

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Webová aplikace pro ubytovací zařízení
Bakalářská práce

Autor: Dominik Kohl
Studijní obor: Aplikovaná informatika

Vedoucí práce: Mgr. Daniela Ponce, Ph.D.

Prohlášení:

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

V Hradci Králové dne 13.08.2022


Dominik Kohl

Poděkování:

Děkuji vedoucímu bakalářské práce Mgr. Daniele Ponce, Ph.D. za metodické vedení práce a velmi cenné rady. Chtěl bych také poděkovat své rodině za podporu během mého studia.

Anotace

Bakalářská práce je orientovaná na vývoj webové aplikace pro ubytovacího zařízení penzion U Dolu. Ubytovací zařízení oslovilo autora práce z důvodu zastaralé webové aplikace. Zařízení poskytlo pokyny a požadavky na dohodnutých schůzkách. Hlavním zadavatelským cílem bylo vytvořit webovou aplikaci, která umožní stávajícím zákazníkům snazší prožitek z ubytování, bez nutnosti kontaktovat provozovatele ubytovacího zařízení.

V počátku práce bylo vytvořeno porovnání konkurenčních aplikací, následované analýza aplikace, kde byly sbírány požadavky. Podle těchto požadavků vznikl návrh celkové aplikace, poté byla webová aplikace implementována za použití jazyka Javascript a pomocí MERN stack. V závěru bylo opět provedeno vyhodnocení nově vytvořené aplikace, která slouží jako potvrzení o vylepšení webové aplikace.

Klíčová slova: webová aplikace, rezervace, MERN stack, webová stránka, SEO, responzivní optimalizace, ubytování

Annotation

Title: Web application for accomodation

The bachelor thesis is focused on the development of a web application for the accommodation facility Pension U Dolu. The accommodation facility approached the author of the thesis because of an outdated web application. The establishment provided instructions and requirements at the agreed meetings. The main objective of the client was to develop a web application that would allow existing customers to have an easier accommodation experience, without having to contact the accommodation provider.

At the beginning of the work, a comparison of competing applications was created, followed by an analysis of the application where requirements were collected. According to these requirements the overall application design was created, then the web application was implemented using Javascript language and using MERN stack. At the end again evaluation of the newly developed web application was done to serve as a confirmation of the web application improvement.

Key words: web app, booking, MERN stack, website, SEO, responsive optimization, accommodation.

Obsah

1	Úvod.....	1
2	Cíl práce, metodika	2
3	Porovnání konkurenčních aplikací.....	3
3.1	Zpracování porovnání	3
3.1.1	SEO	3
3.1.2	Responzivní optimalizace.....	3
3.1.3	Funkčnost a obsah.....	4
3.2	Způsob hodnocení webových aplikací	4
3.3	Postup hodnocení.....	5
3.3.1	Nalezení webové aplikace pro hodnocení	5
3.3.2	Hodnocení aplikací.....	6
3.3.2.1	Hodnocení aplikace ubytovacího zařízení U Dolu.....	6
3.3.2.2	Hodnocení aplikace ubytovacího zařízení Hotýlek pod smrkem... 7	
3.3.2.3	Hodnocení aplikace ubytovacího zařízení Chata pod Lipami.....	8
3.4	Vyhodnocení aplikací.....	9
3.4.1	Webová aplikace U Dolu	9
3.4.2	Webová aplikace Hotýlek pod smrkem	10
3.4.3	Webová aplikace Chata pod Lipami	11
3.5	Porovnání	12
4	Analýza aplikace	13
4.1	Požadavky aplikace	13
4.2	Případy užití	14
4.2.1	Nepřihlášený uživatel	14
4.2.2	Přihlášený uživatel.....	15
4.2.3	Provozovatel.....	15

5	Návrh aplikace	17
5.1	Přihlašovací systém	17
5.2	Rezervační systém	17
5.3	Mechanismus hlídacího psa	19
5.4	Konfigurace aplikace	19
5.5	Rozložení stránek	22
6	Technologie a implementace aplikace	24
6.1	Technologie	24
6.1.1	Javascript	24
6.1.2	HTML a CSS	24
6.1.3	MERN Stack	24
6.1.3.1	React.js	25
6.1.3.2	Expres.js	25
6.1.3.3	Node.js	26
6.1.3.4	MongoDB	27
6.2	Implementace	27
6.2.1	Struktura	27
6.2.2	Aplikační vrstva	29
6.2.2.1	Nasazení aplikace	29
6.2.2.2	Přihlášení	29
6.2.2.3	Registrace	30
6.2.2.4	Obnovení hesla	31
6.2.2.5	Rezervace	32
6.2.2.6	Profil	35
6.2.2.7	Úprava rezervace	36
6.2.2.8	Administrátorský panel	40

7	Zhodnocení aplikace	42
7.1	Hodnocení nové webové aplikace.....	42
7.2	Vyhodnocení aplikace.....	43
8	Shrnutí výsledků.....	44
9	Závěry a doporučení	46
10	Seznam použité literatury.....	48
11	Přílohy.....	49

Seznam obrázků

Obrázek 1 – Úvodní stránka zařízení U Dolu. Generováno: [5]	9
Obrázek 2 – Hodnocení zařízení U Dolu. Zdroj: autor	9
Obrázek 3 – Úvodní stránka zařízení Hotýlku pod smrkem. Generováno: [5]	10
Obrázek 4 – Hodnocení zařízení Hotýlku pod smrkem. Zdroj: autor	10
Obrázek 5 – Úvodní stránka zařízení Chata pod Lipami. Generováno: [5]	11
Obrázek 6 – Hodnocení zařízení Chata pod Lipami. Zdroj: autor	11
Obrázek 7 – Diagram případů užití. Zdroj: autor	16
Obrázek 8 – Návrh Databáze Penzionu U Dolu. Zdroj: autor	21
Obrázek 9 – Mapa rozložení stránek. Zdroj: autor	22
Obrázek 10 – Vrstvy MERN stack. Zdroj: [8]	25
Obrázek 11 – Node.js Server. Zdroj: [13]	26
Obrázek 12 – Grafické znázornění maximální obsazenosti. Zdroj: autor	33
Obrázek 13 – Úvodní stránka nové webové aplikace. Generováno: [5]	43
Obrázek 14 – Hodnocení nové webové aplikace. Zdroj: autor	43

Seznam tabulek

Tabulka 1 – Výsledky testu ubytovacího zařízení U Dolu	6
Tabulka 2 – Výsledky testu ubytovacího zařízení Hotýlek pod smrkem	7
Tabulka 3 – Výsledky testu ubytovacího zařízení Chata pod Lipami	8
Tabulka 4 – Výsledky testu nové webové aplikace	42
Tabulka 5 – Slovní hodnocení testu. Zdroj: [5]	6

1 Úvod

Bakalářská práce se zabývá vytvořením nové webové aplikace pro ubytovací zařízení penzion U Dolu. Práce obsahuje všechny aplikované kroky při vytváření webové aplikace pro ubytovací zařízení, podle požadavků zadavatele. Ubytovací zařízení kontaktovalo autora z důvodu zastaralé aplikace. Zadavatel požaduje vylepšení stávající webové aplikace a udržení stávajících zákazníků. Hlavním nárok na aplikaci, je dovolit zákazníkům vytvářet a spravovat rezervace, bez nutnosti kontaktovat provozovatele ubytovacího zařízení.

Při vytváření aplikace bude nejprve nastíněno, co by aplikace měla splňovat a dodržovat, následně budou sbírány zadavatelovy požadavky na aplikace v předem dohodnutých schůzkách. Bude vytvořen návrh aplikace, který poté bude implementován a bude prokázáno, jestli je aplikace dostačující.

2 Cíl práce, metodika

Cílem bakalářské práce je návrh a implementace webové aplikace pro ubytovací zařízení dle požadavků zadavatele. Webová aplikace má za úkol udržení stávajících zákazníků. Hlavním úkolem je umožnit lepší interakci s webovou aplikací, bez nutnosti kontaktovat provozovatele zařízení.

V první fázi bylo vytvořeno porovnání konkurenčních webových aplikací se zadavatelovou aplikací. Tato fáze má za úkol nastínit, co by taková webová aplikace ne/měla obsahovat a splňovat.

V následující části zabývající se analýzou webové aplikace, bylo na toto porovnání bráno zřetel a za pomoci zadavatele byly vytvořeny požadavky a typové úlohy kladené na webovou aplikaci.

Následující návrhová fáze obsahuje samotný návrh aplikace vyplývající z dvou předchozích fází. Zabývá se především návrhem entit a rozložením a popsáním jednotlivých stránek webové aplikace.

V implementační fázi, která následuje po návrhové, jsou vysvětleny použité technologie a implementace zadavatelových požadavků.

V poslední fázi je zhodnocení nově vytvořené webové aplikace. Vyhodnocení je provedeno jako v kapitole zabývající se porovnáním konkurenčních webových aplikací a slouží k potvrzení, že webová aplikace je dostatečná a že je vylepšením od stávající webové aplikace ubytovacího zařízení.

3 Porovnání konkurenčních aplikací

V této kapitole je provedeno porovnání zadavatelovou stávající webovou aplikací s konkurenčními aplikacemi.

Obsahuje podkapitulu o zpracování porovnání, kde se nacházejí různá hlediska pro hodnocení aplikace. Dále obsahuje způsob hodnocení, kde jsou znovupoužitelné postupy hodnotící webovou aplikaci pro ubytovací zařízení. Nachází se zde také podkapitola postup hodnocení, která je zaměřena na samotné nalezení konkurenčních ubytovacích zařízení a jejich webových aplikací, obsahuje zároveň samotné hodnocení jednotlivých aplikací. Obsahuje také grafické vyhodnocení jednotlivých aplikací. V závěru je provedeno slovní porovnání aplikací s doporučeními.

3.1 Zpracování porovnání

V následující sekci jsou ve stručnosti popsány různá hlediska porovnání a hodnocení webových aplikací pro ubytovací zařízení.

3.1.1 SEO

Optimalizace pro vyhledávače (SEO – Search Engine Optimization) je proces vylepšování webových aplikací tak, aby se zvýšila kvalita a kvantita organické návštěvnosti z vyhledávačů. Úspěšná SEO znamená, že se webová aplikace bude s větší pravděpodobností zobrazovat výše na stránce s výsledky vyhledávání. Nejoblíbenějším vyhledávačem je Google, ale i další vyhledávače (Bing, Yahoo, DuckDuckGo atd.) mají své vlastní jedinečné algoritmy pro procházení webových stránek a vracení nejlepších výsledků vyhledávání. [1] (překlad)

Čím je lepší SEO, tím se webová aplikace zobrazuje výše po vyhledání klíčových slov související s webovou aplikací.

3.1.2 Responzivní optimalizace

Responzivní optimalizace nám říká, jak se zobrazuje webová aplikace na jiných zařízeních než počítači např. tablet, telefon. Responzivní optimalizace je velmi důležitá v dnešní době, protože hodně uživatelů používá mobilní zařízení a je velmi

vhodné, aby se webová aplikace zobrazovala dobře jak na stolním počítači, tak také na ostatních zařízeních s jiným rozlišením obrazovky.

3.1.3 Funkčnost a obsah

Funkčnost webových stránek je velmi důležitá, uživatel chce používat webovou aplikaci za nějakým účelem a očekává od ní nějakou funkcionalitu. Aplikace zároveň musí mít obsah, protože bez obsahu webových stránek nevíme, k čemu webová stránka slouží nebo čím se zabývá a nedokážeme bez cizí pomoci stránku používat.

3.2 Způsob hodnocení webových aplikací

Webové aplikace pro ubytovací zařízení je hodnocena z několika hledisek na stupnici od 0 do 100, při čemž 0 je nejhorší a 100 je nejlepší možný počet bodů.

Prvním hlediskem je SEO (Search engine optimization), které nám říká, jestli je webová aplikace optimalizovaná pro vyhledávač. Pro konzistentní vyhodnocení tohoto hlediska je používán vývojový nástroj Lighthouse v prostředí webového prohlížeče Chrome. Nástroj vyčíslí na stupnici od 0 do 100, jak je daná webová aplikace optimalizovaná pro vyhledávač. Nástroj také poskytuje různé návrhy a doporučení pro zvýšení/vylepšení SEO.

Druhým hlediskem je vyhodnocení responzivní optimalizace webové aplikace. Pro konzistentní vyhodnocení tohoto hlediska je použita stránka Silktide: Nibbler [2], která umožňuje dostat hned několik kategorií hodnocení. V tomto celkovém hodnocení se bere v potaz jen kategorie Mobile. Stránka hodnotí webovou aplikaci od 0 až do 10, toto hodnocení je propočítáváno na jednotnou stupnici hodnocení 0 až 100.

Posledním hlediskem je funkčnost a obsah. Pro toto hledisko se používá vlastní test, který pomůže určit, jak si webová aplikace vede po stránce obsahové a funkcionální. Test je vytvořen na základě webových stránek How good is your website? [3]. Ze stránek jsou vhodně a pečlivě vyselektovány a vytvořeny otázky vhodné k ubytovacím zařízením. Hodnocení je upraveno, aby lépe odpovídalo jednotné stupnici, a je také přeložené slovní hodnocení výsledků testu.

Test v 15 otázkách posuzuje aplikaci především z pohledu obsahového a funkčního, je zaměřen pro webové aplikace pro ubytovací zařízení. Hodnocení jednotlivých

otázek je od 0 bodů (nejhorší) po 2 body (nejlepší). Možné hodnocení je tedy 0 až 30 bodů. Přesné a kompletní znění testu je uvedeno v příloze č. 3, společně se slovním hodnocením pro výsledky testu.

3.3 Postup hodnocení

3.3.1 Nalezení webové aplikace pro hodnocení

K nalezení vhodné webové aplikace k porovnání se zadanou webovou aplikací se pro nalezení konkurenčních ubytovacích zařízení a jejich stránek používá několik parametrů pro vyhledávání.

První z těchto parametrů je kapacita zařízení, která je okolo 16 míst. Dalšími parametry jsou: zařízení má v pokojích vlastní koupelnu, zařízení nabízí wellness, parkoviště, snídaně anebo večeře, jestli zařízení poskytuje nějaký druh zabavení hostů. Zařízení k porovnání by se mělo podobat vzhledem a rozsahem zadaného objektu, se kterým se bude porovnávat. Lokalizace ubytovacího zařízení by měla být v oblasti hor.

K nalezení vhodných zařízení jsou použity webové stránky Penzion.cz [4], na nich je spuštěné vyhledávání s parametry: „16 hostů, 6 pokojů, strava lze dokoupit, snídaně v ceně, parkování, Wi-fi a wellness“ s lokalizací v Krkonoších.

Po pečlivém prohlédnutí zobrazených výsledků jsou k porovnání se zadaným ubytovacím zařízením (penzion U Dolu - <http://www.udolu.cz/>) zvoleny Hotýlek pod smrkem(<https://www.hotylekpodsmrkem.cz/>) a Chata pod Lipami (<https://www.chatapodlipami.cz/>).

3.3.2 Hodnocení aplikací

V následující třech podkapitolách jsou vyhodnoceny webové aplikace podle dříve definovaných hledisek.

3.3.2.1 Hodnocení aplikace ubytovacího zařízení U Dolu

Z prvního hlediska SEO získaly webové stránky udolu.cz 69 bodů ze 100. Pro zlepšení SEO je doporučení pro tyto stránky zlepšit meta data, doplnit atributy alt.

Z druhého hlediska responzivní optimalizace získaly webové stránky udolu.cz 3,0 bodů z 10 (30/100).

Výsledek hodnocení z posledního hlediska funkčnosti a obsahu je 13,33 bodů ze 100, konkrétní otázky jsou shrnuté v tabulce níže. Hodnocení v 15 jednotlivých otázkách testu je součástí autorovi práce a je uvedeno v příloze.

Otázka	Počet bodů	Otázka	Počet bodů
1.	0	9.	0
2.	0	10.	0
3.	1	11.	0
4.	0	12.	0
5.	0	13.	1
6.	0	14.	1
7.	0	15.	0
8.	1		
Celkem bodů: 4 / 30 (13,33/100)			

Tabulka 1 – Výsledky testu ubytovacího zařízení U Dolu

3.3.2.2 Hodnocení aplikace ubytovacího zařízení Hotýlek pod smrkem

Z prvního hlediska SEO získaly webové stránky hotylekpodsmrkem.cz 93 bodů ze 100. Pro zlepšení SEO je doporučení pro tyto stránky doplnit meta data.

Z druhého hlediska responzivní optimalizace získaly webové stránky hotylekpodsmrkem.cz 10 bodů z 10 (100/100).

Výsledek hodnocení z posledního hlediska funkčnosti a obsahu je 66,66 bodů ze 100, konkrétní otázky jsou shrnuté v tabulce níže. Hodnocení v 15 jednotlivých otázkách testu je součástí autorovi práce a je uvedeno v příloze.

Otázka	Počet bodů	Otázka	Počet bodů
1.	1	9.	0
2.	2	10.	1
3.	1	11.	1
4.	2	12.	2
5.	2	13.	1
6.	1	14.	2
7.	0	15.	2
8.	2		
Celkem bodů: 20 / 30 (66,66/100)			

Tabulka 2 – Výsledky testu ubytovacího zařízení Hotýlek pod smrkem

3.3.2.3 Hodnocení aplikace ubytovacího zařízení Chata pod Lipami

Z prvního hlediska SEO získaly webové stránky chatapodlipami.cz 92 bodů ze 100.

Pro zlepšení SEO je doporučení pro tyto stránky doplnit meta data.

Z druhého hlediska responzivní optimalizace získaly webové stránky chatapodlipami.cz 10 bodů z 10 (100/100).

Výsledek hodnocení z posledního hlediska funkčnosti a obsahu je 53,33 bodů ze 100, konkrétní otázky jsou shrnuté v tabulce níže. Hodnocení v 15 jednotlivých otázkách testu je součástí autorovi práce a je uvedeno v příloze.

Otázka	Počet bodů	Otázka	Počet bodů
1.	1	9.	0
2.	1	10.	1
3.	1	11.	0
4.	2	12.	2
5.	1	13.	1
6.	1	14.	2
7.	0	15.	1
8.	2		
Celkem bodů: 16 / 30 (53,33/100)			

Tabulka 3 - Výsledky testu ubytovacího zařízení Chata pod Lipami

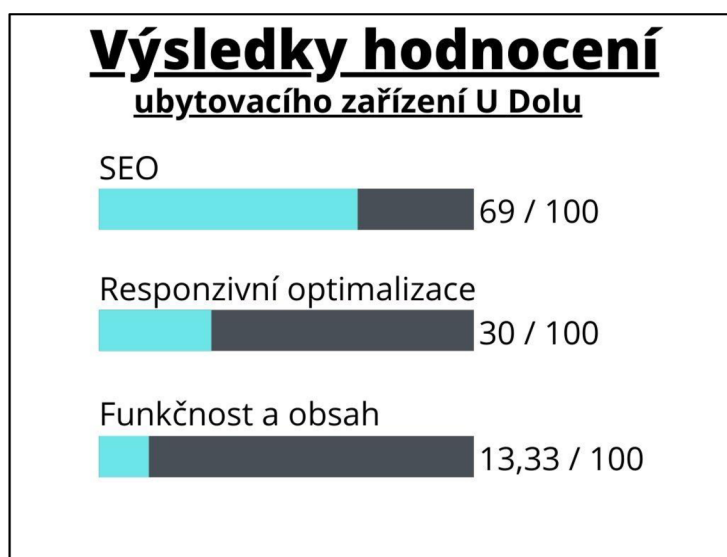
3.4 Vyhodnocení aplikací

V následující kapitole je zobrazená úvodní stránka a grafické znázornění hodnocení každé webové aplikace, které jsou určeny k hodnocení.

3.4.1 Webová aplikace U Dolu

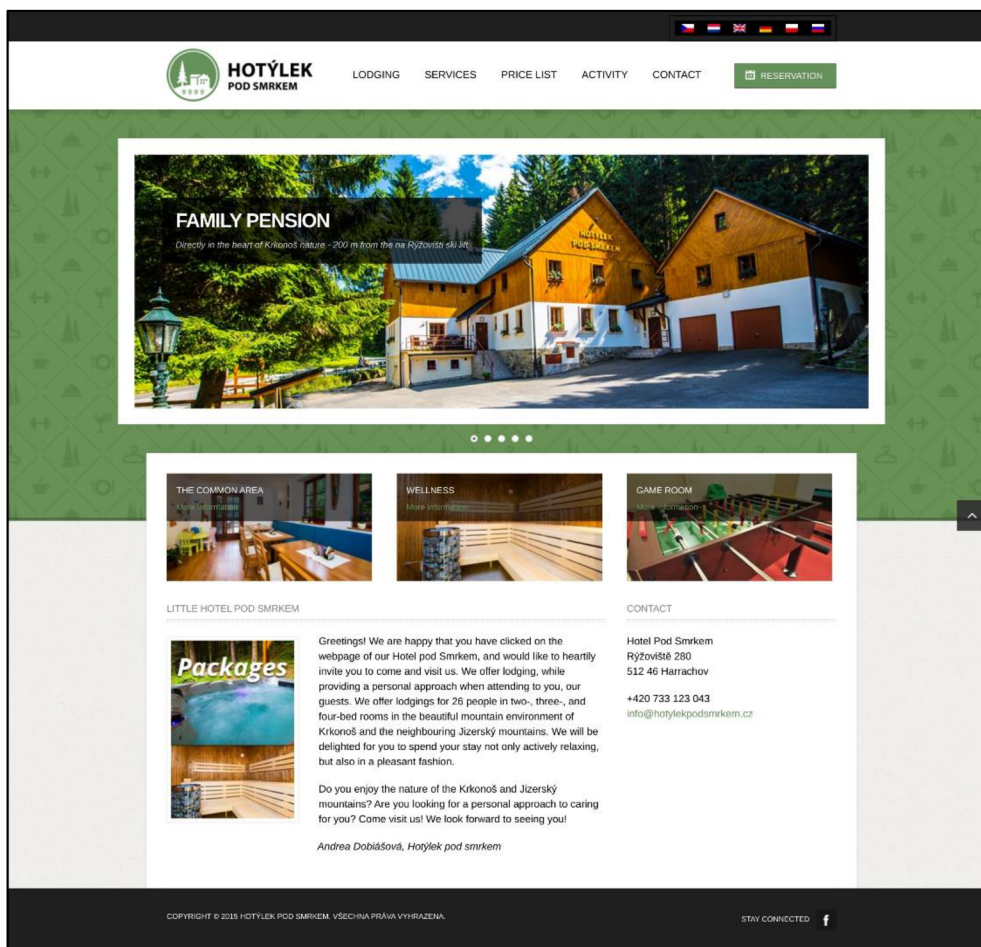


Obrázek 1 – Úvodní stránka zařízení U Dolu. Generováno: [5]



Obrázek 2 – Hodnocení zařízení U Dolu. Zdroj: autor

3.4.2 Webová aplikace Hotýlek pod smrkem

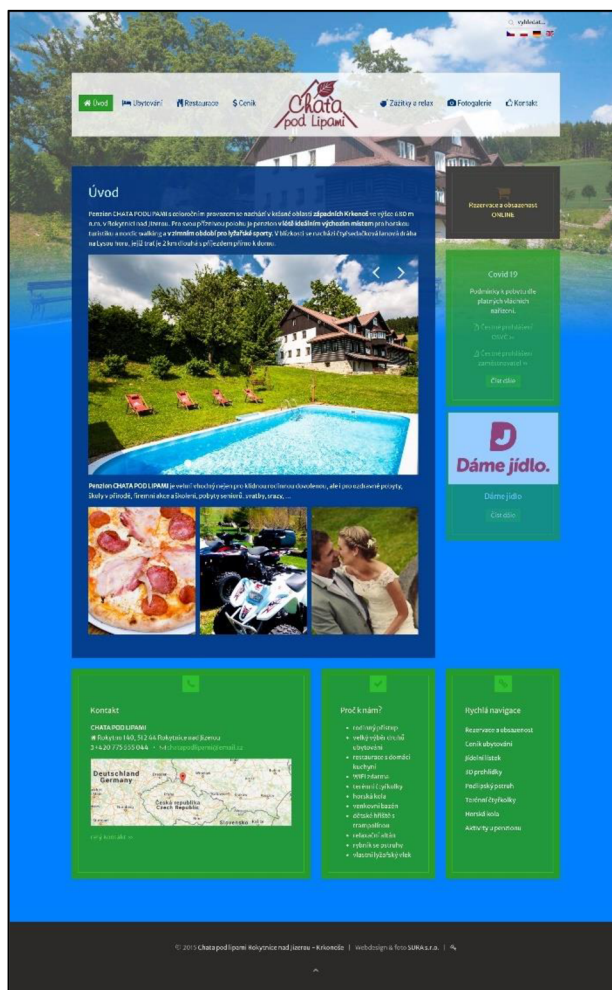


Obrázek 3 – Úvodní stránka zařízení Hotýlku pod smrkem. Generováno: [5]

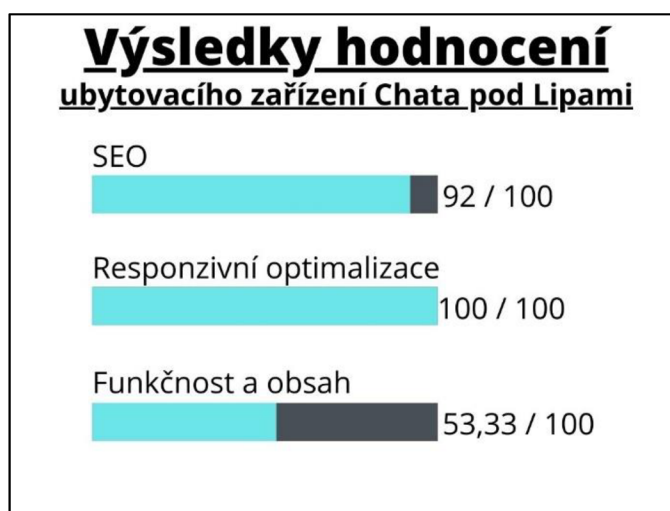


Obrázek 4 – Hodnocení zařízení Hotýlku pod smrkem. Zdroj: autor

3.4.3 Webová aplikace Chata pod Lipami



Obrázek 5 – Úvodní stránka zařízení Chata pod Lipami. Generováno: [5]



Obrázek 6 – Hodnocení zařízení Chata pod Lipami. Zdroj: autor

3.5 Porovnání

Obě nalezené webové aplikace, se kterými je zadaná aplikace porovnávána se od sebe příliš neliší, z jednotlivých hledisek si vedly velmi podobně. Penzion U Dolu si vedl ze všech hledisek hůře oproti ostatním aplikacím. Nejvíce lze rozdíl pozorovat u hlediska funkčnost a obsah.

Při tvorbě webové aplikace by se měl brát především zřetel na funkčnost a obsah webových stránek a na jejich správné rozložení. V úvahu by se měl brát zřetel na snadné používání webové aplikace, bez nutnosti komunikovat s provozovatelem. Každá webová aplikace by měla být optimalizovaná pro další typy zařízení, hlavně pro mobilní telefony, protože je tato metoda v dnešní době hodně využívána.

Žádná z webových aplikací neobsahuje jiné médium než fotky. Videá dokážou možné zákazníky přilákat a je větší šance, že si vyberou právě dané ubytovací zařízení.

4 Analýza aplikace

V následující kapitole bakalářské práce jsou popsány požadavky zadavatele kladené na webovou aplikaci, společně s případy užití aplikace.

4.1 Požadavky aplikace

V následující sekci jsou představeny požadavky zadavatele na webovou aplikaci. Požadavky byly sbírány autorem práce v několika předem dohodnutých schůzkách se zadavatelem. Zadavatelovým hlavním cílem bylo především vyvinout novou aplikaci, která by usnadnila ubytování pravidelným zákazníkům ubytovacího zařízení. Na každé schůzce byly některé požadavky schváleny nebo upraveny. V následující kapitole se podíváme na finální podobu těchto požadavků.

Aplikace umožní vytvořit rezervaci. Uživatel uvidí vlastní vytvořené rezervace a bude je moci spravovat a zjistit aktuální stav rezervace. Stavy rezervace budou tři a to: čekající, schválena a zamítnuta. Rezervaci si bude moci každý uživatel upravit a smazat, ale pouze před příjezdem, do předem nadefinovaném času, určeným provozovatelem. Rezervace jde vytvořit jak pro jeden pokoj, tak i pro více pokojů. Rezervace se nesmějí překrývat a zasahovat jedna do druhé. Příjezd a odjezd jedné rezervace může být ve stejný den. Rezervace může vytvářet pouze přihlášený uživatel.

Aplikace bude mít implementován základní přihlašovací systém, který umožní registraci, přihlášení, obnovení hesla, potvrzení mailové adresy a dovolí změnit heslo přihlášenému uživateli. Uživatel bude mít přístupný svůj panel s profilem, kde uvidí své rezervace a bude je moci spravovat. Aplikace bude rozlišovat tři typy uživatelů a to provozovatel, přihlášený uživatel a nepřihlášený uživatel.

Provozovatel bude jen jeden a bude mít nejvyšší práva. Provozovatel bude mít přístupný administrátorský panel přes, který bude moci spravovat celou aplikaci. Přes toto rozhraní bude provozovatel moci: spravovat rezervace – upravovat jednotlivé rezervace, případně i smazat, nastavit základní konfiguraci aplikace – vypnout/zapnout vytváření rezervaci, vypnout/zapnout vyhledávání volných pokojů v určitém termínu, vypnout/zapnout možnost registraci nových uživatelů, nastavit minimální počet nocí před vytvořením rezervace, definovat maximální a

minimální počet strávených nocí na jednu rezervaci, nastavit kolik dní před příjezdem může zákazník dělat změny v rezervaci, nastavit fixní kurz Eura na Korunu českou pro znázornění ceny rezervace, nastavit cenu v eurech pro fixně nadefinované sezóny po celý rok, vytvářet blokace pro celé ubytovací zařízení, vytvořit blokaci pro jednotlivé pokoje. Provozovatel bude také moci vidět registrované uživatele.

Přihlášený uživatel bude moci vytvářet rezervace a spravovat své vlastní rezervace. Bude také umožněno vytvořit tzv. hlídacího psa, který zašle email uživateli, jestli že se uvolní kapacita pro dříve nenalezený termín. Tyto hlídací psy bude moci mít každý uživatel maximálně tři a bude je moci odebrat a přidávat dle libosti.

Nepřihlášený uživatel bude moci prohlížet obsah webové aplikace jako každá výše uvedená role, dokáže také najít volné pokoje v zadaném termínu s požadovanou kapacitou. Nebude mu umožněno vytvořit rezervaci a ani hlídacího psa.

Aplikace bude podporovat jazykovou lokalizaci, a to v podobě češtiny a němčiny. Aplikace bude optimalizovaná pro mobilní zařízení. Aplikace bude zobrazovat hlavní stránku s úvodním videem, galerii, vyhledání volných pokojů, sekci o nás a také sekci o okolí. Hlavní obsah stránky poskytne zadavatel práce. Aplikace by měla být bezpečná k používání, bude implementována komunikace pomocí SSL certifikátu a šifrování uložených hesel v databázi. Aplikace bude řešit autorizaci jednotlivých uživatelů pro správné fungování jejich rolí a pravomocí.

4.2 Případy užití

V následující sekci jsou popsány případy užití v závislosti na rolích jednotlivých uživatelů ve webové aplikaci. Rozlišují se role provozovatel, nepřihlášený uživatel a také přihlášený uživatel. V poslední části je zobrazen konkrétní diagram případů užití pro webovou aplikaci.

4.2.1 Nepřihlášený uživatel

Jedná se o návštěvníka webových stránek, který neprovedl autorizaci. Tento uživatel bude moci prohlížet prezentační obsah webové aplikace. Bude mu také umožněno si vyhledat volné pokoje ve zvoleném termínu s požadovanou kapacitou. Bude se moci registrovat, pokud ještě nemá vytvořený účet, v tom případě je mu

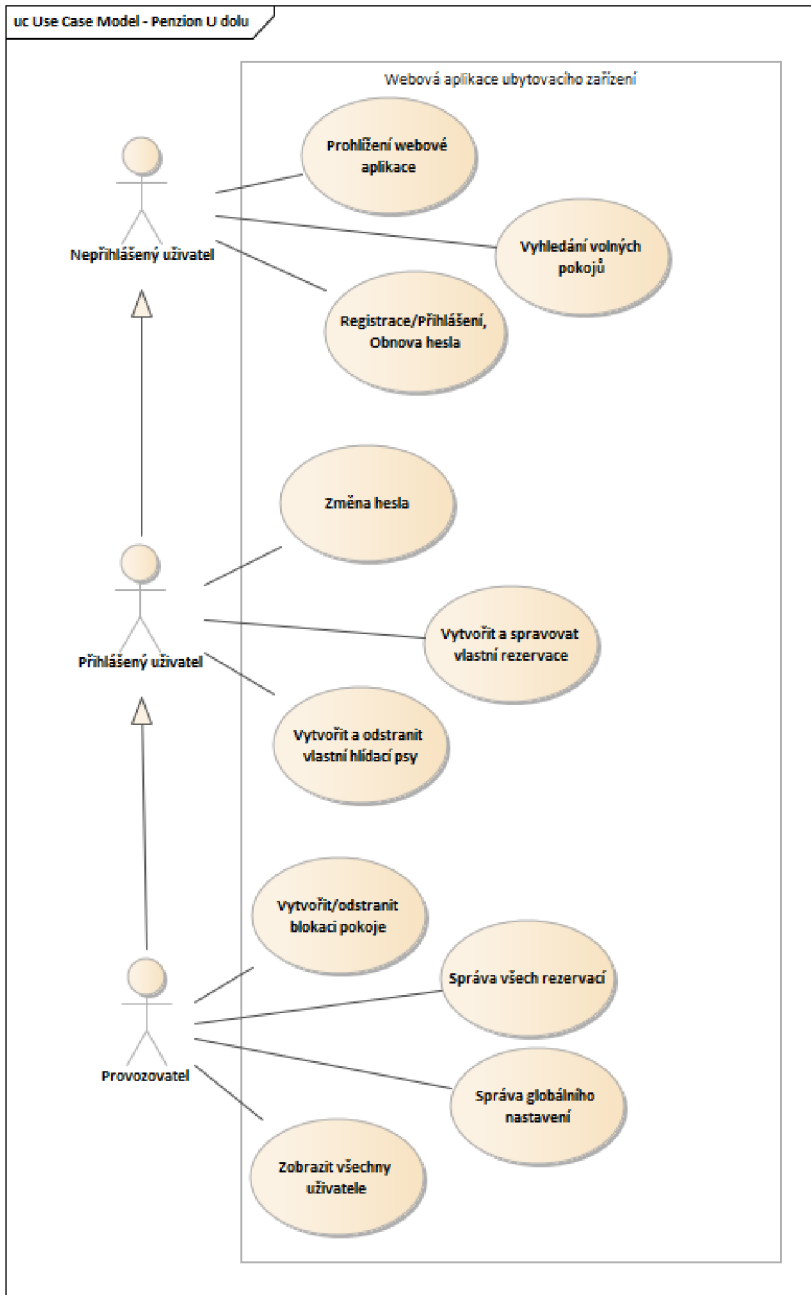
umožněno se přihlásit a tím se autorizovat webové aplikaci. Pokud uživatel má účet, ale zapomněl heslo je možnost obnovení hesla pomocí linku odeslaného na emailovou adresu.

4.2.2 Přihlášený uživatel

Jedná se o uživatele, který již provedl registraci a byl úspěšně autorizován webovou aplikací jako přihlášený uživatel. Bude mít stejné možnosti využívání webové aplikace jako nepřihlášený uživatel. Jeho možnosti budou rozšířené o vytváření a spravování vlastních rezervací. Dále si dokáže změnit vlastní heslo a bude mu také umožněno vytvořit a odstranit vlastní hlídací psy. Hlídací pes je ve své podstatě připomínka uživateli, že je možné vytvořit rezervaci ve zvoleném termínu se zvolenou kapacitou. Uživatel bude upozorněn zasláním emailové zprávy s možností provedení rezervace.

4.2.3 Provozovatel

Provozovatel má stejné možnosti a funkce využívání webové aplikace jako přihlášený uživatel. Jelikož se jedná o provozovatele daného ubytovacího zařízení, jeho možnosti budou rozšířené. Bude moci nahlížet na všechny vytvořené rezervace, spravovat jejich stav a parametry. Dokáže vytvořit a odstranit blokaci pokojů. Blokací pokojů se rozumí zabránění vytváření rezervací v daném termínu pro daný pokoj. Dokáže si zobrazit všechny registrované uživatele. Bude moci spravovat globální nastavení webové aplikace.



Obrázek 7 – Diagram případů užití. Zdroj: autor

5 Návrh aplikace

V následující kapitole bakalářské práce jsou vysvětleny funkcionality vyplývající z analýzy webové aplikace v předchozí kapitole. Pro lepší znázornění návrhu webové aplikace obsahuje také znázornění struktury databáze pomocí diagramu a mapu rozložení stránek.

5.1 Přihlašovací systém

Přihlašovací systém musí dovolit uživateli se registrovat, přihlásit, odhlásit, obnovit uživatelské heslo a také potvrdit emailovou adresu nově registrovaného uživatele.

Pro vytvoření přihlašovacího systému je hlavní entitou uživatel, který obsahuje:

- **username** pro identifikaci uživatele, hodnota v databázi nemusí být unikátní, více uživatelů může používat stejné jméno.
- **status** je hodnota string, ale využívá enum. Nabývá pouze hodnoty Pending(čekající) a Active(aktivní). Je využívána pro identifikaci, zda již uživatel potvrdil svou emailovou adresu.
- **email** je emailová adresa uživatele, kterou používá k přihlášení. Emailová adresa je unikátní pro každého uživatele.
- **password** je využíváno k uchování zašifrovaného hesla uživatele
- **isAdmin** je hodnota boolean a určuje, zda je uživatel administrátorem, nebo zákazníkem. Hodnota true značí, že je administrátorem a hodnota false, že je zákazníkem.
- **confirmationCode** je speciální stringový řetězec generovaný při registraci a je využíván pro ověření uživatelské emailové adresy.

5.2 Rezervační systém

Rezervační systém umožní uživatelům: vytvářet, upravovat a smazat rezervace. Systém dokáže vrátit požadované informace o rezervacích. Skládá se z několika hlavních entit: rezervace, skupinové rezervace, pokoje a konfigurace. V následující části je každá tato entita popsána, kromě entity konfigurace, ta je popsána samostatně v kapitole níže.

Entita pro rezervaci se skládá z atributů:

- **fromDate** je datum příjezdu zákazníka.
- **toDate** je datum odjezdu zákazníka.
- **adultsNumber** je celočíselná hodnota, označující počet dospělých v rezervaci.
- **childsNumber** je celočíselná hodnota, označující počet dětí v rezervaci.
- **isBlocked** je boolean hodnota, která určuje, jestli je rezervace blokována nebo rezervace vytvořena zákazníkem.
- **notesFromCustomer** jsou poznámky od zákazníka, který chce něco dodat informace ke své rezervaci.
- **notesFromAdmin** jsou poznámky od administrátora, zákazníkovi jsou tyto poznámky skryty.
- **status** je hodnota string, využívající enum. Nabývá pouze hodnoty pending (čekající na schválení administrátorem), approved (rezervace byla potvrzena administrátorem) a denied (administrátor zamítl rezervaci). Je využívána k určení stavu rezervace.
- **totalAmount** je číselná hodnota označující celkovou cenu rezervace, hodnota nemusí být celočíselná.
- **totalNights** je celočíselná hodnota, označující počet strávených nocí.

Skupinové rezervace obsahuje atributy:

- **reservations** je seznam rezervací, ze kterých se skupinová rezervace skládá.
- **fromDate** je datum příjezdu zákazníka.
- **toDate** je datum odjezdu zákazníka.
- **adultsNumber/childsNumber** je celočíselná hodnota, označující celkový počet dospělých/dětí ve všech rezervacích, ze kterých se skupinová rezervace skládá.
- **status** je hodnota string, využívající enum. Nabývá stejné hodnoty jako status u samostatné rezervace výše. Status skupinové rezervace, je vždy stejný jako jednotlivé rezervace, ze kterých se skládá. Nemůže

nastat stav, kdy by jedna rezervace měla jiný status než ostatní ve skupinové rezervaci.

- **totalAmount** je číselná hodnota označující celkovou cenu všech rezervací, ze kterých se skupinová rezervace skládá.

Entita pro pokoj se skládá z atributů:

- **capacity** je celočíselná hodnota, označující celkovou možnou kapacitu pokoje.
- **name** je stringová hodnota, určující název pokoje.

5.3 Mechanismus hlídacího psa

Mechanismus umožní vytvářet, smazat a získávat informace o jednotlivých hlídacích psech daného uživatele.

Pro vytvoření mechanismu hlídacího psa je hlavní entitou uživatel, který obsahuje:

- **fromDate** je datum, kdy bych chtěl uživatel přijet.
- **toDate** je datum, kdy bych chtěl uživatel odjet.
- **capacity** je celočíselná hodnota, označující kapacitu, kterou by chtěl zákazník obsadit.

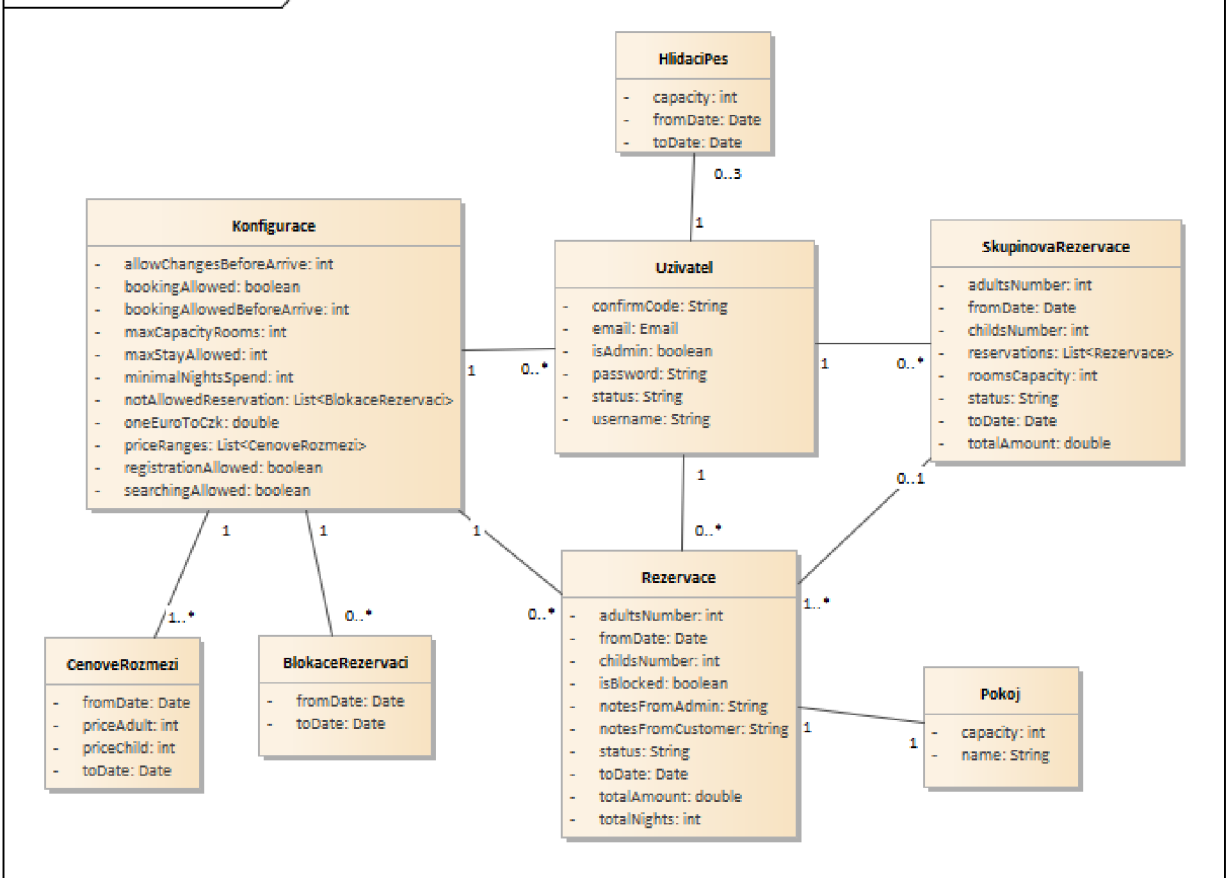
5.4 Konfigurace aplikace

Konfigurace aplikace je klíčovým nastavením celé webové aplikace, lze o ní získat informace, měnit ji a resetovat na základní. Skládá se ze tří entit: Konfigurace, Blokace rezervací a z Cenového rozmezí.

Konfigurace obsahuje atributy:

- **bookingAllowedBeforeArrive** je celočíselná hodnota, určující, kolik dní před příjezdem musí rezervace být vytvořena. Vztahuje se na zákazníka, administrátor toto pravidlo nemusí dodržet.
- **allowChangesBeforeArrive** je celočíselná hodnota, určující, kolik dní před příjezdem může být rezervace upravena, nebo smazána. Vztahuje se na zákazníka, administrátor toto pravidlo nemusí dodržet.

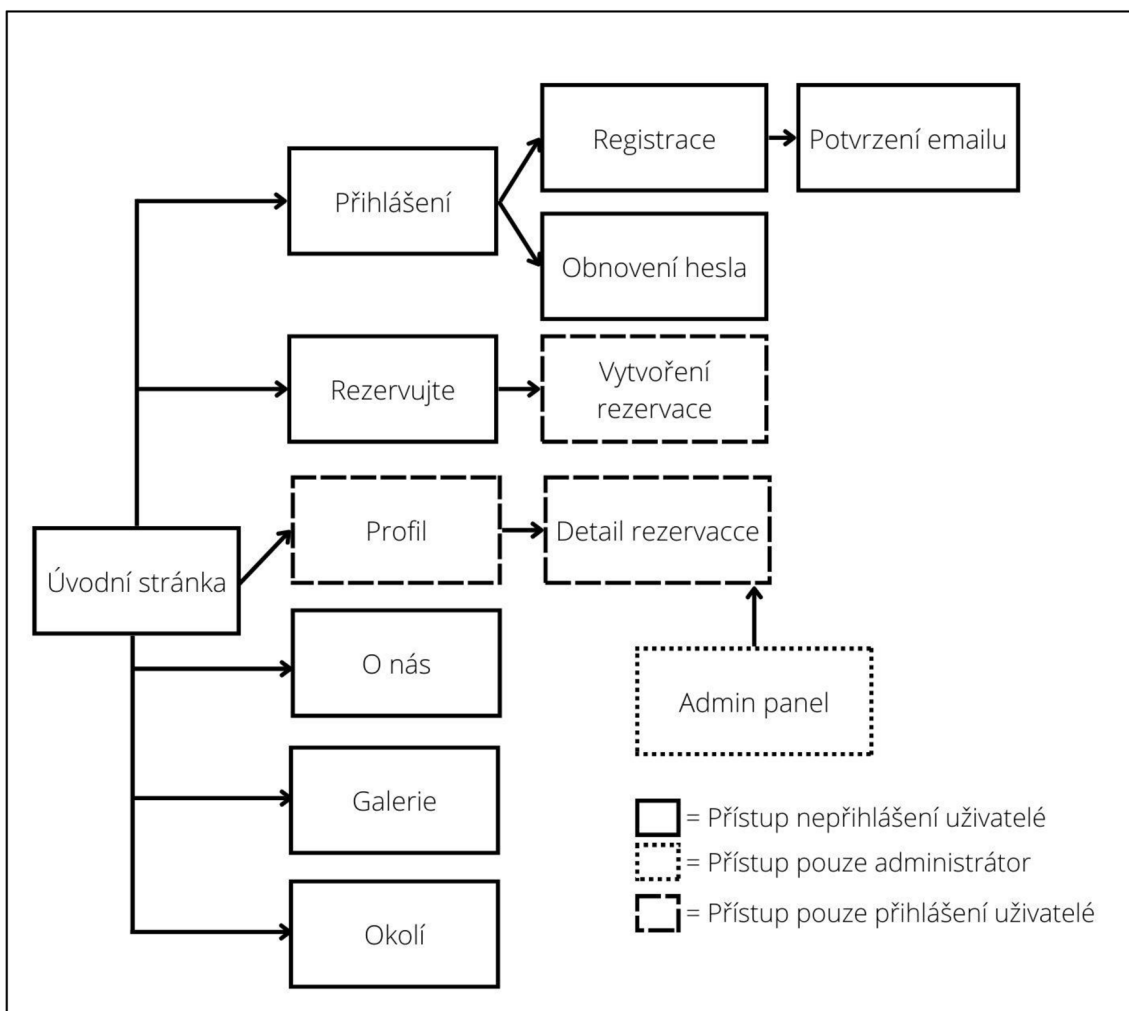
- **minimalNightsSpend/maxStayAllowed** je celočíselná hodnota, určující minimální/maximální počet strávených nocí v jedné rezervaci. Vztahuje se na zákazníka, administrátor toto pravidlo nemusí dodržet.
- **oneEuroToCzk** je číselná hodnota, určující cenu jednoho eura na České koruny. Základní výpočet cen je v Eurech.
- **bookingAllowed** je boolean hodnota, zajišťující možnost vypnout nebo zapnout celý rezervační systém. Hodnota true umožňuje vytvářet a spravovat rezervace, hodnota false vypíná rezervační systém.
- **registrationAllowed** je boolean hodnota, zajišťující možnost vypnout nebo zapnout vytváření nových registrací. Hodnota true umožňuje vytvářet nové registrace a hodnota false vypíná možnost nových registrací.
- **searchingAllowed** je boolean hodnota, zajišťující možnost vypnout nebo zapnout vyhledávání rezervací. Hodnota true umožňuje vyhledávání nových rezervací a hodnota false vypíná možnost vyhledávání rezervací.
- **priceRanges** je seznam cenových rozmezí, která určují cenu strávených nocí podle datumu. Každé cenové rozmezí se skládá z atributů: fromDate (datum příjezdu), toDate (datum příjezdu), priceAdult(cena za dospělého za 1 noc) a priceChild(cena za dítě za 1 noc). Atributy jsou využívány pro výpočet celkové ceny podle datumů v rezervaci.
- **notAllowedReservation** je seznam blokace rezervací. Každá blokace se skládá ze dvou atributů: fromDate a toDate, atributy jsou datумы určující, kdy není možné vytvářet rezervace pro celé ubytovací zařízení.



Obrázek 8 – Návrh Databáze Penzionu U Dolu. Zdroj: autor

5.5 Rozložení stránek

V následující sekci je na diagramu mapa rozložení stránek, s popisem jejich funkcionalit každé z těchto stránek a možnostmi navigace mezi nimi.



Obrázek 9 – Mapa rozložení stránek. Zdroj: autor

Úvodní stránka: Jedná se o stránku, která se zobrazí jako první při spuštění webové aplikace, obsahuje základní informace o ubytování. Každá následující stránka obsahuje také hlavní menu, které umožňuje procházet stránky dle mapy výše. Menu obsahuje možnost přesunovat po 1 a 2 úrovni mapy, následující úrovně a stránky jsou přístupné podle toho, na které stránce se uživatel nachází.

Přihlášení: Stránka umožňuje registrovaným uživatelům se přihlásit. Přes tuto stránku je následně možnost se dostat na stránku pro registraci a obnovení hesla.

O nás: Obsahuje základní informace o ubytovacím zařízení.

Galerie: Obsahuje náhled na ubytovací zařízení, v podobě obrázkového média.

Okolí: Obsahuje informace o aktivitách, které lze provádět v okolí ubytovacího zařízení.

Admin panel: Na tuto stránku je přístup, pouze přes speciálně zvolenou cestu. Vstup je umožněn pouze přihlášenému uživateli, který je administrátorem(provozovatelem). Stránka umožňuje nastavovat konfiguraci aplikace, zobrazuje informace o registrovaných uživateli a stránka umožňuje vytvářet a spravovat všechny rezervace. Stránka dovoluje vstup na stránku s detaily rezervací všech uživatelů a upravovat je.

Rezervujte: Stránka obsahuje možnosti pro vybrání termínu k rezervaci a vytvoření hlídacího psa (pokud není volný termín nalezen). Následně, pokud je uživatel přihlášený a našel volný termín, je mu umožněno se přesunout pro stránku pro vytvoření rezervace.

Profil: Na stránku s profilem je umožněn vstup pouze pokud je uživatel přihlášený. Stránka obsahuje informace o uživatelem vytvořených rezervacích, hlídacích psech a je mu, zde také umožněno změnit heslo. Stránka umožňuje přesun na stránku s detaily k příslušné rezervaci. Stránka umožňuje odstranit vytvořené hlídací psy.

6 Technologie a implementace aplikace

V následující kapitole bakalářské práce jsou pospány autorem použité technologie při implementaci webové aplikace, společně s postupy implementace webové aplikace.

6.1 Technologie

6.1.1 Javascript

Kod'ousková [6] uvádí, že „JavaScript (JS) je skriptovací jazyk učený pro tvorbu moderních dynamických webů. JavaScript je prioritně určen ke zhotovování klientských částí aplikací, s příchodem Node.js a podobných technologií však dokáže plnohodnotně figurovat i na straně serveru.”

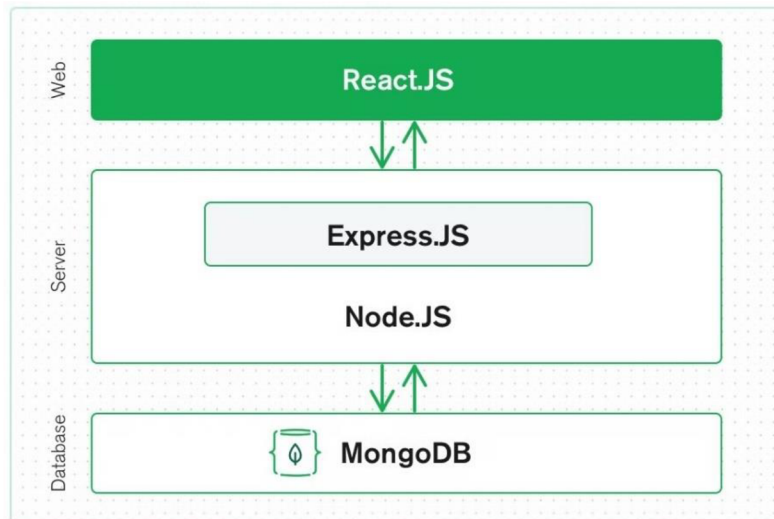
6.1.2 HTML a CSS

HTML a CSS základní stavební kameny při tvorbě webové stránky. Jazyk HTML vznikl na počátku 90.let minulého století a je vyvíjen až do současnosti, vymezuje strukturu a obsah webových stránek. Naproti tomu CSS definuje vzhled obsahu webových stránek. Jazyk CSS se nevyvíjel současně s HTML, jazyk CSS se objevil o několik let později. [7]

Při vyvážení webové aplikace je použita technologie HTML a CSS při tvorbě vzhledu a obsahu webových stránek.

6.1.3 MERN Stack

MERN je stack, který skládá ze 4 technologií, a to: MongoDB, Express.js, React.js a Node.js. Jedná se o full-stack řešení, které je rozděleno do 3 vrstev, a to konkrétně: vrstva front-end, která se skládá z technologie React.js, dále obsahuje aplikační vrstvu, která se skládá ze dvou technologií Express.js a Node.js a poslední vrstvou, kterou řešení obsahu je databázová vrstva, která se skládá z technologie MongoDB. [8] V následujících podkapitolách jsou postupně představeny tyto technologie, společně s grafickým znázorněním MERN stack vrstev.



Obrázek 10 – Vrstvy MERN stack. Zdroj: [8]

6.1.3.1 React.js

Oficiální stránka React [9] říká, že „*React je knihovna jazyka JavaScript pro vytváření uživatelských rozhraní.*” S Reactem se navrhují různé pohledy pro různé stavy aplikace, React podle stavu aplikace aktualizuje a generuje potřebný obsah. React je založen na vytváření zapouzdřených komponent. Každá komponenta si udržuje svůj vlastní stav, to umožňuje tvorbu složitých uživatelských rozhraní. [9]

Při vyváření webové aplikace je použita technologie React k vytváření různých komponent (více informací v kapitole Implementace), které se následně generují na front-endu. React používá syntaxi JSX, která umožňuje za určitých podmínek a úprav, relativně snadně používat HTML syntaxi v JavaScript kódu.

6.1.3.2 Express.js

Na oficiální stránce Express [10] se říká, že „*Express je minimální a flexibilní framework pro webové aplikace Node.js, který poskytuje robustní sadu funkcí pro webové a mobilní aplikace.*”

Express.js je webový framework pro vytváření API. Dokáže vytvářet předem definované URL cesty, tím přijímat požadavky na server ze strany klienta a díky těmto cestám spouštět specifický obslužný kód. Používá také tzv. middleware, to je něco jako prostředník pro zpracování požadavku kladeného na server. Ve své

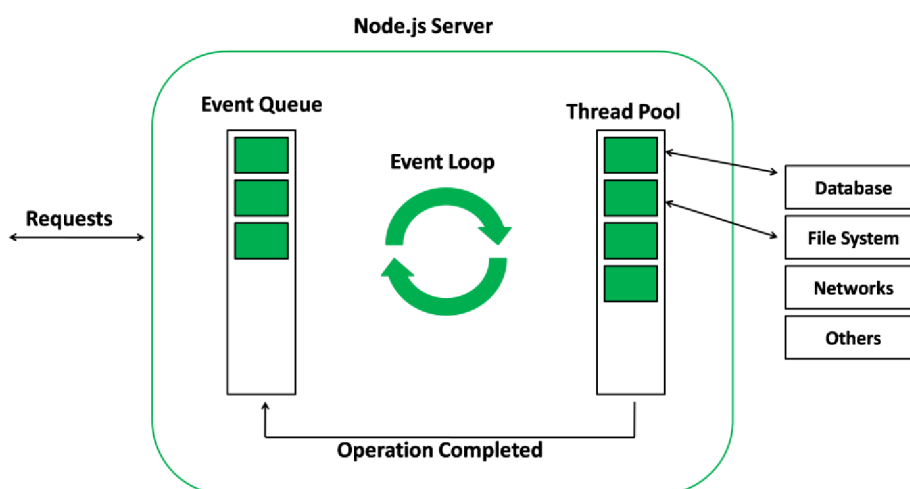
podstatě se nejedná o nic jiného, než o program, který se má spustit, než požadavek pokračuje do hlavního obslužného kódu cesty. [11]

Ve webové aplikaci je Express využit ke směrování všech požadavků na server webové aplikace.

6.1.3.3 Node.js

Máca [12] říká, že „Node.js je prostředí umožňující spouštět JavaScript kód mimo webový prohlížeč. Je postaveno na Chrome V8 JavaScript enginu, takže základ tohoto JS prostředí je stejný jako ve webovém prohlížeči Google Chrome.“ Javascript je totiž především určen ke spouštění ve webovém prohlížeči, ale Node.js nám dovoluje využívat Javascript na straně serveru.

Node.js má takzvanou neblokující I/O(asynchronní) povahu, je jedno vláknově orientovaný. Toto jedno vlákno má za úkol zpracovávat všechny příchozí požadavky, využívá k tomu tzv. smyčku událostí (Event Loop), která zajišťuje vysokou škálovatelnost. Smyčka využívá frontu FIFO pro zpracování požadavků/událostí. Přejde-li nový požadavek a vlákno již pracuje na jiném, je nový požadavek vložen do fronty a čeká se na dokončení původního běžícího, nebo jiných ve frontě před ním. Obsahuje-li požadavek neblokující událost např. čtení ze souboru nebo databáze vlákno nečeká na data z databáze, ale začne zpracovávat další požadavek, poté databáze vrátí callback funkci do fronty a ta čeká na obsluhu. [13+14]



Obrázek 11 – Node.js Server. Zdroj: [13]

Ve webové aplikaci je Node.js využit jako hlavní článek společně s express modulem a dalšími. Pomocí Node.js je umožněno sjednotit front-end a back-end na jeden server.

6.1.3.4 MongoDB

MongoDB je NoSQL databáze, která používá kolekce dokumentů. Každý vložený dokument do databáze je podobný JSON objektu, skládajícího se z klíčů a příslušných hodnot spjatých s klíčem. Dokumenty jsou identifikovány speciálním unikátním klíčem. [8]

Ve webové aplikaci je MongoDB použito společně s Mongoose modulem, který dokáže vytvářet schéma dokumentu a validovat tyto dokumenty před vložením do databáze.

6.2 Implementace

Při vytváření webové aplikace bylo použité vývojové prostředí VScode, které nabízí ideální řešení pro vývoj webových aplikací s možností instalací rozšíření, která vývoj ještě více usnadňují.

6.2.1 Struktura

V kořenovém adresáři webové aplikace se nachází soubor server.js, který má na starosti celkové spuštění webové aplikace (Node.js), tento soubor spojuje modul Express a React a to tak, že Express má na starosti všechny příchozí požadavky, ale když se nejedná o požadavek na API je přesměrován na REACT aplikaci, tím je možné celou webovou aplikaci spouštět na jednom serveru.

Dalším důležitým souborem v kořenovém adresáři je soubor Procfile, ten má na starosti příkazy, které se mají vykonat po nasazení na server Heroku. V tomto případě se zde nachází příkaz ke spuštění celé webové aplikace.

Soubor .env v kořenovém adresáři má na starosti ukládání všech důležitých a unikátních údajů, které by neměly být sdíleny a jsou využívány ve webové aplikaci, jako je například URI pro připojení databáze, hesla a hashovací klíč pro šifrování hesel. Tento soubor je využíván jen při lokálním spuštění webové aplikace při

nasazení na server Heroku, jsou využívány Konfigurační proměnné poskytované Heroku.

Dále se v adresáři nachází dvě složky konkrétně složka `api` a `client`. Složka `api` je učená pro celou logiku a správu back-endu a naopak složka `client` pro logiku front-endu, v tom to případě technologie React.

Ve složce **api** se nacházejí složky:

- **middleware** – tato složka obsahuje funkce přes které požadavky mohou procházet, zároveň se zde nachází pomocné funkce pro fungování webové aplikace. Příkladem middleware je například, autorizace a zda je rezervaci možné vytvořit.
- **models** – v této složce se nacházejí všechna modely/schéματα entit ukládaných do databáze.
- **routes** – zde se nacházejí soubory s logikou požadavků na API server, nachází se zde například soubory pro, rezervace, pokoje, uživatele, konfiguraci a hlídacého psa. V každém tomto souboru jsou definované funkce, které se mají vykonat pro požadavky.

Ve složce **client** se nacházejí složky:

- **public** – zde se nachází statické soubory, jako například obrázky v podsložce **images**, nebo také soubory pro jazykovou lokalizaci v podsložce **localization** a videa v podsložce **videos**.
- **src** – v této složce se nachází celé srdce React aplikace. Složka obsahuje soubor **index.js**, který má na starosti spuštění celé front-end části webové aplikace, dále obsahuje soubor **App.js**, který má na starosti směrování všech požadavků na front-end a zobrazovat příslušné pohledy. V souboru **App.cs** je nastavení vzhledu webové aplikace. Složka **src** zároveň obsahuje další podsložky a to konkrétně:
 - **components** – v této složce se nachází znovu použitelné komponenty v aplikaci.

- **css** – zde se nachází css soubory pro Bootstrap, společně s css importovaným pro Admin panel.
- **screens** – složka obsahuje pohledy, které se zobrazují uživateli podle cesty, kterou vyžaduje.

6.2.2 Aplikační vrstva

V následujících podkapitolách jsou vysvětleny hlavní logické prvky webové aplikace. Všechny uvedené URL adresy jsou uváděny jako konečné části URL. Například cesta „/login“ by mohla znamenat „www.domenaPříklad.cz/login“.

6.2.2.1 Nasazení aplikace

Pro zveřejnění aplikace a zpřístupnění aplikace na internetu je použit server Heroku(<https://www.heroku.com/>). Heroku nabízí spuštění jedné aplikace zdarma s omezením, a to, že poté, co je aplikace neaktivní několik minut je pozastavena. Tento problém je vyřešen pomocí služby New Relic(<https://newrelic.com/>). Je využívána služba monitor, která je opět zdarma a zasílá každých 15 minut požadavek pro otevření aplikace. Tímto způsobem je pozastavování aplikace zamezena.

Heroku poskytuje URL adresu zdarma se zabezpečením SSL. Pro nastavení vlastní domény je využívána služba Cloudflare(<https://www.cloudflare.com/>). Pro kompletní šifrování SSL je lepší využívat placené funkce Heroku.

Heroku také nabízí propojení s GitHub větví pro automatické nasazení tato funkce je využívána pro usnadnění nasazení aplikace.

6.2.2.2 Přihlášení

Stránka pro přihlášení je dostupná na adrese „/login“, při příchodu na tuto adresu se načítán pohled „client/src/screens/LoginScreen.js“. Stránka obsahuje jako hlavní prvek formulář, který vyžaduje emailovou adresu a heslo. Formulář nepovolí své odeslání na server, dokud se nejedná o validní e-mailovou adresu a heslo se neskládá, alespoň z jednoho znaku (pro zvýšení bezpečnosti).

Formulář je následně odeslán na adresu „/api/users/login“ jako POST požadavek, kde se vyhledá uživatel v databázi pomocí e-mailové adresy a potvrdí se správnost

jeho hesla pomocí modulu bcrypt a pokud je ověření úspěšné, je mu vygenerován token, který je následně uložen do cookie a odeslán na front-end, kde následně probíhá přesměrování na domovskou stránku.

Nastane-li v nějakém kroku chyba, server vrátí chybou odpověď na požadavek a uživatel je graficky upozorněn. Pohled zároveň umožňuje přesměrování na pohled pro registraci nebo obnovení hesla, odkazy pro přesměrování jsou umístěny pod formulářem pro přihlášení.

6.2.2.3 Registrace

Registrační stránka je dostupná na adrese „/register“ hlavním prvkem je opět formulář, který se načítá z pohledu „client/src/screens/RegisterScreen.js“. Formulář vyžaduje uživatelské jméno, e-mail, heslo a potvrzení hesla. Validace na straně klienta probíhá pro celý formulář, validuje se, zda je uvedeno uživatelské jméno, jestli v položce e-mail je opravdu e-mailová adresa, u hesla a potvrzení hesla se kontroluje, jestli je heslo dost silné a jestli se hesla shodují.

Formulář se odesílá na adresu „/api/users/register“ jako POST požadavek. Na straně serveru se nejdříve požadavek pokusí projít přes middleware, který ověřuje, zda jsou povolené registrace, pokud ano, pokračuje požadavek dál, jestli jsou registrace vypnuté, vrací chybovou odpověď. Ověří se, zda je e-mailová adresa již používána, pokud je již používána, vrací server odpověď s příslušnou chybovou hláškou a uživatel je graficky upozorněn. Na straně serveru je prováděná opětovná validace síly hesla pomocí modulu validator. Po úspěšné validaci a vložením uživatele do databáze se server pokusí odeslat e-mail s linkem pro potvrzení e-mailové adresy. Link obsahuje JWT token vygenerovaný před uložením uživatele do databáze, díky tomuto tokenu se uživatel autorizuje při kliknutí na link. Při neúspěšném odeslání e-mailu je uživatel z databáze odebrán a je vrácena chybová odpověď.

Příchozí e-mail obsahuje link pro potvrzení e-mailové adresy, po kliknutí na tento link je uživatel přesměrován na adresu „/confirmEmail/confirmationCode“, kde confirmationCode je serverem vygenerovaný JWT token. Pohled „client/src/screens/EmailConfirmScreen.js“ je vykreslován a při načítání je odeslán požadavek POST na adresu „/api/users/confirmEmail“ s tímto tokenem pro

potvrzení e-mailové adresy uživatele. Na serveru se token validuje. Po úspěšné validaci se v databázi vyhledá uživatel se stejným tokenem a pokud ještě nebyl aktivován, tak se změní jeho stav na aktivovaný a je uložen do databáze. Nastane-li nějaké chyba v tomto procesu, nebo uživatel je již aktivován je vrácená chybová odpověď.

6.2.2.4 Obnovení hesla

Stránka pro obnovení hesla je přístupná na adrese „/forgot-password“, načítá pohled „client/src/screens/ForgotPasswordScreen.js“. Nachází se na ní opět formulář, v tomto případě pouze s kolonkou pro e-mail, který se validuje na straně uživatele a nedovolí odeslání nevalidní e-mailové adresy.

Po odeslání formuláře na server se opět validuje příchozí požadavek, zda obsahuje opravdu e-mailovou adresu, server se pokusí najít uživatele a po úspěšném nalezení uživatele je generován link pro změnu hesla, k tomu je zapotřebí vytvořit speciální JWT token, který je validní pouze 30 minut a k hlavnímu klíči pro generování JWT tokenů je přidáno jeho zašifrované heslo z databáze.

Uživateli se odešle mail s linkem pro změnu hesla pomocí modulu nodemailer. Link má podobu „/reset-password/:id/:token“, kde id je uživatelovo id, který žádá o obnovení hesla a token je serverem speciálně vygenerovaný token.

Po kliku na link je vykreslen pohled „client/src/screens/ResetPasswordScreen.js“. Pohled před načtením odešle na server POST požadavek na adresu „/api/users/reset-password/isValid“ s tělem obsahující id a token. Server následně validuje, zda je možné pokračovat ke změně hesla. Pokud není link validní, je serverem odeslána adekvátní odpověď a uživatel je následně přesměrován na pohled pro obnovení hesla. Pokud je ale link v pořádku, zobrazí se formulář pro zadání hesla a potvrzení hesla, formulář opět nedovolí odeslání nevalidních dat.

Po kliknutí na potvrzení změny hesla je na server odeslán POST požadavek na adresu „/api/users/reset-password/change“ s tělem obsahující id, token a nové heslo. Server validuje atributy a při úspěchu je nové zašifrované heslo uloženo do databáze. Při tomto uložení se znehodnotí všechny doposud vygenerované linky pro změnu hesla, protože k vytvoření těchto tokenů bylo použito staré zašifrované heslo

jako součást hlavního klíče pro generování JWT tokenů. Při úspěšné změně hesla je uživatel přesměrován na stránku pro přihlášení.

6.2.2.5 Rezervace

Před vytvořením rezervace je nutné nejdříve vyhledat volné pokoje s požadovanou kapacitou, k tomu slouží pohled „client/src/screens/SearchScreen.js“, který se vykresluje na adrese „/search“. Pohled vykresluje informace o cenách, minimální a maximální počet strávených nocí.

Všechny tyto hodnoty se načítají dynamicky s vykreslováním stránky z globální konfigurace. Pohled odešle GET požadavek na adresu „/api/config“, server vyhledá konfiguraci upraví maximální datумы a odešle aktuální konfiguraci jako odpověď.

Dále se na stránce nachází kalendář, ve kterém si uživatel může vybrat termín, kdy chce vytvořit rezervaci, následně se mu zobrazí možnost výběru požadované kapacity a tlačítko pro vyhledání volných pokojů. Po kliknutí na tlačítko pro vyhledání volných pokojů se odešle požadavek na server na adresu „/api/rooms/findAvailableRooms?fromDate=DD-MM-YYYY&toDate=DD-MM-YYYY“. Server ověří, zda je možné vyhledávat, validuje příchozí datумы, prohledá všechny pokoje a vrátí volné pokoje jako odpověď.

Po vybrání parametrů uživatelem mohou nastat dva stavy a to, že server vrátil volné pokoje, díky tomu se vykreslí tabulka, ze které si uživatel může vybrat, které pokoje chce rezervovat a po kliknutí na tlačítko vytvořit rezervaci je přesměrován na stránku pro vytvoření rezervace. Druhou možností je, že pro daný termín a kapacitu nejsou volné pokoje, v tom případě je uživatel na tento stav upozorněn a je mu umožněno vytvořit hlídacího psa/připomínku, přejeli si uživatel vytvořit hlídacího psa, klikne na tlačítko vytvořit připomínku.

Pro vytvoření rezervace a hlídacího psa musí být uživatel přihlášen, když není, je na to graficky upozorněn a není mu dovoleno pokračovat. Po kliknutí na tlačítko vytvořit připomínku, je odeslán požadavek POST obsahující v těle informace o požadované kapacitě a dnech odjezdu a příjezdu. Požadavek je odeslán na adres „/api/watchDog“ na straně serveru se ověří možnost vytvoření psa a je přidat do databáze. Kontrola připomínek probíhá každých 30 s pomocí CRON plánovače v Node.js, kdy se

pokaždé zkontroluje, zda je kapacita v daném termínu volná, pokud ano, je odeslán e-mail s upozorněním a pes se odstraněn.

Kalendář zároveň obsahuje, graficky znázorněné dny, kdy ubytovací zařízení nemá volné žádné pokoje. Toto znázornění se dynamicky generuje pro každý zobrazený měsíc.

Posílá se POST požadavek na server na adresu „/api/rooms/checkDays“ s některým dnem v měsíci, pro který chceme zjistit obsazenost pokojů. Server ověří, zda je možné vyhledávat – porovná globální nastavení aplikace, proběhne validace požadované měsíce, zjistí se, jestli požadovaný měsíc není v minulosti, nebo moc v budoucnosti, následně se pro každý den zjistí jestli, existuje nějaký volný pokoj, pokud ano, je den adekvátně označen, po kontrole všech dnů v měsíci je odeslána odpověď uživateli s dny v měsíci se statusem, jestli má den volné pokoje nebo ne. Následně jsou tyto hodnoty graficky znázorněny pro zjednodušení volby termínu.

srpen		2022				
pon	úte	stř	čtv	pát	sob	ned
1	<u>2</u>	3	4	5	6	7
8	9	10	11	12	13	14
15	16	✗	✗	✗	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Obrázek 12 – Grafické znázornění maximální obsazenosti. Zdroj: autor

Červené křížky znamenají, že všechny pokoje jsou v daný den obsazeny, šedivé označené dny jsou nedostupné dny, v závislosti na nastavení konfigurace, jedná se o blokace nebo o minimální počet dní před příjezdem nové rezervace. Modře podtržený je dnešní den.

Poté, co si uživatel vybral termín a zvolil volné pokoje nebo pokoj, který chce rezervovat, tak po kliknutí na tlačítko vytvořit rezervaci, otevře nový pohled „client/src/screens/BookingScreen.js“, který je na adrese „/book/DD-MM-YYYY/

DD-MM-YYYY/+idPokoje+idPokoje“, kde první datum odpovídá příjezdu a druhé datum odjezdu a idPokoje jsou id pokojů, kterých se vytvoření rezervace týká.

Na stránce se vykreslí tabulka pro vyplnění počtu dětí a dospělých pro jednotlivé pokoje. Stránka obsahuje, také možnost vyplnění poznámky k rezervaci od uživatele a další základní informace o rezervaci, jako je například přibližná cena.

Po kliknutí na tlačítko vytvořit rezervaci je odeslán požadavek POST na příslušnou adresu, záleží, jestli se jedná o rezervaci s více nebo jedním pokojem. Pro jeden pokoj je to adresa „/api/bookings/createBooking“ a pro rezervaci s více pokoji je to adresa „/api/bookings/createGroupBooking“.

Jedná-li se o rezervaci s jedním pokojem server nejdříve autorizuje uživatele (uživatel musí být přihlášený), potvrdí možnost vytváření rezervace (porovnání s globálním nastavením aplikace), validují se vstupní data a zjistí se, jestli je rezervace opravdu možné vytvořit – jestli nezasahuje do jiných rezervací a další pravidla, poté co je rezervace v pořádku, je uložena do databáze.

Jedná-li se o rezervaci s více pokoji server provádí stejné kroky jako u rezervace s jedním pokojem jen s tím rozdílem, že je vytvořena nová rezervace pro každý pokoj obsahující id skupinové rezervace (rezervace s více pokoji) pro identifikaci dílčích rezervací.

K vytvoření je použita transakce, při neúspěšném vytvoření nějaké z dílčích rezervací nebo samotné skupinové rezervace je databáze obnovena do konzistentního stavu. Nastane-li chyba v tomto procesu, je odeslána adekvátní odpověď. Po vytvoření rezervace je uživatel graficky upozorněn a přesměrován a svůj profil.

```

1  const jwt = require('jsonwebtoken')
2  //Autorizace uživatele pomocí cookie
3  const auth = (req, res, next) => {
4    try {
5      const token = req.cookies.token
6      const user = jwt.verify(token, process.env.JWT_SECRET)
7      req.user = user
8      next()
9    } catch (e) {
10     res.clearCookie("token")
11     res.clearCookie("name")
12     res.send({ error: 'Prokažte svou totožnost.(Přihlašte se)', status: "error" })
13   }
14 }
15 module.exports = { auth }

```

Obrázek 13 – Middleware pro autorizaci. Zdroj: autor

Middleware nalezne cookies s názvem token a poté tento token ověří pomocí JWT a pokud je token validní požadavek pokračuje dál. Při neúspěchu jsou cookies vymazány a vrácena chybová hláška.

6.2.2.6 Profil

Stránka s profilem se nachází na adrese „/profile“, při příchodu se vykresluje pohled „client/src/screens/ProfileScreen.js“, stránku může navštívit pouze přihlášený uživatel. Stránka obsahuje tabulku všech dosavadních rezervací uživatele, tabulku vytvořených připomínek a také obsahuje formulář pro změnu uživatelského hesla.

Při vykreslování stránky jsou odeslány požadavky GET pro získání rezervací a připomínek. Pro získání rezervací je adresa požadavku „/api/bookings/getAllBookingsUser“, na straně serveru se vyhledají všechny rezervace uživatele a skryjí se nepotřebné údaje, například poznámka od administrátora a výsledek se odešle uživateli. Pro získání všech připomínek uživatele je to adresa „/api/watchDog/getAllUsersDogs“, na serveru se vyhledají připomínky uživatele a jsou odeslány zpět uživateli v odpovědi.

Tabulka pro znázornění rezervací uživatele obsahuje informace o pokojích, příjezdu, odjezdu, počtu nocí, stavu rezervace, obsazenosti a ceně. Každá rezervace má také speciální tlačítko pro otevření změn rezervace. Po kliknutí je uživatel přesměrován na pohled pro úpravu rezervace, je zohledněno, jestli se jedná o rezervaci s více nebo jen jedním pokojem.

Tabulka zobrazující hlídací psy (připomínky) obsahuje informace o tom, kdy by chtěl uživatel přijet a odjet, společně s kapacitou, kterou by chtěl obsadit. Dalším prvkem tabulky je tlačítko pro každou připomínku, které odstraní příslušnou připomínku. Po kliknutí na tlačítko odstranit je odeslán DELETE požadavek s id připomínky, kterou uživatel chce odstranit. Požadavek je odeslán na adresu „/api/watchDog“, na straně serveru se uživatel autorizuje, vyhledá se příslušná připomínka a ta je vymazána z databáze. Při úspěchu/neúspěchu je odeslána adekvátní odpověď a uživatel je graficky upozorněn. Po odstranění je opět odeslán požadavek pro získání současných připomínek pro aktualizaci tabulky s připomínkami.

Posledním prvkem pohledu je formulář pro změnu uživatelského hesla. Formulář nedovolí odeslání nevalidního hesla. Po kliknutí na tlačítko změnit heslo je odeslán požadavek POST s novým heslem uživatele na adresu „/api/users/changePassword“. Server zkontroluje, zda je uživatel přihlášený, validuje heslo a uloží ho zašifrované do databáze. Server odešle odpověď o úspěchu/neúspěchu změny hesla a uživatel je graficky upozorněn.

6.2.2.7 Úprava rezervace

Existují dva typy rezervací, a to rezervace jednoho pokoje a více pokojů. Adresa pro úpravu rezervace s jedním pokojem je „/bookdetails/idRezervace“ a pro správu rezervace s více pokoji je to adresa „/updateGroup/ idRezervace“, kde idRezervace je unikátní identifikátor každé rezervace. Každý typ rezervace zobrazuje svůj vlastní pohled, odkaz pro zobrazení konkrétní rezervace se nachází v tabulce rezervací v pohledu pro profil uživatele.

Při otevření pohledu pro rezervaci s jedním pokojem se načítá pohled „/client/src/screens/BookingDetailsScreen.js“, na adrese „/bookdetails/idRezervace“. Pohled při načítání odešle na server požadavek POST na adresu „/api/bookings/getBookingDetails“ s idRezervace v těle požadavku. Server následně validuje vstup a zjistí se, jestli se jedná o administrátora, který odeslal požadavek k obdržení informací o rezervaci, pokud se nejedná o administrátora, je před odesláním odpovědi vymazáno pole notesAdmin, aby uživatel neměl přístup k poznámce od administrátora k dané rezervaci. Následně je odeslán požadavek GET na adresu „/api/users/isAdmin“ pro získání informace,

jestli se jedná o administrátora, aby bylo možné rozlišit různé nastavení pohledu pro administrátora a zákazníka. Dále je odeslán požadavek GET na adresu „/api/config“ pro získání aktuálního konfigurace pro aplikaci. Konfigurace se získává z důvodu validace na straně klienta, ještě před odesláním změny rezervace na server. V poslední radě se odešle požadavek GET na adresu „/api/rooms/getAllRooms“ pro získání všech pokojů, aby bylo možné změnit pokoj.

Pohled obsahuje kalendář pro změnu datumu a příjezdy rezervace, dále obsahuje tabulku s informacemi jako je: pokoj rezervace, počet nocí, status, počet dětí a dospělých a celková cena. Pohled zároveň obsahuje poznámky k rezervaci od zákazníka, které může zákazník měnit. Ve spodní části podhledu se nachází informace a pravidla pro změnu rezervace, zároveň je zde tlačítko pro zrušení rezervace a pro uložení změn rezervace.

Po kliknutí na tlačítko pro zrušení rezervace je odeslán požadavek DELETE na adresu „/api/bookings/deleteSingleBooking“ s tělem obsahujícím ID rezervace, o kterou chce uživatel odstranit.

Na straně serveru se ověří, zda je uživatel přihlášen, zda je zapnutý systém pro rezervace (porovná se nastavení konfigurace), ověří se, zda je uživatel oprávněn smazat rezervaci (jedná-li se o administrátora nebo majitele rezervace). Administrátor může smazat libovolnou rezervaci bez omezení, ale u zákazníka se ověřuje, zda je majitelem rezervace a není-li datum příjezdu příliš blízko dni definovaném konfigurací, kdy se může rezervace měnit/smazat.

Po kliknutí na tlačítko pro uložení změn rezervace je odeslán požadavek PATCH na adresu „/api/bookings/updateBooking“ s tělem obsahujícím všechny atributy, které chce uživatel na rezervaci změnit. Tělo požadavku obsahuje atributy: bookingId (pro identifikaci rezervace), fromDate (datum příjezdu ve formátu DD-MM-YYYY), toDate, roomId, adultsNumber (počet dospělých), childsNumber (počet dětí), status, notesFromCustomer, notesFromAdmin. Na straně serveru se ověří, zda je uživatel přihlášený, jestli je zapnutý systém pro rezervace, ověří se, zda je uživatel oprávněn upravit rezervaci (jedná-li se o administrátora nebo majitele rezervace), zkontroluje se, jestli tělo obsahuje všechny požadované atributy ve správném formátu, zkontroluje se, jestli se jedná skutečně o změnu rezervace (kontrola, zda atributy nejsou úplně stejné jako současná rezervace), provede se kontrola, zda je

možné rezervaci uložit s ohledem na to jestli uživatel, který požaduje změnu je administrátor nebo majitel rezervace.

Při kontrole, zda je možné rezervaci uložit se kontroluje: validita datumu příjezdu a odjezdu, jestli nová rezervace není v některém rozsahu blokace rezervací (kontroluje se jen pro uživatel, ne pro administrátora), počet strávených nocí (kontrola podle nastavení konfiguračního souboru), zkontroluje se, zda je dostatečná kapacita pro pokoj, následně se zkontrolují všechny rezervace pro daný pokoj a zjistí se, jestli nově upravená rezervace nebude zasahovat do některé z již uložených rezervací, v poslední části se ověří zda se nejedná o rezervaci s více pokoji, jestli rezervace není blokací a následně se aktualizuje rezervace dle atributů a změny se uloží do databáze. Při neúspěchu je odeslána adekvátní odpověď a uživatel je graficky upozorněn.

Při otevření pohledu pro rezervaci s více pokoji se načítá pohled „/client/src/screens/GroupBookingDetails.js“, na adrese „/updateGroup/idRezervace“. Pohled při načítání odešle na server požadavek POST na adresu „/api/bookings/getGroupBookingDetails“ s idRezervace v těle požadavku. Server následně provede stejnou validaci, jako u rezervace pro jeden pokoj, zahrnující notesAdmin atribut. Následně je odeslán požadavek GET na adresu „/api/users/isAdmin“ pro získání informace, jestli se jedná o administrátora, aby bylo možné rozlišit různé nastavení pohledu pro administrátora a zákazníka. Dále je odeslán požadavek GET na adresu „/api/config“ pro získání aktuální konfigurace pro aplikaci. Konfigurace se získává z důvodu validace na straně klienta, ještě před odesláním změny rezervace na server. Následně jsou odeslány požadavky POST na adresu požadavek POST na adresu „/api/bookings/getBookingDetails“ s idRezervace v těle požadavku, idRezervace jsou získávány z prvního požadavku odeslaného podhledem. Id rezervací se nachází v odpovědi pod atributem reservations. Požadavky jsou odeslány pro zjištění informací o jednotlivých dílčích rezervacích, které jsou obsazené ve skupinové rezervaci. V poslední radě se odešle požadavek GET na adresu „/api/rooms/getAllRooms“ pro získání všech pokojů, aby bylo možné změnit pokoj.

Pohled obsahuje kalendář pro změnu datumu a příjezdu rezervace, dále obsahuje tabulku s informacemi jako je: pokoj rezervace, počet nocí, status, počet dětí a

dospělých a pro každý záznam v tabulce (dílní rezervaci), existuje tlačítko, které odebere příslušný pokoj. Pohled zároveň obsahuje poznámky k rezervaci od zákazníka, které může zákazník měnit. Ve spodní části podhledu se nachází informace a pravidla pro změnu rezervace, zároveň je zde tlačítko pro zrušení rezervace a pro uložení změn rezervace.

Po kliknutí na tlačítko pro zrušení rezervace je odeslán požadavek DELETE na adresu „/api/bookings/deleteGroupBooking“ s tělem obsahujícím ID rezervace, o kterou chce uživatel odstranit. Na straně serveru se ověří, zda je uživatel přihlášen, zda je zapnutý systém pro rezervace (porovná se nastavení konfigurace), ověří se, zda je uživatel oprávněn smazat rezervaci (jedná-li se o administrátora nebo majitele rezervace). Administrátor může smazat libovolnou rezervaci bez omezení, ale u zákazníka se ověřuje, zda je majitelem rezervace a není-li datum příjezdu příliš blízko dni definovaném konfigurací, kdy se může rezervace měnit/smazat. Následně se odstraní jednotlivé dílní rezervace, společně se skupinovou rezervací. Pro zajištění konzistence databáze je využívána transakce.

Po kliknutí na tlačítko pro uložení změn rezervace je odeslán požadavek PATCH na adresu „/api/bookings/updateGroupBooking“ s tělem obsahujícím všechny atributy, které chce uživatel na rezervaci změnit. Tělo požadavku obsahuje atributy: bookingGroupId (pro identifikaci skupinové rezervace), fromDate (datum příjezdu ve formátu DD-MM-YYYY), toDate, status, notesFromCustomer, notesFromAdmin a bookings. Atribut bookings obsahuje list dílních rezervací, které chce uživatel upravit. Každý prvek listu obsahuje atributy jako: bookingId (id dílní rezervace), roomId, adultsNumber, childsNumber, notesFromCustomer a notesFromAdmin. Na straně serveru ověří, zda je uživatel přihlášený, jestli je zapnutý systém pro rezervace, ověří se, zda je uživatel oprávněn upravit rezervaci (jedná-li se o administrátora nebo majitele rezervace), zkontroluje se, jestli tělo obsahuje všechny požadované atributy ve správném formátu, zkontroluje se, jestli se jedná skutečně o změnu rezervace (kontrola, zda atributy nejsou úplně stejné jako současná rezervace), provede se kontrola, zda je možné rezervaci uložit s ohledem na to, jestli uživatel, který požaduje změnu, je administrátor nebo majitel rezervace. Pro zajištění konzistence databáze je opět využívána transakce. Následně se zkontrolují

pravidla, jestli lze rezervaci uložit a při neúspěchu je odeslána adekvátní odpověď a uživatel je graficky upozorněn.

6.2.2.8 Administrátorský panel

Stránka pro administrátora se nachází na adrese „/admin“, při příchodu se vykresluje pohled „client/src/screens/AdminScreen.js“, stránku může navštívit pouze přihlášený uživatel, který je ale administrátor. Stránka obsahuje tabulku všech rezervací k úpravě a znázornění jednotlivých rezervací, sekci pro změnu konfigurace aplikace a také tabulku všech registrovaných uživatelů. Při vykreslování stránky jsou odeslány požadavky GET pro autorizaci, ověření, zda je uživatel administrátor a pro získání konfigurace, všech pokojů, konfigurace, uživatelů a rezervací. Pro získání všech rezervací je adresa požadavku „/api/bookings/getAllBookings“, na straně serveru se ověří, zda uživatel je administrátor a vyhledají se všechny rezervace a odešle se odpověď.

Pro získání všech uživatelů je to adresa „/api/users/getAllUsers“, na serveru se vyhledají všichni uživatelé a jsou odesláni v bezpečné podobě v odpovědi. Atributy v odpovědi jsou: email, username, status, id (z odpovědi je pro větší bezpečnost odebráno zašifrované uživatelské heslo) jsou získávané z požadavku uvedeného výše. Tabulka s rezervacemi obsahuje vždy rezervace pro aktuální měsíc. Uživatel má možnost si zvolit měsíc, který potřebuje. Tabulku lze filtrovat, a to pomocí tří tlačítek, uživatel si může vybrat, které rezervace chce skrýt podle stavu daných rezervací (schválené, čekající a zamítnuté rezervace). Po kliknutí na uživatelem zvolenou rezervaci se zobrazí nabídka, v které mu je umožněno otevřít pohled s detaily konkrétní rezervace, upravovat parametry rezervace, nebo rezervaci smazat.

Sekce pro změnu konfigurace aplikace obsahuje tlačítka pro vypnutí/zapnutí vytváření/úprav rezervací, vyhledávání rezervací a registrací nových uživatelů. Dále se v této sekci nachází bloky pro vyplnění dalších nastavení jako například: minimální počet nocí (ke dni příjezdu) před vytvořením rezervace, maximální a minimální počet nocí jedné rezervace, počet dní před příjezdem, kdy je možné upravovat rezervaci, přepočítání 1 Eura na CZK a cena dospělého a dítěte podle sezóny. Sekce obsahuje také tabulky blokáží rezervací, v této části je umožněno

vyvážet/smazat blokace rezervací pro celé ubytovací zařízení (pro všechny pokoje), při vyvážení jednotlivých bloků je umožněno administrátorovi vybrat určité rozmezí bloků anebo může zvolit maximální a minimální datum (to zajišťuje pomyslnou blokaci do nekonečna v čase, na straně serveru řešeno pomocí maximální a minimální hodnoty povolené datovým formátem Date). V poslední spodní části sekce se nachází tlačítko pro povolení změn nastavení z důvodu nechtěné změny konfigurace, dále se zde nachází tlačítko pro uložení změn konfigurace a tlačítko pro resetování nastavení konfigurace na základní hodnoty.

Po kliknutí na tlačítko uložit změny je odeslán PATCH požadavek s tělem obsahujícím všechny atributy konfigurace s příslušnými hodnotami. Požadavek je odeslán na adresu „/api/config/update“, na straně serveru se uživatel autorizuje, ověří se, zda se jedná o administrátora, ověří se, zda tělo obsahuje všechny atributy s hodnotami v příslušném formátu a změny se uloží do databáze. Při kliknutí na tlačítko pro resetování nastavení je odeslán požadavek GET na adresu „/api/config/loadDefault“. Na straně serveru se zkontroluje, zda uživatel odesílající požadavek je přihlášen a zároveň jestli je administrátorem a poté jsou nahrány základní hodnoty do konfigurace. Při úspěchu/neúspěchu je odeslána adekvátní odpověď a uživatel je graficky upozorněn.

Posledním prvkem pohledu je tabulka obsahující všechny registrované uživatele, společně se základními informacemi. Tabulka je generována pomocí dat z požadavku GET „/api/users/getAllUsers“, uvedeného výše.

7 Zhodnocení aplikace

Tato kapitola obsahuje hodnocení nově vytvořené webové aplikace pro ubytovací zařízení. Hodnocení probíhá stejně jako při hodnocení jednotlivých webových aplikací v kapitole o porovnání konkurenčních webových aplikací.

7.1 Hodnocení nové webové aplikace

Z prvního hlediska SEO získaly nové webové stránky 92 bodů ze 100. Pro zlepšení SEO je doporučení pro tyto stránky doplnit soubor robots.txt.

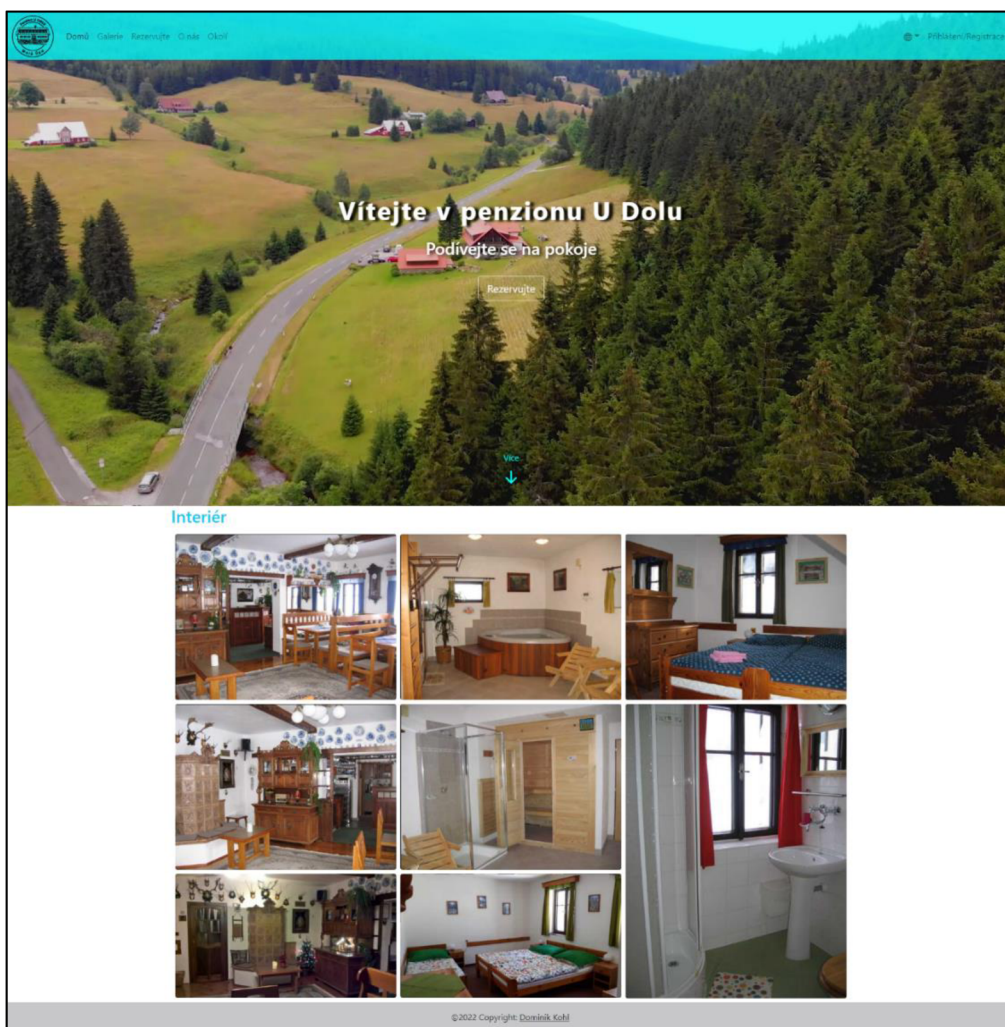
Z druhého hlediska responzivní optimalizace získaly nové webové stránky 10 bodů z 10 (100/100).

Výsledek hodnocení z posledního hlediska funkčnosti a obsahu je 70 bodů ze 100, konkrétní otázky jsou shrnuté v tabulce níže. Hodnocení v 15 jednotlivých otázkách testu je součástí autorovi práce a je uvedeno v příloze.

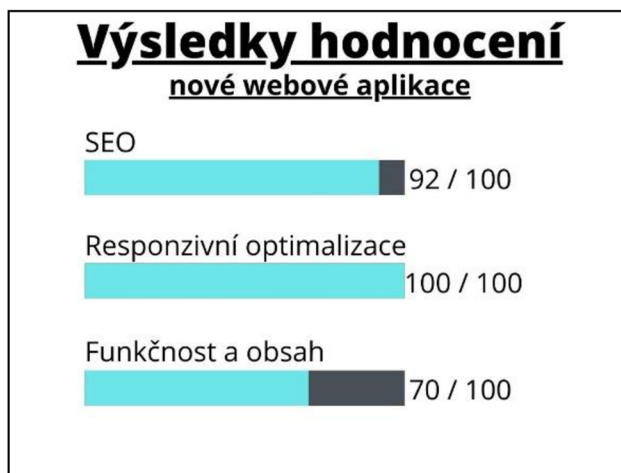
Otázka	Počet bodů	Otázka	Počet bodů
1.	0	9.	1
2.	2	10.	0
3.	2	11.	2
4.	2	12.	2
5.	1	13.	2
6.	1	14.	2
7.	1	15.	1
8.	2		
Celkem bodů: 21 / 30 (70/100)			

Tabulka 4 – Výsledky testu nové webové aplikace

7.2 Vyhodnocení aplikace



Obrázek 14 – Úvodní stránka nové webové aplikace. Generováno: [5]



Obrázek 15 – Hodnocení nové webové aplikace. Zdroj: autor

8 Shrnutí výsledků

Výsledkem práce je webová aplikace pro ubytovací zařízení, která byla vyvíjena podle požadavků zadavatele. Aplikace umožňuje uživatelům získat informace o ubytovacím zařízení. Poskytuje lepší prožitek z hlediska rezervací, bez nutnosti kontaktovat provozovatele ubytovacího zařízení s ohledem na udržení stávajících zákazníků.

Na začátku bylo vytvořené porovnání dvou konkurenčních aplikací se zadanou aplikací. Kapitola pojednává o tom, jak porovnání probíhá. Skládá se z tří kategorií a to SEO, Responzivní optimalizace a z kategorie Funkčnosti a obsahu. Bylo přesně vymezeno, jak se každá kategorie hodnotí. Kapitola současně obsahuje část o tom, jak se konkurenční ubytovací zařízení vyhledávala. Následně v této části proběhlo hodnocení jednotlivých aplikací a v závěru bylo doporučeno, co by měla webová aplikace pro ubytovací zařízení dodržovat a splňovat.

V následující kapitole byla vytvořena analýza aplikace, v této části byly představeny požadavky zadavatele na webovou aplikaci. Tyto požadavky byly sbírány v několika předem dohodnutých schůzkách se zadavatelem. V závěru se nachází případy užití s popisem jednotlivých rolí. Na základě těchto požadavků a případů užití byla následně vytvořena část zabývající se návrhem aplikace.

V části s návrhem aplikace byly vysvětleny funkcionality vyplývající z analýzy webové aplikace. Obsahuje hlavní funkcionality s příslušnými entitami, které obsahují atributy s informacemi a k čemu jsou využívány. V závěru této kapitoly bylo navrženo rozložení stránek, společně s informacemi o každé této stránce webové aplikace.

V další části byly popsány technologie a implementace webové aplikace. Část s technologiemi obsahuje informace o použitých technologiích při vytváření nové webové aplikace. Část s implementací obsahuje informace o struktuře kořenového adresáře, dále zde byly vysvětleny a popsány hlavní logické prvky webové aplikace. Mezi tyto prvky patří: nasazení aplikace, přihlášení, registrace, obnovení hesla, rezervace, profil, úprava rezervace a administrátorský panel.

V poslední kapitole zhodnocení aplikace, byla nová webová aplikace vyhodnocena jako aplikace v kapitole o porovnání konkurenčních aplikací. Tato kapitola slouží

k tomu, aby bylo možné s přesností určit, jestli nová webová aplikace je vylepšením od stávající. Z výsledků této kapitoly lze vyhodnotit, že nová webová aplikace je vylepšením od stávající.

9 Závěry a doporučení

Cílem bakalářské práce bylo vytvořit novou webovou aplikaci pro ubytovací zařízení penzionu U Dolu. Zadání práce se podařilo splnit, všechny zadavatelovy požadavky byly implementovány a bylo dokázáno, že nově vytvořená aplikace je vylepšením od stávající.

V aplikaci by se dalo změnit část zabývající se skupinovou rezervací. Bylo by vhodné změnit část zabývající se vytvořením nových rezervací, tak aby se vždy vytvářela skupinová rezervace. Tím by se omezilo vytváření rezervací s jedním pokojem, dovolilo by to lepší úpravy rezervací. Dalo by se při úpravě rezervací s jedním pokojem přidávat a odebírat pokoje. Při vytvoření rezervací s jedním pokojem je rezervace uzamčena, pokoj je umožněno změnit, ale nejde přidat další pokoj. Tímto způsobem by se vylepšil celkový zážitek z používání webové aplikace, bez velkých úprav v celkové struktuře aplikace.

Nejlépe řešením tohoto problému s rezervacemi s více pokoji by bylo odstranit kompletně rezervace s jedním pokojem, a to i z modelu a vytváření pouze skupinovou rezervací. Současně by se musela přidat možnost přidávat a odebírat pokoje k těmto rezervacím a dostatečně upravit implementaci. Tato část byla při návrhu aplikace zvolena nevhodně a dalo by se to takto vylepšit, celkovou strukturu aplikace a optimalizovat využití databáze.

Dalo by se také vylepšit optimalizace aplikace, například při kontrole možností vytvořit rezervace, nebo při vyhledávání rezervací. V některých částech webové aplikace se vyhledávají všechny rezervace, co jsou v celé aplikaci i když nejsou všechny potřeba, toto by se dalo optimalizovat tím, že vyhledáme jen určený rozsah, který je dostatečný a zbytečně by nezpomaloval aplikaci, například v závislosti na datumu a maximální délky rezervace. Na některých místech webové aplikace toto již bylo z části implementované.

Bylo by vhodné, aby aplikace používala vlastní e-mailovou doménu, a ne doménu třetí strany (outlook.cz). Vylepšením by také bylo, kdyby aplikace využívala přihlašovací systém třetí strany pro správu uživatelů (registrace, přihlášení, obnovení hesla, uchování dat o uživateli) pro vylepšení celkové bezpečnosti aplikace, protože tyto služby poskytují kvalitní zabezpečení, které stále vyvíjejí.

Další prvkem pro zlepšení je celkový vzhled a obsah stránek. Zadavatel je spokojený s celkovým výsledkem webové aplikace, ale daly by se zlepšit některé detaily, jako například vhodněji zvolené barvy, přidat vhodné fonty pro text.

10 Seznam použité literatury

- [1] BEAL, Vangie. SEO (Search Engine Optimization). Webopedia [online]. [cit. 2021-6-16]. Dostupné z: <https://www.webopedia.com/definitions/seo/>
- [2] Silktide: Nibbler [online]. [cit. 2021-6-16]. Dostupné z: <https://nibbler.silktide.com/en>
- [3] HUNTBACH, Martin. How good is your website? Grade it yourself. Jammy Digital [online]. 2017-05-20 [cit. 2021-5-24]. Dostupné z: <https://jammydigital.com/website-good-vs-bad/>
- [4] Penzion.cz [online]. ©1997-2021 [cit. 2021-5-24]. Dostupné z: <https://www.penzion.cz/>
- [5] Cloudconvert [online]. ©2021 [cit. 2021-6-16]. Dostupné z: <https://cloudconvert.com/website-jpg-screenshot>
- [6] KOŘOUSKOVÁ, Barbora. JAVASCRIPT PRO ZAČÁTEČNÍKY: CO TO JE A JAK FUNGUJE [online]. 2022-01-28 [cit. 2022-04-04]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-javascript-pro-zacatecniky>
- [7] CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. Brno: Computer Press, 2012. 16 s. ISBN 978-80-251-3733-8.
- [8] MERN Stack Explained [online]. [cit. 2022-04-03]. Dostupné z: <https://www.mongodb.com/mern-stack>
- [9] React [online]. [cit. 2022-04-03]. Dostupné z: <https://reactjs.org/>
- [10] Express [online]. [cit. 2022-04-04]. Dostupné z: <https://expressjs.com/>
- [11] Express/Node introduction [online]. 2022 [cit. 2022-04-04]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/Introduction
- [12] MÁČA, Jindřich. Lekce 1 - Úvod do Node.js [online]. [cit. 2022-04-04]. Dostupné z: <https://www.itnetwork.cz/javascript/nodejs/uvod-do-nodejs>
- [13] The Node.js Event Loop, Timers, and process.nextTick() [online]. [cit. 2022-04-04]. Dostupné z: <https://nodejs.org/en/docs/guides/event-loop-timers-and-nexttick/>
- [14] SEN, Aritra. Node.js Event Loop [online]. 2021-10-11 [cit. 2022-04-04]. Dostupné z: <https://www.geeksforgeeks.org/node-js-event-loop/>

11 Přílohy

- 1) Nově vytvořená webová aplikace pro ubytovací zařízení
- 2) Klíčové moduly při vytváření webové aplikace
- 3) Test pro hodnocení webových aplikací pro ubytovací zařízení
- 4) Podklad pro zadání bakalářské práce

Nově vytvořená webová aplikace:

K bakalářské práci je přiložen soubor, ve kterém se nachází zdrojový kód nově vytvořené webové aplikace. Soubor obsahuje také návod pro lokální spuštění, ale pro správné fungování všech funkcí aplikace, bez nutnosti doplňovat některých údajů (e-mailové údaje pro odesílání mailů...), je vhodnější použít online zkušební verzi aplikace viz. odkazy níže, aplikace je zde plně funkční.

Zkušební online verze i lokální verze obsahuje základní data:

- 6 pokojů (3 pokoje lokální verze) a základní konfigurační nastavení
- User1: e-mail: user1udolu@domkohl.cz, heslo: Heslo123
- User2: e-mail: user2udolu@domkohl.cz, heslo: Heslo123
- Admin: e-mail: adminudolu@domkohl.cz, heslo: Heslo123

Pro vyzkoušení funkcí, které odesílají e-mail je potřeba registrovat nového uživatele a použít při registraci platný e-mail (nejlépe gmail.com pro rychlejší obdržení e-mailů). Admin panel je přístupný po změně základní adresy přidáním „/admin“.

Webová aplikace je pro vyzkoušení dostupná na adrese:

<https://udolu.herokuapp.com/> nebo <https://bpudolu.domkohl.cz/>

Zdrojový kód je také dostupný na adrese:

<https://github.com/domkohl/bpudoluKohl>

Klíčové moduly:

- 1) Bcrypt - <https://www.npmjs.com/package/bcrypt>
- 2) cookie-parser - <https://www.npmjs.com/package/cookie-parser>
- 3) dotenv - <https://www.npmjs.com/package/dotenv>
- 4) jsonwebtoken - <https://jwt.io/> a
<https://www.npmjs.com/package/jsonwebtoken>
- 5) moment - <https://www.npmjs.com/package/moment> a
<https://momentjs.com/>
- 6) mongoose - <https://www.npmjs.com/package/mongoose> a
<https://mongoosejs.com/>
- 7) nodemailer - <https://www.npmjs.com/package/nodemailer>
<https://nodemailer.com/about/>
- 8) nodemon - <https://www.npmjs.com/package/nodemon>
- 9) validator - <https://www.npmjs.com/package/validator>
- 10) axios - <https://www.npmjs.com/package/axios>
- 11) react-cookie - <https://www.npmjs.com/package/react-cookie>
- 12) express-validator - <https://www.npmjs.com/package/express-validator>
- 13) node-cron - <https://www.npmjs.com/package/node-cron>
- 14) nodemailer - <https://www.npmjs.com/package/nodemailer>
- 15) sweetalert2 - <https://www.npmjs.com/package/sweetalert2>
- 16) simple-react-lightbox - <https://www.npmjs.com/package/simple-react-lightbox>
- 17) react-table - <https://www.npmjs.com/package/react-table>
- 18) react-helmet-async - <https://www.npmjs.com/package/react-helmet-async>

Test pro hodnocení:

1. Jak dostupné jsou kontaktní údaje?
 - A) Kontaktní údaje se těžko hledají. Na domovské stránce není umožněno lidem snadno najít kontaktní telefonní číslo nebo e-mail. Lidé se musí proklikat na jinou stránku, aby našli klíčové informace. (0 bodů)
 - B) Kontaktní údaje se nachází na domovské stránce, ale nejsou na viditelném místě a mohou být přehlédnuty. Mohou být poměrně malé a umístěné v zápatí stránky. (1 bod)
 - C) Kontaktní údaje jsou snadno viditelné na hlavní stránce a jsou obsaženy v nějaké formě na každé stránce. Kontaktní číslo je velké a lze na něj kliknout z mobilního zařízení. Obsahují odkaz, který přesměruje na kontaktní stránku na webových stránkách. (2 body)
2. Jak vypadá logo webových stránek?
 - A) Logo není ostré. Může vypadat rozpixelovaně a některé prvky v logu jsou špatně čitelné. Při změně velikosti nevypadá správně. (0 bodů)
 - B) Logo je ostré, ale je příliš velké. Při změně velikosti stránek zabírá logo nepřírodně velký nebo malý prostor a ostatní elementy se posouvají a stránka tak nevypadá dobře. (1 bod)
 - C) Logo je ostré. Dokonale zapadá a nezabírá mnoho místa. Vypadá proporčně s ostatními prvky na stránce a jeho velikost se mění bez ztráty přehlednosti. (2 body)
3. Obsahuje webová stránka výzvu k akci?
 - A) Stránky nemají jasnou výzvu k akci. Webové stránky nesměřují klienty tam, kam by měly např. k vytvoření rezervace pokoje v ubytovacím zařízení. Není promyšleno, jakou akci by lidé měli provést. (0 bodů)
 - B) Na webových stránkách existuje výzva k akci, např. tlačítka "kontaktujte nás" a "rezervujte pokoj", ale nejsou na první pohled dobře viditelná. Mohou se nacházet dále na stránce a můžou být snadno přehlédnuta. (1 bod)
 - C) Na domovské stránce nalezneme výraznou výzvu k akci, která je na první pohled zřetelná. Je lákavá a inspiruje k nějaké akci. (2 body)

4. Jak dobře stránka využívá obrázky?
- A) Webové stránky neobsahují mnoho obrázků. Stránky vypadají nevýrazně a obsahují bloky textu, které jsou obtížně stravitelné. (0 bodů)
 - B) Na stránkách je několik obrázků, ale nejsou konzistentní. Jedná se o nepříliš profesionální fotografie, které nepřidávají mnoho pro vyniknutí z davu. (1 bod)
 - C) Jsou použité obrázky, které přispívají k celkovému vzhledu. Obrázky jsou konzistentní na celém webu a navozují u návštěvníků pocit příjemného prostředí. Použité profesionální fotografie s vysokou kvalitou. Obrázky doplňují obsah a zvyšují jeho čitelnost. (2 body)
5. Je jasné, že se webová stránka zabývá ubytováním?
- A) Při pohledu na domovskou stránku není jasné, jaké služby jsou nabízeny. Popisy jsou vágní a služby nejsou uváděny příliš podrobně. (0 bodů)
 - B) Na domovské stránce se mluví o službách, ale na první pohled je to těžko vidět. Musíte je vyhledat nebo se prokliknout na jinou stránku, abyste je našli. (1 bod)
 - C) Nabízené služby jsou jasně vysvětleny hned na domovské stránce a lze je snadno pochopit. Informace jsou na webových stránkách snadno k nalezení a služby jsou umístěny blízko horní části domovské stránky. Mohou být dokonce zahrnuty do položek hlavního menu. (2 body)
6. Jsou webové stránky zaměřené na zákazníka?
- A) Webové stránky jsou především o ubytovacím zařízení. Hodně mluví o tom, jak je zařízení dobré a jak dlouho je v provozu, ale nemluví o problémech zákazníků a o tom, jak je může vyřešit a jaké výhody zařízení nabízí. (0 bodů)
 - B) Polovina obsahu je o ubytovacím zařízení a polovina o problémech zákazníků a o tom, jak je může vyřešit a jaké výhody zařízení nabízí. Můžou být uvedeny vlastnosti a výhody pokojů, produktů a služeb, ale mohou obsahovat mnohem více podrobností. (1 bod)

- C) Obsah na webových stránkách je zaměřen na cílovou skupinu a hodně mluví o problémech, se kterými se potýkají. Zákazníci jsou středobodem obsahu, který je vytvořen. Umožňuje jim pochopit více o službách a o tom, jaké výhody získají. (2 body)
7. Zaznamenávají webové stránky e-mailovou adresu zákazníka?
- A) Webová stránka nezachycuje žádným způsobem e-mailovou adresu a další kontaktní údaje. (0 bodů)
- B) E-mailové adresy jsou zaznamenány při registraci zákazníků, ale není použit žádný bezplatný dárek, který by klienty nalákal např. "Přihlaste se k odběru novinek". (1 bod)
- C) Pro registraci uživatele a zaznamenání jeho e-mailové adresy je nabízen výměnou nějaký druh odměny. (2 body)
8. Jak dobré je menu/navigace?
- A) Hlavní nabídka nebo navigační panel se špatně používají. Nemusí být na první pohled viditelná a nemusí vypadat jako standardní menu, tj. horizontální nebo vertikální. (0 bodů)
- B) Hlavní nabídka je přehledná a snadno viditelná, ale na první pohled není zcela zřejmé, co dělá. Návštěvník se možná bude muset prokliknout na jinou stránku, aby zjistil, čím se zabývá, a dozvěděl se více. (1 bod)
- C) Hlavní nabídka je přehledná a snadno viditelná. Snadno se používá a při pohledu na položky nabídky je zřejmé, jaké služby jsou nabízeny. Každá stránka na webu je dostupná na tři kliknutí myší. (2 body)
9. Používají webové stránky různá média?
- A) Webové stránky nepoužívají videa ani zvuk. Ke komunikaci s návštěvníky je použit pouze text a obrázky. (0 bodů)
- B) Použito jedno nebo dvě videa, která slouží k vysvětlení toho, co zařízení nabízí. Nepoužívají se však k tomu, aby návštěvníkům webových stránek pomáhaly, vzdělávaly je anebo je bavily. (1 bod)

- C) Na domovské stránce je použito video, které vysvětluje více o zařízení a službách, které nabízí. Může být použito video s hodnocením ostatních zákazníků. Použity videa a/nebo zvukové záznamy, které pomáhají, vzdělávají nebo baví návštěvníky webových stránek. (2 body)

10. Obsahují webové stránky hodnocení předchozích zákazníků?

- A) Neobsahují žádné hodnocení předchozích zákazníků. Neexistuje žádný důkaz, že služby mají na někoho pozitivní dopad. (0 bodů)
- B) Webové stránky obsahují hodnocení od zákazníků, ale nejsou zde žádné podrobné informace o tom proč zákazník hodnocení vytvořil. (1 bod)
- C) Obsahují autentické hodnocení od předchozích zákazníků. Jsou pro potenciální zákazníky srozumitelné a pomáhají lidem představit si, že by mohli prožít stejné zkušenosti. (2 body)

11. Byl použit software typu „drag and drop“, který může zpomalit webové stránky?

- A) Webové stránky jsou kompletně vytvořeny pomocí softwaru „drag and drop“. To může způsobovat, že se webové stránky budou načítat pomaleji a budou hůře dostupné pro vyhledávače. (0 bodů)
- B) Je použit systém pro správu obsahu, jako je např. WordPress, ale k provádění změn se používáte další software. Tyto softwary vytváří spoustu zbytečného kódu a mají tendenci ovlivňovat rychlost načítání stránek. (1 bod)
- C) Webové stránky nepoužívají žádný další software. S výjimkou témat a pluginů. Webové stránky jsou spravovány pomocí standardních postupů pro tvorbu webových stránek, což zajišťuje jejich bezproblémový chod. (2 body)

12. Dokáže si zákazník vytvořit a upravovat rezervaci?

- A) Webové stránky nenabízí možnost rezervace, je potřeba kontaktovat ubytovací zařízení. (0 bodů)
- B) Rezervace lze vytvořit pomocí webové stránky, nelze jí ale upravovat. (1 bod)

- C) Rezervaci lze kompletně spravovat za pomoci webových stránek. (2 body)

13. Vypadají webové stránky nepřehledně?

- A) Na domovské stránce je mnoho možností. Je obtížné vidět důležité prvky, protože se toho děje příliš mnoho. (0 bodů)
- B) Stránky je jeví přehledným dojmem, ale budí dojem, že by se dalo něco zlepšit. (1 bod)
- C) Na stránkách není žádný nepořádek, který by vadil. Jsou k dispozici pouze klíčové možnosti. Méně možností je pro návštěvníky webových stránek často přitažlivé. (2 body)

14. Obsahují webové stránky informace o cenách?

- A) Webové stránky neobsahují žádné ceny. Zákazník by mohl mít pocit, že před ním zadržujete informace. (0 bodů)
- B) Stránky obsahují základní informace o cenách, ale ne s příliš detaily. (1 bod)
- C) Obsahují plně transparentní informace o cenách a jejich detailech jako např. jejich účtování. (2 body)

15. Obsahují webové stránky dodatečné informace k ubytování?

- A) Webové stránky neobsahují žádné dodatečné informace k ubytování. (0 bodů)
- B) Stránky obsahují základní informace. (1 bod)
- C) Zákazníci najdou informace, o který by ani nevěděli, že by je mohli potřebovat. (2 body)

[5]

Počet získaných bodů	Slovní hodnocení dosažených bodů
26-30	Vážně hodně dobré stránky.
21-25	Dobře vytvořené stránky.
16-20	Docela dobré, ale dá se něco zlepšit.
11-15	Není to dobré, je zde spousta práce.
6-10	Špatné, zpět k rýsovacímu prknu.
0-5	Velmi špatné, ztráta zákazníků.

Tabulka 5 – Slovní hodnocení testu. Zdroj: [5]

Počet bodů z toho testu je přepočítáván na jednotnou stupnici 0 až 100.

UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Akademický rok: 2020/2021

Studijní program: Aplikovaná informatika
Forma studia: Prezenční
Obor/kombinace: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení: Dominik Kohl
Osobní číslo: I1800182
Adresa: Stará alej 538, Svoboda nad Úpou, 54224 Svoboda nad Úpou, Česká republika
Téma práce: Webová aplikace pro ubytovací zařízení
Téma práce anglicky: Web application for accomodation
Vedoucí práce: Mgr. Daniela Ponce, Ph.D.
Katedra informačních technologií

Zásady pro vypracování:

Cíl:
Návrh a implementace webové aplikace pro ubytovací zařízení dle požadavků zadavatele.
Osnova:
1. Úvod
2. Cíl práce, metodika
3. Porovnání konkurenčních aplikací
4. Analýza aplikace
5. Návrh aplikace
6. Technologie a implementace aplikace
7. Zhodnocení aplikace
8. Shrnutí výsledků
9. Závěr a doporučení

Seznam doporučené literatury:

1. SUBRAMANIAN, Vasan. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node. Apress, 2019.
2. HOQUE, Shama. Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node. js. Packt Publishing Ltd, 2020.
3. WILSON, Eddy. MERN Quick Start Guide: Build Web Applications with MongoDB, Express, js, React, and Node. Packt Publishing Ltd, 2018.

Podpis studenta: 

Datum: 13.08.2022

Podpis vedoucího práce:

Datum: