

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Stravování na míru



2024

Vedoucí práce:
Mgr. Jan Laštovička, Ph.D.

Vojtěch Barnet

Studijní program: Informační technologie,
prezenční forma

Bibliografické údaje

Autor: Vojtěch Barnet
Název práce: Stravování na míru
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2024
Studijní program: Informační technologie, prezenční forma
Vedoucí práce: Mgr. Jan Laštovička, Ph.D.
Počet stran: 47
Přílohy: elektronická data v úložišti katedry informatiky
Jazyk práce: český

Bibliographic info

Author: Vojtěch Barnet
Title: Tailor-made catering
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2024
Study program: Information Technologies, full-time form
Supervisor: Mgr. Jan Laštovička, Ph.D.
Page count: 47
Supplements: electronic data in the storage of department of computer science
Thesis language: Czech

Anotace

Výsledkem této práce je aplikace "Culinary PlanWise" pro platformu iOS, vyvíjená pomocí technologií React Native a Firebase. Aplikace slouží ke generování osobních jídelníčků podle individuálních potřeb uživatele. Zahrnuje sběr uživatelských údajů, na jejichž základě aplikace vypočítá nutriční složky a sestaví jídelníček. Kromě praktické implementace aplikace je součástí této práce také teoretická část, která se zabývá teoriemi výpočtu nutričních hodnot a principy tvorby jídelníčku. Závěr práce pak obsahuje poznatky a doporučení pro budoucí vylepšení aplikace a zhodnocení dosažených výsledků.

Synopsis

The result of this thesis is the "Culinary PlanWise" application for the iOS platform, developed using React Native and Firebase technologies. The application serves to generate personalized meal plans according to the individual needs of the user. It includes collecting user data, based on which the application calculates nutritional components and constructs a meal plan. In addition to the practical implementation of the application, this work also includes a theoretical section addressing theories of nutritional value calculation and principles of meal planning. The conclusion of the work contains insights and recommendations for future improvements to the application and an evaluation of the achieved results.

Klíčová slova: Aplikace; Firebase; React; React Native; Stravování; Alergie; Nutriční hodnoty

Keywords: Application; Firebase; React; React Native; Cattering; Allergies; Nutritional values

Tímto bych chtěl poděkovat Mgr. Janu Laštovičkovi, Ph.D. za vedení bakalářské práce a užitečné rady při tvorbě aplikace i textu.

Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

Obsah

1	Úvod	8
2	Stravování a vytváření jídelníčku	9
2.1	Význam kalorií a nutričních složek	9
2.1.1	Kalorie jako základní jednotka energie	9
2.1.2	Nutriční složky jídelníčku	9
2.1.3	Výpočet nutričních složek	10
2.2	Potravinová alergie, potravinová intolerance a oblíbené či neoblíbené potraviny	13
2.3	Správné rozložení jídelníčku	13
2.4	Specifické požadavky a pravidla uživatele	15
2.5	Tvorba jídelníčku	16
2.6	Příklad	16
2.6.1	Výpočet optimálního kalorického příjmu	16
2.6.2	Výpočty zbylých nutričních složek	18
2.6.3	Sestavení jídelníčku podle shromážděných podkladů	19
3	Implementace	22
3.1	Založení a zprovoznění projektu	22
3.1.1	Instalace závislostí a vývojového prostředí	22
3.1.2	Založení a spuštění projektu	23
3.2	Registrace	23
3.2.1	Použití Reactu v procesu registrace	23
3.2.2	Autentizace	24
3.2.3	Sběr uživatelských dat a specifikací pro tvorbu jídelníčku	25
3.2.4	Uložení a odeslání uživatelských dat do databáze Firestore	26
3.3	Implementace výpočtu nutričních složek a tvoření jídelníčku	27
3.3.1	Implementace výpočtu nutričních složek	27
3.3.2	Tvoření jídelníčku	27
3.3.3	Uložení a odeslání jídelníčku do databáze Firestore	32
3.4	Organizace kódu	33
3.5	Uživatelské rozhraní	34
3.5.1	Navigace a obrazovky	34
3.5.2	Tvorba vlastních komponent pomocí technologie React Native	35
3.5.3	Stylizace pomocí vlastních stylů	37
3.5.4	Ukázka finálního uživatelského prostředí	38
4	Zhodnocení	41
4.1	Hodnocení vytvořené aplikace	41
4.2	Možná vylepšení	41
	Závěr	43

Conclusions	44
A Obsah elektronických dat	45
Literatura	46

Seznam tabulek

1	Koeficienty denní aktivity	11
2	Koeficienty pohybových aktivity	11
3	Seznam alergenů	14
4	Průměrné procentuální rozdělení kalorií pro starší děti a dospělé .	15
5	Průměrné procentuální rozdělení kalorií pro nejmladší a děti . . .	16
6	Tabulka na zápis podkladů pro tvorbu jídelníčku	17
7	Tabulka Šárčiných podkladů pro tvorbu jídelníčku	19
8	Šárky procentuální rozložení kalorií a nutričních hodnot během dne	20
9	Součet nutričních složek ve vytvořeném jídelníčku pro Šárku . . .	20
10	Pravidla	28

1 Úvod

V dnešní době, kdy se čas stává stále vzácnějším, lidé aktivně hledají způsoby, jak co nejefektivněji využít svůj čas. S narůstajícím problémem obezity je stále důležitější, aby lidé měli povědomí o svém stravování a zajišťovali, že jejich jídelníček odpovídá individuálním potřebám výživy.

Na základě těchto výzev a zároveň s cílem seznámit se s technologiemi React Native a Firebase jsem se rozhodl zaměřit svoji bakalářskou práci na vývoj stravovací aplikace pomocí těchto technologií. Aplikace nazvaná "Culinary PlanWise", je určena pro platformu iOS a má za cíl generovat jídelníčky, které pokryjí denní nutriční potřeby uživatele po dobu celého týdne.

Práce je rozložena na tři hlavní části:

- Teoretická část: Je věnována teorii spojené s výpočtem nutričních složek a principů tvorby jídelníčku v dnešním světě. Zahrnuje zobrazení současných metod a postupů, včetně ukázky na konkrétním příkladu.
- Implementace: Druhá část práce se zaměřuje na implementaci těchto teoretických poznatků do praktické aplikace. Popisuje proces vývoje aplikace a použité technologie, které umožňují aplikaci efektivně vypočítávat nutriční složky a generovat personalizované jídelníčky.
- Závěr práce: V závěrečné části jsou shrnuty dosažené výsledky a možné vylepšení aplikace. Zároveň je zobrazeno zhodnocení dosažených výsledků a jejich přínos pro uživatele.

2 Stravování a vytváření jídelníčku

Tato kapitola se důkladněji zaměřuje na proces tvorby osobního jídelníčku a související stravovací praktiky. Přináší kompletní řadu uživatelských informací nezbytných pro tento proces, včetně metodiky, kterou je potřeba následovat při vytváření vyváženého jídelníčku, odpovídajícího konkrétním potřebám a cílům jednotlivce. Většina informací v této kapitole je čerpána z [1][2].

2.1 Význam kalorií a nutričních složek

Předtím, než se pustíme do tvorby konkrétního jídelníčku, je důležité pochopit význam kalorií a dalších nutričních složek v naší stravě.

2.1.1 Kalorie jako základní jednotka energie

Kalorie jsou jednotkou energie, kterou obdržíme z potravy a nápojů. Kalorie tedy poskytují tělu energii potřebnou k pohybu, udržení teploty, růstu, myšlení a dalším životně důležitým funkcím. Běžně se hovoří o kaloriích, ale ve skutečnosti se jedná o *kilokalorie* (kcal). Tato zkratka se ustálila v běžném jazyce, přestože přesněji odpovídá pojmu kilokalorie.

2.1.2 Nutriční složky jídelníčku

Kromě kalorií je důležité dbát na rovnováhu dalších nutričních složek, jako jsou sacharidy, tuky, bílkoviny a vláknina.

Sacharidy: Jsou hlavním zdrojem energie pro naše tělo a měly by být součástí každého jídelníčku v rovnoměrném množství. Nejvíce zdravých sacharidů nalezneme například v ovoci (jako jsou jablka, banány), celozrnných obilovinách (jako je hnědá rýže, oves) a luštěninách (jako jsou fazole, čočka).

Tuky: Jsou důležité pro správné fungování buněk a mohou ovlivnit náladu a psychickou pohodu jedince. Zároveň přidávají jídlu specifickou chuť. Proto nám většinou více chutnají nezdravá jídla. Nedostatečný nebo nadměrný příjem tuků může přinášet různá zdravotní rizika. Nejvíce zdravých tuků nalezneme například v olivovém oleji, avokádu, ořechích a rybách bohatých na omega-3 mastné kyseliny.

Bílkoviny: Jsou stavebním kamenem svalů a tkání nezbytné pro opravu a regeneraci těla. Nejvíce bílkovin nalezneme například v kuřecím mase, rybách, luštěninách a mléčných výrobcích.

Vláknina: Je důležitá pro zdravé trávení a přispívá k pocitu sytosti po jídle. Nejvíce vlákniny nalezneme například v celozrnných obilovinách, luštěninách, ovoci a zelenině.

2.1.3 Výpočet nutričních složek

Pro výpočet nutričních složek na denní bázi je nezbytná znalost specifických informací od člověka, pro kterého výpočet děláme. Mezi tyto údaje patří věk, hmotnost, výška, pohlaví, úroveň fyzické aktivity a cíl týkající se hmotnosti, tj. zda uživatel chce zhubnout, udržet současnou váhu nebo nabrat na váze. Tyto informace jsou klíčové pro stanovení přesných hodnot jednotlivých živin a kalorií, které tělo každodenně potřebuje.

Kalorie Pro výpočet *optimálního kalorického příjmu* (OKP) je nejprve nutné zjistit *bazální metabolismus* (BMR), což je množství energie, které tělo potřebuje k udržení životních funkcí v klidovém stavu. Pro výpočet BMR se nejčastěji využívá buď *Mifflin-St. Jeorova* rovnice [4] a nebo *Katch-McArdlova* rovnice [3]. Obě jsou udávány v kilokaloriích(kcal).

Mifflin-St. Jeorova rovnice je dána vztahem

$$BMR = 10 \cdot m + 6,25 \cdot h - 5 \cdot a + c,$$

kde

- m je hmotnost člověka v kilogramech,
- h je výška člověka v centimetrech,
- a je věk člověka v letech,
- c je konstanta, která je specifická pro každou pohlavní skupinu.

Pro muže: 5 a pro ženy: -161

Například pro ženu s hmotností 80 kg, výškou 180 cm a věkem 27 let by **BMR** bylo **1629 kcal**.

Katch-McArdlova rovnice je dána vztahem

$$BMR = c + 21,6 \cdot LBM,$$

kde

- c je konstanta, která je stanovená na 370,
- LBM je Lean Body Mass, neboli hmotnost člověka bez tuku v kilogramech. Pro každou pohlavní skupinu se počítá rozlišně.

Pro muže: $LBM = 0,407 \cdot váha + 0,267 \cdot výška - 19,2$

Pro ženy: $LBM = 0,252 \cdot váha + 0,473 \cdot výška - 48,3$

Tabulka 1: Koeficienty denní aktivity

Typ aktivity	k_1	Popis
Nízká	1,2–1,4	Sedavá práce a životní styl s minimem pohybu během dne
Lehká	1,4–1,6	Lehce aktivní zaměstnání s převahou sedavé práce a občasným pohybem a rutinními domácími pracemi
Střední	1,6–1,7	Středně aktivní zaměstnání s převahou práce v pohybu.
Vysoká	1,7–2,4	Velmi aktivní zaměstnání s převahou manuální práce během celého dne.

Zdroj: gymbeam.cz [1]

Tabulka 2: Koeficienty pohybových aktivity

k_2	Popis
0,2–0,3	Silové sporty (silový trénink v posilovně, kruhový trénink v posilovně, crossfit, cvičení s vlastní vahou, street workout)
0,2–0,45	Kolektivní a raketové sporty (hokej, fotbal, volejbal, basketbal, florbal, futsal, tenis, squash, stolní tenis)
0,2–0,5	Vytrvalostní sporty (chůze, běh, plavání, jízda na kole, veslování)
0,3–0,4	Intenzivní skupinové lekce (aerobic, kruhový trénink, TRX, Body Pump)

Zdroj: gymbeam.cz [1]

Aktivita $OKP = BMR \cdot k$

Dále se BMR vynásobí koeficientem aktivity (k), který určuje množství energie, které se využije při každodenních normálních, tak i sportovních aktivitách. Koeficient aktivity je dán takto:

$$k = k_1 + k_2$$

kde

- k_1 je koeficient denních aktivit daný tabulkou 1,
- k_2 je koeficient pro hodinu pohybové aktivity daný tabulkou 2.

Pokud tedy chceme spočítat přesné k_2 , musíme všechny koeficienty za naše pohybové aktivity během týdne, vydělit počtem dnů v týdnu. Například pro ženu, která chodí třikrát týdně na hodinu na kruhový trénink do posilovny a dvakrát v týdnu si jde na hodinu zaplavat bude výpočet vypadat takto:

$$k_2 = \frac{3 \cdot 0,25 + 2 \cdot 0,3}{7}$$

Změna váhy $OKP = BMR \cdot k \cdot w$

Pokud ale chceme například hubnout nebo nabírat, musíme ještě přidat koeficient hubnutí (w).

- **Hubnutí:** $w = 0,85$
- **Nabírání:** $w = 1,15$

Bílkoviny Doporučený rozsah denního příjmu bílkovin (DPB) se pohybuje od 0,8 do 2 gramů bílkovin (p) na kilogram tělesné hmotnosti (m) denně. Dolní hranice tohoto rozmezí je určena pro jedince s minimální fyzickou aktivitou a sedavým životním stylem, zatímco horní hranice je relevantní především pro sportovce, zejména ty, kteří se věnují silovému tréninku a budování svalové hmoty. Pro jedince se sportovní aktivitou je doporučeno, aby jejich denní příjem bílkovin dosahoval zhruba od 1,4 do 2 gramů bílkoviny na kilogram tělesné hmotnosti. Takže pro člověka, který navštíví posilovnu třikrát týdně, by vyhovovalo 1,8 gramů na kilogram jeho tělesné váhy. Výsledek je udán v gramech, pro zjištění kcal musíme výsledek vynásobit čtyřkou, protože jeden gram bílkoviny obsahuje čtyři kcal [1].

Výpočet DPB: $DPB = m \cdot p$

Tuky Doporučený příjem tuků (DPT) se obvykle pohybuje v rozmezí 20–30 % OKP, který je udáván v kcal. V rovnici je to názorně ilustrováno s průměrnou hodnotou což je 25 %. Pro přepočítání na gramy je třeba výsledek vydělit devíti, protože jeden gram tuku obsahuje devět kcal.

Výpočet DPT: $DPT = \frac{OKP \cdot 0,25}{9}$

Sacharidy Výpočet *doporučeného příjmu sacharidů* (DPS) je relativně jednoduchý a spočívá v odčítání příjmu bílkovin a tuků od OKP. Výsledek je udán v kcal, a pro získání výsledného počtu gramů, je třeba výsledek vydělit čtyřmi, protože jeden gram sacharidů obsahuje čtyři kcal.

Výpočet DPS: $DPS = \frac{OKP - DPB - DPT}{4}$

Vláknina *Denní příjem vlákniny* (DPV) je předem stanovený a nemusíme ho dopočítávat. Pro dospělého člověka by měl být kolem 30 gramů.

2.2 Potravinová alergie, potravinová intolerance a oblíbené či neoblíbené potraviny

Potravinová alergie je reakce imunitního systému na určité složky potravy, která může mít vážné důsledky na zdraví jedince. Existuje několik mechanismů, kterými se tyto nežádoucí reakce projevují, přičemž imunitně zprostředkované reakce jsou nejběžnější a nejzávažnější.

Potravinová intolerance je potíže s trávením určitých potravin, často spojená s nepříjemnými tělesnými příznaky, jako je nadýmání, bolest břicha, bolest hlavy nebo únava.

Tyto dva pojmy postihují miliony lidí a mají významný dopad na jejich životy a zdraví. V důsledku toho bylo nezbytné vyvinout systém, který by usnadnil rozpoznání alergenů v potravinách. Proto v členských státech Evropské unie od 12. prosince 2011 platí předpis o označování potravin. Účelem je ochrana lidí trpících potravinovou alergií nebo intolerancí. Podle tohoto nařízení je vyžadováno, aby všechny přísady potravin, které jsou častými spouštěči alergií, byly uvedeny, i pokud jsou přítomny v malém množství. Proto vznikl list 14 alergenů (14 látek nebo skupin látek), podle kterého se potraviny musí označovat. Tento list je udáván v této tabulce 3 [5][6].

Oblíbené a neoblíbené potraviny: Určení oblíbených a neoblíbených potravin má značný vliv na uživatelův přístup k jídelníčku. Zohlednění těchto preferencí může vytvoření jídelníčku zpříjemnit a zlepšit stravovací zážitek. Naopak ignorování těchto preferencí může vést k nespokojenosti uživatele.

2.3 Správné rozložení jídelníčku

Počet jídel za den: Rozhodnutí o optimálním počtu jídel za den je často ovlivněno mnoha faktory, jako jsou individuální preference, životní styl nebo zdravotní stav. Diskuze o tom, zda je vhodnější jíst častěji nebo méně často, je provázena rozmanitými názory a studii. Někteří odborníci zdůrazňují výhody častějšího stravování, zatímco jiní preferují konzumaci menšího počtu větších jídel. Nedávný výzkum naznačuje, že frekvence jídel nemá významný vliv na celkovou energetickou spotřebu nebo schopnost hubnutí. Důležité je tedy, co vyhovuje konkrétnímu jedinci [7][8]. Nicméně je třeba zmínit, že nejběžnějším modelem stravování bývá čtyři až pět jídel denně, což je často považováno za vyvážený přístup k udržení optimálního metabolismu a potravinového příjmu [9].

Tabulka 3: Seznam alergenů

Název	Číslo	Popis
Obiloviny obsahující lepek	1	Pšenice, žito, ječmen, oves, špalda, kamut nebo jejich hybridní odrůdy a výrobky z nich.
Korýši	2	Korýši a výrobky z nich.
Veje	3	Veje a výrobky z nich.
Ryby	4	Ryby a výrobky z nich.
Podzemnice olejná (arašídy)	5	Podzemnice olejná a výrobky z ní.
Sójové boby (sója)	6	Sójové boby a výrobky z nich.
Mléko	7	Mléko a výrobky z něj.
Skořápkové plody	8	Mandle, lískové ořechy, vlašské ořechy, kešu ořechy, pekanové ořechy, para ořechy, pistácie, makadamie a výrobky z nich.
Celer	9	Celer a výrobky z něj.
Hořčice	10	Hořčice a výrobky z ní.
Sezamová semena (sezam)	11	Sezamová semena (sezam) a výrobky z nich.
Oxid siřičitý a siřičitany	12	Oxid siřičitý a siřičitany v koncentracích vyšších 10 mg, ml/kg, l, vyjádřeno SO ₂
Vlčí bob (lupina)	13	Vlčí bob (lupina) a výrobky z něj.
Měkkýši	14	Měkkýši a výrobky z nich.

Zdroj: NZIP.CZ [5]

Procentuální rozložení kalorií: Analýza provedená v rámci National Health and Nutrition Examination Survey (NHANES) 2003–2004, poskytuje důležité poznatky o rozložení kalorií během dne v případě, že se rozhodneme pro tři hlavní jídla a zbytek rozdělíme do svačin. Starší děti i dospělí průměrně spotřebují přibližně 22 % svého celkového denního příjmu kalorií při snídani, 31 % při obědě a 35 % při večeři, zatímco zbytek je vyčleněn na svačiny. Hodnoty udává tato tabulka 4. Zajímavé je, že nejmladší děti, ve věku 1–4 let, konzumují nižší procento svého denního příjmu během hlavních jídel a vyšší procento při svačinách. Hodnoty udává tabulka 5. Trend tedy odráží skutečnost, že děti v tomto věkovém rozmezí mají tendenci jíst častěji během dne. Starší děti a dospělí mají zase tendenci mít podobné procentuální rozložení kalorií během hlavních jídel, s menším důrazem na svačiny. Tato analýza poskytuje užitečné informace pro navrhování vyvážených jídelních plánů, které odpovídají potřebám jednotlivých věkových skupin. Zohledněním procentuálního rozložení kalorií lze efektivněji podpořit zdravou stravu a životní styl [10].

2.4 Specifické požadavky a pravidla uživatele

Každý jedinec je jedinečný a může mít individuální potřeby a omezení, které je potřeba v jídelníčku zohlednit. Například osoba trpící diabetem potřebuje jídelní plán s nízkým obsahem cukru a sacharidů. Zatímco jedinec, který pravidelně konzumuje práškový protein po cvičení, může preferovat méně bílkovin v normální stravě. Vegetarián může potřebovat alternativní zdroje živin, jako jsou rostlinné bílkoviny a vyhýbat se potravinám živočišného původu. Existuje mnoho dalších specifických potřeb a pravidel, která mohou ovlivnit jídelníček jednotlivce, a proto je velmi důležité vytvářet jídelní plány na míru. Tímto způsobem lze zajistit, že stravovací režim odpovídá individuálním potřebám a cílům uživatele, což přispívá k dosažení optimálního zdravotního a výživového cíle. Díky osobnímu přístupu lze maximalizovat účinnost stravovacích plánů a minimalizovat riziko nežádoucích účinků spojených s nedostatečně přizpůsobenou dietou.

Tabulka 4: Průměrné procentuální rozdělení kalorií pro starší děti a dospělé

	14–18 Let		19–59 Let		>60 Let	
Typ jídla	Muži	Ženy	Muži	Ženy	Muži	Ženy
Snídaně	22 %	22 %	21 %	20 %	24 %	23 %
Oběd	34 %	35 %	33 %	34 %	33 %	33 %
Večeře	39 %	39 %	40 %	39 %	41 %	39 %
Svačina	5 %	4 %	6 %	7 %	2 %	5 %

Zdroj: Child and Adult Care Food Program: Aligning Dietary Guidance for All. [10]

Tabulka 5: Průměrné procentuální rozdělení kalorií pro nejmladší a děti

	1 Rok	2–4 Roky	5–10 Let		11–13 Let	
			Muži	Ženy	Muži	Ženy
Snídaně	20 %	20 %	22 %	21 %	24 %	22 %
Oběd	22 %	27 %	32 %	31 %	30 %	31 %
Večeře	22 %	27 %	32 %	31 %	35 %	37 %
Svačina	36 %	26 %	14 %	17 %	11 %	10 %

Zdroj: Child and Adult Care Food Program: Aligning Dietary Guidance for All. [10]

2.5 Tvorba jídelníčku

Proces tvorby jídelníčku začíná shromážděním všech potřebných informací nezbytných pro jeho sestavení. Ty jsou detailněji popsány v předchozích částech [Význam kalorií a nutričních složek](#), [Potravinová alergie, potravinová intolerance a oblíbené či neoblíbené potraviny](#), [Správné rozložení jídelníčku](#) a [Specifické požadavky a pravidla uživatele](#). Tyto údaje budou zaznamenány do tabulky, která poskytne základ pro sestavení jídelníčku pro danou osobu. Obecně tabulka pro záznam podkladů vypadá takto 6. Za pomoci tabulky budeme postupně vybírat jídla, která splňují stanovená kritéria. Proces výběru jídel probíhá od snídaně přes oběd a večeři až po svačiny. Pokud všechna vybraná jídla odpovídají požadovaným nutričním hodnotám, je jídelníček považován za správně sestavený. Tento proces je ilustrován na příkladu v následující sekci.

2.6 Příklad

Předpokládejme, že máme k dispozici informace o fiktivní osobě jménem Šárka, kterou je 28letá žena. Měří 170 cm a váží 65 kg. Šárka má za cíl udržet si váhu a aktivně se zapojuje do cvičení. Pracuje v kanceláři jako administrativní pracovnice a většinu svého pracovního času tráví v sedě za počítačem. Navštěvuje posilovnu třikrát týdně po dobu jedné hodiny a dvakrát týdně plave také po dobu jedné hodiny. Je důležité zdůraznit, že Šárka má alergii na ryby a nechutná jí červená řepa. Šárka by ráda jídelníček rozložila do pěti jídel denně.

2.6.1 Výpočet optimálního kalorického příjmu

Pro výpočet optimálního kalorického příjmu (OKP) a nutričních složek je nejprve nezbytné provést výpočty pomocí rovnic uvedených v sekci [Kalorie](#).

Tabulka 6: Tabulka na zápis podkladů pro tvorbu jídelníčku

Základní údaje	
Věk	x let
Výška	x cm
Hmotnost	x kg
Cíl	
Alergie	
Oblíbená jídla	
Neoblíbená jídla	
Počet jídel denně	
Aktivita	
Pracovní náplň	
Sport	
Nutriční hodnoty (kcal/g)	
Kalorie	x kcal
Bílkoviny	x kcal
Tuky	x kcal
Sacharidy	x kcal
Vláknina	x g

Výpočet bazálního metabolismu: Klíčovým faktorem je bazální metabolismus (BMR), který je základem pro stanovení OKP. Pro výpočet BMR je třeba obrátit se na [Mifflin-St. Jeorova rovnice](#), protože Šárka neposkytla přesnou hodnotu její tělesné hmotnosti bez tuku, která by umožnila využití [Katch-McArdlova rovnice](#).

$$BMR = 10 \cdot m + 6,25 \cdot h - 5 \cdot a + c,$$

kde

- m je hmotnost subjektu v kilogramech, což v případě Šárky činí 65,
- h je výška subjektu v centimetrech, což je 170,
- a je věk subjektu v letech, Šárce je 28,
- c je konstanta, která je -161 , protože Šárka je ženského pohlaví.

Dosazením hodnot do rovnice získáme:

$$BMR = 10 \cdot 65 + 6,25 \cdot 170 - 5 \cdot 28 - 161 = 1411,5 \text{ kcal.}$$

Výpočet koeficientu aktivity: Následně potřebujeme určit koeficient aktivity (k), který se skládá z koeficientu denních aktivit (k_1) a koeficientu pro hodinu pohybové aktivity (k_2). Jak je uvedeno zde [2.1.3](#).

Pro k_1 vycházíme, že Šárka většinu času tráví v kanceláři, což naznačuje sedavý životní styl s minimální fyzickou aktivitou v pracovní době. Určíme k_1 na hodnotu 1,3 za pomoci tabulky 1.

Pro výpočet k_2 vezmeme v úvahu Šárčiny sportovní aktivity, které zahrnují trénink v posilovně a plavání. Šárce tedy přiřadíme hodnotu 0,25 pro jeden trénink v posilovně a 0,30 pro plavání z tabulky 2. Průměrný denní koeficient sportovní aktivity poté vypočítáme jako

$$k_2 = \frac{\text{počet} \cdot \text{sportovní aktivita}}{7} = \frac{3 \cdot 0,25 + 2 \cdot 0,30}{7} = 0,19$$

Koeficient pohybové aktivity k dopočítáme jako součet k_1 a k_2 , což dává

$$k = k_1 + k_2 = 1,3 + 0,19 = 1,49$$

Zjištění konstanty pro změnu váhy: Poslední částí je zvolení konstanty w , která představuje hodnotu pro změnu váhy, jak je vysvětleno v sekci 2.1.3. Šárka má za cíl udržet si svoji váhu, což znamená, že hodnota w je 1.

Teď už stačí tyto tři hodnoty dosadit do rovnice a máme výsledek. Výsledek je udáván v kcal. Zde dosazená rovnice:

$$OKP = BMR \cdot k \cdot w = 1411,5 \cdot 1,49 \cdot 1 = 2103,1 \text{ kcal}$$

2.6.2 Výpočty zbylých nutričních složek

Nyní budeme pokračovat ve výpočtech dalších nutričních složek pro Šárku jako jsou bílkoviny, sacharidy, tuky a vláknina. Využijeme k tomu znalosti z těchto předchozích sekcí [Význam kalorií a nutričních složek](#) a [Výpočet nutričních složek](#).

Výpočet bílkovin: Pro výpočet denního příjmu bílkovin (DPB) budeme potřebovat vzorec, který je zmíněn v sekci [Bílkoviny](#). Šárka má sice sedavou práci, ale poměrně dost sportuje a věnuje se i silovému tréninku. Proto ji zvolíme p jako 1,5 gramů. Po dosažení hodnot do vzorce dostaneme:

$$DPB = m \cdot p = 65 \cdot 1,5 = 97,5 \text{ g}$$

Výpočet tuků: Pro výpočet doporučeného příjmu tuků (DPT) pro Šárku využijeme vzorec uvedený v sekci [Tuky](#). Po dosažení do vzorce získáme:

$$DPT = \frac{OKP \cdot 0,25}{9} = \frac{2103,1 \cdot 0,25}{9} = 58,4 \text{ g}$$

Výpočet sacharidů: Pro výpočet doporučeného příjmu sacharidů (DPS) pro Šárku využijeme vzorec, který je zmíněn v sekci [Sacharidy](#). Po dosazení do vzorce získáme:

$$DPS = \frac{OKP - (DPB \cdot 4) - (DPT \cdot 9)}{4} = \frac{2103,1 - 390 - 525,6}{4} = 296,9 \text{ g}$$

Vláknina: Denní příjem vlákniny (DPV) pro Šárku není potřeba dopočítávat, jelikož je předem stanovený na přibližně 30 gramů pro dospělého člověka. Jak je psáno zde [Vláknina](#).

2.6.3 Sestavení jídelníčku podle shromážděných podkladů

Nyní máme k dispozici kompletní tabulku podkladů pro tvorbu Šárčina jídelníčku. Můžeme začít sestavovat její denní stravovací plán. Tabulka podkladů je ilustrována v tabulce 7. Procentuální rozložení kalorií do pěti jídel bylo založeno na volbě Šárky, která si vybrala konzumovat pět jídel denně. Podrobnosti ohledně toho, kolik kalorií by mělo každé jídlo obsahovat, jsou uvedeny v části 2.3. Pro Šárku je rozložení kalorií zobrazeno pomocí tabulky 8.

Tabulka 7: Tabulka Šárčiných podkladů pro tvorbu jídelníčku

Základní údaje	
Věk	28 let
Výška	170 cm
Hmotnost	65 kg
Cíl	Udržení váhy
Alergie	Ryby
Oblíbené jídlo	Houby
Neoblíbená jídla	Červená řepa
Počet jídel denně	5
Aktivita	
Pracovní náplň	Administrativní práce
Sport	Posilovna (3x týdně)
	Plavání (2x týdně)
Nutriční hodnoty (kcal/g)	
Kalorie	2103,1 kcal
Bílkoviny	97,5 g
Tuky	58,4 g
Sacharidy	296,9 g
Vláknina	30 g

Tabulka 8: Šárky procentuální rozložení kalorií a nutričních hodnot během dne

Typ jídla	Procento
Snídaně	19 %
První svačina	6 %
Oběd	31 %
Druhá svačina	6 %
Večeře	38 %

Když jsou všechny potřebné informace k dispozici, je možné postupně vybírat jídla a sestavit jídelníček. Začínáme postupem od snídaně, přes oběd a večeři až po svačiny, přičemž neustále dodržujeme tabulku podkladů, abychom zajistili, že nezůstanou nevyplněny žádné nutriční potřeby. Pro usnadnění výběru jídel můžeme využít různé zdroje, jako jsou například nutriční kuchařky, online databáze potravin nebo specializované aplikace. Níže je sepsán jeden z možných jídelníčků pro Šárku.

Tabulka 9: Součet nutričních složek ve vytvořeném jídelníčku pro Šárku

Kalorie	2109 kcal
Bílkoviny	86 g
Tuky	59 g
Sacharidy	332 g
Vláknina	32 g

Snídaně Pro snídani jsem vybral *Smoothie s hrozny a burákovým máslem*, obsahující 21 gramů bílkovin, 13 gramů tuku, 65 gramů sacharidů, 429 kalorií a 5 gramů vlákniny.

První svačina Pro první svačinu jsem zvolil *Kukuřičné placičky s rajčatovou a bazalkovou salsou*, které obsahují 4 gramy bílkovin, 1 gram tuku, 21 gramů sacharidů, 96 kalorií a 3 gramy vlákniny.

Oběd Na oběd jsem zvolil *Čedarovou polentu se sušenými rajčaty a krevetami*, která obsahuje 25 gramů bílkovin, 21 gramů tuku, 117 gramů sacharidů, 753 kalorií a 8 gramů vlákniny.

Druhá svačina Pro druhou svačinu jsem vybral *Salátový mix*, který obsahuje 5 gramů bílkovin, 1 gram tuku, 16 gramů sacharidů, 80 kalorií a 9 gramů vlákniny.

Večeře Pro večeři jsem zvolil *Krémové veganské nudle s koriandrem a limetkou*, které mají 31 gramů bílkovin, 23 gramů tuku, 113 gramů sacharidů, 751 kalorií a 7 gramů vlákniny.

Tímto jsme dokončili sestavení jídelníčku. V tabulce 9 je demonstrováno, kolik celkových nutričních složek má daný jídelníček. Z tabulky je patrné, že i přes pečlivý výběr jídel není vždy dosaženo přesných hodnot. Snažíme se, ale udržet co nejbližší stanoveným kritériím. Jídla a jejich nutriční složky jsem čerpal z databáze receptů Edamam [13].

3 Implementace

3.1 Založení a zprovoznění projektu

V této sekci popíšu proces vytvoření a zprovoznění projektu aplikace. Aplikace je vytvořena pomocí vývojového prostředí Xcode, které je dostupné pouze pro operační systém macOS. To znamená, že vývojáři s jiným operačním systémem nebudou moci použít tento postup pro tvorbu aplikace. Tento postup je podrobně popsán v oficiální dokumentaci React Native Environment Setup [17].

3.1.1 Instalace závislostí a vývojového prostředí

Instalace Node.js, Watchman a CocoaPods pomocí HomeBrew Prvním krokem je instalace Node.js, Watchman a CocoaPods pomocí HomeBrew.

- Node.js je prostředí pro běh JavaScriptu mimo prohlížeč, často používané pro vývoj aplikací nebo serverových aplikací.
- Watchman je nástroj vyvinutý společností Facebook pro sledování změn v souborech.
- CocoaPods je správce závislostí pro iOS projekty.
- HomeBrew je správce balíčků pro macOS, který usnadňuje instalaci softwaru.

Pro instalaci těchto závislostí stačilo zadat příkazy postupně do terminálu. Příkazy jsou sepsány níže:

```
/bin/bash -c "\$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh) "
```

```
brew install node
```

```
brew install watchman
```

```
brew install cocoapods
```

Instalace Xcode, iOS Simulatoru a Xcode Command Line Tools Dalším krokem je instalace Xcode a iOS simulatoru. Xcode je integrované vývojové prostředí (IDE) pro vývoj aplikací, vyvinuté společností Apple. Obsahuje sadu nástrojů potřebných pro vývoj, ladění a nasazování aplikací. Prostřednictvím AppStore, platformy pro distribuci aplikací pro zařízení s operačním systémem macOS, můžete snadno stáhnout a nainstalovat Xcode. Stačí otevřít AppStore, vyhledat Xcode a kliknout na tlačítko pro instalaci.

Pro spouštění a testování aplikace na Macu je dále nutné nainstalovat iOS Simulator. Postup je následující. Zapnout Xcode a přejít do nastavení. Poté stačí vybrat záložku "Platforms" a zvolit iOS Simulator s odpovídající verzí iOS.

Kompletní sestavení a kompilaci aplikace vyžaduje nainstalování Xcode Command Line Tools. Tyto nástroje jsou klíčové pro efektivní práci s Xcode a umožňují kompilaci projektů přímo z příkazové řádky. Nainstalovat je lze přes Xcode v jeho nastavení, kde je potřeba vybrat nejnovější verzi.

3.1.2 Založení a spuštění projektu

Po dokončení instalace závislostí a konfigurace prostředí je čas založit a spustit projekt. Založení projektu probíhá pomocí tohoto příkazu:

```
npx react-native@latest init *jmeno-projektu*
```

Tím se vytvoří projekt i jeho struktura. Teď už zbývá jen poslední část a to spustit projekt. Stačí otevřít terminál a přejít do složky projektu. Poté spustit Metro Bundler pomocí příkazu:

```
npm start
```

Metro Bundler je nezbytný pro správné fungování vaší React Native aplikace. Po spuštění Metro Bundleru je možné spustit aplikaci v iOS simulatoru pomocí následujícího příkazu:

```
npm run ios
```

Poté je možnost spouštět aplikaci i přes Xcode. Stačí otevřít soubor v projektové složce, který má příponu `.xcworkspace` a spouštět pomocí Xcode UI.

3.2 Registrace

Registrace a sběr uživatelských informací jsou kritickými prvky procesu vytváření osobního jídelníčku. Nyní popíši, jak jsem tuto problematiku implementoval v mé aplikaci.

3.2.1 Použití Reactu v procesu registrace

Při implementaci jsem využil React jako hlavní framework. Konkrétně jsem využil funkční komponenty a hooky, které React poskytuje. Například, v komponentě pro registraci `SignUpScreen`, jsem využil hooky `useState` pro sledování stavu jednotlivých polí formuláře, jako je e-mail a heslo. Tím jsem umožnil dynamickou aktualizaci těchto polí v reálném čase podle uživatelovy interakce.

Kromě toho jsem využil React pro zpracování uživatelských událostí, například kliknutí na tlačítko pro odeslání registračního formuláře. Pomocí `onPress` události jsem spustil funkci, která odesílá data z formuláře k provedení registrace. Obě tyto záležitosti jsou zobrazeny v kódu 1 [14][15].

```

1      // useState hook
2      const [email, setEmail] = useState('');
3      <RegistrationTextInputCustom
4      title="E-mail"
5      value={email}
6      textInputFor={setEmail}
7      signUpInScreenStyles={signUpInScreenStyles}
8      secureTextEntry={false}
9      placeholder='e-mail' />
10
11     // Událost
12     <CustomButton onPress={() => SignUp(email, password,
13     againPassword, navigation)} buttonText="Sign Up"
    buttonColor="#8dc325" disabled={false}/>

```

Zdrojový kód 1: Ukázka React hooků a událostí

3.2.2 Autentizace

Pro zajištění autentizace jsem se rozhodl využít osvědčeného řešení *Firebase Authentication*. Tato platforma umožňuje bezpečné ukládání uživatelských dat do cloudu a podporuje ověření pomocí hesel, telefonních čísel a populárních poskytovatelů identit, jako jsou například Google, Facebook a další [11].

Aby bylo možné využívat Firebase a jeho funkce, je nezbytné vytvořit konfigurační soubor. Tento soubor obsahuje klíčové informace pro inicializaci a připojení k projektu ve Firebase. V mém případě jsem pojmenoval soubor `firebaseConfig.js`. V něm je konfigurační objekt obsahující důležité údaje, jako je API klíč, identifikátor projektu, doména a appID. Díky tomuto souboru je možné inicializovat aplikaci pomocí Firebase SDK a získat přístup k funkcím, jako jsou autentizace a ukládání dat do databáze. Ve zdrojovém kódu 2 je zobrazen obsah konfiguračního souboru bez hodnot a export objektů pro využití Firebase funkcí.

V mém řešení jsem se zaměřil na ověření pomocí e-mailové adresy a hesla. Nejprve systém zkontroluje, zda je zadaná e-mailová adresa opravdu e-mailovou adresou. Poté je zkontrolováno jestli není používána jiným uživatelem. Pro zajištění bezpečnosti hesel je vyžadováno, aby heslo obsahovalo alespoň 6 znaků a číslovku.

V následujícím kódu je demonstrováno získání hesla z úložiště klíčů, neboť v průběhu získávání dalších informací, které jsou popsány v sekci 3.2.3, je heslo šifrováno. Následně se za pomoci e-mailu a hesla založí účet. Ukázka tohoto procesu je prezentována ve zdrojovém kódu 3. Výsledná databáze s pěti založenými účty je znázorněna na obrázku 1.


```

1  import firebase from 'firebase/compat/app';
2  import 'firebase/compat/auth';
3  import 'firebase/compat/firestore';
4
5  const firebaseConfig = {
6    apiKey: "",
7    authDomain: "",
8    projectId: "",
9    storageBucket: "",
10   messagingSenderId: "",
11   appId: "",
12   measurementId: ""
13 };
14
15 if (!firebase.apps.length) {
16   firebase.initializeApp(firebaseConfig)
17 }
18
19 const FIREBASE_AUTH = firebase.auth();
20 export { firebase };
21 export { FIREBASE_AUTH };

```

Zdrojový kód 2: Konfigurační soubor Firebase

3.2.3 Sběr uživatelských dat a specifikací pro tvorbu jídelníčku

Po úspěšném získání přihlašovacích údajů je k dispozici možnost volby mezi dvěma základními přístupy k tvorbě jídelníčku:

- jídelníček bez ohledu na jeho nutriční složení,
- jídelníček s ohledem na jeho nutriční složení.

V případě první možnosti jsou požadované informace jméno, příjmení, pohlaví, hmotnost, věk, výška, pravidla, neoblíbené potraviny a alergie. Sběr těchto informací je popsán níže.

Pravidla Po zadání osobních informací má uživatel možnost vybrat si ze seznamu pravidel, která mu pomohou ještě více přizpůsobit svůj jídelníček. Tyto pravidla zohledňují různé faktory. Seznam pravidel, které jsem implementoval udává tabulka 10.

Neoblíbené potraviny Tato část registrace umožňuje uživateli specifikovat, jaké potraviny nechce zařazovat do svého jídelníčku. Zadání neoblíbených potravin, pomáhá vyloučit recepty obsahující ingredience, které uživatel nechce konzumovat. Tímto způsobem se zajišťuje, že jídelníček bude co nejvíce odpovídat individuálním potřebám uživatele.

Alergeny Na závěr uživatel uvádí své alergie z dostupného seznamu 14 alergenů. Seznam alergenů udává tabulka 3.

```

1   const createUserWithStoredCredentials = async () => {
2     try {
3       const credentials = await Keychain.getGenericPassword({
4         service: 'UserCredentials' });
5       if (credentials) {
6         const { username, password } = credentials;
7         await firebase.auth().createUserWithEmailAndPassword(
8           username, password);
9       } else {
10        console.log('Stored login credentials are not available.'
11        );
12      }
13    } catch (error) {
14      console.error('Error creating user from stored login
15      credentials:', error);
16    }
17  };

```

Zdrojový kód 3: Získání hesla z úložiště klíčů a založení účtu

Druhá možnost registračního procesu, vyžaduje stejné informace, jako první možnost, ale dále požaduje specifikaci úrovně fyzické aktivity, motivaci uživatele a preference ohledně výpočtu nutriční hodnot jídelníčku. Tyto informace navíc jsou popsány níže.

Úroveň fyzické aktivity Úroveň fyzické aktivity je vypočítána z denní aktivity viz Tab. 1 a z pohybových aktivit během týdne viz Tab. 2. Důkladný postup je sepsán v sekci [Aktivita](#).

Motivace Dále uživatel upřesní svou motivaci, přičemž možnosti jsou zvoleny podle této sekce [Změna váhy](#).

Předdefinované nebo vlastní nutriční hodnoty V tomto bodě má uživatel možnost vybrat si, zda chce nechat aplikaci automaticky vypočítat nutriční hodnoty jeho jídelníčku, nebo upřednostňuje ruční zadání těchto údajů. Implementace tohoto výpočtu je sepsána v sekci [Implementace výpočtu nutričních složek](#).

3.2.4 Uložení a odeslání uživatelských dat do databáze Firestore

Po dokončení procesu registrace a sběru uživatelských dat jsou všechny relevantní informace, včetně preferencí ohledně jídelníčku, uloženy a odeslány do databáze Firestore. Firestore je cloudová databáze typu NoSQL od společnosti Google, do které může moje iOS aplikace přistupovat přímo pomocí nativního SDK. V této databázi jsou informace uživatele uloženy ve formě dokumentů, které obsahují pole mapující na hodnoty. Tyto dokumenty jsou poté uloženy v kolekcích [12].

Identifier	Providers	Created ↓	Signed In	User UID
vojtech.barnet@me.com	✉	Apr 18, 2024	Apr 18, 2024	iwiicek20rNP6E2sH6sN79v0R...
sarka.nova@gmail.com	✉	Apr 14, 2024	Apr 17, 2024	e0qgZTPcXchLT5y51P00rgXjv...
ales.kercl@gmail.com	✉	Apr 14, 2024	Apr 14, 2024	AF0SqiRJxKN8Zi8RsDToCr6C...
eliska.novakova@gmail...	✉	Apr 13, 2024	Apr 14, 2024	OkzJZQ4tEUOLEPVjNCn2qUbt...
tomas.adamec@gmail...	✉	Apr 13, 2024	Apr 13, 2024	VAd6F07CvecXmLWz0e3qro9...

Obrázek 1: Vzhled autentizační databáze na platformě Firebase

Tato databáze je zobrazena na obrázku 2. Na obrázku je zobrazena struktura databáze. Každý uživatel má svůj vlastní dokument v kolekci *users*, jehož název odpovídá uživatelskému ID. Tento dokument obsahuje všechny informace specifické pro daného uživatele. V tomto dokumentu je následně uložen i jídelníček v vlastní kolekci. O tom více v sekci 3.3.3.

3.3 Implementace výpočtu nutričních složek a tvoření jídelníčku

3.3.1 Implementace výpočtu nutričních složek

Implementace výpočtu nutričních složek probíhá skrze jednu hlavní funkci, která řídí činnost pěti dalších funkcí, z nichž každá je zaměřena na výpočet konkrétní nutriční složky. Výsledné výpočty jsou uloženy do trvalého úložiště v mobilním zařízení, známého jako *AsyncStorage*.

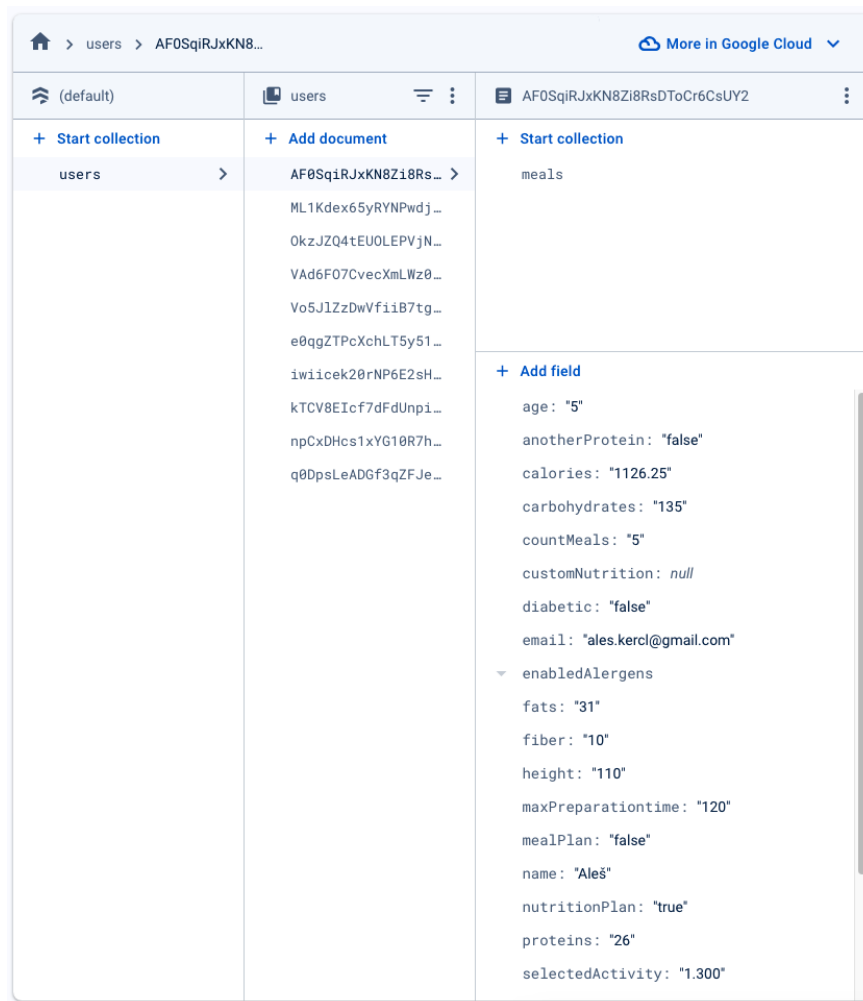
Každá z těchto podfunkcí je vytvořena na základě teorie, která je odborně popsána v sekci [Výpočet nutričních složek](#). Pro lepší představu zobrazím a okomentuji funkci *calculateCalories*, která, jak název naznačuje, provádí výpočet kalorií. Její implementaci naleznete v kódu 4. Tato funkce načte nezbytné informace o uživateli z argumentu *info* a následně je využije pro samotný výpočet. Zásadní je především uživatellovo pohlaví, protože pro muže a ženy platí jiné konstanty ve výpočetním vzorci.

3.3.2 Tvoření jídelníčku

Jídelníček s ohledem na jeho nutriční složení Pro tento proces jsem vytvořil obecnou funkci s názvem *generateMeals*, která poskytne seznam receptů. Pro lepší pochopení funkci podrobněji představím.

Tabulka 10: Pravidla

Název	Popis
Čas	Uživatel zde může vybrat, kolik času je ochoten věnovat přípravě jednoho jídla.
Diabetik	Toto pravidlo umožňuje uživatelům s diabetem upřesnit své potřeby ohledně jídelníčku a získat recepty přizpůsobené jejich dietnímu režimu.
Počet jídel denně	Uživatel zde může specifikovat, zda upřednostňuje tři hlavní jídla denně a jednu svačinu, nebo tři hlavní jídla denně a dvě svačiny.
Příjem proteinů	Uživatel zde může specifikovat, zda během dne konzumuje navíc potraviny nebo nápoje s vysokým obsahem bílkovin.



Obrázek 2: Firestore databáze s uživateli a jejich informacemi

```

1  const calculateCalories = (info: InfoType) => {
2      let genderConst, calories;
3      const {
4          age,
5          weight,
6          height,
7          selectedGender,
8          selectedGoal,
9          selectedActivity,
10     } = info;
11
12     if (selectedGender === 'male') {
13         genderConst = 5;
14     }
15     else {
16         genderConst = -161;
17     }
18     calories = (((10 * weight + 6.25 * height - 5 * age)
19 * selectedGoal * selectedActivity) + genderConst).toFixed(2);
20     return Number(calories);
21 };

```

Zdrojový kód 4: Kód na výpočet kalorií

generateMeals Funkce slouží k sestavení dotazu na databázi receptů a následné zpracování seznamu receptů. Databázi receptů jsem si vybral Edamam, která nabízí bezplatný přístup k jejich API pro studijní a testovací účely. [13] Pro spuštění této funkce je potřeba pět argumentů.

- První argument `userData` obsahuje veškeré informace o uživateli.
- Druhý argument `allergensParams` obsahuje uživatelovy alergie, které jsou přetvořené do parametrů.
- Třetí argument `unfavouriteIngredientsParams` obsahuje seznam ingrediencí, které uživatel nechce zahrnovat do svého jídelníčku přetvořené do parametrů.
- Čtvrtý argument `nutritionData` je pole obsahující informace o tom, kolik nutričních hodnot ještě zbývá dosáhnout nebo o kolik již bylo překročeno v daný den.
- Pátý argument `percentage` je konstanta, která určuje procentuální rozložení nutričních složek a kalorií pro specifický typ jídla. Tato konstanta je odvozena z typu jídla, údajů o věku a pohlaví uživatele.

Po spuštění funkce ověří, zda je uživatel diabetikem a zda využívá nějaké proteinové doplňky stravy. Poté se spustí funkce `calculateMinAndMaxValues`,

která vypočítá minimální a maximální hodnoty pro všechny nutriční složky, aby byl vytvořen rozsah, ve kterém je možné vyhledávat recepty. Tímto způsobem se snižuje riziko, že by nebyly nalezeny žádné vhodné recepty.

Následně je sestavován dotaz na API, který vrací seznam receptů. Postup sestavování dotazu je následující:

- `https://api.edamam.com/api/recipes/v2/`: URL adresa koncového bodu API poskytované společností Edamam, který umožňuje vyhledávání receptů.
- `app_id` a `app_key`: ID a klíč pro autentizaci uživatele při přístupu k API.
- `random`: Parametr pro náhodný výběr receptů.
- `type`: Parametr určující pro koho recepty jsou určeny.
- `mealType`: Parametr určující o jaký typ jídla se jedná (snídaně, oběd, večeře atd.)
- `health`: Parametr sloužící k popisu různých zdravotních aspektů receptu na základě informací o ingrediencích. Například alergie, bezlepkové diety atd.
- `calories`: Parametr pro specifikaci rozmezí kalorií receptů.
- `nutrients[CHOCDF.net]`: Parametr pro specifikaci rozmezí čistých sacharidů (bez vlákniny).
- `nutrients[FAT]`: Parametr pro specifikaci rozmezí tuků v receptech.
- `nutrients[PROCNT]`: Parametr pro specifikaci rozmezí proteinů v receptech.
- `nutrients[FIBTG]`: Parametr pro specifikaci rozmezí vlákniny v receptech.
- `time`: Parametr pro specifikaci maximálního času přípravy receptů.

Výsledný dotaz na API s ohledem na nutriční složení:

```
https://api.edamam.com/api/recipes/v2/?q=
&app_id=secretId
&app_key=secretKey
&random=true
&type=any
&mealType=Dinner
&health=alcohol-free
&health=peanutfree
&excluded=carrot
&calories=1010.8-1117.2
&nutrients[CHOCDF.net]=149.72-165.48
&nutrients[FAT]=28.12-31.08
&nutrients[PROCNT]=29.4-105
&nutrients[FIBTG]=30
&time=120
```

Seznam receptů, který API vrátí, obsahuje více receptů, než je potřeba. Proto je nutné vybrat prvních sedm z nich, které budou dále zpracovány, přičemž každý recept bude reprezentovat jeden den v týdnu. Následně jsou spuštěny všechny ostatní funkce pro nalezení dalších typů jídel. Během provádění těchto funkcí, jsou nutriční hodnoty vytvořených receptů odečítány od doporučených denních hodnot uživatele za účelem kontroly souladu. K tomu slouží již zmíněný čtvrtý argument `nutritionData`. Tento výsledek je pak pomocí funkce `checkNutritionValues` ověřen. Pokud jeho hodnoty splňují stanovené kritérium, jsou recepty uloženy do databáze na platformě Firebase. V případě odchylky od požadovaných hodnot je celý proces sestavování jídelníčku spuštěn znovu. Pokud se tento proces opakuje více než desetkrát, je zobrazena chybová hláška, která informuje uživatele o neúspěchu při nalezení vhodných receptů. V této hlášce se navrhuje, aby uživatel za nějaký čas zkusil proces znovu.

Jídelníček bez ohledu na jeho nutriční složení Pro tento proces je k dispozici obdobná funkce, jako v předchozím případě, avšak bez ohledu na nutriční složky. Funkce `generateMealsWithoutNutrition` slouží k sestavení dotazu na databázi receptů a jediné rozdíly oproti předchozí funkci jsou ve vynechání výpočtu a ověření nutričních složek.

Výsledný dotaz na API bez ohledu na nutriční složení:

```
https://api.edamam.com/api/recipes/v2/?q=
&app_id=secretId
&app_key=secretKey
&random=true
&type=any
&mealType=Dinner
&health=alcohol-free
&health=peanutfree
&excluded=carrot
&time=120
```

3.3.3 Uložení a odeslání jídelníčku do databáze Firestore

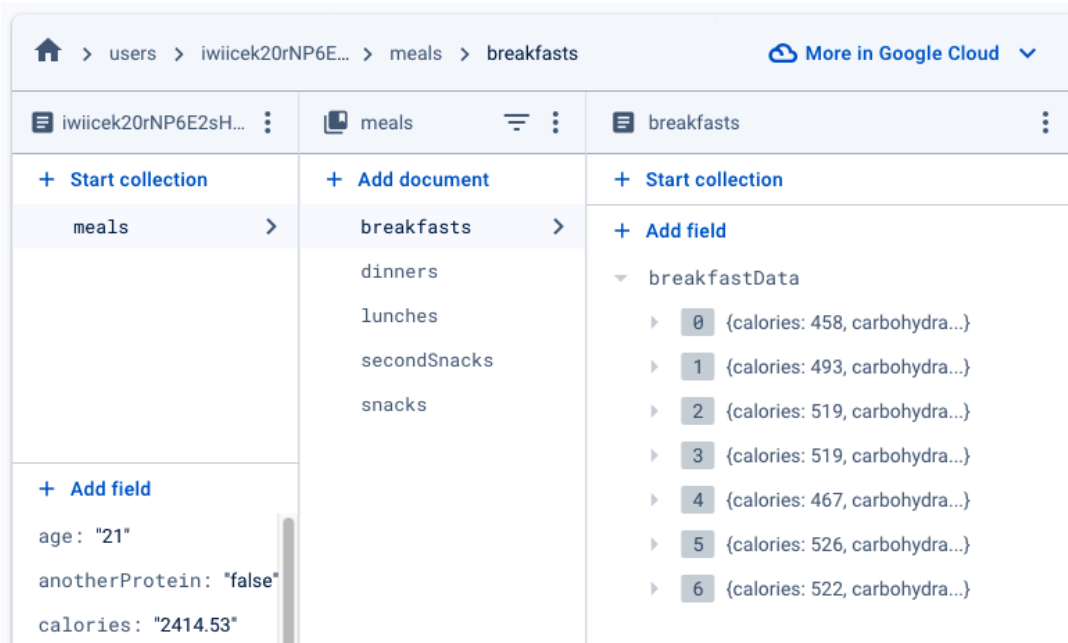
Po sestavení jídelníčku je nezbytné tento jídelníček odeslat a uložit do databáze Firestore. Každý jídelníček je spojen s konkrétním uživatelem a uložen v jeho dokumentu. Příklad databáze můžete vidět na obrázku 3. Struktura databáze je následující. Jak už jsem zmiňoval, každý uživatel má svůj vlastní dokument v kolekci *users*^o, jehož název odpovídá uživatelskému ID. V tomto dokumentu je poté buď vytvořena nebo aktualizována kolekce *meals* podle toho, zda uživatel již v minulosti vytvářel jídelníček, nebo je tento jídelníček jeho prvním. V kolekci *meals* jsou dále jednotlivé dokumenty pro různé typy jídel, jako jsou snídaně, obědy, večeře nebo svačiny. Každý dokument v této kolekci obsahuje seznam receptů a příslušných hodnot pro daný typ jídla.

Zabezpečení databáze Je důležité zdůraznit, že přímý přístup do databáze vyžaduje pečlivé zabezpečení. V mém případě jsem použil pravidla bezpečnosti Firestore neboli Cloud Firestore Rules. Níže jsou pravidla sepsána:

```
1  service cloud.firestore {
2    match /databases/{database}/documents {
3      match /users/{userId} {
4        allow read, write: if request.auth != null && request.auth.uid
5          == userId;
6      }
7      match /users/{userId}/meals/{document=**} {
8        allow read, write: if request.auth != null && request.auth.uid
9          == userId;
10     }
11   }
12 }
```

Zdrojový kód 5: Zabezpečení databáze

Tato pravidla umožňují čtení a zápis do dokumentů uživatelů a jejich jídelníčků pouze autentizovaným uživatelům, kteří odpovídají příslušnému uživatelskému ID.



Obrázek 3: Firestore databáze s uživatelským jídelníčkem

3.4 Organizace kódu

Pro efektivní správu a udržitelnost projektu jsem strukturoval kód do logických skupin neboli adresářů. Každý adresář má přesně stanovené k čemu slouží. Následující seznam popisuje hlavní adresáře a jejich obsah:

- **Functions:** Tato složka obsahuje soubory s funkcemi, které jsou v aplikaci používány opakovaně. Sem patří například soubory pro manipulaci s daty, přihlašování nebo vytvoření jídelníčku.
- **Screens:** V této složce jsou umístěny soubory definující jednotlivé obrazovky aplikace. Každý soubor reprezentuje jednu obrazovku.
- **Components:** Zde se nacházejí opakovaně použitelné komponenty, které jsou sdíleny mezi různými obrazovkami.
- **Styles:** Složka pro stylování aplikace obsahuje soubory s definicemi stylů. Jednotlivé soubory obsahují styly pro různé části aplikace, což usnadňuje údržbu a přehlednost kódu.
- **Firebase:** V tomto adresáři jsou umístěny konfigurační soubory související s Firebaseem.

3.5 Uživatelské rozhraní

Uživatelské rozhraní (UI) představuje klíčový prvek každé mobilní aplikace. Je to prostředí, ve kterém uživatel komunikuje s aplikací a provádí různé úkony. V kontextu mobilních aplikací se UI skládá z jednotlivých obrazovek, což jsou soubory obsahující vizuální komponenty, které jsou zobrazeny na displeji zařízení. K vytvoření uživatelsky přívětivého UI jsou zapotřebí technologie. V této sekci blíže představím ty, které jsem použil. Hlavní technologií je React Native. Dále se také budu zabývat knihovnou React Navigation, která umožňuje pohyb mezi obrazovkami v aplikaci, a samotným jazykem React, na kterém je React Native postaven. Dále představím, jak tyto technologie využívám v rámci mé aplikace a ukáži stylování pomocí vlastních stylů.

3.5.1 Navigace a obrazovky

Jak už jsem zmínil v úvodu, pro pohyb v aplikaci jsem zvolil knihovnu React Navigation. React Navigation je populární knihovna pro správu navigace v mobilních aplikacích postavených na platformě React Native. Umožňuje vytváření různých typů navigačních struktur, jako jsou například stack navigace, tab navigace nebo drawer navigace. Stack navigace, kterou jsem si vybral pro svojí aplikaci, je základní navigační vzor, který funguje na principu zásobníku (obvykle známého jako "stack"). Každá obrazovka (screen) v aplikaci je reprezentována jedním prvkem v tomto zásobníku. Při přechodu na novou obrazovku, je tato obrazovka vložena na vrchol zásobníku a uživateli je zobrazen její obsah. Když uživatel přejde zpět, poslední obrazovka je odebrána ze zásobníku a uživatel vidí obsah předchozí obrazovky. [18]

V mé aplikaci jsem obrazovky rozdělil do dvou zásobníků:

- Obrazovky pro nepřihlášené uživatele (`RootStack`).
- Obrazovky pro přihlášené uživatele (`InsideStack`).

Pokud uživatel není přihlášen, má možnost zobrazit si jen obrazovky `RootStacku`, které slouží k registraci a přihlášení. Po úspěšném přihlášení se uživatel dostane do `InsideStacku` a má možnost si obrazovky z něj zobrazit. Zde může prohlížet recepty, vytvářet jídelníčky a provádět další akce na daných obrazovkách. Ukázkou vytvoření zásobníku obrazovek lze nalézt v následujícím kódu 6. Kód zobrazuje funkci `InsideLayout()`, která definuje zásobník obrazovek pro přihlášené uživatele. Tento zásobník je vytvořen pomocí komponenty `<InsideStack.Navigator>`. Uvnitř této komponenty jsou definovány jednotlivé obrazovky pomocí komponent `<InsideStack.Screen>`. Každá obrazovka má definován svůj název a přidruženou komponentu, která zobrazuje obsah dané obrazovky. Také je nastaveno, že v hlavičce těchto obrazovek není zobrazena navigace.

```

1  function InsideLayout () {
2      return (
3          <InsideStack.Navigator>
4              <InsideStack.Screen name="WeekScreen" component=
                    {WeekScreen} options={{ headerShown: false }} />
5              <InsideStack.Screen name="DayScreen" component=
                    {DayScreen} options={{ headerShown: false }} />
6              <InsideStack.Screen name="UserSettings" component=
                    {UserSettings} options={{ headerShown: false }} />
7          </InsideStack.Navigator>
8      );
9  }

```

Zdrojový kód 6: Kód na vytvoření zásobníku obrazovek

3.5.2 Tvorba vlastních komponent pomocí technologie React Native

React Native komponenty jsou základní stavební bloky pro tvorbu UI v mobilních aplikacích, a také jsou klíčové pro tvorbu vlastních komponent. Jsou to znovupoužitelné JavaScriptové objekty, které reprezentují například textová pole, obrázky, tlačítka a další [16]. Každá komponenta má své vlastnosti. Její chování může být definováno pomocí vlastností a událostí. Příklady běžných React Native komponent:

- **View**: Jedna z nejdůležitějších komponent, která slouží k vytváření kontejnerů pro ostatní komponenty.
- **Text**: Používá se pro zobrazení textu na obrazovce a umožňuje nastavení formátování jako je velikost písma, barva atd.
- **TextInput**: Komponenta pro vstup textu od uživatele, například pro zadávání hesel, e-mailů, nebo dalších informací.
- **Pressable**: Poskytuje schopnost reagovat na stisknutí nebo klepnutí uživatelem. Může obklopovat jiné komponenty a zpracovávat události stisku.
- **Image**: Slouží k zobrazení obrázků na obrazovce a umožňuje načítání obrázků z různých zdrojů jako jsou lokální soubory nebo vzdálené adresy URL.

Nyní ukáži, jak některé z těchto komponent využívám k vytvoření vlastní komponenty.

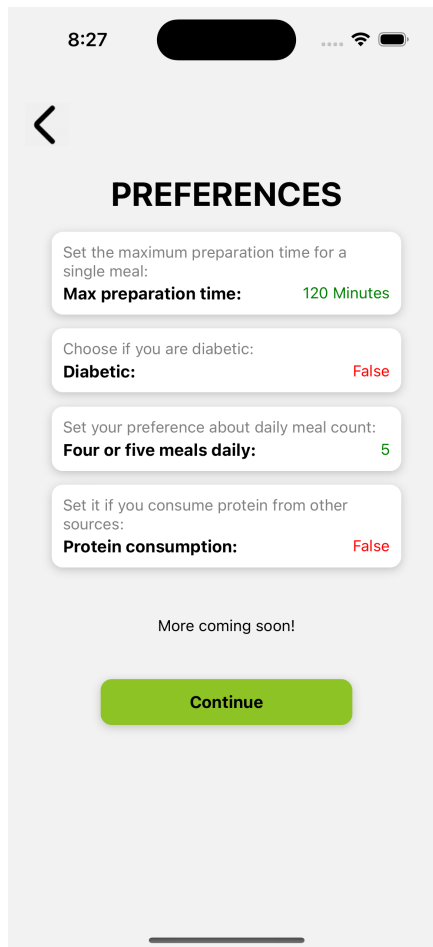
ChooseButton Kód 7 demonstruje komponentu **ChooseButton**. V mém projektu jsem často potřeboval vytvářet tlačítka s podobným vzhledem a chováním, proto jsem vytvořil tuto komponentu, kterou mohu opakovaně používat.

ChooseButton přijímá několik vlastností jako vstup, včetně parametru **onPressOption**, který určuje, co se má stát po stisknutí tlačítka a **text**, co

se má zobrazit na místě popisku a titulku. Dále přijímá i `selectedOption`, což je funkce díky které si uloží vybranou odpověď. A nakonec `optionStyle`, který umožňuje přizpůsobit stylizaci vybrané možnosti, například změnou barvy písma. Stylizace tlačítka je definována pomocí předem definovaných stylů, které jsou importovány z externího souboru `registrationStyles.js`. O stylech a jejich použití je více napsáno v této sekci 3.5.3. Konečnou podobu této komponenty, jak je zobrazena v aplikaci, můžete vidět na obrázku 4. Zobrazena je tam čtyřikrát.

```
1  import React from 'react';
2  import { TouchableOpacity, Text, View } from 'react-native';
3  import { registrationStyles } from '../Styles/registrationStyles'
4  ;
5  const ChooseButton = ({ onPressOption, optionStyle,
6    selectedOption, title, description }) => {
7    return (
8      <TouchableOpacity style={registrationStyles.choosingButton}
9        onPress={onPressOption}>
10       {description && <Text style={registrationStyles.
11         choosingButtonDescription}>{description}</Text>}
12       <View style={registrationStyles.choosingButtonRow}>
13         {title && <Text style={registrationStyles.
14           titleChoosingButton}>{title}</Text>}
15         {selectedOption && <Text style={optionStyle}>
16           {selectedOption}</Text>}
17       </View>
18     </TouchableOpacity>
19   );
20 };
21
22 export default ChooseButton;
```

Zdrojový kód 7: Kód komponenty ChooseButton



Obrázek 4: Ukázka čtyř komponent ChooseButton

3.5.3 Stylizace pomocí vlastních stylů

Jednou z klíčových vlastností tvorby uživatelského rozhraní v React Native je možnost používat vlastní styly k definici vzhledu a formátování komponent. Vlastní styly umožňují centralizované řízení vzhledu aplikace a zjednodušují úpravy a údržbu kódu. V kódu 8 je zobrazena část vytvořeného vlastního stylu. V tomto příkladu soubor `registrationStyles.js` obsahuje definice vlastních stylů pomocí JavaScript objektu. Tyto styly mohou být následně importovány do jiných souborů a použity v komponentách, jak již bylo zobrazeno v kódu 7.

```

1   export const registrationStyles = StyleSheet.create({
2     titleChoosingButton: {
3       fontSize: 15,
4       fontWeight: 'bold'
5     },
6
7     choosingButton:{
8       padding: 10,
9       borderRadius: 10,
10      backgroundColor: 'white',
11
12      elevation: 10,
13      borderRadius: 10,
14      shadowRadius: 4,
15      shadowOpacity: 0.2,
16      shadowOffset: { width: 0, height: 2 },
17      shadowColor: 'black',
18      marginBottom: '4%',
19    },
20    % Zde pokračují další vlastnosti...
21    % Další vlastnosti mohou být přidány dle potřeby.
22  })

```

Zdrojový kód 8: Vlastní styl pro komponentu

3.5.4 Ukázka finálního uživatelského prostředí

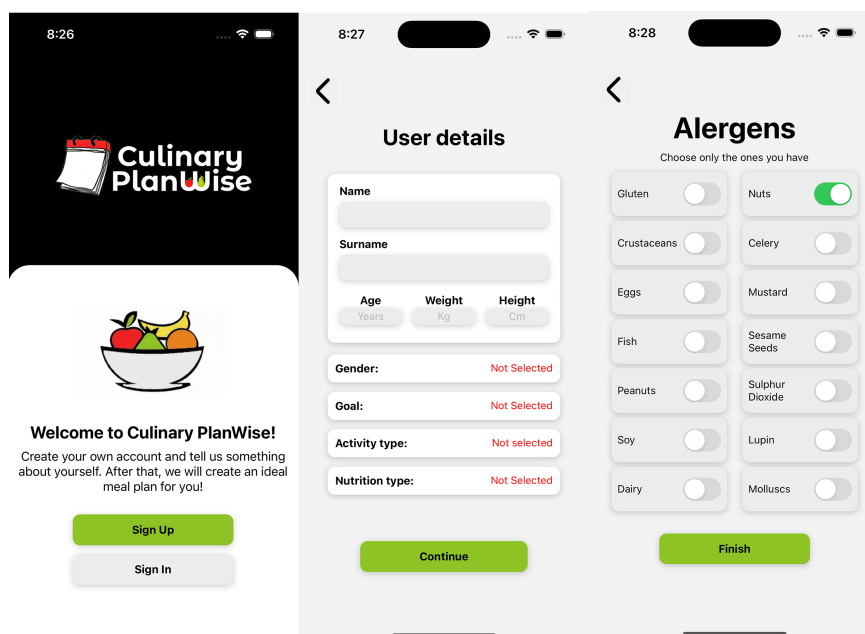
V této sekci prezentuji náhled finálního uživatelského rozhraní mé aplikace na základě dvou snímků.

První snímek První snímek dokumentuje obrazovky související s procesem registrace 5.

Na první části snímku je zobrazena úvodní obrazovka aplikace, která se zobrazuje při jejím prvním spuštění. Obsahuje logo, popisek a tlačítka umožňující uživatelům vytvořit nový účet nebo se přihlásit, pokud již účet vlastní. Tato tlačítka jsou implementována pomocí vlastní komponenty `CustomButton`.

Na druhé části je zobrazena obrazovka, zobrazující formulář pro sběr informací od uživatele. Pomocí klasických textových polí z React Native získáváme informace jako jméno, příjmení, věk, hmotnost a výška. Další informace, jako je pohlaví, jsou získány pomocí vlastní komponenty `ChooseButton`, jejíž vytvoření je popsáno v této sekci 3.5.2.

Ve třetí části je obrazovka, na které uživatel může uvést své alergie. Každý alergen je zobrazen jako klasický přepínač iOS, který jsem implementoval jako vlastní komponentu `AlergenSwitch`.



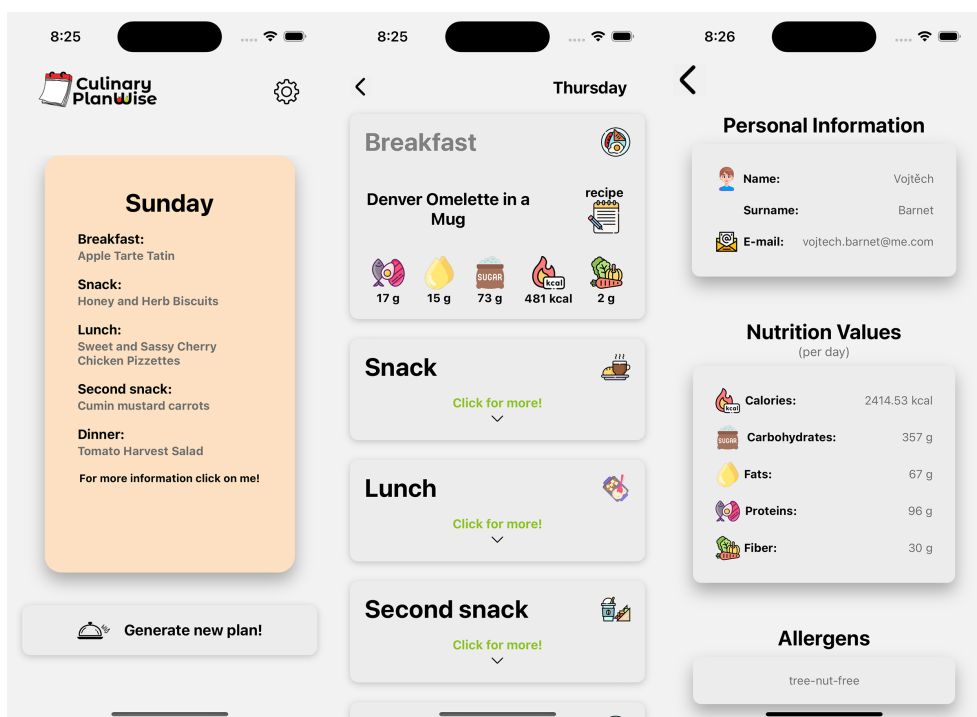
Obrázek 5: Část registračních obrazovek

Druhý snímek Po úspěšném přihlášení nebo registraci uživatele následují následující obrazovky, které jsou zachyceny na druhém snímku 6.

V první části je zobrazena obrazovka, která se zobrazí rovnou po přihlášení. V hlavičce obrazovky je umístěno logo, ikona ozubeného kolečka, která slouží k otevření obrazovky s nastavením. Tuto obrazovku můžete vidět v třetí části snímku. Pod hlavičkou je zobrazena karta, která obsahuje název dne a jídelníček pro tento konkrétní den. Tato karta funguje jako tlačítko a při kliknutí na ni se zobrazí detaily jídelníčku, jak je patrné na druhé části snímku. Pod kartou je umístěno tlačítko, které umožňuje uživateli vygenerovat nový jídelníček.

Druhá část snímku zahrnuje detaily jídelníčku pro daný den. Na ní jsou zobrazeny jednotlivé položky jídelníčku v podobě karet, každá reprezentuje jedno jídlo. Po rozkliknutí karty se zobrazí nutriční složení jídla a odkaz na recept.

Třetí část snímku zobrazuje obrazovku s informacemi o uživatelském účtu. Jsou na ní zobrazeny údaje jako jméno, příjmení, e-mail, alergie. Pokud uživatel zvolil sledování nutričních složek, zobrazuje se zde i jeho denní příjem.



Obrázek 6: Obrazovky po přihlášení a vytvoření jídelníčku

4 Zhodnocení

V této sekci se chci věnovat celkovému zhodnocení projektu Culinary PlanWise a zvážím možná vylepšení. První část se zaměří na hodnocení vytvořené aplikace z hlediska funkčnosti, uživatelského zážitku a použitých technologií. Druhá část se bude zabývat možnostmi vylepšení na základě mých nápadů a inspirace z jiných aplikací.

4.1 Hodnocení vytvořené aplikace

Po dokončení projektu CulinaryPlan Wise jsem celkově spokojen s výsledkem. Aplikace obsahuje všechny hlavní funkcionality, které jsem plánoval implementovat, včetně zohlednění alergií, diabetiků nebo dětí v předškolním věku. Uživatelské rozhraní je intuitivní a přívětivé, což uživatelům usnadňuje plánování jejich jídelníčku a následné vaření podle něho. Během vývoje jsem se snažil o detailní testování a odstraňování chyb, což vedlo k aplikaci, která běží plynule a bez větších problémů.

Přestože jsem s dosaženými výsledky spokojen, při zpětném pohledu na proces vývoje si uvědomuji, že bych některé věci změnil. V případě, že bych měl projekt realizovat znovu nebo ho představit ve veřejném prostoru, vážně bych uvažoval o vytvoření vlastního serveru a databáze. To by mi poskytlo větší kontrolu nad daty a zabezpečením. Rovněž bych se obrátil na odborníka v oblasti stravování, aby mi pomohl s vytvořením receptů a já je integroval do vlastní databáze, což by zaručilo vyšší kvalitu a relevantnost obsahu pro uživatele.

4.2 Možná vylepšení

Nákupní seznam Integrace nákupního seznamu by uživatelům výrazně usnadnila plánování a organizaci nákupů potravin na základě vybraných receptů v jejich jídelníčku. Tato funkce by automaticky vytvořila nákupní seznam surovin potřebných k přípravě receptů z jídelníčku. Následně pomocí další funkce by byl nákupní list zobrazen. Uživatel by mohl jednoduše procházet seznam a odškrtnout položky, které již má zakoupené.

Diety Rozšíření aplikace o funkci podporující různé diety by představovalo významné vylepšení pro uživatele se specifickými dietními potřebami. Tato funkce by vytvořila jídelníček v souladu s jejich konkrétními dietními omezeními nebo preferencemi, jako jsou například vegetariánství, veganská strava, bezlepková dieta nebo ketogenní strava.

Komunitní sdílení receptů a nových pravidel Uživatelé by měli možnost přispívat svými vlastními recepty a sdílet je s ostatními členy komunity. Tyto recepty by prošly procesem schvalování, kde by odborníci na stravování mohli posoudit jejich kvalitu a relevantnost. Kromě toho by uživatelé měli možnost

navrhovat nová pravidla pro tvorbu jídelníčků a sdílet své nápady s vývojářem. Takové sdílení by přispělo k bohatství a rozmanitosti dostupných receptů a pravidel.

Možnost změny cílů a osobních údajů Uživatelé by měli mít možnost flexibilně upravovat své cíle a osobní údaje v rámci svého účtu v aplikaci. Tato flexibilita by jim umožnila aktualizovat své informace v souladu se změnami v jejich životě, jako je například změna hmotnosti. Taková možnost by uživatelům přizpůsobila nutriční složení jídelníčku novým potřebám a cílům.

Závěr

Aplikace představuje komplexní nástroj pro generování osobních jídelníčků s ohledem na specifické potřeby uživatele. Během vývoje byl kladen důraz na praktickou nasaditelnost řešení s možností budoucího rozvoje. Uživatelské rozhraní bylo navrženo s důrazem na jednoduchost a responzivitu, aby poskytovalo pohodlný zážitek uživatelům na různých zařízeních od společnosti Apple. Ačkoli aplikace nabízí prostor pro zlepšení v oblasti tvorby jídelníčků a má potenciál pro optimalizaci výkonu, již v současné době dosahuje dostatečné úrovně k ostrému nasazení v praxi.

Conclusions

The application represents a comprehensive tool for generating personalized meal plans tailored to the specific needs of the user. During development, emphasis was placed on the practical deployability of the solution with the potential for future expansion. The user interface was designed with a focus on simplicity and responsiveness to provide a comfortable experience for users on various Apple devices. Although the application offers the possibility of further improvement in meal plan creation and has the potential for further optimization in performance, it already achieves a sufficient level for practical deployment in real-world scenarios.

A Obsah elektronických dat

CulinaryPlanWise/

Kompletní zdrojový kód iOS aplikace Culinary PlanWise. Struktura adresáře je detailněji popsána v `README.txt` souboru.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu Ki-diplom PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový kód textu, vložené obrázky, apod.

README.txt

Textový soubor pro instalaci a spuštění aplikace.

Literatura

- [1] KLEIN, Ondřej. JAK SPOČÍTAT PŘÍJEM ENERGIE A MAKROŽIVIN PRO HUBNUTÍ, NEBO NABÍRÁNÍ SVALŮ? GymBeam [online]. 2024, 2014-2024 [cit. 2024-03-20]. Dostupné z: <https://gymbeam.cz/blog/jak-spocita-t-prijem-energie-a-makrozivin-pro-hubnuti-nebo-nabirani-svalu/>
- [2] CHERNUS, Andrea; SKOLNIK, Heidi. Výživa pro maximální sportovní výkon: správně načasovaný jídelníček. Grada, 2011. ISBN 978-80-247-3847-5.
- [3] AUBRE, Manuela. Katch McArdle Equation for nutrition professionals [online]. 2023-05-10. [cit. 2024-03-20]. Dostupné z: <https://nutrium.com/blog/katch-mcardle-equation-for-nutrition-professionals/>
- [4] AUBRE, Manuela. Mifflin-St. Jeor for nutrition professionals [online]. 2023-05-03. [cit. 2024-03-20]. Dostupné z: <https://nutrium.com/blog/mifflin-st-jeor-for-nutrition-professionals/>
- [5] GESUNDHEIT.GV.AT. Povinné označování potravin obsahujících alergenů [online]. [cit. 2024-03-20]. Dostupné z: <https://www.nzip.cz/clanek/364-povinne-oznacovani-potravin-obsahujicich-alergeny>
- [6] VALENTA R. ;HOCHWALLNER H.LINHART B. PAHR S.. Food allergies: the basics [online]. 2015-05-01.[cit. 2024-03-20]. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4414527/>
- [7] BELLISLE F.; MCDEVITT R.; PRENTICE AM.. Meal frequency and energy balance. [cit. 2024-03-20]. British Journal of Nutrition. 1997 Apr; doi: 10.1079/bjn19970104.
- [8] GUNNARS Kris. Optimal Meal Frequency — How Many Meals Should You Eat per Day? 2018-04-18. [cit. 2024-03-20]. Dostupné z: <https://www.healthline.com/nutrition/how-many-meals-per-day>
- [9] PAOLI A.; TINSLEY G.; BIANCO A.; MORO T.. The Influence of Meal Frequency and Timing on Health in Humans: The Role of Fasting. Nutrients. 2019 Mar 28 Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6520689/>
- [10] MURPHY SP.; YAKTINE AL.; WEST SUITOR C.. Child and Adult Care Food Program: Aligning Dietary Guidance for All. Washington (DC): National Academies Press (US); 2011. 6. Dostupné z: <https://www.ncbi.nlm.nih.gov/books/NBK209815/>
- [11] GOOGLE. Firebase Authentication. Firebase [online]. 2024-04-16 [cit. 2024-04-26]. Dostupné z: <https://firebase.google.com/docs/auth>

- [12] GOOGLE. Cloud Firestore. Firebase [online]. 2024-04-25 [cit. 2024-04-26]. Dostupné z: <https://firebase.google.com/docs/firestore>
- [13] IGNATIEV I.; PENEV V. Edamam Recipe Search API. Edamam [online]. [cit. 2024-04-26]. Dostupné z: <https://developer.edamam.com/edamam-recipe-api>
- [14] NARAYN H. Just React!: Learn React the React Way. Apress, 2022. ISBN 1484282930.
- [15] META. React [online]. [cit. 2024-04-26]. Dostupné z: <https://react.dev/>
- [16] META. React Native [online]. [cit. 2024-04-26]. Dostupné z: <https://reactnative.dev/>
- [17] META. React Native Environment setup [online]. [cit. 2024-04-26]. Dostupné z: <https://reactnative.dev/docs/environment-setup>
- [18] React Navigation [online]. [cit. 2024-04-26]. Dostupné z: <https://reactnavigation.org/>