

Vývoj aplikace pro sledování zdravotních informací

Lukáš Ondřej

Bakalářská práce
2024



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Lukáš Ondřej
Osobní číslo: A21346
Studijní program: B0613A140020 Softwarové inženýrství
Forma studia: Prezenční
Téma práce: Vývoj aplikace pro sledování zdravotních informací
Téma práce anglicky: Development of an Application for Monitoring Health Information

Zásady pro vypracování

1. Popište současný stav dané problematiky.
2. Popište technologie, které budou v práci použité.
3. Provedte analýzu požadavků pro sledování zdravotnických informací.
4. Navrhněte a realizujte databázi a aplikaci
5. Provedte otestování aplikace.
6. Zhodnoťte provedené řešení a navrhněte případné vylepšení.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. FREEMAN, Adam, 2022. Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages. 9th ed. Edition. Apress. ISBN 978-1484279564.
2. DRESHER, Tamir, Amir ZUKER a Shay FRIEDMAN, 2018. Hands-On Full-Stack Web Development with ASP.NET Core: Learn end-to-end web development with leading frontend frameworks, such as Angular, React, and Vue. Packt Publishing. ISBN 978-1788622882.
3. MICROSOFT CORPORATION, INC., b. r. C# docs – get started, tutorials, reference. | Microsoft Learn [online]. [cit. 2023-11-06]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/csharp/>
4. MICROSOFT CORPORATION, INC., b. r. ASP.NET documentation | Microsoft Learn [online]. [cit. 2023-11-06]. Dostupné z: <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0>
5. J. PRICE, Mark, 2022. C# 11 and .NET 7 – Modern Cross-Platform Development Fundamentals: Start building websites and services with ASP.NET Core 7, Blazor, and EF Core 7, 7th Edition. 7th ed. Edition. Packt Publishing. ISBN 978-1803237800.
6. BUCKEY, A. W., 2019. Personal Data Collection (Privacy in the Digital Age). Core Library. ISBN 978-1532118937.

Vedoucí bakalářské práce: **Ing. Dušan Fojtů, Ph.D.**
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **5. listopadu 2023**

Termín odevzdání bakalářské práce: **13. května 2024**



doc. Ing. Jiří Vojtěšek, Ph.D.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA
ředitel ústavu

Ve Zlíně dne 5. ledna 2024

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 13. 5. 2024

Lukáš Ondřej
v.r.

ABSTRAKT

Byla navržena a vyvinuta webová aplikace FitTrackr, která je určena ke sledování zdravotních informací. Aplikace je určena pro sledování těchto informací, za účelem sledování a podpory zdravého fyzického a mentálního zdraví člověka. Aplikace umožňuje importovat data z MiBand hodinek za pomoci exportu z MiFitness a plánovat aktivity, tréninky a cíle. Mezi poskytované funkce také patří evidence kalorického příjmu a pitného režimu. Taktéž je zde funkcionalita pro sledování spánku a srdečního tepu, na jejichž základě dojde k vyhodnocení a zobrazení ve formě jednoduchého přehledu. Případně je součástí přehledu i doporučení, za účelem, například zlepšení kvality spánku. Pro každou skupinu záznamů poskytuje přehled o sledovaných údajích formou grafů a doporučení pro uživatele. Všechny tyto služby jsou poskytovány za účelem zlepšení osobního zdraví uživatele.

Klíčová slova: ASP.NET, Entity Framework, C#, MS Visual Studio 2022, JetBrains Rider, zdravotní informace, BMI, srdeční tep, kalorie

ABSTRACT

The FitTrackr web application was designed and developed, which is intended for tracking health information. The application is intended for monitoring this information, in order to monitor and support a person's healthy physical and mental health. The application allows you to import data from the MiBand watch using export from MiFitness and plan activities, training sessions and goals. Among the functions provided are also records of caloric intake and drinking regime. There is also functionality for monitoring sleep and heart rate, based on which an evaluation and display in the form of a simple overview will take place. Optionally, the overview also includes recommendations, for the purpose of, for example, improving the quality of sleep. For each group of records, it provides an overview of the monitored data in the form of graphs and recommendations for users. All these services are provided for the purpose of improving the user's personal health.

Keywords: ASP.NET, Entity Framework, C#, MS Visual Studio 2022, JetBrains Rider, health information, BMI, heart rate, calories

PODĚKOVÁNÍ

Děkuji své rodině za podporu během studia a mému vedoucímu Ing. Dušanu Fojtů, Ph. D. za jeho cenné rady a ochotu v průběhu vypracovávání bakalářské práce.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 SOUČASNÝ STAV PROBLEMATIKY	11
1.1 APLIKACE ZAMĚŘENÉ NA FYZICKÉ ZDRAVÍ	11
1.1.1 Apple Health	12
1.1.2 Mi Fitness.....	12
1.1.3 YAZIO	12
1.2 APLIKACE ZAMĚŘENÉ NA DUŠEVNÍ ZDRAVÍ.....	13
1.2.1 VOS.....	13
2 VYUŽITÉ TECHNOLOGIE PRO VÝVOJ	14
2.1 .NET	14
2.1.1 C# (C Sharp)	14
2.1.2 .NET Framework.....	14
2.1.3 .NET Core	15
2.1.4 .NET 8	15
2.2 ASP.NET.....	16
2.2.1 Model-View-Controller (MVC).....	16
2.2.2 Blazor	17
2.3 ASP.NET IDENTITY	18
2.3.1 Identity Autentizace	18
2.3.2 Identity Autorizace.....	18
2.4 DATABÁZOVÝ SYSTÉM	18
2.4.1 Databáze.....	19
2.4.2 Systém báze řízení dat (DBMS).....	19
2.4.3 Způsoby přístupu k databázi při vývoji aplikace	20
2.4.3.1 Code First.....	20
2.4.3.2 Database First	20
2.4.3.3 Model First.....	21
2.5 MYSQL	21
2.5.1 Architektura.....	21
2.5.1.1 Klientská vrstva	21
2.5.1.2 Serverová vrstva	22
2.5.1.3 Úložná vrstva	22
2.5.1.4 Princip komunikace	23
2.6 ENTITY FRAMEWORK CORE	24
2.6.1.1 Popis použití Entity Framework Core	24
2.7 VÝVOJOVÉ PROSTŘEDÍ	25
2.7.1 Visual Studio 2022	25
2.7.2 JetBrains Rider	26
2.8 TESTOVÁNÍ.....	27
2.8.1 xUnit.....	27
2.8.2 Moq	27

2.9	KNIHOVNY UŽIVATELSKÝCH ROZHRANÍ.....	28
2.9.1	AntBlazor	28
2.9.2	ApexCharts.....	29
2.9.2.1	Blazor ApexCharts.....	29
2.10	SOFTWAREVÉ LICENCE.....	29
2.10.1	MIT Licence.....	30
3	HOSTING	31
3.1	TYP HOSTOVANÉ APLIKACE.....	31
3.1.1	Statická webová aplikace	31
3.1.2	Dynamická webová aplikace.....	31
3.2	MOŽNOSTI HOSTINGU	32
3.2.1	Hosting statických aplikací	32
3.2.2	Hosting dynamických aplikací	32
4	ENERGETICKÝ VÝDEJ.....	33
4.1	METABOLICKÝ EKVIVALENT (MET)	33
4.1.1	Výpočet metabolického ekvivalentu	33
4.1.2	BMI	34
II	PRAKTICKÁ ČÁST	35
5	ANALÝZA POŽADAVKŮ APLIKACE	36
5.1	FUNKČNÍ POŽADAVKY	36
5.1.1	Účet	37
5.1.2	Data	37
5.1.2.1	Zdroj dat.....	38
5.1.2.2	Ukládání a načítání dat	38
5.1.2.3	Práce s daty	38
5.1.3	Sociální aspekt	39
5.2	NEFUNKČNÍ POŽADAVKY	39
6	STRUKTURA PROJEKTU	40
6.1	WEBOVÁ VRSTVA.....	40
6.2	APLIKAČNÍ VRSTVA.....	41
6.3	INFRASTRUKTURNÍ VRSTVA.....	41
6.4	DOMÉNOVÁ VRSTVA	42
6.5	TESTOVACÍ VRSTVA	43
7	NÁVRH DATABÁZE	44
7.1	DATABÁZOVÁ STRUKTURA	44
7.1.1	Tabulky údajů v aplikaci	44
7.1.1.1	Aktivity, Trénink a Cíle	44
7.1.1.2	Strava	45
7.1.1.3	Spánek a Srdeční tep.....	46
7.1.1.4	Duševní zdraví	47
7.1.2	Gamifikace, sociální aspekt	48
7.2	ASP.NET IDENTITY	48
8	UŽIVATELSKÉ ROZHRANÍ.....	50

8.1	ÚVODNÍ STRÁNKA	50
8.1.1	Registrace	51
8.1.2	Přihlášení	51
8.2	DOMOVSKÁ STRÁNKA APLIKACE	52
8.3	ÚČET	53
8.3.1	Osobní profil	53
8.3.2	Sociální stránka	54
8.3.3	Cizí profil	55
8.3.4	Import a export dat	55
8.4	AKTIVITY, TRÉNINK A CÍLE	56
8.4.1	Aktivity	57
8.4.2	Trénink	58
8.4.3	Cíle	59
8.5	STRAVA	60
8.6	ZDRAVÍ	61
8.7	SPÁNEK	62
8.8	DUŠEVNÍ ZDRAVÍ	63
8.8.1	Deník	64
8.8.2	Nálada	65
8.9	HISTORIE	66
9	TESTOVÁNÍ APLIKACE	68
10	HOSTOVÁNÍ APLIKACE	69
11	ZHODNOCENÍ A VYLEPŠENÍ	70
11.1	FUNKCE	70
11.2	ADAPTACE VŮČI MODERNÍM TECHNOLOGIÍM	70
11.3	FRONTEND	70
	ZÁVĚR	72
	SEZNAM POUŽITÉ LITERATURY	73
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	75
	SEZNAM OBRÁZKŮ	76
	SEZNAM PŘÍLOH	77

ÚVOD

Duševní a fyzické zdraví každého člověka hraje významnou roli v rámci celé společnosti, neboť tyto aspekty ovlivňují nejenom samotného jednotlivce, ale také jeho okolí. Například v pracovním prostředí, kde důsledkem špatného zdraví může být přímo ovlivněn pracovní výkon a s tím související dopady na ostatní zaměstnance. Stejně tak tomu je v domácím prostředí, kdy duševní zdraví se přímo může projevit ve formě napětí mezi rodinnými příslušníky a působí velmi negativně na všechny členy domácnosti, ať už psychicky, či v horších případech fyzicky.

Je důležité těmto problémům předcházet a jedním z efektivních způsobů je jednoduché pravidelné sledování a péče o celkovou osobní pohodu. Ta zahrnuje jak fyzické, tak mentální aspekty jedince.

V minulosti byli lidé odkázáni pouze na zdravotnická zařízení, kde jim byly tyto informace poskytnuty a doporučena případná medikace, či jiné metodiky pro zlepšení stavu. Nevýhodou bylo pouze občasné zjišťování těchto informací, nutnost cestování a reakce na případné změny, či problémy by ve výsledku mohly přijít opožděně.

Dnes již lze sledovat celou řadu údajů o lidském těle (srdeční tep, index tělesné hmotnosti, saturace kyslíku v krvi atd.) online z pohodlí domova a bez nutnosti navštívit lékařské zařízení. V nejlépším případě za využití technologií, jako jsou chytré hodinky, lze tyto informace získávat v kombinaci s mobilním zařízením takřka okamžitě.

V dnešní době stále se rozšiřujících možností sledovat zdravotní informace pomocí nejrůznějších technologií v oblasti vývoje tzv. chytrých hodinek je na místě zabývat se zpracováním těchto údajů a jejich ochranou.

Tato práce je příspěvkem k dané a výše zmíněné problematice, kdy při využití nejmodernějších nástrojů vývojových prostředí a nejmodernějších programovacích technik byla navržena a vyvinuta webová aplikace, která umožňuje sledovat, zpracovávat a analyzovat zdravotní informace, a na základě vyhodnocení poskytovat přehledy, ze kterých si uživatel může vyvodit vlastní závěr, či přímo obdržet jednoduchá doporučení na základě osobních záznamů.

I. TEORETICKÁ ČÁST

1 SOUČASNÝ STAV PROBLEMATIKY

Aktuální trh, co se týče aplikací se zaměřením na zaznamenávání zdravotních informací, nabízí mnoho možností, ze kterých si mohou uživatelé vybírat. Výsledná volba však záleží na individuálních preferencích a potřebách zákazníka.

Každou z dostupných aplikací lze rozdělit do jedné ze dvou kategorií dle jejich zaměření:

- Fyzické zdraví
- Duševní zdraví

Přesto, že některé aplikace obsahují funkce z obou těchto zaměření, vždy je jedno z nich výrazně dominantní.

Jakmile má uživatel vybráno požadované zaměření, následuje volba dle vlastních požadavků. Z tohoto důvodu neexistuje jedno nejlepší řešení, jelikož má každý uživatel své vlastní potřeby.

Mezi základní rozhodující faktory můžeme zařadit:

- Přesnost – Jak probíhá měření? Jak jsou měření zaznamenávána?
- Platformu – Multiplatformní či výhradně desktopové/mobilní?
- Zpracování dat – zabezpečení, sběr dat.
- Zpoplatnění funkcí – zdarma, předplatné, placená aplikace.

1.1 Aplikace zaměřené na fyzické zdraví

Tyto aplikace jsou soustředěny především na záznamy týkající se fyzických výkonů či údajů uživatele.

Jako některé z typických sledovaných hodnot lze zmínit:

- Aktivity – spálené kalorie, doba trvání aktivity.
- Kalorie – zaznamenávání přijatých kalorií, přehled energických hodnot.
- Pitný režim – dodržování pitného režimu, nastavení připomínek během dne.
- Návrh tréninkového plánu.
- Přehled cviků.

Vzhledem k tomu, že trh pro toto zaměření je rozsáhlý, jsou zde zmíněny pouze některé aplikace spolu s jejich zaměřením a účelem.

1.1.1 Apple Health

Aplikace pro sledování zdraví přímo od společnosti Apple Inc. Jedná se o součást celého ekosystému iOS a mimo sledování aktivit či kroků, je zde celá řada dalších funkcí, které také umožňují průběžné měření tepu, sledování spánku a jeho kvality, pohybové údaje za využití gyroskopu v příslušném zařízení a mnoho dalších.

Přesto, že aplikaci lze zařadit do kategorie, která se soustředí na fyzické zdraví uživatele v rámci celého katalogu funkcionalit, se jedná jen o jednu z dostupných využití. A nelze jej označit za vlajkovou loď tohoto zástupce.

1.1.2 Mi Fitness

Aplikace od společnosti Xiaomi se soustředí na sledování základních fyzických údajů uživatele za využití fitness náramku Mi Band.

Obdobně jako u Apple Health i zde je využití obdobné. Lze sledovat a zaznamenávat kvalitu a délku spánku, průběžně měřit tep, aktivity spolu se spálenými kaloriemi a mnoho dalšího.

Oproti dříve zmíněnému konkurentovi se však jedná o aplikaci, která je zaměřena výhradně na fitness účely.

1.1.3 YAZIO

Jako třetí příklad zástupce aplikací pro sledování fyzických údajů uživatele byla vybrána aplikace YAZIO. Zatím co předešlé dvě aplikace (Apple Health a Mi Fitness), byly vytvořeny a přímo zabudovány do příslušných telefonů výrobců, aplikace YAZIO je jednou z aplikací, které jsou volně dostupné ke stažení na Google Play či App Store. Z tohoto důvodu je v aplikaci zakomponováno předplatné a část funkcionalit je v případě verze zdarma limitována.

Vzhledem k tomu, že se nejedná o nativní aplikaci, není zde příslušenství, které by zasílalo údaje do aplikace automaticky. Veškeré zaznamenávání je prováděno na základě uživatelského vstupu.

I přesto obsahuje YAZIO širokou nabídku funkcí, ať se jedná o zaznamenávání příjmu kalorií, dodržování pitného režimu, návrh tréninkového plánu a dalších obdobných funkcí, je zde i funkce pro sledování přerušovaného půstu.

Tato funkcionalita je založena na principu časovače, kdy si uživatel zvolí příslušný interval, nejčastěji 16 hodin půst, 6 hodin příjem kalorií, a je mu skrze den připomínáno, ve které z fází postění se právě nachází.

1.2 Aplikace zaměřené na duševní zdraví

V porovnání s aplikacemi pro fyzické zdraví je výběr aplikací pro duševní zdraví značně omezen. Většina se skládá z jednoduchých funkcí, jako je zaznamenávání nálady, možnost zápisu do deníku, či přístup k nahraným relaxačním zvukům a meditacím.

Zde je vhodné zmínit českou aplikaci VOS.

1.2.1 VOS

Aplikace VOS poskytuje inovativní přístup k zaznamenávání a sledování duševního zdraví uživatele. Místo jednoduchých, zastaralých způsobů zaznamenávání je zde dostupná široká paleta nástrojů, které míří na duševní pohodu člověka.

Paleta se skládá například z:

- Testů pro lepší sebepoznání – typ osobnosti, zhodnocení kvality života.
- Záznamy do deníku – oproti typickému zápisu je zde implementována AI, analyzuje obsah textu a doporučuje dodatečné otázky, pokud má uživatel zájem je využít.
- Chatbot zaměřený na duševní pohodu.
- Meditace a relaxační zvuky.
- Zaznamenávání nálady a její průběžná analýza.
- Tipy, doporučení a povzbuzující citáty.
- Spojení se s terapeutem.
- Odborné články.

Aplikace nemá v momentální době kvalitního konkurenta, jelikož se jedná o největší aplikaci s tímto zaměřením na trhu.

Vzhledem k tomu, že se nejedná o nativní aplikaci, je zde také část funkcionalit skryta za předplatným.

Veškerá doporučení, návrhy a testy jsou podloženy vědeckými výzkumy, které byly zpracovány experty na mentální zdraví člověka.

2 VYUŽITÉ TECHNOLOGIE PRO VÝVOJ

Pro vývoj samotné aplikace bylo využito mnoho typických technologií, od samotného ekosystému .NET a programovací jazyk, přes framework pro webovou aplikaci až po knihovny komponent, které se starají o vizuální stránku aplikace.

2.1 .NET

.NET je bezplatná vývojářská platforma vyvinutá společností Microsoft. Jedná se o otevřený software, tedy je přístupný všem vývojářům, bez nutnosti vlastnění jakékoliv licence. Platforma podporuje několik programovacích jazyků, jako jsou například Python, či C++, avšak většinou podporu mají jazyky od společnosti Microsoft.

Mezi ty se řadí Visual Basic pro jednoduchý vývoj aplikací s grafickým rozhraním (GUI), funkčně orientovaný jazyk F# a v neposlední řadě objektově orientovaný jazyk C#.

Celkový ekosystém platformy .NET se skládá z několika částí, mezi které řadíme například .NET Framework společně s .NET (Core). [1]

2.1.1 C# (C Sharp)

C# se v dnešní době řadí mezi nejpobulárnější jazyky pro vývoj aplikací. Jedná se o objektově orientovaný způsob programování, kde hlavní princip spočívá ve zpracování logiky kódu prostřednictvím abstrakce objektů.

S těmito objekty se pracuje jako s datovými strukturami, které spojují účely a díky tomu je lze seskupovat. Na základě těchto principů je umožněna efektivní organizace a manipulace s daty. Mimo výše zmíněné abstrakce obsahuje objektově orientované programování i jiné výhody, mezi které se řadí například dědičnost tříd a objektů nebo také polymorfismus či zapouzdření.

Taktéž se jedná o jazyk navržený a vyvinutý společností Microsoft, díky čemuž je pevně integrován s .NET platformou a umožňuje plně využívat výhody tohoto ekosystému.[2]

2.1.2 .NET Framework

Jednou z hlavních částí ekosystému je prostředí .NET Framework, který je soustředěn na vývoj především desktopových aplikací pro platformu Windows, avšak umožňuje i vývoj webových aplikací. Je taktéž součástí operačního systému Windows a není potřeba jej manuálně aktualizovat.

Je složen jak z modulu CLR (Common Language Runtime) pro správu paměti společně se systémovými službami, tak z knihoven, které usnadňují vývoj spolehlivých a rozsáhlých aplikací.

2.1.3 .NET Core

.NET Core je považován za pomyslnou evoluci předchozí platformy .NET Framework. I přesto, že se jedná o vylepšení, byl .NET Core kompletně přepracován, s důrazem na lepší výkon, jednodušší a efektivnější přenositelnost, a především za účelem multiplatformní podpory.

Oproti .NET Framework je možno nově aplikace vyvíjet a spouštět, mimo operačního systému Windows, také na operačních systémech Linux a MacOS. Tato nově integrovaná flexibilita zahrnuje taktéž podporu pro mobilní zařízení s operačními systémy Android a iOS.

Významnou změnou je také rozšíření podpory pro vývoj webových aplikací pomocí platformy ASP.NET, vedle již existující podpory pro aplikace, mířené na desktop.

2.1.4 .NET 8

Aktuální podoba .NET reprezentuje sloučení předešlých platform, tj. .NET Framework a .NET Core do jednotného celku. [1] Tento přístup ke kompletnímu rozhraní umožňuje vývojářům snadnější vývoj aplikací.

Společnost Microsoft neustále investuje do tohoto celku a postupně vydává a upravuje infrastrukturu celé platformy. Neustálá podpora se soustředí na vylepšení výkonu či snadnější přenositelnosti. K aktuálním novinkám lze zařadit kupříkladu integraci nových technologií, jako je například umělá inteligence.

Důležitým aspektem platformy je taktéž komfortnější vývoj za pomoci již vytvořených balíčků, samotnou společností nebo od .NET komunity. Tyto balíčky jsou publikovány prostřednictvím platformy NuGet, která je integrovaná do většiny vývojových prostředí za účelem co nejjednodušší interakce.

V době psaní bakalářské práce je nejnovější verzí .NET verze s číselným označením 8. Tato verze výrazně obohacuje možnosti vývoje webových aplikací za využití frameworku Blazor.

2.2 ASP.NET

.NET platforma se nadále dělí na několik podkategorií, kde každá z nich má vlastní specializaci. Mezi jednu z těchto tříd se řadí také framework ASP.NET. Jedná se o platformu pro vývoj dynamických webových stránek a aplikací.

Vzhledem k tomu, že ASP.NET vychází z platformy .NET, má rozsáhlou integraci s jinými kategoriemi a je zde umožněno využívat jak rozsáhlých, již existujících knihoven, tak jazyků a nástrojů, které jsou zde poskytovány. [1] To umožňuje plně využívat sílu celého ekosystému.

Jedna z výhod využití této platformy společně s jazykem C# jsou příkazy, označované jako LINQ (Language Integrated Query), kterými jsou prováděny dotazy na databázi. [2] Lze tuto funkcionalitu chápat jako SQL příkazy, které jsou prováděny .NET platformou.

Mezi některé z funkcí, které ASP.NET poskytuje, můžeme zmínit například konfiguraci pro zabezpečení dat, funkce pro prevenci útoků, například za pomoci SQL injection, kdy je využit SQL kód pro manipulaci s databází skrze jednoduchý, zdánlivě neškodný uživatelský vstup na stránce.

Taktéž je vhodné zmínit funkce pro testování, ať už testů jednotkových, kdy jde o jednotlivé části aplikace, tak integračních, které kombinují a ověřují funkcionalitu komunikace mezi částmi stránky.

Framework má v dnešní době podporu pro několik frameworků, avšak mezi nejvýznamnější a nejvíce podporované řadíme Model-View-Controller a Blazor. [3]

2.2.1 Model-View-Controller (MVC)

Tato architektura pracuje na principu rozdělení logiky aplikace do jednotlivých částí, kde model reprezentuje data, view reprezentuje uživatelské rozhraní a controller řídí komunikaci mezi těmito dvěma vrstvami a celkově se stará o řízení toku aplikace. [1]

Zdánlivě složité rozdělení však umožňuje přehlednou organizaci a snadnou údržbu celkové architektury.

Přesto, že jsou aplikace napsány za pomoci programovacího jazyka C#, je zde podpora pro využití JavaScriptu za účelem dynamických akcí na stránce bez nutnosti obnovení uživatelského rozhraní. Tím lze dosáhnout interaktivní a plynulou interakci pro uživatele.

2.2.2 Blazor

Blazor je novější a mnohem mladší framework uživatelského rozhraní v porovnání s architekturou MVC. Umožňuje vývoj dynamických webových aplikací a stránek kompletně za použití jazyka C#.

Důležité je zde zmínit, že Blazor podporuje aplikace, které jsou mohou běžet jak na serveru, tak v prohlížeči uživatele, případně lze oba tyto principy zkombinovat. Tímto dosahuje framework mnohonásobně většího výkonu, než tomu je u MVC.

Mezi hlavní rozdíly vůči architektuře MVC můžeme zařadit například mnohonásobně zvýšené využití takzvaných událostí, které zajišťují a zpracovávají komunikaci mezi jednotlivými částmi aplikace a obohacují tak interaktivní zážitek bez nutnosti obnovení stránek. [2]

Dalším významným rozdílem je zdánlivé zrušení části pro řízení toku aplikace (controlleru). Controllery jsou stále podporovány, avšak nejsou povinné pro správnou funkcionalitu samotné aplikace. Komunikace mezi klientem a serverem probíhá za pomoci knihovny SignalR, která umožňuje načítání a interaktivitu v reálném čase. Ta funguje na principu jednoduchého volání API rozhraní. [3]

Místo toho je zde využito komponentového modelu. Ten funguje na principu vytvoření jednotlivých částí stránky jako samostatné Razor soubory, které mají na první pohled stejnou strukturu jako stránky vytvořené za pomoci HTML.

Hlavním rozdílem však je, že při volání těchto komponent a jejich pomyslném skládání na cílové stránce, dochází k předávání parametrů, které se mohou týkat například zobrazení dat, zpracování událostí na komponentách apod. Tento princip lze přirovnat k populárnímu JavaScriptovému frameworku React, případně jej považovat za jakousi odpověď od Microsoftu na popularitu konkurenčního frameworku.

Obsluha služeb a logiky aplikace zpracovávána skrze kód přímo pod HTML částí stránky, případně lze oddělit logiku do samostatné třídy, která je však stále součástí dané stránky.

Blazor taktéž obsahuje podporu pro využití JavaScriptu za pomoci interoperability, čímž lze využít knihoven a rámců, avšak není zde zaručena stejná funkcionalita. Oficiálně je upřednostňováno zpracování jakékoliv interaktivity čistě za použití jazyku C#. [4]

2.3 ASP.NET Identity

Mezi jeden z nejrozsáhlejších ASP.NET balíčků můžeme zařadit balíček Identity pro zpracování funkcionality individuálních účtů ve webové aplikaci. Jedná se o sbírku metod, které efektivně řeší autentizaci a autorizaci uživatelů.

Princip spočívá v ověřování práv klienta společně s dotazováním se na databázi. V té době při migraci k automatickému vytvoření tabulek s předem definovanými sloupci, čímž je implementace individuálních účtů mnohonásobně ulehčena.

Knihovna také nabízí sledování událostí souvisejících s přihlašováním a registrací. Zde se může jednat o zaznamenávání přihlášení, registrace, nebo také neúspěšných pokusů o přihlášení. Tím lze přispět k celkovému zabezpečení aplikace. [4]

2.3.1 Identity Autentizace

V rámci autentizace lze spravovat obsah, který se uživateli zobrazí na základě toho, zda se jedná o přihlášeného uživatele, či nikoliv. Knihovna zajišťuje automatické přesměrování na základě kontrolování prohlížeče uživatele.

Taktéž knihovna umožňuje externí přihlášení skrze existující účty na sociálních sítích, či jiných aplikacích, čímž přispívá k uživatelské přívětivosti.[4]

2.3.2 Identity Autorizace

Autorizací lze ověřit práva účtu a zaručit bezpečnost stránky, kdy nepovolené osobě nejsou zpřístupněny části, které by mohly ovlivnit strukturu či obsah stránek.

Stejně tak lze zpřístupnit oprávněným osobám správu databáze, včetně správy vytvořených uživatelských účtů či umožnit uživatelům přístup k danému obsahu, na základě jejich rolí.

2.4 Databázový systém

Při práci s databází, je potřeba mít k dispozici nástroje, kterými je možno efektivně pracovat s uloženými daty. Program, který tyto operace zprostředkovává, označujeme za databázový systém, který představuje celou strukturu.

Databázový systém můžeme rozdělit na dvě hlavní části:

- Databáze,
- Systém báze řízení dat (DBMS).

Kombinací těchto částí dostaneme systém, který lze pochopit jako prostředníka, který se stará o celkovou správu při práci s databází. [6]

Je důležité zdůraznit, že existuje mnoho databázových systémů, včetně Microsoft SQL Server, MySQL, Oracle a mnoha dalších, jejichž principy fungování jsou sice stejné, ale samotné implementace těchto funkcí se značně liší.

2.4.1 Databáze

Pojem databáze si lze abstraktně představit jako místo, kde jsou všechna data ukládána.

Samotná struktura se skládá z jednotlivých tabulek, které definují vztahy, organizaci dat a celkové chování při práci s daty.

Každá z tabulek je rozdělena do několika sloupců, které jsou definovány za pomoci názvu, datového typu a příznaků.

Mezi klíčové prvky databáze můžeme zařadit:

- Tabulky – pro organizaci a ukládání dat formou záznamů.
- Sloupce – atributy záznamů, definovány za pomoci názvu a datového typu.
- Příznaky – přídavné vlastnosti, jako například primární klíč, cizí klíč, a mnoho dalších.

2.4.2 Systém báze řízení dat (DBMS)

Systém báze řízení dat (DBMS) je softwarové vybavení, které umožňuje manipulaci s daty, která jsou uložena v databázové části. Úkolem je zde zpracovávání příkazů pro operaci s daty, správu uživatelů či zabezpečení a mnoho dalšího. [6]

Mezi klíčové prvky patří:

- Zpracovávání příkazů – kompilace a provádění příkazů z dotazovacího jazyku, nejčastěji SQL.
- Správa bezpečnosti – autorizace pro přístup k datům.
- Optimalizace – efektivní provádění operací s daty.
- Zálohování a obnovování dat.
- Konzistence a integrita dat – ochrana před chybami při manipulaci s daty.

2.4.3 Způsoby přístupu k databázi při vývoji aplikace

Nedílnou součástí vývoje aplikace, která obsahuje jakákoliv data, je komunikace s příslušným databázovým systémem.

V dnešní době existují tři standardizované způsoby pro přístup a práci s databází:

1. Code First
2. Database First
3. Model First

2.4.3.1 Code First

Tento způsob přístupu k databázi je založen na principu, že jako první je vytvořen kód s příslušnými entitami, které představují následně vytvořené tabulky. Jsou v nich určeny jednotlivé sloupce s datovými typy, případně i určeny primární a cizí klíče nebo vztahy vůči ostatním entitám.

Vlastnosti mohou být určeny napevno vývojářem, nebo ponechány příslušnému kompilátoru, který dle názvů interpretuje definice jednotlivých sloupců a vygeneruje příslušné schéma.

Hlavní výhodou tohoto způsobu je možnost úpravy databáze přímo v kódu, bez nutnosti externích programů, či práci přímo s databází. Aktualizace databáze je zde prováděna pomocí operací, které se nazývají migrace. [7]

Mezi jeden z nejpoužívanějších prostředků pro komunikaci mezi kódem a databází je balíček Entity Framework Core.

2.4.3.2 Database First

Database First způsob vychází z principu, že je nejdříve vytvořena databáze s jasně definovaným schématem tabulek, včetně určených definic sloupců a vztahů, pokud nějaké existují, a na jejich základě se dále s touto databází propojuje kód aplikace pro čerpání, či ukládání záznamů.

Realizace tabulkového schématu je prováděna za pomoci systému řízení báze dat. [7]

Způsob je vhodný pro projekty s již existující databází, nebo v případě, že migrace pro rozšíření a modifikaci schématu by byla až příliš složitá.

2.4.3.3 Model First

Princip přístupu typu Model First lze popsat jako kombinaci technik Code First a Database First.

Zde dochází nejdříve k vizuálnímu návrhu schématu, ve kterém se definují tabulky, vlastnosti a vztahy, stejně jako u přístupu Database First, ačkoliv v tomto kroku se ještě nepracuje přímo s databází.

Poté, co je databáze plně navržena, je vytvořeno navržené schéma v databázi, společně s vygenerovaným kódem pro práci s touto nově vzniklou databází. [9]

Tento způsob je vhodný pro projekty, kde je návrh databáze klíčovou složkou vývoje softwaru.

2.5 MySQL

MySQL je jeden z nejpoblárnějších databázových systémů, který funguje na principu relačního databázového modelu, tzn. databáze se skládá z tabulek, které obsahují řádky (záznamy).

Při práci s databází se využívá, jak už název napovídá, příkazů za pomoci jazyka SQL. Tyto příkazy zde označovány jako „*query*“.

Oproti ostatním DBMS se MySQL liší svou architekturou, díky které ji lze použít pro mnoho různých účelů, ať už se jedná o jednoduché, lehké projekty jednotlivců až po komplexní sociální sítě, jako například Facebook, Twitter nebo další globální platformy, mezi které patří Netflix či Airbnb a mnoho dalších. [9]

2.5.1 Architektura

Zmíněná architektura je rozdělena do tří vrstev:

1. Klient
2. Server
3. Úložna

2.5.1.1 Klientská vrstva

Tato vrstva slouží pro interakci s MySQL databázovým systémem za pomoci SQL dotazů (výběr, ukládání/úprava/mazání záznamů apod.) [10]

Komunikace je zde zpracována skrze aplikaci, která potřebuje přístup k databázi a údajům v ní uložených.

2.5.1.2 *Serverová vrstva*

Zde jsou prováděny přijaté příkazy z klientské vrstvy, tudíž vrstva slouží jako část pro logické zpracování požadavků.

Skládá se taktéž z několika částí:

1. Zpracování připojených vláken – po zaslání žádosti klientem a potvrzení od serveru je klient připojen a je mu přiřazeno jeho vlastní vlákno, které zpracovává jeho příkazy.
2. Parser – interpretace a validace přijatých SQL příkazů
 - a. Optimizer – dochází zde k optimalizaci příkazů.
3. Cache a buffer – ukládání výsledků provedených příkazů, kontroluje, zda byl příkaz již dříve proveden, pokud ano, přeskočí provádění a vypíše již uložený záznam.
4. Tabulková cache pro metadata – informace o metadatech objektů.
5. Cache pro klíče – pro identifikování objektů, klíče jsou unikátní.

Mimo zpracování požadavků se také stará o autorizaci a autentizaci přístupů k databázi.

2.5.1.3 *Úložná vrstva*

Úložná vrstva určuje, jaký způsob ukládání byl uložen. Tento způsob se nazývá pluggable storage engine, díky jehož implementaci je ukládání mnohonásobně flexibilnější a může být užito pro splnění různých požadavků v oblasti výkonu či škálovatelnosti se zachováním SQL rozhraní a funkcionality databáze. [10]

Právě tento princip je jeden z největších rozdílů, díky kterým dávají vývojáři přednost MySQL oproti konkurenčním databázovým systémům.

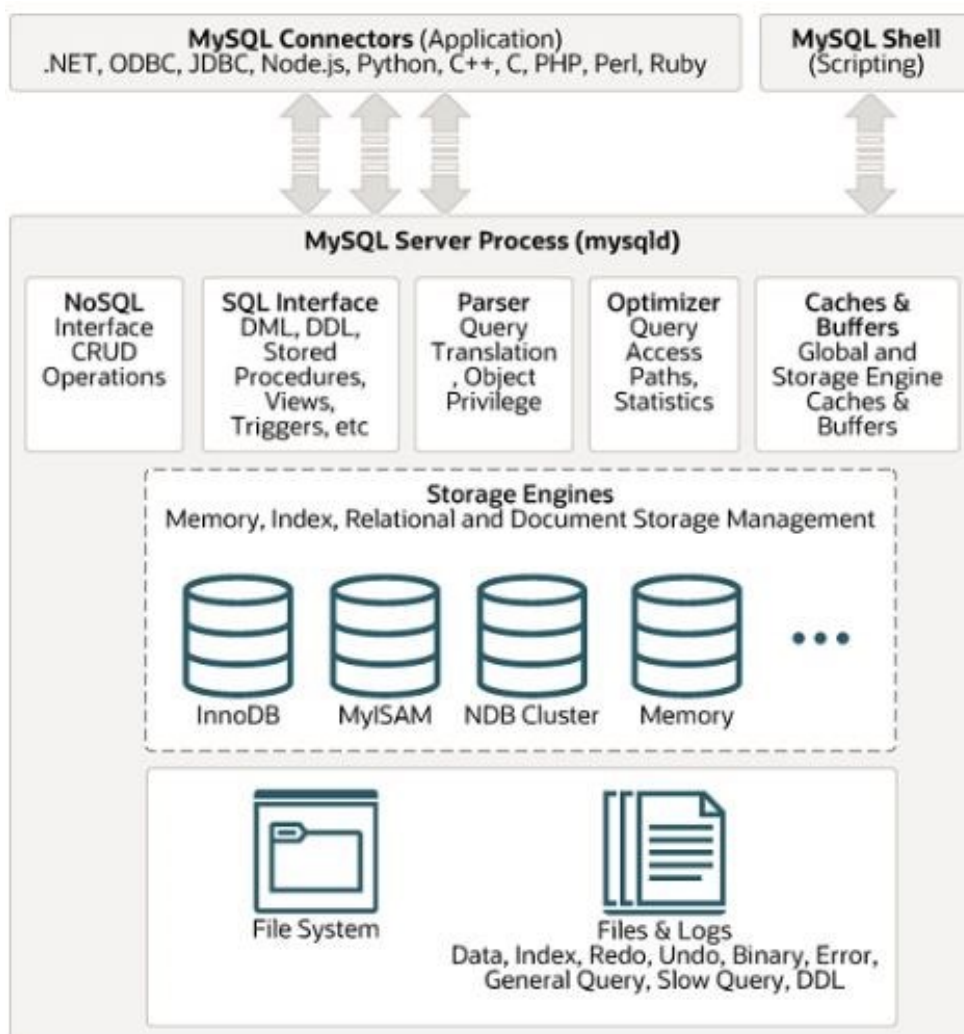
Mezi pluggable storage engine patří například:

- InnoDB – výchozí způsob, tabulky s primárními a cizími klíči.
- MyISAM – tabulky mají limitovaný výkon pro práci s tabulkami.
- CSV – ukládání do CSV souboru.
- Paměť – data jsou ukládána v RAM.

2.5.1.4 Princip komunikace

Komunikace mezi vrstvami probíhá následovně:

1. Klient zašle požadavek formou SQL příkazu.
2. Serverová vrstva požadavek přijme, zkontroluje, zda již nebyl proveden, pokud ano zašle se již uložený výsledek zpět, v opačném případě se pokračuje.
3. Plán pro vykonání je vytvořen kombinací parseru a optimalizace SQL příkazu.
4. Proveďte se příkaz zavoláním API vybraného pluggable storage engine.
5. Výsledek je zaslán zpět na klientskou vrstvu.



Obrázek 1. Architektura MySQL databáze [11]

2.6 Entity Framework Core

Pro práci s databází při využití způsobu code first, je potřeba frameworku pro objektově-relační modelování, který dokáže zpracovat vytvořené třídy, vztahy a další vlastnosti navrženého schématu databáze, vygenerovat z něj příkazy, a následně jej vyvolat v jazyku SQL v příslušném systému řízení báze dat.

Při komunikaci mezi kódem a databází se chová jako pomyslný prostředník.

Nejpopulárnějším ORM frameworkem pro tyto účely je NuGet balíček s názvem Entity Framework Core, který zpracovává výše zmíněný postup při práci s databází.

Lze jej využít jak pro základní operace typu vytvoření, či úprava databáze, až po pokročilejší, automaticky, či ručně definované propojování jednotlivých entit pouze na základě jejich definovaných vlastností. [12]

Další jeho výhodou je zaznamenávání historie migrací, jak v samostatné tabulce při vytvoření databáze, tak ve složce k tomu určené. Na základě těchto souborů lze provádět jak aktualizaci databáze, tak zpětné zotavení dřívější podoby databáze.

Je důležité zmínit, že balíček není vždy součástí vývojového prostředí a u některých je jej potřeba zvlášť stáhnout za pomoci manažera NuGet balíčků, společně s konkrétním přídatným balíčkem, který je specializován na vybraný databázový systém (Microsoft SQL server, MySQL, Oracle...).

2.6.1.1 Popis použití Entity Framework Core

Použití Entity Framework Core pro práci s databází se skládá několika jednoduchých kroků:

1. Vytvoření entit, které představují tabulkové schéma.
2. Určení takzvaného „*ConnectionString*“ který obsahuje přístupovou cestu k databázi, spolu s přihlašovacími údaji.
3. Vytvoření kontextu pro připojení k databázi a práci s daty v ní uloženými.
4. Provedení úvodní migrace, z níž je vytvořena databáze na základě entit.
5. Úprava entit, generování nových migrací a aktualizace existující databáze.

2.7 Vývojové prostředí

Vývojové prostředí je software, který umožňuje programátorům psát kód pro různé účely, od programování mikrokontrolerů, přes tvorbu mobilní či desktopové aplikace, až po webové stránky a pokročilejší, interaktivní webové aplikace.

Ve většině případů se jedná o programy, které se vždy specializují na jeden určitý programovací jazyk.

Mezi hlavní funkce softwaru tohoto typu je vhodné zařadit především editor zdrojového kódu, který je psán programátorem za pomoci syntaxí daného jazyku, následně kompilátoru, který zpracuje zdrojový kód, v průběhu může vývojáře upozornit na chyby, ať už syntaktické, či jiné, například problém s uvolňováním paměti, či nekonečná smyčka, a převede jej na kód strojový, se který je srozumitelný pro procesor, na kterém bude prováděn.

Mimo tyto funkce obsahuje také debugger, při kterém lze sledovat chod programu takzvaně „krok po kroku“ a ladit jej skrze sledování jednotlivých proměnných, obsahu zásobníku či jiných údajů, které jsou zde dostupné. [13]

Stejně tak jsou součástí pomocné funkce, mezi které můžeme zařadit zpracování přídatných balíčků vytvořených komunitou, zakomponované ovládání některých, nejpobulárnějších, balíčků, terminál pro pracování se systémem či jednoduché zpřehlednění skrze využití barev, zvýrazňování částí kódu.

Tyto funkce jsou jen jedny z mnoha, které vývojové prostředí umožňují a zpřijemňují vývoj softwaru pro programátory za účelem zvýšení produktivity.

2.7.1 Visual Studio 2022

Visual Studio 2022 je pobulární vývojové prostředí pro vývoj aplikací v jazyku C# za využití platformy .NET od firmy Microsoft. [14]

Nabízí širokou nabídku funkcí za účelem usnadnění vývoje aplikací pro jakoukoliv platformu, ať že jedná o aplikace mobilní, desktopové či webové.

Mezi hlavní výhody můžeme zařadit například:

- Funkce debuggeru, který umožňuje jednoduché ladění aplikací a sledování stavu programu v reálném čase, případně v daném kroku, pokud je použito krokování.
- Snadné integrované propojení s databázovým systémem, ať už systém Microsoft SQL Server, či jiný.

- Správa balíčků pomocí NuGet manažera, který umožňuje rozšíření funkcí aplikace za pomoci balíčků vytvořených komunitou.
- Integrace s cloudovými uložišti (Git) či sledování změn v týmovém vývoji (společné uložště).
- Velké množství rozšíření za pomoci přídatných pluginů vytvořených komunitou.
- Podpora i pro další programovací jazyky, jako je například Python nebo C++.

Vývojové prostředí je dostupné jak pro operační systém Windows, tak i na zařízeních se systémem macOS, avšak na těchto zařízeních je značně omezeno množství dostupných funkcí a konec podpory je naplánován na 31. srpna 2024.

2.7.2 JetBrains Rider

Mezi další populární vývojové prostředí můžeme zařadit program Rider od společnosti JetBrains. Jedná se o jedno z mnoha prostředí podporované již zmíněnou firmou, a stejně jako Visual Studio, se i Rider soustředí na programovací jazyk C# a prostředí .NET.

Funkčností i vzhledem se navzájem podobají a víceméně by se daly považovat za sobě rovné v případě porovnání. Jako jeden z rozdílů můžeme zmínit například integraci Entity Framework balíčku, který je zde zakomponován do prostředí s plnohodnotným grafickým rozhraním, oproti práci skrze příkazový řádek u Visual Studia.

Důležitý rozdíl však je, že Rider je multiplatformní. Zatímco Visual Studio mělo podporu pro Mac velmi omezenou, Rider ji má plnohodnotnou jak pro operační systém Windows, tak pro zařízení s macOS a také pro operační systém Linux.

2.8 Testování

Pro účely automatizovaného testování je dostupná široká nabídka balíčků, které poskytují všechny požadované funkcionality, ať už se jedná o provádění jednotkových testů, jejichž účelem je ověření jedné funkce aplikace, či integračních testů, kdy se již provádí komunikace mezi různými částmi aplikace.

V rámci webových aplikací, vyvíjených za pomoci ASP.NET je vhodné zmínit nejpopulárnější kombinaci balíčků, kterými jsou xUnit a Moq.

2.8.1 xUnit

xUnit je bezplatný nástroj pro zpracovávání a provádění automatizovaných jednotkových testů, při kterých se testují implementované funkce zvlášť. [15]

Příkladem může být test pro přihlášení či registraci, kde se provede vyplnění formuláře a následné odeslání, které se porovná na závěr s předpokládaným výsledkem.

2.8.2 Moq

Balíček Moq vychází z anglického pojmu „*mocking*“, v překladu by se dal pochopit jako klamání. Na základě klamání se při provádění testů jsou vytvořeny umělé objekty, ať už záznamy či přímo databáze, do kterých se tyto záznamy ukládají, jejichž cílem je následné ověření správnosti fungování implementovaných funkcí.

2.9 Knihovny uživatelských rozhraní

Tvorbu webových stránek a aplikací usnadňují knihovny, které obsahují komponenty, které může vývojář použít a dosáhnout tak jednotného vzhledu stránky, který už obsahuje v některých případech i zpracovanou logiku pro funkčnost komponenty.

Takových knihoven je obrovské množství, ať už se jedná o nejpoblárnější Bootstrap či Tailwind, které jsou postaveny na předem předpřipravených třídách za využití kaskádových stylů (CSS), či jiné knihovny, které již obsahují komponenty, jako jsou např. MaterialUI, který implementuje design od společnosti Google, FluentUI od Microsoftu, AntDesign, či mnoho dalších.

Tyto knihovny jsou ve většině případů postaveny na JavaScriptu a kvůli tomu může docházet k problémům při použití s frameworkem Blazor, který preferuje, aby byla funkcionalita zpracována v jazyce C#.

Naštěstí mnoho z výše zmíněných knihoven má vlastní projekty, které se odráží od hlavního projektu a zaměřují se na předělání do jiných frameworků. Jeden z takových projektů je právě AntBlazor.

2.9.1 AntBlazor

AntBlazor je knihovna komponent uživatelského rozhraní, která vychází z knihovny AntDesign, avšak její funkcionalita a logika je předělána z JavaScriptu do C# speciálně za účelem využívání s Blazorem pro tvorbu interaktivních a bohatých webových aplikací.

JavaScript zde není kompletně vyřazen, ale jeho použití je výrazně zredukováno.

Knihovna umožňuje vytvářet a přizpůsobovat uživatelská rozhraní za pomoci široké nabídky komponent a možností nastavení.

Mimo komponenty, které lze označit za „základní“, jako je například grid pro rozdělení stránky do pomyslných sloupců tak, že jsou rovnoměrně rozloženy, či rozložení stránky na hlavičku, boční menu, patičku a obsah, se zde nachází i mnoho vlastních implementací, jako je například vlastní zpracování vstupů od uživatele a formulář, zobrazení postupu za pomoci ukazatelů, karty, grafy, záložky, modální, šuplíková okna a mnoho dalšího.

Jelikož se jedná o open-source projekt, tak se může kdokoliv zapojit a pracovat na vývoji, či přidávat vlastní implementace, nebo rozšiřovat již existující funkce.

Důležité je zmínit, že knihovna je chráněna licencí typu MIT.

2.9.2 ApexCharts

ApexCharts je jedna z dostupných JavaScriptových knihoven, které se zaměřují na usnadnění vizualizace dat na stránce skrze využití již předpřipravených grafů.

Na výběr je z obrovské nabídky typů grafů od jednoduchých spojnicových, či sloupcových přes koláčové až po složitější, či zobrazení dat za využití teplotní mapy či korelačního (bodového) grafu.

Všechny dostupné grafy jsou flexibilní a plně responzivní společně s širokou nabídkou vlastních přizpůsobení.

Jedná o open source projekt, který je chráněn licenci typu MIT.

2.9.2.1 Blazor ApexCharts

Stejně jako u AntDesignu a AntBlazoru, i u knihovny ApexCharts dochází k problému při jejím využití v Blazoru kvůli tomu, že je primárně navržena pro využití v JavaScriptu.

Jako řešení byl vytvořen balíček s názvem Blazor ApexCharts, který umožňuje využívat grafy z knihovny ApexChart. Jeho princip je postaven na tom, že slouží jako kompilátor, který zpracuje syntaxi jazyka C# tak, aby na jejím základu mohly být grafy sestaveny.

Tím je značně usnadněna integrace do webové aplikace, díky jednotné jazykové platformě a minimalizaci JavaScriptu.

2.10 Softwarové licence

Všechny vyvíjený software by měl být chráněn určitým typem licence, který definuje možnosti distribuce a používání daného programu.

Existuje mnoho softwarových licencí, každá s jinou definicí a odlišnými pravidly.

Mezi nejčastější používané licenční modely patří:

- MIT Licence.
- Apache Licence.
- BSD Licence.

Obsah licencí je definován a zpracováván právníky s ohledem na potřeby a účely daného softwaru. Vývojáři dále vybírají nejvhodnější typ licence pro jejich projekt.

2.10.1 MIT Licence

Všechny doposud uvedené technologie jsou kryty licencí MIT.

Samotná licence vznikla na Massachusettském technologickém institutu a umožňuje daný software užívat k jakýmkoliv účelům.

Uživatel má plnou pravomoc pro modifikaci, kopírování a distribuování softwaru, včetně komerčních účelů. Jedinou podmínkou při využití této licence je, že musí být přiložena jako součást distribuovaného softwaru.

3 HOSTING

Hosting, či hostingová služba, lze popsat jako pronájem prostoru na vzdáleném serveru, do kterého lze umístit svou webovou aplikaci či své webové stránky, čímž následně mohou být přístupné uživatelům za pomoci domény.

Prostor nemusí být nutně jen na cizím zařízení, ale může také být hostován na zařízení vlastním, pokud má hostitel dostatečný výpočetní výkon.

Samotný obsah je uložen v serverech, které jsou připojeny k internetu a v případě, že je obsah uložen na serveru provozovatele služeb, je často poskytnuto rozhraní pro vzdálenou správu prostoru, do které může patřit jak nástroje pro správu stránek, tak pro práci s databází.

V případě, že bude aplikace využívat cizí hostování, cena provozu zařízení se nejčastěji řídí podle zátěže na webovou aplikaci a jejich požadavků na výpočetní výkon. [16]

Obecně lze hosting označit za poslední krok při nasazování aplikace do provozu.

3.1 Typ hostované aplikace

Při výběru hostingu je však důležité dbát na druh produktu, který zde má být provozován, může se jednat buď o webovou aplikaci (stránku) statickou, či dynamickou.

3.1.1 Statická webová aplikace

Statická webová stránka obsahuje nulovou interaktivitu, její obsah se nemění, pokud na ni uživatel neprovede operaci, při které se odešle požadavek na server.

Tak jak byla uložena na server, je posílána klientovi za účelem vykreslení.

3.1.2 Dynamická webová aplikace

Dynamická webová aplikace oproti tomu umožňuje vše, co aplikace statická postrádala. Interaktivitu, reakce na vstup uživatele, odezvu na požadavky, modernější a přívětivější prostředí, ve kterém se klient bude nacházet. Spolu s touto zvýšenou a zmodernizovanou funkcí aplikace bude však také růst zatížení na hosting, jelikož se ve většině případů mnohonásobně zvýší počet odeslaných a přijatých požadavků, které slouží pro aktualizaci obsahu.

3.2 Možnosti hostingu

Jakmile je znám typ aplikace, kterou umístíme na server, je k dispozici obrovské množství poskytovatelů těchto služeb, jak placených, tak těch, kde je možnost hostování zdarma, s omezenými funkcemi.

3.2.1 Hosting statických aplikací

Možností pro hostování statických aplikací a webových stránek je nepřehledné množství především z toho důvodu, že se jedná o původní způsob, jakým byl obsah internetu zobrazován.

Výběr poskytovatele v tomto případě spočívá pouze na nabídce, která bude nejvíce vhodná pro chod aplikace.

Mezi hlavní parametry zde bude patřit poměr ceny/výkonu, nabídka množství uložení, či podpora skrze poskytované vývojářské nástroje.

Pro provoz tohoto typu stránek můžeme využít například služby těchto poskytovatelů:

- Endora – jeden z nejstarších českých poskytovatelů webhostingu.
- Microsoft Azure – plná podpora pro hostování statických webových aplikací, publikování usnadněno díky implementaci funkcí přímo ve Visual Studiu.
- Google Cloud – součást ekosystému od Google, mimo zpracování výpočetních úloh umožňuje také hostování stránek.
- Github Pages – zdarma hostování stránek z daného repozitáře.

3.2.2 Hosting dynamických aplikací

V porovnání se statickými aplikacemi jsou aplikace dynamické výrazně mladší. Vzhledem k jejich zpracování a dostupným funkcionalitám vyžadují o mnoho více výpočetního výkonu pro hostování a provoz tohoto obsahu.

Zde nelze vybrat hosting pouze dle parametrů, ale také je nutné sledovat, zda vůbec umožňuje dynamické aplikace publikovat.

Jako typické zástupce poskytovatelů můžeme zmínit:

- Microsoft Azure – hostování dynamických aplikací společně se statickými.
- Google Cloud.
- Amazon Web Services – cloudová služba obdobně jako Google Cloud, která poskytuje výpočetní výkon, včetně hostování dynamických webových aplikací.

4 ENERGETICKÝ VÝDEJ

Pro zpracovávání energetického výdeje u aktivit je důležitý vhodný výběr způsobu pro výpočet spálených kalorií. Zde je možné použít dvě hlavní metody.

Tou první je výpočet spálených kategorií na základě tepové frekvence. Tento princip je výhodný při použití sledovacího zařízení, nejčastěji chytrých hodinek, či fitness náramku, které průběžně monitorují hodnotu tepu.

Mnohem vhodnější je však využití jednotky metabolického ekvivalentu s označením MET (Metabolic Equivalent of Task), díky kterému dostaneme obecnější přehled, který je méně přesný, co se týče spálených kalorií, avšak jeho použitelnost je o mnoho větší.

4.1 Metabolický ekvivalent (MET)

Hodnota metabolického ekvivalentu slouží pro měření energické náročnosti aktivit vykonávaných člověkem.

Princip tohoto způsobu měření spočívá v tom, že každá z aktivit má přidělenou určitou MET hodnotu, která znázorňuje předpokládaný výdej energie v klidovém stavu. [17]

Pro klidový stav, tedy stav, při kterém člověk nevykonává žádnou intenzivní práci jako je například čtení, čtení televize, sezení, má aktivita hodnotu přibližně 1 MET. S narůstající intenzitou, tedy chůze, poklus, běh a další, roste i hodnota MET.

Na základě MET hodnoty, společně s dobou trvání aktivity a váhou člověka, můžeme dále vypočítat odhadovaný počet spálených kalorií za jednotku času.

4.1.1 Výpočet metabolického ekvivalentu

Pro obecný výpočet spálených kalorií, využíváme hodnoty MET spolu s váhou člověka a dobou trvání dané aktivity v hodinách.

Tyto údaje poté dosadíme do vzorce:

$$\text{Spálené kalorie} = \text{MET hodnot} \times \text{váha člověka} \times \text{trvání aktivity v hodinách}$$

4.1.2 BMI

Vzhledem k energetickému výdeji je také vhodné zmínit tzv. BMI (Body Mass Index), který slouží jako ukazatel toho, zda je váha uživatele, vzhledem k jeho výšce v přiměřeném rozmezí či nikoliv.

Výpočet tohoto indexu je následovný:

$$BMI = \text{tělesná váha} \times (\text{tělesná výška})^2$$

Samotný výpočet může být detailnější a skládat se z mnoha upřesňujících informací, jako je věk, pohlaví či podíl svalové hmoty jedince, avšak pro rychlý a jednoduchý přehled je výše uvedený vzorec dostačující.

II. PRAKTICKÁ ČÁST

5 ANALÝZA POŽADAVKŮ APLIKACE

Mezi základní postupy při vývoji aplikace se řadí analýza vyvíjené aplikace, přesněji je vhodné se zaměřit jako první přímo na požadavky daného produktu.

Tento návrh vychází z potřeb zákazníka, které se získávají za využití dotazníků, rozhovorů či jiných metod. Výsledný přehled usnadňuje vývojářům strukturovanější přístup k vyvíjení jednotlivých částí aplikace.

Tuto vývojovou etapu si lze představit jako pomyslný návrh a plánování toho, jak bude aplikace fungovat. Navržený přehled požadavků můžeme rozdělit do dvou kategorií, kterými jsou požadavky funkční a požadavky nefunkční.

Pro zpracování této části vývoje lze použít mnoho dostupných programů, které jsou zaměřeny na tuto problematiku, jako je například Enterprise Architect, Wireflow, či jakýkoliv jiný.

5.1 Funkční Požadavky

Jako funkční požadavky se rozumí funkcionality aplikace, tj. co bude poskytováno uživateli za služby, jak budou fungovat, co všechno bude vyvíjený software umět.

Lze je zpracovávat buď jako požadavky obecné, například registrace nebo přihlášení, stejně tak jako je lze detailně popisovat, vytvářet grafický návrh toho, jak se budou navzájem realizovat, či jakým stylem budou dány jejich vztahy. Zde záleží na individuálním posouzení toho, jak detailně je daný požadavek vhodné zpracovat.

Jako příklad zde můžeme uvést skupinu požadavků z aplikace vyvíjené pro tuto práci – výpočet postupu cíle, pokud je vybráno hubnutí.

V tomto případě by měla aplikace obsahovat následující požadavky:

- Požadavky pro cíle – práce se záznamy, vytvoření/úprava/smazání záznamu.
- Výpočet postupu cíle – Převod hodnoty na kilogramy a porovnání s cílovým požadovaným množstvím.
- Výpočet kilokalorií u tréninku – suma všech kilokalorií z propojených aktivit
- Výpočet aktivit – výpočet spálených kilokalorií na základě MET hodnoty dané aktivity spolu s dobou trvání a váhou uživatele.

5.1.1 Účet

Mezi funkční požadavky, týkající se účtu v této aplikaci, můžeme zařadit, jak typické funkcionality jako je registrace a přihlášení k danému účtu, můžeme zařadit také změnu údajů. Zde se jedná jak o údaje týkající se účtu, tj. emailová adresa, přihlašovací jméno či heslo, tak o osobní informace jako je jméno a příjmení, pokud je uživatel chce sdílet, spolu s údaji, které jsou následně zpracovány pro výpočet dalších informací a obohacení tak účtu uživatele za účelem přesnějších výpočtů při práci s daty.

Níže jsou uvedeny některé příklady pro tyto zmíněné vztahy:

- Zadání data narození – výpočet věku uživatele.
- Zadání výšky a váhy – výpočet a zobrazení BMI indexu uživatele, přesnější výpočet intenzity dané aktivity.

Do této kategorie spadají také milníky, jimiž je zprostředkován aspekt gamifikace v aplikaci za účelem motivování uživatelů jak k využívání aplikace, tak pro plnění jejich individuálních cílů.

Tyto milníky se zobrazují nejen uživateli na profilu, ale je jim také umožněno tyto úspěchy sdílet s lidmi, kteří je sledují, viz. Sociální aspekt.

5.1.2 Data

Jako požadavky týkající se dat se rozumí všechny způsoby, jak je s uloženými daty nakládáno. Od zdroje dat, jejich vkládání do záznamů až po jejich zpracovávání a využívání k výpočtům.

Veškeré záznamy jsou rozděleny do několika samostatných kategorií, každá z nich je tvořena vlastní stránkou, zde je jejich přehled:

- Profil a stránka pro sociální aspekt, účelem je sdílení milníků a vyhledávání jiných uživatelů, případně sledování jejich úspěchů.
- Aktivity, tréninky a cíle – tyto stránky jsou vzájemně propojeny.
- Spánek, Zdraví.
- Strava – tvořena kaloriemi a pitným režimem.
- Wellbeing – duševní zdraví tvořeno deníkem a záznamy nálad.
- Historie všech záznamů – Vyhledávání či třídění dle jednotlivých kategorií.

5.1.2.1 Zdroj dat

Požadavky pro zdroje dat se zde rozumí způsoby, jakými jsou údaje zadávány do aplikace. Ať už se jedná o import ze souboru, nebo častěji, v případě této aplikace, ručním zadáváním za použití formulářů ve webové aplikaci.

Tyto formuláře jsou unikátní pro každou požadovanou kategorii záznamů, s vlastními vstupními parametry.

5.1.2.2 Ukládání a načítání dat

Ukládání dat se soustředí na otázku toho, kde jsou data ukládána a načítána uživateli. Při zadání požadovaných parametrů daného záznamu jsou data zpracována, a pomocí metod vložena do příslušné databázové tabulky jako nový řádek.

Zjednodušený postup vypadá následovně:

1. Uživatel vyplní formulář a odešle jej pro vytvoření záznamu.
2. Při spuštění metody se do záznamu přidá ID uživatele.
3. Metoda s nově zadaným, rozlišovacím ID uživatele je uložena do databáze.

Opačným způsobem pak funguje jejich načítání, kdy se načtou ta data z požadované tabulky, kde je ID uživatele shodné s ID uživatele, který o data žádá.

Zde je důležité zmínit, že rozlišování dat je zprostředkováno za využití ASP.NET Identity frameworku.

5.1.2.3 Práce s daty

Do kategorie, týkající se práce s daty, můžeme zařadit požadavky, které uložené záznamy následně zpracovávají za různými účely.

Těchto požadavků je v aplikaci celá řada, pro lepší představu účelu jsou zde jsou některé příklady:

- Výpočet průměrné délky spánku.
- Průměrná nálada.
- Nejčastější typ aktivit.
- Přehled nejvíce propojených aktivit s jedním tréninkem.
- Převod spálených kilokalorií na kilogramy u cílů.
- Již zmíněný výpočet BMI díky výšce a váze uživatele.

5.1.3 Sociální aspekt

Sociální aspekt je zde zpracován formou sdílení dosažených milníků, vyhledávání a sledování jiných uživatelů. Zde si je možno povšimnout jednoho ze vztahů propojených požadavků:

- Milníky uživatele jsou propojeny a zobrazují jiným uživatelům díky zpracování požadavku na sledování – závislost jednoho požadavku na dalších.

5.2 Nefunkční požadavky

Nefunkční požadavky se již netýkají nabídky funkcí aplikace, ale samotného systému, způsobu, jakým bude aplikace tvořena a jakých cílů by měla dosahovat v rámci technologií.

Pod tímto popisem si můžeme představit následující požadavky, které byly brány v potaz při vývoji aplikace:

- Výkon – Aplikace by měla být dostatečně optimalizovaná, aby nebyla negativně ovlivněna uživatelská zkušenost, a to i v případě velkého množství dat.
- Škálovatelnost – Snadná možnost přidání nových funkcí bez nutnosti přílišného zásahu do existujícího kódu.
- Bezpečnost – Záznamy by měly být dostatečně zabezpečené, zpracováno hlavně pomocí ASP.NET Identity a MySQL praktik a principů pro ochranu dat.
- Uživatelská přívětivost – Aplikace by neměla být složitá na úkor uživatelské zkušenosti a zároveň poskytovat co největší nabídku funkcí.
- Responzivita – Přizpůsobení způsobu, jakým je obsah zobrazován i na menších obrazovkách.

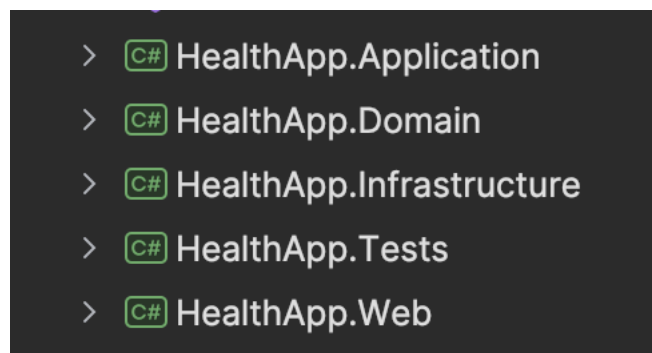
Nefunkční požadavky jsou stejně důležité jako požadavky funkční z toho důvodu, že jsou zaměřeny na kostru daného produktu, ze které vychází všechny dostupné funkcionality.

6 STRUKTURA PROJEKTU

Při vývoji aplikace a pro její funkcionalitu byla vybrána třívrstvá architektura aplikace. Tímto je myšleno, že celková logika je rozdělena do 3, případně 5 vrstev, kde každá z nich má svůj přidělený účel:

1. Webová vrstva.
2. Aplikační vrstva.
3. Infrastrukturní vrstva.
4. Doménová vrstva.
5. Testovací vrstva.

Tuto logiku lze pochopit jako izolování skupin souborů a zpřehlednění kódu při vývoji.



Obrázek 2. Rozdělení vrstev

6.1 Webová vrstva

Ve webové vrstvě, někdy také označované za prezentační vrstvu, se nachází všechny soubory, které zpracovávají a zobrazují obsah uživateli. Ať už se jedná o Razor page soubory, které v sobě kombinují HTML elementy spolu s C# kódem v složce označované jako *Components*, či soubory, které pracují na pozadí této vrstvy. Zde řadíme kaskádové styly (CSS) a JavaScript.

Stejně tak se zde nachází výchozí konfigurace aplikace. Ať už se jedná o nastavení aplikace, ve kterém je definováno připojení k používané databázi, či samotná konfigurace aplikace, kde můžeme najít jak Dependency Injection (DI), skrz kterým umožníme používání služeb, či konfigurace připojení databáze a ASP.NET Identity knihovny.

Také je zde přijímána interakce a vstup uživatele, který je následně předáván do aplikační vrstvy pro zpracování.

6.2 Aplikační vrstva

Aplikační vrstva se stará o celkovou logiku aplikace. Zatím co webová vrstva pouze přijímala a předávala interakci s uživatelským rozhraním, zde se již provádí potřebné operace. Tyto metody jsou implementovány v podobě služeb.

Celkově se dělí vrstva do dvou skupin:

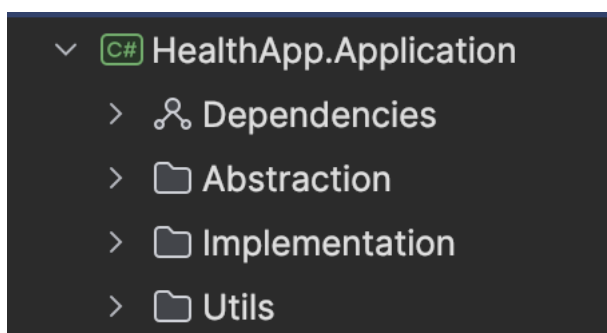
- Abstrakce,
- Implementace.

V prvně jmenované se nachází takzvané rozhraní služeb, které abstraktně popisují metody, jejichž implementace je následně zpracována v implementační části.

Mimo popis služeb a zjednodušený přehled metod, je účelem rozhraní také propojení a zpřístupnění metod v implementaci do jiných vrstev, jmenovitě především pro volání z webové vrstvy.

Mezi nejčastější metody, které zde lze najít, patří CRUD (Create, Remove, Update, Delete) operace, tzn. práce se záznamy v databázi. Mimo tyto metody zde můžeme nalézt například výpočty pro spočítání celkově spálených kalorií, převod kalorií na kilogramy, či spočítání postupu cíle.

Taktéž se zde nachází metody, které zpracovávají obsah stránek, ať už se jedná o počet záznamů, datum posledního záznamu, globální přehled statistik u dané aktivity a mnoho dalšího.



Obrázek 3. Struktura aplikační vrstvy

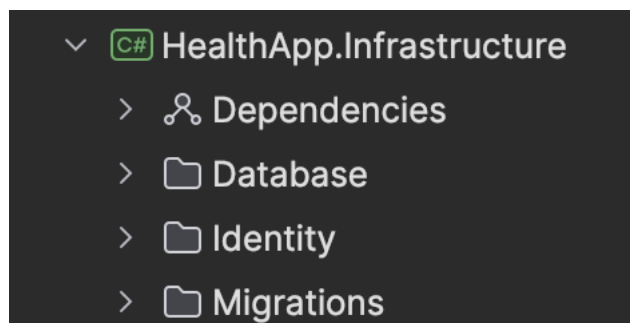
6.3 Infrastrukturní vrstva

Pro práci s databází je však nutné, aby existovalo spojení, které propojuje aplikaci s daným databázovým systémem.

Za tímto účelem je vytvořena infrastrukturní vrstva, díky které můžeme vytvořit a nakonfigurovat propojení s databází a pomocí souboru databázového kontextu. V kombinaci s Entity Framework Core je následně umožněno v aplikační vrstvě provádět operace s údaji v databázi. Při práci s databází jsou vytvářeny migrace, které umožňují upravit databázi, pokud vygenerovaný kód není dostačující. Migrace jsou následně použity k aktualizování databáze, čímž jsou změny provedeny v samotné struktuře.

Mimo výše zmíněného databázového kontextu a migrací, zde můžeme zařadit i výchozí data, kterými můžeme naplnit aplikaci už při jejím vytvoření, bez jakéhokoliv klientského vstupu.

Tato operace se označuje jako „*seedování*“ databáze.



Obrázek 4. Struktura infrastrukturní
vrstvy

6.4 Doménová vrstva

V poslední, doménové vrstvě se nachází entity, se kterými se pracuje při tvoření tabulek ve skutečné databázi za použití code-first přístupu.

Zde se nachází pouze třídy s definovanými sloupci, které jsou následně zpracovány za pomoci Entity Framework Core knihovny a převedeny do databázové struktury. Stejně tak se zde nachází křížové tabulky, či výčtové soubory, ve kterých se nacházejí údaje, u nichž by bylo zbytečné tvořit samostatné tabulky a tím zatěžovat celkovou strukturu databáze.

Přesto, že projekt je tvořen čtyřmi vrstvami, vrstvu doménovou lze označit pouze za pomocnou, pro zpřehlednění celkového projektu při vývoji.

Ve výsledku, jak již bylo zmíněno, je aplikace založena na třívrstvé architektuře.

6.5 Testovací vrstva

Poslední vrstva v sobě uchovává pouze testy pro ověření správného fungování stránky. Vzhledem k tomu, že nijak nezasahuje do komunikace mezi webovou, aplikační a infrastrukturní vrstvou, a tudíž nenabourává třívrstvou architekturu, nebyla zmíněna do celkového počtu a lze si ji představit jako nepovinnou.

Samozřejmě pro správné fungování aplikace je vhodné vytvořit testy vzhledem k tomu, že vývojář nemá šanci odhalit všechny problémy, které mohou nastat.

7 NÁVRH DATABÁZE

Při vývoji jakékoliv aplikace je jednou z nejdůležitějších částí správný návrh databázové struktury, ve které budou data ukládána. Správná či naopak špatná implementace může ovlivnit celý životní cyklus jak při vývoji, tak v horším případě i potom, co dojde k publikaci aplikace.

7.1 Databázová struktura

Databázovou strukturu samotné aplikace lze rozdělit na dvě hlavní části. Tabulky týkající se údajů v aplikaci a na část s tabulkami týkající se autentizace a autorizace za využití balíčku ASP.NET Identity.

Všechny tabulky z aplikační části obsahují vlastnost „*UserId*“, která slouží jako identifikátor uživatele pro získání správných záznamů při načítání. Tento způsob rozlišení dat je zde umožněn díky ASP.NET Identity balíčku a jím vytvořené tabulce o uživateli.

7.1.1 Tabulky údajů v aplikaci

Pro každou oblast aplikace jsou vytvořené příslušné tabulky, které obsahují vlastní atributy pro správné zaznamenávání a případnou práci s těmito vloženými údaji.

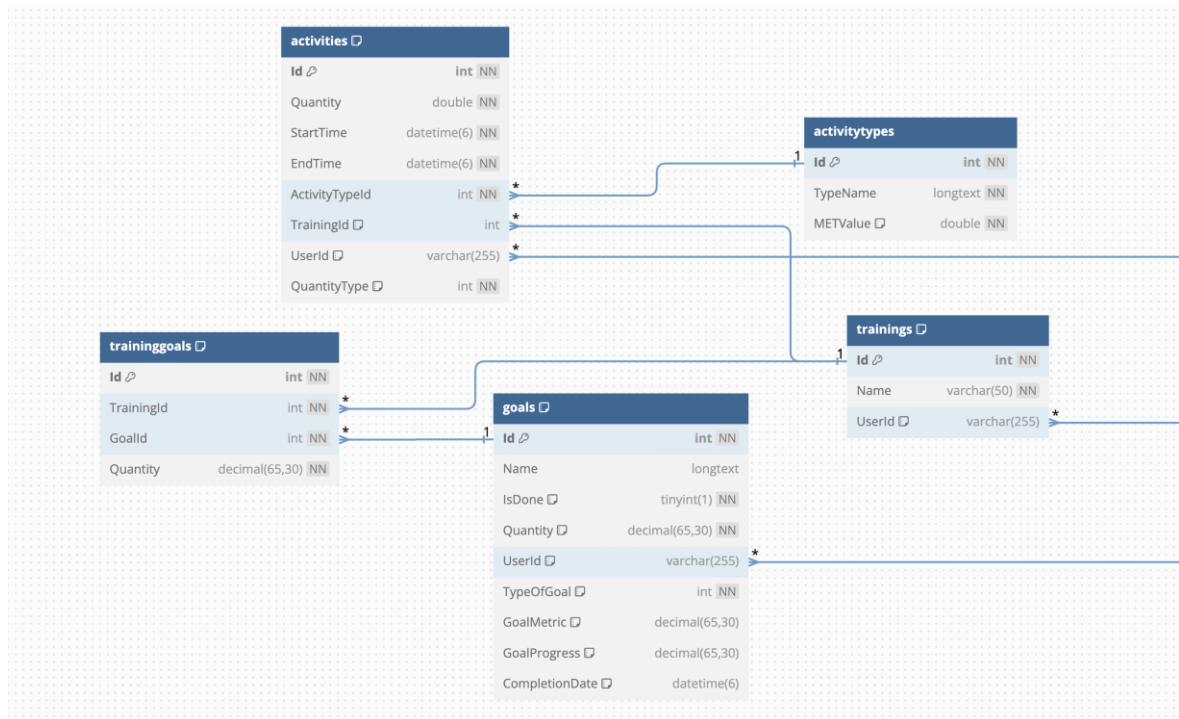
7.1.1.1 *Aktivity, Trénink a Cíle*

Jako první celek struktury lze skloubit tabulky týkající se aktivit, cílů a tréninků. Tabulka aktivit slouží pro zaznamenávání údajů jako je například název, či počáteční společně s konečným časem aktivity a následně samotný typ, tj. běh, plavání, či jiné, který je však zpracován jako samostatná tabulka.

Vzhledem k zjednodušení přehlednosti a případného vylepšení, či přidávání nových typů aktivit, jsou typy zpracovány v samotné tabulce, která je s hlavní propojena vztahem M:1. Součástí této tabulky je mimo názvu aktivity také hodnota MET pro výpočet spálených kalorií.

Stejně tak je aktivitní tabulka propojena s tabulkou sledující tréninky, který slouží jako pomyslný spojovací článek mezi aktivitami a cíli. Vztah je zde definován jako M:1, což odráží samotnou logiku aplikace.

Ovšem u propojení tréninku s cíli je vztah mezi tabulkami dán za pomoci vztahu M:N. V diagramu znázorněn za pomoci křížové tabulky s názvem „*traininggoals*“. Tato usnadňuje přehlednější zaznamenávání a sledování údajů uživatele.



Obrázek 5. Datová struktura aktivit, tréninků a cílů

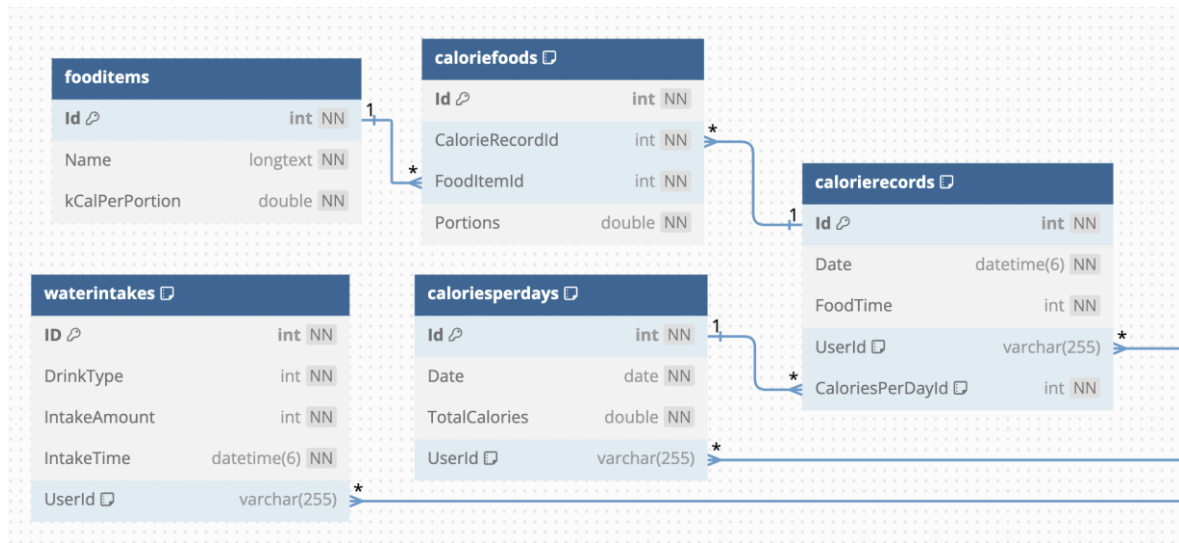
Samotné cíle obsahují mimo typických atributů typu, data a množství, také atribut, kterým může uživatel sledovat, zda již cíl splnil, či nikoliv společně s hodnotou pro splněné množství cíle a jeho postup za účelem grafického znázornění.

7.1.1.2 Strava

Další seskupení tabulek se týká funkcionalit pro zaznamenávání přijaté stravy a pitného režimu.

První část této skupiny se stará o uchovávání kalorických záznamů, tj. záznam skládající se z data a typu jídla s přijatým množstvím. Samotná struktura tabulky, jak lze vidět z obrázku, se skládá z hlavní „*calorierecords*“ tabulky, která je propojená skrze „*caloriefoods*“ křížovou tabulku na záznamy, které obsahují dostupná jídla.

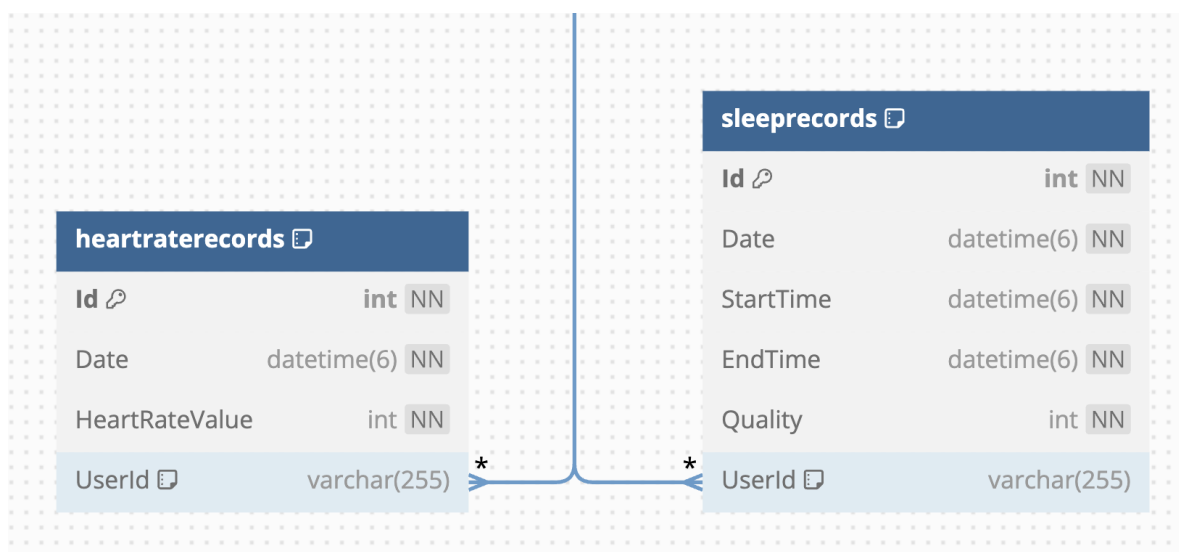
Tento celkový záznam je poté zpracováván a ukládán do tabulky „*caloriesperday*“, díky kterému lze uživateli zobrazit přehlednou historii sumy přijatých kalorií dle dnů jednotlivých dnů.



Obrázek 6. Datová struktura pro oblast stravy

7.1.1.3 Spánek a Srdeční tep

Další částí, na kterou lze strukturu rozdělit, jsou tabulky týkající se údajů o spánku a srdečního tepu.



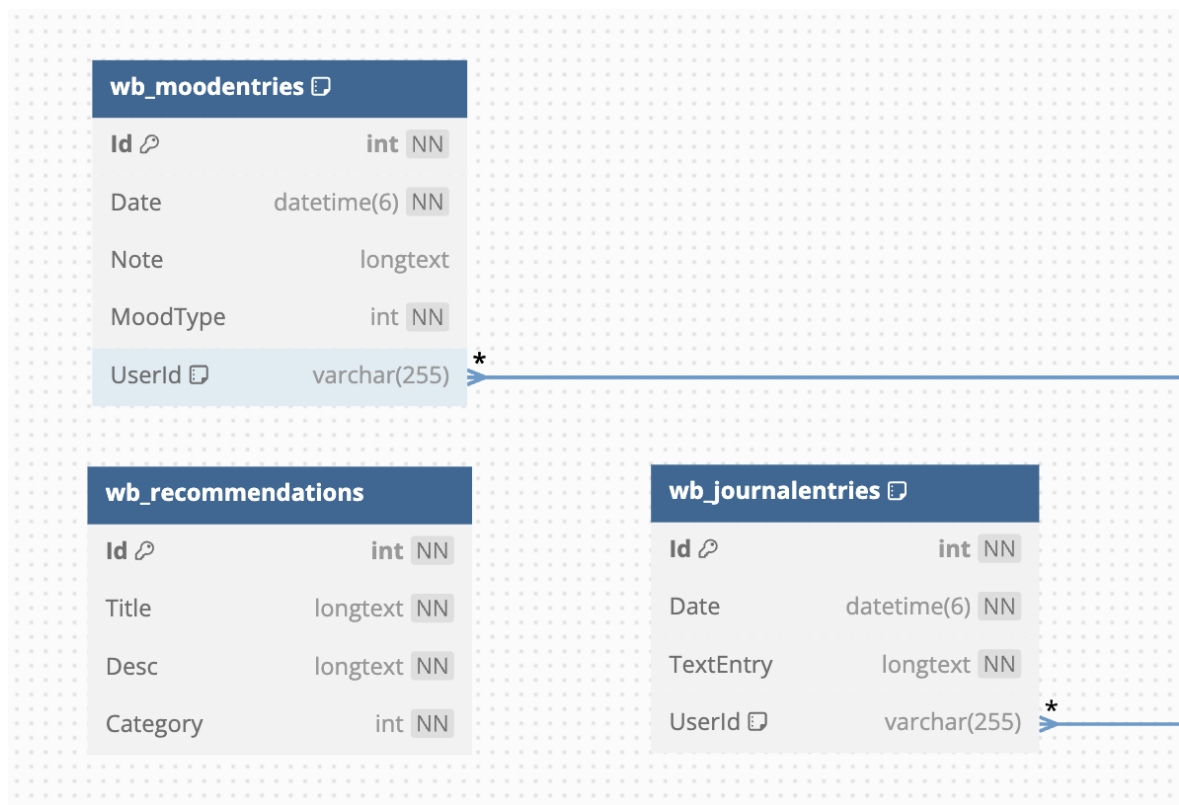
Obrázek 7. Datová struktura pro spánek a srdeční tep

Tabulky se skládají ze základních údajů pro čas, v případě spánku dochází k výpočtu délky spaní rozdílem těchto dvou hodnot, a také z vlastností, které je vhodné zde zaznamenávat, tj, hodnota srdečního tepu a kvalita spánku pro analýzu spánku, ať už se bude jednat o denní záznam nebo týdenní průměr.

I zde, jako v ostatních případech, si lze povšimnout, že kvalita má datový typ int, to je z toho důvodu, že hodnota je propojena s enum souborem, který obsahuje příslušné hodnoty, které jsou pro uživatele příjemnější – dobrá, špatná, hrozná nálada, aj.

7.1.1.4 Duševní zdraví

Další část aplikace, které se týká duševního zdraví, má databázovou strukturu rozdělenou do tří samostatných tabulek. Tabulka pro záznam do deníku, tabulka pro záznam nálady a tabulka, ve které jsou uložena doporučení pro uživatele.

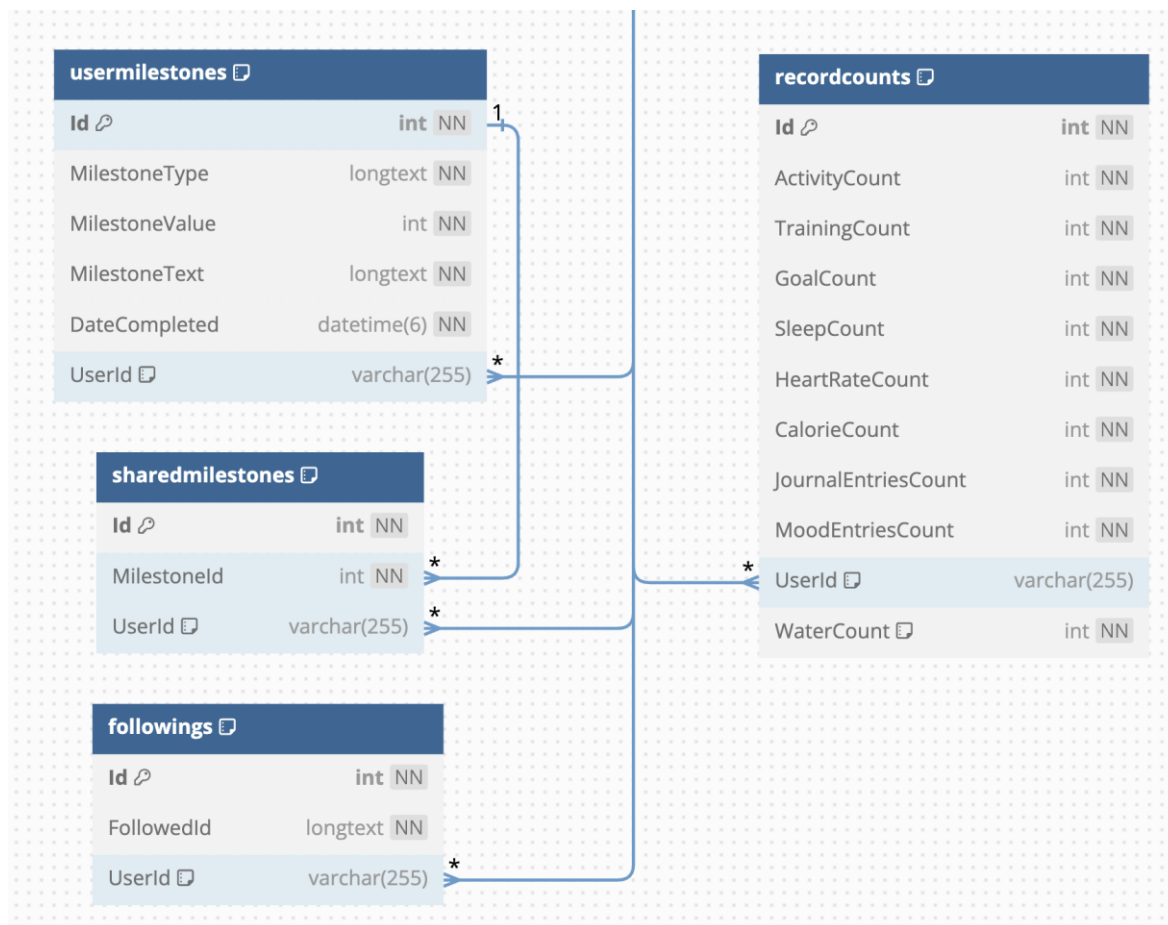


Obrázek 8. Datová struktura pro tabulky duševního zdraví

Samotná tabulka pro záznam do deníku je čistě pro ukládání obsahu záznamu spolu s datem. Oproti tomu tabulka pro nálady se skládá mimo data a poznámky také z typu nálady, který je opět propojen se souborem typu enum, kde každá nálada má vlastní číselnou hodnotu. Tím se sleduje průměrná nálada za pevně dané období – týden, měsíc... Doporučení však obsahují pouze název, popis a kategorii, nejsou zde propojeny s žádnou další tabulkou.

Účelem této separace je načítání náhodných doporučení pro uživatele, které mohou ale nemusí vycházet z jeho nálady.

7.1.2 Gamifikace, sociální aspekt



Obrázek 9. Datová struktura pro gamifikaci a sociální aspekt

7.2 ASP.NET Identity

Při vytvoření databáze za využití ASP.NET Identity dojde k vytvoření tabulek, za jejichž pomocí je zpracována autentizace společně s autorizací v rámci celé aplikace.

Tato část je rozdělena do několika menších tabulek společně s jednou hlavní, nazvanou jako „*aspnetusers*“, která se stará o samostatné zaznamenávání uživatele, ukládání a sledování jeho údajů.

Mimo základních údajů, jako je uživatelské jméno, křestní jméno a příjmení, email a normalizované verze těchto údajů, zde byly přidány i vlastní atributy jako je výška a váha uživatele, či jejich denní kalorický a pitný cíl spolu s datem narození, věkem vypočítaným z tohoto údaje.



Obrázek 10. Datová struktura tabulek vytvořených skrze ASP.NET Identity

Co se týče zbylých tabulek, ty se starají o případné role, které zde však nejsou použity čistě z důvodu toho, jaký má aplikace účel. Zbylé struktury databáze se skládají z údajů pro ověřování při registraci a následném přihlášení.

8 UŽIVATELSKÉ ROZHRANÍ

V následující kapitole je popsána funkcionalita a používání samotné aplikace. Tuto část lze také brát jako pomyslný návod pro neznalého uživatele.

8.1 Úvodní stránka

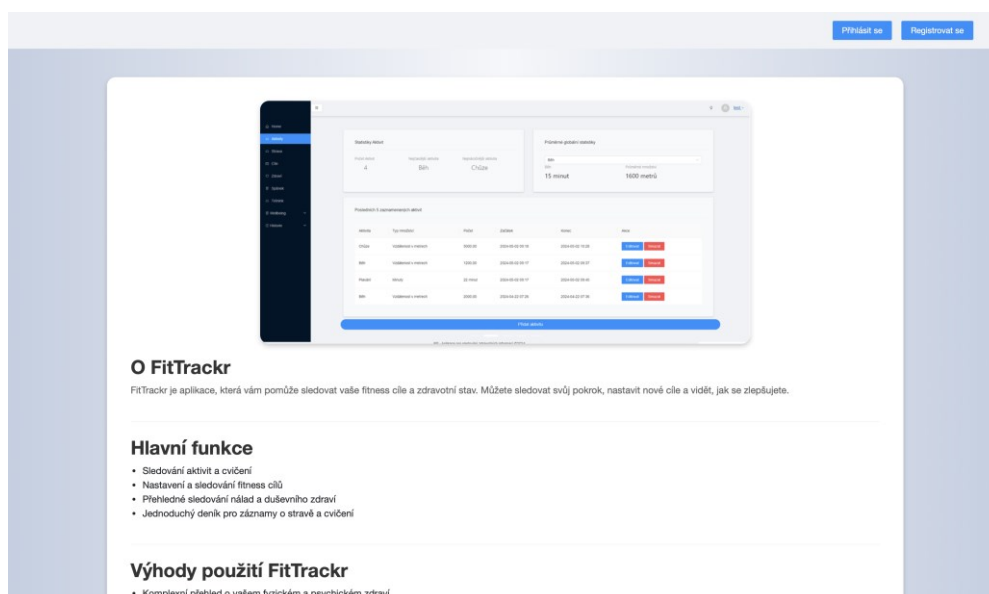
Součástí aplikace je jednoduchá úvodní stránka, která slouží pro stručný popis aplikace, její funkcionality a její účel. Po navštívení je uživatel přivítán jednoduchým rozložením, které se skládá z hlavičky a obsahu stránky.

V hlavičce můžeme nalézt tlačítka pro přesměrování na přihlašovací či registrační formuláře.

Po pokračování na obsah stránky zde nalezneme fotky z aplikace za účelem seznámení uživatele s podobou aplikace. Dále jsou zde jednoduchými body popsány funkce a účely, pro koho je aplikace určená a co od ní může uživatel očekávat.

V poslední části se nachází nejčastější možné dotazy společně s odpověďmi.

Celkově slouží stránka především jako pomyslné uvítání a seznámení předtím, než se uživatel rozhodne pro vytvoření účtu a používání samotné aplikace.



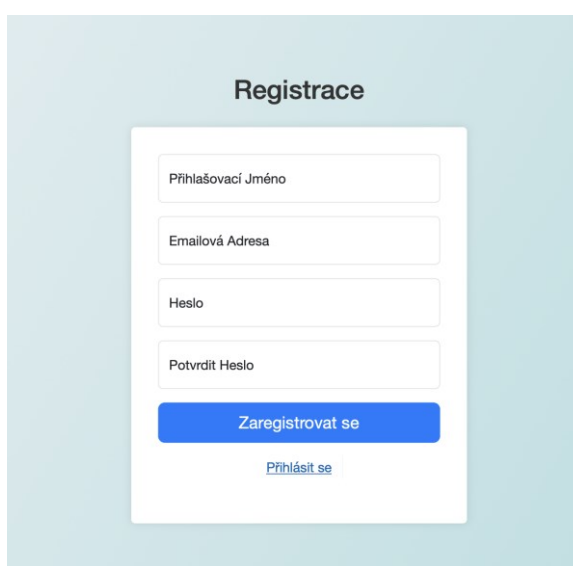
Obrázek 11. Úvodní stránka

8.1.1 Registrace

Po přechodu na registrační formulář v hlavičce domovské stránky můžeme objevit jednoduchý formulář pro vytvoření uživatelského účtu, který se skládá z přihlašovacího jména, emailové adresy a hesla.

Po kliknutí na tlačítko pro vytvoření účtu je formulář validován a v případě, že nedošlo k chybě, je uživatel zaregistrován a přesměrován na stránku pro zadání doplňujících údajů.

Stejně tak je zde umožněno rychlé přesměrování na přihlašovací formulář v případě, že má uživatel již účet vytvořen.



The image shows a registration form titled "Registrace" centered on a light blue background. The form is contained within a white rectangular box and includes the following elements from top to bottom: a text input field labeled "Přihlašovací Jméno", a text input field labeled "Emailová Adresa", a text input field labeled "Heslo", a text input field labeled "Potvrdit Heslo", a prominent blue button labeled "Zaregistrovat se", and a smaller blue link labeled "Přihlásit se" located below the main button.

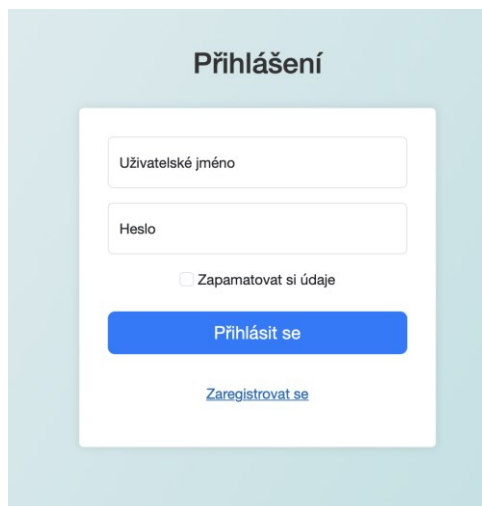
Obrázek 12. Registrační formulář

8.1.2 Přihlášení

Po přesměrování se na přihlašovací formulář, ať už skrze odkaz v hlavičce domovské stránky, či odkazem z registrace, je uživateli zobrazen formulář pro vyplnění uživatelského jména a hesla. Po vyplnění opět, stejně jako tomu je u registrace, dojde k validaci údajů a následně k přihlášení, jsou-li data validní.

Je zde i možnost pro zapamatování si uživatele, díky které není potřeba při každé návštěvě webové aplikace opakovaně vyplňovat údaje.

V neposlední řadě se zde nachází přesměrování na formulář pro vytvoření účtu.



Přihlášení

Uživatelské jméno

Heslo

Zapamatovat si údaje

Přihlásit se

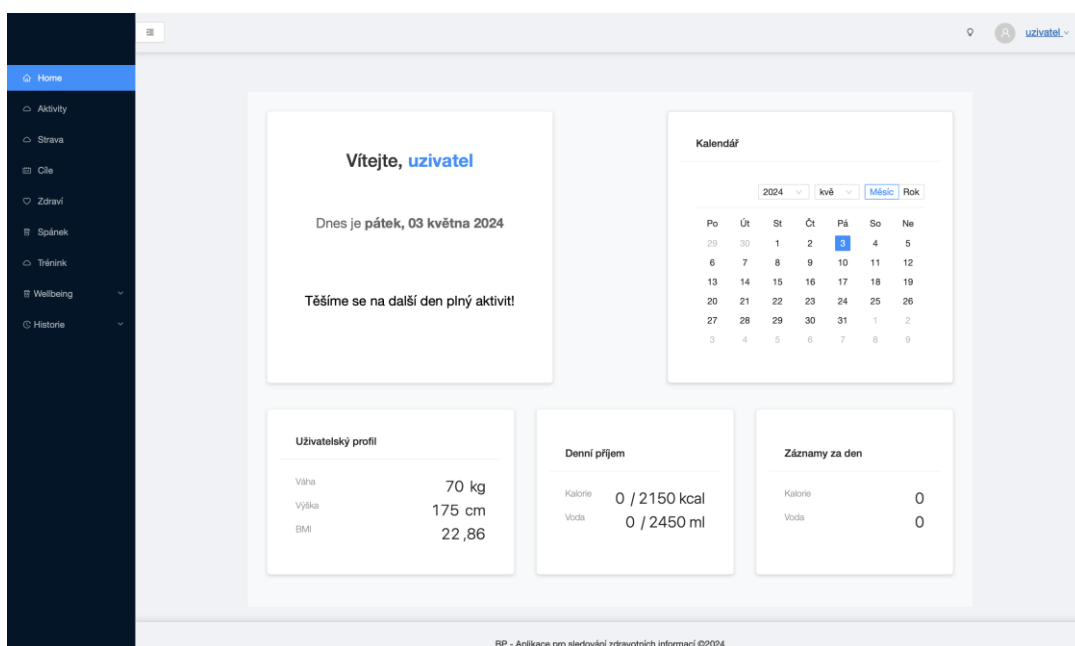
[Zaregistrovat se](#)

Obrázek 13. Přihlášení

8.2 Domovská stránka aplikace

Po přesměrování se na samotnou aplikaci je uživatel uvítán jednoduchým domovským menu.

Jako první se zde vyskytuje jednoduché přivítání uživatele s datem. Dále se zde nachází obecné informace jako je kalendář, váha, výška a jeho BMI, společně s hlídáním denního kalorického příjmu a dodržování pitného režimu, společně s počtem záznamů za den.



Vítejte, uživatel

Dnes je pátek, 03 května 2024

Těšíme se na další den plný aktivit!

Kalendář

2024 kvě Měsíc Rok

Po	Út	St	Čt	Pá	So	Ne
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Uživatelský profil

Váha	70 kg
Výška	175 cm
BMI	22,86

Denní příjem

Kalorie	0 / 2150 kcal
Voda	0 / 2450 ml

Záznamy za den

Kalorie	0
Voda	0

BP - Aplikace pro sledování zdravotních informací ©2024

Obrázek 14. Domovská stránka aplikace

8.3 Účet

Popis stránek týkající se účtu lze rozdělit do 3 samostatných částí, které popisují jak osobní účet, tak sociální aspekt s cizími uživateli, kteří aplikaci taktéž používají.

Části jsou rozděleny následovně:

1. Osobní profil.
2. Sociální stránka.
3. Cizí profil.

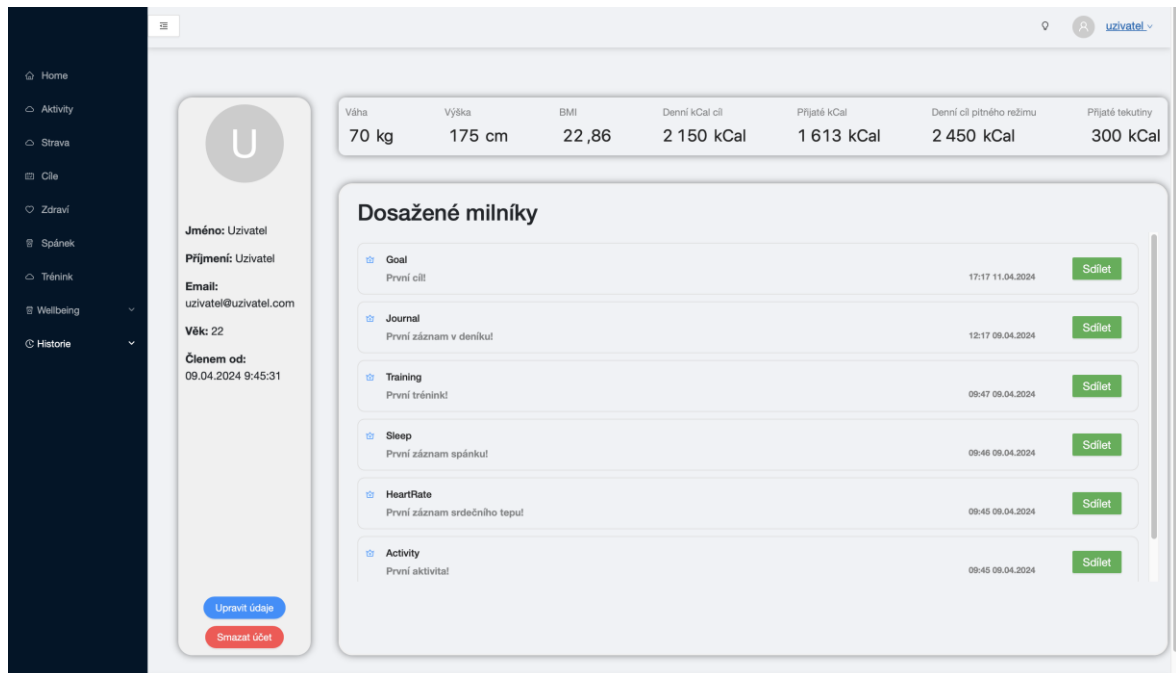
8.3.1 Osobní profil

Při návštěvě uživatelského profilu se za pomoci přesměrování v hlavičce stránky zobrazí osobní přehled celého účtu.

Tento přehled se skládá z několika částí. Jako první si lze v levém sloupci povšimnout osobních údajů, mezi které patří křestní jméno a příjmení, emailová adresa, či věk a datum založení účtu. Spolu s těmito údaji je zde umožněno otevření vysouvacího okna, ve kterém lze příslušné informace změnit. Stejně tak je zde možnost pro kompletní smazání účtu, při kterém dojde k vymazání všech uložených údajů daného uživatele. Tento krok je nevratný.

Ve vertikálním přehledu lze najít informace týkající se fyzického stavu uživatele společně s cíli pro dodržení denního kalorického příjmu a pitného režimu.

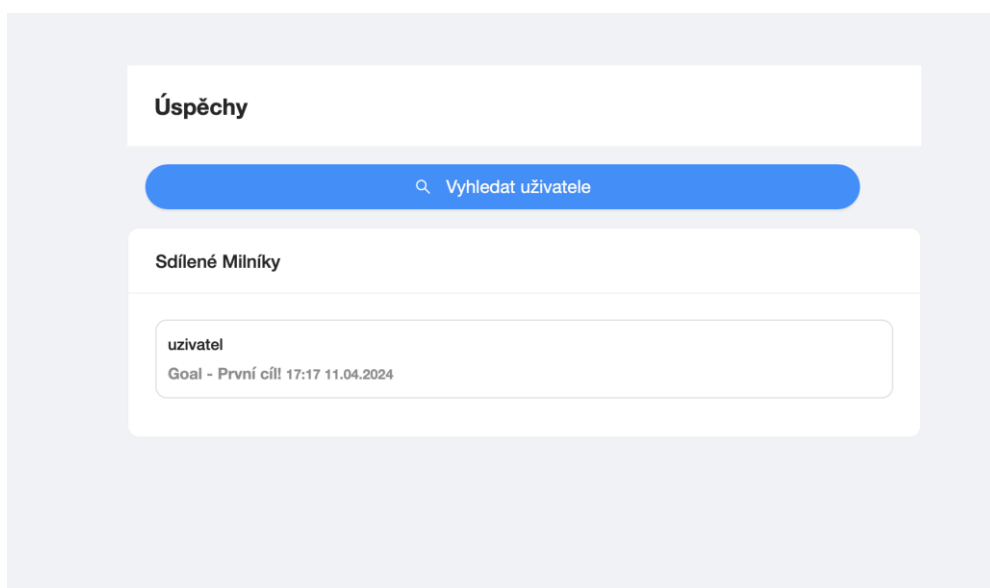
V poslední řadě zde lze najít všechny dosažené milníky, které jsou založeny na počtu vytvořených záznamů. Čím více záznamů v dané kategorii uživatel přidá, tím více milníku mu bude připsáno a zobrazeno na účtu. Tato funkcionality funguje za účelem gamifikace celé aplikace společně s vytvořením jednoduchého sociálního aspektu skrze sdílení milníků.



Obrázek 15. Osobní profil uživatele

8.3.2 Sociální stránka

Pro zobrazení sdílených milníků od uživatelů, slouží jednoduchá sociální stránka. Na té se zobrazuje komponenta pro vyhledávání a sledování uživatele a seznam sdílených milníků. Zde se zobrazí pouze ty, které uživatel sdílel sám od sebe, či ty milníky, které sdílel uživatel, který je na daném účtu sledován.



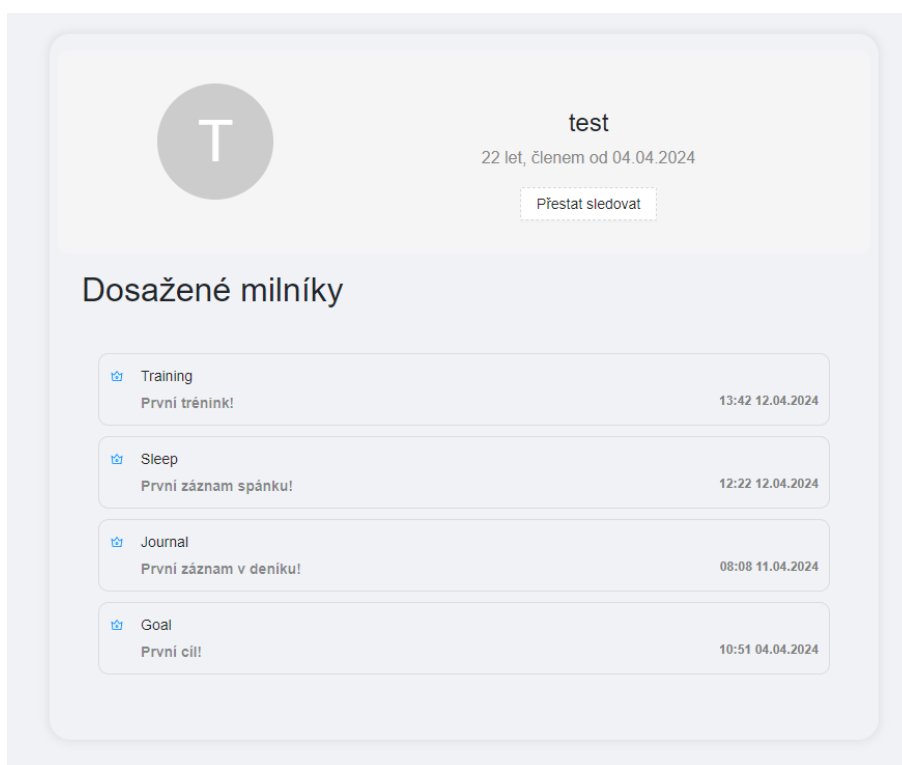
Obrázek 16. Sdílené milníky

8.3.3 Cizí profil

Oproti osobnímu profilu uživatele je cizí profil mnohonásobně zjednodušený. Důvodem je zde samozřejmě soukromí.

Při vyhledání a rozkliknutí profilu uživatele můžeme nalézt uživatelské jméno, datum vytvoření účtu a seznam sdílených milníků. Tyto milníky jsou opět zobrazeny pouze v případě, že je uživatel dobrovolně sdílel.

Po kliknutí na sledování uživatele je vytvořen záznam v databázi pro zpracování sledování a při navrácení se na stránku se všemi sdílenými milníky, budou milníky nového uživatele vypisovány zde.



Obrázek 17. Cizí profil

8.3.4 Import a export dat

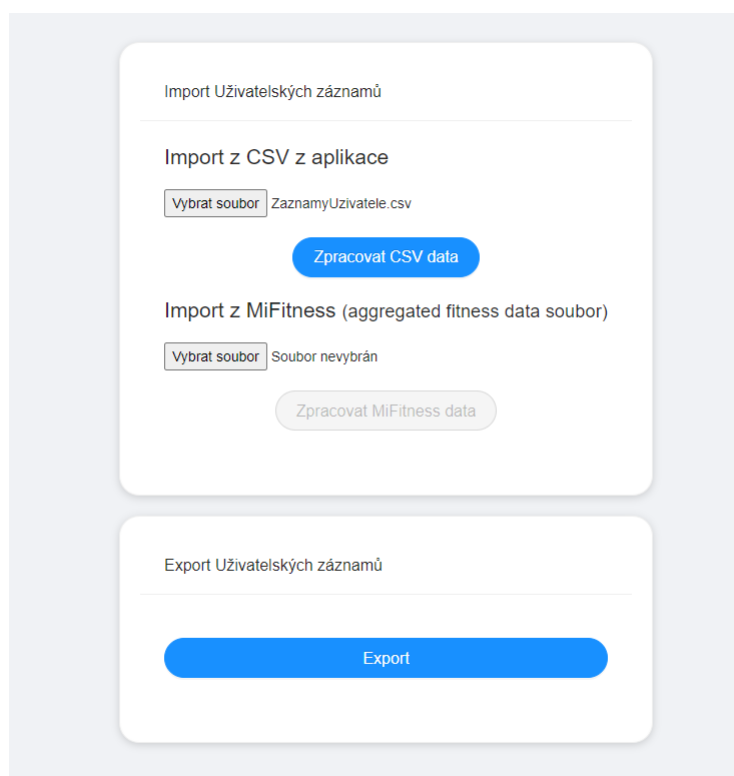
Součástí aplikace je taktéž funkce pro importování záznamů, či naopak export záznamů do souboru CSV.

Po přesměrování se na stránku pro tyto účely za pomoci menu u uživatelského jména v hlavě stránky, je uživateli zobrazeno jednoduché rozložení, které lze rozdělit na dvě hlavní části.

V první části můžeme nalézt funkce pro importování dat do aplikace. Zde jsou dostupné dvě různé možnosti, tou první je import z CSV souboru aplikace. Tím je myšlen soubor, který byl z této aplikace taktéž exportován. Tato funkce může být použita například v případě, že si bude chtít uživatel založit nový účet ale ponechat si již uložené záznamy.

Druhou možností je import z CSV souboru, který byl poskytnut z chytrých hodinek MiBand, přesněji z aplikace MiFitness od společnosti Xiaomi. Vzhledem ke struktuře CSV souboru v porovnání se strukturou tabulek v naší aplikaci je však možno importovat pouze údaje pro srdeční tep a spánek. Zde je místo pro budoucí rozšíření, či přizpůsobení funkce za účelem importu více dat.

V druhé půlce stránky můžeme nalézt export uživatelských dat. Při této operaci se načtou všechna uložená data uživatele do CSV souboru, který se následně stáhne na klientské zařízení.



Obrázek 18. Stránka pro Import a export

8.4 Aktivity, trénink a cíle

Části aplikace lze seskupit do několika větších skupin na základě propojení. Jako první skupinou jsou zde aktivity, tréninky a cíle, které při propojení skrze vytváření či úpravu

záznamů, umožňují sledovat například stav hubnutí v podobě spálených kilogramů ku požadovaným, či jiné dostupné cíle.

8.4.1 Aktivity

První část této skupiny se týká aktivit. Po přechodu na tuto stránku si lze povšimnout, že stránka je rozdělena na dvě poloviny. V horní části můžeme najít statistiky týkající se záznamů uživatele. Jmenovitě se zde ukazuje počet aktivit (záznamů), nejčastěji zapisovaná aktivita spolu s nejnáročnější aktivitou. Ta je vyhodnocena na základě stráveného času.

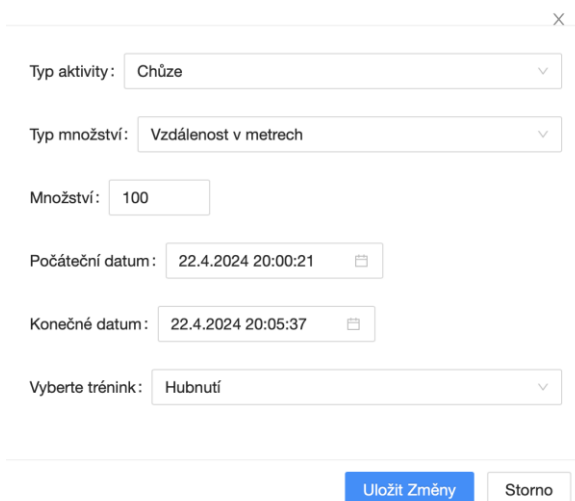
Hned vedle lze nalézt průměrné globální statistiky každé aktivity. Přesněji se po výběru ukáže průměrná doba a množství, podle aktivity se může jednat o metry, počty opakování či délku vykonávání aktivity.

Druhá polovina se skládá z výpisu posledních pěti aktivit a přidáním nového záznamu. Vypsané aktivity lze editovat či smazat.

Aktivita	Typ množství	Počet	Začátek	Konec	Akce
Jóga	Minuty	12 minut	2024-04-22 20:02	2024-04-22 20:15	Editovat Smazat
Běh	Vzdálenost v metrech	100,00	2024-04-22 20:00	2024-04-22 20:05	Editovat Smazat
Posilování - střední zátěž	Minuty	1 hodina 9 minut	2024-04-22 19:59	2024-04-22 21:09	Editovat Smazat
Plavání	Vzdálenost v metrech	1200,00	2024-04-11 17:24	2024-04-11 17:34	Editovat Smazat
Plavání	Minuty	15 minut	2024-04-11 17:24	2024-04-11 17:40	Editovat Smazat

Obrázek 19. Stránka aktivit

Při kliknutí na přidání aktivity je uživateli zobrazeno modální okno, ve kterém si vybere typ aktivity, typ množství, celkové množství, trvání a v poslední řadě propojení s tréninkem. Toto propojení je však dobrovolné a může být provedeno i zpětně skrze editaci.



Typ aktivity: Chůze

Typ množství: Vzdálenost v metrech

Množství: 100

Počáteční datum: 22.4.2024 20:00:21

Konečné datum: 22.4.2024 20:05:37

Vyberte trénink: Hubnutí

Uložit Změny Storno

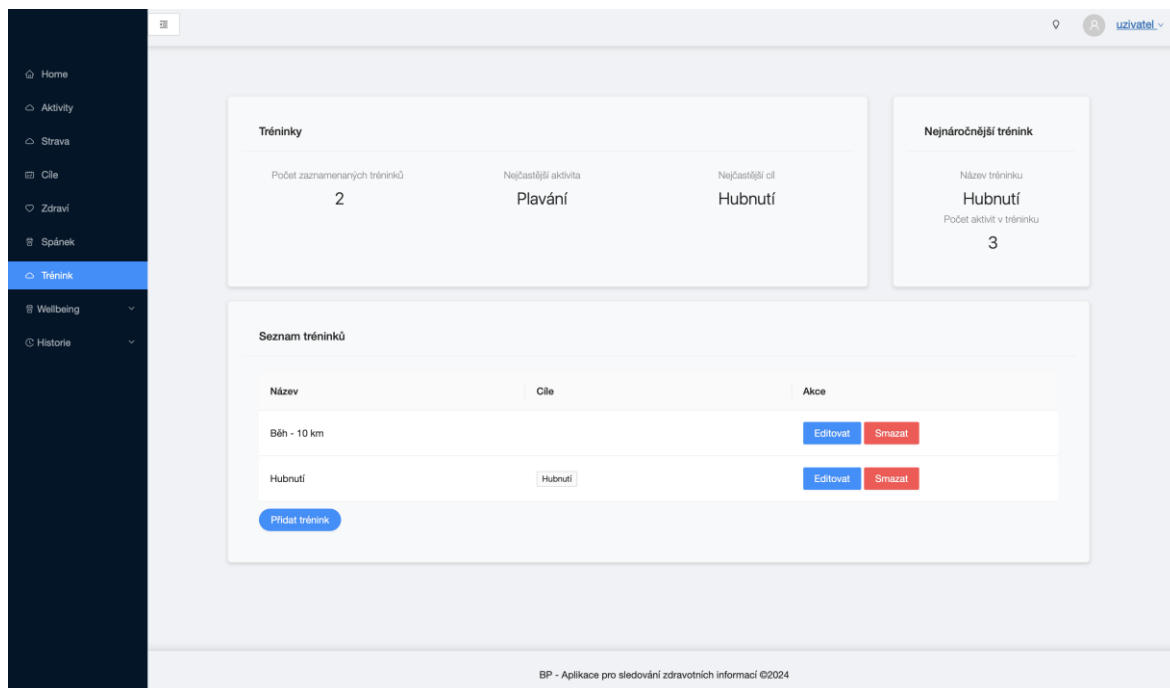
Obrázek 20. Modální formulář
pro přidání aktivity

8.4.2 Trénink

Obdobně jako u aktivit i zde se skládá stránka pro tréninky ze dvou částí. První část obsahuje statistiky založené na záznamech uživatele a druhá obsahuje samotné záznamy.

V statistice můžeme opět nalézt počet celkových záznamů a nejčastěji zaznamenanou aktivitu s cílem. Poslední dvě jmenované jsou založeny na propojení aktivit s cíli skrze tréninky. Avšak oproti aktivitám se zde nenachází globální statistika. Jako poslední části statistik je pouze výpis nejnáročnějšího tréninku, který se zakládá na počtu přiřazených aktivit.

V druhé části se opět vyskytuje posledních 5 záznamů uživatele, které lze editovat či mazat.



Obrázek 21. Stránka s tréninky

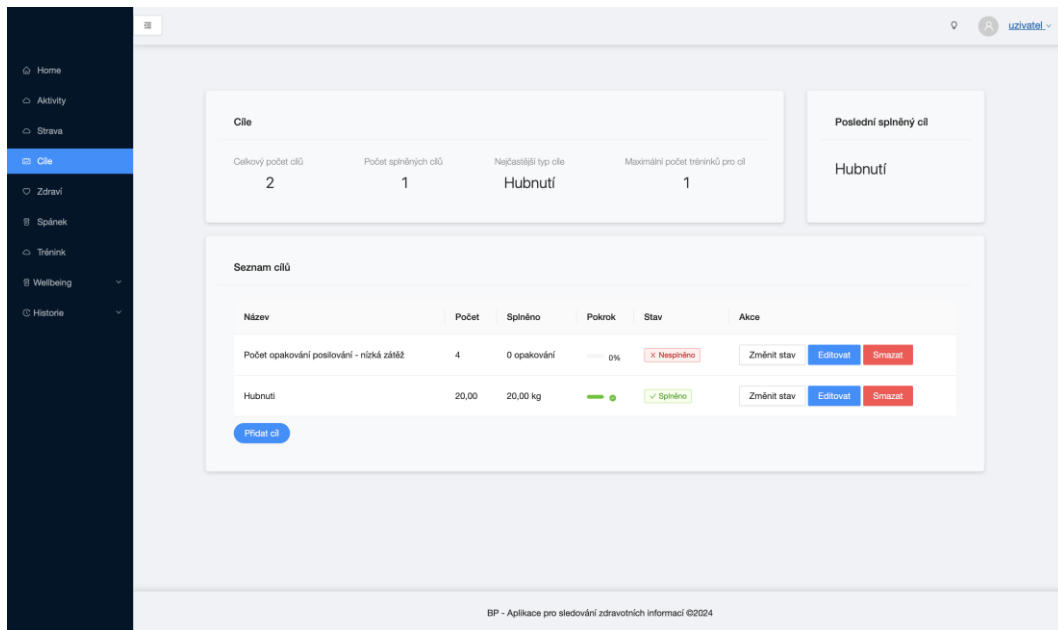
Přidání záznamu tréninku je oproti aktivitě mnohonásobně jednodušší, jelikož se skládá pouze z vlastního názvu tréninku a cíle, se kterým má být propojen.

8.4.3 Cíle

Ve statistické části stránky pro cíle můžeme opět nalézt celkový počet cílů, počet dokončených cílů, s nejčastějším typem a maximálním počtem tréninků, které jsou připojeny k jednomu cíli. Navíc je zde také zobrazován poslední dokončený cíl.

Druhá část, skládající se ze záznamů, je zde již od pohledu o mnoho komplexnější, než tomu bylo doposud. Mimo jednoduché vlastnosti záznamu, jako je název, je zde počet, vypočítaný postup k cílovému počtu, indikátoru postupu cíle a stav, zda je cíl splněný či nikoliv. Vzhledem k tomuto je zde přidáno i tlačítko na manuální změnu stavu.

V případě, že stav přejde ze splněného do nesplněného, je postup vypočítán na základě připojených tréninků s aktivitami.



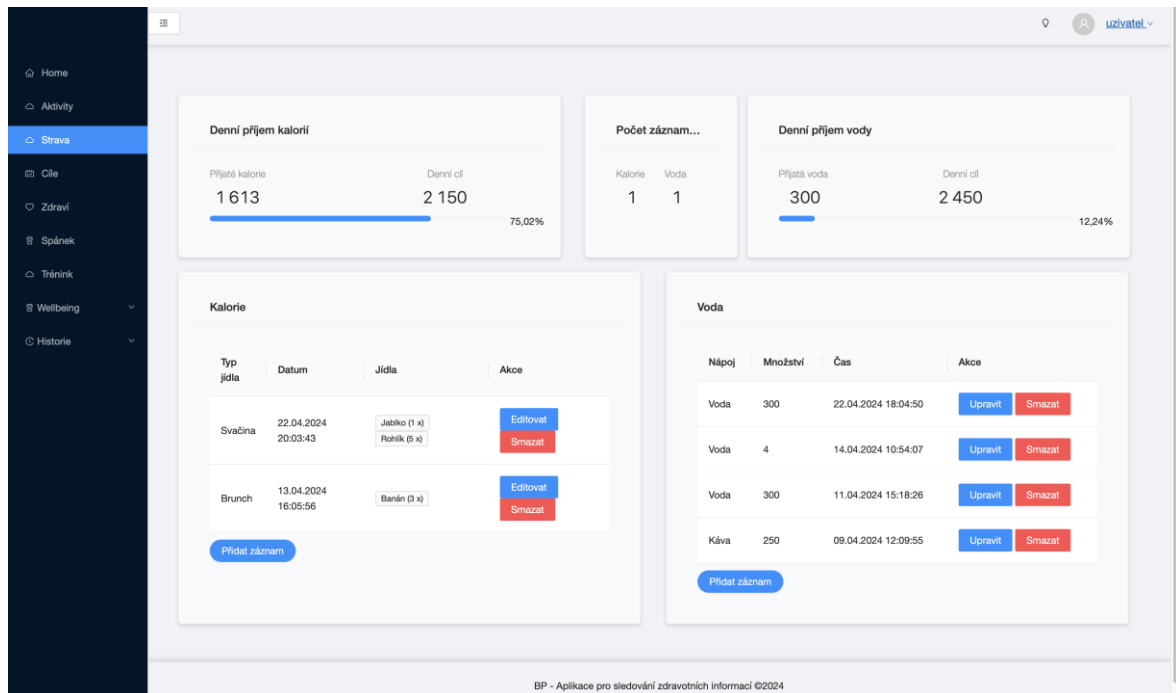
Obrázek 22. Stránka pro cíle

Přidávání záznamů se zde skládá z výběru cíle, který může mít případně svůj vlastní název, a cílového počtu.

8.5 Strava

Další rozsáhlá část, týkající se kalorického příjmu a pitného režimu, byla zpracována jako jedna komplexní stránka, na které se tyto dva účely mísí.

Po zobrazení obsahu si můžeme povšimnout indikátoru postupu jak pro příjem kalorií, tak pro pitný režim, kdy jsou oba z těchto ukazatelů, založeny na záznamech přidaných za jeden den. Pro přehled je zde zobrazen i denní čítač jak kalorický záznamů, tak záznamů pro tekutiny.



Obrázek 23. Sledování stravy

Při přidání záznamů, ať už se jedná o kalorie či nápoje, se opět zobrazí modální okno, kde si mimo množství a datumu lze vybrat i o jaký typ jídla se jednalo (snídaně, oběd, ...), případně o to, jaký nápoj byl zkonsumován.

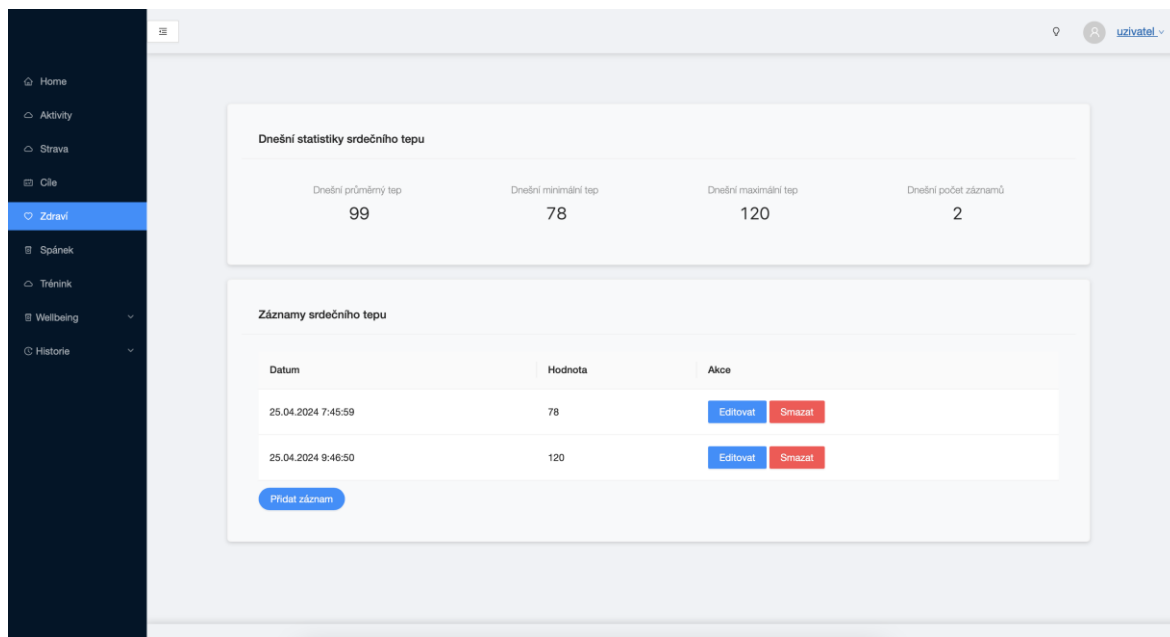
Oproti vodě má kalorický záznam však rozšířený formulář o možnost přidání až 3 jídel do jednoho záznamu.

8.6 Zdraví

Zdraví neobsahuje tolik obsahu, jako tomu bylo u spánku, nicméně i zde se nachází jednoduchý přehled ve formě statistik, které jsou vázány k denním záznamům.

Uživateli se zde zobrazí průměrný srdeční tep, minimální a maximální společně s počtem záznamů, přidanych za jeden den.

Záznamová část se opět skládá z posledních 5 záznamů s možností jejich úprav a mazání.



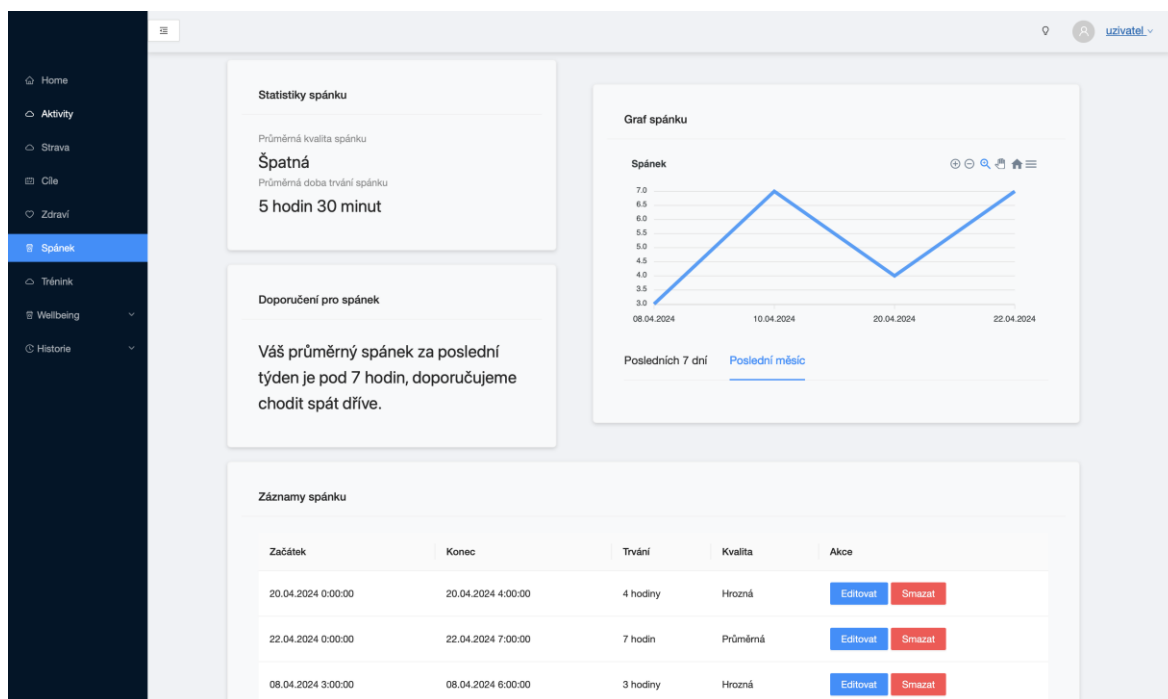
Obrázek 24. Zaznamenávání srdečního tepu

Jak si lze povšimnout ze struktury záznamů, i formulář je zde velice jednoduchý, jelikož je složen pouze z data přidání a hodnoty srdečního tepu.

8.7 Spánek

Obsah stránky pro spánek je opět rozdílný oproti ostatním přehledům. Zde se již statistiky neskládají z počtů, ale z průměrné délky a kvality spánku. Na základě délky je dále vypsáno doporučení, týkající se toho, zda by měl uživatel chodit spát dříve, nespát zbytečně dlouhou dobu, nebo je mu sděleno, že jeho průměrná délka spánku je v doporučených mezích. Toto rozmezí je od sedmi to devíti hodin trvání spánku.

Na základě záznamů, je zde zobrazen i spojnicový graf, který umožňuje grafický přehled záznamů za poslední týden či měsíc.



Obrázek 25. Přehled spánku

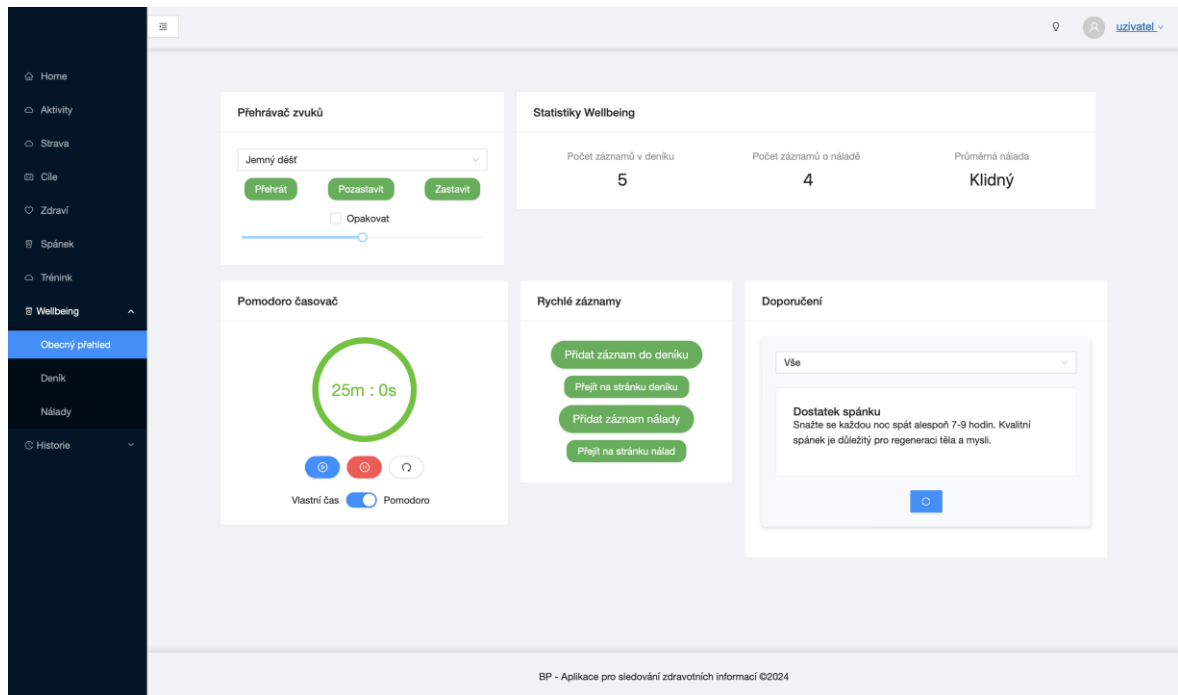
Zde se přidání záznamu skládá z údajů pro začátek a konec spánku a jeho kvality.

8.8 Duševní zdraví

Část aplikace zaměřená na duševní zdraví se skládá z jedné hlavní stránky, která slouží jako pomyslná křižovatka všech komponentů, tykajících se tohoto zaměření. Na samotné domovské stránce se nachází několik dostupných funkcionalit, jmenovitě se jedná o přehrávač, výpis náhodných doporučení, tlačítka pro zobrazení formuláře pro záznam deníku či nálady, společně s tlačítkem pro okamžité přesměrování na tyto stránky, kde si může uživatel procházet všechny vlastní záznamy. V poslední řadě se zde také nachází jednoduchý časovač, který může pomoci uživateli k soustředění. Zde jsou na výběr dvě varianty, pomodoro časovač, při kterém se odpočítává 25 minut, následuje 5 minut přestávky a tento cyklus se opakuje, nebo si může uživatel přepnout spínačem nastavení a vložit svůj vlastní čas.

Formuláře pro náladu a záznam do deníku jsou ve formě vyskakovacího okna, dle zaměření obsahuje příslušné vstupy pro kompletní vyplnění záznamu. U zápisu deníku se jedná o jednoduchý text, u nálady se jedná o datum společně s poznámkou a výběru typu nálady.

Tento vstup následně umožňuje uživateli sledovat jeho průměrnou náladu za poslední týden skrze komponentu, která je také součástí této stránky. Samotná hodnota je vychází z průměru zaznamenaných hodnot za poslední týden.



Obrázek 26. Domovská stránka pro duševní zdraví

8.8.1 Deník

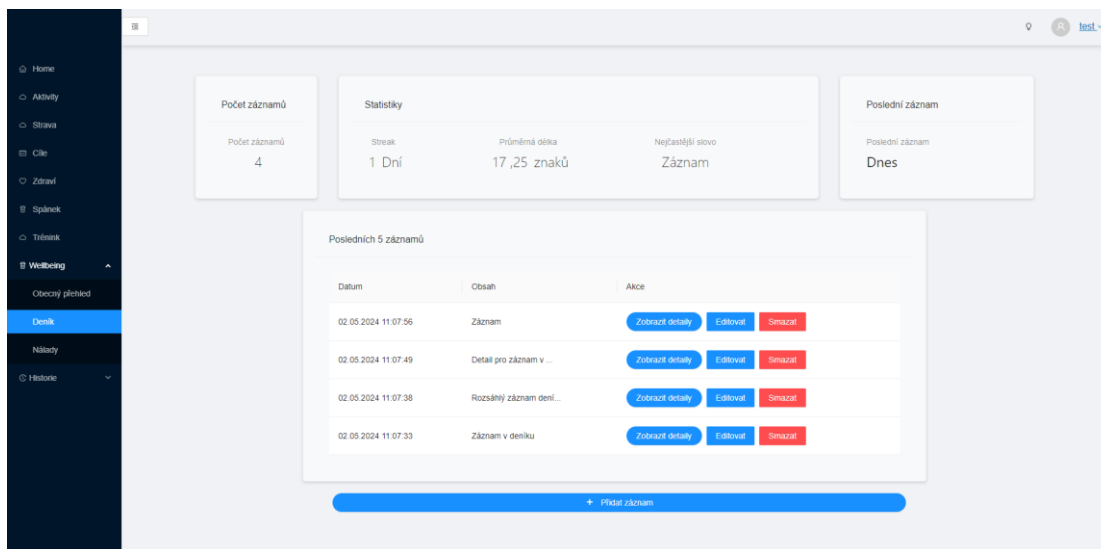
Po přechodu na stránku týkající se obsahu deníku je uživatel přivítán přehledem ve formě statistik společně s posledními pěti záznamy za účelem kompaktnějšího vzhledu stránky.

V horizontálním panelu si můžeme povšimnout čítače, který sděluje celkový počet záznamů do deníku. Na střední kartě se nacházejí průměrné statistiky záznamů, ať už se jedná o počet dní, kdy byly přidávány záznamy po sobě, průměrnou délku všech záznamů či nejčastěji se vyskytující se slovo.

Statistika pro nejčastější slovo nerozlišuje velká či malá písmena.

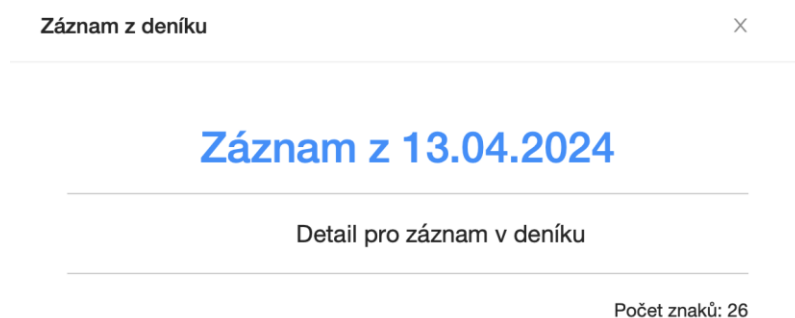
Jako poslední je v statistické části údaj o tom, kdy byl poslední záznam přidán.

Ve výpisu jsou opět přístupné funkce pro editaci či smazání záznamů, avšak zde si můžeme zobrazit i detail daného záznamu za pomoci vysouvacího okna.



Obrázek 27. Obsah deníku

Po stisknutí tlačítka pro zobrazení detailu je vysunuto šuplíkové okno a uživatel si může prohlédnout obsah celého záznamu společně s přehledem pro počet znaků.



Obrázek 28. Detail záznamu v deníku

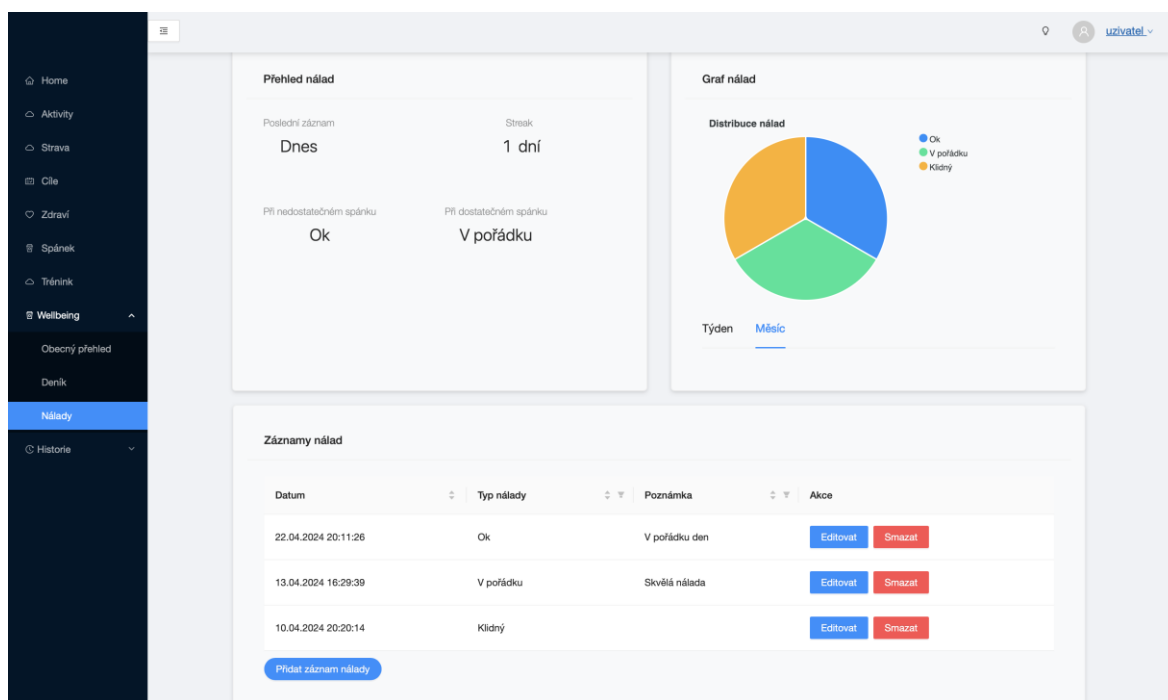
8.8.2 Nálada

Na stránce, týkající se nálady, můžeme objevit rozsáhlé statistiky, s jejichž pomocí může uživatel přehledně sledovat všechny informace týkající se jeho záznamů. V první části si může povšimnout jednoduchých informací, týkajících se posledního záznamu, či naopak propojení nálad a spánku. Zde se zobrazují hodnoty průměrné nálady v případě, že uživatel spal méně než 7 hodin, či odpočíval dostatečně dlouhou dobu.

Napravo od tohoto přehledu můžeme nalézt koláčový graf, na kterém jsou zobrazeny nálady z posledního týdne či měsíce. Tyto nálady vychází ze všech dostupných záznamu, nejen z těch, které jsou zde vypsány.

Na této stránce se opět vyskytuje pouze posledních 5 záznamů uživatele. Je vhodné si povšimnout že záznamy se skládají nejen z data, ale také z typu nálad a dobrovolné poznámky.

Funkcionalita správy záznamů je zde stejná jako u jiných. Uživateli je zde umožněno záznam přidat, editovat, či smazat.



Obrázek 29. Přehled nálad

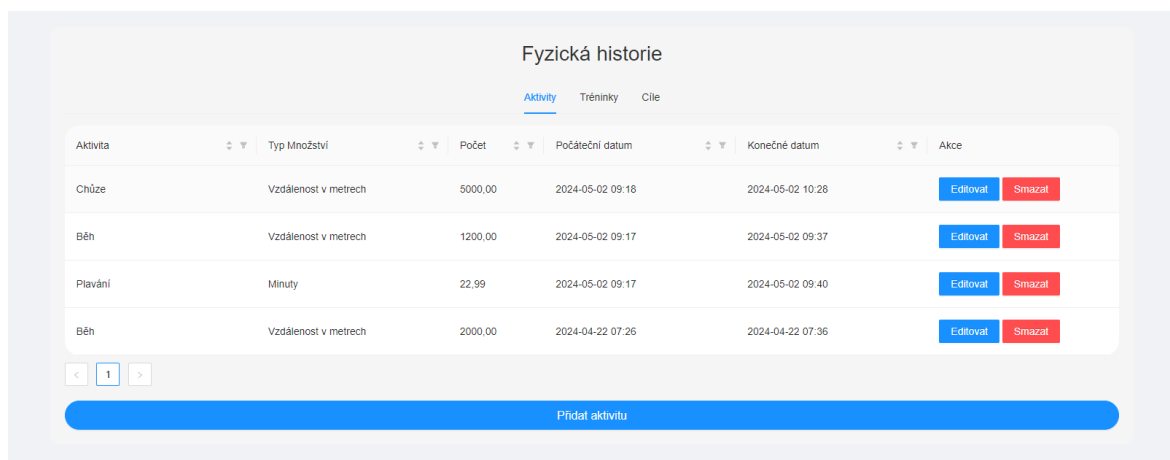
8.9 Historie

Aplikace taktéž obsahuje historii, jejímž účelem je zobrazit uživatelům všechny dostupné záznamy, které kdy přidali. Oproti záznamům na jednotlivých stránkách, které byly již výše zmíněny, se zde nachází všechny záznamy v databázi, které kdy byly přidány přihlášeným uživatelem.

Lze si všimnout že mezi kategoriemi záznamů se lze pohybovat za pomoci záložek, za účelem jednoduchého a přehledného navigačního menu na středu stránky.

Po vybrání požadované kategorie jsou zobrazeny záznamy, které jsou rozděleny do stránek po 7. Toto dělení je čistě z estetického důvodu, za účelem zpříjemnění uživatelského zážitku, jelikož příliš mnoho informací na stránce může uživateli ve většině případů znepříjemnit používání.

Na obrázku je uvedena pouze stránka pro fyzickou historii, avšak princip fungování je stejný jak u zdravotní, tak u duševní historie.



Fyzická historie

[Aktivity](#) [Tréninky](#) [Cíle](#)

Aktivita	Typ	Množství	Počet	Počáteční datum	Konečné datum	Akce
Chůze	Vzdálenost v metrech	5000,00		2024-05-02 09:18	2024-05-02 10:28	Editovat Smazat
Běh	Vzdálenost v metrech	1200,00		2024-05-02 09:17	2024-05-02 09:37	Editovat Smazat
Plavání	Minuty	22,99		2024-05-02 09:17	2024-05-02 09:40	Editovat Smazat
Běh	Vzdálenost v metrech	2000,00		2024-04-22 07:26	2024-04-22 07:36	Editovat Smazat

< 1 >

[Přidat aktivitu](#)

Obrázek 30. Struktura historie

9 TESTOVÁNÍ APLIKACE

Testování je nedílnou součástí vývoje každé aplikace pro ověření toho, že funkce pracují tak jak mají a uživatel na konci dostane zamýšlený výsledek.

Lze jej rozdělit jak na manuální, tak na automatizované testy, kdy se pomocí skriptů provádí různé scénáře za účelem buď nalezení chyby, nebo naopak ověření toho, že systém byl správně naprogramován a nedochází zde k žádnému problému.

Vzhledem k způsobu vývoje aplikace, se zde vyskytovalo především manuální testování, kdy po každé nově implementované metodě, službě, funkci bylo otestováno, zda vrací požadované výsledky. Tento způsob byl vybrán především z toho důvodu, že byl přívětivější možností.

I přesto byly provedeny jednoduché automatizované testy, které ověřují správnost CRUD operací s daty v aktivitách, tréninzích a cílech. Všechny tyto testy fungují správně a funkcionality přidávání, úpravy a mazání záznamů v těchto kategoriích je implementována správně. To však nevyklučuje případné výjimky, jelikož šance na ošetření každé existující možnosti je minimální.

10 HOSTOVÁNÍ APLIKACE

Pro zpřístupnění aplikace je nutností ji mít uloženou na serveru, který zajišťuje její plné fungování a umožňuje uživatelům přístup na stránku, aniž by měli přístup ke zdrojovému kódu, či by se nacházeli v lokální síti, na které je aplikace vyvíjena.

Pro tento účel byla vybrána služba Microsoft Azure, především z toho důvodu, že plně podporuje hostování aplikací vyvinutých s využitím frameworku ASP.NET, ať už se jedná o aplikace vytvořené za pomoci MVC nebo Blazoru. Microsoft Azure i ASP.NET jsou vyvíjené stejnou společností a díky této vlastnosti také poskytují plnou kompatibilitu a širokou podporu společně s vysokou spolehlivostí.

K samotnému hostování byla využita kombinace prostředků označovaných jako webová aplikace a Azure SQL databáze pro MySQL. První jmenovaný prostředek slouží pro rezervaci domény a poskytuje prostor pro hostování aplikace na základě dodaného kódu. Pro plnou funkčnost je však nutnost databáze a k tomu slouží druhý jmenovaný prostředek. Ten byl vybrán především z toho důvodu, že při vývoji byla použita lokální MySQL databáze.

V případě, že by se při publikaci vybral jiný databázový systém, by bylo nutností projít všechny dostupné migrace a přizpůsobit je syntaxi nově vybraného systému. Stejně tak by bylo nutné upravit všechny atributy tabulek společně s jejich vlastnostmi.

Microsoft Azure nabízí plán služby pro hostování, který je dostupný zdarma, avšak na úkor výkonu v porovnání s placenými plány. U databáze tato možnost není dostupná, avšak studentům je poskytován kredit v hodnotě dvou set dolarů, které lze využít dle vlastního uvážení.

Samotnou aplikaci lze nalézt na doméně:

<https://fittrackr.azurewebsites.net/>

Využitím těchto prostředků je docíleno stabilního a spolehlivého prostředí pro hostování vyvinuté aplikace.

11 ZHODNOCENÍ A VYLEPŠENÍ

Vytvořená aplikace působí jako dostačující nástroj pro lidi, kteří žijí aktivně, nebo pro ty, kteří mají zájem o to sledovat přehledně svůj fyzický a mentální stav. Poskytuje mnoho způsobů, jak tyto informace a zdravotní údaje sledovat a je jim za pomoci analýzy záznamů vrácen jednoduchý přehled, který se může týkat čistě z průměrných informací, či přímo doporučení, jako je tomu například u spánku.

I přesto je zde však místo na vylepšení.

11.1 Funkce

V rámci funkcí by bylo vhodné rozšířit nabídku o nové možnosti, mezi které lze zařadit například snímání a sledování tělesné teploty, saturace kyslíku či jiných záznamů, které by uživatel měl zájem sledovat. Tato nabídka je však úzce závislá na vývoji technologií chytrých hodinek, která se neustále vyvíjí a rozšiřuje své, už tak široké, spektrum dostupných měření.

11.2 Adaptační vylepšení moderním technologiím

Vzhledem k tomu, že chytré hodinky jsou nejčastěji propojeny s mobilním zařízením za využití bezdrátových sítí, další z vhodných vylepšení by byl vývoj samostatné mobilní aplikace, která by ve výsledku mnohonásobně usnadnila zaznamenávání a používání aplikace. V případě použití nadsázky by se dalo říct, že s mobilní aplikací by se mohlo používat aplikaci přímo zaznamenávat, jelikož záznamy z hodinek by mohly být přímo sdíleny skrze mobilní zařízení a rovnou ukládány do databáze, bez nutnosti manuálního zasahování uživatele.

Jako další přizpůsobení nejmodernějším technologiím by bylo vhodné zvážit využití umělé inteligence, ať už pro jednoduchou analýzu sentimentu z deníku a propojení se záznamem nálady, či přímo pro generování doporučených aktivit za účelem dosažení daného cíle.

11.3 Frontend

Co se týče samotné aplikace, i zde je stále místo na vylepšení, ať už se jedná o lépe zpracovanou responzivitu, vzhledem k tomu že využitá knihovna komponent se zaměřuje především na desktopové obrazovky, či přímo o využití jiného, populárnějšího frameworku. Zde může dojít k nahrazení Blazoru například populárním JavaScriptovým frameworkem React

či Vue.js, jelikož Blazor je stále vcelku mladý framework a nové funkce nejsou nejlépe vyladěny.

Mimo frontendovou otázku aplikace by se mohla také rozšířit poskytovaná zpětná vazba ve formě statistik, či přímo exportování do přehlednějšího doporučení, které lze určovat pomocí vymezeného období.

Vhodné by také bylo zmínit vylepšení, týkající se výkonnějšího plánu pro hostování webové aplikace, či přímo zajištění vlastního serveru s vlastním výpočetním výkonem, který je určen pouze pro tyto účely. Zde je to však otázka spíše finanční než čistě technologická.

Většina zmíněných vylepšení či změn by bylo však vhodnější zpracovávat ve vícero lidech, jelikož na jednoho vývojáře, s ohledem na rychlost vývoje technologií, může být tvorba nových funkcí příliš pomalá.

ZÁVĚR

Tato bakalářská práce se zabývá vývojem aplikace pro sledování zdravotních informací.

V teoretické části práce jsou popsány všechny použité technologie, včetně jejich historie a funkcí, které poskytují, a možnosti hostingu webových systémů. Dále je zde zpracována problematika energetického výdeje, jsou zde objasněny jednotky, jmenovitě MET jednotky, které jsou využívány v rámci aplikace. V této části byla provedena rešerše na téma aplikací zaměřených na fyzické a duševní zdraví.

Praktická část práce detailně rozebírá analýzu požadavků aplikace, společně se popisem strukturování projektu při vývoji.

Následně je zde popsána databázová struktura, která je nedílnou součástí samotné aplikace a nebylo by bez ní možné fungovat. Dále jsou zde popsány funkcionality aplikace, ať už se jedná o sledování fyzických aktivit uživatele ve formě seskupení aktivit do tréninků, které je následně možno propojit s cíli pro sledování požadovaného postupu, zaznamenávání kalorického příjmu či dodržování pitného režimu, nebo sledování a analýza nálady na základě uložených záznamů.

Mimo již zmíněné funkce se zde nachází taktéž sledování srdečního tepu či spánku, včetně grafického zobrazení, a možnost přidávání záznamů do osobního deníku.

Při využívání aplikace je uživatel odměňován milníky, které následně lze sdílet s uživateli, kteří jej sledují, či naopak sledovat cizí uživatele a jejich sdílené milníky. Tímto je podpořena gamifikace aplikace, která také přispívá k tomu, aby uživatel aplikaci používal i nadále.

Výsledná aplikace poskytuje uživatelům přívětivé a intuitivní prostředí, které umožňuje sledovat své fyzické a mentální zdraví, s důrazem na bezpečnost dat.

SEZNAM POUŽITÉ LITERATURY

- [1] FREEMAN, Adam, 2022. Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages. 9th ed. Edition. Apress. ISBN 978-1484279564.
- [2] MICROSOFT CORPORATION, INC., b. r. C# docs – get started, tutorials, reference. | Microsoft Learn [online]. [cit. 2023-11-06]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/csharp/>
- [3] MICROSOFT CORPORATION, INC., b. r. ASP.NET documentation | Microsoft Learn [online]. [cit. 2023-11-06]. Dostupné z: <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-7.0>
- [4] J. PRICE, Mark, 2022. C# 11 and .NET 7 – Modern Cross-Platform Development Fundamentals: Start building websites and services with ASP.NET Core 7, Blazor, and EF Core 7, 7th Edition. 7th ed. Edition. Packt Publishing. ISBN 978-1803237800.
- [5] DRESHER, Tamir, Amir ZUKER a Shay FRIEDMAN, 2018. Hands-On Full-Stack Web Development with ASP.NET Core: Learn end-to-end web development with leading frontend frameworks, such as Angular, React, and Vue. Packt Publishing. ISBN 978-1788622882.
- [6] ORACLE CORPORATION, b. r. Co je to databáze? [online]. [cit. 2024-04-10]. Dostupné z: <https://www.oracle.com/cz/database/what-is-database/>
- [7] KUMAR, Mukesh, b. r. Code First vs. Database First vs. Model First Approach. C# Corner [online]. 30.11.2015 [cit. 2024-04-11]. Dostupné z: <https://www.c-sharpcorner.com/blogs/code-first-vs-database-first-vs-model-first-approach1>
- [8] GRECH, Shaun, 2013. MODEL FIRST VS CODE FIRST. Result-Driven Web & Mobile Applications by Incredible Web [online]. [cit. 2024-04-11]. Dostupné z: <https://www.incredible-web.com/blog/model-first-vs-code-first>
- [9] What Is MySQL and How Does It Work, 2024. Hostinger Tutorials [online]. [cit. 2024-04-11]. Dostupné z: <https://www.hostinger.com/tutorials/what-is-mysql>
- [10] ORACLE CORPORATION, b. r. MySQL 8.3 Reference Manual [online]. 2024-04-01 [cit. 2024-04-10]. Dostupné z: <https://dev.mysql.com/doc/refman/8.3/en/>

- [11] MySQL Storage Engine Architecture, 2024. In: ORACLE CORPORATION. MySQL:: MySQL 8.3 Reference Manual [online]. [cit. 2024-04-09]. Dostupné z: <https://dev.mysql.com/doc/refman/8.3/en/images/mysql-architecture.png>
- [12] MICROSOFT CORPORATION, 2023. Přehled technologie Entity Framework Core (EF Core) | Microsoft Learn [online]. [cit. 2024-05-01]. Dostupné z: <https://learn.microsoft.com/cs-cz/ef/core/>
- [13] Co je to Vývojové prostředí | DAMI, b. r. DAMI [online]. [cit. 2024-05-01]. Dostupné z: <https://www.damidev.com/slovník/vyvojove-prostredi>
- [14] What is Visual Studio? All You Need to Know, 2023. Hubspot [online]. 17.11.2023 [cit. 2024-05-01]. Dostupné z: <https://blog.hubspot.com/website/what-is-visual-studio>
- [15] XUnit, b. r. XUnit [online]. [cit. 2024-05-02]. Dostupné z: <https://xunit.net/>
- [16] JANÍK, David, 2021. Co je to webhosting a podle čeho vybírat? Virtuální servery s nejlepší administrací pro VPS | Váš-Hosting [online]. [cit. 2024-05-03]. Dostupné z: <https://www.vas-hosting.cz/blog-co-je-to-webhosting-a-podle-ceho-vybirat>
- [17] Metabolický ekvivalent | NZIP, b. r. NZIP – Národní zdravotnický informační portál [online]. [cit. 2024-05-04]. Dostupné z: <https://www.nzip.cz/rejstrikovy-pojem/5293>
- [18] BUCKEY, A. W., 2019. Personal Data Collection (Privacy in the Digital Age). Core Library. ISBN 978-1532118937.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

C#	C Sharp.
F#	F Sharp.
CLR	Computer Language Runtime
MVC	Model-View-Controller
SQL	Structured Query Language
DBMS	Database Management System
ORM	Objektově Relační Mapování
LINQ	Language Integrated Query
API	Application Programming Interface
MET	Metabolic Equivalent of Task
BMI	Body Mass Index
HTML	HyperText Markup Language
CRUD	Create, Read, Update, Delete
Int	Integer
Enum	Enumerable
CSV	Comma-Separated Values

SEZNAM OBRÁZKŮ

Obrázek 1. Architektura MySQL databáze [11]	23
Obrázek 2. Rozdělení vrstev	40
Obrázek 3. Struktura aplikační vrstvy	41
Obrázek 4. Struktura infrastrukturní	42
Obrázek 5. Datová struktura aktivit, tréninků a cílů	45
Obrázek 6. Datová struktura pro oblast stravy	46
Obrázek 7. Datová struktura pro spánek a srdeční tep	46
Obrázek 8. Datová struktura pro tabulky duševního zdraví	47
Obrázek 9. Datová struktura pro gamifikaci a sociální aspekt	48
Obrázek 10. Datová struktura tabulek vytvořených skrze ASP.NET Identity	49
Obrázek 11. Úvodní stránka	50
Obrázek 12. Registrační formulář	51
Obrázek 13. Přihlášení	52
Obrázek 14. Domovská stránka aplikace	52
Obrázek 15. Osobní profil uživatele	54
Obrázek 16. Sdílené milníky	54
Obrázek 17. Cizí profil	55
Obrázek 18. Stránka pro Import a export	56
Obrázek 19. Stránka aktivit	57
Obrázek 20. Modální formulář pro přidání aktivity	58
Obrázek 21. Stránka s tréninky	59
Obrázek 22. Stránka pro cíle	60
Obrázek 23. Sledování stravy	61
Obrázek 24. Zaznamenávání srdečního tepu	62
Obrázek 25. Přehled spánku	63
Obrázek 26. Domovská stránka pro duševní zdraví	64
Obrázek 27. Obsah deníku	65
Obrázek 28. Detail záznamu v deníku	65
Obrázek 29. Přehled nálad	66
Obrázek 30. Struktura historie	67

SEZNAM PŘÍLOH

- P I Export z aplikace
- P II Export z MiFitness

PŘÍLOHA P I: EXPORT Z APLIKACE

Activity	Quantity	StartTime	EndTime	Duration	Id	TypeName	METValue	QuantityType
13	2000	04/22/2024 07:26:44	04/22/2024 07:36:44	00:10:00.0000380	0		0	DistanceM
Goal	Name	TypeOfGoal	Quantity	GoalProgress	GoalMetric	IsDone	CompletionDate	UserId
3	Hubnutí	Hubnutí	5.00000000000000000000000000000000	0.00000000000000000000000000000000	0.00000000000000000000000000000000	False		704cdd4a-6f68
4	Počte opakování posilování - nízká zážeh	PocetOpakovaniPosilovaniNizkaZatez	5.00000000000000000000000000000000	0.00000000000000000000000000000000	0.00000000000000000000000000000000	False		704cdd4a-6f68
5	Hubnutí	Hubnutí	2.00000000000000000000000000000000	0.00000000000000000000000000000000	0.00000000000000000000000000000000	False		704cdd4a-6f68
Training	Name	UserId	Id	UserName	Email	PhoneNumber	FirstName	LastName
Id	Name	UserId	Id	UserName	Email	PhoneNumber	FirstName	LastName
Calories	Date	FoodTime	Id	Date	TotalCalories	UserId	Id	UserName
18	04/19/2024 08:24:10	Breakfast	0	01/01/0001	0			
19	04/25/2024 08:42:00	Breakfast	0	01/01/0001	0			
FoodItem	Name	kCalPerPortion						
Id	Name	kCalPerPortion						
1	Jablko	63						
CalorieFood	CalorieRecordId	Id	Date	FoodTime	Id	Date	TotalCalories	UserId
19	18	18	04/19/2024 08:24:10	Breakfast	0	01/01/0001	0	
20	19	19	04/25/2024 08:42:00	Breakfast	0	01/01/0001	0	
CaloriesPerDay	Date	TotalCalories	UserId	Email	PhoneNumber	FirstName	test	test
14	04/04/2024	252	704cdd4a-6f68-46de-890f-8fc352c65d47	test@test.com	test@test.com	test	test	test
15	04/10/2024	63	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test@test.com	test	test	test
16	04/18/2024	0	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test@test.com	test	test	test
19	04/19/2024	63	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test@test.com	test	test	test
20	04/25/2024	63	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test@test.com	test	test	test
WaterIntake	DrinkType	IntakeAmount	IntakeTime	UserId	Id	UserName	Email	PhoneNumber
12	Voda	220	04/18/2024 10:37:49	704cdd4a-6f68-46de-890f-8fc352c65d47	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test@test.com	
HeartRate	Date	HeartRateValue	UserId	Id	UserName	Email	PhoneNumber	FirstName
1	04/12/2024 11:33:12	120	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test@test.com	test	test	test
2	04/12/2024 09:33:16	76	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test@test.com	test	test	test
3	04/12/2024 09:33:58	69	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test@test.com	test	test	test
Sleep	Date	StartTime	EndTime	HoursSlept	Quality	Id	UserName	test
4	04/15/2024 08:58:48	04/15/2024 02:00:00	04/15/2024 05:00:00	180	Good	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test
5	04/19/2024 09:00:53	04/14/2024 02:00:00	04/14/2024 10:00:00	480	Bad	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test
6	04/19/2024 09:19:15	04/13/2024 00:00:00	04/13/2024 02:00:00	120	Bad	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test
7	04/19/2024 09:19:46	04/16/2024 00:00:00	04/16/2024 07:00:00	420	Perfect	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test
8	04/22/2024 10:24:22	04/22/2024 00:00:00	04/22/2024 07:03:00	423	Perfect	704cdd4a-6f68-46de-890f-8fc352c65d47	test	test
TrainingGoals	TrainingId	GoalId	Quantity	Name	Id	UserName	test	test
Id	TrainingId	GoalId	Quantity	Name	Id	UserName	test	test

PŘÍLOHA P II: EXPORT Z MIFITNESS

MiFitness_hlth_center_aggregated_fitness_data

Uid	Sid	Tag	Key	Time	Value	UpdateTime
8181782621	default	daily_report	sleep	1655424000	{avg_hr:0,avg_spo2:0,day_sleep_evaluation:3,segment_details:[{bedtime:1655466060,duration:22,timezone:3,wake_up_time:1655467380}],long_sleep_evaluation:0,sleep_deep_duration:0,sleep_deep_evaluation:0,sleep_light_dur:1655631179}	1655631179
8181782621	default	daily_report	sleep	1655337600	{avg_hr:0,avg_spo2:0,day_sleep_evaluation:3,segment_details:[{bedtime:1655390820,duration:65,timezone:3,wake_up_time:1655394720}],long_sleep_evaluation:0,sleep_deep_duration:0,sleep_deep_evaluation:0,sleep_light_dur:1655631179}	1655631179
8181782621	default	daily_report	sleep	1655510400	{avg_hr:0,avg_spo2:0,day_sleep_evaluation:3,segment_details:[{bedtime:1655550540,duration:25,timezone:3,wake_up_time:1655552040}],long_sleep_evaluation:0,sleep_deep_duration:0,sleep_deep_evaluation:0,sleep_light_dur:1655631179}	1655631179
8181782621	default	daily_report	heart_rate	1655337600	{abnormal_hr_count:0,avg_hr:77,max_hr:99,min_hr:67}	1655631179
8181782621	default	daily_report	heart_rate	1655424000	{abnormal_hr_count:0,avg_hr:81,max_hr:114,min_hr:63}	1655631179
8181782621	default	daily_report	heart_rate	1655510400	{abnormal_hr_count:0,avg_hr:81,max_hr:110,min_hr:60}	1655631179
8181782621	default	daily_report	heart_rate	1709164600	{abnormal_hr_count:0,aerobic_hr_zone_duration:0,anaerobic_hr_zone_duration:0,extreme_hr_zone_duration:0,fat_burning_hr_zone_duration:0,latest_hr:[{bpm:93,time:1709210880},dbKey]:heart_1712395451}	1712395451
8181782621	default	daily_report	heart_rate	1709856000	{abnormal_hr_count:0,aerobic_hr_zone_duration:0,anaerobic_hr_zone_duration:0,extreme_hr_zone_duration:0,fat_burning_hr_zone_duration:0,latest_hr:[{bpm:98,time:1709922360},dbKey]:heart_1712395451}	1712395451
8181782621	default	daily_report	heart_rate	1709942400	{abnormal_hr_count:0,aerobic_hr_zone_duration:0,anaerobic_hr_zone_duration:0,extreme_hr_zone_duration:0,fat_burning_hr_zone_duration:0,latest_hr:[{bpm:82,time:1709996040},dbKey]:heart_1712395451}	1712395451
8181782621	default	daily_report	heart_rate	1710547200	{abnormal_hr_count:0,aerobic_hr_zone_duration:0,anaerobic_hr_zone_duration:0,extreme_hr_zone_duration:0,fat_burning_hr_zone_duration:0,latest_hr:[{bpm:81,time:1710611763},dbKey]:single_1712395451}	1712395451
8181782621	default	daily_report	heart_rate	1712102400	{abnormal_hr_count:0,aerobic_hr_zone_duration:0,anaerobic_hr_zone_duration:0,extreme_hr_zone_duration:0,fat_burning_hr_zone_duration:0,latest_hr:[{bpm:84,time:1712166960},dbKey]:heart_1712395449}	1712395449
8181782621	default	daily_report	heart_rate	1712534400	{abnormal_hr_count:0,aerobic_hr_zone_duration:0,anaerobic_hr_zone_duration:0,extreme_hr_zone_duration:0,fat_burning_hr_zone_duration:0,latest_hr:[{bpm:91,time:1712602260},dbKey]:heart_1712395449}	1712395449
8181782621	default	daily_report	calories	1655337600	{calories:18}	1655631179
8181782621	default	daily_report	calories	1655424000	{calories:176}	1655631179
8181782621	default	daily_report	calories	1655510400	{calories:104}	1655631179
8181782621	default	daily_report	calories	1707609600	{calories:1}	1712395451
8181782621	default	daily_report	calories	1708008000	{calories:14}	1712395451
8181782621	default	daily_report	calories	1708605600	{calories:1}	1712395451
8181782621	default	daily_report	calories	1709164800	{calories:81}	1712395451
8181782621	default	daily_report	calories	1709337600	{calories:1}	1712395451
8181782621	default	daily_report	calories	1709510400	{calories:1}	1712395451
8181782621	default	daily_report	calories	1709856000	{calories:202}	1712395451
8181782621	default	daily_report	calories	1709942400	{calories:82}	1712395451
8181782621	default	daily_report	calories	1710547200	{calories:1}	1712395451
8181782621	default	daily_report	calories	1710633600	{calories:1}	1712395451
8181782621	default	daily_report	calories	1710720000	{calories:8}	1712395451