



Zadání bakalářské práce

Název:	Inzertní webová aplikace pro řemeslníky
Student:	Peter Večeřa
Vedoucí:	Ing. Jiří Hunka
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Cílem této práce je návrh a realizace inzertní webové aplikace určené pro řemeslníky pro Slovenskou Republiku.

Postupujte v těchto krocích:

1. Vhodnou formou analyzujte požadavky potenciálních budoucích uživatelů - inzerentů a návštěvníků hledajících činnost inzerentů.
2. Na základě analýzy navrhnete budoucí webový systém. Zaměřte se na frontend (včetně uživatelského rozhraní) i na backend (vhodnou volbu technologií backendu apod.).
3. Dle návrhu proveďte minimálně prototypovou implementaci.
4. Nezapomeňte při vývoji na vhodné testování včetně uživatelské testování výsledku.
5. Zhodnoťte použitelnost webové aplikace a navrhnete budoucí rozvoj.



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Bakalárska práca

Inzertní webová aplikace pro řemeslníky

Peter Večeřa

Katedra softwarového inženýrství

Vedúci práce: Ing. Jiří Hunka

12. mája 2021

Pod'akovanie

Rád by som poďakoval vedúcemu bakalárskej práce Ing. Jiřímu Hunkovi za jeho pomoc, rady a nápady. Ako jediný z opýtaných ma bol ochotný viesť počas tvorby bakalárskej práce, za čo mu patrí moje veľké ďakujem.

Ďalej by som chcel úprimne poďakovať svojej rodine za podporu počas celej doby štúdia na vysokej škole.

Zvláštne poďakovanie patrí mojej priateľke za jej povzbudivé slová, trpezlivosť a podporu počas tvorby práce.

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval samostatne a že som uviedol všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Praze 12. mája 2021

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2021 Peter Večeřa. Všetky práva vyhrazené.

Táto práca vznikla ako školské dielo na FIT ČVUT v Prahe. Práca je chránená medzinárodnými predpismi a zmluvami o autorskom práve a právach súvisiacich s autorským právom. Na jej využitie, s výnimkou bezplatných zákonných licencií, je nutný súhlas autora.

Odkaz na túto prácu

Večeřa, Peter. *Inzertní webová aplikace pro řemeslníky*. Bakalárska práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Cieľom práce je vytvorenie webovej aplikácie slúžiacej na inzerciu práce remeselníkov.

Bakalárska práca sa najprv zameriava na analyzovanie požiadaviek budúcich užívateľov a analýzu existujúcich riešení. Práca ďalej pokračuje návrhom systému, v ktorom sú naformulované funkčné a nefunkčné požiadavky, prípady užitia, doménový model a návrh užívateľského rozhrania. Implementačná časť je zameraná na výber jednotlivých technológií, a ich praktické využitie v aplikácii. V rámci implementačnej časti sú zahrnuté ukážky kódu, návrh databázového modelu, ale aj použité princípy a implementačné detaily.

Správny návrh a funkčnosť aplikácie je overená v priebehu testovania, kde je aplikácia podrobená primárne užívateľskému testovaniu. V závere práce sú navrhnuté možné vylepšenia do budúcnosti, ako aj zhodnotenie finálnej verzie aplikácie.

Webová aplikácia umožňuje full textové vyhľadávanie s filtrovaním podľa polohy. Ponúka pestrý výber remesiel, ktorý zjednodušuje a sprehradňuje vyhľadávanie pre užívateľa.

Výsledkom tejto práce je prototypová implementácia inzertného portálu pre remeselníkov, ktorá stavia kvalitné základy pre budúci vývoj.

Kľúčové slová remeselníci, ponuka práce, inzertný portál, webová aplikácia, vývoj aplikácie, Java, Spring, Vaadin

Abstract

The main goal of the thesis is to create a web application to advertise craftsmen's work.

The bachelor thesis firstly focuses on the analysis of future users' requirements and the analysis of existing solutions. The thesis further develops a system design, in which there are formulated functional and nonfunctional requirements, usage cases, domain model and user interface design.

The implementation part centers on the selection of individual technologies and their practical use in the application. Within this part, there are code samples, database model proposal, as well as the principles used and implementation details.

The correct design and functionality of the application is verified primarily by user testing. At the end of the thesis, possible future improvements are suggested and the evaluation of the final version of the application is carried out.

The web application enables full text search with location filtering. It offers a varied selection of crafts, which simplifies and clarifies the search for the user.

The result of the thesis is a prototype implementation of an advertising portal for craftsmen which builds up a quality foundation for future development.

Keywords craftsmen, job offer, advertising portal, web application, application development, Java, Spring, Vaadin

Obsah

Úvod	1
1 Analýza súčasného stavu	3
1.1 Metodológia	3
1.2 Obsah dotazníka	4
1.2.1 Sekcia návštevník	4
1.2.2 Sekcia remeselník	5
1.3 Výsledky dotazníka	6
1.3.1 Sekcia návštevník	6
1.3.2 Sekcia remeselník	9
1.4 Existujúce riešenia	11
1.4.1 Filtrovanie výsledkov	12
1.4.2 Responzívny dizajn	12
1.4.3 Hodnotenie webových stránok pomocou nástroja Light-house	12
1.4.4 Inzercia.sk	13
1.4.5 Bazos.sk	14
1.4.6 Bazar.sk	15
1.4.7 Súhrn	16
2 Návrh systému	19
2.1 Funkčné požiadavky	19
2.1.1 F01 Spravovanie a prehliadanie inzerátov	19
2.1.2 F02 Spravovanie užívateľského účtu	20
2.1.3 F03 Vyhľadávanie a filtrovanie remeselníkov	21
2.1.4 F04 Administrátorská sekcia	21
2.2 Nefunkčné požiadavky	21
2.2.1 Webová aplikácia	21
2.2.2 Responzivnosť	22

2.2.3	Rozširiteľnosť	23
2.3	Prípady užitia	23
2.3.1	Vyhľadávanie a filtrovanie	25
2.3.2	Spravovanie inzerátov	26
2.3.3	Vytvorenie inzerátu	28
2.3.4	Spravovanie účtu	29
2.4	Tabuľka pokrytia	31
2.5	Návrh doménového modelu	32
2.6	Návrh užívateľského rozhrania	33
2.6.1	Domovská stránka	33
2.6.2	Detail inzerátu	35
2.6.3	Kategória a filtrovanie	36
2.6.4	Vytvorenie inzerátu	37
2.6.5	Prihlásenie	38
2.6.6	Registrácia	39
2.6.7	Resetovanie hesla	40
2.6.8	Zmena hesla	41
2.6.9	Moje ponuky	42
2.6.10	Užívatelia - Administrátorská sekcia	43
2.6.11	Inzeráty - Administrátorská sekcia	44
2.7	Súhrn	45
3	Implemetácia funkčného prototypu	47
3.1	Framework	47
3.2	Anotácie	48
3.3	Adresárová štruktúra	49
3.4	Backend	51
3.4.1	Java	51
3.4.2	Spring	52
3.4.3	Spring Boot	52
3.4.4	Spring Security	54
3.4.5	Spring Data	55
3.4.6	Mailsender	57
3.4.7	Gradle	58
3.5	Frontend	60
3.5.1	Vaadin	60
3.5.2	Css	63
3.5.3	Javascript	63
3.6	Databáza	65
3.7	Autentizácia a autorizácia	67
3.8	Dokumentácia	68
3.9	Použité princípy	69
3.10	Súhrn	70

4	Testovanie	71
4.1	Automatické testy	71
4.1.1	Spúšťanie automatických testov	73
4.1.2	Pokrytie kódu	73
4.2	Užívateľské testovanie	74
4.2.1	Scenár A	75
4.2.2	Scenár B	76
4.2.3	Scenár C	77
4.2.4	Scenár D	78
4.2.5	Scenár E – Administrátorská časť	79
4.3	Hodnotenie Lighthouse	80
5	Zhodnotenie	81
5.1	Zhodnotenie aplikácie	81
5.1.1	Použitelnosť výsledného riešenia	82
5.2	Možné vylepšenia	83
5.2.1	Administrátorská časť	83
5.2.2	Zákaznícka podpora	83
5.2.3	Zpracovanie užívateľských údajov	83
5.2.4	Návod aplikácie	83
5.2.5	Správa užívateľského účtu	84
5.2.6	Autentizácia pomocou externého účtu	84
5.2.7	Komunikácia a hodnotenie	84
5.2.8	Prístupnosť	84
5.2.9	Topovanie inzerátu	84
	Záver	85
	Bibliografia	87
	A Zoznam použitých skratiek	91
	B Obsah priloženej SD karty	93

Zoznam obrázkov

1.1	Rozdelenie užívateľov podľa roly, ktorú by na inzertnom portáli zastávali	6
1.2	Rozdelenie užívateľov podľa spôsobu vyhľadávania remeselníka . . .	7
1.3	Rozdelenie remeselníkov podľa využitia inzertných portálov na vlastnú propagáciu	9
1.4	Dizajn hornej lišty pri zariadeniach so šírkou obrazovky ≥ 768 px.	13
1.5	Dizajn hornej lišty pri zariadeniach so šírkou obrazovky < 768 px.	13
1.6	Inzercia.sk výsledky Lighthouse hodnotenia	14
1.7	Bazos.sk výsledky Lighthouse hodnotenia	15
1.8	Dizajn hornej lišty pri zariadeniach so šírkou obrazovky ≥ 992 px.	15
1.9	Dizajn hornej lišty pri zariadeniach so šírkou obrazovky < 992 px.	15
1.10	Výsledky Lighthouse hodnotenia	16
2.1	Podiel internetových prehliadačov na celosvetovom trhu podľa [6]	22
2.2	Podiel internetových prehliadačov na Slovenskom trhu podľa [7]	22
2.3	Štatistiky rozlíšení obrazoviek na celosvetovom trhu pre stolné počítače podľa [8]	22
2.4	Prípady využitia webovej aplikácie	24
2.5	Detailne pokryté prípady využitia vyhľadávania a filtrovania	25
2.6	Detailne pokryté prípady využitia spravovania inzerátov	27
2.7	Detailne pokryté prípady využitia vytvorenia inzerátu	28
2.8	Detailne pokryté prípady využitia spravovania účtu	30
2.9	Doménový model webovej aplikácie	32
2.10	Domovská stránka webovej aplikácie	34
2.11	Detail inzerátu	35
2.12	Obsah kategórie	36
2.13	Obrazovka vytvorenia inzerátu	37
2.14	Prihlasovací formulár	38
2.15	Registračný formulár	39
2.16	Formulár na resetovanie hesla	40

2.17	Formulár na zmenu hesla	41
2.18	Obrazovka vlastnených ponúk užívateľom	42
2.19	Zobrazenie užívateľov v administrátorskej sekcii	43
2.20	Zobrazenie ponúk v administrátorskej sekcii	44
3.1	Adresárová štruktúra aplikácie	50
3.2	20 najpopulárnejších programovacích jazykov (údaje sú získane ku dňu 06. 04. 2021)	52
3.3	Ilustračný obrázok Spring Bootu podľa [14]	53
3.4	Štruktúra Vaadin frameworku podľa [23]	62
3.5	Databázový model webovej aplikácie	66
4.1	Lighthouse hodnotenie vytvorenej webovej aplikácie	80

Zoznam tabuliek

1.1	Úvodný text dotazníka propagovaný na fórach a sociálnych sieťach	3
1.2	Inzertné portály využívané návštevníkmi	7
1.3	Užívateľské odpovede k otázkam 5 až 9 (tučným písmom sú označené funkcie, ktoré dosiahli v priemernom hodnotení aspoň 4/5 bodov).	8
1.4	Inzertné portály využívané remeselníkmi	9
1.5	Užívateľské odpovede k otázkam 5 až 8. (tučným písmom sú označené funkcie, ktoré dosiahli v priemernom hodnotení aspoň 4/5 bodov).	10
1.6	Prehľad inzertných portálov na trhu (údaje o počte inzerátov sú získané z jednotlivých inzertných portálov ku dňu 21. 03. 2021.)	11
1.7	Rozdelenie kategórií podľa šírky obrazovky	12
2.1	Tabuľka pokrytia funkčných požiadaviek	31
4.1	Tabuľka pokrytia backend časti kódu	74
4.2	Scenár A tabuľka splnenia úloh	75
4.3	Scenár B tabuľka splnenia úloh	76
4.4	Scenár C tabuľka splnenia úloh	77
4.5	Scenár D tabuľka splnenia úloh	78
4.6	Scenár E tabuľka splnenia úloh	79

Zoznam blokov kódu

3.1	Trieda zabezpečuje zahešovanie a overenie hesla v aplikácii. . .	55
3.2	Metóda ruší topovanie inzerátov	56
3.3	Metóda využívaná na posielanie správ pomocou e-mailovej pošty	57
3.4	Nastavenia pluginu Spotless pre aplikáciu	59
3.5	Konfigurácia Javadoc pre Gradle	60
3.6	Trieda rozširuje funkcionality Vaadin upload komponentu . . .	61
3.7	CSS nastavenia pre upload komponent	63
3.8	JS funkcia, spúšťajúca udalosť	64
3.9	Vaadin trieda naviazaná na JS skript	64
3.10	Trieda autentizuje užívateľa a konfiguruje prístupové práva k jednotlivým stránkam v aplikácii	67
4.1	Metóda testuje aktivovanie užívateľského účtu	72
4.2	Zdrojový kód konfiguračného súboru kontinuálnej integrácie na GitLabe	73

Úvod

V súčasnej dobe existuje veľa rôznych inzertných portálov, ktoré ponúkajú možnosť inzercie služieb, avšak veľa z užívateľov takýchto portálov o tejto možnosti nevie. Väčšinou je kategória služby rovnocenná s ostatnými kategóriami, avšak jej delenie na jednotlivé typy služieb má nízku granularitu. Inzertné portály vyžadujú častokrát uvedenie ceny, ktoré nie vždy vyhovuje všetkým užívateľom. Príkladom môže byť taký remeselník, ktorý vyčísluje cenu svojej práce vzhľadom na jednotlivé objednávky. Momentálne sa na slovenskom trhu nachádza iba veľmi obmedzený počet inzertných portálov, ktoré sa zameriavajú na inzerciu práce remeselníkov. Webové stránky, ktoré sa na túto činnosť špecializujú sú často zastarané a verejnosti neznáme. Práve v aktuálnom období, kedy je takmer celá spoločnosť odkázaná na prácu z domu, je propagácia činnosti remeselníka výrazne skomplikovaná.

Problém inzercie práce remeselníkov nebol podľa môjho názoru uspokojivo vyriešený, a preto by som sa mu chcel v práci detailnejšie venovať.

Hlavným cieľom práce je vytvoriť webovú aplikáciu na inzerciu práce remeselníkov. Špecifické zameranie aplikácie umožňuje plné prispôbenie užívateľského rozhrania pre tento účel. Výsledná aplikácia si kladie za cieľ lepšiu propagáciu začínajúcich, ale aj skúsených remeselníkom.

Práca je logicky rozčlenená do viacerých kapitol. V kapitole *Analýza súčasného stavu* je realizovaný zber užívateľských požiadaviek na takúto aplikáciu spoločne s analýzou existujúcich riešení. *Návrh systému* sa zameriava na návrh webovej aplikácie počínajúc funkčnými a nefunkčnými požiadavkami, ktoré vznikli primárne na základe predchádzajúcej kapitoly. Ďalej sa v kapitole nachádza návrh prípadov užívania, doménového modelu a užívateľské rozhrania. V kapitole *Implementácia funkčného prototypu* sú detailne uvedené použité technológie, princípy a ich praktické využitie v aplikácii. V poradí štvrtá kapitola *Testovanie* je zameraná na overenie správnej funkčnosti a prehľadnosti aplikácie pre užívateľov. Posledná piata kapitola *Zhodnotenie* hodnotí výslednú aplikáciu, a predstavuje nápady a vylepšenia do budúcnosti.

Analýza súčasného stavu

1.1 Metodológia

Na začiatku zberu užívateľských požiadaviek bolo treba definovať formu zberu. Užívateľský zber má dve roviny, kvalitatívnu a kvantitatívnu. Keďže ide o aplikáciu, ktorá ma fungovať pre širokú verejnosť, tak bola uprednostnená kvantitatívna časť. Je veľa spôsobov ako zbierať užívateľské požiadavky. Jedným z týchto spôsobov je aj dotazník.

Dotazník bol vybraný na realizáciu zberu užívateľských požiadaviek pre inzertný portál z viacerých dôvodov:

- Časová nenáročnosť
- Dá sa vyplniť online
- Jednoduché získanie respondentov

Dotazníky boli aktívne v období od 1. 11. 2020 do 31. 11. 2020 na viacerých diskusných fórach ako aj na sociálnych sieťach. Dotazníky boli rozdelené podľa roly užívateľa, ktorú by na portáli zastával. Dotazníky boli realizované pomocou Google formuláru. Úvodný text dotazníka môžete nájsť v tabuľke 1.1.

<p>Dobrý deň, volám sa Peter Večeľa a študujem na ČVUT v Prahe. Témou mojej bakalárskej práce je inzertná webová aplikácia pre remeselníkov. Pomôžte mi, prosím, vyplnením tohto krátkeho dotazníka. Výsledkom mojej práce by mal byť webový inzertný portál slúžiaci na inzerciu práce remeselníkov. Za vyplnenie dotazníka úprimne ďakujem.</p>

Tabuľka 1.1: Úvodný text dotazníka propagovaný na fórach a sociálnych sieťach

1.2 Obsah dotazníka

1.2.1 Sekcia návštevník

1. Hľadanie remeselníka na inzertnom portáli, je pre mňa:
 - hlavný spôsob, akým hľadám remeselníka
 - jeden zo spôsobov, akým hľadám remeselníka
 - zbytočné, remeselníka hľadám iným spôsobom

2. Na hľadanie remeselníkov využívam:
 - inzercia.sk
 - bazar.sk
 - bazos.sk
 - wilio.sk
 - avizo.sk
 - iné

3. Nevyhovujú mi nasledujúce vlastnosti/funkcie/správanie:

4. U webového portálu by som ocenil(a), aby umožňoval:

Na stupnici

1. zbytočné
2. nevýznamné
3. z časti významné
4. významné
5. veľmi významné

Ohodnoťte, ako podstatné sú pre Vás nasledujúce vlastnosti.

5. Možnosť kontaktovať remeselníka rovno pod inzerátom
6. Možnosť filtrovať remeselníkov na základe rôznych kritérií
7. Možnosť hodnotenia remeselníkov
8. Možnosť sledovania obľúbených remeselníkov
9. Možnosť komunikácie s remeselníkom priamo na stránke

1.2.2 Sekcia remeselník

1. Inzercia práce na inzertnom portáli, je pre mňa:
 - hlavný spôsob, akým zháňam klientov
 - jeden zo spôsobov, akým zháňam klientov
 - zbytočná, klientov zháňam iným spôsobom
2. Na inzerciu práce využívam:
 - inzercia.sk
 - bazar.sk
 - bazos.sk
 - wilio.sk
 - avizo.sk
 - iné
3. Nevyhovujú mi nasledujúce vlastnosti/funkcie/správanie:
4. U webovej aplikácie by som ocenil(a), aby umožňovala:

Na stupnici

1. zbytočné
2. nevýznamné
3. z časti významné
4. významné
5. veľmi významné

Ohodnoďte, ako podstatné sú pre Vás nasledujúce vlastnosti.

5. Možnosť vytvorenia vlastného užívateľského účtu
6. Možnosť zvýrazniť svoj inzerát nad ostatnými
7. Možnosť pridať videovizitku
8. Možnosť komunikácie so zákazníkom priamo na stránke

1.3 Výsledky dotazníka

Na dotazník bolo zaznamenaných celkovo 109 odpovedí v pomere 9:1.

Grafické rozdelenie užívateľov podľa roly môžete nájsť v grafe 1.1.



Obr. 1.1: Rozdelenie užívateľov podľa roly, ktorú by na inzertnom portáli zastávali

1.3.1 Sekcia návštevník

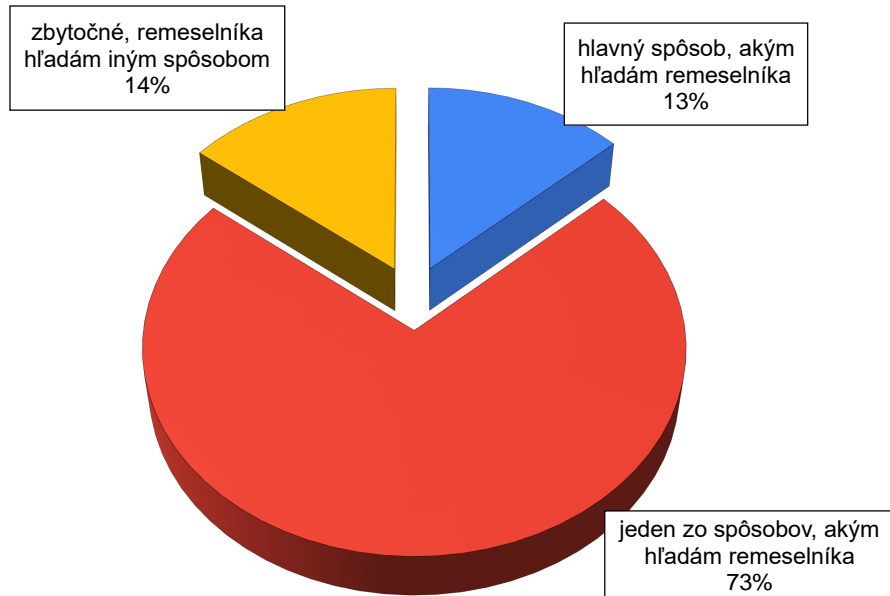
V sekcii návštevník sa podarilo získať až 98 unikátnych odpovedí, čo predstavuje takmer 90 % všetkých odpovedí.

Pre 73 % respondentov je inzertný portál jeden zo spôsobov vyhľadávania remeselníka a pre 13 % opýtaných to predstavuje hlavný spôsob vyhľadávania remeselníka. Iba 14 % z opýtaných nevyužíva inzertný portál na hľadanie remeselníka, a považuje tento spôsob za zbytočný.

Z obrázku 1.2 jasne vyplýva dôležitosť propagácie remeselníckej činnosti v online priestore.

1.3.1.1 Inzercia.sk

Získala druhý najväčší počet hlasov medzi inzertnými portálmi v dotazníku. Užívateľom najviac prekážala neintuitívnosť, ako aj veľa neaktuálnych inzerátov a neaktívnych inzerentov. Návrhy na zlepšenie sa týkali predovšetkým lepšieho rozdelenia remeselníckych povolání, zlepšenia prehľadnosti a zjednodušenia komunikácie.

Hľadanie remeselníka na inzertnom portáli, je pre mňa:

Obr. 1.2: Rozdelenie užívateľov podľa spôsobu vyhľadávania remeselníka

Inzertné portály	Počet hlasov
Inzercia.sk	49
Bazar.sk	29
Bazos.sk	54
Willio.sk	2
Avizo.sk	5
Iné	16

Tabuľka 1.2: Inzertné portály využívané návštevníkmi

1.3.1.2 Bazar.sk

Užívateľom na tomto inzertnom portáli najviac prekážala jeho zbytočná zložitost', ťažké manipulovanie, ale aj občasný spam. Vítané zmeny zahŕňali hlavne jednoduchšie ovládanie, intuitívnejšie prostredie, ale aj detailnejšie rozdelenie remeselníckych povolání do kategórií.

1.3.1.3 Bazos.sk

Napriek faktu, že sa nejde o inzertný portál zameriavajúci sa na inzerciu práce, získal Bazos.sk najvyšší počet hlasov. Respondenti sa sťažovali najmä na zlú vizuálnu stránku a ťažkú orientáciu. Naopak priali by si, aby bol prehľadnejší a ponúkal možnosť filtrovania podľa viacerých kritérií.

1.3.1.4 Willio.sk

Tento inzertný portál získal najmenší počet hlasov spomedzi opýtaných. Dôvodom mohol byť fakt, že sa nejde o klasický inzertný portál. Tento portál funguje na inom princípe. Z dôvodu malej vzorky užívateľov by nevýhody a zlepšenia portálu neboli objektívne, a preto nie sú v texte uvedené.

1.3.1.5 Avizo.sk

Inzertný portál Avizo.sk získal spoločne s Willio.sk najmenší počet hlasov, a preto užívateľmi vnímané nevýhody a navrhované zlepšenia nebudú v texte obsiahnuté.

Na otázku 3. odpovedalo vyše 50 % opýtaných napriek faktu, že bola dobrovoľná, čo môže značiť vysokú mieru nespokojnosti s aktuálnymi možnosťami inzercie práce na trhu. Suverénne najčastejším nedostatkom, ktorý respondenti na inzertnom portáli pociťujú, je jeho neprehľadnosť a neintuitívnosť.

Na otázku 4. odpovedalo opäť viac ako 50 % opýtaných. Neprekvapivo najspomínanejšia pripomienka bola sprehľadnenie a zjednodušenie inzertného portálu, ktorú tvorilo zhruba 60 % z odpovedí.

Funkcia	Priemerné hodnotenie
Kontaktovanie remeselníka rovno pod inzerátom	4.3
Filtrovanie remeselníkov na základe rôznych kritérií	4.6
Hodnotenie remeselníkov	4.3
Sledovanie obľúbených remeselníkov	3.34
Komunikácia s remeselníkom priamo na stránke	3.76

Tabuľka 1.3: Užívateľské odpovede k otázkam 5 až 9 (tučným písmom sú označené funkcie, ktoré dosiahli v priemernom hodnotení aspoň 4/5 bodov).

Filtrovanie remeselníkov dosiahlo v dotazníku najvyššie hodnotenie. Z pozorovania vyplynulo, že ľuďom sa nechce zdĺhavo prechádzať inzertný portál, kým nájdu remeselníka, ktorého potrebujú. Žiadosť po tejto funkcii sa prejavila aj v užívateľských odpovediach v dotazníku, kde respondenti kritizovali neprehľadné a neintuitívne rozdelenie povolaní do kategórií.

Funkcia kontaktovať remeselníka rovno pod inzerátom by nemala určite chýbať na takomto inzertnom portáli. Respondenti túto funkcionality ohodnotili v priemere 4.3 bodu, čo môže signalizovať mieru jej esenciality. Ide o jednoduchú funkciu, ktorá šetrí užívateľom čas.

V neposlednom rade bola zvolená funkcia hodnotenia remeselníka, ktorá so sebou prináša veľkú výhodu pre užívateľa. Užívateľ vďaka tejto funkcii môže filtrovať remeselníkov podľa hodnotení predošlých zákazníkov, a tak zamedziť

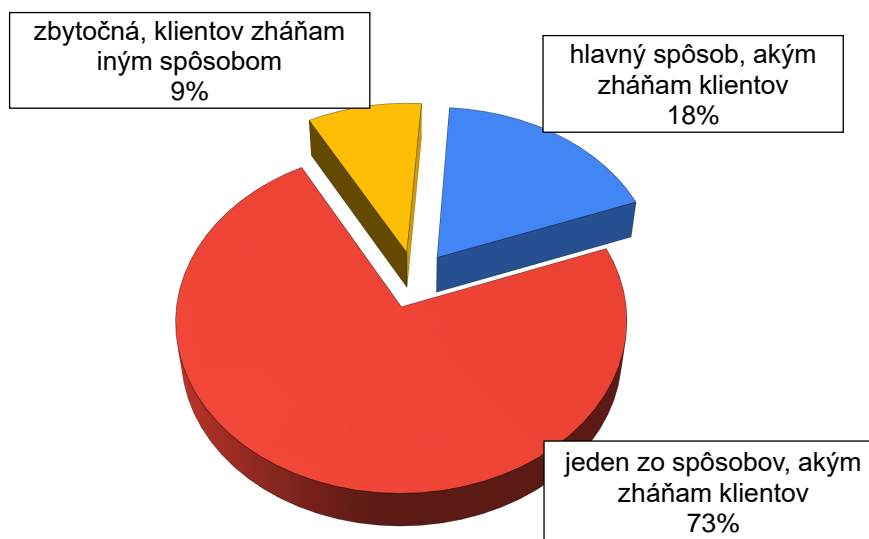
případným nepříjemnostiam v budúcnosti, avšak zber objektívnych recenzií od zákazníkov by bolo veľmi komplikované sprostredkovať, a preto sa v prototypovej implementácii zatiaľ nebude nachádzať.

1.3.2 Sekcia remeselník

V sekcii bolo zaznamenaných, bohužiaľ, iba 11 užívateľských odpovedí. Užívateľská základňa bola v tejto sekcii výrazne menšia ako v predošlej. Odhliadnúc od tohto faktu sa odpovede výrazne nelíšili. Iba 9 % opýtaných (1 respondent) uviedol, že je pre neho inzertný portál zbytočný. Zvyšných 10 respondentov používa inzertný portál na propagáciu svojej činnosti.

Dôležitosť propagácie remeselníckej činnosti na inzertnom portáli je graficky zobrazená na obrázku 1.3.

Inzercia práce na inzertnom portáli, je pre mňa:



Obr. 1.3: Rozdelenie remeselníkov podľa využitia inzertných portálov na vlastnú propagáciu

Inzertné portály	Počet hlasov
Inzercia.sk	3
Bazar.sk	2
Bazos.sk	8
Willio.sk	0
Avizo.sk	0
Iné	2

Tabuľka 1.4: Inzertné portály využívané remeselníkmi

1. ANALÝZA SÚČASNÉHO STAVU

Remeselnícke odpovede na otázku 2. sa výrazne neodlišovali od odpovedí návštevníkov. Z dôvodu malej vzorky respondentov, nie sú zmienené nevyhovujúce vlastnosti, ako aj odporúčania na zlepšenie konkrétnych inzertných portálov, avšak tieto odpovede sú ďalej v texte spoločne zhrnuté.

Na otázku 3. bolo zaznamenaných 7 odpovedí z 11 možných, čo predstavuje vyše 60 % opýtaných. Z dôvodu malej vzorky respondentov sa odpovede výrazne líšili. Boli spomenuté nasledujúce nevyhovujúce vlastnosti:

- neprehľadnosť a chaotickosť
- komplikovanosť a zložité ovládanie
- zložité vyhľadávanie lokality
- automatické odstránenie inzerátov
- deficit mobilnej aplikácie

Na otázku 4. bolo zaznamenaných opäť 7 užívateľských odpovedí. Remeselníci by ocenili nasledujúce vlastnosti:

- intuitívnejšie ovládanie
- prehľadnejšie prostredie
- možnosť vyhľadávať v okruhu podľa adresy

Funkcia	Priemerné hodnotenie
Vytvorenie vlastného užívateľského účtu	3.81
Zvýraznenie svojho inzerátu nad ostatnými	4.27
Pridanie videovizitky	3.56
Komunikácia so zákazníkom priamo na stránke	4

Tabuľka 1.5: Užívateľské odpovede k otázkam 5 až 8. (tučným písmom sú označené funkcie, ktoré dosiahli v priemernom hodnotení aspoň 4/5 bodov).

Zvýraznenie inzerátu, často uvádzane aj ako Topovanie, získalo z uvedených funkcií najlepšie hodnotenie. Za takéto zvýraznenie sa typicky platí na určité časové obdobie. Topovanie prináša výhodu oproti konkurencii vo forme prednostného zobrazenia. Z tabuľky 1.5 možno vidieť záujem remeselníkov o takúto funkciu.

Komunikácia so zákazníkom získala v priemere rovné 4 body. Takáto funkcionalita má hneď viacero výhod. Komunikácia so zákazníkmi je podstatne rýchlejšia ako prostredníctvom e-mailových správ. Taktiež by to bolo prospešné pre inzertný portál z dôvodu väčšej aktivity užívateľov. Na druhej strane by bol inzertný portál komplikovanejší, čo užívatelia výrazne kritizovali.

1.4 Existujúce riešenia

V nasledujúcej časti by som sa rád venoval už existujúcim riešeniam na trhu. Webové inzertné portály stále naberajú na popularite a pomaly nahrádzajú papierovú formu. Väčšina webových inzertných portálov ma podobnú štruktúru. Na hlavnej stránke užívateľ zväčša nájde rozcestník kategórií, vyhľadávacie textové pole, ale aj okolie, v ktorom sa vyhľadávanie má uskutočniť, avšak ani jeden z týchto inzertných portálov sa priamo nešpecifikuje na inzertné služby pre remeselníkov, až na Willio.sk, to však funguje na inom princípe. Z užívateľských odpovedí v dotazníku vyplynulo, že princíp, na ktorom inzertný portál Willio.sk funguje nie je vyhovujúci, a užívatelia sú zvyknutí na klasické inzertné portály. Z tohto dôvodu nie je tento princíp považovaný za vhodný. Na inzertných portáloch sa remeselnícke práce väčšinou nachádzajú v kategórii *Služby*. V tejto kategórii sú remeselnícke práce často delené s nízkou granularitou. V tabuľke 1.6 sú uvedené inzertné portály spomenuté v otázkach v dotazníku obohatené o nasledujúce údaje.

- počet inzerátov
- zameranie
- rok vzniku

Inzertný portál	Počet inzerátov	Zameranie	Rok vzniku
Inzercia.sk	156 000+	Všeobecné	2000
Bazar.sk	786 000+	Všeobecné	2015
Bazos.sk	1 030 515	Všeobecné	2004
Avizo.sk	16 967	Všeobecné	2011
Willio.sk	50 983	Služby	2018

Tabuľka 1.6: Prehľad inzertných portálov na trhu (údaje o počte inzerátov sú získané z jednotlivých inzertných portálov ku dňu 21. 03. 2021.)

Respondeti sa v dotazníkoch sťažovali na zlé vyhľadávanie a filtrovanie remeselníkov, preto je jedna z pozorovaných vlastností na túto funkciu zameraná. Responzívny dizajn je taktiež súčasťou pozorovaných vlastností, keďže nám zabezpečuje konzistentné zobrazenie obsahu stránky na všetkých typoch obrazoviek. V prípade absencie responzívneho dizajnu sa portál stáva pre užívateľa neprehľadný, čo bolo výrazne kritizované respondentami v dotazníku, avšak najdetailnejší a zároveň najobjektívnejší výsledok nám poskytol nástroj Lighthouse, ktorý sledoval viacero vlastností a zároveň hodnotil celkovú kvalitu stránky.

Detailnejší popis sledovaných vlastností nájdete v nasledujúcich podkapitolách. Analyzované budú 3 najčastejšie zvolené riešenia užívateľmi v dotazníku.

1.4.1 Filtrovanie výsledkov

Možnosť filtrovania výsledkov je v dnešnej dobe základom každej dobrej webovej aplikácie. Ide o krátke zhodnotenie možností nastavenia filtra na danom inzertnom portáli. Ide predovšetkým o zameranie sa na filtrovanie podľa polohy a vzdialenosti k nej, ako aj ku špecifickým filtrom jednotlivých kategórií.

1.4.2 Responzívny dizajn

Skúma prispôsobenie inzertných portálov pre všetky druhy zariadení. Zariadenia sú rozdelené podľa šírky obrazovky do 5 kategórií, XS (Extra small), SM (Small), MD (Medium), LG (Large) a XL (Extra large). Jednotlivé kategórie a ich kritéria sú uvedené v tabuľke 1.7. Dôležité je, aby bol obsah stránky prispôsobený pre všetky šírky obrazovky, a zároveň, aby stránka disponovala rovnakou funkcionalitou pre každú kategóriu.

Extra small	Small	Medium	Large	Extra large
< 576 px	≥ 576 px	≥ 768 px	≥ 992 px	≥ 1200 px

Tabuľka 1.7: Rozdelenie kategórií podľa šírky obrazovky

1.4.3 Hodnotenie webových stránok pomocou nástroja Lighthouse

Lighthouse je bezplatný open-source nástroj, ktorý robí detailnú SEO (Search Engine Optimization) analýzu webových stránok. [1]

Tento nástroj som si vybral hlavne kvôli jeho popularite a počtu parametrov, s ktorými pri analýze kalkuluje. Taktiež ponúka možnosť hodnotiť aj stránky, ktoré bežia na lokálnom prostredí, čo sa neskôr využije pri testovaní výslednej aplikácie. Lighthouse hodnotí nasledujúce vlastnosti:

Performance: Rýchlosť načítania webovej stránky, veľkosť súborov, ktoré sa načítajú, ale aj mieru latencie na jednotlivé príkazy.

Accessibility: Skúma ako sa stránka správa na rôznych typoch zariadení s rôznou šírkou obrazoviek.

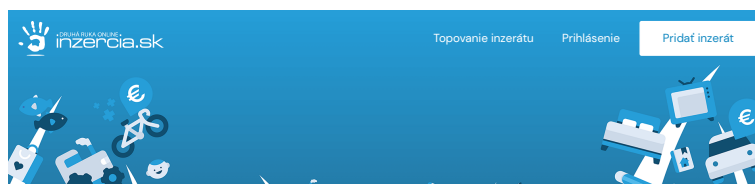
Best practices: Ide o obsah a dizajn webovej stránky, ale aj používanie vhodných formátov.

SEO: Dôležitá vlastnosť stránky, ktorá uľahčuje prehliadačom porozumieť obsahu a hodnotiť na základe kľúčových slov.

1.4.4 Inzercia.sk

1.4.4.1 Responzívny dizajn

Dizajn inzertného webového portálu Inzercia.sk dostupného z [2] bol responzívny, naprieč všetkými kategóriami. Takmer každá kategória mala rovnakú funkcionálnosť až na kategóriu XS. Na najmenšom rozlíšení, ktoré je určené najmä pre smartfóny, bol pozorovaný jeden rozdiel. Z pravej hornej časti obrazovky zmizlo políčko *Topovanie inzerátu*.



Obr. 1.4: Dizajn hornej lišty pri zariadeniach so šírkou obrazovky ≥ 768 px.



Obr. 1.5: Dizajn hornej lišty pri zariadeniach so šírkou obrazovky < 768 px.

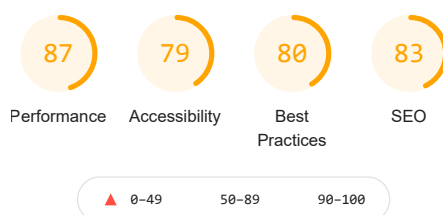
1.4.4.2 Filtrovanie výsledkov

Inzercia.sk nemá špecifické filtre pre jednotlivé kategórie. Umožňuje však full textové vyhľadávanie, ako aj vyhľadávanie pomocou značiek. Užívateľ môže filtrovať ponuky podľa polohy. Inzeráty je možné radieť podľa ceny alebo dátumu ich pridania.

1.4.4.3 Lighthouse hodnotenie

V podrobnej SEO analýze dopadol inzertný portál najhoršie v porovnaní s konkurenciou. Najhoršie hodnotenie získal v časti Accessibility, a naopak najlepšie v Performance. Lighthouse uviedol viacero sémantických chýb, ktoré sa na stránke nachádzali ako napr.: tlačidlá nemajú prístupné pomenovanie, obrázky nemajú tzv. alt atribút, odkazy nemajú rozoznateľné mená.

Stručné Lighthouse hodnotenie môžete vidieť na obrázku 1.6.



Obr. 1.6: Inzercia.sk výsledky Lighthouse hodnotenia

1.4.5 Bazos.sk

1.4.5.1 Responzívny dizajn

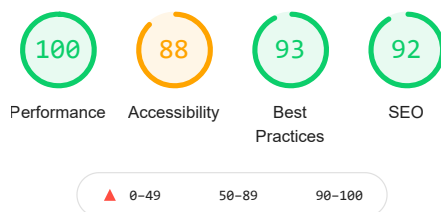
Bazos.sk dostupný z [3] ponúka plne responzívny dizajn, avšak dalo by sa mu vyčítať zobrazenie hlavnej lišty pre smartfóny. Mohol by disponovať lepším prispôbením pre tento typ zariadení, vzhľadom na fakt, že ide o najväčší inzertný portál na Slovensku. Funkcionalita je konzistentná a zachovaná naprieč všetkými kategóriami. Ďalej je potrebné spomenúť, že v prípade použitia na obrazovkách s nižším rozlíšením sú tlačidlá v hornej lište nevýrazné.

1.4.5.2 Filtrovanie výsledkov

Podobne ako v predchádzajúcom prípade nedisponuje ani tento portál špeciickými filtrami pre jednotlivé kategórie. Portál ponúka full textové vyhľadávanie. Na vyhľadávanie podľa adresy môže užívateľ zvoliť názov obce, alebo jej PSČ (Poštové Smerovacie Číslo). Filtrovanie výsledkov sa dá taktiež uskutočniť podľa ceny, pričom je možnosť zadať interval, v ktorom sa má cena pohybovať, alebo podľa počtu zobrazení daného inzerátu.

1.4.5.3 Lighthouse hodnotenie

Inzertný portál má veľmi dobrú optimalizáciu, a preto získal v kategórii Performance najvyšší možný počet bodov. Stránka má kvôli svojmu jednoduchému dizajnu veľmi rýchlu odozvu. Najhoršie hodnotenie získal v kategórii Accessibility. Napriek konzistentnej funkcionalite na všetkých typoch zariadení je toto hodnotenie opodstatnené. Zlá viditeľnosť tlačidiel pri zariadeniach s obrazovkami XS už bola spomenutá v časti 1.4.5.1. Ako príklad vylepšenia uvádza Lighthouse pridanie klávesových skratiek na uľahčenie ovládania. Stručné Lighthouse hodnotenie môžete vidieť na obrázku 1.7.

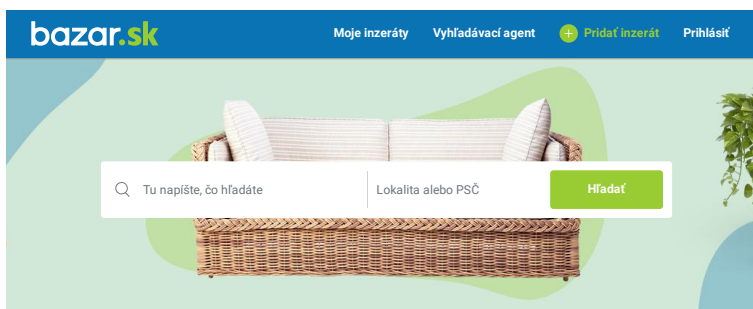
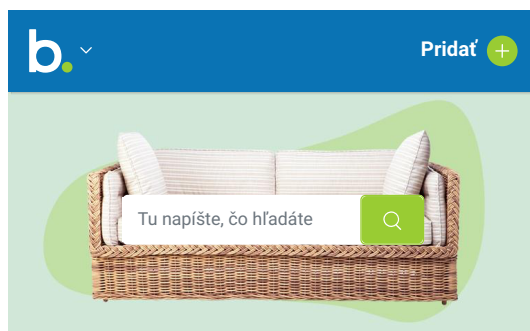


Obr. 1.7: Bazos.sk výsledky Lighthouse hodnotenia

1.4.6 Bazar.sk

1.4.6.1 Responzívny dizajn

Responzívny dizajn nechýbal ani v prípade inzertného portálu Bazar.sk dostupného z [4]. Pre obrazovky šírky XS, SM, MD mal inzertný portál odlišnú hornú lištu, ktorá obsahovala iné funkcionality, ako pre obrazovky, ktoré sa radia do kategórií LG a XL. Na obrázku 1.8 môžete vidieť hornú lištu pre obrazovky so šírkou obrazovky ≥ 992 px. Na porovnanie uvádzam obrázok 1.9, kde môžete sledovať správanie hornej lišty pre obrazovky < 962 px.

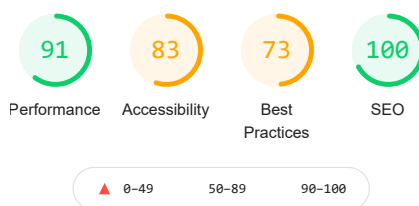
Obr. 1.8: Dizajn hornej lišty pri zariadeniach so šírkou obrazovky ≥ 992 px.Obr. 1.9: Dizajn hornej lišty pri zariadeniach so šírkou obrazovky < 992 px.

1.4.6.2 Filtrovanie výsledkov

Spôsob filtrovania výsledkov je veľmi podobný ako u predchádzajúcich riešení. Inzertný portál ponúka full textové vyhľadávanie. Zároveň ponúka filtrovanie podľa geolokačných údajov. Je možné triedenie podľa ceny, ale aj dátumu vytvorenia inzerátu. Zobrazenie iba ponúk s fotografiou je funkcionalita, ktorou jeho predchodcovia nedisponovali. Navzdory obohateniu o túto funkcionalitu, pretrváva absencia špecifických filtrov pre jednotlivé kategórie.

1.4.6.3 Lighthouse hodnotenie

Bazar.sk získal v kategórii SEO plný počet bodov. Stránka je výborne optimalizovaná pre vyhľadávače, čo má za následok jeho výborne hodnotenie. Rýchlosť načítania je síce v porovnaní s Bazos.sk horšia, avšak inzertný portál je oveľa prepracovanejší, čo má za následok trochu horšie ohodnotenie v kategórii Performance. Najhoršie skóre bolo udelené v kategórii Best Practices. Lighthouse objavil použitie JavaScript knižníc, ktoré majú preukázané bezpečnostné problémy. Stručné Lighthouse hodnotenie môžete nájsť na obrázku 1.10.



Obr. 1.10: Výsledky Lighthouse hodnotenia

1.4.7 Súhrn

Všetky webové inzertné portály majú plne responzívny dizajn. Niektoré webové stránky nie sú pre mobilné zariadenia správne optimalizované a užívateľovi môžu prísť neprehľadné.

Taktiež bola vyzozorovaná zmena funkcionality hornej lišty v závislosti od šírky obrazovky u Inzerca.sk a Bazar.sk.

V kategórii filtrovania výsledkov ani jeden z testovaných subjektov nedisponoval možnosťou špecifického filtrovania vzhľadom na kategóriu. Možnosti nastavenia filtra boli v rámci testovania veľmi podobné.

V Lighthouse hodnotení obstál najlepšie inzertný portál Bazos.sk, ktorého dominantnou stránkou bola rýchlosť. Naopak najhoršie sa umiestnila Inzerca.sk, ktorá mala síce konzistentné, ale nižšie skóre, naprieč všetkými kategóriami. Bazar.sk dominoval v kategórii SEO, v ktorej získal maximálny možný

počet bodov, avšak slabé hodnotenie v kategórii Best Practices sa negatívne podpísalo na jeho celkovom hodnotení.

Získané poznatky z kapitoly 1.4 budú ďalej prakticky využité pri návrhu systému v kapitole 2 a implementácii funkčného prototypu v kapitole 3.

Návrh systému

Nasledujúca kapitola sa bude zaoberať návrhom webovej aplikácie. Návrh bude podmienený doposiaľ analyzovanými užívateľskými požiadavkami v kapitole 1. Taktiež budú zohľadnené výsledky dotazníkov, ale aj získane poznatky z analýzy existujúcich riešení v kapitole 1.4. Ďalej budú navrhnuté funkčné a nefunkčné požiadavky, prípady použitia, doménový model a návrh užívateľského rozhrania.

2.1 Funkčné požiadavky

Cielom tejto bakalárskej práce je vytvoriť inzertnú webovú aplikáciu pre remeselníkov. Na korektný postup je vhodné identifikovať funkčné požiadavky na takúto aplikáciu. Funkčné požiadavky zahŕňujú funkcionality, ktorou by mala aplikácia disponovať, taktiež ich zanedbanie je častou príčinou zlyhania celého softwarového projektu. Na základe poznatkov získaných v predchádzajúcej kapitole, sú ďalej v texte zhrnuté funkčné požiadavky na aplikáciu.

2.1.1 F01 Spravovanie a prehliadanie inzerátov

Úplne najdôležitejšia časť celého systému záleží na inzerátoch od remeselníkov. Webová aplikácia musí obsahovať funkcionality pridania a možného editovania inzerátu. Inzerát by mal obsahovať krátky nadpis, obsah, e-mail, meno, platnú adresu a nepovinne môže remeselník uviesť telefónne číslo.

Na zadanie adresy sa využíva Google places API (Application Programming Interface) podľa [5], ktoré overí integritu zadanej adresy a predá cenné informácie systému akými sú napr. GPS (Global Positioning System) súradnice podľa, ktorých vie systém neskôr efektívne vyhľadávať a filtrovať inzeráty.

E-mailová adresa ako aj meno budú predvyplnené hodnotami zadanými pri registrácii, avšak bude možná ich úprava.

Nahratie fotografií by mal systém umožniť, pokiaľ o takúto možnosť prejaví remeselník záujem.

Počas celého procesu sú zadávané údaje validované a v prípade chybného vstupu je užívateľ systémom informovaný.

Topovanie ponuky remeselníkom je funkcionalita, ktorá sa vzhľadom na jej komplexnosť v prototypovom riešení aplikácie nachádza iba v obmedzenej forme. Navzdory jej obmedzenej funkčnosti je pre systém považovaná za dôležitú, a preto by nemala chýbať jej kompletná funkčnosť pred nasadením systému do produkčného prostredia.

Kontaktovanie remeselníka pomocou kontaktného formulára pod inzerátom bude umožnené všetkým návštevníkom. Návštevník bude musieť uviesť svoju e-mailovú adresu a text, ktorým chce remeselníka osloviť. V prípade, že takúto funkciu bude využívať už prihlásený užívateľ bude jeho e-mailová adresa vo formulári vopred vyplnená.

2.1.2 F02 Spravovanie užívateľského účtu

Neprihlásení užívateľa budú mať k dispozícii takmer kompletnú funkcionalitu webovej aplikácie. Na pridanie, úpravu a topovanie inzerátu bude však potrebné autentifikovanie.

Registrácia bude prvým krokom k získaniu užívateľského účtu na inzertnom portáli. Takáto registrácia bude obsahovať užívateľský e-mail, meno, heslo a jeho opätovné potvrdenie, ktoré by malo zamedziť možným chybám pri prvotnom zadaní. Užívateľove vstupy sú počas celej fázy registrácie monitorované, a v prípade nesprávneho vstupu, je užívateľ na chybu okamžite upozornený.

Po úspešnom vyplnení všetkých údajov bude užívateľovi odoslaný aktivačný kód na uvedenú e-mailovú adresu. Bez aktivácie účtu sa nebude môcť užívateľ na svoj účet prihlásiť, a preto je to nevyhnutný krok k úspešnej registrácii. V momentálnej situácii nie je možné prihlásenie pomocou Googlu, Facebooku a iných systémov poskytujúcich autentifikačné služby. Vyššie spomenutí autentifikační poskytovatelia sa stávajú bežnou súčasťou webových aplikácií, a preto je integrácia funkcie do aplikácie v budúcnosti vítaná.

Proces prihlásenia vyžaduje id užívateľa e-mail, pod ktorým sa zaregistroval, a jeho užívateľské heslo. V rámci prihlásenia je požadované zadanie korektných údajov užívateľom, na všetky možné nesprávne vstupy je užívateľ oboznámený pomocou notifikácie zobrazenej v okne webovej aplikácie.

V prípade, že užívateľ zabudne heslo, bude mať možnosť si heslo resetovať. Funkcia resetovania hesla bude spočívať vo vyplnení užívateľského e-mailu, na ktorý bude odoslané nanovo vygenerované heslo.

Po prihlásení bude užívateľovi ponúknutá možnosť zmeniť vygenerované heslo. Táto možnosť bude taktiež integrovaná v systéme, a bude prístupná všetkým prihláseným užívateľom.

2.1.3 F03 Vyhľadávanie a filtrovanie remeselníkov

Súčasťou riešenia aplikácie bude full textové vyhľadávania remesiel. Takéto vyhľadávanie by malo byť *case insensitive* a *accent insensitive* pre zaistenie čo najlepšej zhody.

Filtrovanie bude umožnené spočiatku podľa kategórie, adresy a vzdialenosti k nej, počtu pozretí a dátumu pridania inzerátu.

Inzeráty sa budú natívne zobrazovať od najnovšieho po najstarší. Počítanie pozretí bude realizované pomocou tzv. session id, kvôli ktorému sa obmedzí umelé navyšovanie pozretí. Session id je unikátne číslo, ktoré systém priradí užívateľovi pri vstupe na stránku. Toto číslo je platné počas celej návštevy užívateľa.

2.1.4 F04 Administrátorská sekcia

V administrátorskej sekcii bude spočiatku implementovaná iba nevyhnutná funkcionálna pre zabezpečenie plynulého fungovania webovej aplikácie. Administrátori budú mať rovnaké prístupové práva na všetky obrazovky ako bežný užívateľ.

Administrátorská sekcia bude obnášať možnosť odstránenia užívateľskej ponuky, ale aj jej topovanie. Toto topovanie bude spočiatku robiť administrátor na požiadanie, a bude trvať 14 dní. Po uplynutí tejto doby sa topovanie inzerátu automaticky zruší.

Administrátor bude mať navyše možnosť zablokovať užívateľský účet pri porušení podmienok webovej aplikácie. Pri zablokovaní užívateľského účtu dôjde taktiež k automatickému odstráneniu všetkých užívateľových inzerátov.

Zmena užívateľskej roly sa bude spočiatku odohrávať taktiež v administrátorskej sekcii.

2.2 Nefunkčné požiadavky

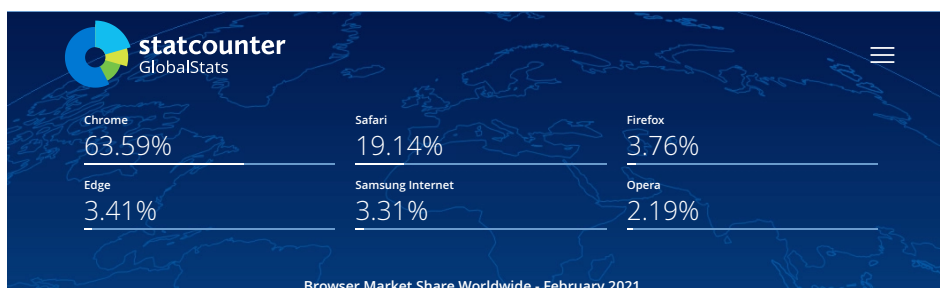
Nefunkčné požiadavky sú doplnkom k funkčným požiadavkám. Takéto požiadavky môžu napr. zahŕňať dostupnosť aplikácie. Nefunkčné požiadavky na webovú aplikáciu sú zhrnuté v nasledujúcich podkapitolách.

2.2.1 Webová aplikácia

Webová aplikácia bude prístupná prostredníctvom internetového prehliadača. U prototypovej implementácie bude zaručená minimálne podpora internetového prehliadača Google Chrome (verzia 89 a vyššie). Google Chrome som si vybral, kvôli jeho veľkej celosvetovej popularite. Podľa [6] má práve Google Chrome viac ako 63 % podiel na celosvetovom trhu. Tieto údaje sú graficky zobrazené na obrázku 2.1. Na Slovensku, ktoré predstavuje cieľový trh tejto webovej aplikácie je podiel Google Chrome na trhu viac ako 68 % podľa [7].

2. NÁVRH SYSTÉMU

Prehľad najpoužívanejších internetových prehliadačov na Slovensku je zobrazený na obrázku 2.2.



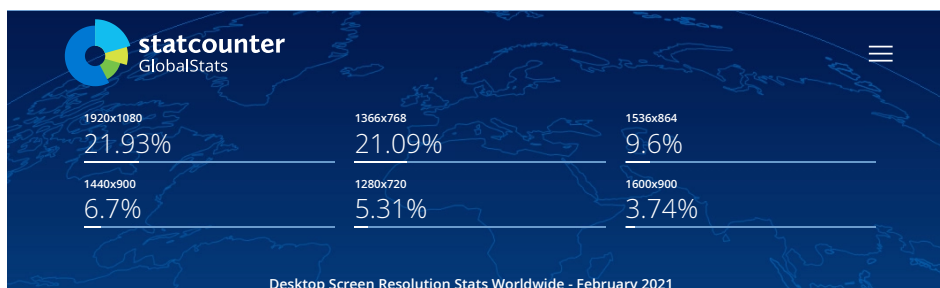
Obr. 2.1: Podiel internetových prehliadačov na celosvetovom trhu podľa [6]



Obr. 2.2: Podiel internetových prehliadačov na Slovenskom trhu podľa [7]

2.2.2 Responzivnosť

Webová aplikácia bude cieľiť na rozlíšenie obrazovky 1920x1080 pixelov, ktoré je zároveň najpoužívanejšie rozlíšenie pre stolné počítače podľa [8], avšak jej optimalizácia bude garantovaná naprieč všetkými typmi obrazoviek. Webová aplikácia bude na každej šírke obrazovky reponzivná a pre užívateľa prehľadná. Prehľad najpoužívanejších rozlíšení je zobrazený na obrázku 2.3.



Obr. 2.3: Štatistiky rozlíšení obrazoviek na celosvetovom trhu pre stolné počítače podľa [8]

2.2.3 Rozšíriteľnosť

Webová aplikácia je navrhnutá tak, aby bola v budúcnosti ľahko rozšíriteľná. Počas implementácie bude autor postupovať podľa osvedčených programátorských praktík. Dodržovanie tzv. best practices je dôležitou súčasťou vývoja. Medzi spomínané praktiky patria napr.:

- menné konvencie
- DRY (Don't Repeat Yourself) - neopakovať svoj vlastný kód
- KISS (Keep It Simple Stupid) - písať čo najjednoduchší kód

a veľa ďalších, ktoré sú ďalej spomenuté v kapitole 3.9. Vývoj inzertného portálu je časovo náročný, a preto sa na konci implementácie počíta s funkčným prototypom inzertného portálu, ktorý bude ľahko rozšíriteľný.

2.3 Prípady užitia

V tejto sekcii sa nachádzajú prípady užitia webovej aplikácie pre inzerciu práce remeselníkov, ktoré boli navrhnuté na základe prechádzajúcej analýzy užívateľských požiadaviek v kapitole 1 a funkčných požiadaviek v kapitole 2.1. Na modelovanie prípadov užitia bol použitý nástroj EA (Enterprise Architect). Autor bol počas štúdia na vysokej škole s týmto nástrojom dobre oboznámený, čo malo výrazný vplyv pri rozhodovaní, aký nástroj použiť.

Na obrázku 2.4 sú zobrazené prípady užitia webovej aplikácie, z ktorých sú niektoré časti ďalej podrobnejšie rozobraté v nasledujúcich podkapitolách.

- **UC01 Prihlásenie**

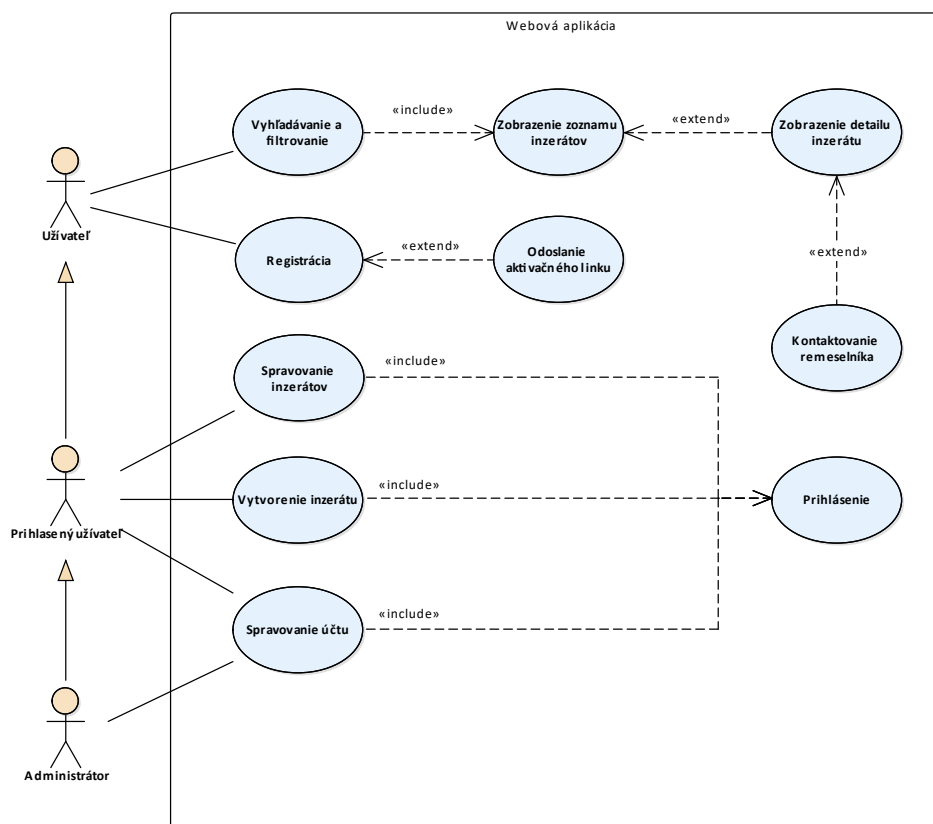
Užívateľovi sa na začiatku procesu zobrazí prihlasovací formulár. Počas celého priebehu je kontrolovaný užívateľský vstup, ktorý v prípade chyby užívateľa adekvátne upozorní. Po vyplnení formulára, systém autentifikuje užívateľa. Autorizácia k špecifickým funkcionalitám systému je exekúovaná pri autentifikácii užívateľa.

- **UC02 Registrácia**

Podobne ako pri prihlásení je užívateľovi ponúknutý online formulár, v ktorom sa od užívateľa požadujú nasledujúce údaje.

- meno
- e-mailová adresa
- heslo
- potvrdenie hesla

2. NÁVRH SYSTÉMU



Obr. 2.4: Prípady použitia webovej aplikácie

V prípade akejkoľvek chyby by mal byť užívateľ náležite upozornený. Po úspešnom vyplnení formulára je užívateľ registrovaný v systéme. Pri každej registrácii je užívateľský profil vytvorený so statusom deaktivovaný. Táto funkcia ponúka jednoduchý spôsob, ako overiť e-mailovú adresu a predísť neplatným registráciám.

- **UC03 Odoslanie aktivačného linku**

Po korektnom vyplnení registračného formulára je na registrovaný užívateľský profil odoslaný aktivačný link, ktorý slúži na overenie totožnosti užívateľa.

- **UC04 Kontaktovanie remeselníka**

Funkcia kontaktovanie remeselníka sa nachádza pri každom detaile inzerátu. Po stlačení tlačidla kontaktovať sa užívateľovi zobrazí kontaktný formulár. Formulár sa uchádza o čo najjednoduchšiu štruktúru, preto bude užívateľovi stačiť iba e-mailová adresa a text adresovaný remeselníkovi.

- **UC05 Zobrazenie detailu inzerátu**

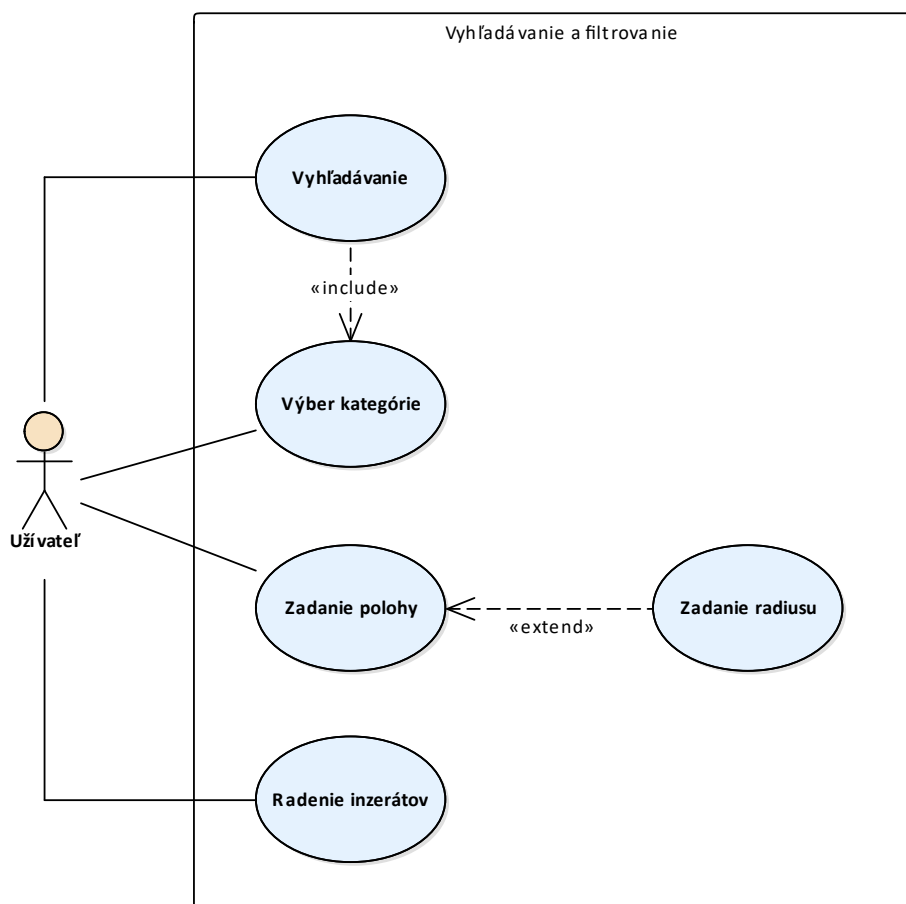
Detail inzerátu bude zahŕňať krátky a výstižný nadpis, opis ponúkanej služby, adresu, fotografie, e-mail, telefónne číslo a tlačidlo na kontaktovanie remeselníka. Na detail inzerátu sa bude užívateľ môcť dostať po kliknutí na fotografiu alebo nadpis inzerátu.

- **UC06 Zobrazenie zoznamu inzerátov**

Zoznam inzerátov bude užívateľovi zobrazený po vstupe na inzertný portál v obmedzenom počte. Po výbere kategórie, alebo kliknutí na ďalšiu stránku bude zoznam inzerátov patrične rozšírený.

2.3.1 Vyhľadávanie a filtrovanie

Na obrázku 2.5 sú podrobne rozobraté prípady užitia týkajúce sa vyhľadávania a filtrovania na inzertnom portáli.



Obr. 2.5: Detailne pokryté prípady užitia vyhľadávania a filtrovania

- **UC07 Vyhľadávanie**

Webová aplikácia bude podporovať full textové vyhľadávanie. Užívateľ zadá hľadaný výraz do vyhľadávacieho poľa, ktoré mu automaticky ponúkne možnosti s najvyššou zhodou.

- **UC08 Výber kategórie**

Užívateľ bude môcť vybrať kategóriu viacerými spôsobmi. Kategórie budú vyhľadateľné pomocou full textového vyhľadávania. Ďalšia možnosť bude výber kategórie zo zoznamu kategórií na úvodnej stránke webovej aplikácie. Posledná z možností, ktorú systém ponúka je prístup na kategóriu pomocou rozkliknutia linku v detaile inzerátu.

- **UC09 Zadanie polohy**

Užívateľ zadá adresu, v ktorej by chcel vyhľadať remeselníkov. Systém mu počas vyhľadávania bude ponúkať zhody s jeho adresou.

- **UC10 Zadanie rádiusu**

Táto funkcionálna časť systému sa užívateľovi zobrazí, až po zadaní adresy. Užívateľ bude môcť nastavovať rádius v okruhu adresy, ktorú zadal. Systém umožní zadanie rádiusu v predefinovaných hodnotách.

- **UC11 Radenie inzerátov**

Užívateľovi bude umožnené radenie inzerátov v zozname spočiatku podľa týchto kritérií.

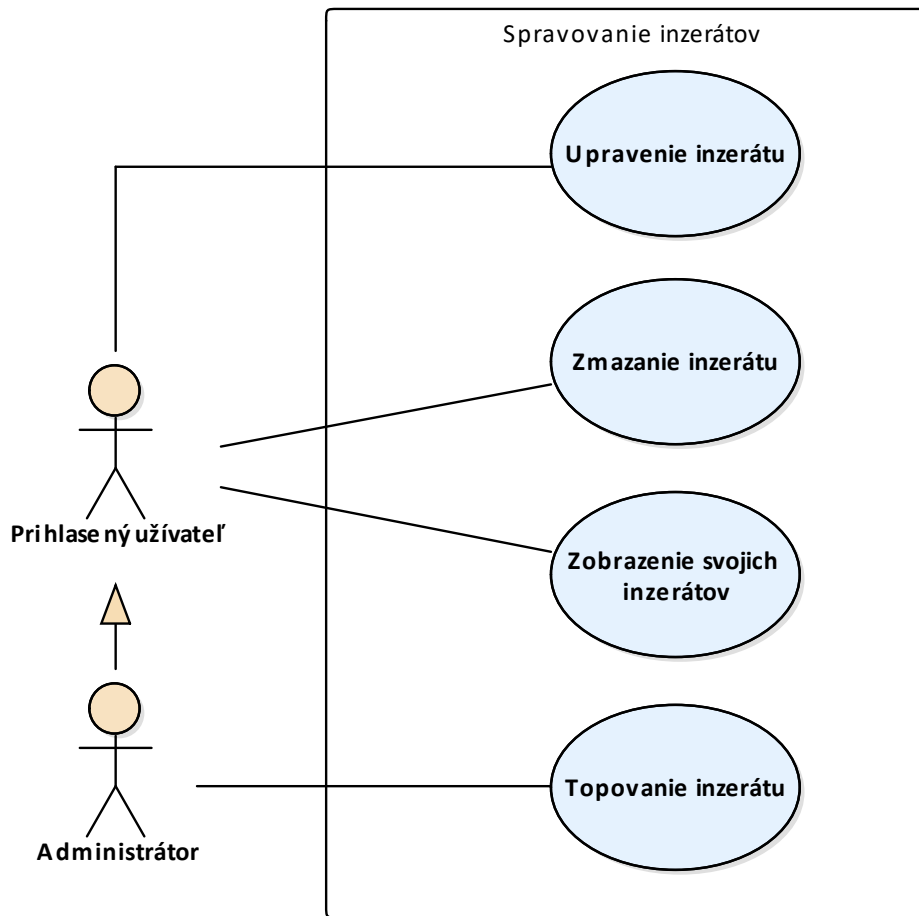
- od najnovších
- od najstarších
- od najpopulárnejších - radenie podľa počtu zobrazení inzerátu

2.3.2 Spravovanie inzerátov

Na obrázku 2.6 sú podrobné rozobraté prípady využitia týkajúce sa spravovania inzerátov na inzertnom portáli.

- **UC12 Upravenie inzerátu**

Pri upravení inzerátu bude inzerát zobrazený užívateľovi vo forme, v ktorej sa nachádzal pri pridaní. Užívateľ bude schopný zmeniť každý atribút inzerátu, ktorý mohol pri vytvorení vyplniť. Po otvorení detailu inzerátu bude možnosť inzerát upraviť. Táto možnosť sa užívateľovi zobrazí iba v prípade, že je vlastníkom takéhoto inzerátu. Administrátor bude môcť upraviť ľubovoľný inzerát.



Obr. 2.6: Detailne pokryté prípady užitia spravovania inzerátov

- **UC13 Zmazanie inzerátu**

Funkcia odstránenia inzerátu bude prístupná po rozkliknutí detailu inzerátu. Užívateľ bude môcť zmazať iba inzerát, ktorý sám vytvoril. Administrátor webovej aplikácie bude mať možnosť vymazať akýkoľvek inzerát zo systému.

- **UC14 Zobrazenie svojich inzerátov**

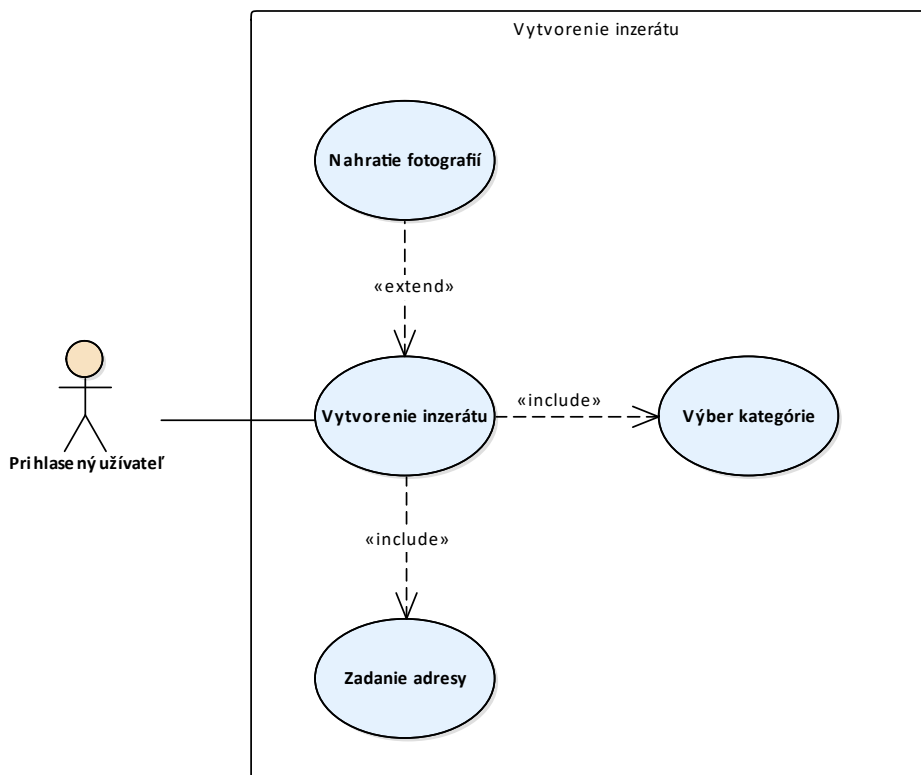
Systém bude ponúkať prehľadné zobrazenie vlastnených inzerátov pomocou tlačidla Moje ponuky.

- **UC15 Topovanie inzerátu**

Administrátor bude môcť topovať ľubovoľnú ponuku manuálne, avšak užívateľ nebude schopný topovať svoj vlastný inzerát. Topovanie ponuky bude aktívne po dobu 14 dní.

2.3.3 Vytvorenie inzerátu

Na obrázku 2.7 sú podrobne rozobraté prípady použitia týkajúce sa vytvorenia inzerátu na inzertnom portáli.



Obr. 2.7: Detailne pokryté prípady použitia vytvorenia inzerátu

- **UC16 Vytvorenie inzerátu**

Funkcionalita, na ktorej bude stavať celý systém musí byť prehľadná, intuitívna a jednoduchá, aby neodradila potencionálnych inzerentov. V pravej hornej lište sa bude nachádzať tlačidlo Pridať ponuku, ktoré automaticky presmeruje užívateľa na online formulár. V prípade, že užívateľ nie je prihlásený, bude najskôr o autentifikáciu požiadany a následne presmerovaný. Vytvorenie inzerátu bude pozostávať zo šiestich povinných a dvoch nepovinných údajov. Po korektnom vyplnení formulára bude inzerát automaticky nahratý do systému a zobrazený medzi najnovšími inzerátmi na hlavnej stránke.

- **UC17 Nahratie fotografií**

Funkcia bude ponúkať možnosť užívateľovi nahrať fotografie zo svojho počítača do systému. Nahratie fotografie nebude povinný krok počas

vypĺňania online formulára, avšak v prípade, že užívateľ nenahrá ani jednu fotografiu, bude použitá základná fotografia. Užívateľ bude môcť nahrať maximálne 3 fotografie.

- **UC18 Výber kategórie**

Ide o povinný krok k úspešnému vytvoreniu inzerátu. Na výber bude vyše 50 kategórií. Kategóriu môže užívateľ zvoliť zo zoznamu, ktorý sa mu pri kliknutí na výber kategórie otvorí, alebo použiť full textové vyhľadávanie. V aktuálnej verzii bude systém podporovať zvolenie maximálne jednej kategórie.

- **UC19 Zadanie adresy**

Jednou zo šiestich povinných funkcií je aj zadanie adresy. Počas užívateľského vstupu, budú užívateľovi ponúkané čo najpresnejšie zhody hľadanej adresy. Po úspešnom zadaní adresy budú z externého zdroja stiahnuté GPS súradnice do systému.

2.3.4 Spravovanie účtu

Na obrázku 2.7 sú podrobné rozobraté prípady užitia týkajúce sa spravovania účtu na inzertnom portáli.

- **UC20 Zmena hesla**

Funkcia bude voľne dostupná všetkým prihláseným užívateľom webovej aplikácie. V online formulári, ktorý bude užívateľovi ponúknutý pri kliknutí na tlačidlo Zmeniť heslo bude otvorený formulár s týmito textovými poľami: staré heslo, nové heslo a zopakovať nové heslo.

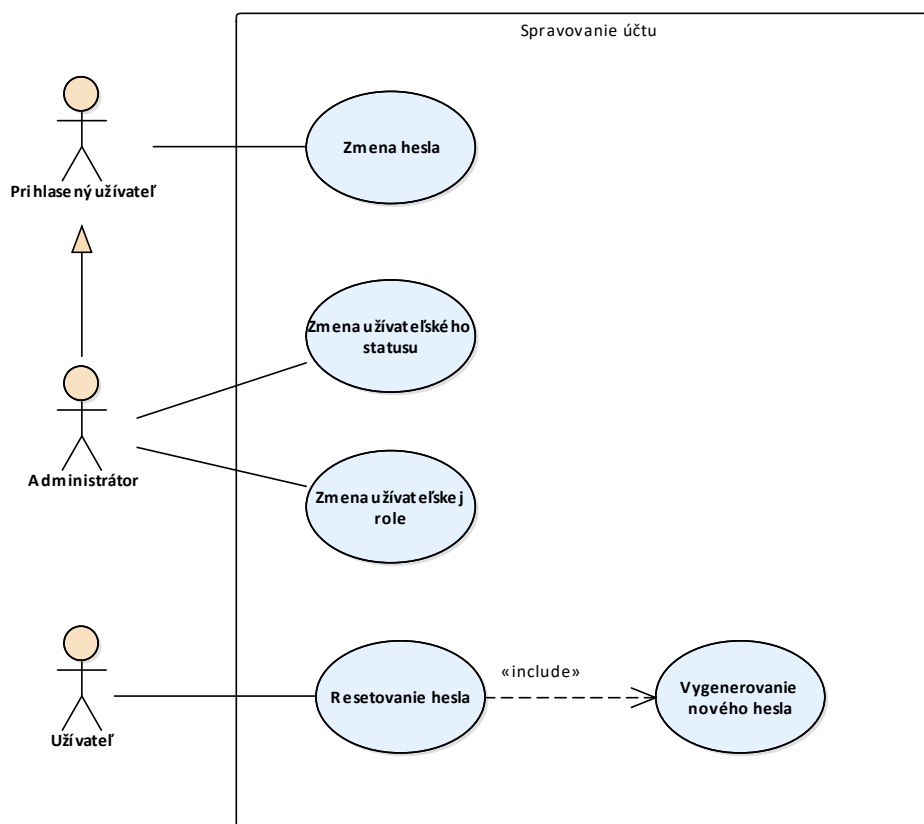
Po zmene hesla zostáva užívateľ naďalej prihlásený.

- **UC21 Zmena užívateľského statusu**

Touto funkciou bude disponovať iba administrátor. Spočiatku budú v systéme registrované tieto užívateľské statusy:

- aktívovaný
- deaktivovaný
- zablokovaný

Deaktivovaný status bude mať každý užívateľ, ktorý neaktivoval svoj profil pomocou zaslaného aktivačného linku. V prípade porušenia pravidiel webovej aplikácie bude môcť administrátor zablokovať užívateľský účet. Pri zablokovaní užívateľského účtu dochádza k automatickému odstráneniu všetkých ponúk vlastnených týmto profilom.



Obr. 2.8: Detailne pokryté prípady užitia spravovania účtu

- **UC22 Zmena užívateľskej role**

System bude evidovať dve užívateľské roly:

- administrátor
- zákazník

Na základe príslušnej roly budú užívateľovi v systéme nastavené prístupové práva do rôznych častí webovej aplikácie.

- **UC23 Resetovanie hesla**

Užívateľ v prípade zabudnutia hesla bude môcť požiadať o jeho resetovanie prostredníctvom formulára, ktorý bude pozostávať z e-mailovej adresy. V prípade, že je adresa systémom evidovaná bude na túto adresu zaslané nanovo vygenerované heslo.

- **UC24 Vygenerovanie nového hesla**

Generácia hesla je exekúovaná po úspešnom resetovaní hesla pomocou online formulára. Heslo je desať znakov dlhé a vygenerované náhodne.

2.4 Tabuľka pokrytia

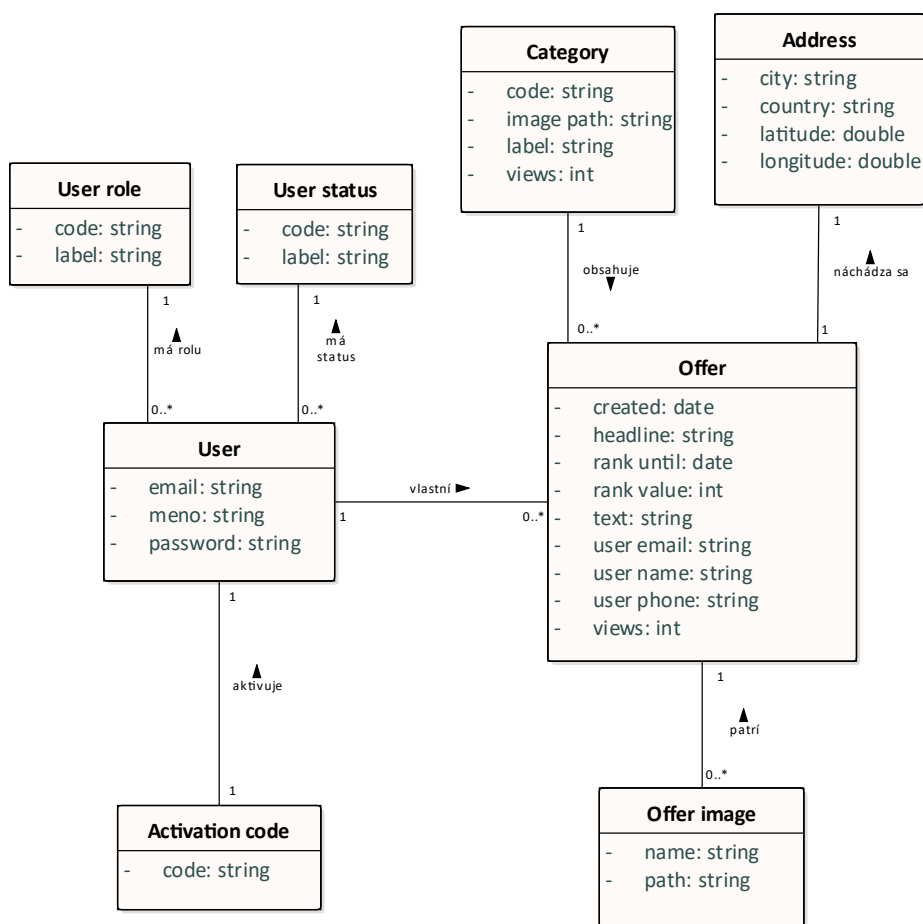
V nasledujúcej sekcii je zobrazená tabuľka pokrytia. Pokiaľ by ľubovoľná funkčná požiadavka nebola pokrytá žiadnym prípadom užitia, išlo by o zbytočnú funkčnú požiadavku, alebo požiadavku, na ktorú sa zabudlo.

Prípady užitia/Funkčné požiadavky	F01	F02	F03	F04
UC01		✓		
UC02		✓		
UC03		✓		
UC04	✓			
UC05	✓			
UC06	✓			
UC07			✓	
UC08			✓	
UC09			✓	
UC10			✓	
UC11			✓	
UC12			✓	
UC13	✓			
UC14	✓			
UC15				✓
UC16	✓			
UC17	✓			
UC18	✓			
UC19	✓			
UC20		✓		
UC21				✓
UC22				✓
UC23		✓		
UC24		✓		

Tabuľka 2.1: Tabuľka pokrytia funkčných požiadaviek

2.5 Návrh doménového modelu

Na základe doposiaľ analyzovaných požiadaviek z predchádzajúcich kapitol bol navrhnutý doménový model zobrazený na obrázku 2.9. Doménový model aplikácie bol spracovaný nástrojom EA podobne ako prípady použitia v kapitole 2.3. V doménovom modeli je pokryté všetko, čo je potrebné k realizácii funkčných požiadaviek z kapitoly 2.1.



Obr. 2.9: Doménový model webovej aplikácie

2.6 Návrh užívateľského rozhrania

Nasledujúca sekcia obsahuje návrh užívateľského rozhrania webovej aplikácie. Pri návrhu sa postupovalo princípom desktop-first, preto sú aj nasledujúce návrhy na obrázkoch v tomto rozlíšení. Desktop-first je postup, pri ktorom sa kladie najväčší dôraz na optimalizovanie rozhrania webovej aplikácie pre obrazovky s vysokým rozlíšením.

Podľa návrhu webovej aplikácie pre obrazovku s rozlíšením XL budú odvodené návrhy pre obrazovky s rozlíšením XS, SM, MD a LG. Počas celého vývoja sa bude dbať na zachovanie responzivity systému naprieč všetkými typmi zariadení.

Na vytvorenie tzv. papierového modelu webovej aplikácie bol použitý nástroj Balsamiq verzie 4.2.3. Intuitívnosť a prehľadnosť boli hlavnými faktormi pri tvorbe dizajnu webovej aplikácie.

2.6.1 Domovská stránka

Domovská stránka bola navrhnutá, aby bola pre užívateľov intuitívna a zároveň prehľadná. Stránka používa zaužívané princípy, ktoré užívateľom zjednodušujú orientáciu. Vizualna podoba domovskej obrazovky je zobrazená na obrázku 2.10.

Umiestnenie loga na domovskej stránke sa môže javiť ako nepodstatné, avšak opak je pravdou. 89 % užívateľov si skôr zapamätá logo umiestnené na tradičnej pozícii vľavo hore ako logo umiestnené na pravej strane obrazovky [9]. Všetky inzertné portály analyzované v kapitole 1.4 disponovali logom, ktoré bolo umiestnené v ľavom hornom rohu. Dôsledkom týchto pozorovaní je logo webovej aplikácie umiestnené v ľavom hornom rohu.

Vyhľadávacie pole je vycentrované na stred a nachádza sa v hornej časti aplikácie. Je dostatočne veľké, a v prípade obrazoviek s nižším rozlíšením sa adekvátne prispôsobí. Poskytuje full textové vyhľadávanie evidovaných remesiel v systéme.

Základná trojica tlačidiel Moje ponuky, Pridať ponuku a Prihlásiť sa nachádza v pravom hornom rohu webovej aplikácie. Tlačidlo Moje ponuky po kliknutí presmeruje užívateľa na obrazovku so zoznamom užívateľových ponúk. V prípade, že užívateľ nie je prihlásený, bude takýto užívateľ presmerovaný na prihlasovací formulár. Pridanie ponuky sa vo webovej aplikácii realizuje pomocou tlačidla Pridať ponuku, ktorá užívateľa presmeruje na formulár vytvorenia ponuky. Tlačidlo Prihlásiť presmeruje užívateľa na prihlasovací formulár.

Pod vyhľadávacím poľom je umiestnený text s aktuálnym počtom aktívnych ponúk. Číslo, ktoré vyjadruje počet aktívnych ponúk, je od textu farebne odlíšené.

Ďalej nasledujú obrázky remesiel, na ktorých je remeslo vizuálnou formou stručne a výstižne opísané. Na zabezpečenie jednoznačnosti je v rámci každého obrázku aj názov remesla, ktorý sa nachádza v dolnej časti. Po kliknutí

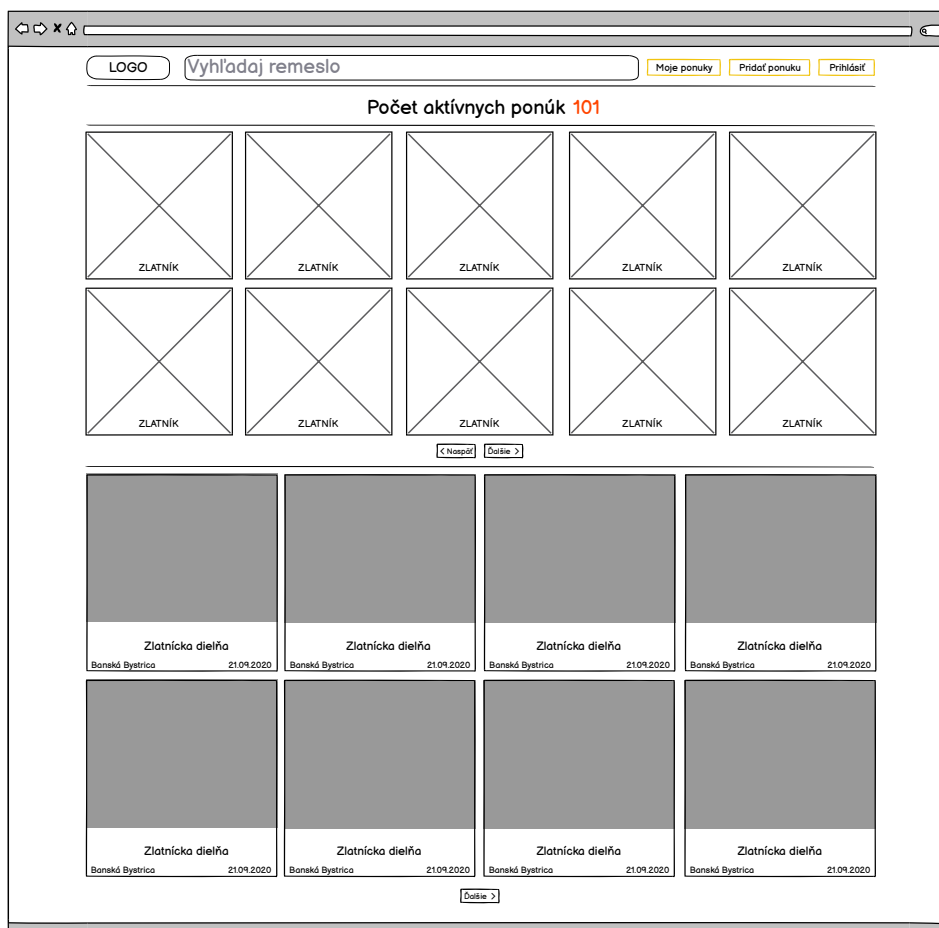
2. NÁVRH SYSTÉMU

na obrázok s remeslom je užívateľovi zobrazená obrazovka so zoznamom evidovaných inzerátov konkrétneho remesla. Aby sa predišlo zahlteniu užívateľa a zachovaniu dobrých praktík tvorby užívateľského rozhrania je na domovskej stránke zobrazených iba prvých 10 najpopulárnejších remesiel.

Pod obrázkami remesiel sa nachádzajú tlačidlá Naspäť a Ďalej slúžiace na stránkovanie remesiel.

V spodnej časti obrazovky možno nájsť výpis najnovšie pridaných inzerátov. Na zamedzenie neprehľadnosti a zahltenia užívateľa veľkým množstvom informácií je zobrazených iba 8 najnovších inzerátov. Každý inzerát pozostáva z fotografie, nadpisu ponuky, adresy a dátumu vytvorenia. Po kliknutí na plochu inzerátu je užívateľ presmerovaný na detail konkrétneho inzerátu.

V prípade, že užívateľovi nestačia zobrazené inzeráty, môže stlačiť tlačidlo Ďalšie, ktoré ho presmeruje na zoznam všetkých ponúk na inzertnom portáli zoradených podľa dátumu vytvorenia.



Obr. 2.10: Domovská stránka webovej aplikácie

2.6.2 Detail inzerátu

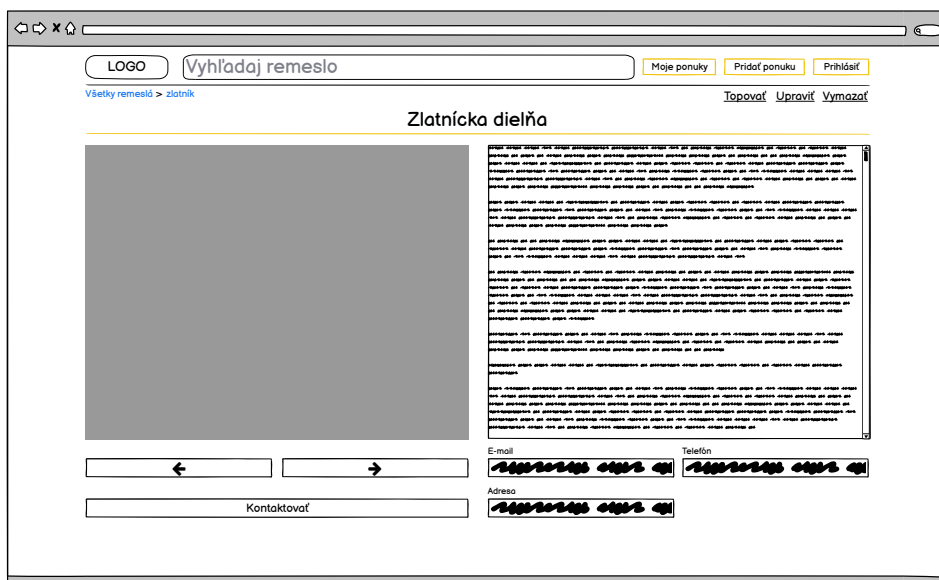
Detail inzerátu má rovnakú distribúciu a funkcionálnosť prvkov hornej lišty (logo, vyhľadávacie pole, moje ponuky, pridať ponuku a prihlásiť) aká sa nachádza na domovskej stránke.

Vizuálna forma detailu inzerátu sa nachádza na obrázku 2.11.

V ľavej hornej časti obrazovky pod hornou lištou sa nachádza cesta k inzerátu v podobe odkazov na jednotlivé obrazovky.

Na rovnakej vertikálnej úrovni ako odkazy sú umiestnené aj tlačidlá **Topovať**, **Upraviť** a **Vymazať**, ktoré sa zobrazia iba vlastníčkovi inzerátu.

Ďalej nasleduje nadpis inzerátu. Fotografia sa nachádza na ľavej časti obrazovky, zatiaľ čo text na pravej. Pod fotografiou sa nachádzajú tlačidlá na prepínanie medzi fotografiami inzerátu a tlačidlo na kontaktovanie remeselníka. Detaily inzerenta ako: e-mailová adresa, telefón a adresa sú uvedené pod textom ponuky.



Obr. 2.11: Detail inzerátu

2.6.3 Kategória a filtrovanie

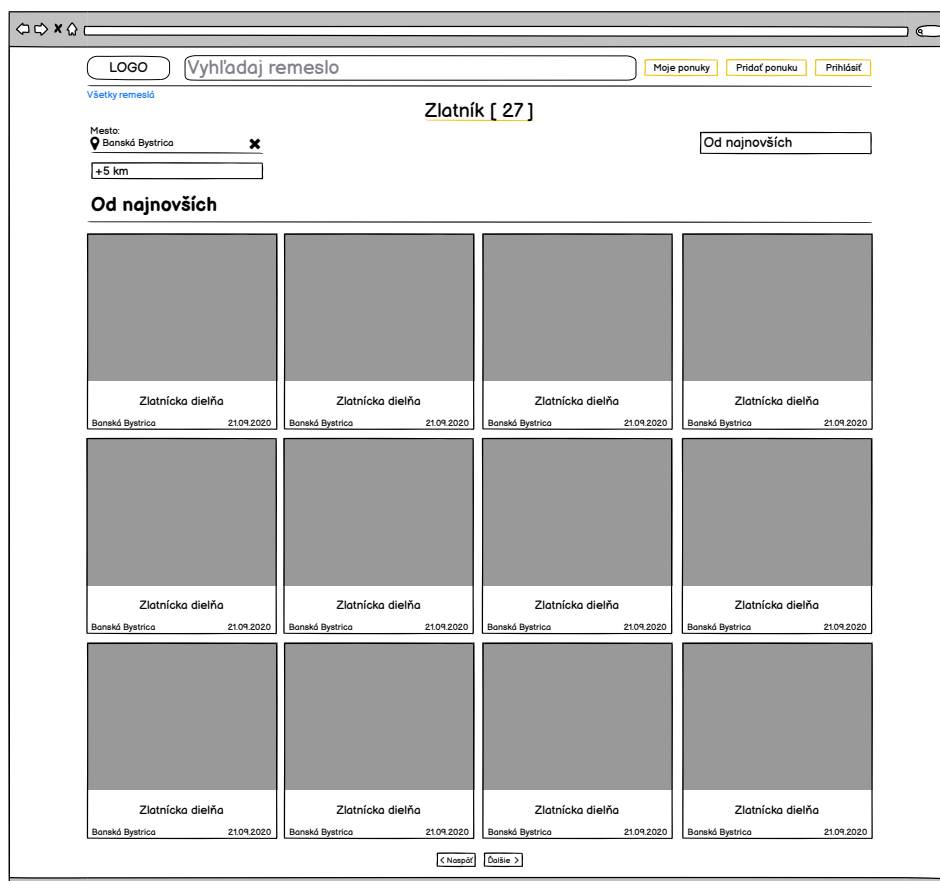
Každá kategória bude obsahovať názov remesla s počtom inzerátov. Na obrázku 2.12 je zobrazený obsah kategórie. Užívateľ bude mať možnosť vyhľadávať inzeráty v kategórii podľa adresy. Po zadaní validnej adresy bude zobrazené políčko udávajúce, v akom rádiuse má webová aplikácia filtrovať ponuky.

V pravej časti obrazovky bude políčko na radenie inzerátov podľa dátumu pridania alebo podľa počtu pozretí. Vo východiskovom nastavení bude kategória zobrazovať inzeráty z celého Slovenska zoradené podľa dátumu pridania od najnovších.

Zoradenie inzerátov bude mať jednu hlavnú podmienku, a to že topovaný inzerát bude vždy v zozname uprednostnený pred bežným inzerátom.

Štruktúra inzerátov je identická so štruktúrou na domovskej stránke. Rozdiel je v počte zobrazovaných inzerátov na jednu stránku.

Tlačidlá v dolnej časti obrazovky budú slúžiť na stránkovanie obsahu spomínanej kategórie.



Obr. 2.12: Obsah kategórie

2.6.4 Vytvorenie inzerátu

Na obrazovku vytvorenia inzerátu sa dá dostať pomocou tlačidla Pridať ponuku, ktoré je situované v pravom hornom rohu webovej aplikácie. Vytvorenie inzerátu obsahuje čo najmenej údajov, aby bolo pre užívateľa čo najjednoduchšie.

Prvé pole, ktoré je užívateľ vyzvaný vyplniť za predpokladu, že vyplňuje údaje odhora dole je výber kategórie. Vybrať kategóriu môže užívateľ zo zoznamu, ktorý sa otvorí po kliknutí do textového poľa, alebo pomocou full textového vyhľadania priamo v textovom poli. Nadpis a textový popis nasledujú po výbere kategórie. E-mail aj meno sú predvyplnené údajmi zadanými pri registrácii. Pole na vyplnenie telefónu je explicitne označené ako nepovinné. Z tohto označenia obzvlášť benefitujú inzerenti, ktorí nechcú zverejniť svoje telefónne číslo, ale zároveň vnímajú každé pole ako povinné. Adresa je posledný povinný údaj pre vytvorenie inzerátu. Pod adresou sa už nachádza iba tlačidlo na pridanie fotografie, ktoré po rozkliknutí zobrazí systémové okno na výber súborov. Nahrať fotografiu sa dá aj pomocou tzv. funkcie drag and drop.

Taký istý formulár ako je na obrázku 2.13 zobrazený, je využívaný aj pri úprave inzerátu s rozdielom, že všetky údaje sú pre užívateľa predvyplnené, a tlačidlo Pridať ponuku je zmenené na tlačidlo Upraviť, aby to bolo pre užívateľa prirodzenejšie.

Po úspešnom vytvorení inzerátu je užívateľ presmerovaný na prehľad svojich aktívnych inzerátov.

Obrázok 2.13 zobrazuje obrazovku vytvorenia inzerátu.

The screenshot shows a web browser window with the following elements:

- Navigation bar: LOGO, search bar with 'Vyhľadaj remeslo', 'Moje ponuky', 'Pridať ponuku', and 'p.večera' buttons.
- Form title: 'Vytvorenie inzerátu'.
- Form fields:
 - Kategória:** 'Vyberte kategóriu' with a dropdown arrow.
 - Nadpis:** 'Nadpis ponuky' with a text input field.
 - Textový popis:** 'Text ponuky.' with a large text area.
 - E-mail:** 'Váš e-mail' with a text input field.
 - Meno:** 'Vaše meno' with a text input field.
 - Telefón (nepovinný):** 'Vaše meno' with a text input field.
 - Adresa:** with a text input field and a close icon (x).
 - Photo upload:** A dashed box containing 'Pridať fotky' (with a plus icon) and 'Tu pretiahnite fotky (max 3)' (with a plus icon).
- Submit button: 'Pridať inzerát' at the bottom center.

Obr. 2.13: Obrazovka vytvorenia inzerátu

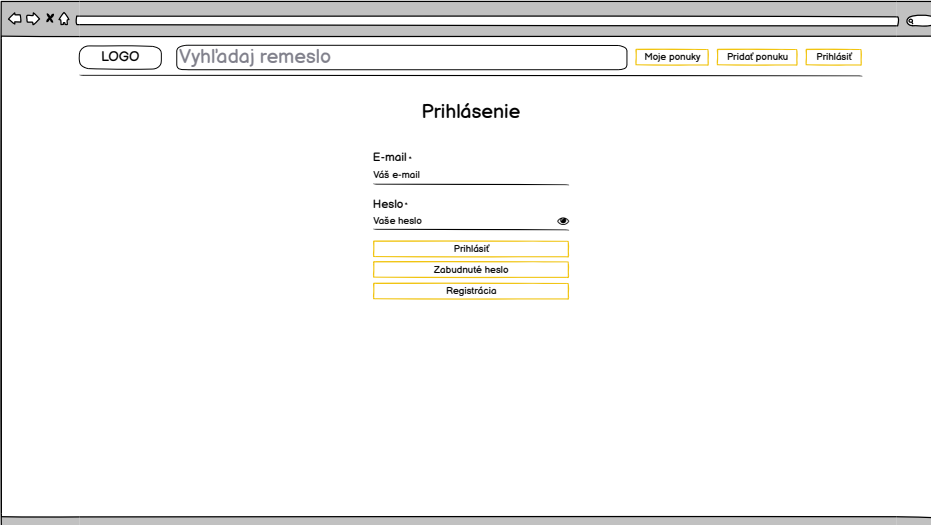
2.6.5 Prihlásenie

Obrazovka prihlásenia obsahuje jednoduchý formulár obsahujúci e-mailovú adresu a heslo. Táto obrazovka sa zobrazí po kliknutí na tlačidlo Prihlásiť, alebo sa neprihlásený užívateľ pokúsi prístup k obsahu webovej aplikácie, ku ktorému nemá oprávnenie.

Pod formulárom sa nachádzajú odkazy smerujúce k resetovaniu hesla, alebo registráciu nového užívateľského účtu.

Po úspešnom prihlásení je užívateľ presmerovaný na stránku, z ktorej bolo volanie uskutočnené.

Obrazovka prihlasovacieho formulára je zobrazená na obrázku 2.15.



The image shows a web browser window displaying a login form. At the top left, there is a 'LOGO' button and a search bar containing the text 'Vyhľadaj remeslo'. To the right of the search bar are three buttons: 'Moje ponuky', 'Pridať ponuku', and 'Prihlásiť'. The main content area is titled 'Prihlásenie'. It contains two input fields: 'E-mail - Váš e-mail' and 'Heslo - Vaše heslo'. Below the password field is a small eye icon for toggling visibility. At the bottom of the form are three buttons: 'Prihlásiť', 'Zabudnuté heslo', and 'Registrácia'.

Obr. 2.14: Prihlasovací formulár

2.6.6 Registrácia

Rozloženie obrazovky registrácie je veľmi jednoduché. Obrazovka obsahuje formulár, ktorý zahŕňa meno, e-mailovú adresu, heslo a zopakované heslo užívateľa. Pod týmito poľami sa nachádza tlačidlo Registrovať.

Po registrácii nie je užívateľ automaticky prihlásený, lebo je potrebná aktivácia nanovo vytvoreného užívateľského účtu, dôsledkom tohto faktu je užívateľ presmerovaný na prihlasovací formulár.

Obrazovka registračného formulára je zobrazená na obrázku 2.15.



The image shows a web browser window displaying a registration form. At the top of the browser, there is a search bar with the text "Vyhľadaj remeslo" and a "LOGO" button. To the right of the search bar are three buttons: "Moje ponuky", "Pridaj ponuku", and "Prihlásiť". The main content of the page is a registration form titled "Registrácia". The form consists of the following fields and buttons:

- Meno - Vaše meno
- E-mail - Váš e-mail
- Heslo - Vaše heslo (with an eye icon for visibility)
- Heslo - Vaše heslo (with an eye icon for visibility)
- Registrovať (button)

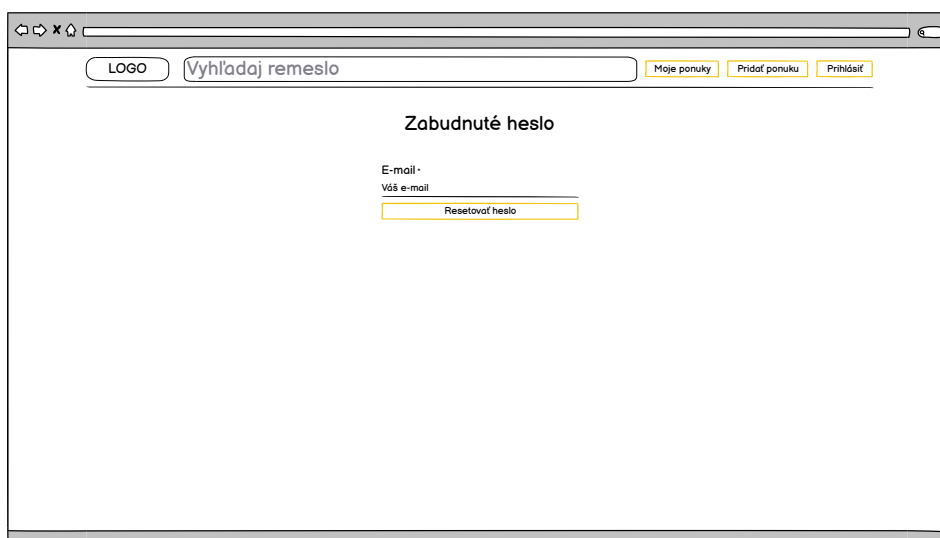
Obr. 2.15: Registračný formulár

2.6.7 Resetovanie hesla

Obrazovka resetovania hesla je najjednoduchšia obrazovka zo všetkých doposiaľ uvedených obrazoviek. Obsahuje pole pre vyplnenie e-mailovej adresy, na ktorú je odoslané nanovo vygenerované heslo.

Funkcia resetovania hesla je prístupná z prihlasovacieho formulára ako aj z menu, ktoré sa zobrazí po rozkliknutí užívateľského účtu.

Vizualizácia tejto obrazovky sa nachádza na obrázku 2.16.



The image shows a web browser window with a search bar at the top containing the text "Vyhľadaj remeslo". To the right of the search bar are three buttons: "Moje ponuky", "Pridať ponuku", and "Prihlásiť". Below the search bar, the page title is "Zabudnuté heslo". Underneath the title, there is a label "E-mail" followed by "Váš e-mail" and a text input field. Below the input field is a button labeled "Resetovať heslo".

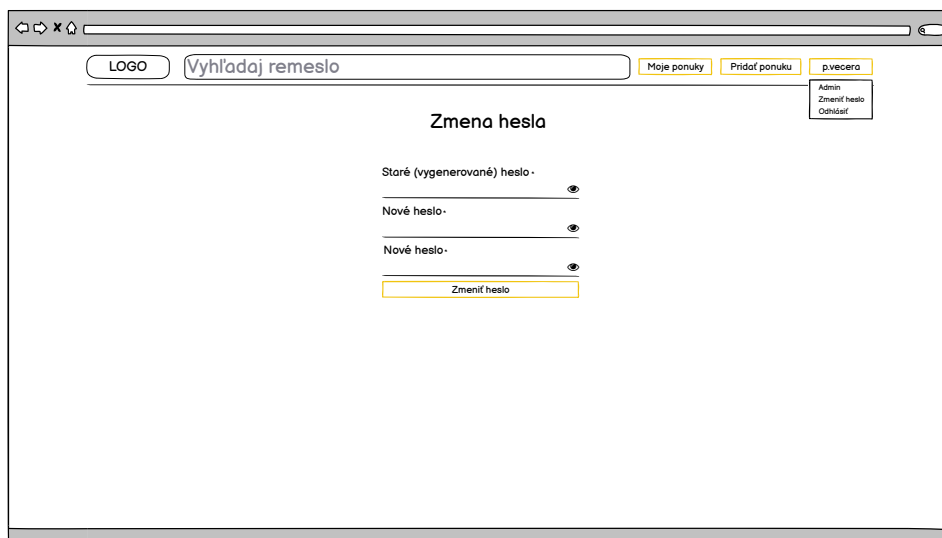
Obr. 2.16: Formulár na resetovanie hesla

2.6.8 Zmena hesla

Pristúpiť na obrazovku zmeny hesla je v systéme uskutočniteľné dvoma spôsobmi. Užívateľ buď požiadal o zmenu hesla a webová aplikácia ho automaticky presmerovala, alebo ku tejto funkcii pristúpil priamo z menu po rozkliknutí užívateľského účtu.

Po prístupe na obrazovku sa zobrazí formulár, ktorý obsahuje tri textové polia na vyplnenie hesla. Prvé textové pole slúži na overenie totožnosti užívateľa vyplnením starého hesla, zvyšné dve textové polia slúžia na vytvorenie nového hesla.

Obrazovku možno vidieť na obrázku 2.17.



The image shows a web browser window with a search bar containing 'Vyhľadaj remeslo' and a navigation menu with 'Moje ponuky', 'Pridaj ponuku', and 'p vecera'. The main content area is titled 'Zmena hesla' and contains three text input fields: 'Staré (vygenerované) heslo', 'Nové heslo', and 'Nové heslo'. Each field has a toggle icon to the right. Below the fields is a yellow button labeled 'Zmeniť heslo'. A user profile dropdown menu is visible in the top right corner, showing 'Admin', 'Zmeniť heslo', and 'Odtlačiť'.

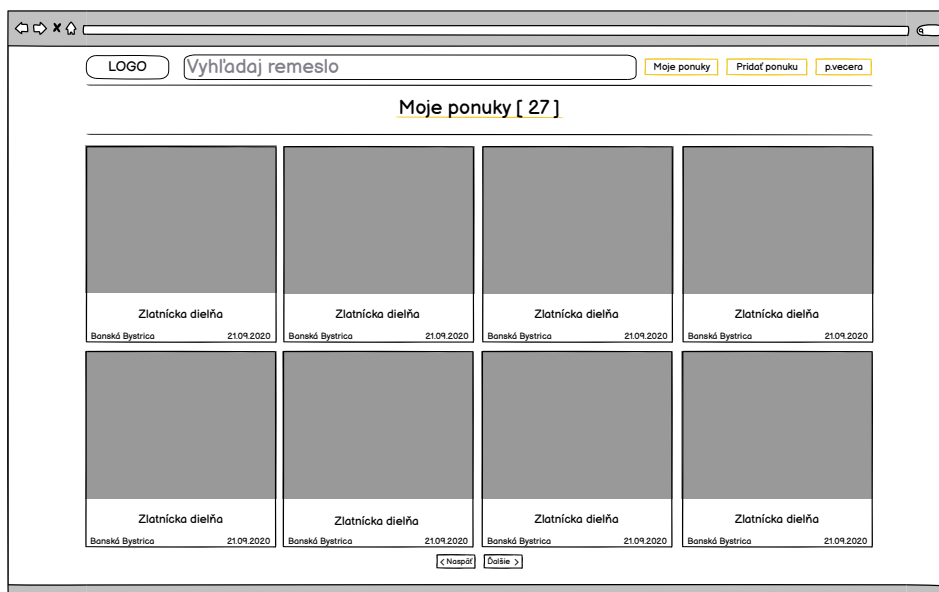
Obr. 2.17: Formulár na zmenu hesla

2.6.9 Moje ponuky

Zobrazenie aktuálne vlastnených inzerátov môže užívateľ pomocou tlačidla Moje ponuky. Po rozkliknutí tohto tlačidla sa zobrazí obrazovka obsahujúca užívateľove aktívne ponuky. Na túto obrazovku je užívateľ presmerovaný aj po pridaní alebo odstránení ponuky.

Pod hornou lištou je uvedený počet aktívnych inzerátov. Inzeráty majú rovnakú štruktúru ako inzeráty na obrázkoch 2.10 alebo 2.12.

Návrh spomínanej obrazovky je zobrazený na obrázku 2.18.



Obr. 2.18: Obrazovka vlastnených ponúk užívateľom

2.6.10 Užívatelia - Administrátorská sekcia

Obrazovka administrátorskej sekcie spravujúca užívateľské profily obsahuje veľkú mriežku s údajmi o evidovaných užívateľoch. Na obrazovku možno prísť po kliknutí na tlačidlo admin nachádzajúce sa v menu, ktoré sa zobrazí po kliknutí na tlačidlo užívateľského účtu. Spomínané menu sa nachádza v pravom hornom rohu na obrázku 2.17.

Ľavá časť obrazovky obsahuje administrátorskú lištu, vďaka ktorej sa dá pristupovať na jednotlivé administrátorské obrazovky.

Mriežka obsahuje nasledujúce údaje o užívateľovi: meno, e-mail, užívateľská rola a užívateľský status.

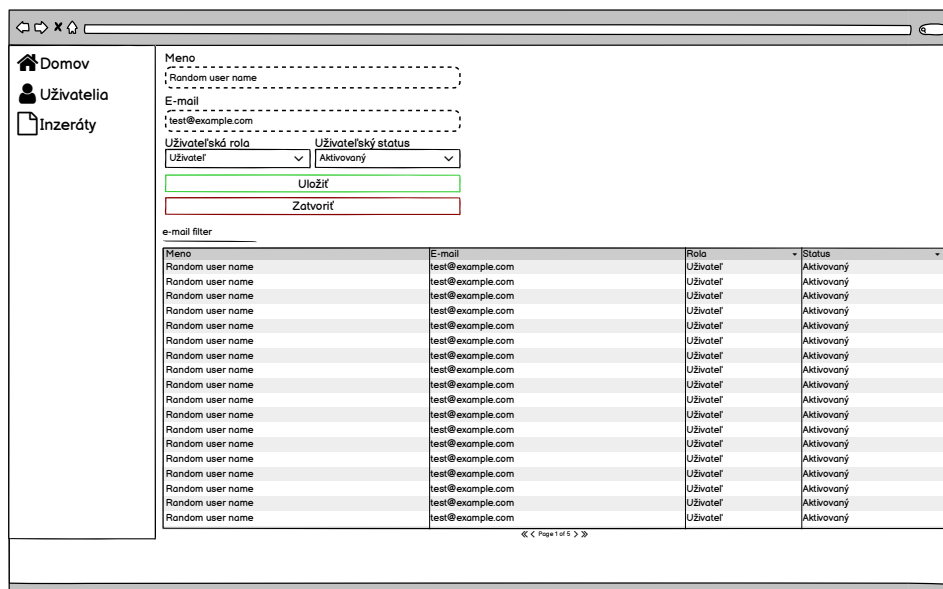
Pod mriežku sa nachádza klasické stránkovanie, ktoré umožňuje prechádzať celý zoznam evidovaných užívateľov.

Nad mriežkou je vyhľadávacie pole, ktoré vyhľadáva užívateľov pomocou ich e-mailovej adresy.

Po rozkliknutí položky mriežky je zobrazený detail užívateľa, v ktorom je možná zmena užívateľovej role alebo statusu. Pri zmene užívateľovho statusu na Zablokovaný sa automaticky zo systému vymažú všetky ponuky užívateľa.

Tlačidlo Uložiť slúži na uloženie zmien a administrátorovi sa následne zobrazí dialógové okno so zmenami, ktoré ide vykonať. Pokiaľ užívateľ klikne na tlačidlo Zatvoriť, tak sa zatvorí formulár obsahujúci užívateľove detaily.

Táto obrazovka je zobrazená na obrázku 2.19.



Obr. 2.19: Zobrazenie užívateľov v administrátorskej sekcii

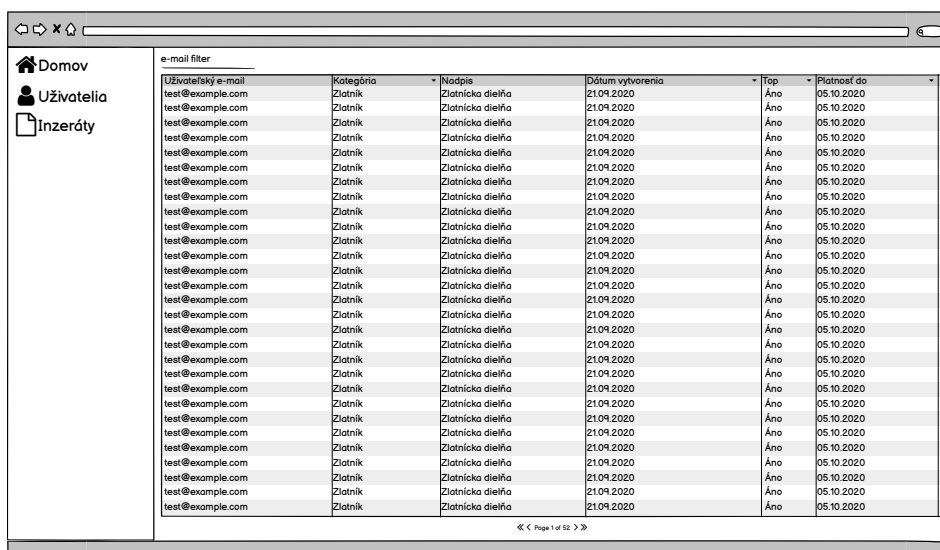
2.6.11 Inzeráty - Administrátorská sekcia

Obrazovka administrátorskej sekcie spravujúcej inzeráty sa výrazne od obrazovky v sekcii 2.6.10 nelíši. Ľavá časť obrazovky a vyhľadávacie pole sú identické ako v predchádzajúcej sekcii. Obrazovka disponuje taktiež mriežkou, ktorá však obsahuje odlišné údaje. Mriežka obsahuje nasledujúce údaje:

- užívateľský e-mail
- kategória
- nadpis
- dátum vytvorenia
- top - atribút značí či je ponuka topovaná, alebo nie
- platnosť do

Po kliknutí na položku mriežky je administrátor presmerovaný na detail inzerátu.

Zobrazenie administrátorskej sekcie spravujúcej inzeráty je zobrazené na obrázku 2.20.



The screenshot shows a web browser window with a sidebar on the left containing navigation links: Domov, Užívateľia, and Inzeráty. The main content area displays a table titled 'e-mail filter' with the following columns: Užívateľský e-mail, Kategória, Nadpis, Dátum vytvorenia, Top, and Platnosť do. The table contains 20 rows of data, all with the same values: 'test@example.com' for the email, 'Zlatník' for the category, 'Zlatnícka dieňša' for the title, '21.09.2020' for the creation date, 'Áno' for the top status, and '05.10.2020' for the validity date.

Užívateľský e-mail	Kategória	Nadpis	Dátum vytvorenia	Top	Platnosť do
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020
test@example.com	Zlatník	Zlatnícka dieňša	21.09.2020	Áno	05.10.2020

Obr. 2.20: Zobrazenie ponúk v administrátorskej sekcii

2.7 Súhrn

Cielom kapitoly bolo vytvoriť návrh webovej aplikácie so zameraním na inzerciu práce remeselníkov. Tento návrh bol rozdelený do viacerých častí.

V kapitole boli spracované funkčné a nefunkčné požiadavky, od ktorých sa odvíjal ďalší postup. Funkčné resp. nefunkčné požiadavky sú popísané v kapitole 2.1 resp. 2.2.

Prípady použitia boli navrhnuté na pokrytie všetkých funkčných požiadaviek, a zároveň obsahujú požiadavky z analýzy v kapitole 1. Prípady použitia boli modelované pomocou nástroja *Enterprise Architect*.

Doménový model na obrázku 2.9 popisuje vnútornú štruktúru aplikácie, a vytvára podklad pre vytvorenie databázového modelu. Podobne ako prípady použitia bol aj doménový model vytvorený prostredníctvom rovnakého modelovacieho nástroja.

Na základe užívateľských požiadaviek z kapitoly 1 bol prostredníctvom nástroja *Balsamiq* vytvorený návrh užívateľského rozhrania popísaný v kapitole 2.6.

Po spracovaní uvedených častí vznikol kompletný návrh webovej aplikácie. Na základe tohto návrhu bola webová aplikácia ďalej implementovaná.

Implementácia funkčného prototypu

V tejto kapitole budú podrobne rozobraté použité technológie a postup, akým webová aplikácia pre inzerciu práce remeselníkov vznikala. Ďalej bude popísaná súborová štruktúra a konfigurácia aplikácie. Použité frameworky a uplatnené princípy pri vývoji budú taktiež obsahom nasledujúceho textu.

Keďže sa aplikácia skladá z vyše 11 000 riadkov kódu, budú v kapitole zobrazené iba malé fragmenty zdrojového kódu.

3.1 Framework

Podľa [10] je framework, alebo softwarový framework platforma na vývoj softwarových aplikácií. Poskytuje základy, na ktorých môžu softwaroví vývojári vytvárať programy pre špecifické platformy. Napríklad framework môže obsahovať predefinované triedy a funkcie, ktoré môžu spracovávať užívateľský vstup, spravovať hradwarové zariadenia a interagovať so systémovým softwarom. Tento fakt zjednodušuje vývojový proces, pretože umožňuje programátorom upriamiť pozornosť na podstatu vyvíjanej aplikácie.

Framework poskytuje pre vývojára veľmi veľa výhod a predstava vývoja aplikácie bez použitia akéhokoľvek frameworku je v dnešnej dobe absurdná. V stručnosti, framework urýchľuje, sprehľadňuje a zjednodušuje proces vývoja aplikácií, keďže sa programátor nemusí zaoberať základnou funkcionalitou, ale môže sa plno venovať riešeniu aplikácie.

V bakalárskej práci bolo použitých viacero frameworkov, ktoré výrazne uľahčili a zrýchlili proces vývoja. Najdôležitejšie frameworky použité v aplikácii sú ďalej v texte elaborované.

3.2 Anotácie

V backendovej časti webovej aplikácie boli využité anotácie, ktoré ponúkajú prevažne moduly Springu. Trend využívania anotácií bol zachovaný aj vo frontendovej časti aplikácie s obohateným o ďalší framework, a to konkrétne o framework Vaadin, ktorý je podrobnejšie rozobratý v kapitole 3.5.1.

Použité anotácie poskytujú informácie pre program, ktoré ďalej určujú akým spôsobom majú byť jednotlivé triedy, rozhrania, metódy a atribúty spracované.

Malá ukážka použitých anotácií v aplikácii a ich stučný popis:

- @Component

Anotácia je v Springu používaná pre všeobecné označenie komponentov. Triedy, ktoré sú označené touto anotáciou predstavujú v Springu tzv. beans.

- @Entity

Anotácia sa využíva pre triedy dátovej vrstvy. Táto trieda by mala obsahovať primárny kľúč a konštruktor bez parametrov. Trieda je namaľovaná na príslušnú tabuľku v databáze. Jedna instancia odpovedá jednému záznamu v databázovej tabuľke, pričom atribúty instance majú vlastné anotácie a predstavujú jednotlivé údaje v riadku.

- @Data

Ide o veľmi užitočnú anotáciu, ktorá zabezpečuje automatické vygenerovanie setterov a getterov pre triedu, ktorá je touto anotáciou označená. Anotácia šetrí čas a úsilie programátorovi, a zároveň robí program prehľadnejším a kratším.

- @Repository

Trieda označená touto anotáciou je spôsobilá na výkon CRUD (Create, Read, Update, Delete) operácií nad databázou. Ďalej je možnosť napísania vlastných otázok tzv. *@Queries* v podobe metód, ktoré sú nad databázou exekúované. Takéto metódy môžu byť písane formou JPQL (Jakarta Persistence Query Language) alebo SQL (Structured Query Language) otázok. JPQL forma ponúka rozdielny spôsob písania otázok. Možno využiť už aj existujúcu funkciu stránkovania, ktorú ponúka Spring Data JPA (Java Persistence API).

- @Transactional

Anotácia sa definuje nad metódami servisnej vrstvy v aplikácii, ktoré komunikujú s databázou. Pri zlyhaní databázovej transakcie dochádza k automatickému rollbacku do pôvodného stavu, ktorý je zabezpečený práve spomínanou anotáciou.

- `@Autowired`

Anotácia slúži na vytváranie automatickej závislosti v aplikácii. V prípade, že Spring nedokáže požadovanú závislosť vytvoriť je vyhodnená výnimka a kompilácia aplikácie končí neúspechom. Anotácia sa častokrát používa nad triedami servisnej a dátovej vrstvy.
- `@Service`

Anotácia sa využíva pre triedy servisnej vrstvy. Biznis logika aplikácie by mala byť obsiahnutá v triedach s týmto označením. Atribútom triedy sú väčšinou repozitáre označené anotáciou `@Autowired`. Metódy, ktoré pracujú a komunikujú s databázou sú označené anotáciou `@Transactional`.
- `@Route`

Anotácia sa využíva vo frontendovej časti aplikácie, kde označuje triedy, ktoré ďalej slúžia ako jednotlivé webové stránky. Hodnota vyplnená v anotácii pracuje ako URL (Uniform Resource Locator) fragment. Pomocou tohto princípu je smerovanie v aplikácii výrazne zjednodušené. Vaadin je so Springom dobre integrovaný, a výsledkom je možnosť vytvorenia automatických závislosti na backendovú časť aplikácie priamo v konštruktoze triedy pomocou anotácie `@Autowired`.
- `@CssImport`

Anotácia ma za úlohu vytvoriť závislosť medzi css súborom a triedou, nad ktorou je použitá. Pomocou tejto závislosti je css súbor importovaný, a je súčasťou vizuálnej stránky webovej aplikácie.

3.3 Adresárová štruktúra

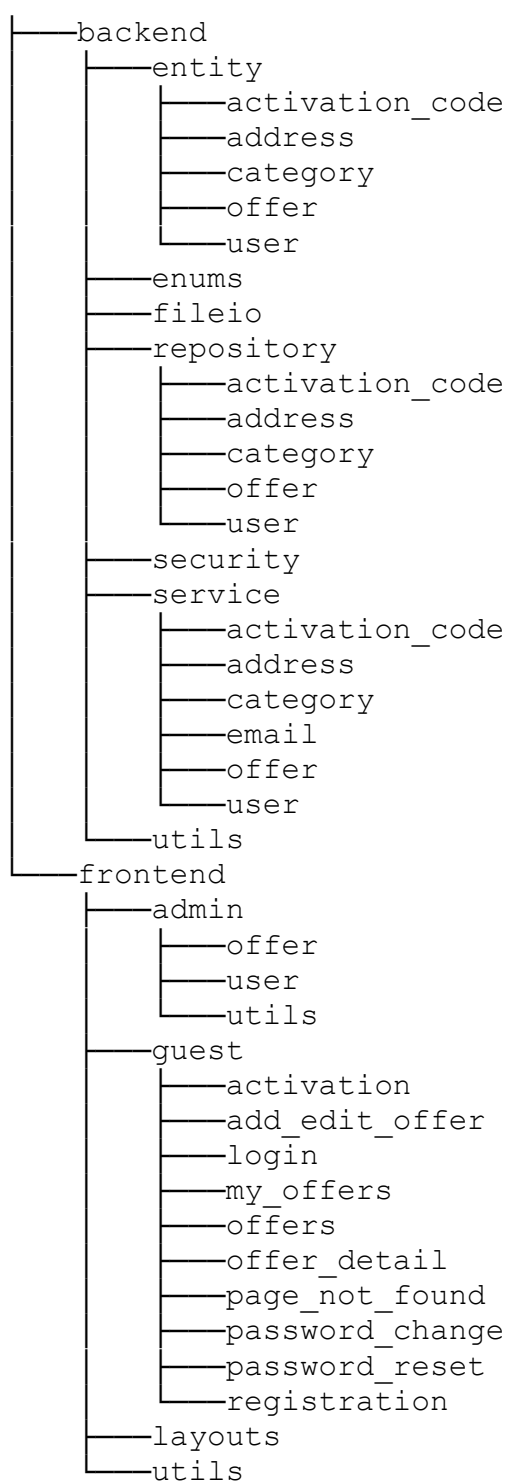
Adresárová štruktúra je z hľadiska lepšej orientácie v aplikácii pre vývojára dôležitá. Pri vývoji bol kladený dôraz na logické a intuitívne členenie tried, rozhraní a všeobecne súborov do adresárov.

Webová aplikácia neobsahuje odporúčanú Spring MVC (Model-view controller) adresárovú štruktúru podľa [11]. Táto štruktúra je odporúčaná v prípade, že ide o trojvrstvovú architektúru, a je potrebné rozhranie medzi prezenčnou a dátovou vrstvou.

Vďaka frameworku Vaadin, ktorý je ďalej podrobnejšie rozobratý v kapitole 3.5.1, je systém založený na dvojvrstvovej architektúre. Výhoda dvojvrstvovej architektúry spočíva v jednoduchom prepojení medzi prezenčnou a dátovou vrstvou.

Časť adresárovej štruktúry webovej aplikácie je zobrazená na obrázku 3.1.

3. IMPLEMETÁCIA FUNKČNÉHO PROTOTYPU



Obr. 3.1: Adresárová štruktúra aplikácie

3.4 Backend

3.4.1 Java

Prvá zmienka o Jave, vtedajšom *Oak*, je datovaná v roku 1991 Jamesom Goslingom. V roku 1995 bola uvedená prvá oficiálna verzia Javy na trh ako nový objektovo-orientovaný programovací jazyk spoločnosťou Sun Microsystems.

Podľa [12] je Java aktuálne 2. najpopulárnejší programovací jazyk. Webová stránka, ktorej dáta boli použité sa snaží, o čo najvyššiu mieru objektivity pri hodnotení programovacích jazykov. Používa vyše 25 internetových vyhľadávačov.

Hodnoty sú vypočítané podľa počtu výsledkov vrátených najpopulárnejšími internetovými vyhľadávačmi.

Použitá vyhľadávacia otázka je:

- +”<language> programming”

Obrázok 3.2 zobrazuje rebríček najpopulárnejších programovacích jazykov. Medzi výhody programovania v Jave patrí:

- objektovo-orientované programovanie
Java je čisto objektový jazyk, čo znamená, že všetko je objekt s výnimkou primitívnych dátových typov.
- automatická správa pamäti
Garbage collector funguje v Jave na princípe mark and sweep. Takýto algoritmus v prvej fáze označí všetky dosiahnuteľné objekty z koreňa. V druhej fáze je celá pamäť na halde označená ako voľná, až na objekty dosiahnuteľné počas prvej fázy.
- nezávislosť od platformy
Zdrojový kód sa neprekladá do strojového, ale do tzv. bajt kódu, čo má za následok spustiteľnosť aplikácie v akomkoľvek prostredí, ktoré má k dispozícii JVM (Java Virtual Machine).

Faktor, ktorý mal pri výbere programovacieho jazyka najväčší vplyv je silne subjektívneho charakteru, a tým je autorova znalosť programovacieho jazyka. Z dôvodu komplexnosti a časovej náročnosti vývoja webovej aplikácie bolo štúdium nového programovacieho jazyka vylúčené, a preto je backendová časť webovej aplikácie realizovaná práve pomocou programovacieho jazyka Java. Pri implementácii bola použitá konkrétne verzia Java SE 11 (Java Standard Edition 11).

3. IMPLEMETÁCIA FUNKČNÉHO PROTOTYPU

Apr 2021	Apr 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	14.32%	-2.40%
2	1	▼	Java	11.23%	-5.49%
3	3		Python	11.03%	+1.72%
4	4		C++	7.14%	+0.36%
5	5		C#	4.91%	+0.16%
6	6		Visual Basic	4.55%	-0.18%
7	7		JavaScript	2.44%	+0.06%
8	14	▲	Assembly language	2.32%	+1.16%
9	8	▼	PHP	1.84%	-0.54%
10	9	▼	SQL	1.83%	-0.34%
11	19	▲	Classic Visual Basic	1.54%	+0.71%
12	22	▲	Delphi/Object Pascal	1.47%	+0.77%
13	13		Ruby	1.23%	-0.02%
14	12	▼	Go	1.22%	-0.13%

Obr. 3.2: 20 najpopulárnejších programovacích jazykov (údaje sú získane ku dňu 06. 04. 2021)

3.4.2 Spring

Spring je jeden z najpopulárnejších frameworkov na vytváranie Java enterprise aplikácií. Milióny ľudí na celom svete používajú Spring framework na vytváranie vysoko výkonného, jednoducho testovateľného a znovu použiteľného kódu.

Spring je nenáročný, čo sa veľkosti a prehľadnosti týka. Základná verzia Spring frameworku má iba okolo 2 MB (Mega Byte).

Kľúčové vlastnosti Spring frameworku môžu byť použité pri vývoji ľubovoľnej Java aplikácie, ale nachádzajú sa tu rozšírenia na tvorbu webových aplikácií na platforme Java EE (Java Enterprise Edition). Spring framework usiluje o uľahčenie používania J2EE (Java 2 Platform, Enterprise Edition), a pri vývoji propaguje dobré programátorské praktiky umožnením programovacieho modelu postaveného na POJO (Plain Old Java Object). [13]

Práve Spring framework a jeho moduly boli pri vývoji backendovej časti webovej aplikácie použité.

3.4.3 Spring Boot

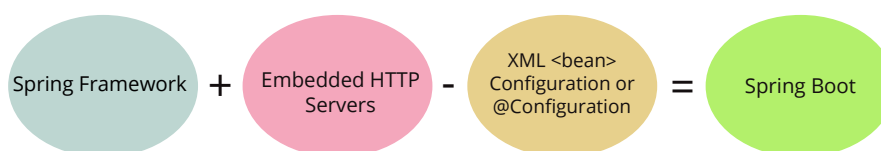
Spring Boot je projekt, ktorý je postavený na Spring frameworku. Poskytuje jednoduchšiu a rýchlejšiu prípravu, konfiguráciu a exekúciu jednoduchých, ale aj webových aplikácií.

Je to modul Springu, ktorý poskytuje tzv. RAD (Rapid Application Development) vlastnosť Spring frameworku. Používa sa na vytvorenie samostatnej Springovo založenej aplikácie, ktorú stačí iba spustiť, lebo potrebuje minimálnu konfiguráciu zo strany Springu.

V Spring Boote nie je žiadna požiadavka na XML (Extensible Markup Language) konfiguráciu. Používa konvencie nad konfiguračným softwarovým vzorom, čo znamená, že znižuje úsilie vývojára. [14]

Podľa [15] ponúka Spring Boot nasledujúce funkcie a benefity:

- Flexibilná možnosť konfigurácie Java Beans, XML konfigurácií a databázových transakcií.
- V Spring Boote je všetko automaticky nakonfigurované, a nie je potreba manuálnej konfigurácie.
- Možnosť už vyššie spomenutých anotácií.
- Uľahčuje spravovanie závislostí.
- Zahŕňa zabudovaný servlet kontajner.



Obr. 3.3: Ilustračný obrázok Spring Bootu podľa [14]

3.4.3.1 Apache Tomcat

Apache Tomcat je open-source Java servlet kontajner, ktorý implementuje niekoľko základných špecifikácií Java EE, konkrétne:

- Java Servlet
- JSP (JavaServer Pages)
- WebSockets API

Projekt spoločnosti Apache Software bol prvýkrát predstavený verejnosti v roku 1999, iba 4 roky po Jave. Tomcat začal ako referenčná implementácia pre prvý Java Servlet API a špecifikáciu JSP. Napriek tomu, že Tomcat už nie je referenčná implementácia ani pre jednu z týchto technológií, ostáva najpoužívanejším Java servletom, pýšiaci sa osvedčeným a otestovaným enginom s dobrou rozšíriteľnosťou. [16]

Apache Tomcat je servlet a JSP kontajner. Java servlet zapuzdruje kód a biznis logiku a definuje ako majú byť žiadosti a odpovede spracované v Java serveri. V aplikácii používa Spring Boot Apache Tomcat verzie 9.0.41.

3.4.4 Spring Security

Podľa [17] je Spring Security považovaný Springom už ako štandardný framework, ktorý poskytuje kompletne zabezpečenie aplikácie. Spring Security ponúka nespočetné množstvo nastavení, ktoré ponúkajú ochranu vývojárom pred útokmi ako napríklad:

- session fixation

Session fixation je technika, ktorá umožní útočníkovi ukradnúť užívateľove session id. Užívateľove session id je nastavené na vopred známu hodnotu, ktorú útočník pozná. Vďaka tomuto faktoru vie útočník toto session id užívateľovi kompromitovať.

- clickjacking

Ide o útok, ktorý používa neviditeľné alebo falošné odkazy, ktoré sa tvária ako normálna súčasť aplikácie. Následne po kliknutí na ne môžu stiahnuť do užívateľovho zariadenia škodlivý software tzv. malware, navštíviť škodlivé webové stránky, alebo získať citlivé informácie o užívateľovi.

- cross-site request forgery

Útok spočíva v ovládnutí užívateľského účtu, ktorý nevedomky dáva príkazy webovej aplikácii. Internetový prehliadač užívateľa je ovládaný škodlivým softwarom, za účelom posielania požiadaviek na webovú aplikáciu. Webové aplikácie sú častokrát po overení totožnosti užívateľa benevolentnejšie, a tento fakt môže mať v konečnom dôsledku devastujúci efekt. Veľké škody možno napáchať v prípade, že ide o webovú aplikáciu banky.

Vo webovej aplikácii sa Spring Security framework používa na viacero miestach. Typickým použitím je zahešovanie užívateľského hesla pri registrácii. Autentizácia a autorizácia užívateľa je taktiež zabezpečovaná Spring Security frameworkom. Počas fázy autentizácie sa užívateľovi overuje totožnosť, ale aj pridávajú prístupové práva v rámci systému. V neposlednom rade Spring Security framework konfiguruje prístup užívateľa k interným stránkam webovej aplikácie na základe pridelenej authority.

Súčasťou Spring Security je aj trieda, ktorá hešhuje a overuje správnosť hesla. Na hešovanie hesla je využitý BCryptPasswordEncoder ponúkaný frameworkom Spring Security.

Táto trieda je zobrazená v nasledujúcom bloku kódu 3.1.

```
1 /** Class represents password security */
2 public class PasswordSecurity {
3     private static final int PASSWORD_STRENGTH = 10;
4
5     /**
6      * Method is encoding password
7      *
8      * @param plainTextPassword - password as a String
9      * @return - Method returns encoded password as a String
10    */
11    public static String getEncodedPasswordFromPlainText(String
12        plainTextPassword) {
13        if (plainTextPassword == null || plainTextPassword.equals(""))
14        {
15            throw new IllegalStateException("Password should not be
16            null or empty String");
17        }
18        return getPasswordEncoder().encode(plainTextPassword);
19    }
20
21    /**
22     * Method is validating password
23     *
24     * @param plainTextPassword - password as a String
25     * @param encodedPassword - encoded password as a String
26     * @return - Method returns true if password as a plain text is
27     *           matching encoded password
28     *           otherwise it returns false
29     */
30    public static boolean validatePassword(String plainTextPassword
31        , String encodedPassword) {
32        return getPasswordEncoder().matches(plainTextPassword,
33        encodedPassword);
34    }
35
36    /**
37     * Method prepares password encoder
38     *
39     * @return password encoder as {@link PasswordEncoder}
40     */
41    private static PasswordEncoder getPasswordEncoder() {
42        return new BCryptPasswordEncoder(PASSWORD_STRENGTH, new
43        SecureRandom());
44    }
45 }
```

Kód 3.1: Trieda zabezpečuje zahešovanie a overenie hesla v aplikácii.

3.4.5 Spring Data

Ďalší z radu užitočných Spring frameworkov je Spring Data framework. Tento framework sa sústreďuje na poskytnutie užívateľsky príjemného rozhrania pre komunikáciu s dátovou vrstvou. Spring Data JPA je súčasťou Spring Data, ktorá

3. IMPLEMETÁCIA FUNKČNÉHO PROTOTYPU

uľahčuje implementáciu repozitárov založených na princípe JPA. Framework si kladie za cieľ výrazne zlepšiť implementáciu prístupu k dátovej vrstve minimalizovaním úsilia na množstvo, ktoré je skutočne potrebné. [18]

Medzi výhody patrí napríklad:

- Sofistikovaná podpora pre budovanie repozitárov na základe Spring a JPA.
- Podpora stránkovania, dynamické vykonávanie otázok, schopnosť integrovať vlastný prístupový kód k dátam.
- Validácia anotovaných otázok @Query.
- Podpora mapovania entít založených na XML.

Vo webovej aplikácii sa Spring Data JPA framework používa na správu entít v databáze. Poskytuje vývojárom všetky CRUD operácie s databázou. Pomocou tohto frameworku sú údaje do databázy ukladané, vyhľadávané ale aj vymazávané.

Integrácia pred vyplnených otázok ušetrí vývojárovi veľa práce. Stránkovanie je v Spring Data JPA vyriešené za užívateľa, a v prípade, že nejde o tzv. native query môže vývojár túto funkcionality využiť. Funkcia stránkovania je dôležitá pri načítaní iba čiastočného obsahu webovej aplikácie, ktorá ma za úlohu urýchliť beh celej webovej stránky.

Zdrojový kód 3.2 automaticky kontroluje, či nevypršala platnosť topovania inzerátu, a v prípade ak áno, je topovanie inzerátu odobrané.

```
1 /**
2  * Method checks offer ranks everyday at 3 am and updates them
3  * if their subscription has expired
4  */
5 @Transactional
6 @Scheduled(cron = "0 0 3 * * *", zone = "Europe/Bratislava")
7 public void updateOffersRank() {
8     LOGGER.info("checking expiration dates of offers' rank");
9     LocalDateTime now = LocalDateTime.now();
10    List<Offer> offers = offerRepository.findAll();
11    for (Offer offer : offers) {
12        if (offer.getRankUntil() != null && offer.getRankUntil().
13            isBefore(now)) {
14            offer.setRankValue(0);
15            offer.setRankUntil(null);
16            saveOrUpdateOffer(offer);
17        }
18    }
19 }
```

Kód 3.2: Metóda ruší topovanie inzerátov

3.4.6 Mailsender

Vo webovej aplikácii je implementovaná trieda `EmailService`, ktorá je označená anotáciou `@Service`, a nachádza sa v servisnej vrstve aplikácie. Využíva rozhranie Springu na vytváranie a odosielanie e-mailových správ. Na korektné použitie poštovej služby ponúkanej Springom je potrebné nakonfigurovať SMTP (Simple Mail Transfer Protocol) v nastaveniach aplikácie.

Metódy triedy sú označené anotáciou `@Async`. Označenie má za následok, že je metóda považovaná programom za asynchrónnu a neblokuje ďalší beh aplikácie. Výsledkom tejto anotácie je výrazne zrýchlená odozva v sekciách aplikácie, kde sa používa komunikácia prostredníctvom e-mailových správ.

Ukážka metódy, ktorú využíva webová aplikácia na e-mail komunikáciu je zobrazená v bloku kódu 3.3.

```

1  /** Class represents email service */
2  @Service
3  public class EmailService {
4      private static final Logger LOGGER = Logger.getLogger(
5          EmailService.class.getName());
6      @Autowired private JavaMailSender javaMailSender;
7      ...
8      /**
9       * Method sends email or throws exception if failure occurs
10     *
11     * @param sender of the message
12     * @param recipient of the message
13     * @param subject of the message
14     * @param text of the message
15     * @throws MessagingException as {@link MessagingException}
16     */
17     @Async
18     public void sendEmail(String sender, String recipient, String
19         subject, String text)
20         throws MessagingException {
21         try {
22             SimpleMailMessage msg = new SimpleMailMessage();
23             msg.setFrom(sender);
24             msg.setTo(recipient);
25             msg.setSubject(subject);
26             msg.setText(text);
27
28             javaMailSender.send(msg);
29         } catch (Exception e) {
30             LOGGER.info("Failed to send email message to recipient:" +
31                 recipient);
32             throw new MessagingException("E-mail was not sent");
33         }
34         LOGGER.info("Email message was successfully sent to recipient
35             : " + recipient);
36     }
37 }

```

Kód 3.3: Metóda využívaná na posielanie správ pomocou e-mailovej pošty

3.4.7 Gradle

Gradle je open-source nástroj na automatické zostavenie aplikácie, ktorý je navrhnutý, aby bol dostatočne flexibilný na zostavenie takmer každého typu softwaru.

Medzi dôležité vlastnosti gradle patria:

- Vysoký výkon

Gradle sa vyhýba zbytočnej práci spúšťaním iba potrebných úloh, ktoré je potrebné spustiť, pretože sa zmenili ich vstupy alebo výstupy.

- JVM

Gradle funguje na JVM a na jeho použitie je potrebné nainštalovanie JDK (Java Development Kit).

- podpora IDE (Integrated Development Environment)

Táto podpora zahŕňa aj IDE, ktoré bolo použité pri vývoji webovej aplikácie, konkrétne IntelliJ IDEA. [19]

Nástroj Gradle bol použitý pre automatické zostavenie aplikácie, a to konkrétne verzia 6.3. Konfigurácia, závislosti od externých knižníc a všetky potrebné náležitosti sú obsiahnuté v súbore build.gradle. Pomocou jednoduchých príkazov je možné generovať dokumentáciu aplikácie, spustiť automatickú úpravu formátovania kódu, ale aj vytvorenie spustiteľného JAR (Java Archive) súboru.

3.4.7.1 Spotless

Spotless je open-source software, ktorý slúži aj ako plugin pre Gradle. Spotless je využívaný na kontrolu formátovania kódu, ale aj na jeho automatickú opravu. Disponuje bohatou konfiguráciou. Umožňuje nastavenie formátovania jednotlivých súborov aplikácie. Pri formátovaní textu je potrebné dodať pravidlá, akými sa má Spotless pri exekúcii riadiť, a prípadne nájdené chyby opraviť. Pravidlá formátovania sa dajú nastaviť podľa vlastných preferencií, ale umožňuje použitie aj známych formátovacích praktík.

Na formátovanie zdrojových súborov Javy boli použité pravidla podľa Google Java Format. Tento štýl formátovania je Googlom považovaný za štandardný. [20]

Práve vďaka nástroju Spotless bolo zachované konzistentné formátovanie aplikácie počas celého vývoja. Nastavenia Spotless sú vizuálne zobrazené v nasledujúcom bloku kódu 3.4.

```
1 spotless {
2   java {
3     target 'src/**/*.java'
4     googleJavaFormat()
5   }
6   format 'css', {
7     target 'frontend/styles/**/*.css'
8     endWithNewline()
9     indentWithSpaces(4)
10    trimTrailingWhitespace()
11  }
12  format 'gradle', {
13    target '*.gradle'
14    endWithNewline()
15    indentWithTabs()
16    trimTrailingWhitespace()
17  }
18  format 'markdown', {
19    target '*.md', 'guides/**/*.md'
20    endWithNewline()
21    indentWithSpaces(4)
22    trimTrailingWhitespace()
23  }
24  format 'yaml', {
25    target '*.yaml'
26    endWithNewline()
27    indentWithSpaces(2)
28    trimTrailingWhitespace()
29  }
30  format 'misc', {
31    target '.env.example',
32           '.gitattributes',
33           '.gitignore',
34           'gradle/wrapper/gradle-wrapper.properties',
35           'src/main/resources/application.properties'
36    endWithNewline()
37    trimTrailingWhitespace()
38  }
39 }
```

Kód 3.4: Nastavenia pluginu Spotless pre aplikáciu

3.4.7.2 Javadoc

Podobne ako Java bol aj Javadoc prvýkrát predstavený verejnosti spoločnosťou Sun Microsystems.

Javadoc nástroj synteticky analyzuje deklarácie a dokumentačné komentáre v zdrojových súboroch Javy a produkuje korešpondujúce HTML (HyperText Markup Language) stránky opisujúce verejné a chránené triedy, vnorené triedy, rozhrania, konštruktory, metódy a atribúty. Môže byť použitý na generovanie dokumentácie pre API alebo na implementačnú dokumentáciu zdrojových kódov. [21]

3. IMPLEMETÁCIA FUNKČNÉHO PROTOTYPU

Javadoc bol použitý v aplikácii na automatické generovanie dokumentácie. Javadoc je považovaný za štandardný nástroj na dokumentáciu Java kódu, preto bol výber tohto nástroja jednoznačný. V bloku kódu 3.5 sa nachádza zobrazená konfigurácia Javadoc pre webovú aplikáciu.

```
1 javadoc {
2     options.encoding = 'UTF-8'
3     options.docEncoding = 'UTF-8'
4     options.charSet = 'UTF-8'
5     source = sourceSets.main.allJava
6     classpath = project.sourceSets.main.runtimeClasspath
7     options.addStringOption("sourcepath", "")
8     options.addStringOption('Xdoclint:all', '-quiet')
9     options.addStringOption('Xwerror', '-quiet')
10 }
```

Kód 3.5: Konfigurácia Javadoc pre Gradle

3.5 Frontend

3.5.1 Vaadin

Vaadin je Java framework určený pre tvorbu užívateľské rozhrania. Ide o open-source software zameraný na frontendovú časť aplikácií. Vaadin ponúka veľa výhod a značne uľahčuje prácu vývojárom. Z cela najväčšou výhodou Vaadinu je fakt, že nie je potrebná znalosť technológií ako JavaScript, HTML alebo CSS k vybudovaniu plne funkčných webových aplikácií. Treba ale podotknúť, že spomínané technológie môžu byť v projekte jednoducho implementované. Väčšina frontendovej časti webovej aplikácie bola písaná práve za pomoci frameworku Vaadin, vďaka ktorému mohla byť takmer celá implementácia realizovaná v programovacom jazyku Java. Aj vďaka tejto unikátnej funkcionalite bol Vaadin framework zvolený ako hlavný nástroj na vývoj frontendovej časti aplikácie.

Práca s frontendovými technológiami ako JavaScript, HTML alebo CSS môže byť pre Java vývojárov vyčerpávajúca a časovo náročná. Vo Vaadine sú všetky UI (User Interface) prvky komponované do tzv. webových komponentov.

Počas vývoja webovej aplikácie bol použitý Vaadin verzie 14.4.4. Ide o jedinú tzv. LTS (Long Term Support) verziu. Verzia softwaru s týmto označením má garantovanú dlhotrvajúcu podporu od sprostredkovateľa.

3.5.1.1 Webové komponenty

Podľa [22] sú webové komponenty sada rôznych technológií, ktoré umožňujú vytvárať znova použiteľne elementy a využívať ich vo webových aplikáciách. Funkčnosť takýchto elementov je oddelená od zbytku kódu.

Jedna z možností, ktoré Vaadin ponúka je aj vytvorenie vlastných webových komponentov. Vývojár tak môže použiť vlastné definované webové komponenty, komponenty ponúkané priamo Vaadinom, ale aj webové komponenty vytvorené inými užívateľmi.

Systém budovania užívateľského rozhrania pomocou skladania webových komponentov do seba bol využívaný počas celého vývoja frontendovej časti webovej aplikácie. Počas implementácie boli navrhnuté webové komponenty, vďaka ktorým sa predišlo opakovaniu kódu, ale bola aj zvýšená jeho čitateľnosť.

Za webový komponent možno považovať aj triedu, ktorá má za úlohu rozšíriť funkcionality už existujúceho webového komponentu.

Trieda rozširujúca vstavaný upload komponent Vaadinu je zobrazená v bloku kódu 3.6.

```

1  /** Class represents my upload which is adding {@link DomEvent}
2     to add functionality of removing file */
3  public class MyUpload extends Upload {
4     public MyUpload() {
5         super();
6     }
7     /** This class represent file remove event */
8     @DomEvent("file-remove")
9     public static class FileRemoveEvent extends ComponentEvent<
10        Upload> {
11
12        private final String fileName;
13
14        public FileRemoveEvent(
15            Upload source,
16            boolean fromClient,
17            @EventData("event.detail.file.name") JreJsonString
18            fileNameJson) {
19            super(source, fromClient);
20            fileName = fileNameJson.getString();
21        }
22        public String getFileName() { return fileName; }
23    }
24
25    /**
26     * Method registers a new method to file remove event class
27     *
28     * @param listener new event listener as ComponentEventListener
29     *        {@link FileRemoveEvent}
30     * @return registration as {@link Registration}
31     */
32    public Registration addFileRemoveListener(
33        final ComponentEventListener<FileRemoveEvent> listener) {
34        return super.addListener(FileRemoveEvent.class, listener);
35    }
36 }

```

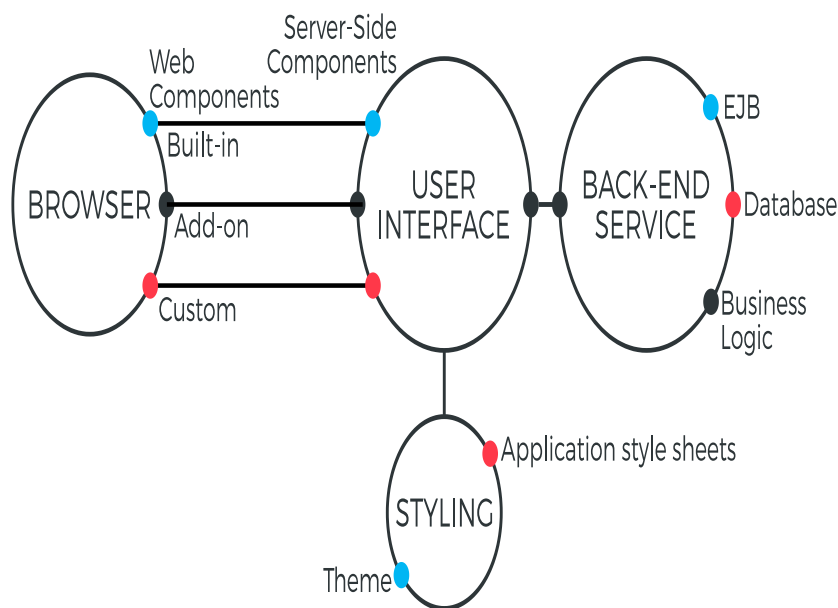
Kód 3.6: Trieda rozširuje funkcionality Vaadin upload komponentu

3.5.1.2 Architektúra

Podľa [23] zahŕňa Vaadin architektúra nasledujúce vlastnosti:

- Typovo bezpečne Java UI API komponenty na strane serveru, ktoré uľahčujú používanie webových komponentov
- Automatizovaná obojstranná komunikácia medzi serverom a prehliadačom, ktorá:
 - Ponúka Java vývojárom prístup ku všetkým moderným webovým vylepšeniam.
 - Zjednodušuje prepojenie UI s dátami pomocou robustného Java backendu, namiesto používania tradičnej REST komunikácie.
- Obojsmerná prepojenie dát: keď sa zmení užívateľské rozhranie na jednej zo strán klient/server, tak sa zmeny prejavujú automaticky na druhej strane

Na obrázku 3.4 je vizuálne zobrazená štruktúra Vaadinu.



Obr. 3.4: Štruktúra Vaadin frameworku podľa [23]

3.5.2 Css

Podľa [24] sú kaskádové štýly alebo CSS (Cascading Style Sheets) v informatike jazyky používané na opis dokumentov napísaných v HTML alebo XML. CSS opisuje ako by mali byť elementy zobrazované na obrazovke, papieri a iných médiach.

V aplikácii bolo CSS použité na štylizáciu vizuálnej stránky. Pomocou anotácie `@CssImport` boli importované CSS súbory s konfiguráciou jednotlivých tried. Pomocou tzv. `@media` anotácie bolo upravené správanie webovej stránky pri rôznych typoch obrazoviek.

V bloku kódu 3.7 možno vidieť upravené štýly pre Vaadin upload komponentu.

```
1 :host{
2
3   --lumo-primary-text-color: black;
4   --lumo-border-radius: 10px;
5   border: var(--custom-border);
6   cursor: pointer;
7
8 }
```

Kód 3.7: CSS nastavenia pre upload komponent

3.5.3 Javascript

Podľa [25] je JavaScript (JS) multiplatformový objektovo orientovaný skriptovací jazyk používaný na vytváranie interaktívnych webových stránok. Javascript je typicky interpretovaný vo webovom prehliadači na strane klienta, avšak existujú aj pokročilé verzie JS ako Node.js, ktoré sú interpretované na strane serveru, a ponúkajú väčšiu funkcionálnu webovej stránke.

JavaScript je neoddeliteľnou súčasťou frontendovej časti aplikácie. Webová aplikácia používa JavaScript prostredníctvom Vaadin frameworku. Napriek faktu, že Vaadin framework ponúka dobrú integráciu JS skriptov nebola táto funkcia príliš využívaná.

Integrácia niektorých API a Vaadinom je však možná iba pomocou JavaScript skriptov, a preto bolo pre niektoré prípady potrebné využitie čistého JavaScriptu.

Webová aplikácia používa Google Places API, ktoré má za úlohu našepťovania adresy užívateľovi. Integrácia tohto API bola realizovaná pomocou JS skriptu. Dôležitá časť webovej aplikácie, ktorej úlohou je získanie údajov po zvolení adresy užívateľom možno nájsť v bloku kódu 3.8. Funkcia spúšťa udalosť `change-complete` a vracia tieto údaje: krajina, mesto, rovnobežka a poludník.

3. IMPLEMETÁCIA FUNKČNÉHO PROTOTYPU

```
1 class PaperInputPlace extends GestureEventListeners (
2     PolymerElement) {
3     ...
4
5     _onChangePlace(e) {
6         var pl = this._places.getPlace();
7         if (pl.geometry) {
8             var p = this._extractPlaceInfo(pl, this.$.nativeInput.value
9             );
10            this._setPlace(p);
11            this._invalid = false;
12            this._setInvalid(false);
13            this._setLatLng({
14                lat: p.latLng.lat,
15                lng: p.latLng.lng
16            });
17
18            this._value = this.$.nativeInput.value;
19            this.value = {
20                search: this.$.nativeInput.value,
21                place_id: p.place_id,
22                latLng: {
23                    lat: p.latLng.lat,
24                    lng: p.latLng.lng
25                }
26            };
27
28            this.placeJSON="{ \"place\": {\"name\":\""+p.search+"\", \"
29            latLng\":{\"lat\":\""+p.latLng.lat+\" ,\"long\":\""+p.latLng.lng+\"
30            }}\";
31            this.dispatchEvent(new CustomEvent('change-complete', {
32                detail: {
33                    country: p.basic.country,
34                    city: p.basic.city,
35                    lat: p.latLng.lat,
36                    lng: p.latLng.lng
37                }
38            }));
39            this.placeName=this._value.place_id;
40        }
41    }
42 }
```

Kód 3.8: JS funkcia, spúšťajúca udalosť

Webový komponent, ktorý je prepojený s týmto JS skriptom je zobrazený v nasledujúcom bloku kódu 3.9, a zobrazuje prepojenie JavaScriptu a Vaadinu. Toto prepojenie je možné iba vďaka anotácii @JsModule ponúkanou priamo Vaadin frameworkom.

```
1 /** Class represent place field connected to google api */
2 @Tag("paper-input-place")
3 @NpmPackage(value = "paper-input-place", version = "2.0.1")
4 @JsModule("./js/paper-input-place.js")
5 public class PlaceAutocomplete extends Component {
6
7     ...
8
9     @DomEvent("change-complete")
10    public static class ValueChangeEvent extends ComponentEvent <
11        PlaceAutocomplete > {
12        private final String country;
13        private final String city;
14        private final Double lat;
15        private final Double lng;
16
17        public ValueChangeEvent(
18            PlaceAutocomplete source,
19            boolean fromClient,
20            @EventData("event.detail.country") String country,
21            @EventData("event.detail.city") String city,
22            @EventData("event.detail.lat") Double latitude,
23            @EventData("event.detail.lng") Double longitude) {
24            super(source, fromClient);
25
26            this.country = country;
27            this.city = city;
28            this.lat = latitude;
29            this.lng = longitude;
30        }
31        public String getCountry() { return country; }
32        public String getCity() { return city; }
33        public Double getLat() { return lat; }
34        public Double getLng() { return lng; }
35    }
36    ...
37 }
```

Kód 3.9: Vaadin trieda naviazaná na JS skript

3.6 Databáza

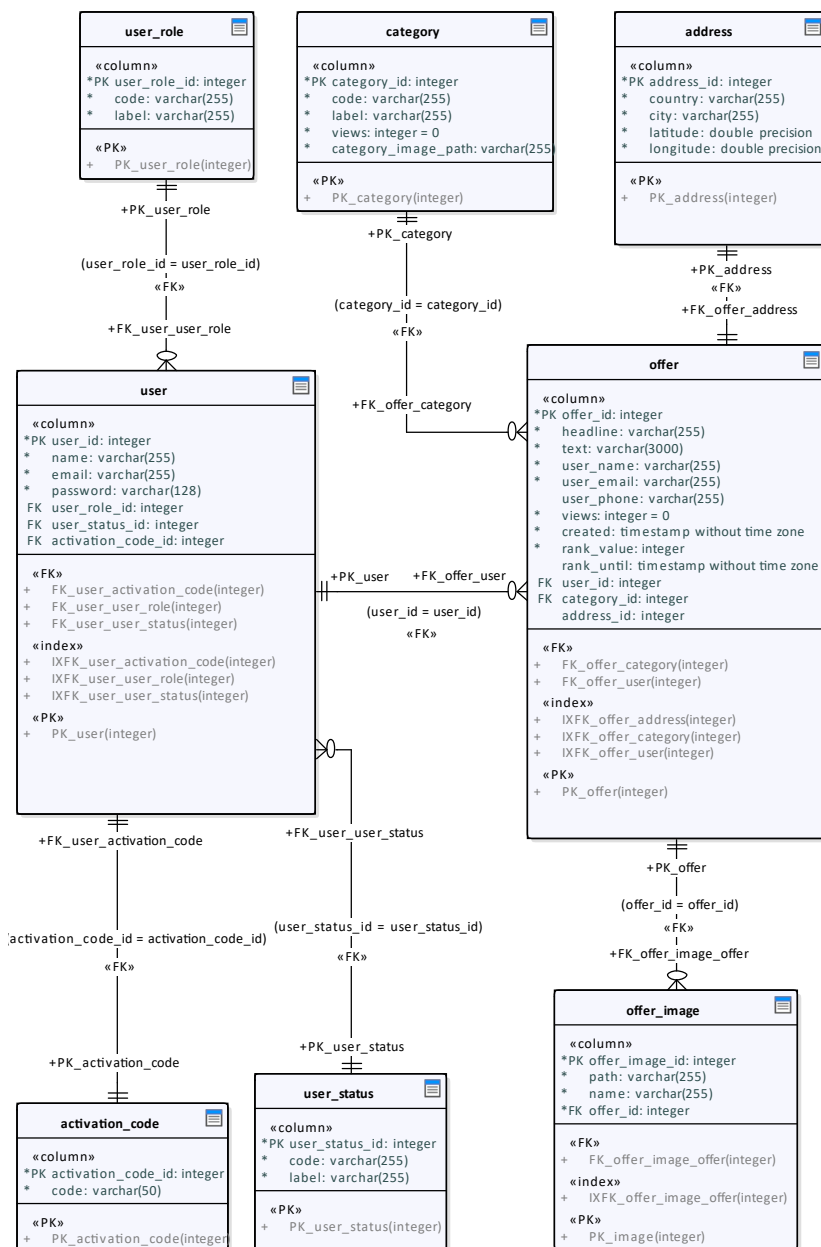
PostgreSQL bol zvolený ako databázový systém pri vývoji webovej aplikácie z viacerých dôvodov:

- open-source projekt.
- objektovo-relačná databáza.
- dobrá znalosť a predchádzajúce skúsenosti autora s týmto databázovým systémom.

3. IMPLEMETÁCIA FUNKČNÉHO PROTOTYPU

V systéme sú vytvorené databázové funkcie navrhnuté špeciálne pre webovú aplikáciu. Účel týchto funkcií je zameraný na rozloženie výpočtovej záťaže v aplikácii.

Na obrázku 3.5 je zobrazený používaný databázový model pri implementácii. Tento model pokrýva všetky funkčné požiadavky z kapitoly 2.1 a kladie dôraz na jeho rozšíriteľnosť v budúcnosti.



Obr. 3.5: Databázový model webovej aplikácie

3.7 Autentizácia a autorizácia

Ako už bolo spomenuté v kapitole 3.4.4 je autorizácia a autentizácia v rámci aplikácie zabezpečovaná Spring Security frameworkom. Únik citlivých dát užívateľov je celosvetový problém tejto generácie.

Bezpečnosť tohto prototypového modelu nie je v porovnaní s veľkými systémami na trhu konkurencieschopná. Neoplýva populárnou dvojfázovou autentifikáciou ako je u väčších systémov známe, avšak aj napriek tomuto faktu bolo usilované, o čo najlepšie zabezpečenie užívateľských údajov.

Ako identifikátor užívateľského účtu bola zvolená e-mailová adresa. Použitie e-mailovej adresy obsahuje jednu veľkú výhodu oproti užívateľskej prezývke alebo užívateľskému menu. Užívateľova totožnosť môže byť jednoducho overená prostredníctvom aktivačného e-mailu. Medzi ďalšie výhody sa radí aj fakt, že užívateľ nepríde do bodu, kedy je ním zvolené užívateľské meno obsadené. Tento fakt môže užívateľa od dokončenia procesu registrácie odradiť.

Aplikácia používa iba jednofázovú autentizáciu, a preto po užívateľovi požaduje heslo, ktoré musí byť najmenej 8 znakov dlhé s povinnosťou aspoň jedného numerického znaku.

V bloku kódu 3.10 je zobrazená trieda zodpovedajúca za nastavenia prístupových práv v systéme.

```

1  /** Class represents web security configurer adapter */
2  @EnableWebSecurity
3  @Configuration
4  public class SecurityConfiguration extends
      WebSecurityConfigurerAdapter {
5
6      private static final String LOGIN_URL = "/login";
7
8      @Autowired
9      private CustomAuthenticationProvider authProvider;
10
11     /**
12      * Method authenticates user
13      *
14      * @param auth {@link AuthenticationManagerBuilder}
15      */
16     @Override
17     protected void configure(AuthenticationManagerBuilder auth) {
18         auth.authenticationProvider(authProvider);
19         auth.eraseCredentials(false);
20     }
21
22     /**
23      * Method require login to access internal pages and configure
24      * login form.
25      *
26      * @param http {@link HttpSecurity}
27      * @throws Exception as {@link Exception}
28      */

```

3. IMPLEMETÁCIA FUNKČNÉHO PROTOTYPU

```
28  @Override
29  protected void configure(HttpSecurity http) throws Exception {
30      http.csrf()
31          .disable()
32          .requestCache()
33          .requestCache(new CustomRequestCache())
34          .and()
35          .authorizeRequests()
36          .antMatchers("/admin/**")
37          .hasAuthority(UserRole.ADMIN_ROLE_CODE)
38          .antMatchers(
39              "/" + MyOffersView.URL + "**",
40              "/" + AddOfferView.URL + "**",
41              "/" + PasswordChangeView.ROUTE + "**",
42              "/" + EditOfferView.ROUTE + "**")
43          .authenticated()
44          .requestMatchers(SecurityUtils::
45              isFrameworkInternalRequest)
46          .permitAll()
47          .antMatchers("**")
48          .permitAll()
49          .anyRequest()
50          .authenticated()
51          .and()
52          .formLogin()
53          .loginPage(LOGIN_URL)
54          .permitAll();
55  }
56  ...
57
58 }
```

Kód 3.10: Trieda autentizuje užívateľa a konfiguruje prístupové práva k jednotlivým stránkam v aplikácii

Za pomoci tejto triedy je možné nastavenie prístupových práv k jednotlivým interným stránkam webovej aplikácie. V riadku 34. je prefixový fragment url adresy, ku ktorej má prístup iba užívateľ s autoritou admin. V riadkoch 37. až 40. sú nakonfigurované url adresy takým spôsobom, že sú prístupné iba pre autentifikovaného užívateľa. V prípade, že užívateľ nemá rolu admin, alebo nie je prihlásený v aplikácii a chce prístup na stránku, kde nemá oprávnenie je aplikáciou presmerovaný na prihlasovací formulár.

3.8 Dokumentácia

Súčasťou každého kvalitného softwarového projektu je dokumentácia. Na základe dokumentácie môže byť napríklad prevedená analýza existujúceho systému, a vypracovaný návrh jeho vylepšení do budúcnosti.

Prechod na nový systém je pre vývojára s detailnou dokumentáciou oveľa hladší a jednoduchší ako v prípade, že ide o neprehľadnú alebo dokonca neexistujúcu dokumentáciu.

Podľa [26] existujú v programovacom jazyku Java tri typy komentárov.

- `/* text */`

Kompilátor ignoruje všetko od `/*` po `*/`.

- `//text`

Kompilátor ignoruje všetko od `//` až po koniec riadku.

- `/** documentation */`

Toto je dokumentačný komentár a vo všeobecnosti sa nazýva doc comment. Javadoc nástroj využíva doc comments pri automatickom generovaní dokumentácie kódu.

Posledný z uvedených komentárov bol použitý k dokumentácii kódu webovej aplikácie. V programe sú popísané všetky podrobnosti ohľadom tried, rozhraní a metód. Po konfigurácii Javadoc nastavení v nástroji Gradle v kapitole 3.4.7.2 bolo generovanie dokumentácie aplikácie výrazne zjednodušené.

3.9 Použité princípy

Medzi nefunkčnými požiadavkami systému bola v kapitole 2.2.3 spomenutá aj rozširiteľnosť systému. V závislosti od týchto požiadaviek bol počas implementácie webovej aplikácie kladený dôraz o jednoduchú rozširiteľnosť v budúcnosti.

Realizácia a písanie zdrojového kódu sa riadili viacerými princípmi. Medzi hlavné princípy možno zaradiť:

- Písanie čo najjednoduchšieho kódu

Počas písania kódu bolo obmedzené použitie zložitých programovacích konštrukcií, aby bol kód pre aktuálnych, ale aj budúcich vývojárov čo najlepšie čitateľný a zrozumiteľný.

- Neopakovanie vlastného kódu

Cieľom bola znova použiteľnosť už napísaných funkcií, tried a komponentov. Vďaka tomuto princípu sa predišlo repetitívnosti kódu v programe.

- YAGNI (You Aren't Going To Need It)

Princíp sa zameriava na písanie iba potrebnej funkcionality. Porušenie princípu nastáva vtedy, keď sa vývojár snaží riešiť problém, ktorý sa v aplikácii nenachádza. Písanie abstraktných programovacích konštrukcií ma častokrát za následok porušenie spomínaného princípu.

- SOLID princípy

Podľa [27] ide o objektovo-orientované softwarové dizajnové vzory. SOLID je skratka pre nasledujúcich 5 princíпов:

- Single Responsibility Principle
Každá trieda by mala byť zodpovedná za jednu časť alebo funkcionálnosť systému.
- Open-Closed Principle
Softwarové komponenty by mali byť rozšíriteľné bez potrebnej modifikácie už existujúcich riešení.
- Liskov Substitution Principle
Rodičovské triedy by malo byť možné nahradiť ich potomkami bez narušenia chodu systému.
- Interface Segregation Principle
Užívatelia by nemali byť závislí od rozhraní, ktoré nepoužívajú.
- Dependency Inversion Principle
Vysoko úrovňové moduly by nemali závisieť od nízko úrovňových modulov, ale oba by mali závisieť od abstrakcií.

3.10 Súhrn

Cieľom tejto kapitoly bolo oboznámiť čitateľa s implementačnými technológiami, princípmi a postupmi, ktoré boli aplikované počas vývoja webovej aplikácie slúžiacej na inzerciu práce remeselníkov.

Na uľahčenie vysvetlenia danej problematiky čitateľovi boli použité ilustračné obrázky, grafy, ale nechýbali ani ukážky konfiguračných nastavení a fragmenty zdrojového kódu aplikácie. Jednotlivý výber technológií bol výstižne argumentačne podložený, aby presvedčil užívateľa o jeho korektnosti.

Zdrojový kód prototypu webovej aplikácie slúžiacej na inzerciu práce remeselníkov je dostupný na adrese <https://gitlab.fit.cvut.cz/vecerpe5/expertservices>.

Testovanie

Kapitola je zameraná na testovanie aplikácie. Táto kapitola opisuje postup, ktorý bol použitý na odhalenie potencionálnych chýb v aplikácii a obmedzenie ich výskytu v budúcnosti. Testovanie bolo rozdelené do 3 častí. V prvej časti boli vytvorené automatické testy. Následne bola upriamená pozornosť na užívateľské testovanie, a na záver bol využitý opäť nástroj Lighthouse, ktorý bol aplikovaný v kapitole 1.4 pri hodnotení už existujúcich riešení.

4.1 Automatické testy

V aplikácii bolo realizované automatické testovanie na úrovni jednotlivých komponentov.

Pri testovaní jednotlivých komponentov predstavujú automatické testy veľkú úsporu. Takéto testovanie nachádza najväčší úžitok práve pri refaktoringu kódu, alebo pri pridaní nových závislostí od už existujúcich komponentov.

Na zjednodušenie procesu automatického testovania boli použité nasledujúce frameworky:

- **JUnit**

JUnit je najrozšírenejší Java framework zameraný na testovanie menších blokov kódu v programe. Korektný test v JUnit by mal testovať jednu funkcionálnu časť v systéme, byť kompletné nezávislý od okolia a opakovateľný.

- **Mockito**

Podľa [28] slúži Mockito framework na tzv. mockovanie tried. Ide o proces, v ktorom nie je volaná konkrétna instancia danej triedy, ale jej Mock. Mock je náhrada za reálny objekt, ktorá získava dáta o tomto objekte.

Pomocou tohto frameworku je možné prednastavenie správania daných objektov.

4. TESTOVANIE

Nasledujúci blok zdrojového kódu 4.1 testuje metódu, ktorá aktivuje užívateľský účet po registrácii. Metóda predstavuje základnú funkcionálnu pre systém, keďže aktivácia užívateľského účtu je neoddeliteľným krokom k vytvoreniu inzerátu.

V metóde sú použité anotácie `@Mock`, ktoré majú nastavené správanie podľa potreby. V riadkoch 26. až 29. je vidieť ako sa majú jednotlivé repozitáre pri zavolaní konkrétnych metód zachovať. Týmto postupom sa zamedzí novej chybe pri komunikácii servisnej triedy s repozitárom, a v prípade zlyhania testu je možná príčina chyby detailnejšie vymedzená.

```
1 @ExtendWith(MockitoExtension.class)
2 public class UserServiceTest {
3     @Mock private UserRepository userRepository;
4     @Mock private UserRepository userStatusRepository;
5     @InjectMocks private UserService userService;
6     ...
7     @Test
8     public void activateUser() {
9         // arrange
10        ActivationCode activationCode = new ActivationCode();
11        User user = new User();
12
13        UserStatus deactivatedUserStatus = new UserStatus();
14        deactivatedUserStatus.setCode(UserStatus.DEACTIVATED_STATUS_CODE);
15
16        UserStatus activatedUserStatus = new UserStatus();
17        activatedUserStatus.setCode(UserStatus.ACTIVATED_STATUS_CODE);
18
19        user.setUserStatus(deactivatedUserStatus);
20
21        given(userRepository.findByActivationCode(activationCode)).
22        willReturn(user);
23        given(userRepository.save(user)).willReturn(user);
24        given(userStatusRepository.findUserStatusByCode(UserStatus.
25        ACTIVATED_STATUS_CODE))
26        .willReturn(activatedUserStatus);
27
28        // action
29        try {
30            userService.activateUser(activationCode);
31        } catch (UserService.AuthException e) {
32            fail();
33        }
34
35        // assert
36        Assertions.assertEquals(UserStatus.ACTIVATED_STATUS_CODE,
37        user.getUserStatus().getCode());
38    }
39 }
```

Kód 4.1: Metóda testuje aktivovanie užívateľského účtu

4.1.1 Spúšťanie automatických testov

Automatické testy sú spúšťané pri každom zostavení aplikácie. Tieto testy sa spúšťajú aj pri každom uložení novej verzie do GitLab repozitára. Kontinuálna integrácia ponúkaná GitLabom, bola nakonfigurovaná tak, aby zostavila aplikáciu a skontrolovala formátovanie kódu. Vo fáze build je aplikácia skompilovaná a pripravená na spustenie. Vo fáze test je skontrolované formátovanie kódu a spustenie automatických testov.

Konfigurácia kontinuálnej integrácie je uvedená v bloku kódu 4.2.

```
1 image: gradle:jdk11
2
3 stages:
4   - build
5   - test
6
7 build:
8   stage: build
9   script:
10    - ./gradlew assemble
11   artifacts:
12     paths:
13       - build/libs/*.jar
14
15 test:
16   stage: test
17   script:
18     - ./gradlew check
```

Kód 4.2: Zdrojový kod konfiguračného súboru kontinuálnej integrácie na GitLabe

GitLab používa Docker pri spúšťaní skriptov. V prvom riadku je zadenované, aké technológie majú byť použité. V našom prípade ide o nástroj na zostavenie aplikácie Gradle a Java 11.

4.1.2 Pokrytie kódu

Pokrytie kódu testami, ktoré bolo merané pomocou IntelliJIDEA IDE konkrétne funkciou *Coverage*, je vidieť v tabuľke 4.1. V tabuľke pokrytia sú uvedené jednotlivé balíčky, ktoré sú obsiahnuté v balíčku backend.

Nižšie celkové hodnotenie je spôsobené najmä balíčkom Security a Entity. V balíčku Security sa nachádzajú metódy konfigurujúce prístup k jednotlivým stránkam webovej aplikácie a v rámci automatického testovania ich nebolo možné rozumne otestovať. Testovanie prístupu k jednotlivým stránkam bolo realizované ručne.

Podiel na nižšom hodnotení má taktiež balíček Entity, v ktorom sa nachádzajú triedy reprezentujúce jednotlivé tabuľky v databáze. Absencia otestovania jednotlivých getterov a setterov má za následok nižšie celkové pokrytie.

Názov	Pokrytie tried	Pokrytie metód	Pokrytie riadkov
<i>Entity</i>	100 %	60 %	39 %
<i>Enums</i>	100 %	100 %	92 %
<i>Repositories</i>	100 %	100 %	100 %
<i>Services</i>	100 %	100 %	95 %
<i>Utils</i>	100 %	100 %	100 %
<i>Fileio</i>	100 %	100 %	93 %
<i>Security</i>	60 %	73 %	62 %
Celkovo	93 %	77 %	76 %

Tabuľka 4.1: Tabuľka pokrytia backend časti kódu

4.2 Uživatelské testovanie

Komplikované uživatelské testovanie je mrhaním zdrojov. Najlepšie výsledky sú pri testovaní nie viac ako 5 užívateľov. [29]

Na základe predchádzajúceho citátu Jakoba Nielsena a jeho štúdie o ideálnom počte testovacích užívateľov, bolo vybratých práve 5 užívateľov, a to vo veku 20, 23, 26, 35 a 42 rokov. Pri výbere užívateľov bol kladený dôraz na výber reprezentantov z jednotlivých vekových skupín. Bolo vypracovaných 5 scenárov, ktoré pokrývajú hlavnú funkcionality webovej aplikácie.

Záznamy z testovania nie sú, bohužiaľ, súčasťou tejto bakalárskej práce, avšak napriek tomu bol užívateľ počas prechodu jednotlivých testovacích scenárov pod drobnohľadom autora, ktorý si všetky informácie dôkladne zaznamenával. Záznamy do textového dokumentu nie sú ideálna forma poznámok, častokrát sú subjektívne a prípadná analýza sa podľa nich realizuje ťažšie. V prípade budúceho uživatelského testovania aplikácie budú všetky testovacie scenáre nahrávané a bude vytvorený videozáznam.

Testovacie scenáre zdieľali nasledujúcu štruktúru:

Ciel:

Čo si kládol testovací scenár za cieľ.

Príbeh:

Príbeh mal za úlohu nasimulovať reálnu situáciu zo života.

Úlohy:

Kroky, v ktorých mal užívateľ postupovať.

Výsledky:

Zhodnotenie výsledkov pozorovaných pri prechode testovacieho scenára užívateľom.

Tabuľka splnenia úloh radí užívateľov podľa veku od najmladšieho po najstaršieho t.j. užívateľ 1 má 20 rokov a užívateľ 5 má 42 rokov.

4.2.1 Scenár A

Cieľ:

Tento scenár má za cieľ otestovať užívateľovu schopnosť vytvoriť jednoduchý inzerát v aplikácii.

Príbeh:

Kvôli koronavírusovej pandémie, ktorá postihla celý svet sa Váš život v práci výrazne skomplikoval. Ako remeselník potrebujete veľa kontaktov, a v prípade ich nedostatku Vám chýba práca. Rozhodli ste sa pokúsiť o zviditeľnenie v online priestore, a preto ste si vybrali práve náš inzertný portál, ktorý slúži presne pre ľudí ako vy. Pokúste sa vytvoriť inzerát, ktorý najlepšie opisuje služby, ktoré ponúkate.

Úlohy:

1. Registrujte sa.
2. Aktivujte si užívateľský účet.
3. Vytvorte inzerát.

Výsledky:

Užívatelia zvládli všetky kroky scenáru bez väčších problémov. Užívateľ 2 sa pokúsil o prihlásenie bez predchádzajúcej aktivácie svojho nanovo vytvoreného užívateľského účtu. Systém užívateľa automaticky upozornil chybovou hláškou, že daný účet je deaktivovaný, a je potrebná jeho aktivácia pomocou aktivačného linku, ktorý mu bol odoslaný na e-mail.

Užívateľ 5 chcel vytvoriť inzerát bez adresy a nebol aplikáciou upozornený, že adresa je neplatná. Nakoniec užívateľ skúsil adresu vyplniť a zvládol vytvoriť inzerát. Chyba, ktorá sa počas testu vyskytla bola opravená pridaním automatickej chybovej hlášky v prípade, že je zadaná adresa neplatná.

úlohy/užívatelia	užívateľ 1	užívateľ 2	užívateľ 3	užívateľ 4	užívateľ 5
úloha 1	✓	✓	✓	✓	?
úloha 1	✓	✓	✓	✓	✓
úloha 1	✓	✓	✓	✓	✓

Tabuľka 4.2: Scenár A tabuľka splnenia úloh

✓ = Úspech | ? = Úspech s komplikáciami | X = Neúspech

4.2.2 Scenár B

Ciel:

Cielom tohto testovacieho scenáru je otestovať, či je užívateľ schopný inzerát upraviť, nastaviť profilovú fotografiu a odhlásiť sa z inzertného portálu podľa zadania.

Príbeh:

Práve ste nahrali svoj inzerát do systému, ale spomenuli ste si, že ste zabudli vyplniť Vaše telefónne číslo a fotografiu. Pokúste sa túto chybu napraviť vašou úpravou spomínaného inzerátu.

Úlohy:

1. Upravte inzerát.
2. Zmeňte telefónne číslo na 0123 456 789.
3. Pridajte fotografie, ktoré sa nachádzajú v priečinku fotografie.
4. Uložte editovaný inzerát.
5. Odhláste sa z vášho účtu.

Výsledky:

Každý jeden užívateľ úspešne absolvoval všetky kroky scenáru. Upravenie inzerátu bolo pre užívateľov dostatočne intuitívne, a nebolo potrebné sa hlbšie zamýšľať. Telefónne číslo aj pridanie fotografií prebehlo v poriadku. Odhlásenie z aplikácie bolo pre dvoch užívateľov nejednoznačné. Užívatelia spočiatku nevedeli, kde sa tlačidlo odhlásiť nachádza. Po chvíli na to, ale obaja užívatelia prišli a úspešne sa odhlásili. Funkcia odhlásiť sa bola v aplikácii upravená, aby bola pre budúcich užívateľov prehľadnejšie označená.

úlohy/užívatelia	užívateľ 1	užívateľ 2	užívateľ 3	užívateľ 4	užívateľ 5
úloha 1	✓	✓	✓	✓	✓
úloha 2	✓	✓	✓	✓	✓
úloha 3	✓	✓	✓	✓	✓
úloha 4	✓	✓	✓	✓	✓
úloha 5	?	✓	✓	?	✓

Tabuľka 4.3: Scenár B tabuľka splnenia úloh

✓ = Úspech | ? = Úspech s komplikáciami | X = Neúspech

4.2.3 Scenár C

Ciel:

Scenár cieľi na vyhľadanie remeselníka. Vyhľadávanie pozostáva z viacerých krokov. Po úspešnom vyhľadaní, je testovaná funkcionálna kontaktovanie remeselníka.

Príbeh:

Pomaly sa končí jeseň a prichádza zima, a vy rozmýšľate ako sa na ňu najlepšie pripraviť. Minulý rok ste používali domáci krb, ktorý výrazne šetril náklady na energiu. Avšak po minuloročnom extenzívnom kúrení, potrebuje váš komín údržbu. Keďže sa bojíte výšok a nepoznáte žiadneho kominára v okolí, tak ste sa rozhodli použiť práve náš inzertný portál.

Úlohy:

1. Pozrite sa na úvodné remeslá, pokiaľ sa v nich nachádza remeslo “kominár” tak ho vyberte.
2. V prípade, že sa na úvodnej stránke príslušne remeslo nenachádza, skúste takéto remeslo vyhľadať vo vyhľadávacom okne ručne.
3. Zvoľte mesto Žilina.
4. Pokiaľ sa žiadna ponuka v tomto meste nenachádza zvýšte váš vyhľadávací okruh.
5. Zoradte ponuky od najpopulárnejších.
6. Kontaktuje 2. remeselníka prostredníctvom kontaktného formulára.

Výsledky:

Testovací scenár dokončili všetci užívatelia úspešne. Počas vyhľadávania, systém správne reagoval na užívateľský vstup.

úlohy/užívatelia	užívateľ 1	užívateľ 2	užívateľ 3	užívateľ 4	užívateľ 5
úloha 1	✓	✓	✓	✓	✓
úloha 2	✓	✓	✓	✓	✓
úloha 3	✓	✓	✓	✓	✓
úloha 4	✓	✓	✓	✓	✓
úloha 5	✓	✓	✓	✓	✓
úloha 6	✓	✓	✓	✓	✓

Tabuľka 4.4: Scenár C tabuľka splnenia úloh

✓ = Úspech | ? = Úspech s komplikáciami | X = Neúspech

4.2.4 Scenár D

Ciel:

Cieľom tohto scenáru je nasimulovať situáciu, kedy užívateľ dlhšie portál nepoužíval, a v dôsledku toho zabudol svoje prihlasovacie heslo. Scenár testuje funkčnosť resetovania hesla užívateľom.

Príbeh:

Uplynulo veľa času od Vašej poslednej návštevy nášho inzertného portálu. Váš inzerát stále púta pozornosť zákazníkov, ale momentálne nie ste schopný vykonávať Vaše povolanie, preto ste sa rozhodli inzerát vymazať. Avšak neviete si spomenúť na Vaše heslo, a preto sa ho rozhodnete resetovať.

Úlohy:

1. Pokúste sa resetovať Vaše heslo.
2. Prihláste sa vygenerovaným heslom, ktoré Vám prišlo na vašu e-mailovú adresu.
3. Zmeňte novo vygenerované heslo systémom na vami definované heslo.
4. Vymažte vami publikovaný inzerát.

Výsledky:

Testovací scenár bol spomedzi všetkých pre užívateľov najjednoduchší. Každý jeden užívateľ zvládol tento scenár v pomerne krátkom čase a hlavne bezproblémovo. Krátka skúsenosť, ktorú užívatelia počas testovania systému získali by mohla byť zodpovedná za bezproblémové zvládnutie testovacieho scenáru.

úlohy/užívatelia	užívatel 1	užívatel 2	užívatel 3	užívatel 4	užívatel 5
úloha 1	✓	✓	✓	✓	✓
úloha 2	✓	✓	✓	✓	✓
úloha 3	✓	✓	✓	✓	✓
úloha 4	✓	✓	✓	✓	✓

Tabuľka 4.5: Scenár D tabuľka splnenia úloh

✓ = Úspech | ? = Úspech s komplikáciami | X = Neúspech

4.2.5 Scenár E – Administrátorská časť

Cieľ:

Scenár testuje schopnosť administrátora spravovať inzeráty a užívateľov vo webovej aplikácii. Po následnom zablokovaní užívateľského účtu je kontrolovaná funkcionálna zabezpečujúca prístup do systému.

Príbeh:

Všimli ste si, že v systéme je užívateľ, ktorý disponuje podivne veľa inzerátmi. Po bližšej inšpekcii ste zistili, že zverejnené inzeráty porušujú pravidlá inzertného portálu, a preto ste sa rozhodli tento účet zablokovať. Pri prezeraní jednotlivých inzerátov ste objavili inzerát, ktorý nespĺňa pravidlá inzertného portálu a boli ste nútený ho vymazať.

Úlohy:

1. Prihláste sa s údajmi:
 - e-mail: admin
 - heslo: admin
2. Po rozkliknutí vášho účtu vyberte sekciu admin.
3. V možnosti Inzeráty vymažte ponuku užívateľa admin.
4. V možnosti Užívateľia zablokujte užívateľský účet, ktorý ste si v predchádzajúcich krokoch vytvorili.

Výsledky:

Ani tento užívateľský scenár nebol výnimkou a zvládol ho každý užívateľ. Počas testovania došlo k jednému problému. Užívateľ neklikol na tlačidlo Inzeráty, lebo bolo aplikáciou zvýraznené. Napriek tomu, že bolo tlačidlo zvýraznené sa užívateľ nenachádzal na správnej obrazovke. Táto skutočnosť bola opravená a aplikácia ďalej takéto tlačidlo chybne nezvýrazní, pokiaľ sa užívateľ nenachádza na danej obrazovke.

úlohy/užívateľia	užívateľ 1	užívateľ 2	užívateľ 3	užívateľ 4	užívateľ 5
úloha 1	✓	✓	✓	✓	✓
úloha 2	✓	✓	✓	✓	✓
úloha 3	✓	?	✓	✓	✓
úloha 4	✓	✓	✓	✓	✓

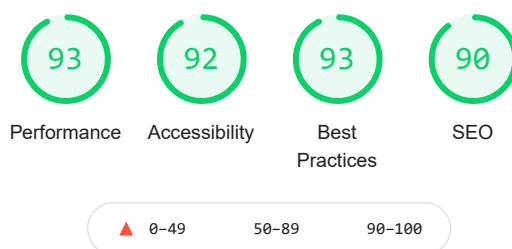
Tabuľka 4.6: Scenár E tabuľka splnenia úloh

✓ = Úspech | ? = Úspech s komplikáciami | X = Neúspech

4.3 Hodnotenie Lighthouse

Webová aplikácia bola podrobená detailnému hodnoteniu za pomoci nástroja Lighthouse, podobne ako už existujúce riešenia v kapitole 1.4. Analýza bola exekúovaná až po modifikovaní webovej aplikácie, ktoré prebehlo po užívateľskom testovaní na dosiahnutie, čo najlepšieho výsledku.

Na obrázku 4.1 je zobrazené hodnotenie pomocou nástroja Lighthouse.



Obr. 4.1: Lighthouse hodnotenie vytvorenej webovej aplikácie

Webová aplikácia dosiahla v každej jednej z kategórií nad 90 bodov, čo predstavuje výborný výsledok. Takéto hodnotenie nebolo dosiahnuté ani v jednom z už existujúcich riešení analyzovaných v kapitole 1.4.

Zhodnotenie

V tejto kapitole bude autorom zhodnotená výsledná implementácia webovej aplikácie. Použitelnosť webovej aplikácie bude analyzovaná na základe výsledkov testovania aplikácie v kapitole 4.

V druhej časti kapitoly budú uvedené možné vylepšenia do budúcnosti. Nevyhnutné funkcie systému, ktoré by pred nasadením do produkčného prostredia nemali chýbať, boli už okrajovo spomenuté v kapitole 2.1.

5.1 Zhodnotenie aplikácie

Webová aplikácia splnila všetky funkčné a nefunkčné požiadavky, ktoré boli na ňu v kapitole 2 kladené. Dizajn užívateľského rozhrania odpovedá návrhu užívateľského rozhrania v sekcii 2.6, ktorý bol počas implementácie striktné dodržiavaný. Vďaka použitým princípom a frameworkom počas realizácie je rozšírenie webovej aplikácie v budúcnosti pre programátorov značne uľahčené.

Na základe užívateľského testovania v sekcii 4.2 vyplynulo, že aplikácia je pre užívateľov dostatočne intuitívna a prehľadná. Aj napriek faktu, že užívatelia, ktorí testovali webovú aplikáciu pochádzali z rôznych vekových skupín, boli dosiahnuté uspokojivé výsledky, čo značí mieru flexibility webovej aplikácie.

Aplikácia získala v sekcii 4.3 výborné hodnotenie, čo je vynikajúci výsledok vzhľadom na fakt, že ešte nejde o plnohodnotnú produkčnú verziu.

Medzi najväčšie nevýhody momentálnej verzie webovej aplikácie sa radí obmedzená funkcionálnosť administrátorskej časti. Táto časť ešte nemôže byť považovaná za kompletnú. Chýbajú tu viaceré nastavenia, ktorými by mal administrátor takéhoto portálu ovládať. V administrátorskej sekcii boli implementované iba najpodstatnejšie funkcie, nevyhnutné pre spravovanie inzerátov a užívateľských účtov.

5.1.1 Použitelnosť výsledného riešenia

Podľa [30] je dôležité si uvedomiť, že použiteľnosť nie je jednodimenzionálna vlastnosť produktu, systému alebo užívateľského rozhrania. Použitelnosť je kombinácia faktorov zahŕňajúcich:

Intuitívny dizajn:

Bezproblémové porozumenie rozloženiu jednotlivých prvkov na obrazovke a navigovaniu na webovej stránke.

Pochopiteľnosť:

Ako náročne je pre užívateľa vykonať úlohu po prvýkrát.

Efektívnosť:

Ako rýchlo dokáže skúsený užívateľ vykonávať jednotlivé úlohy.

Zapamätateľnosť:

Kolko si dokáže užívateľ po návšteve zapamätať, aby bol pri budúcich návštevách efektívny.

Chybovosť:

Ako často urobí užívateľ pri používaní systému chybu, ako je závažná, a ako rýchlo sa dá napraviť.

Spokojnosť:

Do akej miery je činnosť príjemná.

Použitelnosť webovej aplikácie bola otestovaná v kapitole 4.2. Počas testovania užívatelia úspešne absolvovali všetky testovacie scenáre, bez väčších problémov. Nedostatky, ktoré boli užívateľmi počas testovania odhalené boli okamžite autorom opravené.

Jediný z faktorov, ktorý nebol poriadne overený je faktor zapamätateľnosti. Na uistenie sa či bol aj spomínaný faktor obsiahnutý, by bolo potrebné znovu otestovať aplikáciu tými istými užívateľmi. Opätovná realizácia užívateľského testovania tými istými užívateľmi nebola z časových dôvodov možná.

Zostávajúce faktory použiteľnosti boli počas užívateľského testovania dôkladne overené. Na intuitívnosť, pochopiteľnosť a spokojnosť bol kladený dôraz pri návrhu užívateľského rozhrania v kapitole 2.6. Tento návrh sa odzrkadlil na užívateľskom testovaní, kedy bol pre užívateľov inzertný portál intuitívny, pochopiteľný a celkový dojem užívateľov bol kladný. Užívatelia porozumeli princípom webovej aplikácie v relatívne krátkom čase a nasledujúce testovacie scenáre boli efektívne. Chyby, ktoré užívatelia počas používania systému urobili, boli analyzované a autorom práce opravené.

Z tohto testovania vyplýva, že použiteľnosť aplikácie je na dobrej úrovni.

5.2 Možné vylepšenia

V tejto sekcii budú autorom navrhnuté možné vylepšenia webovej aplikácie do budúcnosti.

Vzhľadom na komplexnosť a rozsah webovej aplikácie nebolo počas implementácie dosiahnuté produkčné riešenie, ktoré by mohlo už existujúcim riešeniam plnohodnotne konkurovať. Aplikácia sa momentálne nachádza iba v prototypovej podobe a k dosiahnutiu jej produkčnej podoby je potrebné rozšírenie jej funkcionalít.

5.2.1 Administrátorská časť

Ako už bolo spomenuté v sekcii 5.1 je administrátorská časť webovej aplikácie najviac zanedbaná. Administrátor dokáže vykonávať iba limitovaný počet operácií. V ideálnom scenári by bolo pred nasadením do produkcie vhodné prídanie nasledujúcich funkcií (všetky funkcie sú zamerané na správu užívateľského účtu):

- úprava e-mailovej adresy a mena
- vymazanie účtu

Z právneho hľadiska je užívateľ oprávnený požiadať inzertný portál o vymazanie osobných údajov, a preto by bolo vhodné túto funkcionalitu doimplementovať.

5.2.2 Zákaznícka podpora

V aktuálnej podobe webovej aplikácie sa nenachádza žiadna možnosť, akou by užívateľ mohol kontaktovať zákaznícku podporu. Riešením by bolo vytvorenie obrazovky s kontaktnými informáciami. Zákaznícku podporu by zo začiatku mohol poskytovať administrátor webovej aplikácie.

5.2.3 Zpracovanie užívateľských údajov

Webová aplikácia momentálne neopĺyva tzv. podmienkami používania, ktoré by bolo potrebné pred nasadením na produkciu zabezpečiť. V takýchto podmienkach by boli uvedené právne ustanovenia webovej aplikácie voči užívateľovi.

5.2.4 Návod aplikácie

Jedným z 10 princípov použiteľnosti podľa Jakoba Nielsena je aj Help and documentation [31].

V prípade, že si je užívateľ neistý ako previesť danú operáciu, mal by mať možnosť náhľadu do návodu, ktorý by mal byť na webovej stránke k dispozícii.

5.2.5 Správa užívateľského účtu

V momentálnej verzii môže užívateľ zmeniť svoje užívateľské heslo. Funkcionalita spravovania užívateľského účtu je zatiaľ minimalistická, avšak postačujúca. Pridanie nasledujúcich funkcií pre užívateľa by pridalo na flexibilitu správy užívateľského účtu.

- úprava e-mailovej adresy a mena
- vymazanie účtu

5.2.6 Autentizácia pomocou externého účtu

Webová aplikácia neumožňuje prihlásenie pomocou externého účtu. Takýto typ autentizácie naberá na popularite a šetrí čas užívateľov. Ide o jedno z možných vylepšení webovej aplikácie do budúcnosti.

5.2.7 Komunikácia a hodnotenie

O funkcionalite hodnotenia remeselníkov bol prejavovaný veľký záujem v dotazníku. Zabezpečenie objektívnych hodnotení nie je jednoduchý proces, a preto sa autor rozhodol upriamiť pozornosť na základnú funkcionalitu aplikácie. Funkcia hodnotenia remeselníkov sa v momentálnej verzii aplikácie nenachádza. Ide však o jednu z funkcií, o ktoré bol prejavovaný najväčší záujem, a preto by mala byť prioritou v nasledujúcom vývoji.

Medzi menej podstatné, avšak vítane rozšírenie by spadala funkcia kontaktovania remeselníka priamo na stránke pomocou vbudovaného komunikačného kanálu. Funkcia by zvýšila návštevnosť a zrýchlila komunikáciu medzi návštevníkom a remeselníkom.

5.2.8 Prístupnosť

Google Chrome je aktuálne jediným podporovaným internetovým prehliadačom. Obsah budúceho vývoja webovej aplikácie by mal zahŕňať podporu pre ďalšie veľké internetové prehliadače ako Safari, Mozilla Firefox atď.

Ďalším možným vylepšením prístupnosti by mohlo byť vytvorenie mobilnej aplikácie. V danom prípade ide o veľký krok a bolo by potrebné prevedenia tzv. ROI (Return On Investment) analýzy.

5.2.9 Topovanie inzerátu

Pokiaľ by bol záujem o komerčné využitie, bolo by potrebné, aby nebola webová aplikácia stratová. Možnosť predplatenia tzv. topovania inzerátu sa stala zvykom na inzertných portáloch, v momentálnej situácii takéto topovanie zaobstaráva administrátor manuálne. Rozšírenie funkcionality o automatizovaný proces topovania inzerátu, by bol nevyhnutý pri komerčnom využití.

Záver

Cieľom bakalárskej práce bolo zvolenie vhodnej formy zberu užívateľských požiadaviek, ich zber a následná analýza, navrhnutie a implementovanie aspoň prototypovej verzie webovej aplikácie, ktorá uľahčí propagáciu remeselníckej práce v online priestore. Pre získanie čo najlepších poznatkov boli analyzované už existujúce riešenia.

V časti návrhu sa nachádza návrh funkčných a nefunkčných požiadaviek na systém. Následne boli vymodelované všetky prípady užitia, na základe ktorých bolo pomocou tabuľky pokrytia overené pokrytie všetkých funkčných požiadaviek. Detailné spracovanie návrhu užívateľského rozhrania bolo podmienené získanými vedomosťami v predchádzajúcich častiach.

Po ukončení návrhu bol implementovaný funkčný prototyp webovej aplikácie, ktorý ešte nie je nasaditeľný na produkciu. V časti implementácia boli uvedené použité technológie, a k nim ich praktické využitie v aplikácii, ktoré bolo patrične argumentačne podložené.

Výsledná implementácia bola obohatená o automatické testy, a ďalej podrobená užívateľskému testovaniu. Chyby, ktoré boli počas testovania odhalené boli autorom práce náležite odstránené.

V poslednej kapitole bola snaha o čo najobjektívnejšie hodnotenie vzniknutej webovej aplikácie v súlade so zadaním. V tejto časti boli taktiež zmienené možné vylepšenia aplikácie do budúcnosti.

Vytvorená aplikácia implementuje všetky funkcie, ktoré boli na ňu kladené. Z toho vyplýva, že sa podarilo splniť všetky ciele zadania práce.

Ide zatiaľ iba o prototypovú implementáciu, ktorú je potrebné pred nasadením do produkcie ešte rozšíriť. Všetky autorove návrhy na možné vylepšenie webovej aplikácie boli predstavené v poslednej kapitole bakalárskej práce. Na základe zvolených postupov, dobrej dokumentácií a použitým princípom je v budúcnosti rozšírenie webovej aplikácie jednoduché.

Bibliografia

1. GOOGLE LCC. *Lighthouse* [online]. 2021-02 [cit. 2021-03-16]. Dostupné z : <https://developers.google.com/web/tools/lighthouse>.
2. INZERCIA.SK S.R.O. *Inzercia.sk - inzeráty a bazár zdarma* [online] [cit. 2021-03-16]. Dostupné z : <https://www.inzercia.sk/>.
3. SMIČKA, Radim. *Bazos.sk - Inzercia, inzeráty* [online] [cit. 2021-03-16]. Dostupné z : <https://www.bazos.sk/>.
4. UNITED CLASSIFIEDS S.R.O. *Bazar.sk* [online] [cit. 2021-03-16]. Dostupné z : <https://www.bazar.sk/>.
5. GOOGLE LCC. *Overview* [online]. 2021-03 [cit. 2021-03-19]. Dostupné z : <https://developers.google.com/maps/documentation/places/web-service/overview>.
6. STATCOUNTER. *Browser Market Share Worldwide* [online]. 2021-03 [cit. 2021-03-25]. Dostupné z : <https://developers.google.com/maps/documentation/places/web-service/overview>.
7. STATCOUNTER. *Browser Market Share Slovakia (slovak Republic)* [online]. 2021-03 [cit. 2021-03-25]. Dostupné z : [https://gs.statcounter.com/browser-market-share/all/slovakia-\(slovak-republic\)](https://gs.statcounter.com/browser-market-share/all/slovakia-(slovak-republic)).
8. STATCOUNTER. *Desktop Screen Resolution Stats Worldwide* [online]. 2021-03 [cit. 2021-03-25]. Dostupné z : <https://gs.statcounter.com/screen-resolution-stats/desktop/worldwide>.
9. WHITENTON, Kathryn. *Website Logo Placement for Maximum Brand Recall* [online]. 2016-02 [cit. 2021-03-27]. Dostupné z : <https://www.nngroup.com/articles/logo-placement-brand-recall>.
10. SHARPENED PRODUCTIONS, INC. *Framework* [online]. 2013-03 [cit. 2021-04-08]. Dostupné z : <https://techterms.com/definition/framework>.

11. TUTORIALS POINT (I) PVT. LTD. *Spring - MVC Framework* [online] [cit. 2021-04-05]. Dostupné z : <https://gs.statcounter.com/screen-resolution-stats/desktop/worldwide>.
12. TIOBE SOFTWARE BV. *TIOBE Index for April 2021* [online]. 2021-04 [cit. 2021-04-06]. Dostupné z : <https://www.tiobe.com/tiobe-index/>.
13. TUTORIALS POINT (I) PVT. LTD. *Why to Learn Spring?* [Online] [cit. 2021-04-07]. Dostupné z : <https://www.tutorialspoint.com/spring/index.htm>.
14. JAISWAL, Sonoo. *What is Spring Boot* [online] [cit. 2021-04-07]. Dostupné z : <https://www.javatpoint.com/spring-boot-tutorial>.
15. TUTORIALS POINT (I) PVT. LTD. *Spring Boot - Introduction* [online] [cit. 2021-04-07]. Dostupné z : https://www.tutorialspoint.com/spring_boot/spring_boot_introduction.htm.
16. TYSON, Matthew. *What is Tomcat? The original Java servlet container* [online]. 2019-12 [cit. 2021-04-10]. Dostupné z : <https://www.infoworld.com/article/3510460/what-is-apache-tomcat-the-original-java-servlet-container.html>.
17. PIVOTAL SOFTWARE, INC. *Spring Security - Overview* [online] [cit. 2021-04-07]. Dostupné z : <https://spring.io/projects/spring-security>.
18. PIVOTAL SOFTWARE, INC. *Spring Data JPA - Overview* [online] [cit. 2021-04-07]. Dostupné z : <https://spring.io/projects/spring-data-jpa>.
19. GRADLE, INC. *What is Gradle?* [Online] [cit. 2021-04-09]. Dostupné z : https://docs.gradle.org/current/userguide/what_is_gradle.html.
20. GOOGLE LLC. *Google Java Style Guide* [online] [cit. 2021-04-08]. Dostupné z : <https://google.github.io/styleguide/javaguide.html>.
21. ORACLE CORPORATION. *javadoc - The Java API Documentation Generator* [online] [cit. 2021-04-09]. Dostupné z : <https://docs.oracle.com/javase/7/docs/technotes/tools/windows/javadoc.html>.
22. MDN WEB DOCS. *Web Components* [online]. 2021-03 [cit. 2021-04-11]. Dostupné z : https://developer.mozilla.org/en-US/docs/Web/Web_Components.
23. KOIVUVIITA, Jouni. *Application Architecture* [online]. Mar-2021 [cit. 2021-04-12]. Dostupné z : <https://vaadin.com/docs/v14/guide/introduction>.
24. MDN WEB DOCS. *CSS: Cascading Style Sheets* [online]. 2021-04 [cit. 2021-04-12]. Dostupné z : <https://developer.mozilla.org/en-US/docs/Web/CSS>.

-
25. MDN WEB DOCS. *What is JavaScript?* [Online]. 2021-03 [cit. 2021-04-13]. Dostupné z : <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>.
 26. TUTORIALS POINT (I) PVT. LTD. *Java - Documentation Comments* [online] [cit. 2021-04-09]. Dostupné z : https://www.tutorialspoint.com/java/java_documentation.htm.
 27. JONÁŠ, Martin. *Návrhové principy: SOLID* [online]. 2012-05 [cit. 2021-04-14]. Dostupné z : <https://zdrojak.cz/clanky/navrhove-principy-solid/>.
 28. KUNČAR, Petr. *Mockito - unit test framework* [online] [cit. 2021-04-15]. Dostupné z : <https://www.itnetwork.sk/java/pokrocilych/mockito-unit-test-framework>.
 29. NIELSEN, Jakob. *Why You Only Need to Test with 5 Users* [online]. 2000-03 [cit. 2021-04-16]. Dostupné z : <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
 30. U.S. DEPARTMENT OF HEALTH AND HUMAN SCIENCES. *Usability Evaluation Basics* [online] [cit. 2021-04-18]. Dostupné z : <https://www.usability.gov/what-and-why/usability-evaluation.html#:~:text=Usability%20refers%20to%20the%20quality,overall%20satisfaction%20of%20the%20user..>
 31. NIELSEN, Jakob. *10 Usability Heuristics for User Interface Design* [online]. 2020-11 [cit. 2021-04-19]. Dostupné z : <https://www.nngroup.com/articles/ten-usability-heuristics/>.

Zoznam použitých skratiek

- XS** Extra small
- SM** Small
- MD** Medium
- LG** Large
- XL** Extra large
- SEO** Search Engine Optimization
- PSČ** Poštové Smerovacie Číslo
- API** Application Programming Interface
- GPS** Global Positioning System
- DRY** Don't Repeat Yourself
- KISS** Keep It Simple Stupid
- EA** Enterprise Architect
- CRUD** Create, Read, Update, Delete
- JPQL** Jakarta Persistence Query Language
- SQL** Structured Query Language
- JPA** Java Persistence API
- URL** Uniform Resource Locator
- MVC** Model-view controller
- JVM** Java Virtual Machine

A. ZOZNAM POUŽITÝCH SKRATIEK

Java SE 11 Java Standard Edition 11

MB Mega Byte

Java EE Java Enterprise Edition

J2EE Java 2 Platform, Enterprise Edition

POJO Plain Old Java Object

RAD Rapid Application Development

XML Extensible Markup Language

JSP JavaServer Pages

SMTP Simple Mail Transfer Protocol

JDK Java Development Kit

IDE Integrated Development Environment

JAR Java Archive

HTML Hyper-Text Markup Language

UI User Interface

LTS Long Term Support

CSS Cascading Style Sheets

JS JavaScript

YAGNI You Aren't Going To Need It

ROI Return On Investment

Obsah priloženej SD karty

	readme.txt	stručný popis obsahu SD karty
	src	
	thesis	zdrojová forma práce vo formáte L ^A T _E X
	questionnaire.xlsx	odpovede z dotazníka vo formáte XLSX
	text	text práce
	thesis.pdf	text práce vo formáte PDF