

UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A
INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2025

JAKUB HAVEL

Univerzita Pardubice
Fakulta Elektrotechniky a Informatiky

Aplikace pro sledování jídelníčku
Bakalářská práce

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Jakub Havel**
Osobní číslo: **I22094**
Studijní program: **B0688A140009 Informační technologie**
Téma práce: **Aplikace pro sledování jídelníčku**
Zadávací katedra: **Katedra informačních technologií**

Zásady pro vypracování

Cílem této bakalářské práce je vyvinout mobilní aplikaci, která pomáhá uživatelům zlepšit jejich fyzické zdraví a poskytuje nástroje pro sledování tohoto procesu. Aplikace bude vytvořena ve frameworku Flutter, s využitím jazyka Dart pro multiplatformní vývoj. Hlavní funkcionality aplikace budou zahrnovat sledování stravy, gamifikaci s výzvami a odměnami, motivační notifikace, sledování chůze a běhu pomocí telefonních senzorů a ukládání dat uživatelů.

Rozsah pracovní zprávy: **min. 30 stran**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

Flutter – A. Miola, *Flutter Complete Reference: Create beautiful, fast and native apps for any device*. Independently published, 2020. Dart – M. Schwarzmüller, *Learn Flutter and Dart to Build iOS and Android Apps – First Edition: A complete guide to Flutter SDK and Flutter framework to build native iOS and Android apps*. Packt Publishing, 2018. Fitness – C. Armbruster, *Fitness and Wellness: A Way of Life*. Human Kinetics, 2018.

Vedoucí bakalářské práce: **Ing. Jan Panuš, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **15. prosince 2024**
Termín odevzdání bakalářské práce: **16. května 2025**

prof. Ing. Petr Doležel, Ph.D. v.r.
děkan

L.S.

Ing. Jan Panuš, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 28. února 2025

Prohlašuji:

Práci s názvem Aplikace pro sledování jídelníčku jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 28.02.2025

Jakub Havel

PODĚKOVÁNÍ

Rád bych poděkoval panu Ing. Janu Panušovi, Ph.D. za jeho odborné vedení, vstřícný přístup a poskytnutí cenných rad pro vytvoření této práce.

ANOTACE

Tato bakalářská práce se zabývá vývojem mobilní aplikace pro sledování konzumace potravin a pohybové aktivity. Aplikace je vytvořena ve frameworku Flutter s využitím programovacího jazyka Dart a ukládá data lokálně pomocí databáze Hive. Uživatelé mohou vyhledávat potraviny prostřednictvím Open Food Facts API, přidávat je do svého denního jídelníčku a sledovat svůj pokrok pomocí vizuálních indikátorů denního příjmu živin.

Součástí aplikace je také sledování pohybové aktivity pomocí senzorů mobilního zařízení. Pro práci s polohou a pohybem byly použity knihovny flutter_map, latlong2, location a pedometer.

KLÍČOVÁ SLOVA

Flutter, Dart, Hive, Open Food Facts API, sledování stravy, mobilní aplikace, senzory pohybu, lokalizace

TITLE

Aplikace pro sledování jídelníčku

ANNOTATION

This bachelor's thesis focuses on the development of a mobile application for tracking food consumption and physical activity. The application is built using the Flutter framework with the Dart programming language and stores data locally using the Hive database. Users can search for food products through the Open Food Facts API, add them to their daily meal plan, and monitor their progress using visual indicators of daily nutrient intake.

The application also includes physical activity tracking using mobile device sensors. For working with location and movement data, the libraries flutter_map, latlong2, location, and pedometer were utilized.

KEYWORDS

Flutter, Dart, Hive, Open Food Facts API, food tracking, mobile application, motion sensors, localization

Obsah

SEZNAM ZKRATEK A ZNAČEK	11
ÚVOD.....	12
1 POUŽITÉ TECHNOLOGIE.....	13
1.1 Flutter.....	13
1.2 Dart	14
1.3 Hive.....	14
1.4 Open Food Facts API.....	15
1.5 Senzory pohybu a lokace	15
2 Analytická část.....	17
2.1 Framework Flutter	17
2.2 Databázové systémy v mobilních aplikacích.....	17
2.3 Open Food Facts API a práce s potravinovými databázemi	18
2.4 Senzory mobilního zařízení pro sledování pohybu.....	18
2.5 UML Diagram aplikace	20
2.5.1 Diagram úvodní stránky.....	20
2.5.2 Diagram stránky pro sledování kalorií.....	21
2.5.3 Diagram třetí a čtvrté stránky	22
2.6 Funkční požadavky	23
2.7 Možnosti dalšího rozšíření.....	23
2.8 Porovnání s existujícími aplikacemi	24
3. Vývoj a realizace aplikace	25
3.1 Instalace	25
3.2 Popis funkce aplikace	26
3.3 Úvodní strana.....	26
3.3.1 Úvod.....	26
3.3.2 Funkcionalita	26
3.3.3 Použité výpočetní vzorce	26

3.3.4 Ukládání do databáze	28
3.4 Stránka pro sledování kalorií	28
3.4.1 Úvod.....	28
3.4.2 Vyhledávání potravin.....	28
3.4.3 Přidávání potravin do seznamu.....	29
3.4.4 Seznam zkonsumovaných potravin	30
3.4.5 Ukazatele průběhu	31
3.4.6 Zdroj dat – Hive.....	32
3.5 Stránka pro monitorování aktivit	33
3.5.1 Úvod.....	33
3.5.2 Získání oprávnění	33
3.5.3 Sledování aktivity	33
3.6 Historie aktivit	34
3.6.1 Úvod.....	34
3.6.2 Popis stránky.....	34
4 Unit Testy	35
4.1 Úvod.....	35
4.2 Fitness Plan Service	35
4.3 Food Calorie Tracker Service	35
4.4 Hive Database Service	36
ZÁVĚR	37
POUŽITÁ LITERATURA	38
SEZNAM PŘÍLOH.....	40

SEZNAM ILUSTRACÍ A TABULEK

Obrázek 1: UML diagram první stránky.....	20
Obrázek 2: UML diagram stránky pro sledování kalorií.....	21
Obrázek 3: UML diagram třetí a čtvrté stránky.....	22
Obrázek 4: Vyhledávání potravin.....	29
Obrázek 5: Zadávání počtu zkonsumovaných gramů potravin.....	30
Tabulka 1: Tabulka koeficientů aktivity.....	27
Tabulka 2: Tabulka úrovně aktivity.....	27

SEZNAM ZKRATEK A ZNAČEK

UML	Unified Modeling Language
BMR	Basal Metabolic Rate
TDEE	Total Daily Energy Expenditure
API	Application Programming Interface
JSON	JavaScript Object Notation

ÚVOD

V dnešní době si stále více lidí uvědomuje důležitost zdravé výživy a aktivního životního stylu. S rostoucí dostupností informací o potravinách a nutričních hodnotách roste i zájem o digitální nástroje, které pomáhají uživatelům sledovat jejich stravovací návyky a pohybovou aktivitu. Mobilní aplikace zaměřené na monitoring živin a fyzické aktivity mohou významně přispět k lepšímu povědomí o vlastním zdraví a podpořit dosažení osobních cílů v oblasti výživy a pohybu.

Tato bakalářská práce se zaměřuje na vývoj mobilní aplikace umožňující sledování konzumace potravin a pohybové aktivity. Aplikace je vyvinuta pomocí moderního frameworku Flutter s využitím programovacího jazyka Dart a ukládá data lokálně prostřednictvím databáze Hive. Klíčovou funkcionalitou aplikace je vyhledávání potravin v databázi Open Food Facts, jejich přidávání do denního jídelníčku a vizualizace nutričních hodnot pomocí progres barů, které znázorňují dosaženou část denního cíle. Kromě sledování příjmu živin aplikace také monitoruje pohyb uživatele pomocí senzorů mobilního zařízení, přičemž využívá knihovny flutter_map, latlong2, location a pedometer.

Hlavním cílem práce je vytvořit intuitivní a efektivní nástroj, který pomůže uživatelům lépe porozumět jejich stravovacím návykům, optimalizovat příjem živin a zvýšit povědomí o jejich denní fyzické aktivitě. Práce se věnuje jak teoretickým aspektům vývoje mobilních aplikací a zpracování dat o potravinách, tak praktické implementaci, testování a vyhodnocení výsledků.

1 POUŽITÉ TECHNOLOGIE

Pro vývoj mobilní aplikace byly využity moderní a efektivní technologie, které umožňují multiplatformní vývoj s důrazem na výkon, stabilitu, bezpečnost a uživatelskou přívětivost. Vhodná volba technologií zásadním způsobem ovlivňuje nejen výsledný vzhled a funkčnost aplikace, ale také samotný proces vývoje, ladění a údržby. V této práci byly využity především následující nástroje a knihovny: framework Flutter, programovací jazyk Dart, databázový systém Hive pro lokální úložiště, rozhraní Open Food Facts API pro získávání dat o potravinách a různé senzory mobilního zařízení pro sledování pohybu, polohy a fyzické aktivity uživatele.

1.1 Flutter

Flutter je open-source framework vyvinutý společností Google pro vývoj nativně kompilovaných aplikací z jediného zdrojového kódu. Umožňuje vývoj aplikací pro Android, iOS, web a desktop s důrazem na vysoký výkon, vizuální konzistenci a rychlou iteraci během vývoje.

Flutter využívá deklarativní přístup k tvorbě uživatelského rozhraní, kde je každá komponenta reprezentována jako tzv. widget. Widgety jsou znovupoužitelné prvky, které mohou být základní (např. tlačítko, text, ikona) nebo složené (např. seznam, formulář, karta). Tato architektura umožňuje jednoduše vytvářet komplexní rozhraní s možností plné přizpůsobitelnosti.

Flutter používá grafický engine Skia, který přímo vykresluje komponenty na plátno bez závislosti na nativních prvcích operačního systému. Díky tomu je zajištěna konzistentní vizuální podoba aplikací napříč platformami, a to i na starších zařízeních. Výsledkem je velmi plynulý chod aplikace, i při náročnějších grafických operacích.

Funkce hot reload je dalším velkým přínosem Flutteru. Vývojář může po změně kódu téměř okamžitě vidět výsledek v běžící aplikaci bez nutnosti jejího úplného restartování, což výrazně urychluje ladění i testování nových funkcí.

Díky rozsáhlé komunitě a bohatému ekosystému pluginů (např. pro databáze, senzory, mapy, oprávnění) je Flutter jednou z nejrychleji rostoucích technologií pro vývoj mobilních aplikací. V této práci byl Flutter zvolen jako hlavní framework díky svému multiplatformnímu přístupu, rychlosti, modernímu vývojářskému prostředí a rozsáhlé dokumentaci. (1)

1.2 Dart

Dart je programovací jazyk, který byl navržen a vyvíjen společností Google jako jazyk pro tvorbu uživatelských rozhraní s důrazem na jednoduchost, výkonnost a moderní syntaxi. Dart se stal přirozeným jazykem pro vývoj aplikací ve Flutteru, jelikož je s tímto frameworkem úzce provázán. Je objektově orientovaný a jeho syntaxe je inspirována jazyky jako Java, JavaScript nebo C#, což vývojářům usnadňuje přechod a rychlé pochopení jeho principů.

Mezi hlavní přednosti jazyka Dart patří jeho schopnost provádět Ahead-of-Time kompilaci, která překládá kód do nativního strojového jazyka ještě před samotným spuštěním aplikace. Díky tomu je výsledná binární aplikace optimalizovaná pro konkrétní zařízení, což přispívá k rychlosti a stabilitě běhu. Dart zároveň podporuje i Just-in-Time kompilaci, která umožňuje rychlé testování a hot reload při vývoji. (2; 3)

Důležitým aspektem jazyka Dart je také jeho přirozená podpora asynchronního programování pomocí konstrukcí jako Future a Stream. Tyto mechanismy umožňují provádět operace, jako je přístup k databázím, čtení ze senzorů nebo načítání dat z internetu, bez zablokování hlavního vlákna aplikace. To je zásadní pro zajištění plynulého a responzivního uživatelského rozhraní. Díky systému null safety, který Dart podporuje, je navíc minimalizováno riziko výskytu chyb spojených s hodnotou null, což přispívá k bezpečnosti a stabilitě celého řešení.

1.3 Hive

Hive je lehká, rychlá a výkonná NoSQL databáze určená pro Flutter aplikace. Je navržena tak, aby byla snadno použitelná a efektivní při práci s lokálně uloženými daty, což ji činí ideální pro mobilní aplikace, které potřebují ukládat uživatelská data přímo na zařízení.

Tato databáze používá binární serializaci, což umožňuje rychlé čtení a zápis dat. Namísto tradičních relačních tabulek se Hive spoléhá na klíč-hodnota (key-value) model, kde jsou data ukládána ve formě objektů. To přináší spoustu výhod, jako například rychlost, Hive je výrazně rychlejší než SQLite a jiné databázové systémy, protože nepoužívá SQL dotazy, ale přímo čte a zapisuje binární data. Tato optimalizovaná struktura také umožňuje efektivní ukládání dat bez zbytečných režijních nákladů. Hive také využívá Future a async/await, což umožňuje provádět operace bez blokování hlavního vlákna aplikace.

Hive organizuje data pomocí tzv. boxů (Hive boxes), což jsou kontejnery, ve kterých jsou uloženy objekty. Každý box může obsahovat několik položek, přičemž každá položka má svůj unikátní klíč.

1.4 Open Food Facts API

Open Food Facts je veřejná a komunitně spravovaná databáze potravin, která poskytuje informace o složení, nutričních hodnotách, alergenech, kategoriích a dalších parametrech jednotlivých produktů. Databáze obsahuje data o statisících produktů z celého světa a je pravidelně aktualizována uživateli i výrobci. Přístup k těmto datům je umožněn prostřednictvím REST API, které vrací výsledky ve formátu JSON.

Pomocí tohoto API je možné vyhledávat potraviny podle názvu, čárového kódu nebo jiných parametrů a následně zobrazit detailní nutriční informace. API poskytuje nejen číselné hodnoty, jako jsou kalorie, tuky, sacharidy a bílkoviny, ale i složení výrobku, jeho původ nebo obrázky. Díky tomu je možné v aplikaci vytvořit komplexní výživový profil dané potraviny.

V rámci této aplikace je API využito ke zjednodušení procesu zadávání potravin do systému, kdy uživatel nemusí ručně vypisovat nutriční hodnoty, ale jednoduše vyhledá produkt a načte jeho informace. To výrazně zvyšuje komfort používání aplikace a zároveň zajišťuje přesnost a konzistenci zaznamenaných dat. (4)

1.5 Senzory pohybu a lokace

Aplikace využívá různé senzory mobilního zařízení ke sledování fyzické aktivity a polohy uživatele. K jejich implementaci bylo využito několik knihoven.

První z nich byla knihovna flutter_map. Tato knihovna je využita pro zobrazování interaktivních map v aplikaci. Je založena na Leaflet.js, což je populární knihovna pro práci s mapami. Této aplikaci umožňuje vizualizaci tras pohybu uživatele při běhu, či chůzi a zobrazení aktuální polohy na mapě. (5)

Druhou knihovnou, kterou tato aplikace využívá je latlong2. Tato knihovna poskytuje nástroje pro práci se souřadnicemi GPS, což je důležité pro výpočty vzdáleností a tras. V aplikaci se využívá pro výpočtu vzdálenosti mezi dvěma body a transformaci souřadnic GPS na formát vhodný pro mapové zobrazení. (6)

Třetí knihovnou je location, která umožňuje přístup k údajům o poloze zařízení pomocí GPS a dalších senzorů. V aplikaci se využívá k získání aktuální polohy uživatelova zařízení, detekci změn polohy v reálném čase a zaznamenávání trasy uživatele během aktivity. (7)

Čtvrtou knihovnou je pedometer. Slouží k počítání kroků pomocí akcelerometru a gyroskopu. V aplikaci je využit pro počítání kroků při spuštěné aktivitě jako je například chůze. (8)

Poslední knihovnou je `permission_handler`. Tato knihovna umožňuje všem předešlým přístup k sensorům uživatelského zařízení, jelikož spravuje oprávnění v mobilních aplikacích. Vzhledem k tomu, že aplikace využívá senzory a služby, jako je GPS a krokoměr, je nutné zajistit, aby uživatel aplikaci udělil potřebná oprávnění. Je důležitou součástí aplikace, protože zajišťuje, že uživatelé mohou správně využívat funkce vyžadující specifická oprávnění. (9)

2 Analytická část

2.1 Framework Flutter

Tradiční vývoj mobilních aplikací často vyžadoval samostatnou implementaci pro Android a iOS, což bylo časově, technicky i finančně náročné. Vývojář musel zvládat dva různé programovací jazyky (např. Java/Kotlin pro Android a Swift/Objective-C pro iOS) a přizpůsobovat aplikaci specifickým požadavkům každé platformy. Tento přístup často vedl ke zdoluhavému ladění chyb a složitému testování.

S příchodem multiplatformních frameworků, jako je Flutter, se vývoj výrazně zjednodušil. Flutter je open-source framework vyvinutý společností Google, který umožňuje vytvářet nativně kompilované aplikace pro Android, iOS, web i desktop z jediného sdíleného kódu. Jednou z hlavních výhod Flutteru je jeho vysoká rychlost a výkonnost, která je umožněna díky použití grafického enginu Skia. Tento engine dokáže vykreslovat rozhraní přímo bez závislosti na nativních komponentech jednotlivých platforem, což vede k jednotnému vzhledu aplikace na všech zařízeních.

Flutter se vyznačuje také použitím jazyka Dart, který je jednoduchý a snadno čitelný. Samotný vývoj probíhá pomocí widgetů, což jsou základní stavební prvky uživatelského rozhraní – každý prvek na obrazovce je widget, což vývoj výrazně zjednodušuje a zpřehledňuje. Kromě toho vývojář ocení funkci hot reload, která umožňuje okamžitě zobrazit změny v kódu přímo v běžící aplikaci, bez nutnosti jejího restartování.

Díky těmto vlastnostem byl Flutter zvolen jako hlavní nástroj pro vývoj aplikace v této práci. Nabízí jak vysokou míru flexibility, tak i rychlý a efektivní vývoj napříč platformami, což je zvláště výhodné pro menší projekty a individuální vývoj. (10)(11)

2.2 Databázové systémy v mobilních aplikacích

Databázové systémy představují klíčovou součástí téměř každé moderní aplikace, neboť umožňují efektivní správu, ukládání a načítání dat. V prostředí mobilních aplikací se databáze obvykle dělí na lokální a cloudové. Lokální databáze ukládají data přímo v zařízení, bez nutnosti připojení k internetu – typicky se využívají v případech, kdy aplikace musí fungovat offline. Cloudové databáze, jako je Firebase nebo MongoDB Atlas, naproti tomu umožňují synchronizaci dat mezi více zařízeními a nabízejí výhody jako zálohování a škálovatelnost, ale vyžadují internetové připojení.

V této aplikaci byla zvolena lokální NoSQL databáze Hive, která je optimalizovaná pro mobilní prostředí. Hive se vyznačuje vysokým výkonem, díky ukládání dat ve vysoce komprimovaném binárním formátu, a přitom zůstává velmi jednoduchá na implementaci. Oproti tradičnějším databázím, jako je SQLite, nevyžaduje definování schémat, a je tak velmi flexibilní.

Další výhodou Hive je podpora reaktivního programování – změna dat se ihned odrazí v uživatelském rozhraní, což je ideální pro moderní aplikace s dynamickým UI. V této aplikaci Hive slouží pro ukládání přidávaných potravin, jejich nutričních hodnot, historii konzumace i zaznamenaných aktivit uživatele. Díky tomu je aplikace plně funkční i bez připojení k internetu a uživatel má okamžitý přístup ke svým datům. (12)

2.3 Open Food Facts API a práce s potravinovými databázemi

Open Food Facts je rozsáhlá a volně dostupná databáze potravin, která byla založena komunitou nadšenců a odborníků v oblasti výživy. Obsahuje informace o složení, nutričních hodnotách, alergenech, přísadách, kategoriích produktů a dalších attributech desítek tisíc výrobků po celém světě. Díky své otevřenosti je ideálním nástrojem pro vývojáře, kteří potřebují integrovat potravinová data do svých aplikací.

Aplikace vytvořená v rámci této práce využívá REST API této databáze. Uživatel může pomocí vyhledávacího pole zadat název potraviny, načež aplikace odešle požadavek na Open Food Facts API a obdrží odpověď s detailními informacemi. Ty zahrnují např. kalorickou hodnotu, obsah bílkovin, sacharidů, tuků a vlákniny, ale také například obrázek produktu nebo informace o jeho nutriční známce.

Velkou výhodou této databáze je její pravidelná aktualizace a otevřenost, díky čemuž se neustále rozrůstá. Tato funkce přináší uživateli možnost rychlého a jednoduchého přidání potraviny bez nutnosti manuálního zadávání dat, čímž šetří čas a zvyšuje přesnost sledování příjmu.

2.4 Senzory mobilního zařízení pro sledování pohybu

Moderní chytré telefony jsou dnes vybaveny celou řadou senzorů, které umožňují monitorování pohybu, polohy a fyzické aktivity uživatele. Tato data jsou cenná nejen pro fitness aplikace, ale také pro personalizaci obsahu nebo zdravotní sledování.

V této aplikaci jsou využity lokalizační služby prostřednictvím knihovny `location`, která poskytuje informace o aktuální GPS poloze zařízení. Tato data jsou využívána k zaznamenávání trasy pohybu uživatele během fyzické aktivity, například chůze nebo běhu.

Dále je použit akcelerometr a krokomeř, konkrétně pomocí knihovny `pedometer`, která umožňuje sledovat počet kroků a detekovat chůzi či běh. Tyto senzory pracují na základě měření změn zrychlení zařízení a jsou energeticky úsporné, což je ideální pro dlouhodobé používání.

Součástí řešení je také vizualizace aktivity na mapě pomocí knihoven `flutter_map` a `latlong2`. Tyto knihovny umožňují nejen zobrazení polohy na mapových podkladech, ale i vykreslení trasy pohybu uživatele, čímž se zvyšuje srozumitelnost a přehlednost dat.

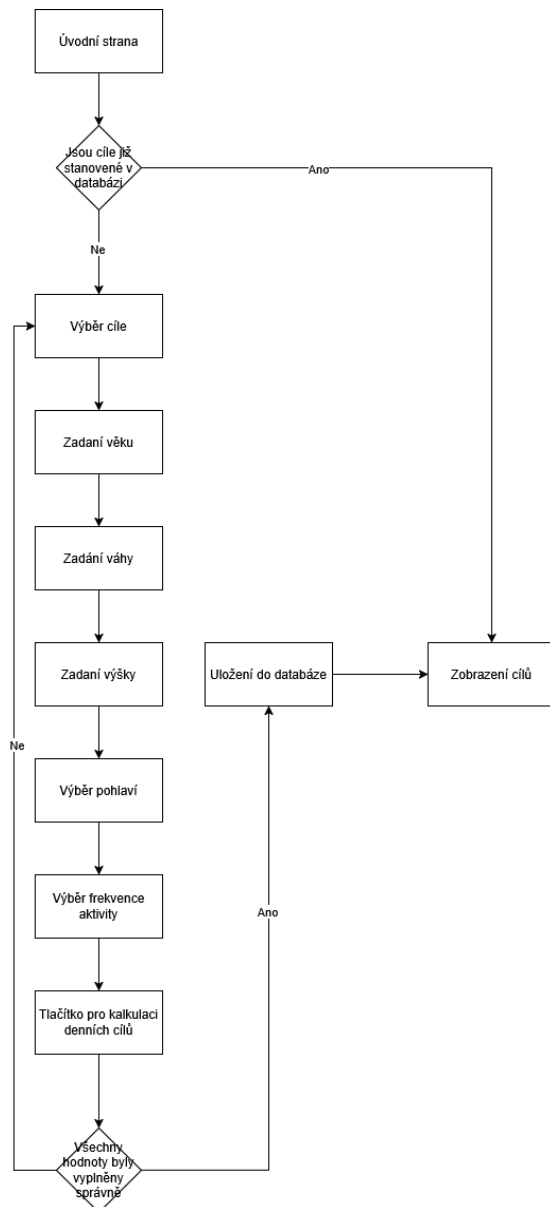
Pro správné fungování těchto funkcí bylo rovněž implementováno získání uživatelského oprávnění pro přístup k poloze a pohybovým sensorům, což je důležité z hlediska ochrany soukromí a bezpečnosti dat. Uživatel je vyzván, aby toto oprávnění udělil při prvním spuštění funkce, která je vyžaduje.

2.5 UML Diagram aplikace

V této části jsou uvedeny UML diagramy, které přehledně znázorňují základní funkčnost jednotlivých částí mobilní aplikace. Diagramy byly vytvořeny pomocí nástroje Draw.io a slouží k ilustraci hlavních komponent, jejich vzájemné interakce a posloupnosti uživatelských akcí. Každý z diagramů se zaměřuje na konkrétní stránku či část aplikace a pomáhá pochopit její vnitřní logiku a navigační strukturu. (13)

2.5.1 Diagram úvodní stránky

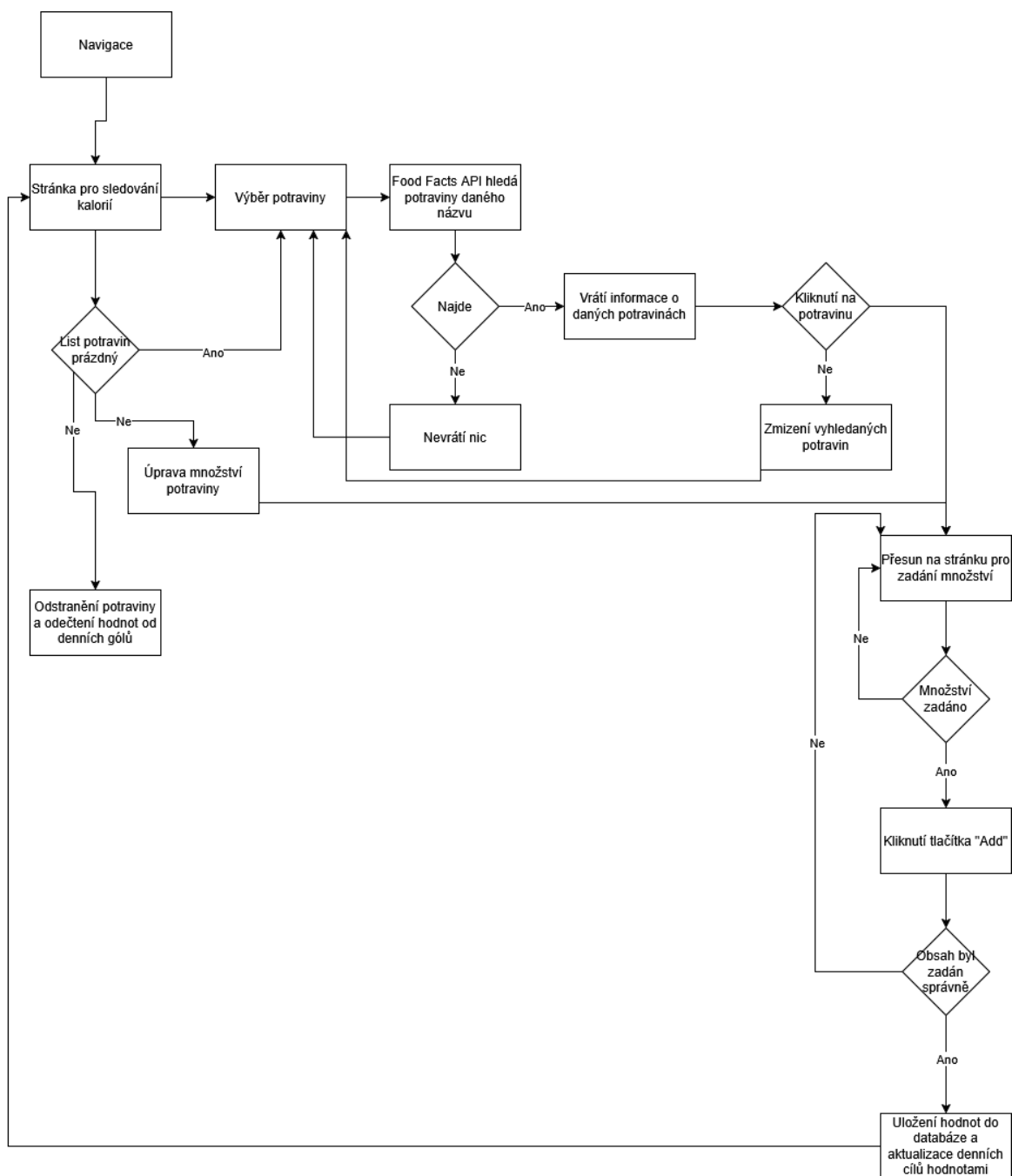
První UML diagram se zaměřuje na úvodní obrazovku aplikace, která představuje vstupní bod pro uživatele. Diagram znázorňuje jednotlivé kroky, které uživatel podniká od spuštění aplikace až po přechod na další stránky.



Obrázek 1: UML diagram úvodní stránky (zdroj: vlastní)

2.5.2 Diagram stránky pro sledování kalorií

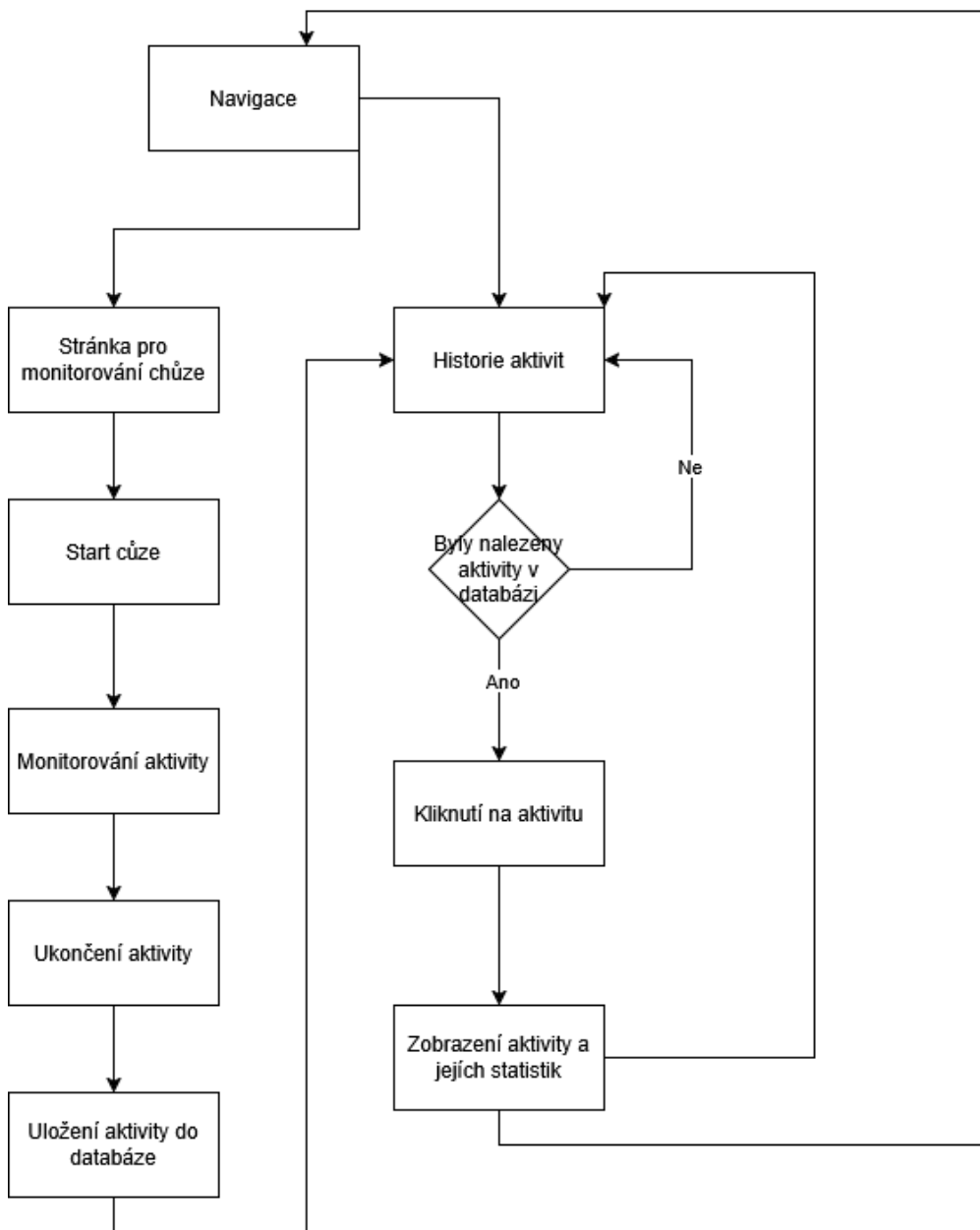
Druhý UML diagram znázorňuje komplexní logiku stránky určené pro sledování denního příjmu kalorií a živin. Tento diagram reflektuje klíčové funkce, jako je vyhledávání potravin pomocí API Open Food Facts, zadávání množství ve formě gramů, přidávání potravin do denního seznamu a jejich případné odebrání. Zahrnuje také mechanismus automatické aktualizace součtů jednotlivých nutričních hodnot a zobrazení výsledků pomocí progres barů. Diagram rovněž zachycuje uložení a načítání dat prostřednictvím HiveDB, čímž zajišťuje uchování uživatelských údajů i po uzavření aplikace.



Obrázek 2: UML diagram stránky pro sledování kalorií (zdroj: vlastní)

2.5.3 Diagram třetí a čtvrté stránky

Poslední diagram spojuje funkce dvou propojených stránek – záznamu a přehledu fyzické aktivity. Z důvodu jejich menší rozsáhlosti a vzájemné návaznosti jsou zobrazeny společně. Diagram ukazuje, jak aplikace zachytává fyzický pohyb uživatele prostřednictvím senzorů pohybové aktivity a GPS, a jakým způsobem dochází k ukončení a uložení aktivity. Dále zobrazuje proces, jak si uživatel může zobrazit historii již uložených aktivit a případně s nimi dále pracovat. Vše je navrženo tak, aby byla data přístupná offline i online, opět za využití databáze HiveDB.



Obrázek 3: UML diagram třetí a čtvrté stránky (zdroj: vlastní)

2.6 Funkční požadavky

Funkční požadavky určují konkrétní funkce, které má aplikace poskytovat. Mezi hlavní patřily:

- Generování personalizovaného jídelního plánu na základě údajů uživatele (věk, pohlaví, výška, váha, úroveň aktivity, cíl).
- Výpočet doporučeného denního příjmu kalorií a makroživin (bílkoviny, sacharidy, tuky, vláknina).
- Možnost zaznamenávat a sledovat denní příjem potravy pomocí databáze potravin (Open Food Facts API).
- Sledování fyzické aktivity – konkrétně chůze – včetně počtu kroků, trvání a trasy s využitím GPS.
- Ukládání všech dat lokálně pomocí databáze Hive.
- Správa oprávnění k přístupu k poloze a senzorům pohybu v souladu s pravidly pro Android.

Všechny výše uvedené funkční požadavky byly implementovány a ověřeny pomocí testování. Aplikace tak poskytuje ucelený přehled o výživě a základní pohybové aktivitě.

2.7 Možnosti dalšího rozšíření

Původně byly plánovány i další funkce, které však z časových nebo technických důvodů nebyly realizovány:

- Skenování čárových kódů a vizuální rozpoznávání potravin pro rychlejší zadávání – vyžadovalo by integraci pokročilých knihoven a služeb třetích stran.
- Synchronizace s nositelnými zařízeními (např. chytré hodinky) pro automatický přenos dat o aktivitě.
- Pokročilé vizualizace – grafické zobrazení vývoje hmotnosti, kalorického příjmu nebo fyzické aktivity.
- Možnost evidence různých typů tréninků (běh, cyklistika, posilování apod.).

Tyto funkcionality představují příležitost pro další rozvoj aplikace, a jejich implementace by mohla výrazně zvýšit uživatelskou hodnotu a konkurenceschopnost řešení.

2.8 Porovnání s existujícími aplikacemi

Při porovnání s konkurenčními aplikacemi na trhu se tato aplikace odlišuje především svou schopností propojit dvě klíčové oblasti zdravého životního stylu – výživu a pohyb – do jednoho přehledného a jednoduše ovladatelného rozhraní. Zatímco většina dostupných aplikací se specializuje pouze na jednu z těchto domén, tato práce kladla důraz na komplexní přístup k uživatelským potřebám.

Například aplikace jako Fitify nebo Nike Training Club se zaměřují výhradně na tréninkové plány, vedení cvičení a sledování sportovních výkonů, avšak zcela opomíjejí oblast stravování. Uživatelé těchto aplikací musí výživu řešit zvlášť, což často vede k potřebě instalace další aplikace nebo ručnímu zaznamenávání údajů, čímž dochází k roztříštěnosti uživatelského zážitku.

Na druhé straně existují aplikace jako Yazio, které se zaměřují především na sledování kalorického příjmu, makroživin a hubnutí, ale nenabízejí žádné sofistikované nástroje pro sledování pohybové aktivity, kromě základního propojení se zdravotními aplikacemi systému. V případě, že uživatel hledá i evidenci svých sportovních výkonů, opět je nucen využít jiný software.

Tato aplikace se snaží vyplnit zmíněnou mezeru na trhu tím, že integruje výživové plánování a současně základní formu sledování pohybové aktivity, konkrétně chůze, pomocí GPS a senzorů pohybu. Díky tomu má uživatel ucelený přehled o svém zdraví a životním stylu v jedné aplikaci bez nutnosti přepínání mezi různými nástroji.

Je však třeba podotknout, že oproti některým zavedeným konkurentům zůstává aplikace v určitých aspektech funkcionality omezená. Zejména v oblasti sportovní aktivity nenabízí tak širokou škálu možností – například chybí možnost zaznamenávání různých typů tréninků, propojení s nositelnou elektronikou (např. chytré hodinky) nebo analýzy srdečního tepu a dalších fyziologických dat. Tyto nedostatky však mohou být považovány spíše za prostor pro další rozšíření aplikace v budoucím vývoji.

3. Vývoj a realizace aplikace

Cílem této bakalářské práce je navrhnout a implementovat mobilní aplikaci pro sledování konzumace potravin a pohybové aktivity. (15) Aplikace bude vyvinuta pomocí multiplatformního frameworku Flutter a bude využívat lokální databázi Hive pro ukládání uživatelských dat.

Hlavním cílem je vytvořit intuitivní a efektivní nástroj, který uživatelům umožní snadno zaznamenávat svůj denní kalorický příjem a výdej a získat přehled o svých stravovacích a pohybových návycích. Důraz bude kladen na uživatelskou přívětivost, rychlost odezvy aplikace a efektivní práci s daty.

Pro dosažení tohoto cíle budou splněny následující dílčí úkoly:

- ✓ **Analýza existujících řešení** – průzkum dostupných aplikací pro sledování stravy a pohybu, identifikace jejich silných a slabých stránek.
- ✓ **Návrh aplikace** – specifikace klíčových funkcí, návrh uživatelského rozhraní a datového modelu.
- ✓ **Implementace aplikace** – vývoj mobilní aplikace ve Flutteru, integrace databáze Hive pro lokální ukládání dat.
- ✓ **Testování a optimalizace** – ověření funkčnosti, odstranění chyb a optimalizace výkonu aplikace.
- ✓ **Zhodnocení výsledků a návrh možného rozšíření** – posouzení dosažených výsledků a návrhy pro budoucí rozvoj aplikace.

Tato práce tak poskytne jak teoretický základ v oblasti sledování stravy a pohybu, tak praktickou realizaci funkční mobilní aplikace.

3.1 Instalace

Průběh instalace je velice jednoduchý, vše, co uživatel musí udělat je stáhnout instalační soubor, otevřít ho a aplikace se sama nainstaluje. Tento soubor lze jednoduše vytvořit ve vscode pomocí příkazu flutter build apk --release. V případě, že by uživatel potřeboval aktualizovat aplikaci novější verzí stačí opět pouze rozkliknout instalační soubor a ten provede veškeré aktualizace sám.

3.2 Popis funkce aplikace

V této části se popisuje, jak se aplikace operuje, všechny její funkce a jak jimi může uživatel procházet.

3.3 Úvodní strana

3.3.1 Úvod

První stránka, kterou uživatel uvidí poté co poprvé zapne tuto aplikaci je stránka pro výpočet denního cíle kalorií, bílkovin, tuků, sacharidů a vlákniny na základě individuálních parametrů zadaných uživatelem. Tyto informace aplikace zpracuje a použije je k výpočtu, vezme v potaz uživatelův věk, váhu, výšku, cíl a úroveň fyzické aktivity. Výsledkem je doporučený kalorický příjem a jeho rozdělení do jednotlivých mikroživin.

3.3.2 Funkcionalita

Jako první možnost stránka umožňuje uživateli vybrat jeho fitness cíl, a to, jestli chce uživatel zhubnout, nabrat svalovou hmotu anebo udržet váhu. Dále uživatel zadá svůj věk v letech, s tím že aplikace kontroluje, zda nebylo zadané negativní číslo, věk větší než sto anebo text místo čísla. Dalším údajem je uživatelova váha, která je zadána v kilogramech, a i v tomto poli probíhá kontrola, zda input uživatele je číslo a zda je větší než nula. Po zadání váhy je třeba zadat výšku, a to v centimetrech, a i u tohoto pole probíhá kontrola inputu jako v předešlých polích. Dále je také nezbytné, aby uživatel zadal své pohlaví z combo boxu, jelikož pohlaví je také potřebné vzít v potaz při výpočtu plánu uživatele. Poslední informace, kterou musí uživatel zadat je jeho úroveň aktivity, a to z combo boxu, který má pět různých možností: Sedentary (neaktivní životní styl), Lightly active (Mírně aktivní), Moderately active (Středně aktivní), Very active (Velice aktivní), Super active (Super aktivní).

Po zadání všech těchto informací může uživatel kliknout na tlačítko Calculate, které vezme v potaz všechny zadané hodnoty a vypočítá z nich doporučený denní kalorický příjem, optimální množství bílkovin, tuků, sacharidů a vlákniny a zobrazí grafické rozložení mikroživin.

3.3.3 Použité výpočetní vzorce

Poté co uživatel zadá všechny potřebné informace a aplikace se ujistí, že žádné pole nezůstalo prázdné nebo nebyly zadány ve špatném formátu, je začne zpracovávat.

Nejprve se vypočítá bazální metabolický výdej (BMR), což je množství kalorií, které tělo spálí v klidovém stavu. Pro výpočet BMR se používá Mifflin-St Jeorova rovnice, a to ve dvou různých variantách, jelikož výpočet BMR vypadá odlišně pro muže a pro ženy. Výpočet pro

muže vypadá následovně: $BMR = 10 * \text{váha (kg)} + 6.25 * \text{výška (cm)} - 5 * \text{věk (roky)} + 5$. Výpočet pro ženy vypadá následně s malým rozdílem: $BMR = 10 * \text{váha (kg)} + 6.25 * \text{výška (cm)} - 5 * \text{věk (roky)} - 161$. Výpočet se provádí s tímto rozdílem z důvodu fyziologických rozdílů mezi pohlavími, které ovlivňují energetickou spotřebu těla. Muži mají více svalové hmoty, zatímco ženy mají více tělesného tuku a jelikož svaly spalují více energie než tuk, tak je nutné, aby muži s vyšším podílem svalové hmoty měli vyšší bazální příjem.

Dále je nutné vypočítat celkové denní výdaje (TDEE). K výpočtu této hodnoty je třeba BMR, které poté vynásobíme úrovní zadané aktivity uživatelem. Využijeme jednoduchého vzorce $TDEE = BMR * \text{koeficient aktivity}$. Koeficient aktivity získáme z následující tabulky. (16)

Úroveň aktivity	Koeficient aktivity	Popis aktivity
Sedavý	1.2	žádné cvičení
Lehce aktivní	1.375	1-3 dny cvičení týdně
Středně aktivní	1.55	3-5 dní cvičení týdně
Velmi aktivní	1.725	6-7 dní cvičení týdně
Extrémně aktivní	1.9	sportovci, těžká práce

Tabulka 1: Tabulka koeficientů aktivity

Posledním krokem jest vypočítat doporučený příjem makroživin. Doporučené množství makroživin se odvíjí od cíle, který si uživatel vybral na začátku. (17)(18)

Jako první vypočítáme denní počet bílkovin. Bílkoviny jsou klíčové pro regeneraci svalů a celkovou tělesnou funkci. Pro tento výpočet využijeme následující vzorec: $\text{Bílkoviny (g)} = \text{váha (kg)} * \text{doporučený rozsah}$. Doporučené množství závisí na úrovni aktivity.

Cíl	Doporučený příjem bílkovin (g/kg tělesné hmotnosti)
Hubnutí	1.6–2.2 g/kg
Udržení váhy	1.2–2.0 g/kg
Nabírání svalové hmoty	1.8–2.5 g/kg

Tabulka 2: Tabulka úrovně aktivity

Tuky jsou důležitým zdrojem energie a podporují hormonální rovnováhu. Doporučený příjem se pohybuje mezi 20-35 % celkového energetického příjmu. Pro výpočet tuků použijeme vzorec: $\text{Tuky (g)} = (TDEE * \text{procento tuků}) / 9$. Použijeme číslo 9 jelikož to je energetická hodnota jednoho gramu tuku.

Sacharidy slouží jako hlavní zdroj energie a jejich množství se dopočítává podle zbývajících energetického příjmu po odečtení bílkovin a tuků, použijeme následující vzorec: $\text{Sacharidy (g)} = (\text{TDEE} - (\text{Bílkoviny} * 4 + \text{Tuky} * 9)) / 4$. Číslo 4 použijeme, jelikož to je energetická hodnota jednoho gramu sacharidů.

Vláknina je klíčová pro správné trávení, regulaci, krevního cukru a udržení zdravého střevního mikrobiomu. Její doporučené množství se stanovuje na základě celkového energetického příjmu (TDEE). K výpočtu využijeme vzorce: $\text{Vláknina (g)} = \text{TDEE} * 0.014$.

3.3.4 Ukládání do databáze

Poté co jsou všechny nutné informace vypočítány, je třeba je uložit na nějaké místo kam bude mít aplikace stále přístup, jinak by tyto hodnoty byly zapomenuty okamžitě po změně okna. Pro ukládání je používána databáze Hive. Hive je lehká a rychlá NoSQL databáze pro Flutter/Dart, která umožňuje ukládání strukturovaných dat přímo na zařízení uživatele. Hive používá key-value (klíč-hodnota) strukturu pro ukládání dat. To znamená, že každá hodnota je přiřazena unikátnímu klíči v databázi. V našem případě se ukládají data uživatele do tzv. Boxu, což je obdoba tabulky v relačních databázích. Box je kontejner pro ukládání dat v Hive. Lze jej chápat jako soubor, kde se uchovávají páry klíč-hodnota. Boxy mohou být otevřené a zavřené podle potřeby aplikace. Klíče jsou unikátní řetězce nebo číselné hodnoty, které identifikují záznam.

3.4 Stránka pro sledování kalorií

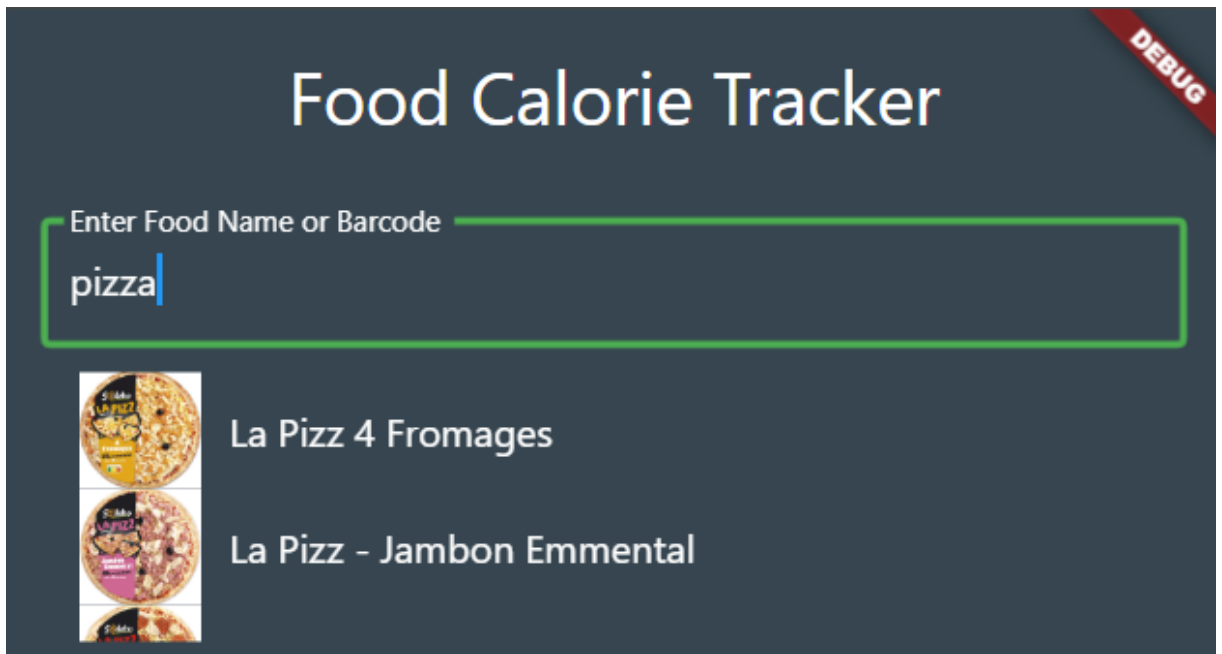
3.4.1 Úvod

Tato stránka umožňuje uživateli vyhledávat potraviny pomocí Open Food Facts API, přidávat je do svého denního seznamu konzumovaných potravin a sledovat svůj pokrok směrem k denním nutričním cílům.

3.4.2 Vyhledávání potravin

Na vrchu stránky se nachází textové okno, které umožňuje uživateli vyhledávat potraviny a nápoje s jejich potřebnými detaily. Poté co uživatel napíše nějaký text do tohoto okna aplikace vyšle API požadavek na Open Food Facts databázi, aby navrátila potraviny s jejich obrázky, korespondující se zadaným textem ve vyhledávacím okně. Pokud aplikace obdrží kód odpovědi 200, tak byly informace z API obdrženy v pořádku. Informace o dané potravine aplikace obdrží ve formátu JSON, ze kterého si poté vybere název a obrázek produktu, který se zobrazí v nápovědě pod vyhledávacím oknem. Pokud aplikace nezobrazí žádné potraviny poté co se je uživatel pokusil vyhledat mohla nastat jedna z následujících chyb: Uživatel není

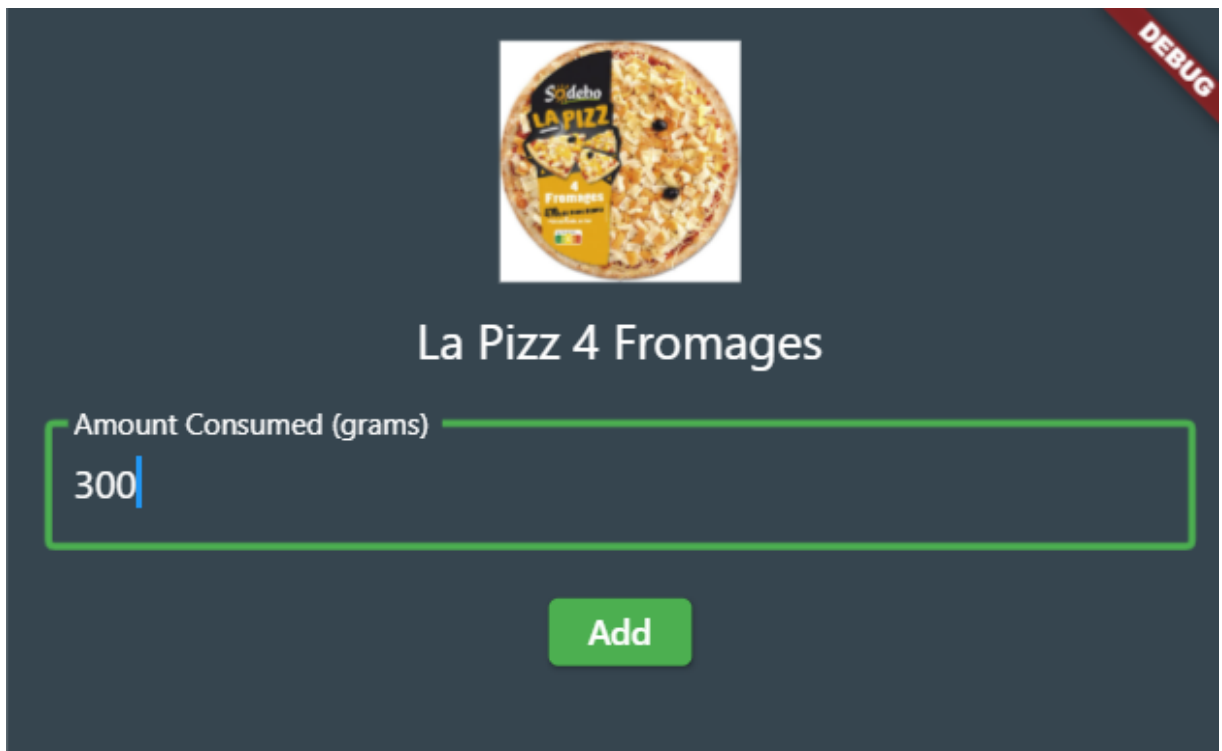
připojen k internetu, toto znemožňuje aplikaci dosáhnout Open Food Facts databáze a navrátit potřebné informace. Druhým důvodem, proč nebyly zobrazeny žádné produkty je, že žádný produkt se zadaným názvem se nenachází v Open Food Facts databázi, a tudíž aplikace neobdrží žádné informace.



Obázek 4: Vyhledávání potravin (zdroj: vlastní)

3.4.3 Přidávání potravin do seznamu

Poté co uživatel klikne na potravinu, kterou si přeje přidat tak je přesměrován na oddělenou stránku. Na této stránce musí zadat množství produktu v gramech, aby mohla aplikace spočítat kolik kalorií, bílkovin, tuků, vlákniny a sacharidů z daného produktu musí připočítat k dennímu počtu. Tyto informace samozřejmě získá z Open Food Facts databáze. Když uživatel zadá, kolik daného produktu zkonsumoval může kliknout na tlačítko OK, které nejprve provede kontrolu, zda zadaná hodnota je kladné číslo a pokud tomu tak je, tak je uživatel vrácen zpět na předešlou stránku kde se automaticky vyprázdní obsah vyhledávacího okna i nápovědy a v seznamu zkonsumovaných potravin se objeví potravina, kterou uživatel přidal společně s názvem, obrázkem a všemi mikroživinami. Počet kalorií a mikroživin se odvíjí od zadaného počtu zkonsumovaných gramů produktu.



Obrázek 5: Zadávání počtu zkonsumovaných gramů potraviny (zdroj: vlastní)

3.4.4 Seznam zkonsumovaných potravin

V polovině této stránky se nachází seznam s nadpisem „Foods Consumed“, do kterého se přidávají veškeré potraviny, které uživatel během dne zkonsumuje. V tomto seznamu se zobrazí obrázek dané potraviny, pokud se nachází v databázi, když potraviny nemá v databázi přiřazen žádný obrázek, tak aplikace využije zástupný obrázek. Napravo od obrázku se nachází plný název potraviny. Pod názvem se zobrazují počty kalorií, bílkovin, tuků, sacharidů a vápníku vypočítány podle zadaného počtu gramů na předchozí stránce.

Tento seznam se v aplikaci vytváří pomocí widgetu `ListView.builder`, který dynamicky generuje položky seznamu. Bere dva parametry, první z nich je `itemCount`, který je určen délkou seznamu jídel, která jsou držena v listu. Druhý parametr je `itemBuilder`, který bere v potaz context aplikace a index, podle kterého generuje jednotlivé položky. Každá položka je reprezentována widgetem `ListTile`, který obsahuje název potraviny, nutriční hodnoty, tlačítko pro odstranění a možnost editace.

V případě, že by si uživatel přál změnit počet gramů zkonsumované potraviny, má možnost kliknout kamkoliv na potravinu v seznamu, kromě tlačítka na odstranění nacházejícího se úplně vpravo. Poté co je potravina v seznamu kliknuta, uživatel je převeden opět na stránku se zadáváním gramů potraviny, zde má možnost přepsat původní hodnotu a poté změnu potvrdit

a vrátit se zpět na původní stránku. Po provedení této změny se také upraví veškeré údaje jako je počet kalorií, mikroživin jak v seznamu zkonsumovaných potravin, tak v progres barech, které zobrazují celkový počet zkonsumovaných kalorií a mikroživin za celý den.

Na pravé straně potravin v seznamu se nachází tlačítko s ikonou popelnice, které umožňuje danou potravinu odstranit ze seznamu společně s hodnotami kalorií a mikroživin, které přinášelo do denního cíle těchto hodnot. Odstranění je možno pomocí `onPressed` vlastnosti dané ikonky. K této vlastnosti dojde poté co je ikonku stisknuta, v daný okamžik se provede obsah metody `_removeFood`, která je touto akcí vyvolána. Tato metoda si přebírá index, který určí které jídlo ze seznamu odstranit a také přebírá hodnotu `food`, aby aplikace věděla jaký počet kalorií a mikroživin musí odstranit od celkového denního počtu.

3.4.5 Ukazatele průběhu

Tato sekce stránky se nachází v samotném spodku obrazovky a skládá se ze čtyř ukazatelů procesů. Je to jedna z nejdůležitějších částí této aplikace, jelikož se stará o spravování denních cílů a jejich modifikací. Cílem je poskytnout uživateli rychlý přehled o tom, jak blízko je k dosažení svého denního cíle pro kalorie, bílkoviny, tuky, sacharidy a vlákninu.

Každý progres bar zobrazuje aktuální hodnotu přijatých živin v porovnání s cílovou hodnotou stanovenou uživatelem. Tyto hodnoty se automaticky aktualizují na základě potravin, které uživatel přidal do aplikace.

Progres bary jsou vykresleny uvnitř sloupce (Column), kde každý jednotlivý řádek představuje konkrétní živinu – kalorie, bílkoviny, tuky, sacharidy a vlákninu. Každý řádek obsahuje název živiny a aktuální stav, například že bylo splněno 1200 z 2000 kalorií („Calories: 1200/2000 kcal“). Dále zobrazuje grafickou reprezentaci progresu pomocí `LinearProgressIndicator`, tato komponenta slouží jako progres bar, který se naplňuje s tím kolik dané živiny uživatel zkonsumoval. Progres bar také barevně odlišuje na základě dosažení cíle.

Funkce `_buildProgressBar` dynamicky vytváří jednotlivé progres bary a počítá jejich aktuální hodnotu vůči cílové hodnotě. Tento poměr se použije k nastavení hodnoty `LinearProgressIndicator`.

```

Widget _buildProgressBar(String label, double current, int
goal, String unit) {
  int safeCurrent = current < 0 ? 0 : current.round();
  double progress = (goal > 0) ? safeCurrent / goal : 0;

  return Column(
    children: [
      Center(
        child: Text(
          '$label: $safeCurrent / $goal $unit',
          style: const TextStyle(color: Colors.white),
          textAlign: TextAlign.center,
        ),
      ),
      const SizedBox(height: 5),
      LinearProgressIndicator(
        backgroundColor: const Color.fromARGB(255, 239, 189,
189),
        minHeight: 7,
        value: progress.clamp(0.0, 1.0),
        valueColor:
AlwaysStoppedAnimation<Color>(_getProgressColor(progress)),
      ),
      const SizedBox(height: 15),
    ],
  );
}

```

Pokud například uživatel přijal 1200 kcal z denního cíle 2000 kcal, progres bar bude naplněn na 60 %.

```
double progress = (goal > 0) ? safeCurrent / goal : 0;
```

Tento výpočet zajišťuje, že pokud je hodnota goal nulová (což by mohlo způsobit chybu dělením nulou), nastaví se hodnota progresu na 0.

Barva progres baru se mění v závislosti na tom, jak blízko je uživatel ke svému cíli. Tato změna je řízena funkcí `_getProgressColor(progress)`, která přizpůsobuje barvu na základě dosaženého procenta, pokud je progres pod 80% tak je barva progres baru oranžová, pokud mezi 80% až 100% tak zelená, signalizující, že byl cíl splněn a pokud nad 100% tak se bar změnil v červený, signalizující, že uživatel zkonsumoval více než by měl.

3.4.6 Zdroj dat – Hive

Všechny zobrazované hodnoty jsou uloženy a načítány z Hive databáze, což umožňuje aplikaci udržovat data i po jejím zavření a znovuotevření. Každý přidání pokrm zvyšuje

odpovídající hodnoty živin a aktualizuje progres bary v reálném čase. Tyto hodnoty se aktualizují při přidání nebo odebrání potravin v aplikaci a ukládají se lokálně, aby je bylo možné použít i po restartu aplikace.

3.5 Stránka pro monitorování aktivit

3.5.1 Úvod

Tato stránka slouží ke sledování aktivit uživatele a zaznamenávání relevantních údajů, jako je počet kroků, trasa pohybu a doba trvání aktivity. Tyto údaje jsou následně ukládány do lokální databáze Hive, kde si je uživatel může zpětně prohlédnout. Sledování aktivity je realizováno pomocí různých technologií, včetně senzorů mobilního zařízení pro detekci kroků a GPS pro záznam polohy. Mapová vizualizace umožňuje zobrazit trasu, kterou uživatel při chůzi urazil, a poskytuje tak lepší přehled o jeho fyzické aktivitě.

3.5.2 Získání oprávnění

Při spuštění stránky aplikace požaduje od uživatele oprávnění k přístupu k poloze a k rozpoznávání aktivity. Funkce `_requestLocationPermission()` zajišťuje, že aplikace může získat přesnou GPS polohu zařízení, což umožňuje sledování trasy na mapě. Druhá funkce `_requestPermissions()` pak žádá o oprávnění k detekci kroků prostřednictvím senzoru akcelerometru. Pokud uživatel tato oprávnění neudělí, některé funkce aplikace nebudou moci fungovat správně.

3.5.3 Sledování aktivity

Po získání oprávnění aplikace určí aktuální polohu uživatele a použije ji k nastavení středu mapy. To je realizováno funkcí `_getCurrentLocation()`, která získá GPS souřadnice a posune mapu na tuto polohu pomocí `_mapController.move()`. Pokud uživatel začne aktivitu, funkce `_startTracking()` zahájí proces sledování. Nejprve vynuluje počet kroků a seznam souřadnic trasy. Následně spustí časovač pro měření délky aktivity a začne naslouchat změnám polohy uživatele. Při každé změně polohy je nový bod přidán do seznamu `_routeCoords`, čímž se trasa postupně vykresluje na mapě.

Kromě záznamu polohy aplikace sleduje také počet kroků uživatele pomocí funkce `_initStepCounter()`. Ta naslouchá streamu kroků pomocí knihovny Pedometer a při každé změně aktualizuje hodnotu `_stepCount`, což uživateli poskytuje přehled o jeho chůzi v reálném čase.

Jakmile uživatel aktivitu ukončí, může stisknout tlačítko, které zavolá funkci `_stopTracking()`. Tato funkce nejprve zastaví časovač a následně uloží záznam o aktivitě do Hive databáze ve formě objektu `WalkingActivity`. Tento objekt obsahuje počet kroků, celkovou délku aktivity v sekundách a seznam souřadnic trasy. Po úspěšném uložení aplikace automaticky přesměruje uživatele na stránku `ActivityHistoryScreen`, kde si může zobrazit seznam všech svých minulých aktivit.

Uživatelské rozhraní této stránky je rozděleno na dvě hlavní části. Horní část tvoří mapa, která je implementována pomocí knihovny `FlutterMap`. Mapa vizualizuje trasu chůze pomocí vrstvy `PolylineLayer`, která propojuje jednotlivé zaznamenané body. Dále je zde vrstva `MarkerLayer`, která zobrazuje aktuální polohu uživatele jako červený pin. Jako mapový podklad aplikace využívá dlaždice `OpenStreetMap`.

Pod mapou se nachází statistiky a ovládací prvky. Uživatel vidí aktuální počet kroků a dobu trvání aktivity v minutách a sekundách. Dále se zde nachází tlačítko pro spuštění a ukončení aktivity, jehož barva se mění podle aktuálního stavu sledování – zelená pro spuštění a červená pro zastavení.

3.6 Historie aktivit

3.6.1 Úvod

Tato další stránka slouží k zobrazení historie všech zaznamenaných chodeckých aktivit uživatele. Data jsou načítána z lokální databáze Hive a zobrazována v přehledném seznamu. Každá zaznamenaná aktivita obsahuje informace o počtu kroků a délce trvání.

3.6.2 Popis stránky

Při inicializaci obrazovky se volá metoda `_loadActivities()`, která načítá všechna uložená data o aktivitách pomocí služby `HiveService`. Tyto data jsou následně uložena do seznamu `_activities`, který se používá k dynamickému vykreslení uživatelského rozhraní. Pokud nejsou k dispozici žádné zaznamenané aktivity, zobrazí se informační zpráva "No activities recorded yet." v centru obrazovky.

Pokud existují zaznamenané aktivity, jsou zobrazeny v seznamu pomocí widgetu `ListView.builder()`. Každá položka seznamu je reprezentována kartou (`Card`), která zobrazuje počet kroků, délku trvání aktivity a ikonu chůze. Kliknutím na kartu se uživatel dostane na stránku `ActivityDetailScreen`, kde se zobrazí detailní informace o vybrané aktivitě.

Stránka `ActivityDetailScreen` poskytuje podrobnější pohled na konkrétní chodeckou aktivitu. Obsahuje mapovou vizualizaci trasy, kterou uživatel během aktivity prošel. Mapu poskytuje knihovna `FlutterMap`, která využívá dlaždice `OpenStreetMap`. Trasa je vykreslena pomocí `PolylineLayer`, kde jednotlivé body představují GPS souřadnice zaznamenané během chůze. Pod mapou jsou uvedeny další detaily o aktivitě. Uživatel zde vidí počet kroků a dobu trvání aktivity ve formátu minut a sekund. Tyto údaje jsou zobrazeny ve vizuálně přívětivém stylu s bílým textem na tmavém pozadí.

4 Unit Testy

4.1 Úvod

Tato dokumentace popisuje testy napsané v rámci ověření funkčnosti několika klíčových komponent aplikace. Testy se zaměřují na ověření správného výpočtu hodnot v rámci fitness plánu, správné funkce sledování kalorií, a interakci s databází `Hive`. (10) Testy jsou napsány v jazyce `Dart` s použitím knihovny `flutter_test` pro jednotkové testování a `mockito` pro simulaci interakcí s externími službami. (19; 20)

4.2 Fitness Plan Service

Tento testovací soubor ověřuje správnost výpočtů pro fitness plán, který zahrnuje makroživiny na základě specifikovaných parametrů (věk, váha, výška, pohlaví, úroveň aktivity, cíl). Testy zahrnují:

- ✓ Výpočet makroživin pro muže, který chce nabírat svalovou hmotu
- ✓ Výpočet makroživin pro ženu, která chce zhubnout a má sedavý životní styl
- ✓ Výpočet pro muže s cílem udržet váhu při velmi aktivním životním stylu
- ✓ Hraniční testy pro velmi nízkou váhu a výšku
- ✓ Hraniční testy pro velmi vysokou váhu a výšku

Každý test kontroluje, že výsledky výpočtů odpovídají očekávaným hodnotám pro kalorie, bílkoviny, tuky, sacharidy a vlákninu. (21)

4.3 Food Calorie Tracker Service

Tento soubor obsahuje testy pro službu, která sleduje přidání potravin a jejich kalorické hodnoty, makroživiny a vlákninu. Testy zahrnují:

- ✓ Přidání jídla
- ✓ Odebrání více potravin
- ✓ Odebrání poslední potraviny
- ✓ Kontrola logiky barvy pro sledování pokroku

4.4 Hive Database Service

Testy v tomto souboru ověřují správnost interakce s databází Hive, která je použita pro ukládání aktivit, jako je chůze, do místní databáze. Testy zahrnují:

- ✓ Získání hodnoty z Hive
- ✓ Uložení hodnoty do Hive
- ✓ Odstranění klíče z Hive
- ✓ Uložení aktivity do Hive
- ✓ Získání všech aktivit z Hive

Testy byly napsány tak, aby pokryly různé scénáře, včetně hraničních případů a zajišťují robustnost a stabilitu aplikace. (22)

ZÁVĚR

Tato bakalářská práce se zaměřila na návrh, implementaci a testování mobilní aplikace pro sledování fyzické aktivity a kalorického příjmu. Cílem bylo vytvořit efektivní a uživatelsky přívětivý systém, který by umožnil uživatelům sledovat své kroky, trasu a zároveň evidovat příjem kalorií a živin.

V rámci práce byl analyzován současný stav mobilních aplikací pro sledování aktivity a výživy, přičemž bylo identifikováno několik klíčových požadavků na funkcionalitu. Na základě této analýzy byla navržena architektura aplikace s využitím moderních technologií, jako je Flutter pro multiplatformní vývoj, Hive pro lokální databázové úložiště a knihovny jako LatLong2 pro práci s geografickými souřadnicemi.

Implementace aplikace byla podrobena důkladnému testování, které zahrnovalo unit testy využívající framework flutter_test a mockito pro simulaci interakcí s databází Hive. Výsledky testování potvrdily správnost klíčových funkcionalit, včetně ukládání a načítání dat o aktivitách, manipulace s databázovými záznamy a výpočtu kalorického příjmu.

Přestože aplikace splnila stanovené cíle, existují možnosti pro její další rozšíření. Mezi potenciální vylepšení patří integrace s nositelnými zařízeními, implementace synchronizace s cloudovými službami a přidání funkcí pro analýzu dlouhodobého vývoje fyzické aktivity a výživových návyků uživatelů.

Celkově lze konstatovat, že práce úspěšně demonstrovala možnosti využití moderních mobilních technologií pro sledování zdravého životního stylu a položila základy pro další vývoj aplikace v této oblasti.

POUŽITÁ LITERATURA

- (1) *Flutter Docs*. Online. Flutter Docs. 2025. Dostupné z: <https://docs.flutter.dev/>. [cit. 2025-03-14].
- (2) *Dart programming language*. Online. Dart programming language. 2025. Dostupné z: <https://dart.dev/docs>. [cit. 2025-03-14].
- (3) *Dart API Docs*. Online. Dart API Docs. N.d. Dostupné z: <https://pub.dev/documentation/hive/latest/>. [cit. 2025-03-14].
- (4) *Open Food Facts*. Online. Open Food Facts. 2025. Dostupné z: <https://world.openfoodfacts.org/data>. [cit. 2025-03-14].
- (5) *Flutter map Docs*. Online. Flutter map Docs. N.d. Dostupné z: <https://docs.fleaflet.dev/>. [cit. 2025-03-14].
- (6) *Latlong2 | Flutter package*. Online. Latlong2 | Flutter package. N.d. Dostupné z: <https://pub.dev/packages/latlong2>. [cit. 2025-03-14].
- (7) *Location | Flutter package*. Online. Location | Flutter package. N.d. Dostupné z: <https://pub.dev/packages/location>. [cit. 2025-03-14].
- (8) *Pedometer | Flutter package*. Online. Pedometer | Flutter package. N.d. Dostupné z: <https://pub.dev/packages/pedometer>. [cit. 2025-03-14].
- (9) *Permission_handler | Flutter package*. Online. Permission_handler | Flutter package. N.d. Dostupné z: https://pub.dev/packages/permission_handler. [cit. 2025-03-14].
- (10) MOORE, Kevin. *Mastering Flutter*. BPB Publications, 2025. ISBN 9365899176.
- (11) BIESSEK, Alessandro. *Flutter for Beginners*. Packt Publishing, 2019. ISBN 1788996089.
- (12) LI, Yishan. A performance comparison of SQL and NoSQL databases. Online. *ResearchGate*. 2013, s. 19. Dostupné z: <https://doi.org/10.1109/PACRIM.2013.6625441>.

- [cit. 2025-03-14].
- (13) *Draw.io*. Online. Draw.io. N.d. Dostupné z: <https://app.diagrams.net/>. [cit. 2025-03-14].
- (14) HOOBER, Steven a BERKMAN, Eric. *Designing Mobile Interfaces*. Online. Oreilly. 2011. Dostupné z: <https://www.oreilly.com/library/view/designing-mobile-interfaces/9781449318451/>. [cit. 2025-04-11].
- (15) *Visual Studio Code*. Online. Visual Studio Code. 2025. Dostupné z: <https://code.visualstudio.com/>. [cit. 2025-03-14].
- (16) *9round*. Online. 2025. Dostupné z: <https://www.9round.com/blog/how-to-calculate-a-calorie-deficit>. [cit. 2025-04-17].
- (17) GONZALEZ, Drew E. International society of sports nutrition position stand: tactical athlete nutrition. Online. *National Library of Medicine*. 2022, roč. 2022, č. 1, s. 267-315. Dostupné z: <https://doi.org/10.1080/15502783.2022.2086017>. [cit. 2025-03-14].
- (18) CLIN NUTR, Am J. A new predictive equation for resting energy expenditure in healthy individuals. Online. Dostupné z: <https://doi.org/10.1093/ajcn/51.2.241>. [cit. 2025-03-14].
- (19) *Testing Flutter apps*. Online. Testing Flutter apps. 2025. Dostupné z: <https://docs.flutter.dev/testing/overview>. [cit. 2025-03-14].
- (20) *An introduction to widget testing*. Online. An introduction to widget testing. N.d. Dostupné z: <https://docs.flutter.dev/cookbook/testing/widget/introduction>. [cit. 2025-03-14].
- (21) *Mockito | Flutter package*. Online. Mockito | Flutter package. N.d. Dostupné z: <https://pub.dev/packages/mockito>. [cit. 2025-03-14].
- (22) *TestAsyncUtils class*. Online. TestAsyncUtils class. N.d. Dostupné z: https://api.flutter.dev/flutter/flutter_test/TestAsyncUtils-class.html. [cit. 2025-03-14].

SEZNAM PŘÍLOH

Příloha A: Instalační soubor aplikace app_release.apk

PŘÍLOHA A: Instalační soubor aplikace

Vložit přílohu, popis přílohy A, zdroj.