

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická

Návrh a implementace platformy pro prodej přebytků jídla

Stanislav Chaban

Školitel: Ing. Božena Mannová, Ph.D.
Leden 2024

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Chaban** Jméno: **Stanislav** Osobní číslo: **503197**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh a implementace platformy pro prodej přebytků jídla

Název bakalářské práce anglicky:

Design and implementation of a platform for selling surplus food

Pokyny pro vypracování:

- Cílem je vytvořit aplikaci, která bude pomáhat při prodeji přebytků jídla a pomůže tak jídlu neplýtvat. Prvním cílem práce je provedení analýzy existujících řešení, která poskytnou přehled o funkcích, které tyto aplikace obsahují. Druhým cílem práce je na základě provedené analýzy specifikovat požadavky nové aplikace a navrhnout její architekturu.
1. Navrhněte a implementujte aplikaci pro prodej přebytků jídla. Aplikace by měla pomoci omezit plýtvání potravinami a umožnit využít přebytků pro další spotřebu.
 2. Seznamte se s problematikou, s podobnými existujícími aplikacemi i s platnými předpisy pro tento typ aplikace.
 3. Na základě provedené analýzy navrhněte základní funkcionality navrhované aplikace.
 4. Zvolte architekturu aplikace a vyberte nejvhodnější technologie pro implementaci. Výběr technologií zdůvodněte.
 5. Aplikaci implementujte a otestujte včetně uživatelských testů.
 6. Zhodnoťte výsledky a navrhněte případné další funkcionality nebo jiná zlepšení.
 7. Při řešení využívejte vhodných prostředků softwarového inženýrství.

Seznam doporučené literatury:

- [1] Roger S. Pressmann Bruce Maxim: Software Engineering: A Practitioner's Approach , ISBN-10: 9780078022128
- [2] Aplikace , která pomáhá neplýtvat jídlu <https://nesnezeno.eco/>
- [3] Tesco v Česku testuje aplikaci proti plýtvání jídlu, Lidl zavádí výprodejové bedýnky [https://www.kupi.cz/magazin/clanek/14923-tesco-v-cesku-testuje-aplikaci-proti-plytvani-jidlem-lidl-zavadi-vyprodejove-be-dynk](https://www.kupi.cz/magazin/clanek/14923-tesco-v-cesku-testuje-aplikaci-proti-plytvani-jidlem-lidl-zavadi-vyprodejove-bedynk)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Božena Mannová, Ph.D. kabinet výuky informatiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **22.08.2023**

Termín odevzdání bakalářské práce: **05.02.2024**

Platnost zadání bakalářské práce: **16.02.2025**

Ing. Božena Mannová, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych poděkoval svým přátelům, kteří mě při psaní této práce morálně podporovali.

Prohlášení

I declare that this work is all my own work and I have cited all sources I have used in the bibliography.

Prague, ledna 11, 2024

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 11. January 2024

Abstrakt

Tato bakalářská práce se věnuje inovacím v oblasti potravinového odpadu a klade důraz na vývoj aplikace, která podporuje efektivní a udržitelné řešení této problematiky. Práce identifikuje klíčové faktory vedoucí k vzniku potravinového odpadu a provádí analýzu moderních trendů v potravinářském odvětví. Zaměřuje se na udržitelnost a využití technologií jako prostředek k minimalizaci plýtvání potravinami.

Metodologie výzkumu zahrnuje literární rešerši, analýzu statistických dat a návrh a implementaci aplikace v jazycích Java pro backend a Javascript pro frontend. Vytvoření aplikace je klíčovou součástí práce, která se zaměřuje na aktivní zapojení uživatelů do boje proti potravinovému odpadu. Aplikace bude navržena s ohledem na uživatelskou přívětivost, interaktivitu a možnost efektivního propojení mezi podniky a spotřebiteli.

Výsledky práce očekáváme ve formě funkční aplikace, která umožní uživatelům snadné a interaktivní zúčastnění se snižování potravinového odpadu. Závěry práce budou zahrnovat hodnocení úspěšnosti aplikace a diskuzi o jejím potenciálu v oblasti udržitelnosti a odpovědné spotřeby potravin. Klíčová slova této práce zahrnují potravinový odpad, trendy, inovace, udržitelnost, a vývoj aplikace v jazycích Java a Javascript.

Klíčová slova: Potravinový odpad, Trendy, Inovace, Udržitelnost, Vývoj aplikací, Java, Javascript

Školitel: Ing. Božena Mannová, Ph.D.
Praha 2, Karlovo náměstí 13, E-430

Abstract

This bachelor's thesis focuses on innovations in the field of food waste and emphasizes the development of an application that supports an efficient and sustainable solution to this issue. The thesis identifies key factors contributing to the generation of food waste and analyzes modern trends in the food industry, with a focus on sustainability and the use of technology as a means to minimize food wastage.

The research methodology includes literature review, analysis of statistical data, and the design and implementation of an application using Java for the backend and Javascript for the frontend. The creation of the application is a crucial part of the thesis, aiming to actively engage users in the fight against food waste. The application will be designed with user-friendliness, interactivity, and effective connectivity between businesses and consumers in mind.

The expected results of the thesis include a functional application that allows users easy and interactive participation in reducing food waste. The conclusions will involve an assessment of the application's success and a discussion of its potential in the realm of sustainability and responsible food consumption. Keywords for this thesis include food waste, trends, innovations, sustainability, and application development in Java and Javascript.

Keywords: Food waste, Trends, Innovations, Sustainability, Application development, Java, Javascript

Title translation: Design and implementation of a platform for selling surplus food

Obsah

1 Úvod	1		
1.1 Motivace	1		
1.2 Cíle	1		
2 Úvod do problematiky potravinového odpadu	3		
2.1 Potravinový odpad v podnicích	3		
2.2 Dopady potravinového odpadu	3		
2.3 Současné snahy o řešení	4		
2.4 Závěr	4		
3 Analýza trhu	7		
3.1 Úvod	7		
3.2 Dostupné aplikace	7		
3.2.1 OLIO	7		
3.2.2 Nesněženo	8		
3.2.3 Too good to go	9		
3.2.4 ResQ Club	10		
3.2.5 Food for all	10		
3.2.6 Karma	12		
3.3 Závěr	13		
4 Softwarová analýza	15		
4.1 Uživatelské role	15		
4.1.1 Administrátor	15		
4.1.2 Zákazník	16		
4.1.3 Prodejce	16		
4.1.4 Nezaregistrovaný Uživatel	16		
4.2 Funkční požadavky	16		
4.3 Nefunkční požadavky	18		
4.4 Diagram případů užití	18		
5 Analýza technologií	23		
5.1 Klientská část	23		
5.1.1 React	23		
5.1.2 Angular	23		
5.1.3 Vue.js	24		
5.1.4 Ember.js	24		
5.1.5 Backbone.js	24		
5.1.6 Závěr	24		
5.2 Serverová část	24		
5.2.1 Java	24		
5.2.2 Python	25		
5.2.3 C#	25		
5.2.4 Závěr	25		
5.3 Databaze	25		
5.3.1 MySQL	25		
5.3.2 PostgreSQL	26		
5.3.3 Microsoft SQL Server	26		
5.3.4 Oracle Database	26		
5.3.5 MongoDB	26		
5.3.6 Závěr	26		
5.4 Souhrn	26		
6 Návrh	29		
6.1 Diagram tříd	29		
6.2 Návrh architektury	31		
6.3 Návrh uživatelského rozhraní	32		
6.4 Databaze	34		
6.5 Hibernate	34		
6.6 JPA	34		
6.7 Maven	35		
6.8 REST	35		
7 Implementace	37		
7.1 Serverová část	37		
7.2 Uživatelské rozhraní	37		
7.3 Popis aplikace	38		
7.3.1 Přihlášení	38		
7.3.2 Registrace	39		
7.3.3 Nepřihlášené uživatele	40		
7.3.4 Přihlášený uživatel	43		
7.3.5 Prodejce	44		
7.3.6 Administrator	46		
7.4 Zabezpečení	47		
7.5 Závěr	49		
8 Testování	51		
8.1 Testování s Postmanem	51		
8.2 Uživatelské testování	51		
8.2.1 Výběr testerů	52		
8.2.2 Testovací Scénáře	52		
8.2.3 Závěr uživatelského testování	53		
8.3 Automatizované testy jednotek	53		
8.4 Výsledky a Závěry	53		
8.5 Budoucí Rozvoj a Možné Vylepšení	54		
9 Závěr	55		
Literatura	57		

Obrázky

3.1 OLIO[5]	8	7.14 Uživatelské rozhraní pro zobrazení objednávek	46
3.2 Nesněženo [2]	9	7.15 Uživatelské rozhraní pro zobrazení detailní informace o objednávce ..	46
3.3 Too good to go [?]	10	7.16 Uživatelské rozhraní pro zobrazení seznamu požadavků o převedení profilu na prodejce	47
3.4 ResQ Club [4]	11		
3.5 Food for all [6]	12		
3.6 Karma	13		
4.1 Katalog Aktérů	15		
4.2 Sprava účtu.	19		
4.3 Hlavní interakce v systému.	20		
4.4 Hlavní interakce administrátora v systému.	21		
6.1 Diagram tříd.	30		
6.2 3 vrstevná architektura	31		
6.3 Prototyp stránky pro přihlášení	32		
6.4 Prototyp stránky pro registraci	32		
6.5 Prototyp stránky pro zobrazení všech produktů	33		
6.6 Prototyp stránky pro zobrazení detailní informace produktu	33		
6.7 Prototyp košíku	33		
7.1 Uživatelské rozhraní pro přihlášení	39		
7.2 Uživatelské rozhraní pro Přihlášení pomocí Google	39		
7.3 Uživatelské rozhraní pro registraci	40		
7.4 Uživatelské rozhraní hlavní stránku se seznamem produktů	40		
7.5 Komponenta filtr	41		
7.6 Uživatelské rozhraní pro hlavní stránku s využitím pole pro hledání	42		
7.7 Uživatelské rozhraní pro detailní informace o produktu	42		
7.8 Košík	43		
7.9 Uživatelské rozhraní pro zaplacení objednavky	43		
7.10 Uživatelské rozhraní pro zobrazení profilu uživatele	44		
7.11 Uživatelské rozhraní pro zobrazení profilu prodejce	44		
7.12 Uživatelské rozhraní pro přidání nového podniku část 1	45		
7.13 Uživatelské rozhraní pro přidání nového podniku část 2	45		

Tabulky



Kapitola 1

Úvod

Ve své bakalářské práci se zaměřuji na návrh a vývoj informačního systému určeného k minimalizaci plýtvání potravin v restauracích, rychlých občerstveních a mezi běžnými lidmi. Cílem tohoto systému je zjednodušit a zefektivnit procesy spojené s redistribucí nevyužitého nebo neprodaného jídla. Systém by měl umožňovat snadnější propojení mezi těmi, kteří mají přebytky jídla, a těmi, kteří je chtějí využít, čímž přispívá k řešení problému plýtvání potravin. Rozhodl jsem se proto vytvořit informační systém, který by podpořil efektivnější využití potravinových zdrojů a napomohl k ekologičtějším chování v oblasti stravování.

1.1 Motivace

V dnešní době se neustále zvyšuje povědomí o problematice plýtvání potravinami. Restaurace, obchody i jednotlivci často čelí situacím, kdy se potraviny, které zůstaly neprodané nebo nevyužité, musí zbytečně vyhazovat. Podle odhadů se celosvětově ročně vyhodí přibližně třetina vyrobených potravin [1]. Tento problém má značný dopad jak na životní prostředí, tak na ekonomiku. Řešením může být efektivní využití neprodaných potravin prostřednictvím aplikace pro jejich prodej či darování.

Díky moderním technologiím a rostoucímu zájmu o udržitelný životní styl je nyní ideální doba pro vývoj aplikace, která by propojovala restaurace, obchody a jednotlivce s potenciálními zákazníky nebo příjemci těchto potravin. Mnoho restaurací a obchodů již hledá způsoby, jak snížit množství plýtvání potravinami, ale často jim chybí efektivní prostředek k dosažení tohoto cíle.

Proto se v této bakalářské práci zaměřuji na vývoj prototypu aplikace, která umožní snadnou distribuci nevyužitých potravin.

1.2 Cíle

Cílem bakalářské práce je analyzovat současný trh a existující řešení v oblasti snižování plýtvání potravin, navrhnout strukturu a implementovat informační systém, který umožní prodej či darování nevyužitých potravin od restaurací, obchodů i soukromých osob.

Aplikace by měla umožnit jednoduchou interakci mezi poskytovateli a zákazníky nebo příjemci potravin, efektivně spravovat nabídky nevyužitých potravin a usnadnit proces jejich prodeje nebo darování. Tímto způsobem aplikace přispěje ke snížení plýtvání potravinami a podpoří udržitelné nakládání s potravinovými zdroji.

Výsledkem bakalářské práce bude funkční prototyp aplikace s implementovanou serverovou a klientskou částí, která bude sloužit jako základ pro další rozvoj a zdokonalení.

Kapitola 2

Úvod do problematiky potravinového odpadu

V úvodní kapitole této bakalářské práce se zaměříme na aktuální a naléhavou problematiku odpadu potravin. Potravinové plýtvání se stalo závažným globálním problémem s negativními dopady na životní prostředí, ekonomiku a sociální rovnováhu. Budeme zkoumat rozsah této výzvy, přičemž se zaměříme na faktory a důvody, které k potravinovému plýtvání přispívají. Současně se podíváme na snahy a inovace, které se objevují v oblasti záchrany potravin a minimalizace odpadu. Tato kapitola nám poskytne základní pohled na kontext této problematiky a připraví nás na hlubší zkoumání konkrétních aplikací a jejich přínosů v boji proti potravinovému plýtvání.

2.1 Potravinový odpad v podnicích

Problém potravinového odpadu v podnicích a restauracích může být vysvětlen několika klíčovými faktory. Prvním z nich je nadbytek produkce potravin v těchto zařízeních, kdy se často připravují či zakupují suroviny ve větších množstvích, než kolik je reálně zapotřebí. Estetický standard také hraje důležitou roli, neboť potraviny, které nesplňují přesné vizuální normy, jsou často vyřazovány, ačkoli jsou plně konzumovatelné.

Problém se prohlubuje i kvůli nedostatečné efektivitě při správě skladování a distribuce potravin v restauracích a podnicích. Krátká doba trvanlivosti některých potravin a neexistující systémy sledování stavu způsobují, že mnohé suroviny končí jako odpad. Nedostatek informací a vzdělávání ohledně potravinového plýtvání v průmyslovém odvětví může také hrát svou roli.

Vzhledem k těmto aspektům se zaměříme na podniky a restaurace jako klíčové aktéry, kteří mohou přispívat k významnému množství potravinového odpadu, a snažíme se identifikovat možné přístupy a inovace, které by mohly pomoci tuto problematiku řešit.

2.2 Dopady potravinového odpadu

Problematika potravinového odpadu má značné a rozsáhlé dopady na ekonomiku, životní prostředí a sociální sféru. Z ekonomického hlediska představuje

které se snaží tento problém řešit. Snahy o optimalizaci potravinového řetězce, využití moderních technologií, a důraz na udržitelnost a odpovědnost podnikání jsou klíčovými prvky v boji proti plýtvání potravinami.

Podniky a restaurace přistupují k revidování svých postupů skladování a distribuce, s cílem omezit ztráty potravin a zlepšit udržitelnost. Technologické inovace, zejména v oblasti chytrých senzorů a analýzy dat, nabízejí možnosti pro lepší řízení poptávky a nabídky potravin a minimalizaci znehodnocení.

Aplikace zaměřené na záchranu potravin představují další úroveň účinného a uživatelsky přívětivého řešení. Tyto mobilní platformy umožňují uživatelům aktivní účast na snižování potravinového odpadu tím, že propojují podniky a spotřebitele s cílem efektivně využít nepoužité potraviny.

Celkově je třeba zdůraznit, že boj proti potravinovému odpadu vyžaduje komplexní a kolektivní úsilí od všech aktérů potravinového řetězce a spotřebitelů. Trendy a inovace, které jsme zkoumali, představují slibný směr pro budoucí snahy o vytvoření udržitelnějšího a odpovědnějšího potravinového systému.

Kapitola 3

Analýza trhu

3.1 Úvod

Vstupujeme nyní do kapitoly, která se zaměří na analýzu a hodnocení současných aplikací v oblasti záchrany jídla s cílem identifikovat klíčové trendy, přístupy a úspěchy v této inovativní sféře. Tato analýza poskytne podrobný pohled na aplikace, které se snaží minimalizovat potravinové plýtvání a napomáhat udržitelnějšímu využívání potravin.

Během této kapitoly budou zkoumány různé aspekty těchto aplikací, včetně technologických inovací, sociálních a komunitních aspektů, ekologické udržitelnosti a způsobu, jakým tyto platformy zasahují do potravinového řetězce. Analýza bude směřovat k identifikaci klíčových prvků, které přispívají k efektivitě a úspěchu těchto aplikací v boji proti potravinovému plýtvání.

Cílem této kapitoly je poskytnout komplexní pohled na stávající situaci v oblasti záchrany jídla prostřednictvím moderních technologických nástrojů a podnítit diskuzi o možnostech dalšího rozvoje a zdokonalení těchto aplikací. Důkladná analýza nám umožní lépe porozumět jednotlivým přístupům a přínosům jednotlivých platform, což může posloužit jako inspirace pro budoucí inovace v této důležité oblasti potravinové udržitelnosti.

3.2 Dostupné aplikace

V této části se zaměříme na několik aplikací, které nabízejí platformu pro záchranu potravin, ale momentálně nefungují v České republice, a proto se můžu těmito aplikacemi inspirovat při tvorbě vlastního řešení.

3.2.1 OLIO

Aplikace OLIO přináší unikátní přístup k boji s potravinovým plýtváním prostřednictvím sdílení jídla v komunitě. Tato inovativní platforma umožňuje jednotlivcům, obchodům a domácnostem nabízet přebytečná jídla a potraviny svým sousedům, čímž vytváří lokální síť vzájemné spolupráce.

Jedním z klíčových rozdílů OLIO [5] je důraz na propojení místních komunit a podpora lokálního sdílení potravin. Uživatelé mohou snadno sdílet nevyužité



Obrázek 3.1: OLIO[5]

potravin s lidmi ve svém okolí, což vytváří silné sociální pouto a posiluje komunitní duch. Tato decentralizovaná a komunitně orientovaná filozofie dělá z OLIO nejen nástroj pro boj proti plýtvání potravinami, ale také pro budování silnějších vztahů v rámci místních komunit .

Dalším významným prvkem OLIO je jeho flexibilita v nabídce různých typů potravin, včetně čerstvého ovoce a zeleniny, konzervovaných potravin a domácích pokrmů. Tímto způsobem aplikace pokrývá široké spektrum potravinových kategorií a nabízí uživatelům možnost sdílet a získávat různorodé produkty.

OLIO také zdůrazňuje svůj ekologický rozměr tím, že pomáhá snižovat uhlíkovou stopu spojenou s potravinovým plýtváním. Sdílením potravin na lokální úrovni se minimalizuje potřeba dlouhých přepravních tras a snižuje se množství potravinového odpadu na skládkách.

Celkově lze říci, že aplikace OLIO se vyznačuje svým lokálním a komunitním přístupem k boji s potravinovým plýtváním, nabízí širokou škálu potravin a klade důraz na ekologickou udržitelnost, čímž poskytuje uživatelům holistický a sociálně odpovědný způsob řešení tohoto globálního problému.

■ 3.2.2 Nesnězeno

Nesnězeno[2], inovativní česká aplikace založená v roce 2022, přináší revoluční přístup k potravinovému plýtvání. Jejím hlavním cílem není pouze zastavit zbytečné plýtvání jídlem, ale také snižovat ekologické emise spojené s produkcí a zpracováním potravin. Tato platforma umožňuje podnikům prodat lahůdky z daného dne nebo pochoutky s blížícím se datem expirace, což nejenže šetří potraviny, ale zároveň poskytuje uživatelům příležitost podílet se na udržitelných a ekonomicky výhodných opatřeních.

Nesnězeno není pouze aplikací, je to komunita, která spojuje lidi s podobnými hodnotami a zájmy ohledně udržitelného stravování. Kromě základní funkcionality nabízí bohatý obsah ve formě článků, které podporují myšlenku

¹<https://olioapp.com/en/>



Obrázek 3.2: Nesnězeno [2]

šetření s jídlem a ekologického zodpovědného chování. Tato komunita aktivně působí na sociálních sítích, kde sdílí inspirativní příběhy, tipy a triky týkající se minimalizace plýtvání.

Výjimečnost aplikace spočívá nejen ve vyspělých funkcích vyhledávání jídla v okolí, ale také v různých filtrech, které umožňují uživatelům personalizovat své hledání. Filtry zahrnují kritéria jako vzdálenost, hodnocení, způsob balení a specifické potřeby zákazníka, například preferenci veganství či vegetariánství. Aplikace si navíc získala důvěru a partnerství s prominentními českými institucemi, mezi něž patří Česká televize a Aktuálně.cz.

3.2.3 Too good to go

Ačkoliv existuje několik aplikací s podobným cílem snižovat potravinové plýtvání, "Too Good To Go" vyniká několika klíčovými prvky, které ji odlišují od ostatních. Prvním významným rozdílem je geografická flexibilita a šíře dosahu aplikace. "Too Good To Go"[3] umožňuje uživatelům snadno lokalizovat přebytečné potraviny ve svém okolí, a to včetně obchodů, restaurací a dalších podniků. Tato široká dostupnost výrazně zvyšuje možnosti uživatelů najít a využít nabídky nevyužitých potravin.

Dále se "Too Good To Go" zaměřuje na interaktivitu a komunikaci mezi obchody a spotřebiteli. Aplikace poskytuje uživatelům možnost sledovat a pochopit proces, jakým potraviny končí jako přebytek. To nejenže zvyšuje povědomí o potravinovém plýtvání, ale také vytváří silnější propojení mezi uživateli a provozovateli obchodů.

Dalším významným prvkem, který "Too Good To Go" vyzdvihuje, je jeho vzdělávací aspekt. Aplikace se snaží informovat uživatele o problematice potravinového plýtvání a o impaktech, které toto plýtvání má na životní prostředí. Tím vytváří nejen prostor pro udržitelný nákup, ale také podporuje změnu chování směrem k odpovědnější spotřebě potravin.

Celkově lze konstatovat, že "Too Good To Go" nejen poskytuje praktický způsob, jak bojovat proti potravinovému plýtvání, ale také zdůrazňuje vzdělávací, interaktivní a geograficky dostupné prvky, které ji činí výjimečnou a efektivní v porovnání s jinými aplikacemi na trhu.

²<https://nesnezeno.eco/>



Obrázek 3.3: Too good to go [?]

3.2.4 ResQ Club

Aplikace ResQ Club se zaměřuje na redukci potravinového plýtvání prostřednictvím nabídky zlevněných jídel z přebytků v gastronomických podnicích a restauracích. Klíčovým prvkem, který odlišuje ResQ Club, je jeho partnerství s restauracemi a obchody, které nabízejí jídlo za sníženou cenu až na poslední chvíli, aby minimalizovaly potravinový odpad.

Jedním z významných rozdílů ResQ Club je jeho zaměření na odbyt přebytečného jídla v reálném čase. Uživatelé mohou prostřednictvím aplikace sledovat aktuální nabídky, což umožňuje okamžitou reakci a nákup jídla za zvýhodněnou cenu. Tato dynamická a okamžitá povaha ResQ Club přináší efektivní a pohotový způsob, jak minimalizovat ztráty potravin.

Dále ResQ Club vytváří ekosystém partnerských podniků, což znamená, že aplikace spolupracuje s různými typy gastronomických zařízení, od kaváren po restaurace. To poskytuje uživatelům širokou škálu možností a nabídek, což zvyšuje atraktivitu aplikace pro různorodé chutě a preference spotřebitelů.

V rámci svého přístupu k udržitelnosti ResQ Club zdůrazňuje snižování uhlíkové stopy spojené s potravinovým plýtváním tím, že podporuje využívání již vyrobených jídel a minimalizuje nutnost výroby nových. Tímto způsobem aplikace napomáhá snižování environmentálního dopadu potravinového průmyslu.

Celkově lze konstatovat, že ResQ Club vyniká v oblasti okamžitého snižování potravinového plýtvání díky svému partnerství s gastronomickými podniky a nabídce jídel v reálném čase, což přináší uživatelům efektivní a praktickou alternativu ke snižování potravinových ztrát. [4]

3.2.5 Food for all

Aplikace Food for All přichází s originálním konceptem řešení potravinového plýtvání tím, že nabízí uživatelům možnost zakoupit nevyužitá jídla

³<https://www.toogoodtogo.com/>

⁴<https://www.resq-club.com/>



Obrázek 3.4: ResQ Club [4]

z restaurací a obchodů za výrazně nižší cenu. Jedním z klíčových aspektů, který odlišuje Food for All, je jeho zaměření na poskytování jídel v daném časovém okně, obvykle na konci obchodního dne, kdy by jídlo jinak mohlo být zahozeno.

Jednou z výrazných výhod Food for All je jeho snaha o maximální využití potravin, které by jinak skončily jako odpad. Tímto způsobem aplikace napomáhá restauracím a obchodům snižovat ztráty potravin, což má pozitivní dopad na celkové potravinové plýtvání. Uživatelé zase mohou využívat atraktivní ceny za kvalitní jídlo a zároveň přispívat k boji proti plýtvání potravinami.

Dalším klíčovým prvkem Food for All je jeho důraz na komunitní aspekt. Aplikace se snaží vytvářet pouto mezi místními podniky a spotřebiteli, kteří mají zájem o udržitelnou a odpovědnou spotřebu. To přináší do aplikace sociální rozměr a podporuje lokální ekonomiku a udržitelnost.

V rámci své filozofie Food for All také zdůrazňuje význam edukace spotřebitelů o problému potravinového plýtvání. Aplikace se snaží informovat uživatele o dopadech jejich rozhodnutí na životní prostředí a podporuje uvědomělý přístup k nakupování a konzumaci potravin.

Celkově lze říci, že Food for All vyniká svým inovativním přístupem k časově omezenému nabízení jídel za snížené ceny, přičemž klade důraz na komunitní spolupráci a edukaci spotřebitelů v boji proti potravinovému plýtvání.[6]

⁵<https://foodforalluk.com/>



Obrázek 3.5: Food for all [6]

■ 3.2.6 Karma

Posledním příkladem aplikace zaměřující se na snižování potravinového plýtvání je Karma. Tato platforma nabízí uživatelům možnost zakoupit nevyužité jídlo z restaurací, kaváren a obchodů za sníženou cenu. Karma vyniká především svou uživatelsky přívětivou platformou a důrazem na interaktivní zážitek spotřebitelů.

Jedním z klíčových prvků, který Karma odlišuje, je systém odměn a loajality pro uživatele. Ti mohou sbírat body za každý nákup a tyto body pak vyměňovat za slevy nebo další výhody. Tento systém motivace podporuje opakované využívání aplikace a vytváří pro uživatele přidanou hodnotu.

Karma také klade velký důraz na transparentnost a informovanost spotřebitelů. Uživatelé mohou prostřednictvím aplikace získat informace o potravinách, které zakupují, včetně popisu, ingrediencí a způsobu přípravy. Tímto způsobem Karma podporuje vzdělávání spotřebitelů o potravinách a zvyšuje povědomí o potravinovém plýtvání.

Karma dále rozšiřuje svůj dosah tím, že spolupracuje s různými podniky po celém světě, což poskytuje uživatelům široké možnosti výběru jídel a zvyšuje globální dopad v boji proti potravinovému plýtvání.

Celkově lze konstatovat, že Karma přináší inovativní přístup k řešení potravinového plýtvání, kombinuje motivaci uživatelů, transparentnost v nabídce



Obrázek 3.6: Karma

jídel a globální dosah, čímž poskytuje komplexní řešení pro udržitelnější a odpovědnější nakupování potravin. [7]

3.3 Závěr

V závěru této kapitoly, která byla věnována analýze aplikací v oblasti záchrany jídla, lze konstatovat, že tyto inovativní platformy přinášejí efektivní a udržitelné řešení pro boj proti potravinovému plýtvání. Každá z aplikací, tj. "Too Good To Go", "OLIO", "ResQ Club", "Food for All" a "Karma", se vyznačuje unikátním přístupem k této problematice, přičemž klade důraz na různé aspekty, jako jsou komunitní spolupráce, okamžité nabídky, využívání technologie, ekologická udržitelnost a edukace spotřebitelů.

"Too Good To Go" se zaměřuje na propojení spotřebitelů a podniků prostřednictvím geolokační technologie a výrazně zdůrazňuje vzdělávání a vědomí ohledně potravinové udržitelnosti. Aplikace "OLIO" staví na lokálním sdílení potravin v komunitě a klade důraz na různorodost nabídky potravin. "ResQ Club" se vyznačuje dynamickým modelem nabídky jídel v reálném čase, čímž minimalizuje potravinový odpad.

"Food for All" přináší koncept časově omezených nabídek jídel za zvýhodněnou cenu a zdůrazňuje komunitní spolupráci a edukaci. Aplikace "Karma" pak kombinuje uživatelskou motivaci formou odměn s transparentností v nabídce potravin, což vytváří zajímavý prostor pro odpovědnější spotřebu.

Celkově lze konstatovat, že tyto aplikace nejen snižují potravinové plýtvání, ale také formují nový přístup k nakupování a konzumaci potravin s ohledem na sociální odpovědnost, udržitelnost a komunitní hodnoty. Jejich diverzita a inovativnost představují pozitivní kroky směrem k efektivnějšímu využívání potravin a tvorbě udržitelnější potravinového systému.

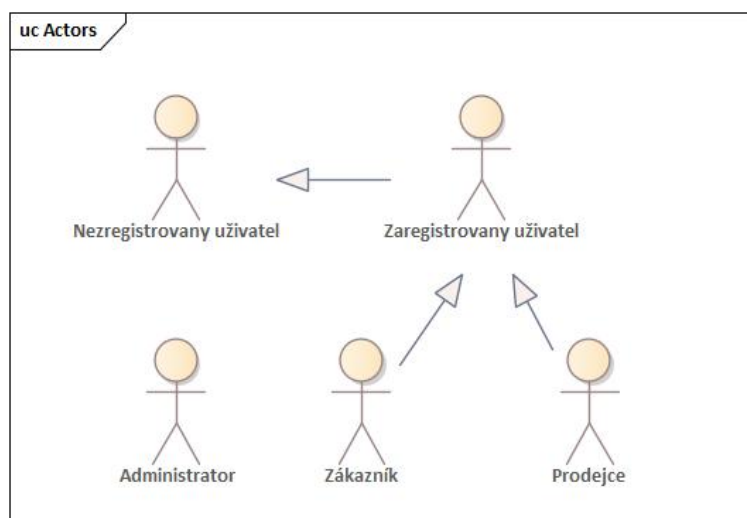
⁵<https://save.karma.life/>

Kapitola 4

Softwarová analýza

Tato kapitola definuje role uživatelů systému, systémové požadavky aplikace a případy užití aplikace

4.1 Uživatelské role



Obrázek 4.1: Katalog Aktérů

4.1.1 Administrátor

Admin má kontrolu nad celým systémem, včetně schvalování žádostí o změnu účtu na Seller, správy uživatelů a možnosti prohlížení a odstraňování jakýchkoli účtů, restaurací, obchodů nebo produktů. Tato role je klíčová pro udržení správného fungování systému, zajištění jeho bezpečnosti a efektivního managementu.

■ 4.1.2 Zákazník

Zákazník je základní uživatel systému, který může prohlížet produkty, restaurace a obchody, nakupovat produkty, přidávat je do oblíbených a sledovat svůj uživatelský profil, včetně historie nákupů a oblíbených produktů. Tato role je zaměřena na koncové uživatele, kteří hledají produkty ke koupi a chtějí přispět k řešení problému plýtvání potravinami.

■ 4.1.3 Prodejce

Prodejce má stejnou funkcionalitu jako Customer, ale navíc má možnost přidávat, upravovat a spravovat své restaurace nebo obchody a produkty v nich. Tato role je určena pro majitele restaurací, obchodů nebo jednotlivce, kteří chtějí nabízet své produkty skrze systém a tím přispět k distribuci nadbytečných potravin.

■ 4.1.4 Nezaregistrovaný Uživatel

Nezaregistrovaný uživatel může v systému nakupovat produkty a prohlížet si detailní informace o produktech. Na rozdíl od zaregistrovaných uživatelů však nemá přístup k funkcím, jako je prohlížení historie nákupů nebo přidávání produktů do seznamu oblíbených a jejich následné odstraňování.

■ 4.2 Funkční požadavky

[9] Funkční požadavky jsou požadavky, které popisují, co by měl software nebo systém dělat. Tyto požadavky se týkají funkčnosti systému a jsou klíčové pro jeho správné fungování. Jsou to specifikace, které popisují, jakým způsobem by měl systém reagovat na různé vstupy, jakým způsobem by měl zpracovávat data a jakým způsobem by měl vykonávat své funkce. Tyto požadavky jsou důležité pro návrh a implementaci systému a jsou nutné pro to, aby byl systém schopen splnit své účely. Funkční požadavky by měly být jasně a konkrétně definovány, aby bylo jasné, jakým způsobem by měl systém fungovat.

Uživatel (Zákazník a Prodejce)

- **FR01: Registrace** Uživatelé se mohou zaregistrovat do systému jako Customer nebo Seller.
- **FR02: Přihlášení do systému** Uživatelé se mohou přihlásit do svého účtu.
- **FR03: Odstranění účtu** Uživatelé mají možnost odstranit svůj účet z systému.
- **FR04: Prohlížení produktů** Uživatelé mohou prohlížet seznam dostupných produktů.
- **FR05: Prohlížení restaurací a obchodů** Uživatelé mohou prohlížet seznam restaurací a obchodů.

- **FR06: Detailní informace o produktu**
Uživatelé mohou zobrazit detailní informace o konkrétním produktu.
- **FR07: Nákup produktu**
Uživatelé mohou zakoupit vybraný produkt.
- **FR08: Přidání produktu do oblíbených**
Uživatelé mohou přidat produkt do seznamu oblíbených.
- **FR09: Prohlížení osobního profilu**
Uživatelé mohou prohlížet své osobní informace v profilu.
- **FR10: Historie nákupů**
Uživatelé mají přístup k historii svých nákupů.
- **FR11: Oblíbené produkty**
Uživatelé mohou prohlížet seznam svých oblíbených produktů.
- **FR12: Žádost o změně statusu na prodejce**
Uživatelé mohou podat žádost o změnu svého účtu na status Seller.

Prodejce

- **FR13: Přidání restaurace/obchodu** Prodejce může přidat novou restauraci nebo obchod.
- **FR14: Úprava restaurace/obchodu**
Prodejce může upravovat informace o svých restauracích/obchodech.
- **FR15: Správa restaurací/obchodů**
Prodejce může prohlížet a spravovat své restaurace/obchody.
- **FR16: Přidání produktů do restaurace/obchodu**
Prodejce může přidávat produkty do svých restaurací/obchodů.
- **FR17: Úprava produktů**
Prodejce může upravovat produkty nabízené v jeho restauracích/obchodech.

Administrátor

- **FR18: Správa žádostí o status Seller** Administrátor může prohlížet a schvalovat či zamítnat žádosti o změnu statusu na Seller.
- **FR19: Správa uživatelů**
Administrátor může prohlížet seznam všech uživatelů a má možnost odstranit jakéhokoli uživatele.
- **FR20: Správa restaurací/obchodů/produktů**
Administrátor může prohlížet a spravovat všechny restaurace, obchody a produkty v systému a má možnost odstranit jakýkoli z těchto prvků.

4.3 Nefunkční požadavky

Nefunkční požadavky jsou klíčové specifikace systému, které popisují jeho vlastnosti a omezení, ale nesouvisí přímo s konkrétními funkcemi, které systém vykonává. Tyto požadavky se týkají kvality a standardů, které systém musí splňovat, aby byl efektivní, spolehlivý a přijatelný pro uživatele.

- **NFR1: Dostupnost** Systém by měl být dostupný 24 hodin denně, 7 dní v týdnu, s minimálními výpadky a s rychlým časem odezvy.
- **NFR2: Bezpečnost**
Systém musí zajistit vysokou úroveň ochrany osobních a transakčních údajů uživatelů před neautorizovaným přístupem a zneužitím.
- **NFR3: Škálovatelnost**
Systém by měl být navržen tak, aby efektivně zvládal rostoucí počet uživatelů a transakcí, aniž by došlo ke zhoršení výkonu.
- **NFRč: Použitelnost**
Systém by měl být navržen s důrazem na jednoduchost a intuitivnost používání, aby byl přístupný a snadno použitelný pro všechny uživatelské role.
- **NFRř: Kompatibilita**
Systém by měl být kompatibilní s různými operačními systémy, prohlížeči a zařízeními, aby byl přístupný co nejširšímu spektru uživatelů.
- **NFRž: Zpětná kompatibilita**
Systém by měl být schopen efektivně pracovat s daty z předchozích verzí a umožnit hladký přechod na novější verze bez ztráty funkcí nebo dat.

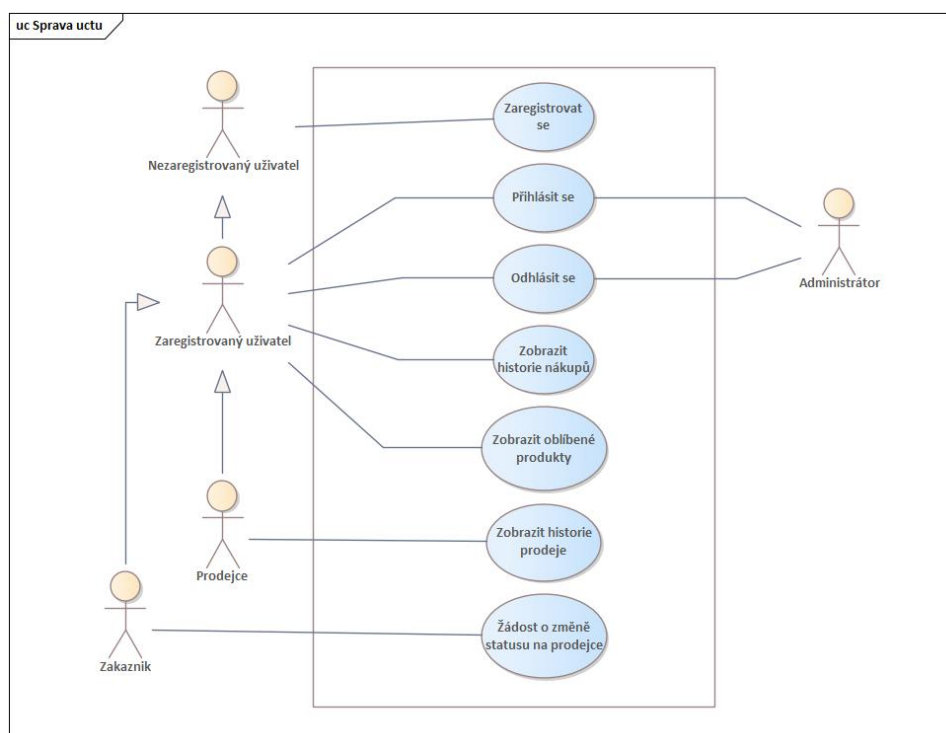
4.4 Diagram případů užití

Případy užití jsou užitečné pro pochopení, jakým způsobem bude systém používán, a pro identifikaci různých typů uživatelů (aktérů) a jejich interakcí se systémem.

Sprava uctu:

- UC 1 - Přihlásit se
- UC 2 - Zaregistrovat se
- UC 3 - Odhlásit se
- UC 4 - Zobrazit historie nákupů
- UC 5 - Zobrazit historie prodeje
- UC 6 - Zobrazit oblíbené produkty

■ UC 7 - Žádost o změnu statusu na prodejce

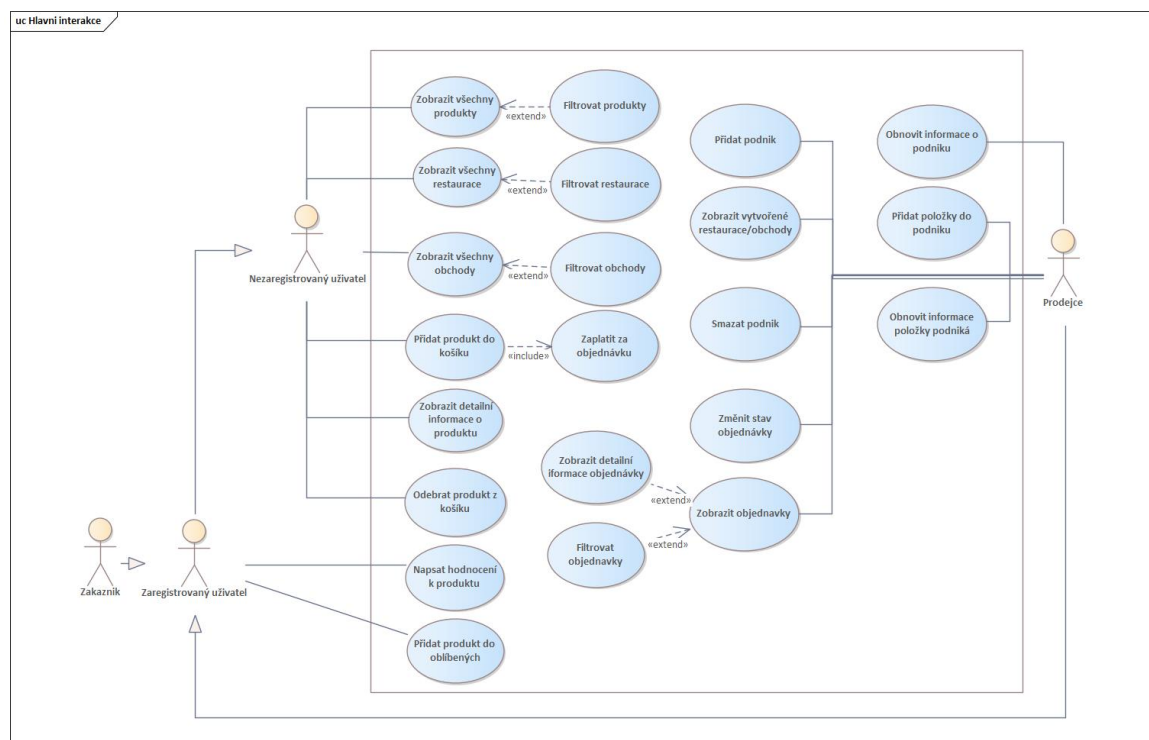


Obrázek 4.2: Sprava účtu.

Hlavní interakce v systému:

- UC 8 - Zobrazit všechny produkty
- UC 9 - Zobrazit všechny restaurace
- UC 10 - Zobrazit všechny obchody
- UC 11 - Zobrazit detailní informace o produktu
- UC 12 - Přidat produkt do košíku
- UC 13 - Odebrat produkt z košíku
- UC 14 - Filtrovat produkty
- UC 15 - Filtrovat restaurace
- UC 16 - Filtrovat obchody
- UC 17 - Zaplatit za objednávku
- UC 18 - Přidat podnik
- UC 19 - Zobrazit vytvořené restaurace/obchody

- UC 20 - Smazat podnik
- UC 21 - Zobrazit objednávky
- UC 22 - Změnit stav objednávky
- UC 23 - Obnovit informace položky podniká
- UC 24 - Přidat položky do podniku
- UC 25 - Obnovit informace o podniku
- UC 25 - Zobrazit detailní onformace objednávky
- UC 25 - Filtrovat objednávky
- UC 25 - Napsat hodnocení k produktu
- UC 25 - Přidat produkt do oblíbených

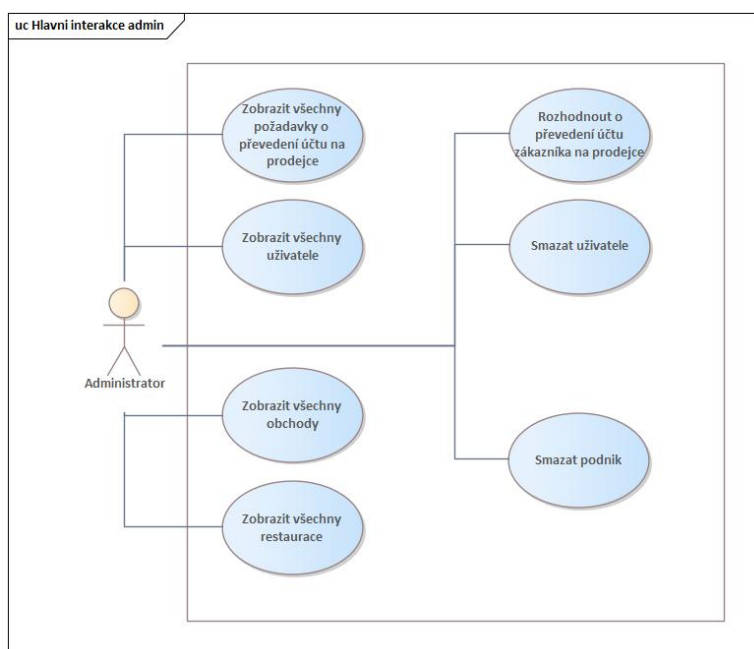


Obrázek 4.3: Hlavní interakce v systému.

Hlavní interakce administrátora v systému:

- UC 26 - Zobrazit všechny požadavky o převedení účtu na prodejce
- UC 27- Zobrazit všechny uživatele
- UC 28 - Zobrazit všechny obchody

- UC 29 - Zobrazit všechny restaurace
- UC 30 - Rozhodnout o převedení účtu zákazníka na prodejce
- UC 31 - Smazat uživatele
- UC 32 - Smazat podnik



Obrázek 4.4: Hlavní interakce administrátora v systému.

Kapitola 5

Analýza technologií

Tato kapitola se věnuje detailnímu popisu a porovnání různých technologií, které by mohly být použity pro realizaci daného systému. Analyzuje se jejich výhody a nevýhody, aby bylo možné co nejlépe zvážit, které z nich jsou pro daný projekt nejvhodnější. Na základě této analýzy bude následně vybrána konkrétní technologie, nebo sada technologií, které budou použity k realizaci systému. Tato kapitola tak poskytuje důležité informace pro rozhodování o tom, jakým způsobem bude systém vytvořen a jakým technologiím bude využívat.

5.1 Klientská část

Klientská část se vztahuje na část webové aplikace, která je viditelná a interaguje s uživatelem. Zahrnuje uživatelské rozhraní, rozložení a designové prvky jako text, obrázky, formuláře, tlačítka a další interaktivní prvky, které jsou vykresleny webovým prohlížečem. Též se zná jako klientská část webové aplikace a je zodpovědná za prezentaci a interakci s aplikací. Vývoj frontendu se typicky týká používání jazyků jako HTML, CSS a JavaScript k vytvoření rozložení a designu webové stránky nebo webové aplikace.

Nejpopulárnější frontendové frameworky pro programování jsou:

5.1.1 React

Vytvořený společností Facebook, React je JavaScript knihovna specializující se na vytváření uživatelských rozhraní. Je známý svou flexibilitou a rychlostí, stejně jako svou podporou pro komponentní architekturu. [10]

5.1.2 Angular

Vytvořený společností Google, Angular je kompletní řešení pro vývoj webových aplikací. Je známý svým silným typováním a podporou pro MVC architekturu. [11]

■ 5.1.3 Vue.js

Vytvořený skupinou nezávislých vývojářů, Vue.js je jednoduchý framework pro vývoj interaktivních webových aplikací. Je známý svou jednoduchostí a snadnou implementací, stejně jako svou flexibilitou.[12]

■ 5.1.4 Ember.js

Ember je JavaScript framework, který vám umožňuje vytvářet ambiciózní webové aplikace. Poskytuje silný soubor funkcí a CLI, který vám pomůže rychle spustit vaši aplikaci. [13]

■ 5.1.5 Backbone.js

Backbone.js je lehký JavaScript framework, který umožňuje strukturovat klientský kód v MVC stylu. Poskytuje malou sadu funkcí, které jsou snadno pochopitelné a rozšiřitelné.[14]

■ 5.1.6 Závěr

V závěru lze říci, že React je nejvhodnější volbou pro vývoj webového projektu pro hotel. Jeho flexibilita, rychlost a podpora pro komponentní architekturu dělají z něj ideální nástroj pro vytváření moderních webových aplikací. Navíc, má silnou podporu a rozsáhlou komunitu vývojářů, což zajišťuje trvalou podporu a aktualizace. Mám malou zkušenost s tímto frameworkem a jsem si jistý, že React bude nejlepší volbou pro naše potřeby a požadavky projektu.

■ 5.2 Serverová část

Serverová část je složená z hardwarové a softwarové části, která je uložena na vzdáleném serveru, nikoli na počítači nebo prohlížeči uživatele. Jejím úkolem je zpracovat data, která jsou mu poskytnuta a následně odpovědět na dotazy nebo požadavky, které k ní přicházejí. Serverová část tak hraje důležitou roli v komunikaci mezi klientem a databází, zajišťuje bezpečnost a dostupnost dat a zpracovává logiku aplikace. Je to důležitá součást každého webového nebo síťového systému, která umožňuje správu a zpracování velkého množství dat.

■ 5.2.1 Java

Java je populární objektově orientovaný programovací jazyk, který se široce používá k vývoji webových aplikací. Je znám svou platformovou nezávislostí, bezpečností a škálovatelností. Také se používá k vývoji podnikových aplikací a mobilních aplikací. Jeho popularity je díky svým rozsáhlým knihovnám a frameworkům jako Spring, Hibernate a JSF.[15]

■ 5.2.2 Python

Python je vysokoúrovňový interpretovaný programovací jazyk, který se proslavil svou jednoduchostí použití, čitelností a flexibilitou. Často se používá pro webový vývoj, vědecké výpočty, analýzu dat a umělou inteligenci. Jeho popularity je díky jeho jednoduchosti a popularity svých webových frameworků jako jsou Django, Flask a Pyramid.[16]

■ 5.2.3 C#

C# je moderní objektově orientovaný programovací jazyk, který se široce používá pro vývoj Windows desktop aplikací a webových aplikací. Je známý svými moderními funkcemi a používá se pro vývoj podnikových aplikací a her. Jeho popularity je díky své podobnosti s Javou a použití v ekosystému Microsoftu a také svých webových frameworků jako ASP.NET Core, NancyFX a ServiceStack.[17]

■ 5.2.4 Závěr

V závěru, po porovnání těchto třech populárních programovacích jazyků, lze říci, že Java pro mě je nejlepší volbou pro psaní webových aplikací. Java nabízí kombinaci bezpečnosti, škálovatelnosti a platformové nezávislosti, což ji činí vhodnou pro vývoj webových aplikací. Navíc má k dispozici velké množství knihoven a frameworků, což ulehčuje vývoj, testování a údržbu webových aplikací. Kromě toho má Java velkou komunitu vývojářů, která poskytuje trvalou podporu a aktualizace, což ji činí spolehlivou volbou pro vývoj webových aplikací.

■ 5.3 Databaze

Databáze je systém, který umožňuje efektivně a organizovaně skladovat a spravovat velké množství dat. Tyto data jsou uložena ve formě tabulek, kde každá tabulka obsahuje řádky a sloupce. Řádky představují jednotlivé záznamy a sloupce představují různé atributy těchto záznamů. Databáze umožňuje snadné vyhledávání, úpravy a mazání dat, a také poskytuje možnosti pro tvorbu dotazů a reportů. Databáze jsou často používány pro ukládání informací v podnikových aplikacích, jako jsou například účetní systémy, CRM nebo ERP systémy.

■ 5.3.1 MySQL

MySQL je open-source relační databázový management systém, který je známý svou spolehlivostí a jednoduchostí použití. Je velmi populární mezi malými a středními podniky a je často používán pro webové aplikace.[18]

■ 5.3.2 PostgreSQL

PostgreSQL je open-source relační databázový management systém, který je známý svými pokročilými funkcemi a škálovatelností. Je to vhodná volba pro aplikace s vysokými požadavky na dostupnost a bezpečnost dat.[19]

■ 5.3.3 Microsoft SQL Server

Microsoft SQL Server je proprietární relační databázový management systém, který je vyvíjen společností Microsoft. Je velmi populární v podnikovém prostředí a nabízí širokou škálu funkcí a nástrojů pro správu dat.[20]

■ 5.3.4 Oracle Database

Oracle Database je proprietární relační databázový management systém, který je vyvíjen společností Oracle. Je to silná volba pro velké organizace s náročnými požadavky na správu dat a bezpečnost.[21]

■ 5.3.5 MongoDB

MongoDB je open-source NoSQL dokument-orientovaný databázový systém, který je navržen pro škálovatelnost a flexibilitu. Je velmi populární pro aplikace, které potřebují rychle pracovat s velkými objemy dat.[22]

■ 5.3.6 Závěr

Na závěr, na základě analýzy různých databází, lze říci, že PostgreSQL je nejlepší volbou pro webový systém pro správu hotelu. PostgreSQL je open-source relační databázový management systém, který se proslavil svými rozsáhlými možnostmi a flexibilitou. Je to ideální volba pro webové aplikace, které požadují vysokou dostupnost a zabezpečení dat, což je důležité pro webovou stránku hotelu. Navíc skutečnost, že mám zkušenosti s PostgreSQL, činí tuto volbu atraktivnější, protože budu moci využít své stávající znalosti a dovednosti k implementaci a správě databáze pro webovou stránku hotelu.

■ 5.4 Souhrn

V procesu výběru technologií pro bakalářskou práci jsem se rozhodl pro kombinaci Java pro serverovou část, React pro klientskou část a PostgreSQL pro databázové řešení. Toto rozhodnutí bylo pečlivě zváženo a vychází z několika klíčových faktorů, které tyto technologie činí ideální volbou pro můj projekt.

Java, s její stabilní a robustní architekturou, poskytuje vysokou míru bezpečnosti, spolehlivosti a škálovatelnosti. Je to osvědčená volba pro vývoj serverových aplikací a má široké uplatnění v průmyslu. Díky platformové nezávislosti a podpoře silných frameworků jako je Spring, Java nabízí pevný základ pro spolehlivý a dlouhodobě udržitelný vývoj aplikací.

React, vytvořený společností Facebook, je významným nástrojem pro vývoj klientské části webových aplikací. Jeho komponentní struktura, efektivní vykreslování a bohatý ekosystém nástrojů umožňují vytvářet moderní, rychlá a responzivní uživatelská rozhraní. Jeho popularita a podpora ze strany vývojářské komunity zajišťují, že React bude i nadále dobře podporován a aktualizován.

Pokud jde o databázové řešení, zvolil jsem PostgreSQL, což je open-source relační databázový systém známý svou pokročilostí, výkonností a škálovatelností. Tato volba byla motivována jak mými osobními zkušenostmi s PostgreSQL, tak i jeho schopností efektivně řešit komplexní datové struktury a dotazy, což je nezbytné pro náš systém zaměřený na správu potravin a transakcí.

V kombinaci tyto technologie - Java, React a PostgreSQL - poskytují silný základ pro vývoj spolehlivého, efektivního a uživatelsky přívětivého systému pro minimalizaci plýtvání potravinami. Má zkušenost s těmito technologiemi mi umožní efektivně pracovat na projektu a přinést kvalitní řešení pro tento důležitý sociální problém.

Kapitola 6

Návrh

6.1 Diagram tříd

Diagram tříd je základním prvkem v modelování objektově orientovaných systémů a slouží k vizualizaci tříd a vztahů mezi nimi v rámci softwarové aplikace.[23]

Třída 'User' představuje hlavní entitu v našem systému, která reprezentuje koncové uživatele aplikace. Tato třída je zásadní pro autentizaci, správu uživatelských profilů a interakci s různými moduly systému.

Hlavní atributy třídy "User":

- `userId` (ID uživatele): Unikátní identifikační číslo uživatele.
- `uniqIdentificationCode` (Unikátní identifikační kód): Unikátní kód pro identifikaci uživatele který se nastaví při registraci pomocí google.
- `provider` (Poskytovatel registrace): Typ registrace uživatele (např. aplikace, Google)
- `email` (E-mail): E-mailová adresa, unikátní pro každého uživatele. Používá se při přihlašování
- `password` (Heslo): Heslo uživatele. Používá se při přihlašování

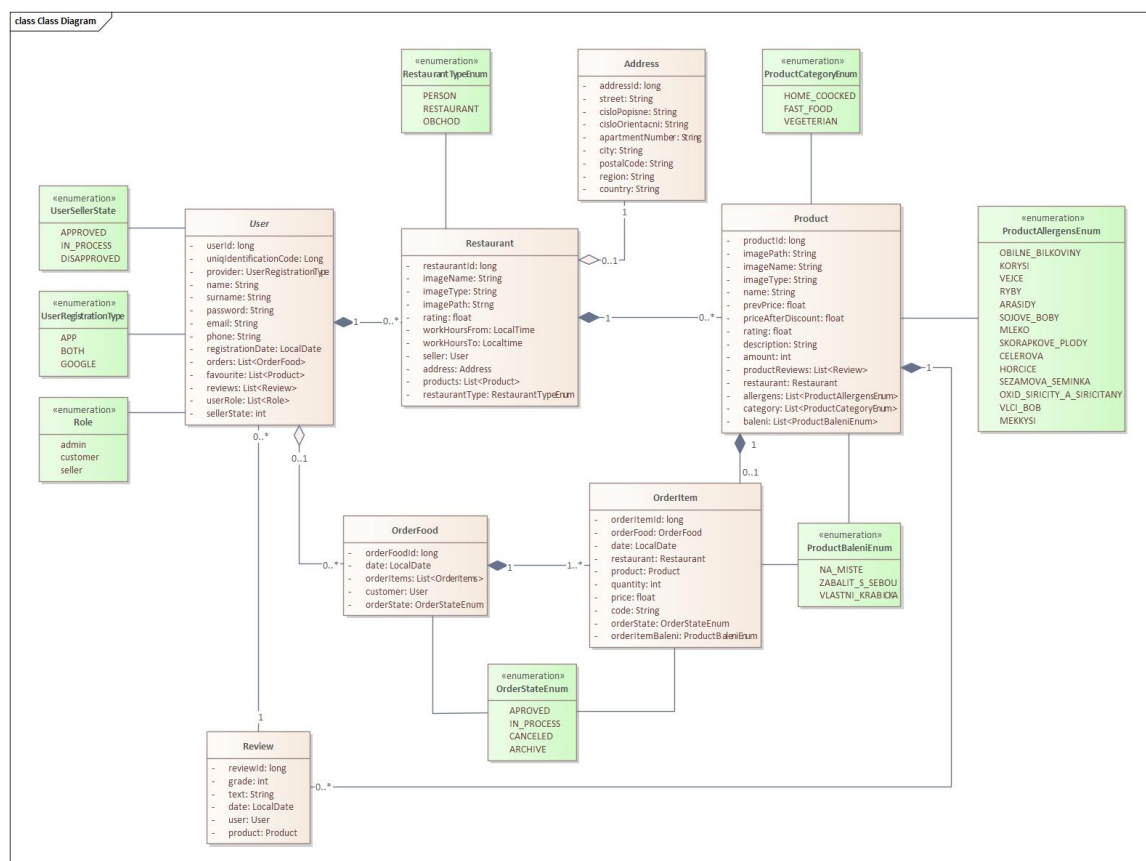
Další atributy:

- `username` (Uživatelské jméno): Uživatelské jméno.
- `name` (Jméno): Jméno uživatele.
- `surname` (Příjmení): Příjmení uživatele.
- `phone` (Telefon): Telefonní číslo uživatele.
- `registrationDate` (Datum registrace): Datum, kdy byl uživatel zaregistrován.

- registrationDate (Datum registrace): Datum, kdy byl uživatel zaregistrován.

Vztahy s dalšími třídami:

- orders (Objednávky): Seznam objednávek uživatele.
- favourite (Oblíbené produkty): Seznam oblíbených produktů uživatele.
- reviews (Recenze): Seznam recenzí napsaných uživatelem.
- userRole (Uživatelské role): Seznam rolí přiřazených uživateli (např. prodejce, uživatel, administrator).
- sellerState (Stav prodejce): Stav uživatele jako prodejce (např. schválený, neschválený, v procesu).
- restaurant (Restaurace): Stav uživatele jako prodejce (např. schválený, neschválený, v procesu).



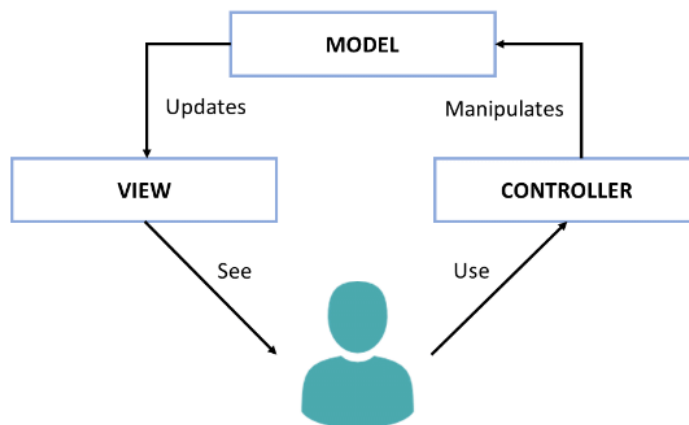
Obrázek 6.1: Diagram tříd.

Tato diagram tříd poskytuje ucelený přehled o struktuře aplikace a je klíčový pro pochopení vztahů a závislostí mezi jednotlivými komponentami systému.

6.2 Návrh architektury

MVC (Model-View-Controller) je široce používaný architektonický vzor v programování, který rozděluje aplikaci do tří hlavních komponent: Model, View a Controller. Tato struktura vám pomáhá organizovat kód způsobem, který odděluje aplikační logiku od uživatelského rozhraní, čímž zlepšuje čitelnost a usnadňuje testování a údržbu.

- Model je zodpovědný za data a obchodní logiku aplikace. Zpracovává požadavky kontroléru a spolupracuje s databází.
- View zobrazuje data uživateli, obvykle prostřednictvím uživatelského rozhraní.
- Controller slouží jako prostředník mezi modelem a pohledem, zpracovává uživatelský vstup a předává modelu příkazy.



Obrázek 6.2: 3 vrstevná architektura

Proč jsem si vybral MVC?

Architekturu MVC jsem si vybral z několika důvodů:


- Jasné oddělení obav: MVC umožňuje oddělit aplikační logiku a uživatelské rozhraní, což výrazně zjednodušuje proces vývoje a údržby. To mi pomáhá organizovat můj kód strukturovanějším a logičtějším způsobem.
- Flexibilita a škálovatelnost: Díky modularitě MVC mohu snadno upravovat nebo rozšiřovat určité části aplikace bez výrazných změn ve zbytku. Díky tomu je architektura ideální pro můj projekt, který se může časem vyvíjet a růst.
- Snadné testování: Jednotlivé komponenty v MVC lze testovat nezávisle, což zajišťuje vyšší spolehlivost a stabilitu aplikace.

- Popularita a podpora komunity: MVC je jednou z nejpůvodnějších architektur a je široce podporována komunitou vývojářů. Díky tomu mám přístup k mnoha zdrojům, nástrojům a osvědčeným postupům.

Společně tyto faktory dělají z MVC nejlepší volbu pro můj projekt, protože je efektivní, rozšiřitelný a snadno se udržuje

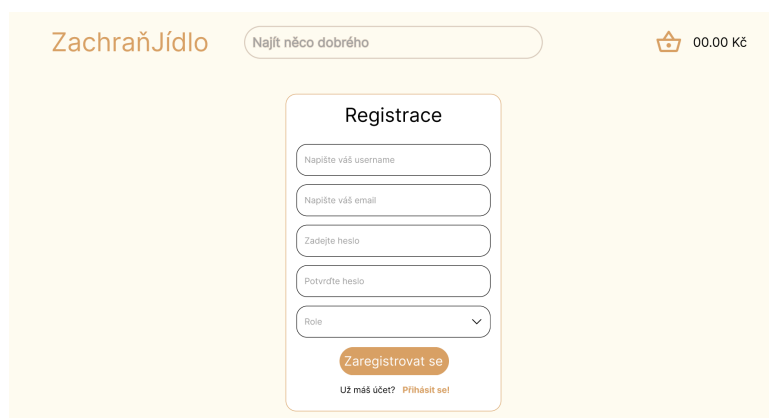
6.3 Návrh uživatelského rozhraní

Vývoj uživatelského rozhraní pro tento projekt zahrnoval několik zásadních fází, aby se zajistila jeho efektivnost a organizovanost. Iniciálně byly pomocí designového nástroje Figma vypracovány detailní návrhy rozhraní.



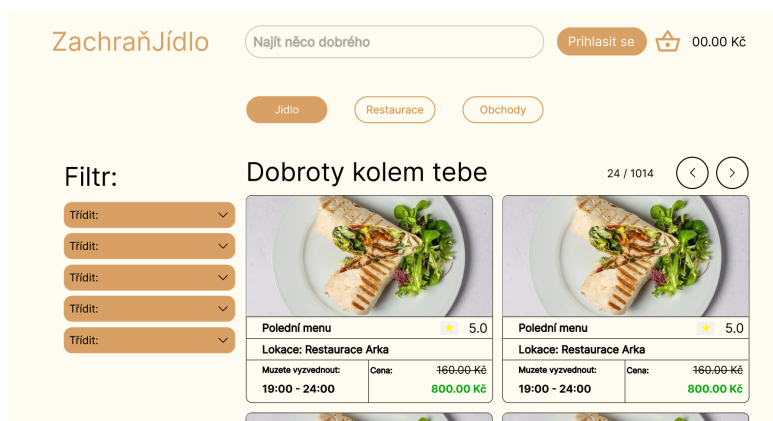
Prototyp stránky pro přihlášení. V horní části je logo 'ZachraňJídlo', vyhledávací pole s textem 'Najít něco dobrého' a ikona košíku s cenou '00.00 Kč'. V centru je formulář s titulem 'Přihlášení'. Obsahuje dvě vstupní pole: 'Napište váš login' a 'Napište heslo'. Pod nimi je oranžový tlačítko 'Přihlásit se'. Na konci formuláře je odkaz: 'Ještě nemáš účet? Zaregistrovat se!'.

Obrázek 6.3: Prototyp stránky pro přihlášení

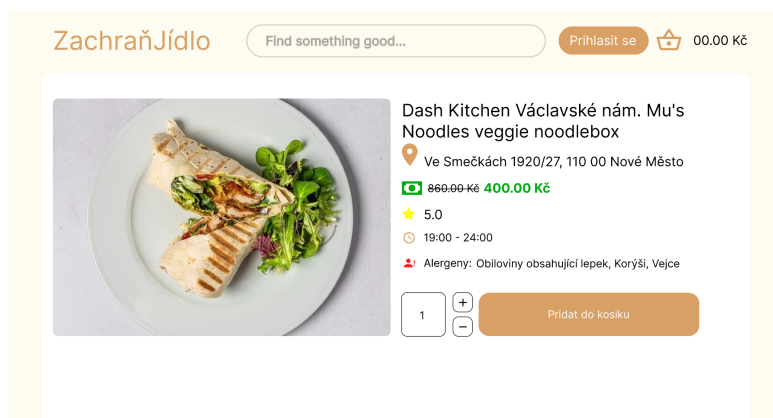


Prototyp stránky pro registraci. V horní části je logo 'ZachraňJídlo', vyhledávací pole s textem 'Najít něco dobrého' a ikona košíku s cenou '00.00 Kč'. V centru je formulář s titulem 'Registrace'. Obsahuje čtyři vstupní pole: 'Napište váš username', 'Napište váš email', 'Zadejte heslo' a 'Potvrďte heslo'. Pod nimi je dropdown menu 'Role' a oranžové tlačítko 'Zaregistrovat se'. Na konci formuláře je odkaz: 'Už máš účet? Přihlásit se!'.

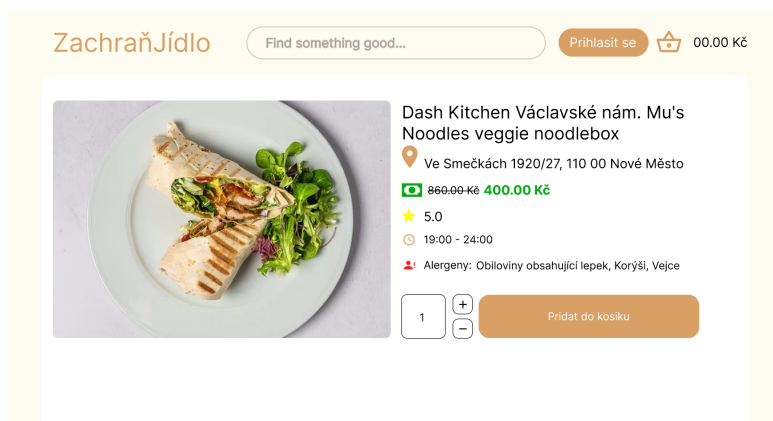
Obrázek 6.4: Prototyp stránky pro registraci



Obrázek 6.5: Prototyp stránky pro zobrazení všech produktů



Obrázek 6.6: Prototyp stránky pro zobrazení detailní informace produktu



Obrázek 6.7: Prototyp košíku

6.4 Database

Během vývoje aplikace rozhodl jsem se používat databázový systém PostgreSQL. Toto rozhodnutí bylo založeno na mých předchozích pozitivních zkušenostech s tímto systémem. Mezi klíčové výhody PostgreSQL patří jeho spolehlivost, vysoká míra flexibility a plná podpora ACID principů. Pro práci s databází jsem využíval nástroj pgAdmin, který poskytuje pohodlné rozhraní pro správu a monitorování databáze.

```
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect
spring.jpa.hibernate.ddl-auto=create-drop
spring.jpa.hibernate.show-sql=true
spring.datasource.url=jdbc:postgresql://localhost:5432/postgres
spring.datasource.username=postgres
spring.datasource.password=admin
```

Listing 6.2: Konfigurace aplikace pro přístup k databázi.

6.5 Hibernate

V rámci mého projektu jsem využil Hibernate, což je objektově relační mapovací (ORM) framework pro jazyk Java. Hibernate mi umožnil efektivně mapovat objekty z Java aplikace do databázových tabulek a naopak, což zjednodušilo práci s databází. Díky Hibernate jsem mohl snadno zpracovávat a manipulovat s daty bez nutnosti psát složité SQL dotazy. Tento nástroj také poskytuje výkonné funkce pro dotazování a transakční správu, což bylo klíčové pro správu a integritu dat v mém projektu.

6.6 JPA

Ve svém projektu jsem se rozhodl používat Java Persistence API (JPA), což je nástroj, který poskytuje standardizovaný způsob práce s relačními databázemi v Java aplikacích. Toto rozhodnutí bylo motivováno potřebou efektivně spravovat databázová data prostřednictvím objektově orientovaného přístupu, což JPA umožňuje. Jeho použití značně zjednodušilo proces mapování objektů aplikace na databázové tabulky a zefektivnilo celkovou práci s daty v projektu.

```
@ManyToMany(cascade = CascadeType.ALL)
@JoinColumn(name = "order_food_id")
private List<OrderFood> orders;
```

Listing 6.2: Příklad použití JPA.

6.7 Maven

Ve svém projektu jsem použil Maven, což je nástroj pro správu a automatizaci sestavení softwarových projektů v prostředí Java. Maven mi umožnil snadněji spravovat závislosti projektu, zjednodušit proces sestavení a zvýšit přehlednost celého vývojového cyklu. Jeho konfigurační soubory založené na XML poskytují efektivní a flexibilní způsob definování a správy projektové struktury, což bylo pro úspěšnou realizaci mého projektu nezbytné.[30]

6.8 REST

V mém projektu jsem použil Spring REST, což je součást Spring frameworku určená pro vývoj RESTful webových služeb. Spring REST mi umožnil snadno vytvářet webové API, které jsou jak efektivní, tak snadno použitelné. Díky jeho podpoře pro anotace a konfiguraci založenou na Javě jsem mohl efektivně definovat koncové body služeb, routování požadavků a zpracování odpovědí. Tento přístup mi pomohl vytvořit čisté a udržovatelné API, které je dobře integrovatelné s ostatními komponentami mého projektu a podporuje komunikaci mezi klientskou a serverovou částí aplikace.

```
@GetMapping
@PreAuthorize("hasAnyRole('ADMIN')")
public ResponseEntity<List<UserDto>> getAllUsers() {
    List<UserDto> users = customerService.findAll();
    if (users.isEmpty()) {
        return ResponseEntity.noContent().build();
    }
    return ResponseEntity.ok(users);
}
```

Listing 6.2: Příklad použití REST

Kapitola 7

Implementace

Pro vývoj webové aplikace jsem se rozhodl použít vícevrstvý architektonický model, který obsahuje vrstvy spolupracující mezi sebou. Konkrétně to zahrnuje prezentační vrstvu (uživatelské rozhraní) realizovanou pomocí frameworku React, vrstvu pro zpracování logiky (business logiku) využívající Java Spring Boot a datovou vrstvu pro ukládání informací, kde je použit PostgreSQL.

7.1 Serverová část

Pro serverovou část projektu je použit programovací jazyk Java 17. Pro automatizaci sestavování projektu je použit nástroj Maven. Soubor `src/main/resources/application.properties` obsahuje nastavení konfigurace aplikace a parametry nezbytné pro připojení backendu k databázi. Navíc, aby byla zajištěna funkčnost webového serveru, REST[25] a Spring Boot [24], byla do souboru `pom.xml` přidána následující závislost:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

7.2 Uživatelské rozhraní

V rámci implementace frontendové části projektu jsem se rozhodl využít React, populární JavaScriptovou knihovnu pro tvorbu uživatelských rozhraní. React poskytuje efektivní způsob, jak spravovat stav aplikace a zajišťuje hladké uživatelské interakce díky svému reaktivnímu a komponentnímu přístupu.

React umožňuje vytvářet opakovaně použitelné UI komponenty, což zjednodušuje vývoj a udržování kódu. Jsem si vybral tento framework pro jeho flexibilitu, širokou podporu komunity a schopnost integrace s různými backendovými technologiemi.

Kromě samotného Reactu jsem také použil řadu dalších nástrojů a knihoven, které pomohly zefektivnit vývoj. Mezi ně patří

```
"dependencies": {
  "@emotion/react": "^11.11.1",
  "@emotion/styled": "^11.11.0",
  "@mui/icons-material": "^5.15.0",
  "@mui/lab": "^5.0.0-alpha.158",
  "@mui/material": "^5.15.0",
  "@react-oauth/google": "^0.12.1",
  "@testing-library/jest-dom": "^5.17.0",
  "@testing-library/react": "^13.4.0",
  "@testing-library/user-event": "^13.5.0",
  "axios": "^1.4.0",
  "jwt-decode": "^3.1.2",
  "react": "^18.2.0",
  "react-dom": "^18.2.0",
  "react-icons": "^4.12.0",
  "react-router-dom": "6.20.1",
  "react-scripts": "5.0.1",
  "web-vitals": "^2.1.4"
}
```

Tyto nástroje umožňují lepší správu stavů, navigaci v aplikaci a integraci s externími API.

Celkově jsem s pomocí Reactu a přidružených technologií byl schopen vytvořit responzivní, uživatelsky přívětivé a funkční frontendové řešení, které efektivně komunikuje s backendovou částí aplikace.

Nyní, pro lepší představu o skutečném uživatelském rozhraní, bych rád přešel k vizuální prezentaci uživatelského rozhraní. Následující část nabízí pohled na klíčové aspekty a funkce aplikace, které ilustrují, jak teorie a technologie ožívají v praxi."

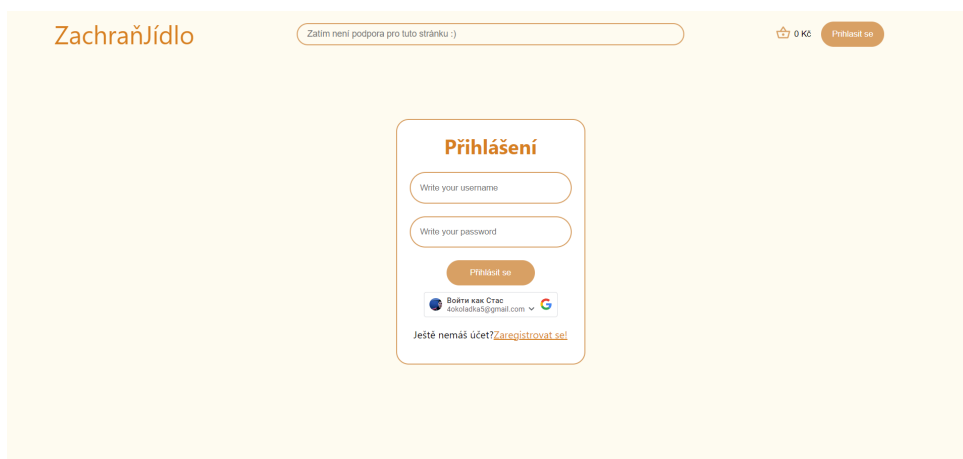
7.3 Popis aplikace

V této části bakalářské práce se zaměřím na podrobný popis aplikace, jejíž vývoj a implementace tvoří hlavní obsah tohoto projektu. Cílem této sekce je poskytnout komplexní přehled o funkcionalitě, struktuře a vlastnostech vyvinutého softwaru.

Aplikace, kterou zde prezentují, byla navržena s cílem řešit konkrétní problém – v tomto případě plýtvání potravinami.

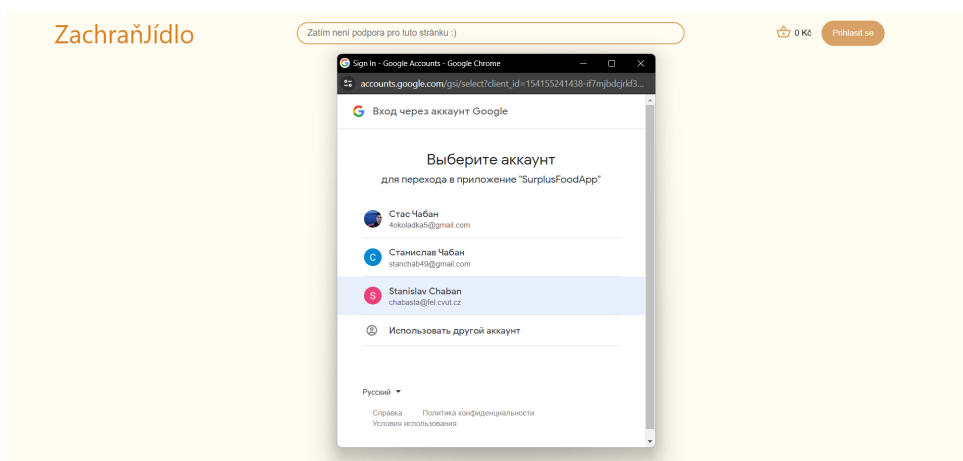
7.3.1 Přihlášení

Na stránce přihlášení uživatel bude moci se přihlásit dvěma způsoby, buď standartním vyplňováním povinných polí, jako je email a heslo



Obrázek 7.1: Uživatelské rozhraní pro přihlášení

Nebo taky je možnost se přihlásit pomocí Google profilu



Obrázek 7.2: Uživatelské rozhraní pro Přihlášení pomocí Google

7.3.2 Registrace

Na stránce pro registraci si uživatel může vytvořit účet a ihned si vybrat, jakou roli by chtěl na webu zastávat.

Obrázek 7.3: Uživatelské rozhraní pro registraci

7.3.3 Nepřihlášené uživatele

Nepřihlášený uživatel může prohlížet seznam všech produktů, restaurace a obchodů. Na každé stránce je dvě možnosti filtrace dat.

Obrázek 7.4: Uživatelské rozhraní hlavní stránky se seznamem produktů

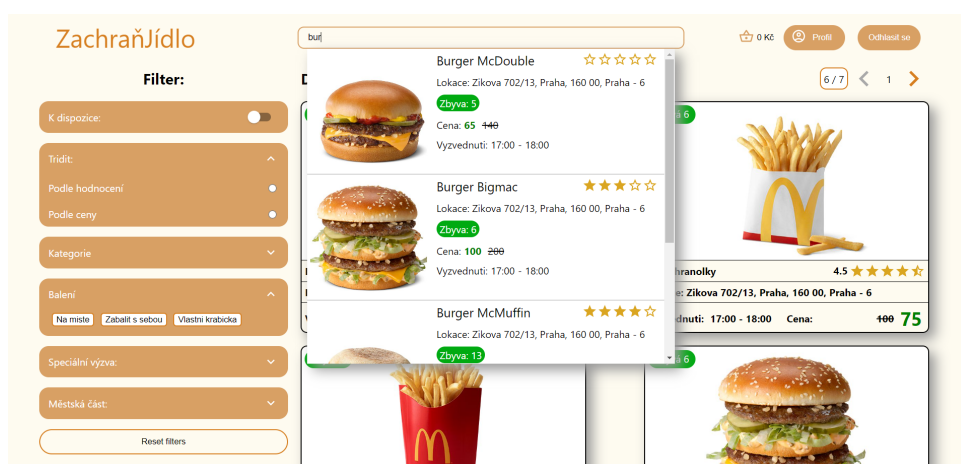
První způsob je použít speciální komponentu v levé části obrazovky, kde si uživatel může například zobrazit produkty které nemají nějaké alergie a td.



Obrázek 7.5: Komponenta filtr

Druhý způsob filtrace dat je = použití vyhledávací lišty která se nachází ve hlavičce webu. Jakmile uživatel začne zadávat nějaký text, objeví se mu všechny položky, které mají tuto část v názvu.

7. Implementace

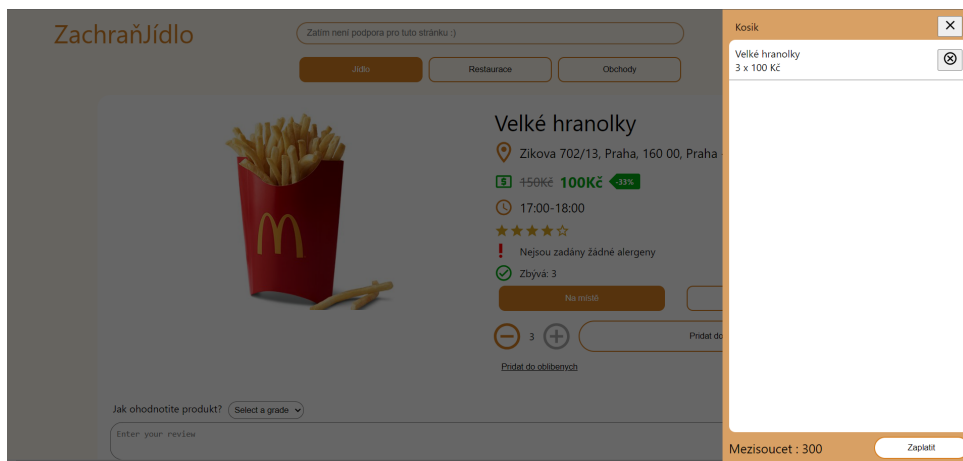


Obrázek 7.6: Uživatelské rozhraní pro hlavní stránku s využitím pole pro hledání

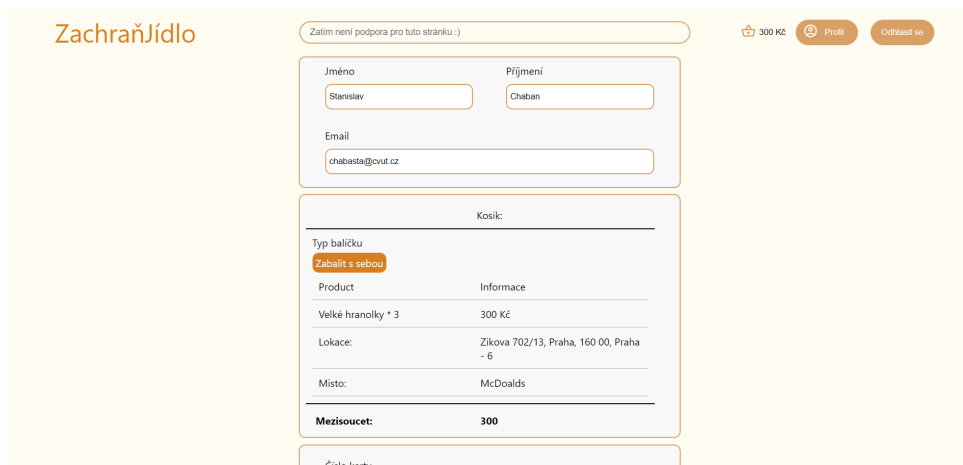
když si uživatel najde něco, co se mu líbí, může si zobrazit podrobné informace o této položce, přidat jej do kosíku a případně zaplatit



Obrázek 7.7: Uživatelské rozhraní pro detailní informace o produktu



Obrázek 7.8: Košík



Obrázek 7.9: Uživatelské rozhraní pro zaplacení objednávky

7.3.4 Přihlášený uživatel

Přihlášený uživatel může dělat to všechno co nepřihlášený ale navíc má k dispozici profil, kde může změnit osobní údaje nebo heslo. Taký může prohlédnout historie nákupů a oblíbené produkty. A vycházející z předchozí věty ještě jeden maličký rodil je že přihlášený uživatel může přidat produkt do seznamu oblíbených. Taký přihlášený uživatel může ve svém profilu požádat o převedení na status prodejce

Obrázek 7.10: Uživatelské rozhraní pro zobrazení profilu uživatele

7.3.5 Prodejce

Přihlášený uživatel s rolí prodejce může vidět ve svém profilu jednu sekce navíc, což je historie prodeje.

Obrázek 7.11: Uživatelské rozhraní pro zobrazení profilu prodejce

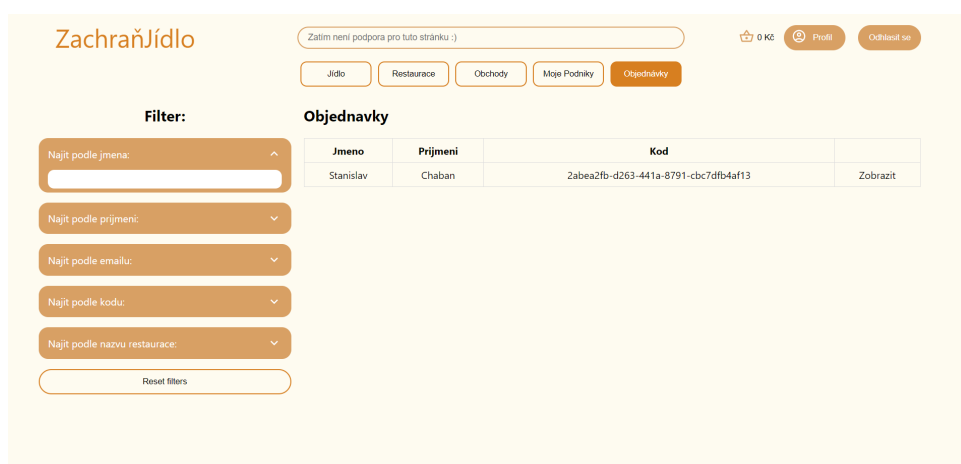
Uživatel přihlášený jako prodejce má k dispozici dvě nové stránky, první je "Moje podniky" kde může vytvořit nový podnik a zároveň přidat položky které chce prodávat. Také všechny svoje podniky může editovat nebo smazat.

Obrázek 7.12: Uživatelské rozhraní pro přidání nového podniku část 1

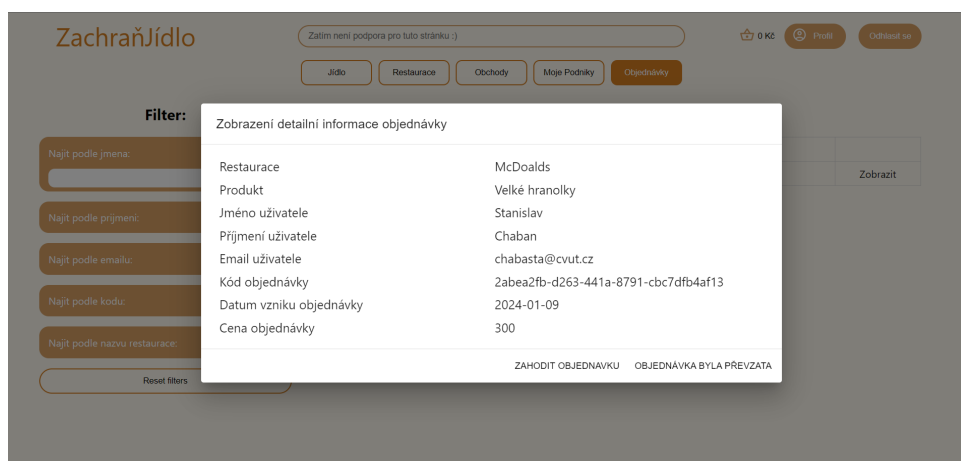
a druhá stránka je "Objednávky" kde si může vidět všechny objednávky a může je filtrovat podle jména, příjmení nebo emailu zákazníka.

Obrázek 7.13: Uživatelské rozhraní pro přidání nového podniku část 2

Navíc prodejce může otevřít detailní informace objednávky kliknutím na nějakou položku v tabulce.



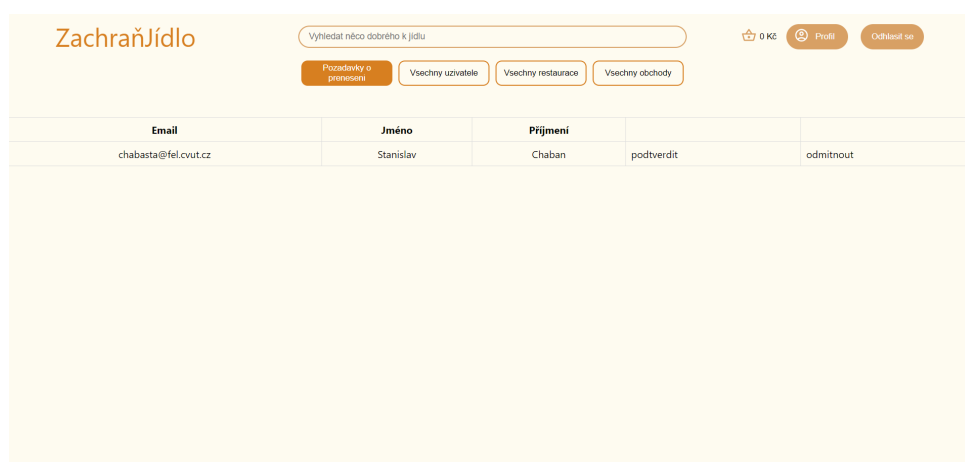
Obrázek 7.14: Uživatelské rozhraní pro zobrazení objednávek



Obrázek 7.15: Uživatelské rozhraní pro zobrazení detailní informace o objednávce

7.3.6 Administrator

Administrátor má u sebe stránku kde může vidět všechny požadavky uživatele o změně stavu profilu na prodejce a může to buď schválit nebo odmítnout.



Obrázek 7.16: Uživatelské rozhraní pro zobrazení seznamu požadavků o převedení profilu na prodejce

7.4 Zabezpečení

V aplikaci příkládám velký význam bezpečnosti, zejména pokud jde o ochranu soukromí uživatelů a zabezpečení autentizace. Pro zajištění bezpečnosti používám následující metody:

1. Hash hesla pomocí BCryptPasswordEncoder: BCryptPasswordEncoder je metoda hašování hesel, která poskytuje silné zabezpečení. Proměnění hesla uživatele na hash kód, který se následně uloží do databáze. Hašování hesla znamená, že i když někdo získá přístup k databázi, nebude schopen přechít skutečné heslo, protože hash nelze vrátit zpět. BCryptPasswordEncoder používá sůl pro každé heslo, díky čemuž je každý hash jedinečný a odolný vůči útokům hrubou silou.

```
bCryptPasswordEncoder.encode(userDto.getPassword())
```

Listing 6.1: Příklad použití BCryptPasswordEncoder [26]

```
@Configuration
public class BCryptPasswordEncoderConfig {

    @Bean
    public BCryptPasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }

}
```

Listing 6.2: Vytváření BCryptPasswordEncoder [26]

2. Generování tokenu JWT[27] po standardním přihlášení: Po úspěšném přihlášení přes standardní formu autentizace vygeneruji JSON Web Token

(JWT), který slouží k ověření identity uživatele v následných požadavcích na server. Tím je zajištěna bezpečnost relace uživatele, protože JWT obsahuje zašifrované informace o uživateli, které jsou ověřovány v každé fázi přístupu k serveru.

```
public String generateToken(UserDetails userDetails) {
    Map<String, Object> claims = new HashMap<>();
    List<String> rolesList = userDetails.getAuthorities().stream()
        .map(GrantedAuthority::getAuthority)
        .collect(Collectors.toList());
    claims.put("roles", rolesList);

    Date issuedDate = new Date();
    Date expiredDate = new Date(issuedDate.getTime() + jwtLifetime);

    return Jwts.builder()
        .setClaims(claims)
        .setSubject(userDetails.getUsername())
        .setIssuedAt(issuedDate)
        .setExpiration(expiredDate)
        .signWith(SignatureAlgorithm.HS256, secret)
        .compact();
}
```

Listing 6.3: Příklad generování jwt

3. Autentizace přes Google a ověření pomocí GoogleIdTokenVerifier: Pokud se uživatel rozhodne přihlásit přes Google, použijí token přijatý na frontendu a ověřím jej na serveru pomocí GoogleIdTokenVerifier. Tato třída od Googlu ověřuje, že token je platný a že jeho zdroj je důvěryhodný. Pokud je token ověřen jako platný, extrahuji z něj uživatelská data a použiji je ke generování vlastního JWT. To poskytuje další úroveň zabezpečení, protože identitu uživatele ověřuje důvěryhodný externí poskytovatel služeb Google.

Použití těchto metod zajišťuje, že aplikace poskytuje vysokou úroveň zabezpečení a ochrany uživatelských dat, což je klíčový faktor důvěry platformy.

```
public String getEmail(String token) throws GeneralSecurityException, IOException {
    GoogleIdTokenVerifier verifier = new GoogleIdTokenVerifier.Builder(
        new NetHttpTransport(), new GsonFactory())
        .setAudience(Collections.singletonList(clientId))
        .setIssuer("https://accounts.google.com")
        .build();
    GoogleIdToken idToken = verifier.verify(token);
    if (idToken != null) {
        Payload payload = idToken.getPayload();
        return payload.getEmail();
    } else {
        return null;
    }
}
```



```
}  
}
```

Listing 6.4: Příklad použití GoogleIdTokenVerifier [28]

■ 7.5 Závěr

V této kapitole jsme se podrobně zaměřili na proces implementace vyvíjeného systému, přičemž byly probrány zásadní aspekty celého procesu implementace. V dalších kapitolách se budeme věnovat testování aplikace, detailnímu popisu jejích funkcí a možnostem pro její další vylepšení, a to vše na základě zjištění z této kapitoly.

Kapitola 8

Testování

Testování hraje klíčovou roli v jakémkoli procesu vývoje softwaru. To nejen pomáhá detekovat a opravovat chyby, ale také zajišťuje, že aplikace splní všechny zadané požadavky a očekávání uživatelů. Ve svém projektu jsem použil různé testovací metody, abych zajistil kvalitu a spolehlivost aplikace, kterou jsem vyvinul.

Testování bylo prováděno na několika úrovních a zahrnovalo různé metody, od ručního testování uživatelského rozhraní až po automatizované testy jednotek. Každá testovací metoda měla jiný účel a pomohla zajistit různé aspekty kvality aplikace.

Účelem této části je tedy podrobně popsat proces testování, včetně metod a nástrojů používaných během vývoje, a také diskuse o výsledcích a závěrech vyvozených z tohoto procesu. Podrobný popis všech fází testování vám umožní lépe pochopit, jak byl zajištěn výkon a kvalita konečného produktu.

8.1 Testování s Postmanem

Pro podrobnější testování aplikačního API byl použit nástroj Postman. To mi umožnilo vytvářet a odesílat požadavky HTTP na server a analyzovat odpovědi. Testování s Postmanem zahrnovalo kontrolu správného fungování koncových bodů, ověřování odpovědí a zpracování chyb.

8.2 Uživatelské testování

Uživatelské testování je proces, při kterém se koncoví uživatelé reálného softwaru nebo produktu účastní testování, aby vyhodnotili jeho funkčnost, použitelnost, rozhraní, celkový dojem z produktu a jeho vhodnost pro potřeby uživatele.

Mezi cíle uživatelských testů patří:

- Posouzení snadného použití: Pochopte, jak snadné je pro koncové uživatele používat aplikaci a jak intuitivní je rozhraní.

- Detekce problémů: Identifikace chyb nebo problémů v uživatelském rozhraní, které nemusí být vývojářům zřejmé.
- Přijímání zpětné vazby: Shromažďování názorů a návrhů od uživatelů ke zlepšení produktu.
- Testování reálného scénáře: Testování toho, jak produkt funguje za skutečných podmínek použití.

Metody "uživatelských testů" mohou zahrnovat rozhovory s uživateli, pozorování jejich interakce s produktem, dotazníky po použití produktu a další techniky. Tyto testy pomáhají zajistit, že produkt je nejen technicky v pořádku, ale také příjemný a srozumitelný pro uživatele.

8.2.1 Výběr testerů

V rámci uživatelských testů mého projektu se zapojilo pět přátel. Výběr účastníků testování byl proveden náhodně, což vedlo k tomu, že skupinu testerů tvořili dva lidé bez výrazných programovacích znalostí a tři zkušení programátoři. Hlavním cílem testování bylo ověřit funkčnost a uživatelskou přívětivost aplikace, a to jak z pohledu běžných kupujících, tak z pohledu prodejců.

8.2.2 Testovací Scénáře

1) Registrace prodejce

- Vytvořte si nový uživatelský účet se statusem prodejce
- Po úspěšné registraci se odhlaste
- Přihlaste pro ověření vytvoření účtu.

Výsledek:

Všechny 5 lidí jednoduše zaregistrovali si uživatelský účet. 3 z 5 lidí hned šli na stránku profilu, a zjistili, že probíhá zpracování převedení profilu na status prodejce. Ostatní nechápali proč se jim ne změnil GUI.

2) Nákup produktů

- Přihlaste se pomocí google
- po úspěšném přihlášení využijte filtr nebo vyhledávací panel nahoře stránky pro výběr vhodného produktu.
- Otevřete detailní informace o produktu
- Přidejte produkt do košíku
- Zaplatíte objednávku

- Ověřte v profilu, v sekce historie nákupů, že do seznamu nákupů se přidala nová položka

Výsledek:

Všechny 5 lidí úspěšně prošli tímto scénářem. Jednomu člověku chyběla ve filtru možnost měnit cenu "Od nejlevnějšího"/"Od nejdražšího". Také jedna osoba by chtěla mít možnost změnit počet produktů v položce košíku

2) Přidání podniku

- Přihlaste se pomocí google
- po úspěšném přihlášení pozadejte o převedení účtu na status prodejce
- Vytvořte nový podnik
- Přidejte do podniku jeden nebo pár položek
- V závislosti na tom, co jste přesně vytvořili, přejděte na příslušnou stránku a zkontrolujte, zda byl sub-pod úspěšně vytvořen

Výsledek:

Všechny 5 lidí úspěšně prošli tímto scénářem. Jednomu člověku chyběla ve filtru možnost měnit cenu "Od nejlevnějšího"/"Od nejdražšího".

8.2.3 Závěr uživatelského testování

Tento typ testování mi pomohl podívat se na mou aplikaci ze dvou stran najednou, ze strany nakupujícího a ze strany prodávajícího. Na základě referencí testerů bylo rozhodnuto o následující verzi: 1) rozbalte filtr, což vám umožní změnit řazení ve vzestupném a sestupném pořadí 2) přidat možnost změnit počet položek v košíku 3) přidejte na hlavní obrazovku nějaké vyskakovací okno, které naznačuje, že můžete převést svůj účet na prodejce ve svém profilu

8.3 Automatizované testy jednotek

Pro testování jednotlivých komponent systému byly napsány a provedeny automatizované jednotkové testy. Tyto testy zahrnovaly použití rámce JUnit a zesměšňování závislostí pomocí Mockito [29]. Testy nám umožnily ověřit správnost obchodní logiky, interakci s databází a správné zpracování požadavků a odpovědí.

8.4 Výsledky a Závěry

Kombinovaný testovací přístup, včetně manuálních i automatických metod, zajistil důkladnou kontrolu aplikace. Manuální testování poskytlo náhled

do uživatelského prostředí a rozhraní, zatímco automatické testy zajistily spolehlivost a stabilitu kódu. Testování obecně přispělo ke zlepšení kvality a spolehlivosti vyvíjené aplikace.

8.5 Budoucí Rozvoj a Možné Vylepšení

V současné verzi aplikace je funkcionalita pro administrátora omezena pouze na schvalování nebo zamítání žádostí o změnu účtů uživatelů na status prodejců. Tato funkcionalita je základní, a v budoucích verzích by bylo vhodné rozšířit možnosti, které má administrátor k dispozici.

Co se týče prodejců, v současné době nemají možnost měnit status objednávek, ale mají k dispozici detailně zpracovanou stránku s filtrací, kde mohou vyhledávat a prohlížet detailní informace o objednávkách. Plánem pro další vývoj je zahrnout možnost odeslání kódu objednávky prostřednictvím emailu, což zvýší efektivitu komunikace mezi prodejci a zákazníky.

Na základě zpětné vazby z uživatelských testů byly identifikovány následující body pro zlepšení:

- Rozšíření funkce filtru, které umožní uživatelům měnit řazení položek ve vzestupném a sestupném pořadí.
- Přidání možnosti upravit počet položek v košíku, což zlepší uživatelskou zkušenost při nakupování.
- Implementace vyskakovacího okna na hlavní obrazovce, které by informovalo uživatele o možnosti převést svůj účet na prodejce v rámci jejich profilu.

Navíc bych rád zavedl systém odměn, například udělení odznaku za nákup 10 jídel, což by mohlo motivovat uživatele k častějšímu využívání aplikace a přispělo k řešení problému plýtvání potravinami.



Kapitola 9

Závěr

V rámci této bakalářské práce byla provedena důkladná analýza existujících aplikací pro prodej přebytků jídla a technologií, které se v této oblasti používají. Na základě této analýzy byly vytvořeny podrobné návrhy a definovány klíčové funkční a nefunkční požadavky pro nový prototyp aplikace. Toto vše poskytlo pevný základ pro výběr vhodných technologií a pro návrh samotného prototypu aplikace. Výsledkem práce je prototyp aplikace, který usiluje o snížení plýtvání jídlem. Prototyp byl řádně otestován, což zahrnovalo i uživatelské testy, a výsledky testování ukázaly, že prototyp je funkční a splňuje stanovené cíle.



Literatura

- [1] FAO *Food wastage footprint: Impacts on natural resources - Summary report*, [online]. [Cit. 11.12.2023] Dostupné z <https://www.fao.org/3/i3347e/i3347e.pdf>
- [2] *Aplikace Nesněženo*, [online]. [Cit. 11.12.2023] Dostupné z <https://nesnezeno.eco/>
- [3] *Aplikace Too good to go*, [online]. [Cit. 11.12.2023] Dostupné z <https://www.toogoodtogo.com/>
- [4] *Aplikace ResQ Club*, [online]. [Cit. 11.12.2023] Dostupné z <https://www.toogoodtogo.com/>
- [5] *Aplikace OLIO*, [online]. [Cit. 11.12.2023] Dostupné z <https://olioapp.com/en/>
- [6] *Aplikace Food for all*, [online]. [Cit. 11.12.2023] Dostupné z <https://foodforalluk.com/>
- [7] *Aplikace Food for all*, [online]. [Cit. 11.12.2023] Dostupné z <https://save.karma.life/>
- [8] *Infografika – Plýtvání potravinami: předcházení vzniku, opětovné využití a recyklace potravin*, [online]. [Cit. 11.12.2023] Dostupné z <https://www.consilium.europa.eu/cs/infographics/food-loss-and-food-waste/>
- [9] *Software requirements third edition* vydavatele Karl Wieggers And Jou Beatty
- [10] *React The library for web and native user interfaces*, [online]. [Cit. 11.12.2023] Dostupné z <https://react.dev/>
- [11] *Introduction to the Angular docs*, [online]. [Cit. 11.12.2023] Dostupné z <https://angular.io/docs>
- [12] *The Progressive JavaScript Framework Vue.js*, [online]. [Cit. 11.12.2023] Dostupné z <https://vuejs.org/guide/introduction.html>

- [13] *A framework for ambitious web developers Ember.js*, [online]. [Cit. 11.12.2023] Dostupné z <https://emberjs.com/>
- [14] *Dokumentace Backbone.js*, [online]. [Cit. 11.12.2023] Dostupné z <https://backbonejs.org/>
- [15] *What is Java?*, [online]. [Cit. 11.12.2023] Dostupné z https://www.java.com/en/download/help/whatis_java.html
- [16] *Python*, [online]. [Cit. 11.12.2023] Dostupné z <https://docs.python.org/3/tutorial/index.html>
- [17] Microsoft *A tour of the C# language*, [online]. [Cit. 11.12.2023] Dostupné z <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- [18] Microsoft *MySQL Documentation*, [online]. [Cit. 11.12.2023] Dostupné z <https://dev.mysql.com/doc/>
- [19] PostgreSQL *What Is PostgreSQL?*, [online]. [Cit. 11.12.2023] Dostupné z <https://www.postgresql.org/docs/15/intro-what-is.html>
- [20] Microsoft *SQL Server technical documentation*, [online]. [Cit. 11.12.2023] Dostupné z <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>
- [21] *What Is Oracle Database*, [online]. [Cit. 11.12.2023] Dostupné z <https://www.oracletutorial.com/getting-started/what-is-oracle-database/>
- [22] IBM *What is MongoDB?*, [online]. [Cit. 11.12.2023] Dostupné z <https://www.ibm.com/topics/mongodb>
- [23] *UML Modeling Guidelines*, [online]. [Cit. 11.12.2023] Dostupné z https://opennetworking.org/wp-content/uploads/2018/08/TR-514_UML_Modeling_Guidelines_v1.3-1-1.pdf
- [24] *Spring Boot*, [online]. [Cit. 11.12.2023] Dostupné z <https://spring.io/projects/spring-boot>
- [25] *Building REST services with Spring* [online]. [Cit. 11.12.2023] Dostupné z <https://spring.io/guides/tutorials/rest/>
- [26] *BCryptPasswordEncoder documentation* [online]. [Cit. 11.12.2023] Dostupné z <https://docs.spring.io/spring-security/site/docs/current/api/org/springframework/security/crypto/bcrypt/BCryptPasswordEncoder.html>
- [27] *Introduction to JSON Web Tokens* [online]. [Cit. 11.12.2023] Dostupné z <https://jwt.io/introduction>

- [28] *GoogleIdTokenVerifier documentation* [online]. [Cit. 11.12.2023] Dostupné z <https://cloud.google.com/java/docs/reference/google-api-client/latest/com.google.api.client.googleapis.auth.oauth2.GoogleIdTokenVerifier>
- [29] Mockito, [online]. [Cit. 11.12.2023] Dostupné z <https://github.com/mockito/mockito/wiki/FAQ>
- [30] Apache Maven Project, [online]. [Cit. 11.12.2023] Dostupné z <https://maven.apache.org/what-is-maven.html>
- [31] *Tesco v Česku testuje aplikaci proti plýtvání jídlem, Lidl zavádí výprodejové bedýnky*, [online]. [Cit. 11.12.2023] Dostupné z <https://www.kupi.cz/magazin/clanek/14923-tesco-v-cesku-testuje-aplikaci-proti-plytvani-jidlem-lidl-zavadi-vyprodejove-b>
- [32] ISBN-10: 9780078022128 *Software Engineering: A Practitioner's Approach* vydavatel Roger S. Pressmann Bruce Maxim,
- [33] Dmitriy Shevchenko *Informační systém pro správu autoservisu* Dostupné z <https://dspace.cvut.cz/bitstream/handle/10467/108770/F3-BP-2023-Shevchenko-Dmitriy-Informacni%20system%20pro%20spravu%20autoservisu.pdf?sequence=-1&isAllowed=y>