

# ZÁVĚREČNÁ STUDIJNÍ PRÁCE

## dokumentace



# HabitTrack – Aplikace pro sledování návyků a zdraví

**Autor:** Miroslav Lubeník  
**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
se zaměřením na počítačové sítě a programování  
**Třída:** IT4  
**Školní rok:** 2025/2026



## PODĚKOVÁNÍ

*poděkování (například vedoucímu práce).*

.....

# PROHLÁŠENÍ

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 8. ledna 2026

.....  
*podpis autora práce*

## Abstrakt

Tato závěrečná práce se zabývá vytvořením mobilní aplikace HabitTrack pro sledování návyků a zdraví pomocí Flutter frameworku. Aplikace umožňuje uživatelům vytvářet a spravovat vlastní návyky, sledovat jejich denní plnění, zobrazovat statistiky pokroku a personalizovat prostředí aplikace. Aplikace je navržena s důrazem na moderní uživatelské rozhraní, offline fungování a intuitivní ovládání. Práce obsahuje popis teoretických základů, použité technologie, způsoby řešení a výsledky implementace.

**Klíčová slova** mobilní aplikace, sledování návyků, HabitTrack, personalizace, statistiky

**Abstract** This final thesis deals with the creation of a mobile application HabitTrack for tracking habits and health using the Flutter framework. The application allows users to create and manage their own habits, track their daily completion, display progress statistics, and personalize the application environment. The application is designed with emphasis on modern user interface, offline functionality, and intuitive operation. The thesis contains description of theoretical foundations, used technologies, solution approaches and implementation results.

**Keywords** mobile application, habit tracking, HabitTrack, personalization, statistics

# Obsah

<b>Úvod</b>	<b>7</b>
<b>1 Teoretická a metodická východiska</b>	<b>9</b>
1.1 Problematika sledování návyků . . . . .	9
1.2 Metodika sledování návyků . . . . .	9
1.3 Architektura mobilních aplikací . . . . .	10
1.4 Offline fungování aplikace . . . . .	10
<b>2 Využité technologie</b>	<b>11</b>
2.1 Flutter Framework . . . . .	11
2.2 Dart programovací jazyk . . . . .	11
2.3 SQLite databáze . . . . .	11
2.4 Další použité knihovny . . . . .	11
2.5 Path Provider . . . . .	11
2.6 Zdůvodnění výběru technologií . . . . .	12
<b>3 Způsoby řešení a použité postupy</b>	<b>13</b>
3.1 Návrh architektury aplikace . . . . .	13
3.2 Návrh databáze . . . . .	13
3.3 Návrh uživatelského rozhraní . . . . .	14
3.4 Implementační postupy . . . . .	14
3.5 Způsoby testování . . . . .	17
3.6 Řešení problémů . . . . .	18
<b>4 Výsledky řešení, výstupy, uživatelský manuál</b>	<b>19</b>
4.1 Splněné cíle práce . . . . .	19
4.2 Hlavní funkcionality aplikace . . . . .	19
4.3 Uživatelský manuál . . . . .	22
4.4 Technické specifikace . . . . .	24
<b>Závěr</b>	<b>25</b>
<b>Seznam použitých informačních zdrojů</b>	<b>27</b>
<b>Seznam obrázků</b>	<b>30</b>

# ÚVOD

Tato závěrečná práce se zabývá vytvořením mobilní aplikace HabitTrack pro sledování návyků a zdraví. Aplikace poskytuje uživatelům moderní a intuitivní nástroj pro správu jejich denních návyků s důrazem na vizualizaci pokroku a motivaci k dlouhodobé pravidelnosti.

Sledování návyků je důležitou součástí osobního rozvoje a zdravého životního stylu. Mobilní aplikace mohou uživatelům významně pomoci při budování a udržování pozitivních návyků prostřednictvím vizualizace pokroku, připomenutí a gamifikace. Aplikace HabitTrack umožňuje uživatelům vytvářet vlastní návyky, sledovat jejich denní plnění, zobrazovat statistiky pokroku a personalizovat prostředí aplikace podle svých preferencí.

Volba této problematiky byla motivována potřebou praktické aplikace pro osobní použití, zájmem o vývoj mobilních aplikací s komplexní funkcionalitou a snahou vytvořit aplikaci s moderním designem a uživatelsky přívětivým rozhraním, která pomůže uživatelům udržet si pozitivní návyky a sledovat jejich pokrok.

Hlavní cíle této závěrečné práce jsou: navrhnout a implementovat mobilní aplikaci pro sledování návyků, vytvořit intuitivní uživatelské rozhraní s moderním designem, implementovat systém pro ukládání dat s offline podporou, zajistit personalizaci aplikace (dark mode, barvy, profil), vytvořit systém statistik a vizualizace pokroku a implementovat notifikace pro připomenutí o návycích.

Práce je rozdělena do následujících kapitol. Kapitola 1 – Teoretická a metodická východiska popisuje problematiku sledování návyků a metodiky pro jejich efektivní sledování. Kapitola 2 – Využité technologie představuje konkrétní technologie, nástroje a knihovny použité při vývoji aplikace. Kapitola 3 – Způsoby řešení a použité postupy popisuje návrh architektury aplikace, databázového schématu a implementační postupy. Kapitola 4 – Výsledky řešení, výstupy, uživatelský manuál prezentuje finální aplikaci, její funkcionality a uživatelský manuál. Závěr shrnuje výsledky práce a navrhuje možná vylepšení.



# 1 TEORETICKÁ A METODICKÁ VÝCHODISKA

## 1.1 PROBLEMATIKA SLEDOVÁNÍ NÁVYKŮ

Sledování návyků je důležitou součástí osobního rozvoje a zdravého životního stylu. Návyky představují opakované chování, které se stává automatickým a pomáhá jedincům dosahovat dlouhodobých cílů. Efektivní sledování návyků umožňuje uživatelům identifikovat vzorce chování, měřit pokrok a udržovat motivaci k pravidelnému plnění stanovených cílů.

Moderní mobilní aplikace mohou uživatelům významně pomoci při budování a udržování pozitivních návyků prostřednictvím vizualizace pokroku, připomenutí a gamifikace. Aplikace HabitTrack byla navržena s důrazem na tyto aspekty, aby poskytla uživatelům komplexní nástroj pro správu jejich návyků.

## 1.2 METODIKA SLEDOVÁNÍ NÁVYKŮ

Pro efektivní sledování návyků je důležité stanovit jasné cíle, pravidelně zaznamenávat pokrok a vizualizovat výsledky. Aplikace HabitTrack implementuje několik klíčových principů:

- **Každodenní sledování** – uživatelé mohou označit splnění návyku pro konkrétní den.
- **Vizualizace pokroku** – kalendář s historií plnění a statistiky poskytuje přehled o pokroku.
- **Gamifikace** – systém achievementů motivuje uživatele k pravidelnému plnění návyků.
- **Připomenutí** – notifikace pomáhají uživatelům nezapomenout na plnění návyků.
- **Personalizace** – možnost nastavení barev, dark mode a profilové fotky zvyšuje uživatelskou zkušenosť.

## **1.3 ARCHITEKTURA MOBILNÍCH APLIKACÍ**

Pro vytvoření funkční mobilní aplikace je důležité navrhnout správnou architekturu, která zajistí oddělení logiky aplikace od prezentační vrstvy, umožní snadnou údržbu a rozšíření funkcionalit. Aplikace HabitTrack využívá čistou architekturu s rozdělením na vrstvy pro data, logiku a prezentaci.

## **1.4 OFFLINE FUNGOVÁNÍ APLIKACE**

Důležitým aspektem mobilní aplikace pro sledování návyků je možnost fungování bez síťového připojení. Uživatelé potřebují mít přístup ke svým datům kdykoli a kdekoli, bez ohledu na dostupnost internetového připojení. Aplikace HabitTrack ukládá všechna data lokálně na zařízení, což zajišťuje rychlý přístup a neustálou dostupnost funkcionalit.

## 2 VYUŽITÉ TECHNOLOGIE

### 2.1 FLUTTER FRAMEWORK

Aplikace byla vyvinuta pomocí Flutter frameworku [1], který umožňuje vytvářet cross-platform mobilní aplikace z jednoho zdrojového kódu. Flutter používá programovací jazyk Dart [2] a kompiluje aplikace do nativního kódu, což zajišťuje vysoký výkon.

### 2.2 DART PROGRAMOVACÍ JAZYK

Dart je objektově orientovaný programovací jazyk [2], který poskytuje statickou typovou kontrolu, asynchronní programování a automatickou správu paměti.

### 2.3 SQLITE DATABÁZE

Pro lokální ukládání dat byla použita SQLite databáze přes balíček `sqflite` [4], která umožňuje offline fungování aplikace a rychlý přístup k datům.

### 2.4 DALŠÍ POUŽITÉ KNIHOVNY

Aplikace využívá další knihovny pro konkrétní funkcionality: `shared_preferences` [5] pro ukládání nastavení, `flutter_local_notifications` [?] pro notifikace, `f1_chart` [?] pro vytváření grafů a `image_picker` pro výběr obrázků z galerie.

### 2.5 PATH PROVIDER

Balíček `path_provider` poskytuje přístup k systémovým cestám pro ukládání souborů. V aplikaci je použit pro ukládání profilových fotografií uživatelů na správné místo v systému souborů.

## 2.6 ZDŮVODNĚNÍ VÝBĚRU TECHNOLOGIÍ

Výběr technologií byl proveden s ohledem na požadavek na cross-platform kompatibilitu, potřebu offline fungování aplikace, snahu o moderní a atraktivní uživatelské rozhraní, dostupnost a kvalitu dokumentace a aktivitu komunity a podporu. Při výběru technologií byly využity zdroje jako oficiální dokumentace [3, ?], Flutter Cookbook [10], komunita na Stack Overflow [6] a GitHub [7] pro inspiraci z open source projektů. Všechny použité technologie jsou open-source a bezplatné, což umožňuje volné použití a distribuci aplikace.

## 3 ZPŮSOBY ŘEŠENÍ A POUŽITÉ POSTUPY

### 3.1 NÁVRH ARCHITEKTURY APLIKACE

Aplikace je strukturována podle principů čisté architektury [?] s oddělením vrstev. Struktura projektu je následující:

#### 3.1.1 Struktura projektu

- `lib/screens/` – obrazovky aplikace (UI vrstva).
- `lib/services/` – business logika a služby (databáze, notifikace).
- `lib/widgets/` – znovupoužitelné widgety.
- `lib/styles/` – centralizované styly a design systém.

#### 3.1.2 Design Patterns

V aplikaci jsou použity následující návrhové vzory [?]:

- **Singleton** – pro DatabaseHelper a NotificationService (zajišťuje jedinou instanci).
- **State Management** – pomocí StatefulWidget a setState pro lokální stav.
- **Repository Pattern** – pro práci s databází (abstrakce datové vrstvy).
- **Factory Pattern** – pro vytváření widgetů a stylů.

### 3.2 NÁVRH DATABÁZE

Databáze obsahuje následující tabulky:

### 3.2.1 Schéma databáze

- `users` – informace o uživatelích (id, email, nickname, password\_hash, profile\_photo\_path)
- `habits` – definice návyků (id, user\_id, name, description, color, icon, daily\_target, atd.)
- `habit_logs` – záznamy o plnění návyků (id, habit\_id, date, completed)
- `calendar_notes` – poznámky k datům (id, user\_id, date, note)
- `achievements` – odemčené achievementy (id, user\_id, habit\_id, type, unlocked\_at)

### 3.2.2 Migrace databáze

Databáze podporuje migrace pro plynulé aktualizace schématu. Při změně verze databáze se automaticky provedou potřebné změny (přidání sloupců, vytvoření nových tabulek).

## 3.3 NÁVRH UŽIVATELSKÉHO ROZHRANÍ

UI bylo navrženo s důrazem na:

- Minimalistický a moderní design.
- Konzistentní barevné schéma (gradient Pink → Orange).
- Animace pro lepší uživatelský zážitek.
- Dark mode podpora.
- Responzivní layout pro různé velikosti obrazovek.

## 3.4 IMPLEMENTAČNÍ POSTUPY

### 3.4.1 Inicializace aplikace

Aplikace začíná v souboru `main.dart`, kde se inicializují klíčové služby (databáze, notifikace) a načítají uživatelská nastavení.

```

1 void main() async {
2     WidgetsFlutterBinding.ensureInitialized();
3     await DatabaseHelper.instance.database;
4     await NotificationService.instance.initialize();
5
6     final prefs = await SharedPreferences.getInstance();
7     final savedEmail = prefs.getString('user_email');
8     final isDarkMode = prefs.getBool('dark_mode') ?? false;
9     final themeColor = prefs.getString('theme_color') ?? '#009688';
10
11    runApp(HabitTrackApp(
12        isLoggedIn: savedEmail != null,
13        isDarkMode: isDarkMode,
14        themeColor: themeColor,
15    ));
16}

```

Listing 3.1: Inicializace aplikace v main.dart

### 3.4.2 Databázová vrstva

Databázová vrstva je implementována pomocí Singleton patternu v třídě `DatabaseHelper`. Všechny databázové operace jsou asynchronní a používají SQL dotazy.

```

1 class DatabaseHelper {
2     static final DatabaseHelper instance = DatabaseHelper._init();
3     static Database? _database;
4
5     DatabaseHelper._init();
6
7     Future<Database> get database async {
8         if (_database != null) return _database!;
9         _database = await _initDB('habittrack.db');
10        return _database!;
11    }
12
13    Future<Database> _initDB(String filePath) async {
14        final dbPath = await getDatabasesPath();
15        final path = join(dbPath, filePath);
16        return await openDatabase(path, version: 6,
17                                onCreate: _createDB,

```

```

18             onUpgrade: _onUpgrade);
19         }
20     }

```

Listing 3.2: Implementace Singleton patternu pro DatabaseHelper

### 3.4.3 Správa návyků

Hlavní obrazovka zobrazuje seznam návyků uživatele s možností označit je jako splněné. Při dokončení návyku se zobrazí confetti efekt pro vizuální zpětnou vazbu.

```

1 Future<void> _loadHabits() async {
2     final data = await DatabaseHelper.instance.getHabits(userId);
3     final todayStr =
4         DateTime.now().toIso8601String().split('T').first;
5
6     Map<int, int> todayCount = {};
7     Map<int, int> streaks = {};
8
9     for (var habit in data) {
10         final habitId = habit['id'] as int;
11         final count = await DatabaseHelper.instance
12             .getDailyCompletionCount(habitId, todayStr);
13         todayCount[habitId] = count;
14
15         final streak = await DatabaseHelper.instance
16             .getHabitStreak(habitId);
17         streaks[habitId] = streak;
18     }
19
20     setState(() {
21         habits = data;
22         completedTodayCount = todayCount;
23         habitStreaks = streaks;
24     });

```

Listing 3.3: Načítání návyků a jejich statistik

### 3.4.4 Stylování a design systém

Aplikace používá centralizovaný design systém s oddělenými soubory pro barvy, gradienty, textové styly a dekorace. To umožňuje snadnou údržbu a konzistenci vzhledu.

```
1 class AppColors {  
2     // Prim r n  barvy  
3     static const Color primaryOrange = Color(0xFFFF9800);  
4     static const Color primaryPink = Color(0xFFE91E63);  
5     static const Color primaryPurple = Color(0xFF9C27B0);  
6  
7     // Barvy pro ~n vyky  
8     static const Color habitPink = Color(0xFFE91E63);  
9     static const Color habitRed = Color(0xFFFF44336);  
10    static const Color habitOrange = Color(0xFFFF9800);  
11    // ... dal      barvy  
12 }
```

Listing 3.4: Definice barev v design systému

### 3.4.5 Animace a efekty

Aplikace obsahuje různé animace pro zlepšení uživatelského zážitku:

- AnimatedGradientBackground – animované gradientní pozadí.
- ConfettiEffect – confetti při dokončení návyku.
- BounceAnimation – bounce efekt pro seznamy.
- ScaleOnTap – scale efekt při kliknutí.
- SkeletonLoader – skeleton loading screens.

## 3.5 ZPŮSOBY TESTOVÁNÍ

Aplikace byla testována na několika úrovních:

### 3.5.1 Manuální testování

Aplikace byla manuálně testována na:

- Android emulátoru (různé verze Android).
- Fyzickém Android zařízení.

### **3.5.2 Testované funkcionality**

- Autentizace (registrace, přihlášení).
- Správa návyků (vytváření, úprava, mazání).
- Označení návyku jako splněného.
- Zobrazení statistik a grafů.
- Notifikace a připomenutí.
- Dark mode přepínání.
- Personalizace (barvy, profilová fotka).

## **3.6 ŘEŠENÍ PROBLÉMŮ**

Během vývoje byly řešeny následující problémy:

### **3.6.1 Přetečení layoutu na kartách návyků**

**Problém:** Ikonky a text se překrývaly na kartách návyků při menších obrazovkách.

**Řešení:** Zvýšena výška karet a upraveno rozložení prvků s lepším paddingem.

### **3.6.2 Bílá obrazovka před splash screenem**

**Problém:** Na Android zařízeních se zobrazovala bílá obrazovka před načtením Flutter splash screenu.

**Řešení:** Upraveny native Android launch screen soubory (`launch_background.xml`) pro zobrazení gradientního pozadí.

### **3.6.3 Type mismatch při dark mode**

**Problém:** Některé barvy byly definovány jako funkce vyžadující BuildContext, ale používaly se jako statické hodnoty.

**Řešení:** Rozděleny barvy na statické (pro gradient pozadí) a context-aware (pro karty a text).

## **4 VÝSLEDKY ŘEŠENÍ, VÝSTUPY, UŽIVATELSKÝ MÁNUÁL**

### **4.1 SPLNĚNÉ CÍLE PRÁCE**

Všechny stanovené cíle práce byly úspěšně splněny:

- Navržena a implementována mobilní aplikace pro sledování návyků.
- Vytvořeno intuitivní uživatelské rozhraní s moderním designem.
- Implementován systém pro ukládání dat s offline podporou.
- Zajištěna personalizace aplikace (dark mode, barvy, profil).
- Vytvořen systém statistik a vizualizace pokroku.
- Implementovány notifikace pro připomenutí o návycích.

### **4.2 HLAVNÍ FUNKCIONALITY APLIKACE**

#### **4.2.1 Autentizace**

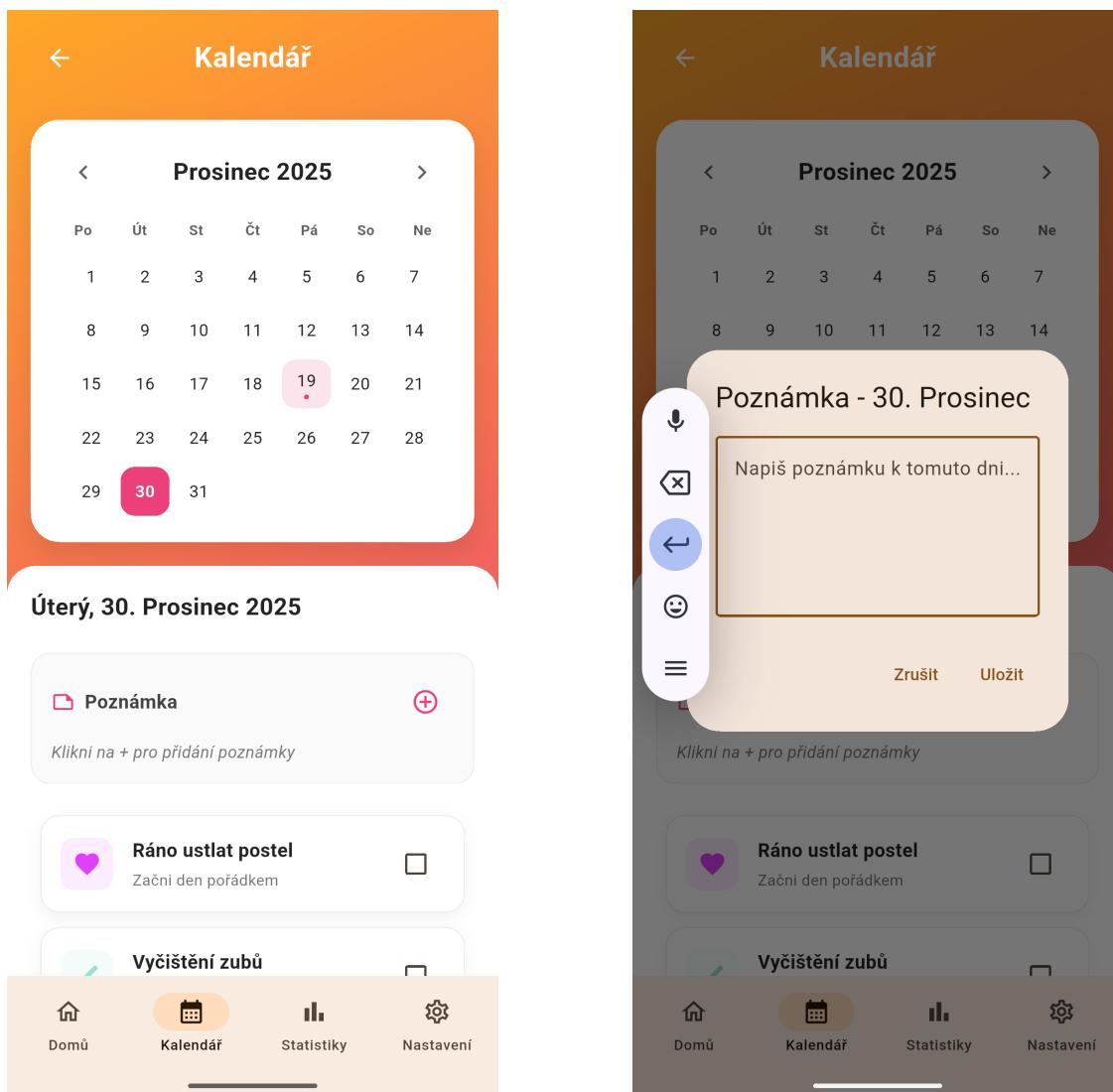
Aplikace obsahuje systém autentizace s registrací, přihlášením a ukládáním přihlašovacích údajů.

#### **4.2.2 Správa návyků**

Uživatel může vytvářet, upravovat a mazat návyky s vlastními parametry (název, popis, barva, ikona, denní cíl), označovat návyky jako splněné pro daný den a zobrazovat progress a streak.

#### 4.2.3 Kalendář

Kalendář zobrazuje historii plnění návyků pro jednotlivé dny, umožňuje přidat poznámky k datům a zobrazuje vizuální indikace splněných/nesplněných návyků.



(a) Kalendář s historií plnění návyků

(b) Přidání poznámky do kalendáře

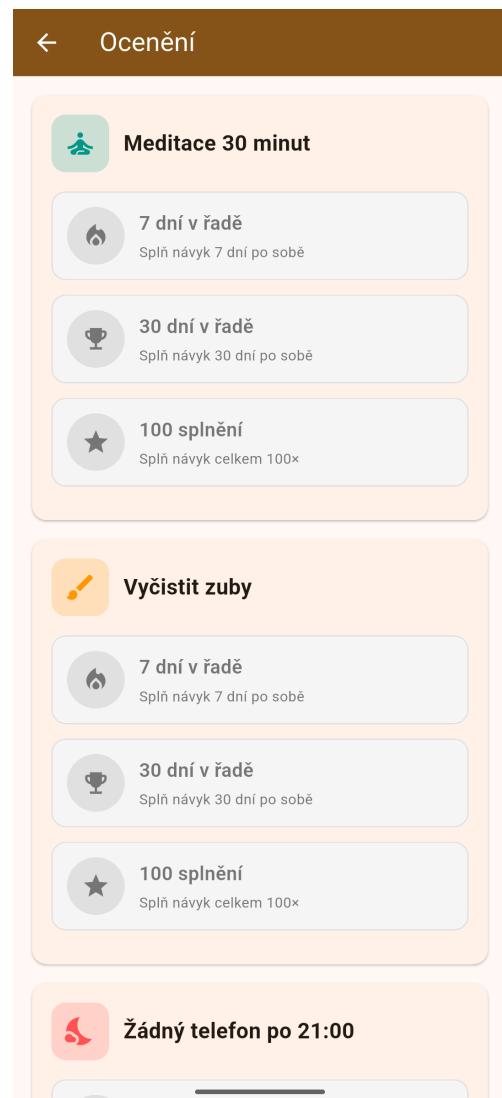
Obrázek 4.1: Kalendář s historií plnění návyků a přidání poznámky

#### 4.2.4 Statistiky a Achievements

Statistiky obsahují týdenní a měsíční přehledy, grafy pokroku a počty splnění návyků. Systém achievementů motivuje uživatele (7, 30, 100 dní v řadě).



(a) Statistiky a grafy pokroku

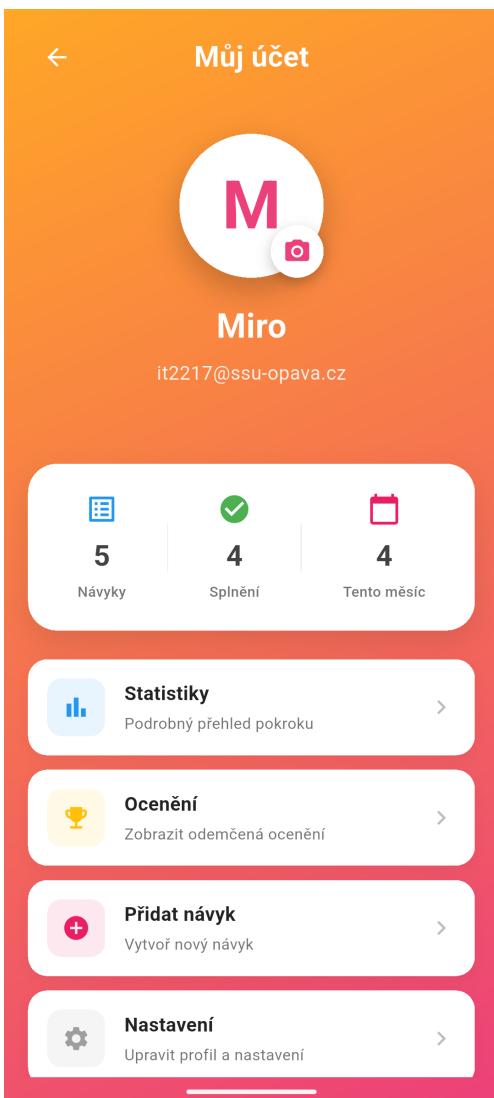


(b) Systém achievementů

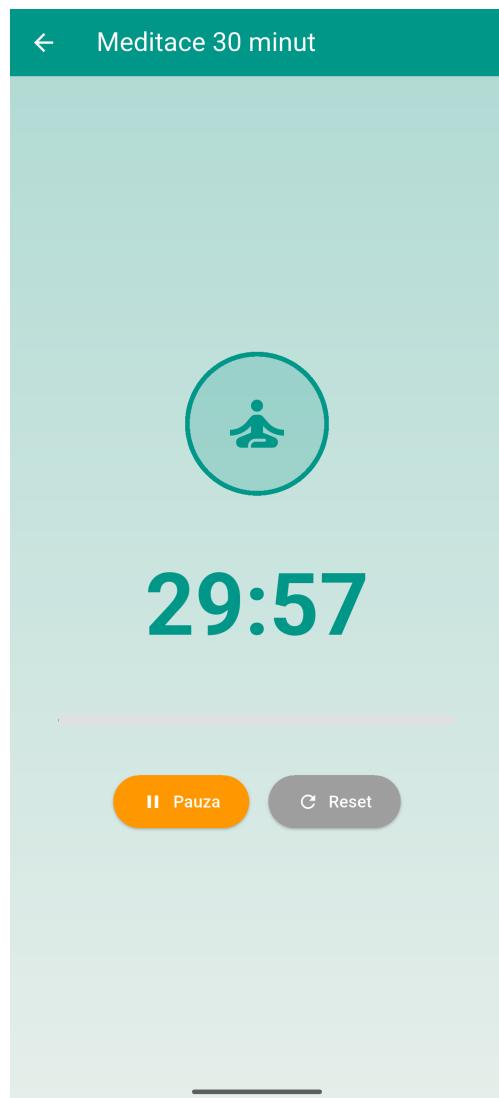
Obrázek 4.2: Statistiky a systém achievementů

#### 4.2.5 Personalizace a Notifikace

Uživatel může přepínat mezi světlým a tmavým režimem, měnit barvu motivu a nastavit profil. Systém notifikací umožňuje naplánovat připomenu pro návyky.



(a) Profil uživatele a nastavení



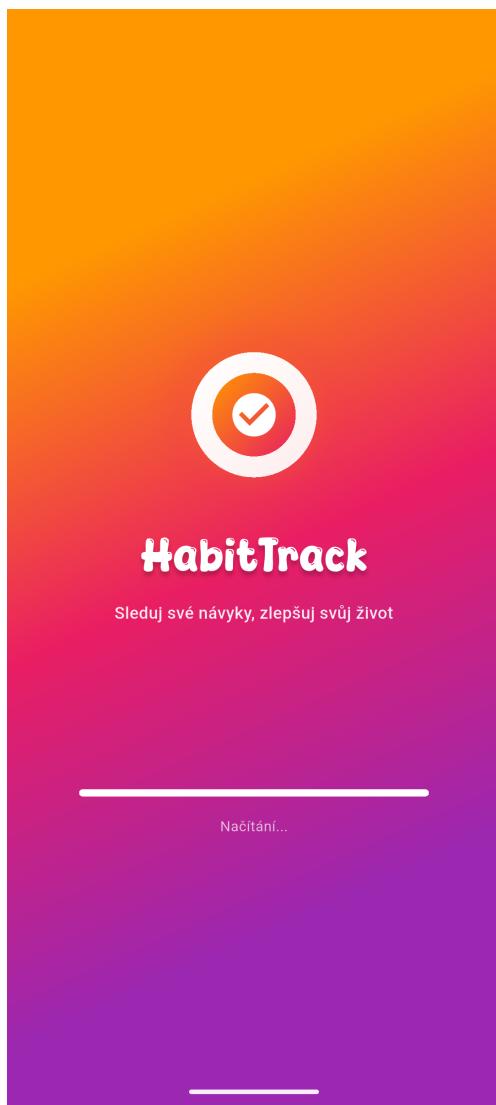
(b) Časovač pro návyky s časovým limitem

Obrázek 4.3: Profil uživatele a časovač

## 4.3 UŽIVATELSKÝ MANUÁL

### 4.3.1 První spuštění a přihlášení

Po spuštění aplikace se zobrazí splash screen s logem aplikace. Následně se uživatel dostane na obrazovku přihlášení nebo registrace. Po úspěšném přihlášení se dostane na hlavní obrazovku.



(a) Loading screen s logem aplikace



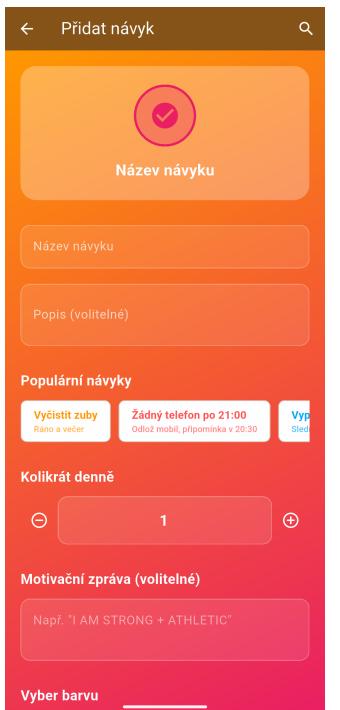
(b) Hlavní obrazovka s návyky

Obrázek 4.4: Loading screen a hlavní obrazovka

### 4.3.2 Vytvoření návyku

Proces přidání nového návyku probíhá ve třech krocích:

1. Na hlavní obrazovce klikněte na tlačítko "+" (vpravo nahore)
2. Vyplňte formulář (název, popis, barva, ikona, denní cíl)
3. Volitelně nastavte připomenutí
4. Klikněte na "Vytvořit návyk"



(a) Krok 1: základní informace



(b) Krok 2: výběr ikony



(c) Krok 3: dokončení

Obrázek 4.5: Proces vytvoření návyku ve třech krocích

### 4.3.3 Základní funkce

Na hlavní obrazovce lze kliknutím na kartu návyku označit návyk jako splněný. V dolní navigaci lze přepínat mezi obrazovkami pro zobrazení statistik a nastavení profilu.

## 4.4 TECHNICKÉ SPECIFIKACE

Aplikace vyžaduje Android 5.0 (API level 21) nebo vyšší a minimálně 50 MB volného místa. Podporuje offline fungování, dark mode, notifikace, profilové fotky a statistiky s grafy.

# ZÁVĚR

Hlavním cílem této závěrečné práce bylo vytvořit plně funkční mobilní aplikaci pro sledování návyků a zdraví pomocí Flutter frameworku. Tento cíl byl úspěšně splněn. Aplikace HabitTrack poskytuje uživatelům moderní a intuitivní nástroj pro správu jejich denních návyků s důrazem na vizualizaci pokroku a motivaci k pravidelnému plnění.

V rámci práce byla implementována kompletní aplikace obsahující systém autentizace, správu návyků, sledování denního plnění, kalendář s historií, statistiky, systém achievementů, personalizaci, notifikace a offline fungování s lokálním ukládáním dat. Aplikace byla vyvinuta s použitím čisté architektury a centralizovaného designu systému, což usnadňuje údržbu a budoucí rozvoj. Díky použití Flutter frameworku je aplikace cross-platform a může být nasazena na obě hlavní mobilní platformy.

Pro budoucí rozvoj aplikace bylo možné implementovat cloud synchronizaci dat mezi zařízeními, sociální funkce pro sdílení pokroku, export dat do různých formátů, podporu více jazyků, widgety pro rychlý přístup, integraci s health aplikacemi pro automatické sledování aktivit a AI doporučení pro optimalizaci návyků.

Zdrojový kód aplikace je dostupný na GitHubu: [github.com/lubenikmiroslav/zaverecnaPrace](https://github.com/lubenikmiroslav/zaverecnaPrace)



## **SEZNAM POUŽITÝCH INFORMAČNÍCH ZDROJŮ**



## LITERATURA

- [1] GOOGLE. *Flutter - Build apps for any screen* [online]. 2024 [cit. 2024]. Dostupné z: <https://flutter.dev>
- [2] GOOGLE. *Dart programming language* [online]. 2024 [cit. 2024]. Dostupné z: <https://dart.dev>
- [3] GOOGLE. *Flutter Documentation* [online]. 2024 [cit. 2024]. Dostupné z: <https://docs.flutter.dev>
- [4] TEKARTIK. *sqflite - SQLite plugin for Flutter* [online]. 2024 [cit. 2024]. Dostupné z: <https://pub.dev/packages/sqflite>
- [5] FLUTTER TEAM. *shared\_preferences - Flutter plugin for storing key-value data* [online]. 2024 [cit. 2024]. Dostupné z: [https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences)
- [6] STACK EXCHANGE INC. *Stack Overflow - Where Developers Learn, Share, & Build Careers* [online]. 2024 [cit. 2024]. Dostupné z: <https://stackoverflow.com>
- [7] GITHUB INC. *GitHub - Where the world builds software* [online]. 2024 [cit. 2024]. Dostupné z: <https://github.com>
- [8] W3SCHOOLS. *W3Schools Online Web Tutorials* [online]. 2024 [cit. 2024]. Dostupné z: <https://www.w3schools.com>
- [9] MOZILLA. *MDN Web Docs* [online]. 2024 [cit. 2024]. Dostupné z: <https://developer.mozilla.org>
- [10] GOOGLE. *Flutter Cookbook* [online]. 2024 [cit. 2024]. Dostupné z: <https://docs.flutter.dev/cookbook>

# Seznam obrázků

4.1	Kalendář s historií plnění návyků a přidání poznámky . . . . .	20
4.2	Statistiky a systém achievementů . . . . .	21
4.3	Profil uživatele a časovač . . . . .	22
4.4	Loading screen a hlavní obrazovka . . . . .	23
4.5	Proces vytvoření návyku ve třech krocích . . . . .	24