



## Zadání bakalářské práce

<b>Název:</b>	Rozšíření projektu Recipeo
<b>Student:</b>	Martin Koutenský
<b>Vedoucí:</b>	Ing. Oldřich Malec
<b>Studijní program:</b>	Informatika
<b>Obor / specializace:</b>	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	do konce letního semestru 2023/2024

### Pokyny pro vypracování

Recipeo je webová aplikace pro správu receptů, která vznikla jako bakalářská práce Vojtěcha Moravce.

Upravte a rozšiřte tuto aplikaci, aby podporovala nové funkce:

- Sdílení receptu mezi více skupinami.
- Pokročilá správa ingrediencí: jejich kategorizace do stromové struktury, vzájemná nahraditelnost, soukromá/veřejná viditelnost.
- Inventář ingrediencí - evidence jejich aktuálního množství
- Načítání ingrediencí pomocí jejich čárových kódů.

Postupujte v následujících krocích:

- Analyzujte stávající stav aplikace
- Formalizujte nové požadavky
- Navrhněte úpravy ve struktuře uložení dat a zobrazení
- Úpravy implementujte
- Výsledné řešení podrobte uživatelskému testování



Bakalárska práca

# ROZŠÍŘENÍ PROJEKTU RECIPEO

**Martin Koutenský**

Fakulta informačních technologií  
Katedra softwarového inženýrství  
Vedúci: Ing. Oldřich Malec  
11. mája 2023

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2023 Martin Koutenský. Všechny práva vyhrazené.

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

Odkaz na túto prácu: Koutenský Martin. *Rozšíření projektu Recipeo*. Bakalárska práca. České vysoké učení technické v Praze, Fakulta informačních technologií, 2023.

## Obsah

<b>Podakovanie</b>	<b>vii</b>
<b>Vyhlásenie</b>	<b>viii</b>
<b>Abstrakt</b>	<b>ix</b>
<b>Zoznam skratiek</b>	<b>x</b>
<b>Slovník</b>	<b>xi</b>
<b>1 Úvod</b>	<b>1</b>
<b>2 Ciele</b>	<b>3</b>
<b>3 Analýza</b>	<b>5</b>
3.1 Aktuálny stav . . . . .	5
3.1.1 Ingrediencie . . . . .	5
3.1.2 Recepty . . . . .	5
3.1.3 Skupiny . . . . .	6
3.2 Existujúce riešenia . . . . .	6
3.2.1 Vareni.cz . . . . .	6
3.2.2 Toprecepty.cz . . . . .	8
3.2.3 Recepty.cz . . . . .	9
3.2.4 Stock and Inventory . . . . .	11
3.2.5 KitchenPal . . . . .	12
3.2.6 Záver . . . . .	13
3.3 Čiarové kódy . . . . .	13
3.4 Výber funkcionalít . . . . .	13
3.5 Požiadavky . . . . .	15
3.5.1 Funkčné požiadavky . . . . .	15
3.5.2 Nefunkčné požiadavky . . . . .	15
3.5.3 Porovnanie požiadaviek . . . . .	15
3.6 Technológie . . . . .	16
3.6.1 Vue.js . . . . .	16
3.6.2 Vuetify . . . . .	16
3.6.3 Vuex . . . . .	16
3.6.4 Vue Router . . . . .	16
3.6.5 Firebase . . . . .	16
3.6.6 Vue Barcode and QR code scanner . . . . .	17
<b>4 Návrh</b>	<b>19</b>
4.1 Ingrediencie . . . . .	19
4.2 Recepty . . . . .	20
4.3 Skupiny . . . . .	20

4.4	Inventár . . . . .	20
4.5	Zmeny viditeľnosti . . . . .	20
4.6	Databáza . . . . .	22
<b>5</b>	<b>Implementácia</b>	<b>25</b>
5.1	Vstup do projektu . . . . .	25
5.2	Štruktúra projektu . . . . .	26
5.3	Transakcie . . . . .	26
5.3.1	Testovanie transakcií . . . . .	28
5.4	Databáza . . . . .	28
5.5	Skupiny . . . . .	29
5.6	Ingrediencie . . . . .	31
<b>6</b>	<b>Testovanie</b>	<b>37</b>
6.1	Používateľské testovanie . . . . .	37
6.1.1	Respondenti . . . . .	37
6.1.2	Priebeh testovania . . . . .	38
6.1.3	Výsledky testovania . . . . .	38
6.1.4	Záver . . . . .	39
<b>7</b>	<b>Možnosti rozšírenia</b>	<b>41</b>
7.1	Integrovanie ingrediencií do postupu receptov . . . . .	41
7.2	Rozšírenie receptu o potrebné nástroje a spotrebiče . . . . .	41
<b>8</b>	<b>Záver</b>	<b>43</b>
<b>A</b>	<b>Scenár testovania</b>	<b>45</b>
A.1	Testovanie na webovom rozhraní (veľký displej) . . . . .	45
A.2	Testovanie na mobilnom rozhraní (malý displej) . . . . .	46
<b>B</b>	<b>Snímky z aplikácie</b>	<b>47</b>
	<b>Obsah priloženého média</b>	<b>55</b>

## Zoznam obrázkov

3.1	Vareni.cz – Používateľské menu. Obsahuje kategóriu vlastných receptov a vlastných kuchárov, no nie receptov označených ako „Uvařil/a jsem“ alebo „Líbí se mi“. [2] . . . . .	7
3.2	Vareni.cz – Neštrukturovaný postup receptu. [2] . . . . .	7
3.3	Vareni.cz – Zobrazenie hodnotenia pod receptom. [2] . . . . .	7
3.4	Vareni.cz – Možnosti hodnotenia a zdieľania receptu. [2] . . . . .	7
3.5	Toprecepty.cz – Používateľské menu. Obsahuje veľa vlastných kategórií, podľa ktorých sa dá upravovať obsah, ktorý používateľ vidí. [3] . . . . .	8
3.6	Toprecepty.cz – Menu používateľovej aktivity. [3] . . . . .	9
3.7	Toprecepty.cz – Práca s ingredienciami. [3] . . . . .	9
3.8	Recepty.cz – Používateľské menu. Obsahuje niekoľko vlastných kategórií, do ktorých vie používateľ zaradiť obsah, ktorý vidí. [4] . . . . .	10
3.9	Recepty.cz – Nákupný zoznam. Položky sa z neho dajú odstrániť separátne alebo skupinovo „po receptoch“. [4] . . . . .	10
3.10	Stock and Inventory – Hlavné menu. Zobrazuje hlavné funkcionality aplikácie. [5]	11
3.11	Stock and Inventory – Podrobné menu funkcionalít pre dokumentovanie tržieb, záväzkov, výdajov, objednávok atď. [5] . . . . .	12
3.12	Porovnanie pôvodných a nových požiadaviek. . . . .	15
4.1	Stav pred zmenou viditeľnosti . . . . .	21
4.2	Stav po zmene viditeľnosti . . . . .	21
4.3	Rozšírený doménový model . . . . .	22
4.4	Rozšírený dokumentový model . . . . .	23
5.1	Adresárová štruktúra[1] . . . . .	26
5.2	Porovnanie použitia <i>batched writes</i> a <i>transakcie</i> . . . . .	27
5.3	Firestore database – Dialóg pre exportovanie databázy Firestore. [24] . . . . .	29
B.1	Detail ingrediencie. . . . .	47
B.2	Formulár pre pridanie a úpravu ingrediencie. . . . .	48
B.3	Použitie skeneru vo formulári pre pridanie a úpravu ingrediencie. . . . .	49
B.4	Detail skupiny. . . . .	50
B.5	Detail inventáru skupiny s použitím skeneru. . . . .	51
B.6	Dialóg pre označenie receptu. . . . .	52

## Zoznam tabuliek

3.1	Porovnanie ponúknutých funkcionalít . . . . .	14
-----	---	----

## Zoznam výpisov kódu

1	Importovanie komponentov <i>StreamBarcodeReader</i> a <i>ImageBarcodeReader</i> pomocou JavaScript-u a ich použitie v HTML . . . . .	18
2	Príkazy pre stiahnutie vývojárskych balíčkov . . . . .	25
3	Transakcia pre pridávanie novej ingrediencie . . . . .	28
4	Funkcia použitá pre pridanie chýbajúcich polí so základnou hodnotou. . . . .	30
5	Funkcia použitá pre určenie vzťahu nadržadenosti dvoch ingrediencií. . . . .	32
6	Funkcie pre aktualizovanie zoznamov povolených ingrediencií. . . . .	33
7	Funkcie pre aktualizovanie zoznamov označených ingrediencií. . . . .	34



*Chcel by som poďakovať hlavne svojmu vedúcemu Ing. Oldřichu Malcovi, ktorý mi vždy vyšiel v ústrety a pomohol s každým problémom, s ktorým som si nevedel dať rady. Ďalej svojej rodine, ktorá mi umožnila bez problémov študovať a svojim kamarátom, ktorí ma pomáhali motivovať.*

## Vyhlásenie

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, avšak pouze k nevýdělečným účelům. Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 11. mája 2023

.....

## Abstrakt

Obsahom práce je postup rozširovania existujúcej aplikácie Recipeo. Aplikácia sa zaoberá varením a receptami. Rozšírenie pridáva funkcie zdieľania receptov a inventáru, aby išlo aj o pomocníka v domácnosti. Práca sa začína analýzou požiadaviek, pokračuje návrhom zmien a ďalej rieši konkrétne implementačné problémy. Aplikácia je písaná vo Vue.js s backendom od služby Firebase od Google. Obsahom je ako frontend, tak aj backend. Výsledkom je verejne dostupná aplikácia.

**Kľúčová slova** webová aplikácia, frontend, receptár, inventár, kategorizácia ingrediencií, Vue, Vuetify, Firebase

## Abstract

This bachelor's thesis describes the extension of an existing application Recipeo. The application deals with cooking and recipes. The extension adds recipe sharing and group inventory in order to elevate itself to home assistant. It starts with an analysis of requirements, followed by a design of changes and then deals with specific implementation issues. Application uses Vue.js as its main framework with a backend in Firebase from Google. This thesis describes both frontend and backend. The result is a publicly accessible application.

**Keywords** web application, frontend, recipe book, inventory, ingredient categorization, Vue, Vuetify, Firebase

## Zoznam skratiek

API	Application Programming Interface
EAN	European Article Numbering Association
GTIN	Global Trade Item Number
npm	Node Package Manager
SDK	Software Development Kit
SFC	Single-File Components
SPA	Single-Page Application
UI	User Interface
UX	User Experience
Yarn	Yet Another Resource Negotiator

## Slovník

batched writes	zázpisy po dávkach
bind	previazat
bucket	kontajner na dáta v cloudovom úložisku
decode	dekódovať
emulator	emulátor
error	chyba
framework	nosná štruktúra
loaded	načítané
result	výsledok
rule	pravidlo
true	pravda
watcher	pozorovateľ





## Kapitola 1

# Úvod

Varenie predstavuje pre mnohých ľudí každodennú činnosť, avšak nie všetci ju zvládajú s ľahkosťou. Aby si proces varenia zjednodušili, používajú cudzie recepty, ktoré nájdu, a to väčšinou na webových portáloch.

Projekt Recipeo[1], na ktorý táto bakalárska práca nadväzuje, je konkurenčný portál, ktorý sa po porovnaní s aktuálne populárnymi portálmi pokúsil vytvoriť vlastné riešenie so zameraním na UI a UX. Kvôli rozsahu práce neboli všetky žiadané funkcionality dodané.

Táto práca zhodnotí funkcionality, ktoré iné portály ponúkajú, porovná ich s tými, ktoré sú pre užívateľa dôležité, a vyberie tie, ktoré ju odlišia.

Projekt Recipeo nadobudne podobu nielen webového receptáru, ale aj platformy na zdieľanie receptov so zhodnou skupinou používateľov a pomocníka pre organizovanie aktuálneho inventára skupiny (napr. domácnosť, klub, kancelária...).







## Kapitola 2

# Ciele

Cielom práce je existujúcu webovú aplikáciu Recipeo rozšíriť o nové funkcionality. Recipeo je webová aplikácia na správu receptov, ktorej hlavným cieľom je priniesť príjemný používateľský zážitok. Tieto funkcionality tvoria čiastkové ciele.

Prvým čiastkovým cieľom je pridať používateľom možnosť zdieľať recepty medzi skupinami, ktorými sú členom. Toto zahŕňa zdieľanie vlastných receptov, ktoré používateľ pridal, ale aj cudzích receptov, ktoré používateľ našiel medzi verejnými a chce, aby k nim mala skupina rýchly prístup.

Druhým čiastkovým cieľom je vytvoriť robustnejší systém správy ingrediencií, ktorý podporuje stromové členenie, ako aj rýchly náhľad pre vzájomnú nahraditeľnosť. Tá je motivovaná faktom, že veľa ingrediencií je vo varení naozaj nahraditeľných. Ingrediencie budú mať rôzne nastavenie viditeľnosti, ktoré ovplyvní, ako sa k nim aplikácia správa. Neverejná viditeľnosť neumožní, aby iní používatelia danú ingredienciu videli v zozname všetkých verejne dostupných ingrediencií a zároveň budú kladené menšie požiadavky na kvalitu názvu, popisu a obrázku. Používatelia však budú môcť vidieť túto ingredienciu, ak bola použitá vo verejnom recepte. Ingrediencie s verejnou viditeľnosťou budú musieť spĺňať prísnejšie požiadavky, aby boli reprezentatívne pred ostatnými používateľmi. Ďalej budú verejne prístupné všetkým používateľom, aby ich videli a používali.

Tretím čiastkovým cieľom je pridať funkcionality inventáru, ktorý eviduje aktuálny stav ingrediencií. Inventár bude viazaný na skupinu používateľov, ktorej patrí.

Štvrtým čiastkovým cieľom je pridať používateľom možnosť načítať ingrediencie pomocou ich čiarového kódu. Toto sa uplatní v rôznych kontextoch, ako napríklad pri vytváraní novej ingrediencie alebo pri pridávaní/odoberaní určitého množstva už existujúcej ingrediencie do/z niektorého inventáru.



## Kapitola 3

# Analýza

### 3.1 Aktuálny stav

Webová aplikácia Recipeo prešla doteraz iba jedným vývojovým obdobím, čo bolo trvanie bakalárskej práce originálneho autora. Z tohto dôvodu boli rozsah a funkcionality obmedzené. K verzii, ktorá bola výsledkom predošlej práce, budem odkazovať ako k „pôvodnej“ a k verzii, ktorá vznikla ako výsledok tejto bakalárskej práce, zase ako k „rozšírenej“.

#### 3.1.1 Ingrediencie

V pôvodnej verzii aplikácie Recipeo [1] bolo možné pridávať ingrediencie používateľom. Nutné informácie boli:

- názov,
- odkaz na rohlik.cz,
- odkaz na kosik.cz.

Ďalšie voliteľné informácie boli:

- štítky,
- obrázok alebo odkaz na obrázok.

Interne neboli ukladané žiadne ďalšie informácie. Štruktúra ingrediencií bola teda veľmi jednoduchá, skoro žiadna – iba kolekcia. Neumožňovala žiadne ďalšie vzťahy medzi ingredienciami.

#### 3.1.2 Recepty

V pôvodnej verzii aplikácie Recipeo [1] bolo možné vytvárať recepty. Nutné informácie boli:

- názov,
- viditeľnosť,
- majiteľ,
- aspoň jeden krok receptu,
- aspoň jedna ingrediencia receptu.

Ďalšie voliteľné informácie boli:

- štítky,
- obrázok.

Pojmy vlastníctva a viditeľnosti boli teda už zavedené. Zatiaľ slúžili iba pre zobrazenie receptov pri prezeraní všetkých receptov.

### 3.1.3 Skupiny

V pôvodnej verzii aplikácie Recipeo [1] existovali skupiny. Mali zatiaľ iba jedno využitie – vedeli vlastníť recepty. Ak bola skupina vlastníkom receptu, každý jej člen mal práva prezerat a upravovať daný recept.

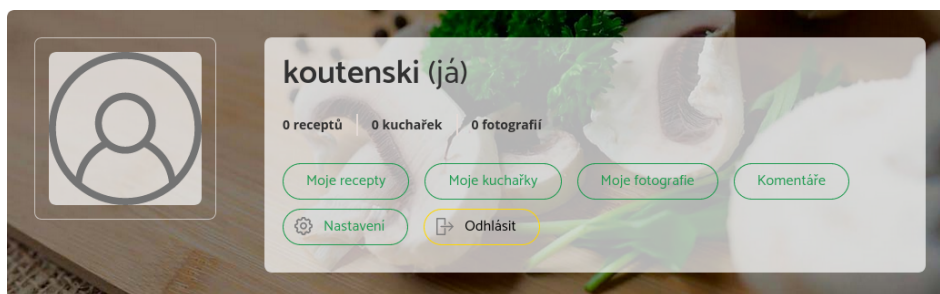
## 3.2 Existujúce riešenia

Tento projekt som prevzal už s istými znalosťami o existujúcich riešeniach. Pôvodná verzia aplikácie totižto mierila k vytvoreniu portálu, ktorý bude na rozdiel od vtedy existujúcich veľmi jednoduchý na používanie s veľkým dôrazom na čistý a intuitívny dizajn. Tieto riešenia však treba podrobiť aj kvalitatívnej skúške ohľadom funkcionalít, ktoré ponúkajú. Spomenuté budú funkcionality relevantné k tejto práci. To, či aplikácia zobrazuje reklamy, odkazy na iné články z iných webových aplikácií alebo zdieľanie na iné platformy nebude zahrnuté v separátnych opisoch.

### 3.2.1 Vareni.cz

Vareni.cz [2] ponúka tieto funkcionality:

1. Vytváranie receptu. Je limitované v oblasti popisu krokov, kedy ide iba o jeden text, ktorý autor vloží. Postup nie je rozdelený teda podľa krokov, ak sa autor nerozhodne tak formátovať svoj popis.
2. Prezeranie receptov. Keďže recepty neobsahujú nič podobné nastaveniu viditeľnosti, sú všetky verejné.
3. Označenie receptu ako „Líbí se mi / Nelíbí se mi“. Toto označenie je čisto pre potreby receptu, vďaka čomu je pre ostatných užívateľov vidno, koľko ľudí ho považuje za dobrý.
4. Označenie receptu ako „Uvařil/a jsem“. Predpokladal som, že sa bude dať nahliadnúť na všetky používateľom uvarené recepty. Nielen, že sa na ne nedá nahliadnúť, no toto označenie vyzerá, že nemá žiaden efekt. Po stlačení príslušného tlačidla nenastane žiadna potvrdzujúca animácia alebo oznámenie, ako u iných tlačidiel. Je možné, že táto funkcionalita je dočasne zablokovaná.



■ **Obr. 3.1** Vareni.cz – Používateľské menu. Obsahuje kategóriu vlastných receptov a vlastných kuchárk, no nie receptov označených ako „Uvařil/a jsem“ alebo „Líbí se mi“: [2]

#### Postup přípravy \*

Toto je pole pro postup přípravy. Uživatel má napsat postup (odporúčane v 1. osobe množného čísla) od přípravy surovin až po servírování.

Rozdelenie do krokov nie je funkcionality aplikácie, ale užívateľovo rozhodnutie.

1. Uvarite vodu.
2. Zalejte čaj.

■ **Obr. 3.2** Vareni.cz – Neštrukturovaný postup receptu. [2]

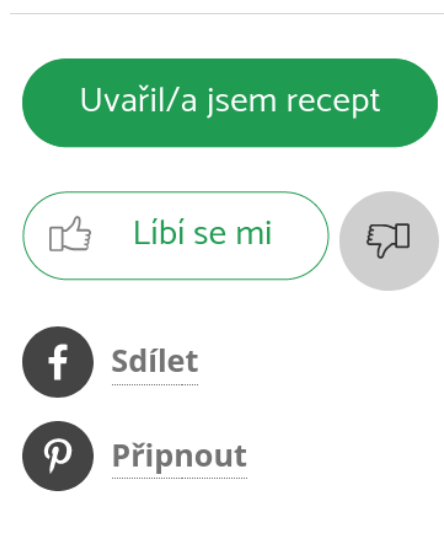


### Jahodové pudinkové palačinky ze zakysané...

Dezerty

★ 4.4 56 hodnocení

■ **Obr. 3.3** Vareni.cz – Zobrazenie hodnotenia pod receptom. [2]



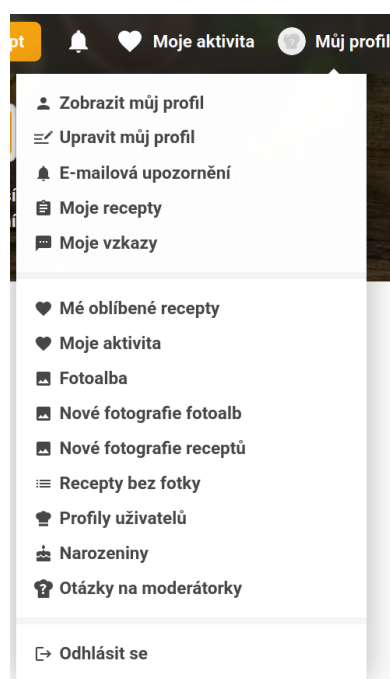
■ **Obr. 3.4** Vareni.cz – Možnosti hodnotenia a zdieľania receptu. [2]

Vareni.cz neponúka žiadne funkcionality ohľadom skupín a zdieľania receptov v rámci aplikácie. Zdieľať je možné len na iné platformy ako Facebook či Pintereset, alebo prostredníctvom odkazu na stránku samotnú. Ďalej ide výhradne o receptár, čiže nič podobné inventáru nie je súčasťou aplikácie.

### 3.2.2 Toprecepty.cz

Toprecepty.cz [3] ponúkajú tieto funkcionality:

1. Vytváranie receptu. Umožňuje zadávať postup po jednotlivých krokoch. Ingrediencie sú ale neštrukturované.
2. Prezeranie receptov. Podobne ako u Vareni.cz sú všetky verejné.
3. Recepty sa dajú označiť za uvarené a za obľúbené. Dá sa k nim pridať hodnotenie, poznámka alebo komentár. Pomocou každej z týchto metód sa dajú recepty filtrovať a prezeráť. Taktiež sa dajú prezeráť nedávno navštívené recepty.
4. Priame správy medzi používateľmi.
5. Zdanlivo prácu s ingredienciami, keď sa dajú niektoré ingrediencie z receptu označiť. Označenie ingrediencie má pravdepodobne za úlohu zjednodušiť varenie komplikovaných receptov, keďže sa tak dá sledovať, ktoré ingrediencie už boli použité. Nenapĺňa však funkciu inventáru.



■ **Obr. 3.5** Toprecepty.cz – Používateľské menu. Obsahuje veľa vlastných kategórií, podľa ktorých sa dá upravovať obsah, ktorý používateľ vidí. [3]

## Moje aktivita

Veškerá Vaše aktivita na webu je zaznamenaná v této sekci **Moje aktivita**, kde se můžete přehledně podívat na recepty, které jste přidali, navštívili, ohodnotili, okomentovali, přidali do oblíbených či uvařili.

Vaším oblíbeným receptům se podrobněji věnuje sekce **Oblíbené recepty**, kde si můžete recepty, které jste si tam uložili, filtrovat je a případně v nich vyhledávat.

Přidané recepty (0)	Přidané fotografie (0)	Oblíbené (1)	Uvařené (1)
Hodnocené (1)	Navštívěné (3)	Komentované (0)	Vytisknulé (1)

■ Obr. 3.6 Toprecepty.cz – Menu používateľovej aktivity. [3]

**Suroviny** — 2 porce +

- 300 g hovézího steaku
- 2 lžíce kukuřičného škrobu
- 4 lžíce rostlinného oleje
- Na omáčku:**
- 1 lžíce rostlinného oleje
- 1 lžička prolisovaného česneku
- 1 lžička nastrohaného zázvoru
- 1 lžíce hnědého cukru
- 4 lžíce tmavé sójové omáčky
- 50-70 ml vody
- sůl (dle chuti)
- čerstvě mletý pepř
- K podávání:**
- 2 zelené jarní cibulky

■ Obr. 3.7 Toprecepty.cz – Práca s ingredienciami. [3]

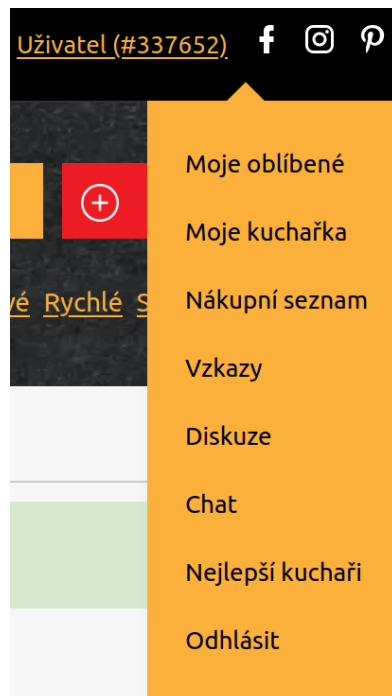
Toprecepty.cz ponúkajú veľa funkcionalít na vlastné kategorizovanie a prezeranie receptov. Ponúkajú aj priame správy, no žiadne skupiny používateľov. Možnosti zdieľania receptov sú preto limitované.

### 3.2.3 Recepty.cz

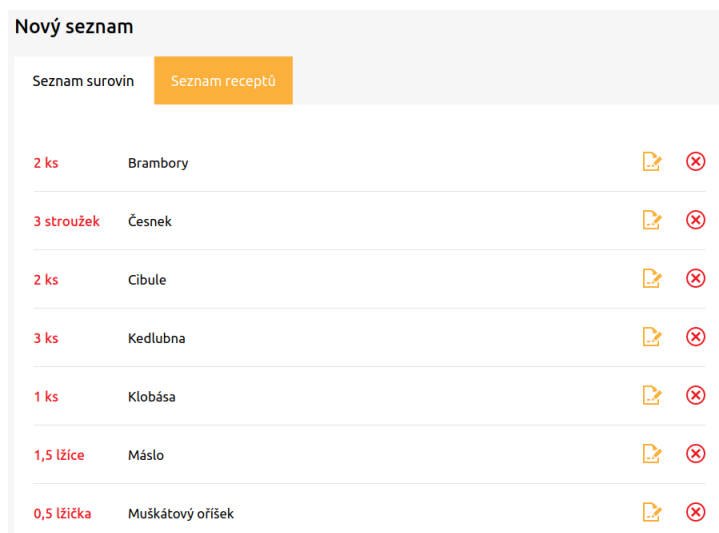
Recepty.cz [4] ponúkajú tieto funkcionality:

1. Vytváranie receptu. Tiež umožňuje zadávať postup po jednotlivých krokoch.
2. Prezeranie receptov. Znovu sú všetky recepty verejné.
3. Recepty sa dajú ohodnotiť a dá sa k nim pridať komentár. Nie je ich však možné podľa týchto hodnotení a komentárov neskôr prezeráť.
4. Vyhľadávania receptov a kategórie je možné zaradiť medzi obľúbené a rýchlo sa k nim vrátiť.
5. Recepty je možné zaradiť do kuchárky, ktorá sa spolu s poznámkami dá vytlačiť. Používateľ si vie teda obľúbené recepty zaradiť sem.
6. Priame správy medzi používateľmi.

7. Nákupný zoznam, do ktorého sa dajú jedným kliknutím pridať všetky ingrediencie receptu. Dajú sa zo zoznamu odstraňovať po jednej alebo po receptoch, ktorým patrili.



■ **Obr. 3.8** Recepty.cz – Používateľské menu. Obsahuje niekoľko vlastných kategórií, do ktorých vie používateľ zaradiť obsah, ktorý vidí. [4]



■ **Obr. 3.9** Recepty.cz – Nákupný zoznam. Položky sa z neho dajú odstrániť separátne alebo skupinovo „po receptoch“. [4]

Recepty.cz ponúkali nejaké funkcionality na vlastné kategorizovanie a prezeranie receptov. Ponúkali aj priame správy, no žiadne skupiny. Možnosti zdieľania sú preto limitované. Existujú

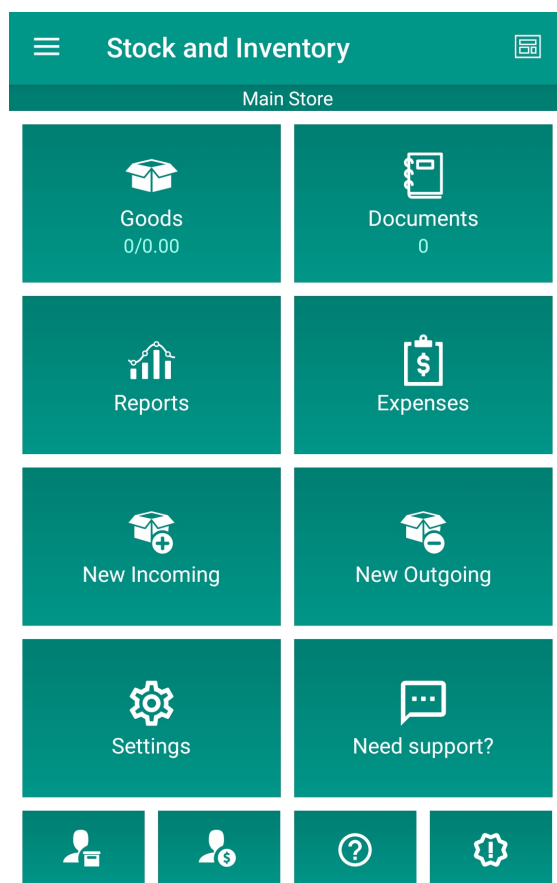


tu však verejné diskusie, no tie by mali plniť iné účely – otázky na bežné problémy, zdieľanie rád a trikov... Používatelia ich však vedia využiť miesto skupín na zdieľanie receptov naraz. Tieto diskusie sú plne verejné.

### 3.2.4 Stock and Inventory

Stock and Inventory [5] je mobilná aplikácia. Vybral som ju ako zástupcu štýlu, ktorý má viacero prevedení.

Tieto mobilné aplikácie ponúkajú komplexný inventár s možnosťami pravidelného pridávania a odberu tovaru, so štatistikami spotreby a ceny služby na zákazníka. Ide preto o aplikáciu vhodnú pre malé a stredné podniky. Bežný používateľ, ktorý chce variť a sledovať obsah svojej chladničky a špajze predtým, ako pôjde nakúpiť, bez toho, aby sa musel cestou z práce zastaviť doma, by nevyužil veľkú väčšinu funkcionalít.



■ Obr. 3.10 Stock and Inventory – Hlavné menu. Zobrazuje hlavné funkcionality aplikácie. [5]

Reports		
Main Store		
Debts by Customer	☆	⋮
Debts by Supplier	☆	⋮
Expenses	☆	⋮
List of documents	☆	⋮
List of transactions	☆	⋮
Profits and Losses	☆	⋮
Purchases by Supplier	☆	⋮
Reorder Report	☆	⋮
Sales by Customer	☆	⋮
Sales by Date	☆	⋮
Sales by Items	☆	⋮
Sales by Month	☆	⋮
Stock Movement	☆	⋮

■ **Obr. 3.11** Stock and Inventory – Podrobné menu funkcionalít pre dokumentovanie tržieb, záväzkov, výdajov, objednávok atď. [5]

### 3.2.5 KitchenPal

Táto aplikácia sa mi dostala do pozornosti až dlho potom, ako som robil prvé kolo prieskumu a začal pracovať na rozšírení Recipeo. Keď som neskôr znovu pozeral konkurenčné riešenia, našiel som ju na Play Store [6] v sekcii sponzorované, čiže predpokladám, že svoju marketingovú kampaň začali až po mojom prieskume, no pred dokončením projektu.

KitchenPal [7] je iba mobilná aplikácia, ktorá ponúka funkcie inventáru v rozsahu vhodného pre jednotlivca. Používateľ vie vyhľadávať recepty a vidí, ktoré ingrediencie už vlastní. Z ingrediencií, čo nevlásta vie vytvárať nákupné zoznamy. Používateľ vie označiť recepty ako obľúbené a prezerať ich v separátnej kategórii. Je možné zdieľať obsah inventáru s inými používateľmi prepojením ich účtov. Je to však limitované na jednu skupinu.

Táto aplikácia je najpodobnejšia tomu, čo má tento projekt v pláne, no stále je tu niekoľko rozdielov:

- Ide iba o mobilnú aplikáciu.
- Naraz existuje z pohľadu používateľa len jedna skupina.
- Zatiaľ podporuje iba anglický a francúzsky jazyk.

- Niektoré služby sú zahrnuté len v spoplatnenom programe Premium.

### 3.2.6 Záver

Spomedzi webových portálov, ktoré sú prominentné na českej scéne, sú Toprecepty.cz najpodobnejšie plánovanému rozšíreniu. Ani tie však neponúkali takú prácu so skupinami, inventárom a viditeľnosťou ingrediencií, ktorú považujem za vhodnú pre takúto aplikáciu.

Mimo webových portálov existuje veľa inventárnych aplikácií, ktoré sú pre použitie v domácnosti až nepotrebné komplexné a nejaké riešenia, v podobe napríklad KitchenPal, ktoré sú podobné plánovanému rozšíreniu, no zatiaľ mimo českej scény.

## 3.3 Čiarové kódy

Čiarový kód je, ako názov napovedá, iba kód. Informácia, ktorú prenáša, má význam iba keď poznáme jej kontext. Čiarové kódy, s ktorými plánuje aplikácia Recipeo pracovať, sú skoro výhradne obchodné označenia potravín a produktov bežne sa vyskytujúcich v domácnosti.

GTIN (Global Trade Item Number) [8] je jedinečné označenie všetkých obchodných produktov podľa štandardu určeného GS1 [9]. Spolu s označením EAN [10] popisuje všetky bežné produkty. Toto označenie sa skladá z viacerých častí. Obsahuje GS1 prefix (dvojciferné označenie krajiny pôvodu), označenie výrobcu, označenie produktu a cifru kontrolného súčtu. Názvy kódov, ktoré sa používajú, sú EAN-8, EAN-13, GTIN-8, GTIN-12, GTIN-13 a GTIN-14. Všetky spadajú pod lineárne čiarové kódy. Keďže táto aplikácia ráta primárne s použitím potravín z európskeho trhu, použitý bude najmä kód EAN-13.

Vďaka tomuto štandardu je nielen možné prečítať kódy produktov od rôznych výrobcov, ale zároveň mať istotu, že nenastane žiadna kolízia.

## 3.4 Výber funkcionalít

Prieskum existujúcich riešení ukázal, že viacero funkcionalít je zdieľaných. Ide preto o nosné funkcionality pre webový receptár. Rôzne riešenia sa častejšie líšili kvalitou a spôsobom prevedenia, ako odlišnou funkcionalitou, hoci rozdiely sa vyskytovali aj v rámci nej. V tabuľke 3.1 je uvedený ich prehľad.

Aplikácia Recipeo niektoré funkcionality už ponúka a ďalšie sú obsahom tejto práce.

Už implementované funkcionality: vytváranie receptov, štrukturovaný postup, objekt ingrediencie, vlastné ingrediencie.

Vybrané funkcionality: označenie receptov, prezeranie označených receptov, zdieľanie receptov, inventár, identifikovanie ingrediencií podľa čiarového kódu.

Typ	vareni.cz	toprecepty.cz	recepty.cz	Stock and Inventory	KitchenPal
Vytváranie receptov	Áno	Áno	Áno	Nie	Áno – Premium
Štrukturovaný postup	Nie	Áno	Áno	Nie	Áno
Označenie receptov	Áno	Áno	Áno – K	Nie	Áno
Prezeranie označených receptov	Nie	Áno	Áno – K	Nie	Áno – Premium
Hodnotenie receptov	Áno	Áno	Áno	Nie	Áno
Zdieľanie receptov	Nie	Nie	Nie	Nie	Áno – Premium
Priame správy medzi používateľmi	Nie	Áno	Áno	Nie	Nie
Objekt ingrediencie	Nie	Nie	Nie	Áno, hocijaká položka	Áno
Vlastné ingrediencie	Nie	Nie	Nie	Áno	Áno – Premium
Nákupný zoznam	Nie	Nie	Áno	Áno	Áno
Inventár	Nie	Nie	Nie	Áno	Áno
Označenie ingrediencií za použité	Nie	Áno	Nie	Netýka sa	Nie
Identifikovanie ingrediencií podľa čiarového kódu	Nie	Nie	Nie	Áno	Áno

Áno – K

Áno – Premium

Áno, s použitím kuchárky (nie vstavaná funkcionálnosť)

Áno, so spoplatneným Premium

■ **Tabuľka 3.1** Porovnanie ponúknutých funkcionalít

## 3.5 Požiadavky

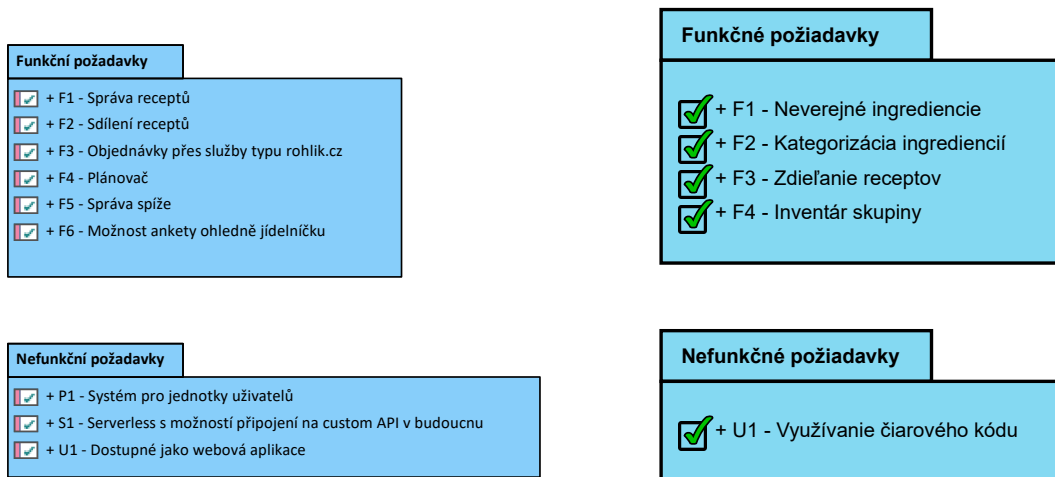
### 3.5.1 Funkčné požiadavky

- F1: Neverejné ingrediencie. Používatelia budú mať možnosť vytvoriť neverejné ingrediencie, ktoré budú patriť buď im, alebo niektorej skupine, ktorej sú členom.
- F2: Kategorizácia ingrediencií. Používateľ bude mať možnosť popísať vzťahy medzi ingredienciami (viac/menej všeobecná). Tento vzťah bude tvoriť stromovú štruktúru, ktorá umožní jednoduchšie prezeranie podobných ingrediencií.
- F3: Zdieľanie receptov. Používateľ bude mať možnosť zdieľať verejný recept skupine, do ktorej patrí. Používateľ bude vedieť zdieľať recept súčasne vo viacerých skupinách. Každý člen skupiny bude vedieť prezerat všetky recepty, ktoré boli skupine zazdieľané.
- F4: Inventár skupiny. Skupina bude môcť viesť inventár ingrediencií a sledovať ich počet. Každý člen skupiny bude mať možnosť tento inventár upravovať.

### 3.5.2 Nefunkčné požiadavky

- U1: Identifikácia ingrediencií bude podporovať ich obchodné označenie čiarovým kódom. Bude sa dať použiť pri vytváraní ingrediencie alebo pri pridávaní ingrediencie do inventára skupiny.

### 3.5.3 Porovnanie požiadaviek



(a) Pôvodné požiadavky aplikácie Recipeo. [1]

(b) Rozšírené požiadavky aplikácie Recipeo.

■ **Obr. 3.12** Porovnanie pôvodných a nových požiadaviek.

Pri porovnaní požiadaviek je vidno nasledovné:

- V pôvodnej verzii aplikácie Recipeo bolo sústredenie na základy aplikácie (nefunkčné požiadavky), na základy receptárnej aplikácie (F1, F2), na integrovanie populárnych českých služieb dovozu potravín (F3) a na vlastné odlišenie od konkurenčných riešení (F4, F5, F6).

- Rozšírená verzia má zhodné požiadavky (F2 -> F3, F5 -> F4), pretože neboli v pôvodnej práci vypracované na žiadanú úroveň (zdieľanie receptov je len s verejnosťou, nie zámerne so skupinou a správa špajze nebola naplnená).
- Rozšírená verzia sa sústreďuje na funkcionality, nie na samotný systém webovej aplikácie.

## 3.6 Technológie

Keďže ide o už zabehnutý projekt, technológie boli vybrané. Samozrejme bola možnosť niektoré vymeniť v prípade, že existuje lepšia alternatíva. Všetky úvahy budú spomenuté pri konkrétnych komponentoch.

### 3.6.1 Vue.js

Hlavná použitá technológia je Vue.js [11]. Vue.js je JavaScriptový framework, ktorý stavia na HTML a CSS. Je určený pre vytváranie používateľských rozhraní pre webové aplikácie. Vue.js pracuje primárne skladaním čiastkových komponentov do väčšieho celku. Tieto komponenty sú znovupoužiteľné naprieč celou aplikáciou a zároveň sú parametrizovateľné. Ďalej sú zvyčajne SFC (Single-File Components), čo znamená, že jeden súbor popisuje celý a práve jeden komponent. Obsahuje teda zároveň HTML, CSS a JavaScript, aby naplnil všetky potrebné funkcie.

Vue.js ponúka aktuálne verziu 2 a 3. V čase vývoja pôvodnej aplikácie nebola verzia 3 ešte plne dokončená, a preto sa použila verzia 2. Tá ponúka všetky funkcionality potrebné pre tento projekt. V budúcnosti je v pláne previesť aplikáciu na verziu 3, no spolu s vedúcim práce sme rozhodli, že to je vhodné urobiť až s plánovaným prechodom z *options API* na *composition API*.

### 3.6.2 Vuetify

Vuetify [12] je grafická knižnica pre Vue.js. Jej hlavným prínosom je, že má veľa preddefinovaných komponentov, ktoré sú stále vysoko parametrizovateľné. Pri práci vývojár teda vytvára svoje špecifické komponenty pomocou použitia a upravenia niekoľkých všeobecných komponentov od Vuetify.

### 3.6.3 Vuex

Vuex [13] je knižnica pre spravovanie globálneho stavu aplikácie. Ten je dostupný z ľubovoľného komponentu v aplikácii, no princípom pravidiel je zaistené, že k dátam bude prístupované a budú upravované iba očakávaným spôsobom.

### 3.6.4 Vue Router

Vue Router [14] je nástroj umožňujúci jednoduchú navigáciu v rámci SPA (Single Page Application). Ku komponentom sa prideliť adresa a pomocou prechodov sa na označené komponenty presunieme. Toto je úzko späté s SPA, ktoré sú stavané tak, aby ponúkali všetku plánovanú funkcionality, no bez potreby obnovenia stránky.

### 3.6.5 Firebase

Firebase [15] je platforma pre vývoj aplikácií od spoločnosti Google. Ponúka veľa rôznych produktov a dá sa na nej vyvíjať vo viacerých jazykoch vďaka multiplatformovému SDK. Všetky

produkty dokážu navzájom komunikovať. Použité produkty sú *Firestore*, *Storage*, *Hosting* a *Functions*. Firebase je prepojitelný s ďalšími službami Google ako sú *Analytics* [16] či *Ads* [17].

V kombinácii použitia týchto produktov nie je potrebné spravovať vlastné servery, na ktorých by sa ukladali dáta či nachádzala aplikácia. Firebase ponúka niekoľko plánov, ktoré majú rôzne úrovne obmedzení. Medzi nimi je aj plne bezplatný plán *Spark* a do prekročenia limitov bezplatný plán *Blaze*.

### 3.6.5.1 Firestore

Firestore [18] je produkt od Firebase, ktorý ponúka NoSQL dokumentovú databázu. NoSQL dokumentové databázy sú navrhnuté so zámerom častého čítania a zmeny dát. Dáta sú uchovávané v podobe JSON dokumentov, ktoré sú zoskupené v kolekciách.

### 3.6.5.2 Storage

Storage [19] je produkt od Firebase, ktorý ponúka cloud úložisko pre užívateľské dáta, ako sú obrázky či videá. Je možné dáta kategoricky ukladať pre štrukturovaný prístup.

### 3.6.5.3 Hosting

Hosting [20] je produkt od Firebase, ktorý zabezpečí dostupnosť aplikácie na ich serveroch.

### 3.6.5.4 Functions

Functions [21] je produkt od Firebase, ktorý ponúka servery, na ktorých sú spúšťané funkcie backendu. Väčšina logiky a vnútorných procesov aplikácie by nemala byť počítaná na klientovi, pretože by zbytočne znížila rýchlosť aplikácie. Týmto spôsobom sú náročnejšie výpočty spracované mimo aplikácie a tej sa iba pošlú finálne výsledky. Tak, ako aj prevádzkovanie vlastného serveru, je tu výhoda oddelenia citlivej práce s dátami od používateľa. Je to preto vykonané bezpečne a používateľ dostane práve tie informácie, ktoré potrebuje.

## 3.6.6 Vue Barcode and QR code scanner

Balíček *vue-barcode-reader* [22] ponúka funkcionality skenovania čiarových kódov z obrázkov alebo spusteného video prenosu. Výhodou je, že má dve verzie, z ktorých je jedna kompatibilná s Vue.js 2 a druhá s Vue.js 3. Je teda možné použiť rovnakú knižnicu aj po plánovanom prechode na verziu Vue.js 3.

Balíček obsahuje dva komponenty: *ImageBarcodeReader*, ktorý načíta čiarový kód z dodaného obrázku, a *StreamBarcodeReader*, ktorý načíta čiarový kód z dodaného video prenosu z kamery zariadenia.

Pre použitie *StreamBarcodeReader*, ktorý je vhodný pre túto aplikáciu, je potrebné vložiť komponent do HTML súboru. Po vytvorení komponentu sa spustí jeho inicializačná funkcia, ktorá získa prístup k prenosu z kamery (používateľ môže toto správanie odmietnuť, ak nedá danej webovej stránke svoje povolenie). Po úspešnom dokončení vytvorí *StreamBarcodeReader* udalosť `loaded` a pošle ju nadradenému elementu. Vtedy sa vytvorí grafický komponent skeneru. Po načítaní čiarového kódu je vytvorená udalosť `decode` s parametrom `result` (výsledok), ktorá je znovu poslaná nadradenému elementu. Nadradený element vie tieto udalosti odchytiť a reagovať na ne zavolaním vlastných metód.

Použitie *ImageBarcodeReader* je podobné, ale grafický element je celý čas viditeľný a ide o dialóg pre výber súboru. Inicializačná funkcia nevytvorí udalosť `loaded`. Úspešné prečítanie zo súboru vytvorí znovu udalosť `decode` s parametrom `result`, zatiaľ čo neúspešné vytvorí udalosť `error` s parametrom `error` (chybová správa).

```
1 <StreamBarcodeReader @decode="onDecode" @loaded="onLoaded">  
2 </StreamBarcodeReader>  
3 <ImageBarcodeReader @decode="onDecode" @error="onError">  
4 </ImageBarcodeReader>
```

```
1 import { StreamBarcodeReader } from "vue-barcode-reader";  
2 import { ImageBarcodeReader } from "vue-barcode-reader";
```

■ **Výpis kódu 1** Importovanie komponentov *StreamBarcodeReader* a *ImageBarcodeReader* pomocou JavaScript-u a ich použitie v HTML



S cieľom dodania plánovanej funkcionality aplikácii bolo potrebné rozšíriť jednotlivé objekty, ktoré sa používali, a navrhnuť, ako sa budú správať v nových situáciách, ktoré nastanú.

### 4.1 Ingrediencie

Zatiaľ čo ingrediencie boli doposiaľ veľmi jednoduché a nemali medzi sebou žiadne vzťahy, teraz budú musieť dodržiavať stromovú štruktúru.

1. Každá ingrediencia bude preto odkazovať na jednu ingredienciu, ktorá je viac všeobecným variantom tej samej ingrediencie. Ak takáto ingrediencia neexistuje, môže byť odkaz prázdny. K tejto ingrediencii bude odkazované ako k priamo nadradenej.
2. Každá ingrediencia bude ďalej odkazovať na zoznam ingrediencií, ktoré sú viac špecifické varianty tej samej ingrediencie. Ak žiadna taká neexistuje, môže byť zoznam prázdny. K týmto ingredienciám bude odkazované ako k priamo podradeným.

Aby išlo o korektný strom, musí ďalej platiť, že vlastnosti 1 a 2 sú previazané, teda ak ingrediencia A odkazuje na ingredienciu B vo svojom zozname viac špecifických ingrediencií, bude ingrediencia B odkazovať na ingredienciu A ako na svoju viac všeobecnú ingredienciu.

3. Ingrediencie budú ďalej rozlišovať viditeľnosť, konkrétne verejnú a neverejnú viditeľnosť.

Verejná viditeľnosť, ako názov implikuje, sprístupní ingredienciu všetkým používateľom. V tomto ohľade sa bude veľmi podobáť na ingrediencie v pôvodnej verzii aplikácie. Všetci používatelia budú môcť taktiež upravovať túto ingredienciu, ako napríklad doplniť chýbajúce odkazy na rohlik.cz a kosik.cz alebo doplniť odkazy na viac všeobecnú či viac špecifické ingrediencie. Zmena vlastníka a viditeľnosti bude však povolená len aktuálnemu vlastníkovi.

Neverejná viditeľnosť sprístupní ingredienciu iba vybraným používateľom. V prípade, že vlastníkom ingrediencie je používateľ, bude ju môcť využívať ako jediný. V prípade, že je vlastníkom skupina, bude ju môcť využívať každý člen danej skupiny. Neverejné ingrediencie budú patriť do stromu ingrediencií, ale bude musieť platiť, že verejná ingrediencia nebude nikdy špecifikáciou neverejnej ingrediencie. Toto pravidlo vychádza z mienky, že ak je ingrediencia A natoľko dôležitá, že by bola v strome kategoricky vyššie ako iná ingrediencia B, ktorá je zároveň natoľko dôležitá, aby bola verejná, mala by aj ingrediencia A byť verejná.

## 4.2 Recepty

Recepty boli skoro v takom stave, aby spĺňali aj nové požiadavky. Je však potrebné ich upraviť tak, aby sa prispôbili novým ingredienciám, ktoré budú používať.

Menovite, recepty môžu využívať verejné a neverejné ingrediencie. Z pohľadu konzistentného správania je nutné zaviesť niekoľko pravidiel.

1. Každý recept smie používať verejné ingrediencie.
2. Súkromný recept vlastnený skupinou smie používať aj všetky neverejné ingrediencie vlastnené práve tou skupinou.
3. Súkromný recept vlastnený používateľom smie používať aj všetky neverejné ingrediencie, vlastnené ním alebo skupinou, ktorej je členom.

Tieto pravidlá zaručia, že každý používateľ bude vidieť iba recepty, ktoré používajú iba ingrediencie, ku ktorým má prístup.

Recepty bude možné zdieľať medzi používateľmi. Táto možnosť bude zahŕňať iba verejné a neverejné recepty. Samotné technické prevedenie bude mať informácie o zdieľaní uložené v objekte skupiny.

## 4.3 Skupiny

Rozšírené skupiny budú obsahovať dve nové veci. Prvou bude inventár, teda zoznam ingrediencií, ktoré skupina vlastní. Vlastníctvo je teraz myslené nie ako autorské práva k ingrediencii, ale ako vlastníctvo ingrediencie mimo tejto aplikácie. Dôvodom je, že jedno z plánovaných využití skupín je pre domácnosť či pre kanceláriu, kde by používatelia zdieľali zásoby. Druhou bude zoznam receptov, ktoré sú skupine zdieľané. Tieto recepty sú teda primárne rýchlymi odkazmi.

## 4.4 Inventár

Inventár bude novozavedený do aplikácie. Zámerom je odlišiť túto aplikáciu od iných konkurenčných portálov. Ako bolo spomenuté v analýze, väčšinou sú tieto portály zamýšľané výhradne na verejné zdieľanie receptov, kde používatelia hľadajú nové recepty.

## 4.5 Zmeny viditeľnosti

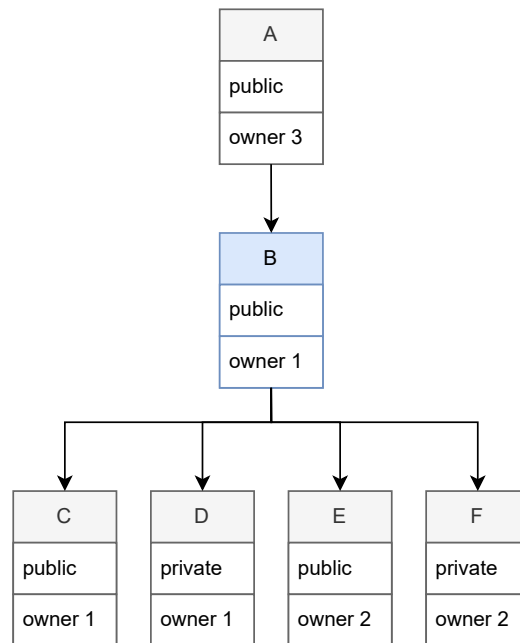
Keďže recepty aj ingrediencie sú upraviteľné po ich vytvorení, je možné zmeniť ich viditeľnosť. Prirodzene, viditeľnosť môže zmeniť iba vlastník. V prípade skupiny, jej lubovoľný člen. Tento fakt ale prináša niekoľko zaujímavých situácií, ktoré treba ošetriť.

- Vlastník zmení verejnej ingrediencii, ktorá bola použitá vo verejnom recepte, viditeľnosť na neverejnú.

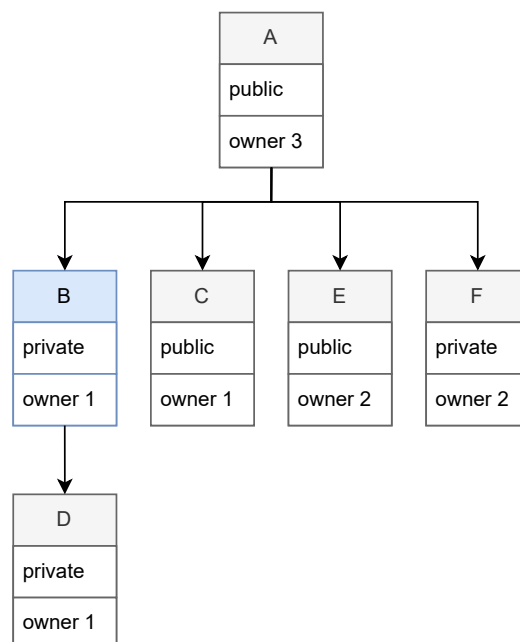
Tieto verejné recepty by teraz odkazovali na neverejnú ingredienciu

- Vlastník zmení verejnej ingrediencii, ktorá mala odkazy na špecifickejšie verejné ingrediencie alebo na neverejné ingrediencie vlastnené iným vlastníkom (ďalej len špecifikácie), viditeľnosť na neverejnú.

Tieto špecifikácie by teraz mali neverejnú ingredienciu ako viac všeobecnú. Toto by porušovalo konzistenciu stromu, a preto sa niektoré väzby zmenia. Týmto špecifikáciám sa zmení odkaz na viac všeobecnú o úroveň vyššie. Vid. obr. 4.14.2.



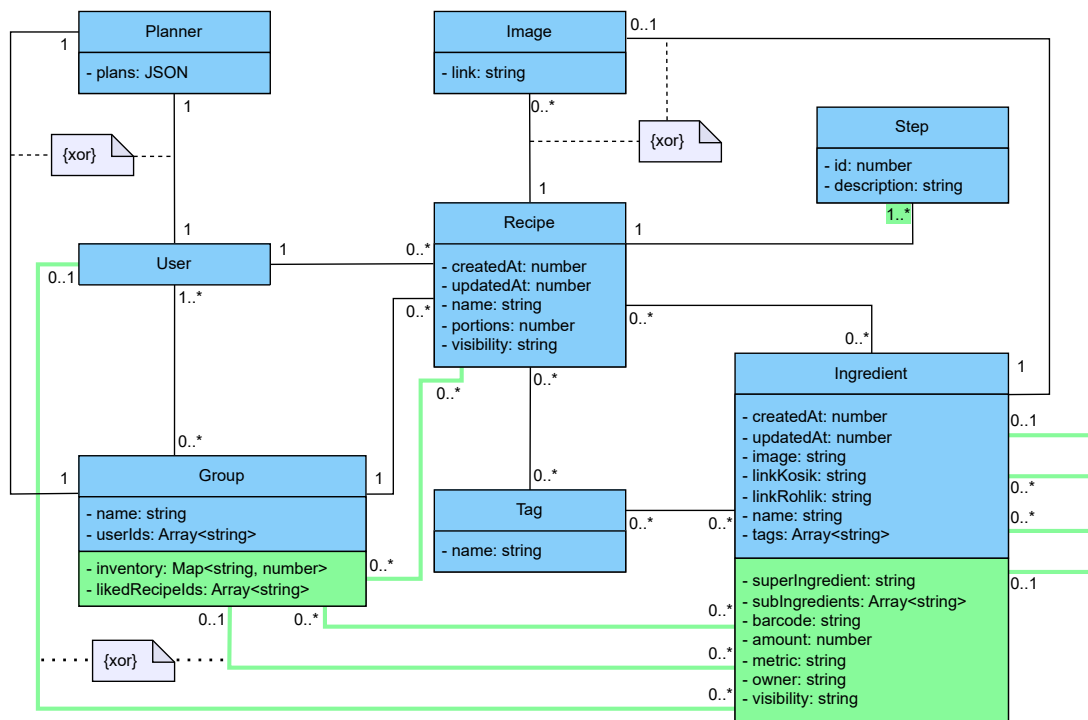
■ Obr. 4.1 Stav pred zmenou viditeľnosti



■ Obr. 4.2 Stav po zmene viditeľnosti

## 4.6 Databáza

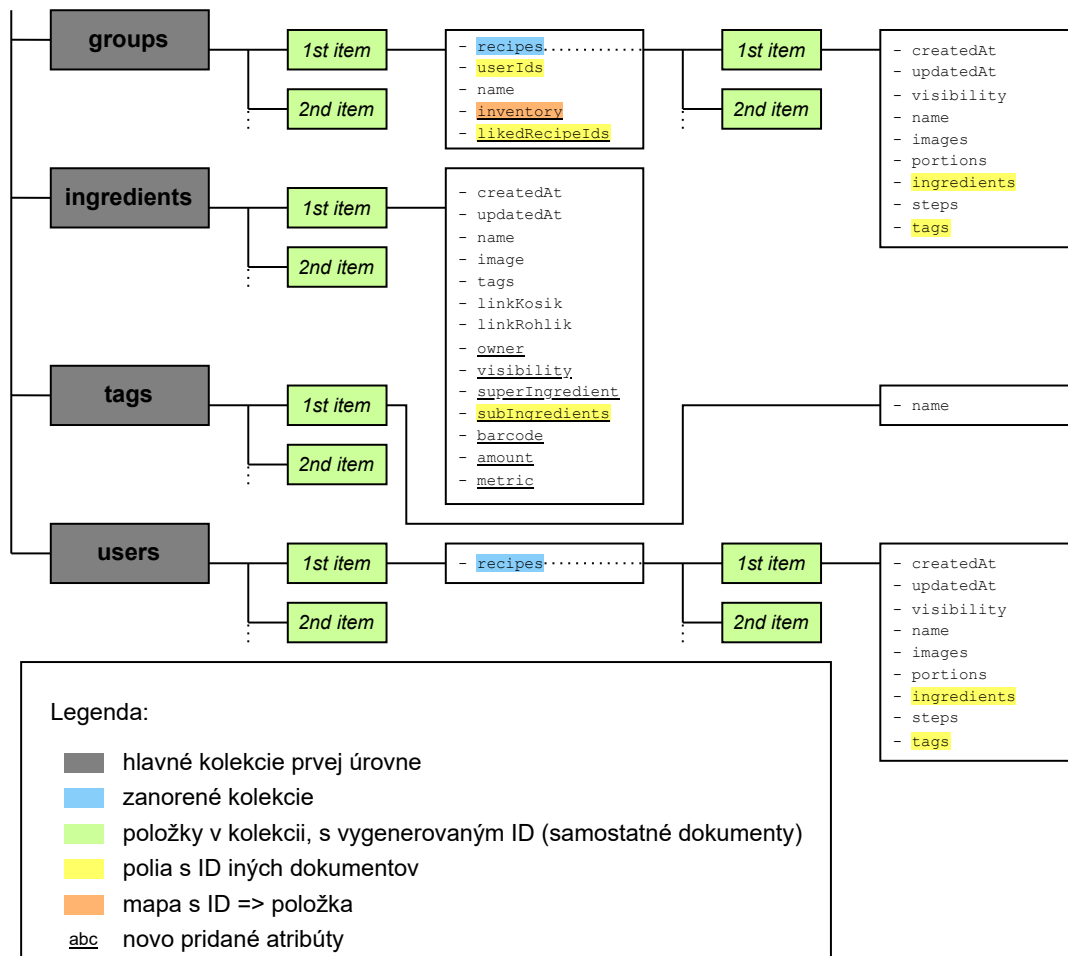
S rozšírením o funkcionality prichádza nutnosť rozšíriť databázu o položky a vzťahy, ktoré tieto funkcionality poskytujú. Vytvoril som doménový model, ktorý vychádza z doménového modelu pôvodnej práce.



■ Obr. 4.3 Rozšírený doménový model

Zelenou farbou sú vyznačené všetky zmeny. Bolo potrebné pridať nové atribúty, ktoré reprezentujú nové dáta a väzby, ktoré reprezentujú nové vzťahy.

Tento doménový model bol následne transformovaný na model dokumentovej databázy. Výsledok je popísaný nižšie.



■ Obr. 4.4 Rozšírený dokumentový model

Na rozdiel od riešenia v pôvodnej aplikácii je vlastníctvo ingrediencií urobené zatiaľ iným spôsobom. Kým u receptov je dokument receptu v kolekcií používateľa alebo skupiny, ktorej patrí, ingrediencie majú atribút, ktorý obsahuje ID vlastníka. Toto je z dôvodu otvoreného pokračovania vo funkcionalite vlastníctva ingrediencií, lebo to umožňuje jednoduché menenie vlastníka.

Motiváciou pre zmenu vlastníka ingrediencie je presunúť ingredienciu z vlastníctva jednotlivca do vlastníctva skupiny, namiesto potreby vytvorenia identickej ingrediencie. Podľa potrieb je do budúcnosti možné ingrediencie prerobiť na podobný systém, ako majú recepty, kde je vlastníkom ten, kto bol autorom. Ostatní používatelia (ako jednotlivci alebo ako zástupcovia skupiny) by vedeli tieto ingrediencie zdieľať tak, ako teraz môžu recepty.



# Implementácia

V tejto kapitole sa nachádza vstup do projektu a proces implementácie navrhnutých riešení. Spomenuté sú dôležité časti, ako aj zaujímavé problémy, ktoré v priebehu nastali.

## 5.1 Vstup do projektu

Na začiatok bolo potrebné splniť niekoľko krokov, ktoré boli nutné pre samotné vyvíjanie.

Z GitHub repozitáru bolo potrebné stiahnuť zdrojový kód. Tento kód popisuje celú aplikáciu s výnimkou jedného súboru. Ide o konfiguračný súbor Firebase, bez ktorého nie je možné pri lokálnom vývoji pripojiť aplikáciu na žiadnu z použitých služieb. Tento súbor nie je verejne dostupný, pretože umožňuje ovplyvňovať produkčnú databázu, meniť použité funkcie a jednoducho dáva plný prístup.

Tento súbor je potrebné si stiahnuť priamo z Firebase, z webového klienta po prihlásení sa do projektu. Na prihlásenie sa musí byť vývojár pozvaný vlastníkom projektu cez ich Google účet. Vlastníkom boli pôvodný autor a jeho vedúci práce.

Po prihlásení sa a stiahnutí potrebného konfiguračného súboru sú všetky potrebné dáta dostupné. Posledným krokom pred vyvíjaním je stiahnutie chýbajúcich balíčkov potrebných pre vyvíjanie v tomto prostredí. Je nutné nainštalovať *Node.js*, *npm* (Node Package Manager), *Firebase* balíčky pre vývoj a *Yarn* (Yet Another Resource Negotiator). Všetky *Firebase* balíčky je možné nainštalovať pomocou *npm*. *Yarn* sa oplatí nainštalovať len pre pohodlie vývoja (niektoré skripty pre spustenie a testovanie používajú *Yarn*). *Yarn* a *npm* sú pre väčšinu účelov zameniteľné, no pôvodný autor vytvoril inštalčný manuál pre *Yarn*.

```
1 sudo apt install nodejs
2 sudo apt install npm
3 npm install --global firebase
4 npm install --global firebase-tools
5 npm install --global firebase-functions
6 npm install --global yarn
```

■ **Výpis kódu 2** Príkazy pre stiahnutie vývojárskych balíčkov

## 5.2 Štruktúra projektu

Štruktúra projektu nepotrebovala z pohľadu zdrojového kódu v skutočnosti žiadne zmeny. Upravil bolo treba databázu, čo bolo zahrnuté v kapitole návrhu.

├── .github.....	adresář s konfiguračními soubory Githubu
├── dist.....	adresář pro sestavenou produkční verzi
├── functions.....	adresář s implementací Firebase Functions
├── node_modules.....	adresář s moduly staženými pomocí Yarn/npm
├── public.....	adresář s obrázky aplikace a základním souborem
├── src.....	zdrojové kódy
│   ├── assets.....	obrázky využité v aplikaci
│   ├── components.....	Vue komponenty
│   ├── enum	
│   ├── lang.....	překlady
│   ├── plugins.....	obaly použitých knihoven
│   ├── router.....	adresář pluginu Vue router
│   ├── service	
│   ├── store.....	adresář pluginu Vuex
│   ├── style.....	CSS soubory
│   └── views.....	Vue komponenty využívané ve Vue router pro zobrazení stránek
├── firebase.json.....	konfigurační soubor Firebase
├── firebaseConfig.js.....	tajný konfigurační soubor Firebase
└── package.json.....	soubor obsahující seznam závislostí a konfiguraci aplikace

■ Obr. 5.1 Adresárová štruktúra[1]

Takto bola štruktúra popísaná v pôvodnej práci. Pre účely aktuálneho zadania boli však zaujímavé najmä adresáre *functions*, *components*, *plugins*, *store* a *views*. Do týchto adresárov sa pridávali nové a upravovali sa existujúce súbory. Zmeny nastali aj inde, ale nie v takej výraznej miere ako vo vyššie spomenutých.

Z vizuálnej stránky UI bolo potrebné rozšíriť a pridať súbory do *views* a tým pádom, v záujme znovupoužitelnosti a prehľadnosti, boli pridané potrebné čiastkové časti do *components*.

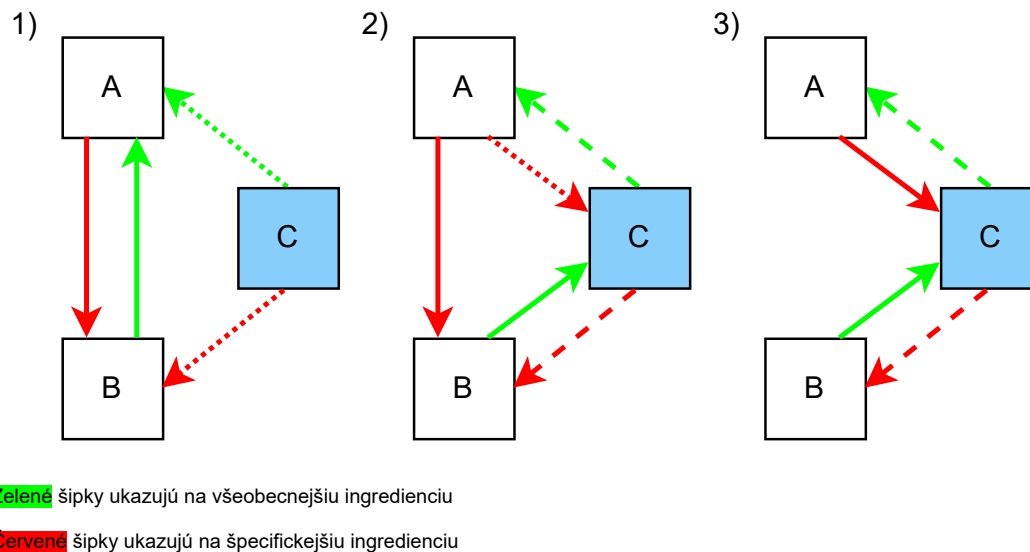
Väčšia časť práce sa však venovala práve zvyšným oblastiam, teda *functions*, kam patria funkcie, ktoré sa vykonávajú na *Firebase* serveroch a spracúvajú komplexnejšie požiadavky; *plugins*, ktorý obsahuje konfigurácie použitých balíčkov a menovite aj definície *Firestore* funkcií, teda jednoduchších funkcií na rýchle a malé čítania a zápisy, hlavne bez potreby výraznejšieho predpočítania a *store*, kam patria pravidlá pre prácu s globálnym stavom.

## 5.3 Transakcie

Prvá funkcionalita, ktorá bola implementovaná, bolo rozšírenie ingrediencií o odkazy na ich viac všeobecné a špecifické varianty. Keďže ide o obojstrannú väzbu, je dôležité dáta upravovať opatrne, aby ostala zachovaná ich konzistencia.

Presnejšie, pridanie či úprava jednej ingrediencie ovplyvňuje viacero ďalších dokumentov, a preto je tieto zmeny potrebné vykonávať atomicky. *Firebase functions* ponúkajú transakcie and tzv. *batched writes* (zápisy po dávkach).[23] *Batched writes* sú zápisy do viacerých súborov naraz, no musia byť nezávislé na ich predošlých hodnotách. Tento prípad sa však netýkal nášho problému a obr. 5.2 znázorňuje proti-príklad.





■ Obr. 5.2 Porovnanie použitia *batched writes* a *transakcie*

V časti 1 je vidno ingrediencie A, B, ktoré sú už v databáze. Ingrediencia C je nová ingrediencia, ktorej pridanie je práve v pláne.

V časti 2 sú použité *batched writes*. Väzby (C, A) a (C, B) sa dajú pridať jednoducho, a keďže všetky väzby sú obojstranné, aj väzby (A, C) a (B, C) sa dajú pridať spolu s nimi. Keďže odkaz na viac všeobecnú ingredienciu je nanajvýš jeden, bola väzba (B, A) úspešne prepísaná. Odkazov na viac špecifické ingrediencie môže byť ale viacero, a preto sa väzba (A, B) neodstránila. V tejto situácii ostala narušená štruktúra s nežiadanou extra väzbou.

V časti 3 sú použité *transakcie*. Väzby (C, A), (C, B), (A, C) a (B, C) sa dajú jednoducho pridať, podobne ako v predošlom prípade. Pred pridaním sa dal ale prečítať ich obsah. Teraz je známe, že ingrediencia B naozaj má odkaz na viac všeobecnú ingredienciu a dá sa na to reagovať. V zozname A na viac špecifické ingrediencie sa odstráni odkaz na B. Teraz ostala korektne upravená štruktúra.

Podobné a ďalšie situácie nastávajú aj pri úprave už existujúcej ingrediencie.

Použitie boli napokon transakcie. Tie dovoľujú dáta najprv prečítať a upraviť ich v závislosti na ich pôvodnom obsahu, čo presne popisuje tento problém.

```

1  return admin.firestore().runTransaction(async (transaction) => {
2      const newItem = req?.body?.data?.newItem;
3
4      let docCache
5      docCache = await addIngredientRead(transaction, newItem)
6      const itemRef = admin.firestore().collection(INGREDIENTS).doc()
7      docCache[itemRef.id] = newItem
8      docCache = await addIngredientEdit(transaction, itemRef.id, docCache)
9      await addIngredientWrite(transaction, itemRef, docCache)
10
11     return itemRef
12 }).then((ref) => {
13     return res.send({ data: { "id": ref.id, "path": ref.path } });
14 }).catch((error) => {
15     console.log("Transaction failed:", error);
16     return res.send({ data: { msg: "Fail" } });
17 });

```

### ■ Výpis kódu 3 Transakcia pre pridávanie novej ingrediencie

Ako je vidno na výpise kódu 3, s transakciami treba pracovať nasledovne:

1. vykonať všetky čítania z databázy (a lokálne ich uložiť),
2. vykonať potrebné výpočty a úpravy na lokálnej kópii dát,
3. vykonať všetky zápisy naspäť do databázy.

Ak toto poradie nie je dodržané – konkrétne vykonať všetky čítania pred prvým zápisom – je transakcia neplatná.

## 5.3.1 Testovanie transakcií

Keďže transakcie vykonávajú komplexné požiadavky, je potrebné ich otestovať. Pri ich vývoji sa na to dá použiť *Firestore emulator*. V tomto prostredí je možné spustiť napísané funkcie (ktoré využívajú spomenuté transakcie) bez toho, aby ich bolo potrebné nasadiť na produkčný server. Toto má hneď niekoľko výhod – aktuálna verzia funkcií sa neprepíše nasadením novej a nie je potrebné čakať, kým sa nasadí. Ďalej, keďže sú požiadavky počítané a riešené na lokálnom zariadení, kde sa vyvíja, má vývojár prístup ku konzole a záznamom.

V tomto momente by vývojár však stále pracoval s produkčnými dátami. Ak je toto nežiadané, je vhodné použiť *Firestore emulator*, ktorý umožňuje lokálne vytvoriť databázu rovnakej štruktúry, aká je použitá v produkcii. Na týchto dátach sa dá potom bezpečne testovať nové funkcie alebo len ich nové verzie. Nie je však nutné tento emulátor využiť. Za cenu spomalenia produkcie je možné pracovať na produkčných serveroch, ale je nutné vytvoriť nové kolekcie s testovacími dátami. Integrita dát, na rozdiel rýchlosti produkcie, nie je totižto diskutabilná.

## 5.4 Databáza

Všetky prevedené zmeny na štruktúre dát sa aplikujú na novopridané záznamy. Staré záznamy potrebujú však tiež upraviť, aby sa dali používať v novozavedených funkcionalitách. Bolo preto potrebné upraviť celú databázu. Tu sa ponúka viacero možností.

← Export

### Source

Choose the data you'd like to export. Keep in mind: you can only import the entirety of the file you export into your same project via upsert.

**Export entire database**

Acts as a full backup of your database at this point in time. Large amounts of data take longer to export.

**Export one or more collection groups**

You can choose up to 60 collection groups per export.

Choose collection group(s) ▼

### Destination

Choose a Google Cloud Storage bucket or folder to export into. Your data from project `recipeo-2e1c9` will be stored in `eur3`.

Bucket \*

BROWSE

Collections are exported in LevelDB format.



You'll be billed based on entity reads and writes, and for storage. [Learn more](#)

EXPORT

CANCEL

■ **Obr. 5.3** Firestore database – Dialóg pre exportovanie databázy Firestore. [24]

*Firestore* ponúka možnosť exportovania a importovania dát. Z *Firestore* projektu sa exportujú do *bucket-u* v *Google Cloud* úložisku. Stadiaľ je možné si ich stiahnuť a následne lokálne upravovať. Upravené dáta by sa nahrali naspäť do *Google Cloud* úložiska a stadiaľ by sa importovali do *Firestore* projektu. Tento proces je však spoplatnený podľa typu programu, do ktorého projekt patrí.

Ďalšou možnosťou je využiť to, čo je už zavedené. K dátam je prístup cez funkcie z *Firestore functions*, ktoré sú použité napríklad pri pridávaní a aktualizovaní ingrediencií. Tento istý systém sa dá použiť pre pridanie nových atribútov pre každý prvok kolekcie. Starým dátam sa pridajú všetky chýbajúce polia so základnými hodnotami. Používateľ vie tieto hodnoty zmeniť priamo v aplikácii cez aktualizáčnu funkcionality.

Druhá možnosť bola vhodnejšia pre potreby tejto aktualizácie, a preto bola použitá.

## 5.5 Skupiny

Skupinám bolo pridaných niekoľko funkcionalít. V rozšírenej verzii aplikácie môžu okrem vlastnenia receptov vykonávať nasledovné:

- Vlastníť ingrediencie. Toto je vhodné, keď existuje špecifická ingrediencia, ktorú daná skupina používa, no z ľubovoľného dôvodu ju nechcú zverejniť. Príkladom môže byť skupina kolegov,

```
1 return cors(req, res, async () => {
2   const collectionName = req?.body?.data?.collectionName
3   const newPropsArray = req?.body?.data?.newPropsArray
4   const collection = admin.firestore().collection(collectionName)
5   collection.get().then((querySnapshot) => {
6     querySnapshot.forEach((doc) => {
7       // doc.data() is never undefined for query doc snapshots
8       const newProps = {}
9       newPropsArray.forEach(item => {
10        // Select only missing data.
11        if ( !!doc.data()[item.label] ) { newProps[item.label] = item.value }
12      })
13      doc.ref.update(newProps)
14      .then(() => {
15        console.log("Document: '" + doc.id + "', successfully updated!");
16      })
17      .catch((error) => {
18        // The document probably doesn't exist.
19        // Happens if it has been removed while this function was in process.
20        console.error("Error updating document:", error);
21      });
22    });
23  });
24  return res.send({data: {msg: "Success!"}});
25 });
```

- **Výpis kódu 4** Funkcia použitá pre pridanie chýbajúcich polí so základnou hodnotou.

ktorí vytvorili skupinu s názvom „Kancelária“ a v spoločných zásobách majú „Nepál Masala - ajurvédsky čierny čaj“ [25]. Pre nich ide o dôležitú ingredienciu, ktorú používajú a chcú odlišiť od iných čajov. Túto ingredienciu majú poznamenanú v inventári skupiny. Pre iných používateľov (ale aj pre tých istých používateľov, keď práve využívajú aplikáciu ako člen inej skupiny) však ide o veľmi špecifickú ingredienciu, pre ktorú nemajú využitie. Keby bolo takýchto ingrediencií veľa, začala by byť aplikácia neprehľadnou. Prejavilo by sa to pri prezeraní všetkých ingrediencií, ako aj pri využívaní polí s dopĺňaním. Údaj o vlastníctve ingrediencie je parametrom ingrediencie.

- Označiť recept za obľúbený. Mať skupinu zhodných používateľov umožňuje zdieľanie v rovnakej komunite. Toto je funkcionálna, ktorá nebola možná v konkurenčných riešeniach (okrem KitchPal, viď. 3.1). Mať možnosť zdieľať priamo v aplikácii je veľkou výhodou pre používateľov, pretože majú všetko na tom istom mieste. Týmto spôsobom si používatelia nepotrebojú pamätať, kto a kde im recept zazdieľal, no stačí im prezrieť si ich skupiny. Následkom je častejšie používanie aplikácie Recipeo. Zdieľanie, ktoré iné riešenia ponúkali, bolo iba rýchlym odkazom na vytvorenie príspevku či správy na inej platforme.

Oblíbené recepty sú implementované ako kolekcie ich ID. K tejto kolekci je pomocou globálneho stavu prístup zovšadiaľ v aplikácii, no na jej úpravu sú použité funkcie *Firebase firestore*. Po upravení sa v prípade úspechu načítajú nové dáta zase zo serveru. Načítavajú sa selektívne v záujme šetrenia zdrojov a rýchlosti aplikácie.

- Spravovať inventár. Aplikácia Recipeo kladie dôraz na fakt, že ľudia sa pohybujú väčšinou v nízkom počte tých istých prostredí, ako je napríklad domácnosť, kancelária, klubovňa alebo majú viacero miest na skladovanie rôznych potravín, ako sú pivnica, špajza či chladnička a mať v tom prehľad je užitočné. Inventár je preto súčasťou skupín a je prístupný iba z nich. Je implementovaný ako kolekcia objektov, ktoré obsahujú ID ingrediencie a jej aktuálny počet. Počet ingrediencie sa dá upravovať po jednej jednotke, po množstve, v akom sa kupuje, ak je nastavené ako základné a dá sa nastaviť aj priamo na presnú hodnotu. Proces úpravy a načítania je obdobný ako pri obľúbených receptoch.

## 5.6 Ingredencie

Implementácia rozšírených ingrediencií sa držala návrhu. Boli pridané atribúty majiteľa, viditeľnosti, nadradenej ingrediencie, podradených ingrediencií, čiarového kódu a pre potreby inventáru množstva a metriky, v ktorom sa kupuje. Použitie v inventári bolo vysvetlené v kapitole 5.5.

Hlavná nová funkcionálna je kategorizovanie do stromovej štruktúry. Najdôležitejšou časťou bolo udržiavať konzistenciu dát pri nízkej výpočetnej náročnosti. Zvolené riešenie bolo rozdelené do dvoch častí.

Prvá bola kontrola formulára pre pridanie a úpravu ingrediencií, ktorá prebiehala na fronte. Táto časť bola rýchla, keďže nepotrebovala internetové pripojenie a nemusela preto asynchrónne čakať na odpoveď zo serveru. Musela byť však zároveň dostatočne jednoduchá, aby nespomaľovala používateľovo zariadenie. Samotná kontrola využívala *rules*, teda pravidlá na komponente formulára od *Vuetify*. Tieto pravidlá sú vlastné funkcie, ktoré sa spustia vždy pri úprave príslušného pola formulára a návratovú hodnotu majú buď **true**, teda „pravda“ v prípade, že bolo pravidlo dodržané, alebo chybovú správu, ak nie. Týchto pravidiel môže byť použitých viacero, a to aj pre to isté pole formulára.

Pravidlá použité pre udržanie konzistencie kontrolovali vzťah nadradenosti. Tento vzťah je tranzitívny a asymetrický. Kontrolované bolo konkrétne:

- že aktuálne upravovaná ingrediencia (netýka sa pridávania novej) nie je nadradená tej, ktorú označuje za svoju priamo nadradenú,

```

1 isASuperOfB (a, b) {
2   if (!a) { return false }
3   while (b) {
4     if (a === b) { return true }
5     b = this.getIngredient(b).superIngredient
6   }
7   return false
8 },

```

■ **Výpis kódu 5** Funkcia použitá pre určenie vzťahu nadradenosti dvoch ingrediencií.

- že aktuálne upravovaná ingrediencia (netýka sa pridávania novej) nie je podradená žiadnej z tých, ktoré označuje za svoje priamo podradené,
- alebo že tá ingrediencia, ktorá je označená ako priamo nadradená aktuálnej, nie je podradená žiadnej z tých, ktoré sú označené ako priamo podradené aktuálnej.

Na určenie tohoto vzťahu sa používala funkcia 5. Presný spôsob použitia sa dá nájsť v zdrojovom kóde v priloženom médiu v súbore `./src/views/IngredientAdd.vue`.

Obsahom prvej časti kontroly bolo aj použitie iba povolených ingrediencií podľa ich vlastníka a viditeľnosti. Pre toto sa použili *watchers*, teda „pozorovatelia“. Vue.js umožňuje označiť niektoré premenné ako pozorované. Vždy, keď sa zmení ich hodnota, Vue.js zavolá vlastnú metódu. Pozorovanými premennými boli atribúty vlastníka a viditeľnosti. Pri ich zmene sa funkciami 6 aktualizujú zoznamy povolených ingrediencií pre nadradenú a podradené ingrediencie. Tieto zoznamy boli, zase pomocou funkcionality *bind* od Vue.js, previazané s poliami formulára. Od tohto momentu ponúkajú polia už nové ingrediencie, ale starý výber mohol byť chybný. Funciami 7 sa tento výber prefiltruje len na tie, ktoré spĺňajú nové požiadavky.

Všetky spomenuté funkcie sú výpočetne nenáročné a celý čas majú všetky potrebné dáta. Nepotrebnú asynchrónne čakať, aby sa pri každom použití stiahli zo servera, pretože ich má aplikácia uložené v cache.

Druhou časťou bolo správne prepísanie existujúcich väzieb na nové. Toto sa dialo pomocou *Firebase functions* na *Firebase* serveri. Tieto funkcie použili vyššie spomenuté transakcie, aby sa zaručila integrita databázy (buď bude v starom stave bez novej/upravenej ingrediencie, alebo v novom korektnom stave). Ich kód sa dá nájsť v zdrojovom kóde v priloženom médiu v súbore `./functions/src/index.js`. Riadili sa nasledujúcimi krokmi:

- Funkcia pre pridanie ingrediencie:
  1. Prečítať a lokálne uložiť všetky ingrediencie, s ktorými sa bude pracovať. Patrí sem priamo nadradená novej (kategória A), priamo podradené novej (kategória B) a priamo nadradené tým z kategórie B (kategória C).
  2. Lokálne sa upraví všetky ingrediencie podľa parametrov funkcie. Najprv sa každej z kategórie C vymaže zo zoznamu podradených im príslušná ingrediencia z kategórie B. Potom sa každej z kategórie B prepíše nadradená na novopridávanú. Potom sa novopridávaná ingrediencia pridá do zoznamu podradených kategórie A.
  3. Všetky ingrediencie sa aktualizujú a novopridávanej ingrediencii sa vytvorí vlastný dokument, kam sa zapíše.
- Funkcia pre aktualizovanie ingrediencie:
  1. Prečítať a lokálne uložiť všetky ingrediencie, s ktorými sa bude pracovať. Patrí sem priamo nadradená upravovanej (kategória A1) a priamo podradené upravovanej (kategória B1).

```
1  getAvailableSuperIngredients () {
2    if (this.ingredient.visibility === PUBLIC) {
3      return this.ingredients.filter(e => e.visibility === PUBLIC)
4    }
5    else if (this.ingredient.visibility === PRIVATE) {
6      return this.ingredients.filter(e =>
7        (e.visibility === PUBLIC) ||
8        (e.owner.localeCompare(this.ingredient.owner) === 0)
9      )
10   }
11   return []
12 },
13
14  getAvailableSubIngredients () {
15    if (this.ingredient.visibility === PUBLIC) {
16      return this.ingredients.filter(e =>
17        (e.visibility === PUBLIC) ||
18        (e.owner.localeCompare(this.ingredient.owner) === 0)
19      )
20   }
21    else if (this.ingredient.visibility === PRIVATE) {
22      return this.ingredients.filter(e =>
23        (e.visibility === PRIVATE) &&
24        (e.owner.localeCompare(this.ingredient.owner) === 0)
25      )
26   }
27   return []
28 },
```

■ **Výpis kódu 6** Funkcie pre aktualizovanie zoznamov povolených ingrediencií.

```
1 watch: {
2   ingredientOwner: {
3     immediate: true,
4     handler () {
5       ingredient.superIngredient = availableSuperIngredients
6         .map(obj => obj.id)
7         .includes(ingredient.superIngredient)
8         ? ingredient.superIngredient : null
9       ingredient.subIngredients = ingredient.subIngredients
10        .filter( e => availableSubIngredients.map(obj => obj.id).includes(e) )
11     }
12   },
13 }
14
15 toggleVisibility () {
16   if (ingredientVisibility === PUBLIC) {
17     ingredient.visibility = PRIVATE
18     ingredient.subIngredients = ingredient.subIngredients
19       .filter( e => availableSubIngredients.map(obj => obj.id).includes(e) )
20   }
21   else if (ingredientVisibility === PRIVATE) {
22     ingredient.visibility = PUBLIC
23     if (ingredient.superIngredient.visibility === PRIVATE) {
24       ingredient.superIngredient = ""
25     }
26   }
27 },
```

■ **Výpis kódu 7** Funkcie pre aktualizovanie zoznamov označených ingrediencií.



Ďalej sem patrí priamo nadradená novej verzii (kategória A2), priamo podradené novej verzii (kategória B2) a priamo nadradené tým z kategórie B2 (kategória C2).

2. Lokálne sa upravujú všetky ingrediencie podľa parametrov funkcie. K úprave sa čiastočne pristupuje ako ku odstráneniu a pridaniu ingrediencie. Najprv sa kategórii A1 vymaže upravovaná ingrediencia zo zoznamu podradených. Potom sa kategória B1 rozdelí na dve časti. Kategória B1.1, tých ktoré používateľ, ktorý ingredienciu upravoval, videl. Týmto ingredienciám sa prepíše nadradená ingrediencia na A1 (využije sa tranzitivita vzťahu k získaniu najbližšej nadradenej ingrediencie). Kategória B1.2 sú ingrediencie zvyšné. O týchto nemal používateľ ako vedieť, a preto sa nemajú meniť. S kategóriami A2, B2 a C2 sa pracuje obdobne ako s kategóriami A, B a C vo funkcii pridávania ingrediencie.
3. Všetky ingrediencie sa aktualizujú, vrátane upravovanej ingrediencie.



K správne mu vyvíjaniu softvérového produktu patrí jeho testovanie a preto bol aj tento projekt testovaný. Najvhodnejšia metóda, vzhľadom k povahe projektu, je používateľské testovanie.

### 6.1 Používateľské testovanie

Používateľské testovanie prebieha dvoma spôsobmi – buď je pod dohľadom osoby, ktorá ho moderuje, a nasleduje sa scenár testovania, alebo moderované nie je a používatelia produkt iba používajú a hlásia problémy. Pre tento projekt bolo zvolené moderované testovanie so scenárom, ktorý je v plnom rozsahu v prílohe A. Scenár obsahuje kroky, ktoré využívajú nové funkcionality a jeho cieľom sú nasledujúce body:

1. Zistiť, či funkcionality fungujú.
2. Zistiť, či je dizajn pre používateľa intuitívny. Vývojári prístupujú k projektu so zadaním, ktoré nasledujú, a preto vedia, aké je plánované využitie jednotlivých funkcionalít a ako sa obsluhujú. Cieľový (nový) používateľ však natrafí na tento projekt pri vyhľadávaní nových receptov alebo od známych, ktorí ho zazdieľajú. Nebude teda vedieť, v čom je odlišný.
3. Zistiť, či dizajn funguje aj za iných podmienok, ako bol vyvíjaný – na iných zariadeniach, na iných internetových prehliadačoch, iným spôsobom prístupu. Vývojári používajú plánovaný prístup, no používateľ sa to musí naučiť iba z toho, čo mu je vysvetlené.

#### 6.1.1 Respondenti

Podľa Nielsena [26] je vhodné použiť 5 respondentov, pokiaľ existuje jedna cieľová skupina, 3-4 respondentov z oboch skupín, pokiaľ existujú dve a 3 respondentov z každej skupiny, pokiaľ ich existuje viac. Toto číslo ďalej neklesá – je potrebné mať aspoň 3 respondentov, pokiaľ chceme vylúčiť individuálny faktor nevhodne vybraného respondenta. Tento projekt je stále na svojom začiatku a teda je cieľová skupina len jedna – ľudia, ktorí varia podľa receptov a chcú ich aj vytvárať alebo zdieľať. Preto bolo zvolených 5 respondentov. Nielsen ďalej popisuje [26], ako sa pozorovaním viacerých testovaní zistilo, že 5 respondentov odhalí až 85 % chýb, pričom je výrazne menej náročné ako použiť veľkej skupiny. Ďalej je vhodné reagovať na problémy, ktoré boli nájdené, a testovanie zopakovať s novou skupinou respondentov.

V budúcnosti bude vhodné rozdeliť respondentov do viacerých skupín, a to minimálne podľa veku. Projekt Recipeo vytvára moderný a štýlový zážitok, čo často znamená minimalistický dizajn, ktorý nemusí byť pre všetkých rovnako intuitívny. Pokiaľ bude projekt Recipeo chcieť vyhovieť širšej cieľovej skupine lepšie, bude musieť vedieť reagovať na ich potreby.

## 6.1.2 Priebeh testovania

S každým respondentom prebiehalo individuálne testovanie, na ktorého začiatku dostal respondent všetky potrebné informácie. Potom bol požiadaný nasledovať scenár a moderátor sledoval jeho priebeh. Moderátor zasiahol iba v prípade, že to bolo nevyhnutné (napr. funkcionality nefungovali plánovane, ale respondent to nemal ako vedieť). Po prejení scenáru sa moderátor spýtal na názor respondenta a na jeho vstup – čo považoval za dobré alebo zlé, čo bolo máťúce a čo užitočné.

Scenár v dodatku A obsahuje pokyny pre respondenta. Skrátená verzia s popisom toho, čo si všima moderátor, je nasledovná:

1. Respondent má pridať niekoľko ingrediencií. Každá je istý typ mlieka (aby respondent vedel medzi nimi vytvoriť sémantické väzby). Moderátor sleduje, ako jednoducho respondent ingrediencie pridá (intuitívnosť dizajnu) a či využíva všetky ponúknuté funkcionality (ich užitočnosť pre používateľa).
2. Respondent má využiť funkcionality, ktoré sa rozhodol nevyužiť sám od seba. Moderátor pozoruje, či sú používané v súlade s návrhom.
3. Respondent má označiť recepty (vlastná kategorizácia) a nájsť, kde má ich prehľad. Moderátor pozoruje, ako jednoducho respondent porozumie používaniu funkcionalít.
4. Pri mobilnom testovaní má respondent zopakovať niektoré kroky z testovania s webovým rozhraním. Moderátor pozoruje, či sú všetky funkcionality rovnako dostupné (rozhranie mobilných zariadení je navzájom veľmi odlišné).
5. Po konci testu sa moderátor pýta na názor respondenta o plnení jednotlivých krokov. Toto zahŕňa problémy, nepochopený zámer, očakávania či spokojnosť s dizajnom.

## 6.1.3 Výsledky testovania

Počas testovania boli zozbierané tieto pozorovania:

1. Skupiny nie sú umiestnené intuitívne.
2. Koncept skupín užívateľov sa prvotne nestretol s porozumením. Išlo o niečo, čo respondenti neočakávali od receptárového webového portálu.
3. Hlavné vyhľadávanie (cez recepty) nie je zjavné. Respondenti ho skúšali použiť pre vyhľadávanie aj ingrediencií, skupín či nápojov.
4. Štruktúra ingrediencií nie je intuitívna. Respondenti používali nadradenú ingredienciu oveľa viac ako podradené ingrediencie, i keď ide o ten istý vzťah.
5. Funkcia viditeľnosti ingrediencií nebola napriek použitiu zaužívanej ikony pre viditeľnosť zjavná.
6. Respondenti očakávali možnosť úpravy počtu ingrediencií v inventári už na prehľade skupiny a nie až na samostatnej stránke inventára. Ďalej očakávali možnosť úpravy priamo pri kliknutí na počet a nie cez formulár.
7. Povinné políčka vo formulári pridávania ingrediencie chceli vyznačené proaktívne, nie reaktívne.
8. Automatická úprava formuláru pridávania ingrediencie, ktorá zaručuje použitie korektných údajov pri zmene iných údajov (v prípade, že sú previazané), bola neočakávaná. Tu bola naopak žiadaná reaktívna poznámka namiesto proaktívnej zmeny.

9. Skener čiarových kódov nefungoval spoľahlivo. Veľakrát fungoval hneď (plánovane), no potom (s tým istým respondentom, zariadením a kódom) viac nefungoval.

#### 6.1.4 Záver

Väčšina problémov pramení z toho, že funkcionality neboli dobre vysvetlené. Samotné popisy tlačidiel, textových polí a vybrané ikony nestačili k presnému preneseniu významu. Toto je možné vyriešiť pridaním jednoduchých poznámok. Buď budú celý čas viditeľné a jasnosť bude teda vyššia, alebo sa zobrazia pri ukázaní na dotýčny element, a tým pádom ostane dizajn naďalej minimalistický.

Aplikácia by ďalej nemala robiť nič bez toho, aby používateľa informovala. Sú operácie, ktoré vykoná aj v prípade, že si ich používateľ neželá, no musí používateľa informovať. Príkladom je úprava formulára pridávania ingrediencie, v záujme zachovania korektnosti štruktúry. Táto štruktúra sa riadi istými pravidlami (napr. ide o strom z pohľadu teórie grafov), a preto nepodporuje dve priamo nadradené ingrediencie tej istej ingrediencii.

Kvalita použitého skenera nie je konzistentná. Dôvodom môže byť použitie staršej verzie, čo by sa opravilo v priebehu plánovaného prechodu na Vue.js 3. V opačnom prípade je potrebné použiť buď iný balíček, alebo úplne iné riešenie.

Časť podnetov zozbieraných počas testovania je už zohľadnená vo výslednej verzii projektu Recipeo. Niektoré však vyžadujú prehodnotenie celkovej implementácie danej funkcionality. Príkladom je zvolená stromová štruktúra pre kategorizáciu ingrediencií. Alternatívou by bola štruktúra štítkov pre ingrediencie, v ktorých by používateľ listoval a ingrediencia by tak mohla patriť do viacerých skupín podľa viacerých označení. Tieto rozhodnutia by však menili zadanie tejto práce, a preto sú mimo jej rozsahu.



## Možnosti rozšírenia

### 7.1 Integrovanie ingrediencií do postupu receptov

Pre každý krok receptu sú dôležité len niektoré ingrediencie. Priradiť tieto ingrediencie k týmto krokom by zjednodušilo čitateľnosť textu, hlavne keď je potrebné pri varení rýchlo pozrieť, čo treba použiť. Pohodlný spôsob pre autora receptu by bol ten, ktorý používajú rôzne sociálne platformy pri označovaní členov konverzácie či priateľov. Priamo v popise kroku by používateľ napísal znak @ a za ním meno ingrediencie, ktoré predtým zapísal do zoznamu ingrediencií.

### 7.2 Rozšírenie receptu o potrebné nástroje a spotrebiče

Každý recept vyžaduje istú sadu nástrojov a spotrebičov, no nie každá kuchyňa má kompletne vybavenie. Recepty by preto mohli obsahovať ako zoznam ingrediencií, tak zoznam nástrojov. Používateľ by vedel filtrovať recepty podľa toho, čo vlastní a čo nie. Registrovaný používateľ by sa alebo skupine, ktorej je členom, vedel vytvoriť základné nastavenie týchto nástrojov, ktoré by sa automaticky aplikovalo podľa zvoleného profilu. Aplikácia by všetkým používateľom ponúkala nejaké predvoľby, ktoré by pokryli známe kombinácie spotrebičov.







## Kapitola 8

# Záver

Cieľom práce bolo rozšíriť aplikáciu Recipeo a pokročilé funkcionality pre kategorizáciu ingrediencií, zdieľanie receptov a sledovanie inventára skupín. Tieto zmeny umožnia používateľom získať viac z používania aplikácie Recipeo. Zadané požiadavky boli naplnené.

Práca sa začína analýzou pôvodnej aplikácie, konkurenčných riešení a funkcionalít, ktoré je vhodné pridať. Pokračuje návrhom, ako budú nové funkcionality fungovať a ich integrovaním do pôvodného riešenia. Ďalej popisuje implementačné kroky od vstupu do projektu cez prípravu vývojového prostredia po samotné implementovanie funkcionalít. Popisuje rôzne možnosti pre riešenie jednotlivých problémov a ktoré a prečo boli použité. Popísaný je rozsah a spôsob testovania aj s výsledkami. Práca je ukončená návrhmi budúcich rozšírení, ktoré sú inšpirované konkurenčnými riešeniami a rozhovormi s respondentmi testov, a s potenciálnymi používateľmi. V prílohe pokračuje vizuálnym porovnaním pôvodnej a rozšírenej verzie.

Aplikácia Recipeo sa posunula bližšie k ponúknutiu kompletného používateľského zážitku po doplnení o užitočné funkcionality.



## Scenár testovania

Tento dodatok obsahuje scenár testovania v takej podobe, v akej ho dostali respondenti.

### A.1 Testovanie na webovom rozhraní (veľký displej)

Nasledujúce kroky splňte po prihlásení sa.

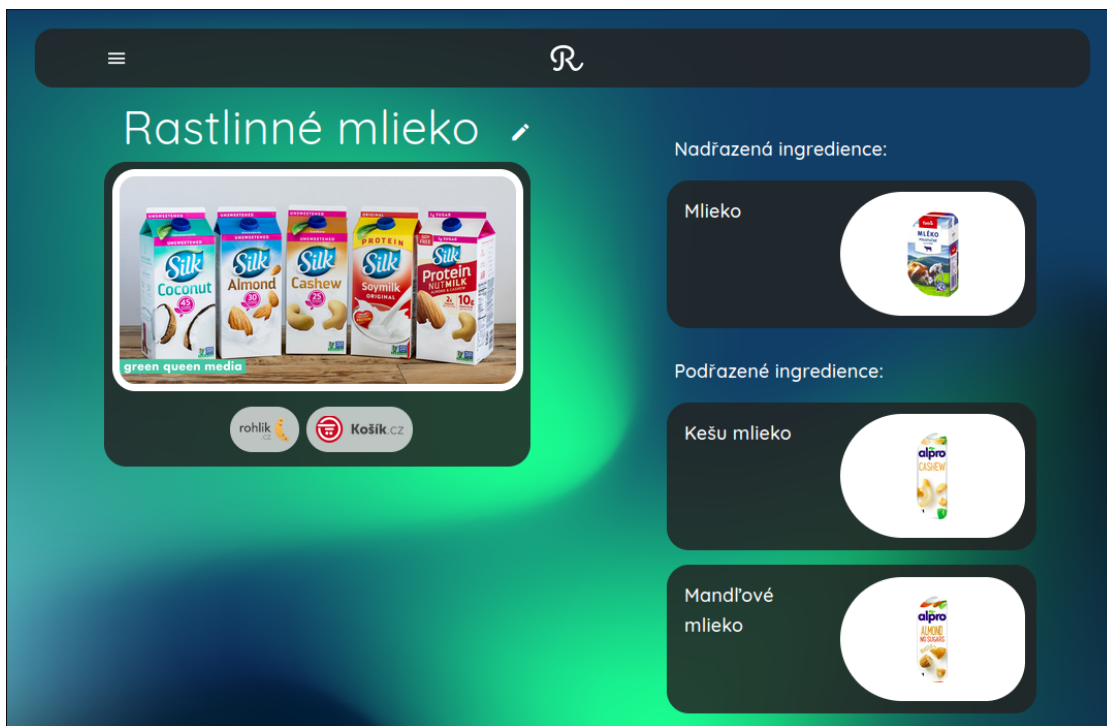
1. Pridajte nasledujúce ingrediencie v tomto poradí:
  - mlieko,
  - mandľové mlieko,
  - kešu mlieko,
  - nízkoúčné mlieko,
  - bezlaktózové mlieko,
  - mlieko v prášku,
  - bezlaktózové mlieko nízkoúčné.
2. Ak ste ich sami nevyužili, využite atribúty „Nadřazená ingrediencie“ a „Podřazené ingrediencie“, aby ste vytvorili vzťah medzi týmito typmi mlieka, s ktorým ste spokojní.
3. Ak ste tak sami neurobili a myslíte si, že je to vhodné, pridajte ďalšie typy mlieka, aby boli vzťahy jasnejšie.
4. Vytvorte dve skupiny. Pomenujte ich tak, ako by ste ich plánovali používať (takže nie „test1“, „123“ alebo „Skupina ABC“).
5. Prezerajte si recepty, ktoré existujú. Ak vás nejaký zaujme, skúste si ho nejako poznačiť. Ak vás žiaden nezaujme, prosím, aj tak si niektoré zvolte.
6. Skúste nájsť vami poznačené recepty.
7. Pridajte do inventára niektorej vašej skupiny tri ingrediencie v množstve, v ktorom by ste ich niekedy nakúpili (malý alebo veľký nákup). Ako jednu použite Tortilla wraps.
8. Zmeňte množstvo Tortilla wraps na 127 ks.

## **A.2** Testovanie na mobilnom rozhraní (malý displej)

Nasledujúce kroky splňte po prihlásení sa.

1. Pridajte ľubovoľnú ingredienciu aj s čiarovým kódom.
2. Upravte túto ingredienciu, použite všetky atribúty (obsah nemusí byť zmysluplný, ide o možnosť použitia na mobilnom rozhraní).
3. Označte ľubovoľný recept.
4. V inventári zvýšte počet niektorej ingrediencie. Pridajte novú ingredienciu do inventára. Použite čiarový kód.
5. Skúste pridať ingredienciu do inventára pomocou čiarového kódu. Vyberte takú, ktorá nie je ešte pridaná.

## Snímky z aplikácie



■ Obr. B.1 Detail ingrediencie.

The image shows a mobile application interface for adding and editing ingredients. The form is set against a dark blue background with a teal-to-green gradient. At the top, there is a navigation bar with a hamburger menu icon on the left and a logo on the right. Below the navigation bar, the form consists of several input fields and buttons:

- Název:** A text input field with the placeholder "Název ingrediencie" and a search icon. To its right is a button labeled "Přidat ingredienci".
- Majitel:** A dropdown menu with the placeholder "Vyhledávání".
- Nadřazená ingredience:** A dropdown menu with the placeholder "Vyhledávání".
- Podřazené ingredience:** A dropdown menu with the placeholder "Vyhledávání".
- Štítky:** A dropdown menu with the placeholder "Vyhledávání".
- Čárový kód:** A text input field with the placeholder "Čárový kód" and a button labeled "Skener".
- Množství:** A text input field containing the number "1" and a unit selection icon.
- Jednotky:** A dropdown menu with the placeholder "ks".
- Odkaz na rohlík.cz:** A text input field.
- Odkaz na kosík.cz:** A text input field.
- Nahrajte obrázek:** A section with a camera icon and the text "Nahrajte obrázek", followed by a horizontal line for image upload.
- Odkaz na obrázek:** A text input field.

■ Obr. B.2 Formulár pre pridanie a úpravu ingrediencie.

The image shows a mobile application interface for adding ingredients. The form is dark-themed with a teal-to-blue gradient background. At the top, there is a menu icon and a logo. Below this, the form consists of several input fields and buttons:

- Název**: A text input field with the placeholder "Název ingredience" and a "Přidat ingredienci" button to its right.
- Majitel**: A dropdown menu with the placeholder "Vyhledávání".
- Nadřazená ingredience**: A dropdown menu with the placeholder "Vyhledávání".
- Podřazené ingredience**: A dropdown menu with the placeholder "Vyhledávání".
- Štítky**: A dropdown menu with the placeholder "Vyhledávání".
- Čárový kód**: A text input field with the placeholder "Čárový k" and a "Skener" button to its right.
- Množství**: A text input field with the value "1".

A semi-transparent grey scanner overlay is positioned over the "Čárový kód" field. The scanner shows a red horizontal line across the middle of a grey square, indicating the scanning process. Below the scanner, there are several fields for links and image uploads:

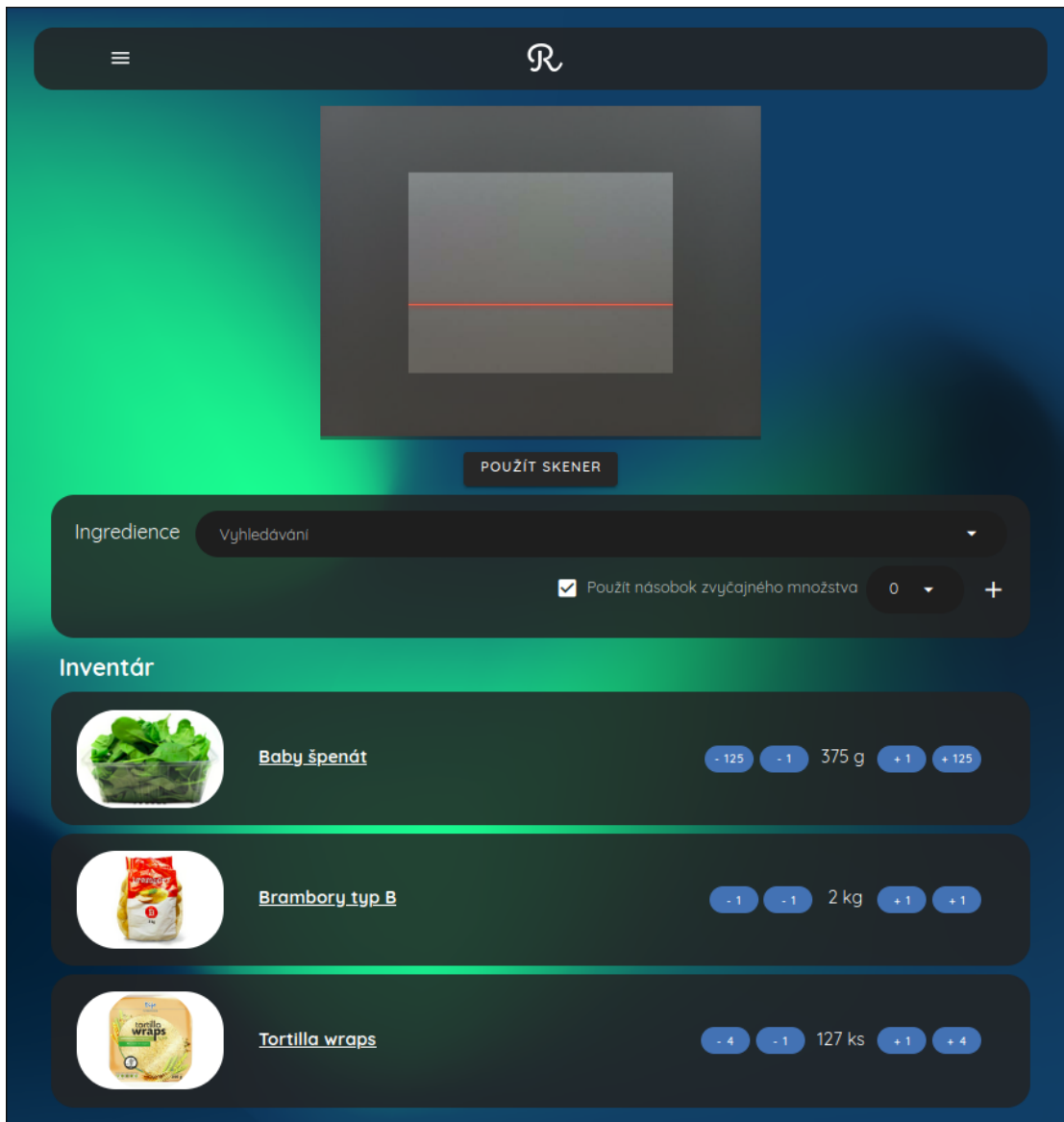
- Odkaz na rohlík.cz**: A text input field.
- Odkaz na kosík.cz**: A text input field.
- Nahrajte obrázek**: A file upload field with a camera icon and a horizontal line for the image.
- Odkaz na obrázek**: A text input field.

■ **Obr. B.3** Použití skeneru vo formuláři pre pridanie a úpravu ingrediencie.

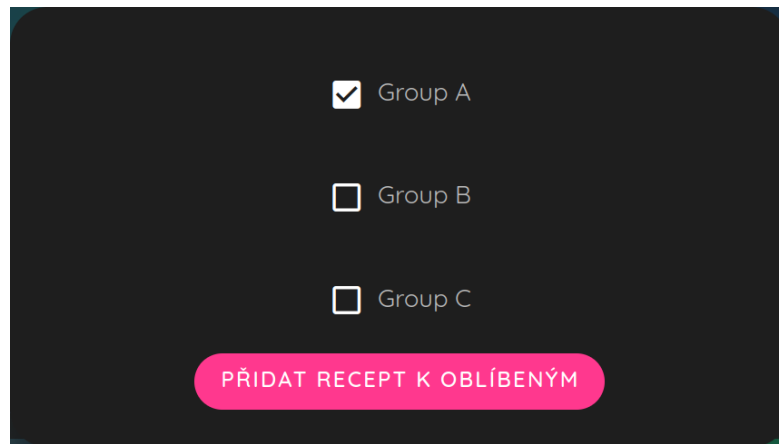


■ Obr. B.4 Detail skupiny.





■ Obr. B.5 Detail inventáru skupiny s použitím skeneru.



■ **Obr. B.6** Dialóg pre označenie receptu.

# Bibliografia

1. MORAVEC, Vojtěch. *Webová aplikace pro správu a sdílení receptů*. 2022. Dostupné tiež z: <https://dspace.cvut.cz/handle/10467/101903>. Fakulta informačních technologií ČVUT. Malec, Oldřich.
2. *Vareni.cz* [online]. Vareni.cz, 2021-06-01 [cit. 2023-03-17]. Dostupné z : <https://www.vareni.cz>.
3. *Toprecepty.cz* [online]. Toprecepty.cz, 2023 [cit. 2023-03-17]. Dostupné z : <https://www.toprecepty.cz>.
4. *Recepty.cz* [online]. Recepty.cz, 2019 [cit. 2023-03-17]. Dostupné z : <https://www.recepty.cz>.
5. *Stock and Inventory Simple* [online]. Chester Software, 2016-08-25 [cit. 2023-03-17]. Dostupné z : [https://play.google.com/store/apps/details?id=com.stockmanagement.next.app&hl=en\\_US](https://play.google.com/store/apps/details?id=com.stockmanagement.next.app&hl=en_US).
6. *Google Play* [online]. Google, 2012-03-06 [cit. 2023-03-17]. Dostupné z : <https://play.google.com/>.
7. *KITCHENPAL: Pantry Inventory* [online]. iCuisto Pte Ltd, 2016-01-31 [cit. 2023-03-17]. Dostupné z : <https://play.google.com/store/apps/details?id=fr.icuisto.icuisto>.
8. *What is a GTIN?* [online]. GS1 US, 2005 [cit. 2023-05-01]. Dostupné z : <https://www.gs1us.org/upcs-barcodes-prefixes/what-is-a-gtin>.
9. *What is a GTIN?* [online]. GS1 US, 2005 [cit. 2023-05-01]. Dostupné z : <https://www.gs1us.org/who-we-are/about-us>.
10. *EAN/UPC barcodes* [online]. GS1, 2005 [cit. 2023-05-01]. Dostupné z : <https://www.gs1.org/standards/barcodes/ean-upc>.
11. YOU, Evan. *VueJS* [online]. Vue.js, 2014-02 [cit. 2023-04-06]. Dostupné z : <https://v3.vuejs.org/>.
12. *Vuetify* [online]. Vuetify, 2016 [cit. 2023-04-06]. Dostupné z : <https://vuetifyjs.com/en/introduction/why-vuetify/>.
13. YOU, Evan. *Vuex* [online]. Vue.js, 2016 [cit. 2023-04-06]. Dostupné z : <https://vuex.vuejs.org/>.
14. YOU, Evan. *Vue Router The official Router for Vue.js* [online]. Vue.js, 2014 [cit. 2023-04-06]. Dostupné z : <https://router.vuejs.org/>.
15. *Firebase* [online]. Google, 2014-04 [cit. 2023-04-06]. Dostupné z : <https://firebase.google.com/>.

16. *Google Analytics* [online]. Google, 2005-11-14 [cit. 2023-04-06]. Dostupné z : <https://analytics.google.com/analytics/web/>.
17. *Google Ads* [online]. Google, 2000-10-23 [cit. 2023-04-06]. Dostupné z : <https://ads.google.com/home/>.
18. *Cloud Firestore* [online]. Google, 2017-10 [cit. 2023-04-06]. Dostupné z : <https://firebase.google.com/products/firestore>.
19. *Firebase Cloud Storage* [online]. Google, 2017-10 [cit. 2023-04-06]. Dostupné z : <https://firebase.google.com/docs/storage>.
20. *Firebase Hosting* [online]. Google, 2014 [cit. 2023-04-06]. Dostupné z : <https://firebase.google.com/products/hosting>.
21. *Cloud Functions for Firebase* [online]. Google, 2017-03-09 [cit. 2023-04-06]. Dostupné z : <https://firebase.google.com/products/functions>.
22. *Vue Barcode and QR code scanner* [online]. olefirenko, 2019-09-20 [cit. 2023-05-01]. Dostupné z : <https://github.com/olefirenko/vue-barcode-reader#readme>.
23. *Transactions and batched writes* [online]. Google, 2017-10 [cit. 2023-04-06]. Dostupné z : <https://firebase.google.com/docs/firestore/manage-data/transactions>.
24. *Google Cloud Firestore Export* [online]. Google, 2017-10 [cit. 2023-04-18]. Dostupné z : <https://console.cloud.google.com/firestore/databases/-default-/export>.
25. *Nepál Masala - ajurvédsky čaj* [online]. Great Tea Garden s.r.o., 2023 [cit. 2023-04-22]. Dostupné z : <https://www.cajova-zahrada.cz/nepal-masala-koreneny-caj/>.
26. NIELSEN, Jakob. *Why you only need to test with 5 users* [online]. Nielsen Norman Group, 2000-03-18 [cit. 2023-05-01]. Dostupné z : <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.

# Obsah priloženého média

	<code>readme.txt</code>	.....	stručný popis obsahu média
	<code>impl</code>	.....	zdrojové kódy implementácie
	<code>latex</code>	.....	zdrojová forma práce vo formáte $\text{\LaTeX}$
	<code>thesis.pdf</code>	.....	text práce vo formátu PDF