

Vývoj mobilní a webové fitness aplikace

Zdeněk Karlík

Bakalářská práce
2022



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

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

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Zdeněk Karlík
Osobní číslo: A19049
Studijní program: B3902 Inženýrská informatika
Studijní obor: Softwarové inženýrství
Forma studia: Prezenční
Téma práce: Vývoj mobilní a webové fitness aplikace
Téma práce anglicky: Development of Mobile and Web Fitness Applications

Zásady pro vypracování

1. Vypracujte literární rešerši na téma vývoj webových a mobilních aplikací, přičemž se zaměřte na technologie, které hodláte při vývoji použít.
2. Navrhněte webovou fitness aplikaci pro administraci dat.
3. Proveďte návrh mobilní fitness aplikace pro běžné uživatele.
4. Naprogramujte navržené aplikace.
5. Vyvinuté aplikace dostatečně popište.

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

Seznam doporučené literatury:

1. HOQUE, Shama. Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js. 2. Birmingham: Packt, 2018. ISBN 9781839215414.
2. SUBRAMANIAN, Vasan. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node. Berkeley, CA: Apress, 2017. ISBN 978-1484226520.
3. BODUCH, Adam a Roy DERKS. React and React Native. 3. Birmingham: Packt, 2020. ISBN 9781839211140.
4. React: Docs [online]. Menlo Park, Kalifornie, USA: Facebook, 2021 [cit. 2021-11-29]. Dostupné z: <https://reactjs.org/docs/getting-started.html>
5. React Native: Docs [online]. Menlo Park, Kalifornie, USA: Facebook, 2021 [cit. 2021-11-29]. Dostupné z: <https://reactnative.dev/docs/getting-started>

Vedoucí bakalářské práce:

Ing. Tomáš Vogeltanz, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **3. prosince 2021**

Termín odevzdání bakalářské práce: **23. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



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

Ve Zlíně dne 24. ledna 2022

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 12.5.2022

Zdeněk Karlík, v.r.
podpis studenta

ABSTRAKT

Bakalářská práce je zaměřená na vývoj mobilní a webové fitness aplikace. Webová část, která má administrační účel, je zaměřená na evidenci dat, které jsou poté součástí mobilního softwaru. Uživatelé, kteří mají vytvořený účet, mohou zaznamenávat data o svých trénincích prostřednictvím vyplnění formuláře v mobilní aplikaci. Pokud uživatel není přihlášený může si vyzkoušet omezenou verzi aplikace.

Klíčová slova: Webová aplikace, mobilní aplikace, MERN, React Native, fitness

ABSTRACT

The bachelor thesis deals with development of mobile and web fitness applications. The Web Part, which has an administrative purpose, deals on recording data, which is then part of the mobile software. Users who have created an account can record data about their trainings by filling out a form in the mobile app. If the user is not signed in, they can try a limited version of the application.

Keywords: Web application, mobile application, MERN, React Native, fitness

Tímto bych chtěl poděkovat Ing. Tomáši Vogeltanzovi Ph.D. za cenné rady, odborný přístup a ochotu při vedení mé 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 VÝVOJ WEBOVÉ APLIKACE	11
1.1 ZÁKLADNÍ WEBOVÉ TECHNOLOGIE.....	11
1.1.1 HTML	11
1.1.2 CSS.....	12
1.1.3 Javascript.....	12
1.2 MERN STACK.....	12
1.2.1 Databáze MongoDB.....	13
1.2.2 Node.js	13
1.2.3 Express.js	14
1.2.4 Knihovna React	14
1.2.4.1 React komponenty	15
1.2.4.2 Hooks	15
1.2.4.3 Atomic design	15
1.2.4.4 React-Router	16
1.2.4.5 React-Bootstrap	16
1.2.4.6 Stylování	16
2 VÝVOJ MOBILNÍ APLIKACE	18
2.1 REACT NATIVE.....	18
2.1.1 Nativní komponenty.....	18
3 POSTUPY A NÁSTROJE PRO VÝVOJ	20
3.1 TECHNOLOGICKÉ POSTUPY PŘI NÁVRHU APLIKACE	20
3.1.1 Požadavky systému	20
3.1.2 Model případu užití	21
3.1.3 Scénáře	22
3.1.4 Návrh UI.....	22
3.2 SYSTÉM PRO SPRÁVU VERZÍ.....	22
II PRAKTICKÁ ČÁST	24
4 NÁVRH WEBOVÉ A MOBILNÍ FITNESS APLIKACE	25
4.1 SBĚR A ANALÝZA POŽADAVKŮ WEBOVÉ APLIKACE	25
4.1.1 Manažerské požadavky	25
4.1.2 Administrátorské požadavky.....	25
4.2 MODEL PŘÍPADŮ UŽITÍ WEBOVÉ APLIKACE.....	25
4.3 SCÉNÁŘE	27
4.3.1 Přihlášení uživatele	27
4.3.2 Editace účtu	27
4.3.3 Obecné scénáře pro zobrazení a CRUD operace	28
4.3.4 Zobrazení registrovaných uživatelů	31
4.3.5 Zobrazení statistik	31
4.3.6 Scénáře pro zobrazení filtrace	31
4.3.7 Scénáře pro zobrazení uživatelů a jejich rolí	32

4.4	NAVRŽENÍ UI	34
5	NÁVRH MOBILNÍ APLIKACE	38
5.1	SBĚR A ANALÝZA POŽADAVKŮ MOBILNÍ APLIKACE	38
5.1.1	Neregistrovaný uživatel	38
5.1.2	Registrovaný uživatel.....	38
5.2	UC DIAGRAM MOBILNÍ APLIKACE.....	38
5.3	SCÉNÁŘE	39
5.3.1	Registrace uživatele	40
5.3.2	Výpočet a kalkulace	40
5.3.3	Zobrazení náhodných jídel.....	41
5.3.4	Zobrazení náhodných tréninků.....	41
5.3.5	Zobrazení tréninků	42
5.3.6	Zobrazení tréninků a operace s nimi	42
5.3.7	Zobrazení menu.....	44
5.3.8	Zobrazení osobních statistik.....	45
5.3.9	Zobrazení oblíbených tréninků	45
5.3.10	Editace účtu	46
5.4	NAVRŽENÍ MOBILNÍHO UI	46
6	NAPROGRAMOVÁNÍ APLIKACÍ A JEJICH POPIS	50
6.1	SERVER A TVORBA API	50
6.1.1	Vytvoření modelů.....	50
6.1.2	Controllery	53
6.1.2.1	Auth controller	53
6.1.2.2	Private controller.....	54
6.1.2.3	User, Bodypart, Workout, Training, Meal controller	54
6.1.3	Middleware	55
6.1.4	Routes.....	56
6.2	POPIS WEBOVÉ APLIKACE	56
6.2.1	Přihlášení a obnova hesla	56
6.2.2	Neoprávněný přístup	58
6.2.3	Dashboard a menu.....	59
6.2.4	Stránky částí těla	60
6.2.5	Stránky cviků	62
6.2.6	Stránky pro tréninky.....	63
6.2.7	Stránky pro jídla.....	65
6.2.8	Uživatelské stránky	67
6.3	POPIS MOBILNÍ APLIKACE	69
6.3.1	Navigační menu a hlavní obrazovka	69
6.3.2	Přihlášení, registrace a zapomenuté heslo.....	71
6.3.3	Kalkulační obrazovka.....	73
6.3.4	Zobrazení tréninků a operace s nimi	74
6.3.5	Zobrazení jídel a jejich detailu	78
6.3.6	Editace profilu.....	80
	ZÁVĚR	81
	SEZNAM POUŽITÉ LITERATURY.....	82
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	84

SEZNAM OBRÁZKŮ	86
SEZNAM TABULEK.....	88
SEZNAM PŘÍLOH.....	89

ÚVOD

Sportovní odvětví fitness je v dnešní době velmi populární. Spousta sportovců při cvičení využívá různé typy aplikací, které zaznamenávají údaje o trénincích prostřednictvím chytrých zařízení. Tato zařízení ale ve většině případů nesvedou rozpoznat veškeré detaily tréninku. Navíc pokud chceme mít přesné výsledky, je nutné mít pořízené vybavení, které je z hlediska finanční stránky velmi nákladné.

Většina mobilních aplikací funguje na principu, který má snahu uživateli buď ukázat, jak správně cvičit nebo mu přímo generuje trénink. Tohle řešení je vhodné pro začínající sportovce, kteří nemají v oblasti fitness mnoho zkušeností.

Sportovec, který je již znalý nepotřebuje znát správné provedení cviku z důvodu, že už většinu cviků zná a ví, jak fungují. Také nepotřebuje, aby mu aplikace tvořila tréninky, protože si je již sestavuje sám a již pozná, co pro něj funguje nejlépe. Daleko prospěšnější je mít zaznamenané údaje o provedených trénincích, aby věděl, na co se příště více zaměřit. Pro některé sportovce je důležité mít také inspiraci v trénincích jiných uživatelů a v případě zájmu mít možnost si konkrétní trénink uložit, pro pozdější možnost zobrazení. Tyto poznatky byly hlavní motivací pro vytvoření nové mobilní aplikace, která bude fungovat především na principu uchovávání dat z tréninků.

Mobilní aplikace, vyvíjená pomocí React Native, je doplněná o webovou aplikaci, která je zaměřena na administraci a správu dat. Mezi technologie využívané při vývoji webu spadá především React, Javascript, HTML a CSS. Naplnění uživatelských požadavků, validaci na straně serveru a operace s daty poté zprostředkovává Node.js a framework Express.js.

I. TEORETICKÁ ČÁST

1 VÝVOJ WEBOVÉ APLIKACE

Na úvod práce je nutné objasnit pojem webová aplikace. Jedná se o typ softwaru, který je uložený na webovém serveru. V případě, že naše aplikace vyžaduje práci s databází, musíme použít také aplikační server. Hlavním účelem aplikace je usnadnit uživatelům komunikaci, získávání a poskytování informací, správu dat a v neposlední řadě zaujmout návštěvníky webu. Komunikace mezi uživatelem a softwarem je zprostředkována pomocí požadavků, které jsou odesílány na webový server a poté předány aplikačnímu serveru, kde dochází k provedení požadovaných operací a práce s daty. Následně je výsledek zaslán zpět na webový server, který uživateli vrátí odpověď na požadavek. U aplikací webového typu se standardně se využívá kombinace klientské a serverové skriptovací strany. Velkou výhodou je, že pro fungování aplikace není vyžadováno její stažení. Mezi největší nevýhody spadá nutnost připojení k internetu, zabezpečení, velikost úložiště a potřeba vysokého výkonu serveru. [1]

1.1 Základní webové technologie

Mezi nejčastější programovací jazyky, které jsou využívány ve webových aplikacích na straně klienta, řadíme HTML, CSS a Javascript. Jejich cílem je prezentace informací a vytvoření rozhraní, které dokáže uživatele zaujmout. Důležitým přínosem je, že mají podporu u většiny webových prohlížečů. [1]

1.1.1 HTML

Značkovací jazyky zajišťují správu a zobrazení webových dokumentů za pomoci softwaru, který v sobě obsahuje webový prohlížeč. Zobrazování jednotlivých částí dokumentů je prováděno pomocí značek, které jsou uzavřeny do speciálních symbolů v podobě lomených závorek. Tyto značky, označované jako tagy, mohou být párové nebo nepárové. U párových se tag nachází před a za obsahem. V případě nepárových mohou stát samostatně, jako třeba zalomení řádku. Jejich hlavním záměrem je vytvořit strukturu stránky pomocí prvků jako jsou nadpisy, odkazy, odstavce a jiné. Jednotlivé tagy lze upravovat pomocí atributů a tím měnit jejich vlastnosti. [2]

HTML5 je od roku 2014 doporučenou verzí značkovacího jazyka HTML. Z důvodu co nejlepší pozice při vyhledávání ve webových prohlížečích, kterou zajišťuje SEO, je nutné psát kód správně a snažit se co nejvíce využít správných tagů. [2][3]

1.1.2 CSS

Jedná se o kaskádový jazyk, který zabezpečuje vzhled a styl HTML stránky. Díky němu jsme schopni měnit velikost, barvu, typ a jiné vlastnosti tagů. Jazyk je založený na dědičnosti, což znamená, že potomci, kterými mohou být nadpisy, média, odstavce a další, uzavření mezi párovými tagy dědí automaticky vlastnosti rodičovského tagu. Velkou předností je, že styl lze použít na více místech najednou a tím dochází k zamezení duplicitního kódu. Součástí CSS je také zajištění responzivního designu a jednoduchých animací. V dnešní době je využívána verze CSS3.[4]

1.1.3 Javascript

Javascript je skriptovací jazyk, který je vykonáván nejčastěji na straně klienta a je využíván k ovládání interakcí na webové stránce. I přesto, že samotné CSS dnes již nabízí spoustu interakcí, tak má Javascript spoustu benefitů. Jedním z největších je určitě jeho schopnost práce s objekty, díky čemuž dokáže manipulovat s jednotlivými částmi webové stránky. Další velkou předností je jeho schopnost reagovat na určité události jako jsou pohyb myši, načtení webu nebo kliknutí na odkaz. V souvislosti s těmito vlastnostmi je často využívám pro ověřování správnosti formulářů a korektnosti vstupů. Navíc dokáže provádět matematické výpočty a tvořit komplikovanější animace. [2]

Samotný Javascript je sice velmi hojně využíváný, ale v dnešní době většina moderních webů využívá výhody frameworků. Jedná se o softwarové knihovny, které jsou založeny na samotném Javascriptu. Hlavním záměrem frameworků je usnadnění a urychlení vývoje a psaní výsledného kódu. Mezi nejpobulárnějšími frameworky, co se týká vývoje front end části, sloužící k prezentaci viditelné části webu, řadíme Vue.js, Angular nebo třeba Ember. Velmi populární je také knihovna React, která není sama o sobě považována za framework. V případě, že chceme využít Javascript i na straně serveru, musíme využít prostředí v podobě Node.js Zde se dá také volit z více frameworků, z nichž nejpobulárnější je Express.js [5][6]

1.2 MERN Stack

Webové aplikace lze vyvíjet pomocí různých technologií, které lze vzájemně kombinovat. Pro sdružení více technologií se používá výraz stack. Jedním z nejpobulárnějších se v minulosti stal LAMP. V dnešní době je ale daleko pobulárnější využití jednostránkové aplikace, které mají snahu vyvarovat neustálému obnovování webu. Místo toho je kladen důraz na

využití volání serveru k získání dat a následné aktualizace webové stránky. Díky tomu, není nutné vždy celou stránku znovu načíst, ale stačí změnit jen menší část. Tohle vedlo k popularizaci frameworků a tvorbě nových stacků jako jsou MERN a MEAN. Oba využívají spojení Node.js a Express.js, který dohromady představují webový server. Pro ukládání dat je využívána NoSQL databáze MongoDB, která nevyužívá tradičních tabulek typických pro relační databáze, ale ukládá data ve formátu BSON, který zastupují tvar binárního JSON. Jediným rozdílem mezi stacky je volba front end technologie. Zatímco MEAN využívá framework Angular postavený na architektuře MVC, MERN využívá knihovnu React, která je používána pouze pro vykreslení View ze vzoru MVC. Obě kombinace technologií se řadí mezi nejpoblárnější pro vývoj nových webových aplikací. [6]

MERN stack má spoustu předností. Tou největší je, že všechny technologie jsou založeny pouze na jediném programovacím jazyce a tím je Javascript. Díky tomu můžeme kombinovat kód na straně klienta společně se serverovou logikou. Toho se dá využít u tvorby formulářů, kdy ošetřování uživatelských vstupů řešíme na straně klienta a ochranu dat ze strany serveru. Další přínosem je, že si ušetříme práci s transformací dat, za pomoci využívání JSON formátu, který lze využívat úplně všude. Mezi další výhody spadá architektura Node.js, npm balíčkovací systém a to, že React není framework, ale pouze knihovna. [6]

1.2.1 Databáze MongoDB

MongoDB je jednou z NoSQL databází, která místo ukládání pomocí tabulek, jako u relačních databází, využívá ukládání pomocí kolekcí, dokumentů a polí. Každý dokument má svůj vlastní identifikátor, který je tvořen automaticky. Tento identifikátor se nazývá index a je unikátní. Jedná se o objektově orientovanou databázi, která je založena na vytváření objektů, díky čemuž je velmi jednoduchá na pochopení. V souvislosti s tím nemusíme řešit problémy s ORM, které nastávají u relačních databází. Při využívání formátu JSON můžeme zanořovat objekty, jakkoliv hluboko, a tím tvořit jeden velký dokument, bez nutnosti vytváření více tabulek. Ve spojení s tím nám ovšem vznikají duplicitní data, což pro nás znamená větší náročnost na velikost datového uložště. [6][7]

1.2.2 Node.js

Node je prostředí pro vývoj aplikací v jazyce Javascript, který se ovšem nenachází na straně klienta, ale na straně serveru. Jeho základem je engine Chrome V8, na němž je postavený i samotný Google Chrome. Poskytuje architekturu založenou na volání asynchronních

událostí a díky tomu nedochází k blokování vstupů a výstupů. Z toho vzniká daleko nižší doba trvání na vyřízení jednotlivých požadavků, velmi dobrá škálovatelnost a tvorba aplikací v reálném čase, využívajících zpracování více požadavků najednou. [8][9]

Common.js je systém založený na modulech, využívaný při používání Node.js. Jeho hlavním cílem je rozdělení kódu do jednotlivých modulů. Tím přispívá především k lepší organizaci. Moduly jsou poté volány pomocí klíčového slova `required`. Dalším pozitivním aspektem je, že jednotlivé moduly na sebe mohou vzájemně odkazovat, což u běžného Javascript souboru není možné. Samotný Node obsahuje po instalaci spoustu základních modulů. [6]

1.2.3 Express.js

Express.js je framework, který se nachází na serverové straně a obstarává vytváření webových aplikací společně s prostředím Node. Využívá se především pro usnadnění napsání zdrojového kódu webového serveru, který by šel napsat i pomocí samotného Node.js, ale jednalo by se o složitější proces. [8]

Velmi důležitou částí je `express middleware`, který je využíván pro funkčnost přihlášení a registrace uživatelů, autentizaci a autorizaci. Pomocí frameworku jsme schopni využívat směrování API na straně serveru, což vede k umožnění přístupu ke statickým souborům uložených v databázi. Je schopný zajistit nastavení odpovědí ze serveru a práci s cookies. Směrování a odpověď na něj jsou zajišťovány pomocí konkrétního URI a HTTP metod požadavků. [6][10]

Mezi typické metody požadavků řadíme především `get`, `post`, `put` a `delete`. `Get` je využívána pouze k vyžádání dat z daného zdroje. Naopak metoda `post` zajišťuje zaslání dat na server, čímž můžeme vytvořit nový záznam nebo nějaký starší aktualizovat. `Put` má stejnou funkcionalitu jako `post` s rozdílem toho, že při vícenásobném spuštění požadavku `put` nám vždy vygeneruje stejný výsledek. Naopak v případě opakovaném spuštění `post` dochází k vytváření stejného záznamu vícekrát. `Delete` je metoda, která obstarává odstranění daného zdroje. [11]

1.2.4 Knihovna React

Jedná se o open-source knihovnu, na jejímž vývoji se podílí především společnost Facebook. Jejím hlavním cílem je vytvoření front end části webové stránky. Díky tomu, že nevyužívá architektonický vzor, jako je MVC, získáváme možnost měnit pouze view, a přitom nezměníme logiku aplikace. [6]

1.2.4.1 React komponenty

Knihovna se dělí na dva dílčí sektory. V prvním sektoru se nachází API pro komponenty, jehož součástí jsou jednotlivé části stránky, které chceme vykreslit v react DOM. Ve druhém sektoru je samotný React DOM, což je segment, který bude skutečně zobrazen na webové stránce. Mezi komponentami se vyskytují data, životní cykly, události a syntaxe JSX. Data zajišťují zobrazení informací, které odněkud pocházejí. Součástí životního cyklu jsou metody a Hooks, které nám zajišťují, aby aplikace byla schopná reagovat v čase. V případě, že je potřeba reakce na uživatelské interakce, využívá se událost. Posledním zásadním prvkem jsou syntaxe JSX. Jejich úlohou je tvorba struktury React komponent pro výsledné uživatelské rozhraní. [12]

Functional komponenty se chovají podobně jako klasické React komponenty. Jejich hlavním rozdílem je, že při jejich psaní vzniká daleko menší množství kódu a jsou jednodušší na pochopení. U klasické varianty také musíme při tvorbě komponenty použít klíčové slovo class a v komponentě metodu render. V případě Functional komponenty se nepoužívá ani jedna část. [13]

1.2.4.2 Hooks

Jedná se o funkce, které přidávají nové chování React komponentám a umožňují jim přístup k aktuálnímu stavu. Výhodou je především v tom, že funkce se mezi moduly sdílejí snadněji než celé třídy. [12]

1.2.4.3 Atomic design

Členění komponent v Reactu nám zajišťuje zlepšení přehledu a znovu použitelnosti kódu. Jednou z metod členění je Atomic design, což je metodika založená na skládání webu z jednotlivých fází, které spolu dohromady představují hierarchii webové stránky. Je složený z pěti základních stupňů a to atomů, molekul, organismů, šablon a stránek. [14]

Základem všech fází jsou atomy, které reprezentují základní HTML prvky. Do této skupiny spadají tagy, do kterých se již běžně nekládají další tagy. Jedná se o jednoduché prvky jako jsou tlačítka, obrázky nebo formuláře. Každý atom má své specifické vlastnosti, které by měly být jedinečné. [14]

Z jednotlivých atomů vznikají molekuly, které primárně zajišťují tvorbu jednotlivých částí uživatelského rozhraní. Molekuly by měly být tvořeny tak, aby sloužily k jednomu

jednoduchému účelu. To souvisí s usnadněním výsledného testování a podporuje znovu použitelnost. [14]

V případě organismů už se jedná o složitější složky UI, které jsou vytvářeny pomocí molekul a atomů, případně i jiných organismů. Při jejich využití nám vznikají větší dílčí celky jako jsou záhlaví, zápatí nebo tělo webové stránky. [14]

Poslední částí, co se týká struktury stránky, je šablona. Jedná se o objekt, které již zobrazuje výslednou webovou stránku, její vzhled a rozložení jednotlivých komponent. [14]

K vytváření konkrétních instancí šablon jsou využívány stránky. Dochází zde k vygenerování konečného obsahu a předání logiky do jednotlivých menších celků. [14]

1.2.4.4 React-Router

Samotný React zajišťuje pouze vytvoření UI a ovládání jednotlivých interakcí. V případě využití více stránek, které mají svůj specifický účel, je nutné řešit i jednotlivé přechody mezi nimi. Změna jednotlivých stránek je prováděna pomocí směrování a adresy URL v prohlížeči. Za účelem směrování, včetně jeho historie, byla vytvořena knihovna React-Router. Díky ní lze snadno zajistit spojitost mezi konkrétními stránkami, a navíc zajišťuje ovládání tlačítka zpět v prohlížeči. [6]

1.2.4.5 React-Bootstrap

Bootstrap je jedním z nejpopulárnějších CSS frameworků, který je využíván pro design UI. Z tohoto důvodu byla vytvořena knihovna, která nabízí většinu běžných funkcí samotného frameworku. Tato knihovna nese název React-Bootstrap, a kromě základních komponent, které nám poskytuje, obsahuje také informace o tom, jak vytvářet a stylovat vlastní komponenty. [6]

1.2.4.6 Stylování

Stylování je důležitou součástí samotného Reactu. Při tvorbě aplikace a psaní CSS máme více způsobů, které se dají rozdělit do pěti základních. [15]

Inline styly jsou velmi jednoduché a rychlé na vytvoření a jsou vloženy přímo v kódu do jednotlivých prvků. Tento způsob není příliš používaný díky tomu, že výsledný kód se velmi rychle stává nečitelným a také díky tomu, že nelze využívat funkce CSS jako jsou animace a selektory. [15]

Namísto inline stylů se často využívají externí CSS soubory, které jsou vkládány pomocí importů a tím dochází k přehlednosti v kódu. Na rozdíl od inline stylů navíc podporují veškerou funkcionalitu a nástroje CSS. Nevýhodou mohou být konflikty vzniklé v případě chybného pojmenování stejných tříd, id nebo využití stejných tagů. [15]

SASS a SCSS využívají funkcionalitu CSS, ale rozšiřují ji o další funkce jako jsou proměnné nebo vnořené stylování. SASS nevyžaduje psaní závorek, tak jako u kaskádových stylů, na rozdíl od SCSS. Oba styly ovšem snižují počet řádků, které by u klasického CSS museli být napsány. K jejich fungování je nutné nainstalovat další knihovnu, která se jmenuje node-sass. [15]

CSS moduly jsou psány jako klasické CSS, ale jsou importovány jako objekty. Velkou výhodou je, že při vložení modulu do konkrétní komponenty nedochází ke stylování podřízených komponent. Díky tomu dochází k zamezení nežádoucích problémů s dědičností. [15]

Poslední stylování využívá psaní CSS přímo v souborech Javascriptu. Při změně stylu komponenty nemůže nastat situace, že ovlivníme komponentu jinou. K tomu, aby psaní bylo pohodlné napomáhají knihovny styled-components a emotion. [15]

2 VÝVOJ MOBILNÍ APLIKACE

Mobilní aplikace je druh softwaru, který je navržený pro práci na mobilních zařízeních. Jejím hlavním cílem je, aby byla uživateli poskytnuta podobná funkcionalita jako u počítačových softwarů. Na rozdíl od integrovaných systémů jsou aplikace zaměřeny na určitý úkol, který mají splnit. Dnes jsou děleny na dva základní typy, a to nativní a webové. Nativní aplikace jsou vyvíjeny pro konkrétní operační systém a mají propracovanější UI. [16]

2.1 React Native

Tvorba aplikací pro mobilní zařízení je rozdílná v závislosti na používané platformě. U vývoje pro operační systém Android se nejvíce využívá programovací jazyk Java. V případě systému, která je postavený na iOS, jsou nejčastěji aplikace vyvíjeny pomocí jazyka Swift. Z důvodu usnadnění vývoje, snížení finančních nákladů a časových důvodů vznikl React Native. Tato technologie, která pracuje se stejnou knihovnou jako v případě webových technologií, využívá asynchronního volání dle určitého operačního systému a za pomoci API dodává komponenty na mobilní platformu. Tyto komponenty jsou ve skutečnosti HTML elementy. [12]

React Native vygeneruje nový DOM, který funguje na mobilních platformách. Důvodem, proč je DOM tvořen zvlášť pro mobilní zařízení a webové stránky je ten, že webové prohlížeče na mobilních zařízeních nejsou pro běžné uživatele tak populární jako nativní aplikace. Mezi hlavní příčiny spadá to, že iterace, které uživatel používá běžně na webu, nejsou vhodné pro mobilní zařízení. Webová aplikace, na rozdíl od té mobilní, která využívá systému gest, počítá, že bude ovládána pomocí myši a klávesnice. Tento problém odstraňuje právě React Native, který je vytvořený pro ovládání pomocí gest. Další výhodou je automatická aktualizace. V případě, kdy se mobilní platforma rozhodne přejít na novou verzi, dojde k aktualizování komponent i v aplikaci. [12]

2.1.1 Nativní komponenty

Prvky, u kterých si React Native vygeneruje odpovídající zobrazení dle operačního systému, nazýváme nativní komponenty. Samotný React Native v sobě obsahuje sadu základních komponent. Kromě této sady lze vytvářet vlastní nativní komponenty, které pokryjí specifické požadavky aplikace. [17]

View je jednou ze základních prvků a zajišťuje vytvoření kontejneru a následného vygenerování základního rozvržení UI za pomoci flexboxu a stylování. [17]

V případě, že view bude větší než samotná obrazovka, musí být navíc přidán prvek scrollView. Jeho funkcí je vytvoření rolovacího zobrazení. [17]

Mezi další základní komponenty spadá text pro zobrazení textových řetězců, image pro znázornění obrázků a textInput pro možnost zadávání textů od uživatele. [17]

3 POSTUPY A NÁSTROJE PRO VÝVOJ

Pro vývoj aplikace je nutné zvolit vhodné nástroje a postupy. V případě, kdy vývojář nepoužívá správné nástroje anebo má nesprávný návrh aplikace, dochází k tvorbě chyb, které mohou ve výsledku vývoj zpomalit. [18]

3.1 Technologické postupy při návrhu aplikace

Návrh aplikace je velmi důležitá část vývoje. Standardem, který se věnuje návrhu a analýze při vývoji aplikací, se stal UML. Tento nástroj dokáže výrazně usnadnit pozdější vývoj. UML diagramy se dají získat více způsoby. Jedním z jednodušších způsobů je načrtnutí diagramu. V této fázi se většinou snažíme zachytit základní návrh systému a získat co nejpřesnější představu klienta o finální aplikaci. Druhou možností je vytvořit návrh s využitím podpurných programů. Pomocí vzniklého plánu jsou programátoři schopni rychlejšího zorientování ve funkcionalitě systému a poté využít diagram jako dokumentaci. Při správném vytvoření UML návrhu je z něj možné vygenerovat základní kód, který lze uplatnit ve výsledné aplikaci. [19]

3.1.1 Požadavky systému

Za požadavek je považována každá jednotlivá část funkce systému. Tyto funkce jsou shrnutím zákaznických potřeb, které se zjišťují pomocí analýzy požadavků. Prvním typem jsou obecné uživatelské požadavky. Jsou zachyceny běžným jazykem a jejich cílem je zachytit služby a jejich omezení, které má systém poskytovat. V případě, kdy je využíván podrobný popis funkcionality systému, musí být sepsány systémové požadavky, které by měly sloužit jako definice a dokumentace výsledných funkcí aplikace. [20]

Při sběru požadavků dochází k rozdělení na funkční a nefunkční. V případě funkčních jde o popis toho, co a jak by měl systém vykonávat. Tyto požadavky jsou závislé na uživateli a jejich interakcích. U tvorby funkčních požadavků dochází k problémům s konzistencí. Mezi hlavní důvody spadá nedostatečné definování specifikací pro složitější systémy a více zúčastněných stran, které mají různé očekávání. Druhým typem jsou požadavky nefunkční, které se netýkají přímo služeb poskytovaných systémem. Tyto požadavky, které jsou u většiny softwaru kritické, se týkají vlastností softwaru jako je spolehlivost, doba odezvy nebo kapacita paměti. Nefunkční požadavky jsou děleny do tří základních skupin a to produktové, organizační a externí. V případě produktových dochází k zaměření na specifikaci chování a doby běhu určitého softwaru jako jsou rychlost spuštění systému, spolehlivost nebo

bezpečnostní požadavky. Mezi organizační řadíme nezbytné zásady, které si může stanovit zákaznická nebo vývojářská strana. Součástí zásad mohou být zvolené programovací jazyky, vývojové prostředí nebo procesní standardy. Poslední skupinou jsou požadavky externí. Jedná se o regulace, které je nutné splnit, aby mohl být systém schválen příslušným orgánem. [20]

3.1.2 Model případu užití

Model případu užití je využíván pro zobrazení interakcí mezi uživateli a systémem. Hlavním účelem modelu je upřesnění funkčních požadavků systému, a také je využíván jako základ návrhu, analýzy a dokumentace. UC model se skládá z více složek. Mezi ty základní jsou řazeni aktéři, případy užití a asociace. Z pokročilých se využívají ohraničení, zobecnění a závislosti. [21]

Aktéři symbolizují uživatelské role, které mají rozdílná přístupová práva. Aktérem může být člověk, ale také jakýkoliv vnější systém. Každý aktér musí mít svůj specifický název a kvůli čitelnosti by měli být nadřazení aktéři umístěni výše než jejich potomci. [21]

Případy užití udávají, jakým způsobem budou splněny cíle, kterých chtějí konkrétní aktéři dosáhnout. Dílčí případy užití by měli obsahovat název, který by měl začínat slovesem. Nadřazené případy by měli být umístěny výše než zděděné. [21]

Asociace jsou posledním prvkem základních složek a zajišťují komunikaci mezi aktéry a případy užití. [21]

Ohraničení je komponenta, která zajišťující vytvoření hranice určité části systému. Jednotlivé případy užití jsou poté zobrazeny v této oblasti na rozdíl od aktérů, kteří se v oblasti nenacházejí. [21]

Zobecnění udává vztah, kdy podřízený případ nebo aktér dědí od nadřazeného. Podřízený je schopný vykonávat veškeré operace jeho rodiče, a navíc využívat svých vlastních operací. [21]

Poslední složkou jsou závislosti, které mohou být dvou typů, a to include a extend. V případě include jde o rozšíření základního případu užití o další funkce, které jsou ovšem nezbytné k funkčnosti systému. Extend je vztah, který umožňuje přidání volitelné funkcionality. Na rozdíl od include na něm není konkrétní případ užití závislý. [21]

3.1.3 Scénáře

Samotný UC model zobrazuje pouze funkce systému. Pro detailnější popis jednotlivých případů užití se využívá scénářů. Každý základní scénář by se měl skládat z bodů jako popisek, aktérů a jednotlivých toků, které nám upřesňují výslednou specifikaci. [22]

Popisek by měl být krátký a výstižný. Jeho cílem je ukázat funkcionalitu celého případu užití z uživatelského pohledu. Dalším bodem je vyjmenování aktérů, kteří jsou v případě užití zahrnuti. Hlavním důvodem, proč jsou scénáře využívány, je základní a alternativní tok. V případě základního toku se jedná o předpokládaný průběh interakce mezi aktérem a jednotlivými případy užití. Tento tok je zaznamenán pomocí scénáře, ve kterém dochází ke střídání funkcí aktérů. Pokud máme v systému možnost, při které může nastat odchýlení od základního scénáře, využijeme alternativní tok. Jedná se o chyby ze strany aktérů jako špatné vyplnění formuláře nebo nesprávné zadání hesla. U některých případů užití mohou být definovány podmínky pro spuštění a podmínky pro dokončení. [22]

3.1.4 Návrh UI

Jednou z nejdůležitějších částí, co se týká struktury a rozložení jednotlivých prvků aplikace, je tvorba wireframů. Jejich hlavním cílem je zachycení základního UI jednotlivých stránek a jejich interakcí. Díky jednoduchosti a rychlému vytváření wireframů lze ušetřit spoustu času. Další výhodou je, že koncový uživatel z něj velmi snadno dokáže pochopit jednotlivé funkcionality systému. Existují tři typy wireframů, a to s nízkou, střední a vysokou přesností. U nízké přesnosti se jedná o papírový model, který je vytvořen nahrubo ručně na papír a nemá pevně dané měřítko. Tento model zobrazuje pouze jednoduché prvky jako nadpisy, obsah stránky a funkce, které mají být vykonány při konkrétních interakcích. V případě wireframu se střední přesností jsou již využívány měřítko a výsledný model je standardně využíván pro komunikaci s koncovým uživatelem. Wireframy s vysokou přesností jsou rozložením velmi podobné přesnosti střední s tím rozdílem, že již obsahují konkrétní texty nadpisů, obrázky a data, která budou v aplikaci využívána. Díky tomu jsou modely s vysokou přesností velmi podobné konečnému designu aplikace. [23]

3.2 Systém pro správu verzí

Systém pro správu verzí obstarává sledování historie změn v projektu. Tyto změny mohou být prováděny jedincem nebo vývojářským týmem. Největší výhodou tohoto systému je, že se lze kdykoliv vrátit ke dřívějším verzím projektu. Díky přístupu k historii lze zpětně získat

informace o provedených změnách, kdo je provedl, kdy byly provedeny a také proč byly provedeny. Nejpopulárnějším systémem pro správu verzí je v dnešní době Git. [24]

II. PRAKTICKÁ ČÁST

4 NÁVRH WEBOVÉ A MOBILNÍ FITNESS APLIKACE

Navržení webové části aplikace je rozděleno do čtyř základních bodů. Prvním bodem je shrnutí uživatelských požadavků pomocí analýzy. Dalším bodem je zakomponování požadavků do jednotlivých případů užití a vytvoření UC modelu. Do vytvořených modelů jsou poté dosazeny konkrétní scénáře. Posledním krokem je návrh UI pomocí wireframů.

4.1 Sběr a analýza požadavků webové aplikace

Hlavním smyslem webové fitness aplikace je správa dat a administrace. Mezi klíčové požadavky spadá přihlášení do role manažera a administrátora. Tyto role mají na rozdíl od běžných uživatelů spoustu funkcí navíc včetně přístupu ke statistikám. Běžní a neregistrovaní uživatelé nemají přístup do administrátorské části. Na role běžných a neregistrovaných uživatelů je zaměřena především mobilní aplikace.

4.1.1 Manažerské požadavky

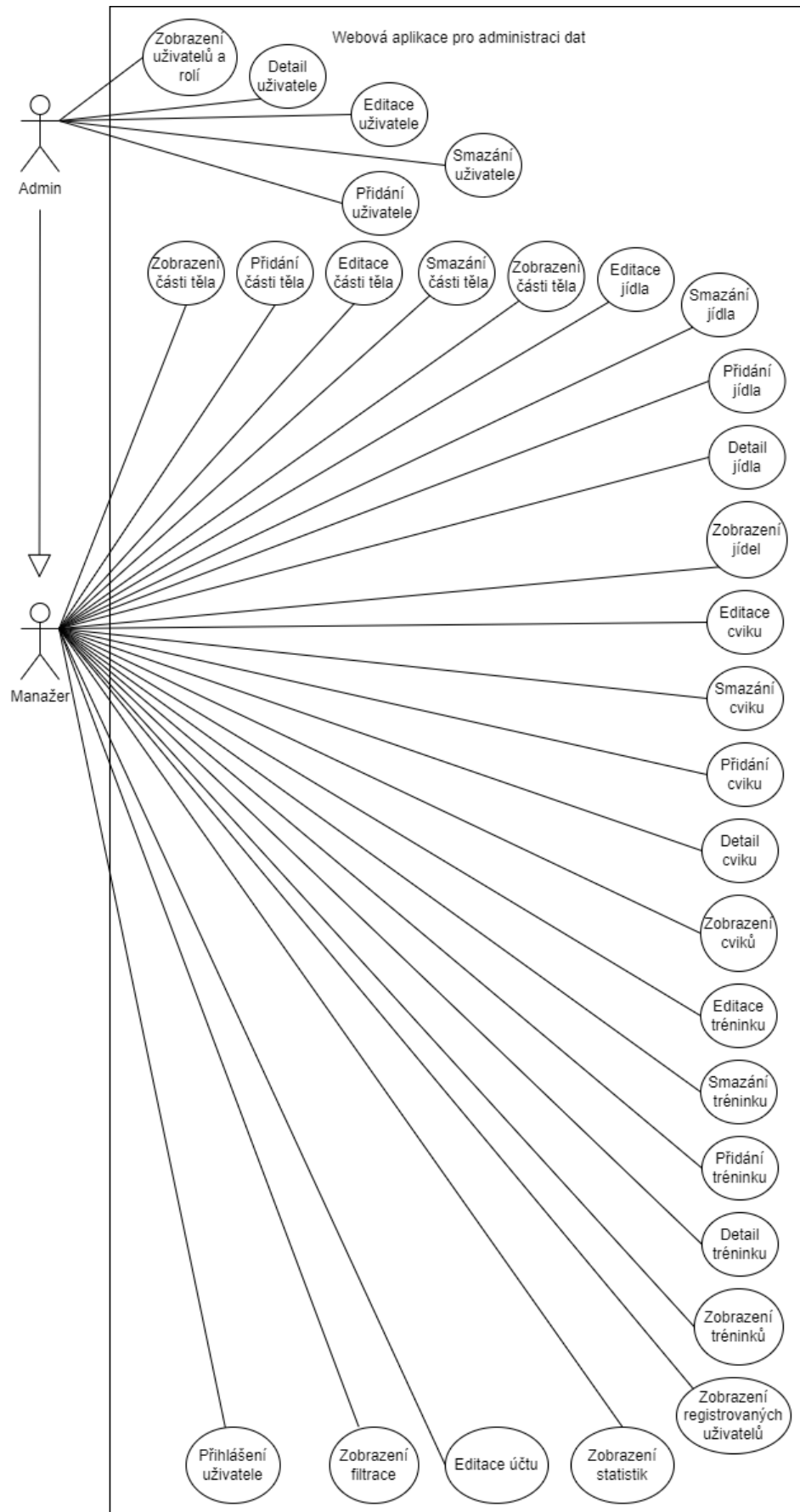
- Má možnost přihlášení do systému
- Může provádět všechny operace, které jsou dostupné běžnému uživateli.
- Vytváření, mazání, zobrazení a editace jídel z jídelníčku.
- Vytváření, mazání, zobrazení a editace částí těla.
- Vytváření, mazání, zobrazení a editace jednotlivých cviků.
- Vytváření, mazání, zobrazení a editace jednotlivých tréninků.
- Zobrazení tabulky se všemi registrovanými uživateli.
- Zobrazení statistiky ohledně tréninků, jídel a registrovaných uživatelů.
- Systém umožní filtrovat a vyhledávat jídla, části těla, cviky a uživatele.

4.1.2 Administrátorské požadavky

- Může provádět veškeré operace co manažer.
- Kromě zobrazení uživatelů je bude moci i vytvářet, editovat, smazat a určovat jejich role.

4.2 Model případů užití webové aplikace

V základním webovém UC modelu jsou zobrazeni dva aktéři, a to manažer a admin. Každý aktér má své specifické případy užití a vytvořené scénáře. Jednotlivé případy byly získány z analýzy požadavků. Celý diagram je zobrazen na obrázku níže.



Obrázek 1. Model případu užití webové aplikace

4.3 Scénáře

Součástí UC modelu jsou jednotlivé scénáře. Pomocí nich jsou prováděny operace s databází a práce s daty, které mohou být vytvářeny, editovány nebo mazány. Mezi další operace, které se ve scénářích vyskytují, spadá zobrazení dat a jejich filtrace.

4.3.1 Přihlášení uživatele

Uživatelé, kteří již mají vytvořený účet, mají mít možnost přihlášení do systému. Přihlášení probíhá pomocí vyplnění formuláře, kde je zadáno heslo a email. V případě úspěšného zadání údajů je uživatel přihlášen do systému, kde je mu následně poskytnuta funkcionální role.

Tabulka 1. Scénář – přihlášení uživatele

Název scénáře	Přihlášení uživatele
Akteři	Registrovaný uživatel
Podmínky pro spuštění	Uživatel musí být zaregistrovaný
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko přihlásit se od uživatele 2. Systém zobrazí uživateli formulář pro přihlášení 3. Uživatel vyplní formulář 4. Uživatel odešle formulář 5. Systém zkontroluje správnost dat 6. Systém přihlásí uživatele
Alternativní tok	5a. Data byla zadána špatně
Podmínky pro dokončení	Zobrazení dialogu o úspěšném přihlášení

4.3.2 Editace účtu

Přihlášení uživatelé, kteří mají oprávnění pro využívání webové aplikace, mohou editovat svůj vlastní účet. Po kliknutí na editaci účtu jim bude zobrazen formulář s vlastními předvyplněnými daty z databáze. Tato data mohou libovolně měnit.

Tabulka 2. Scénář – editace účtu

Název scénáře	Editace účtu
Aktéři	Manažer, admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli manažera nebo admina
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko editovat účet 2. Systém zobrazí uživateli formulář s vlastními předvyplněnými daty 3. Uživatel upraví vstupy ve formuláři 4. Uživatel odešle formulář 5. Systém zkontroluje správnost dat 6. Systém upraví údaje o uživateli v databázi
Alternativní tok	<p>5a. Data byla zadána špatně</p> <p>5b. Povinná data byla odstraněna</p>
Podmínky pro dokončení	Zobrazení dialogu o úspěšné změně dat

4.3.3 Obecné scénáře pro zobrazení a CRUD operace

Úspěšné zobrazení všech tréninků, cviků, částí těla a jídel má k dispozici pouze manažer a admin. Všechny prvky zobrazení mohou být libovolně editovány, smazány, zobrazeny a lze do nich vkládat nové záznamy a data. Veškeré scénáře, co se týká CRUD operací, vychází ze stejného obecného scénáře, u kterého je pouze zaměněno slovo záznam za konkrétní název UC.

Tabulka 3. Scénář – zobrazení všech záznamů

Název scénáře	Zobrazení všech záznamů
Aktéři	Manažer, admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli manažera nebo admina Uživatel musí mít vytvořený alespoň jeden záznam

Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko zobrazit mé záznamy 2. Systém zobrazí tabulku se všemi záznamy
Alternativní tok	2a. Uživatel nemá vytvořený žádný záznam

Tabulka 4. Scénář – přidání záznamu

Název scénáře	Přidání záznamu
Aktéři	Manažer, admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli manažera nebo admina
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko přidat záznam 2. Systém zobrazí formulář pro vytvoření záznamu 3. Uživatel vyplní formulář a odešle ho 4. Systém zkontroluje správnost dat 5. Systém zaznamená data do databáze
Alternativní tok	4a. Data nebyla správně zadána
Podmínky pro dokončení	Zobrazení dialogu o úspěšném přidání záznamu

Tabulka 5. Scénář – editace záznamu

Název scénáře	Editace záznamu
Aktéři	Manažer, admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli manažera nebo admina
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko editovat záznam 2. Systém zobrazí formulář s předvyplněnými daty z databáze 3. Uživatel vyplní formulář a odešle ho 4. Systém zkontroluje správnost dat

	5. Systém upraví data v databázi
Alternativní tok	4a. Data nebyla správně zadána 4b. Povinná data byla odstraněna
Podmínky pro dokončení	Zobrazení dialogu o úspěšné změně dat

Tabulka 6. Scénář – smazání záznamu

Název scénáře	Smazání záznamu
Aktéři	Manažer, admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli manažera nebo admina
Základní tok	1. UC začíná po kliknutí na tlačítko smazat záznam 2. Systém zobrazí dialog s dotazem, zda má být záznam opravdu smazán 3. Uživatel potvrdí smazání 4. Systém vymaže záznam a jeho data z databáze
Alternativní tok	2a. Uživatel zruší smazání
Podmínky pro dokončení	Zobrazení dialogu o úspěšném odstranění záznamu

Tabulka 7. Scénář – zobrazení detailu záznamu

Název scénáře	Zobrazení detailu záznamu
Aktéři	Manažer, admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli manažera nebo admina
Základní tok	1. UC začíná po kliknutí na odkaz záznamu 2. Systém zobrazí uživateli stránku s detaily o zvoleném záznamu

4.3.4 Zobrazení registrovaných uživatelů

Zobrazení tabulky registrovaných uživatelů má k dispozici pouze manažer. Tato funkce ukazuje přehled všech uživatelů v systému. Manažer nemá práva na jakoukoliv úpravu nebo smazání těchto záznamů na rozdíl od admina.

Tabulka 8. Scénář – zobrazení registrovaných uživatelů

Název scénáře	Zobrazení registrovaných uživatelů
Aktéři	Manažer
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli manažera
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko zobrazit uživatele 2. Systém zobrazí tabulku se všemi uživateli

4.3.5 Zobrazení statistik

Zobrazení statistik je součástí dashboardu, kde dochází k vyobrazení dat z databáze jako počet uživatelů nebo počet vytvořených tréninků.

Tabulka 9. Scénář – zobrazení statistik

Název scénáře	Zobrazení statistik
Aktéři	Manažer, admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli manažera nebo admina
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po přihlášení uživatele 2. Systém zobrazí statistiku z databáze a tabulky

4.3.6 Scénáře pro zobrazení filtrace

Uživatelé budou moci využívat filtraci dat. Filtraci lze využívat za pomoci tlačítek.

Tabulka 10. Scénář – zobrazení filtrace

Název scénáře	Zobrazení filtrace
Aktéři	Manažer, admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli manažera nebo admina
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po přihlášení do systému 2. Systém zobrazí možnosti filtrů a vstup pro vyhledávání 3. Uživatel zvolí filtr nebo zadá text k vyhledání 4. Uživatel potvrdí filtr 5. Systém zobrazí vyfiltrovaná data
Alternativní tok	4a. Uživatel zadá text který není v databázi

4.3.7 Scénáře pro zobrazení uživatelů a jejich rolí

Uživatel s rolí admina má vlastnost zobrazení všech registrovaných uživatelů včetně jejich rolí, detailů a operací. Základní operací je vytváření nového uživatele. Navíc admin může uživatele editovat, zobrazovat a mazat. Mezi pole, která jsou vyžadována pro vytvoření účtu, spadá jméno, příjmení, email, uživatelské jméno, heslo a potvrzující heslo.

Tabulka 11. Scénář – zobrazení uživatelů a rolí

Název scénáře	Zobrazení uživatelů a rolí
Aktéři	Admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli admina
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko zobrazit uživatele 2. Systém zobrazí tabulku se všemi uživateli včetně rolí a možných operací

Tabulka 12. Scénář – vytvoření uživatele

Název scénáře	Přidání uživatele
Aktéři	Admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli admina
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko přidat uživatele 2. Systém zobrazí formulář pro vytvoření uživatele 3. Uživatel vyplní formulář a odešle ho 4. Systém zkontroluje správnost dat 5. Systém zaznamená data do databáze
Alternativní tok	4a. Data nebyla správně zadána
Podmínky pro dokončení	Zobrazení dialogu o úspěšném přidání záznamu

Tabulka 13. Scénář – editace uživatelů

Název scénáře	Editace uživatele
Aktéři	Admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli admina
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko editovat uživatele 2. Systém zobrazí formulář s předvyplněnými daty o uživateli z databáze 3. Uživatel vyplní formulář a odešle ho 4. Systém zkontroluje správnost dat 5. Systém upraví data v databázi
Alternativní tok	<p>4a. Data nebyla správně zadána</p> <p>4b. Povinná data byla odstraněna</p>
Podmínky pro dokončení	Zobrazení dialogu o úspěšné změně dat cizího uživatele

Tabulka 14. Scénář – smazání uživatele

Název scénáře	Smazání uživatele
Aktéři	Admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli admina
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko smazat uživatele 2. Systém zobrazí dialog s dotazem, zda má být uživatel smazán 3. Admin potvrdí smazání cizího uživatele 4. Systém vymaže cizího uživatele včetně jeho tréninků z databáze
Alternativní tok	2a. Admin zruší smazání cizího uživatele
Podmínky pro dokončení	Zobrazení dialogu o úspěšném odstranění cizího uživatele

Tabulka 15. Scénář – zobrazení uživatele

Název scénáře	Zobrazení uživatele
Aktéři	Admin
Podmínky pro spuštění	Uživatel musí být přihlášený Uživatel musí mít roli admina
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na odkaz s cizím uživatelem 2. Systém zobrazí adminovi stránku s detaily o zvoleném cizím uživateli

4.4 Navržení UI

Při navrhování aplikace bylo hlavním cílem vytvoření jednoduchého UI. Pomocí wireframů bylo navrženo rozložení jednotlivých komponent webových stránek.

První návrh se týká přihlašovacího formuláře. Prioritou návrhu je, aby byla pozornost uživatele zaměřena pouze na formulář, a ne na jiné prvky. Proto je na wireframu pouze název aplikace a samotný formulář.

The wireframe shows a central login form titled "FitnessApp". The form contains the following elements:

- A label "Nickname/email" above a text input field.
- A label "Password" above a text input field.
- A link "You don't have an account - register" below the password field.
- A "Sign in" button at the bottom of the form.

Obrázek 2. Wireframe – formulář pro přihlášení

Základem každého wireframu, poté co se uživatel přihlásí do systému, je postranní menu složené z jednotlivých odkazů na webové stránky, profilu uživatele a uživatelského jména. Jednou z komponent návrhu dashboardu je i přehled statistických údajů, které budou převzaty z databáze.

The wireframe shows a dashboard layout for "FitnessApp". It features a sidebar menu on the left and a main content area with three statistics.

FitnessApp

- Menu 1
- Menu 2
- Menu 3
- Menu 4
- Menu 5
- Menu 6
- Menu 7

Statistics:

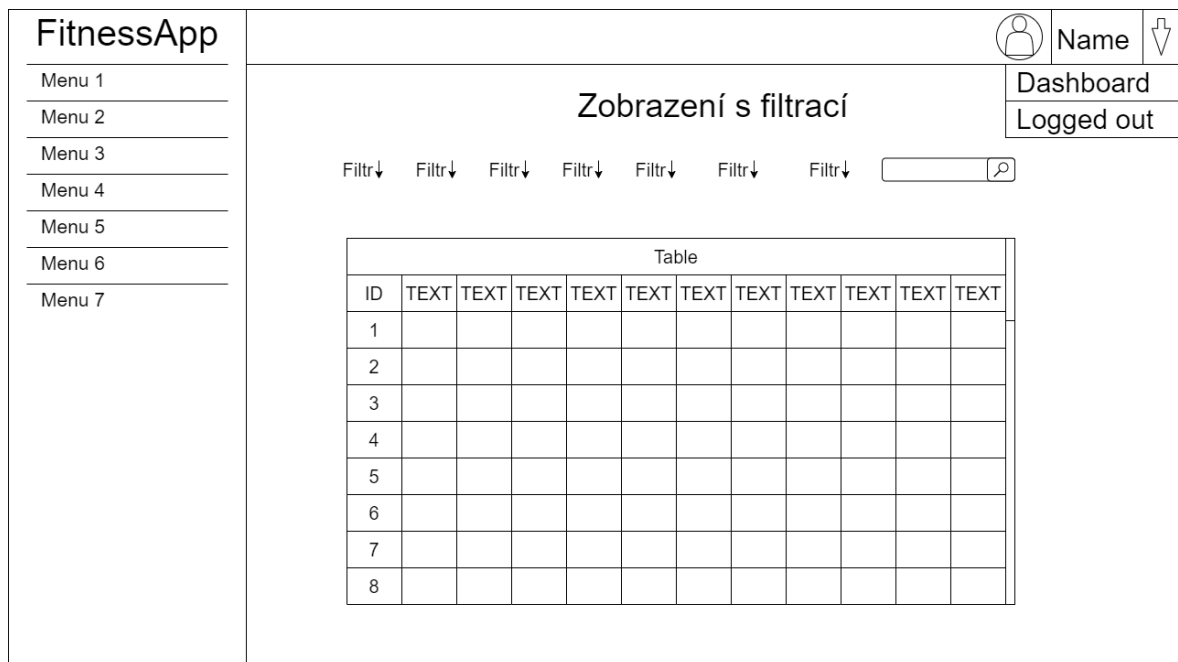
Statistic	Value
Statistic 1	15
Statistic 2	257
Statistic 3	38

User Profile:

- Name
- Dashboard
- Logged out

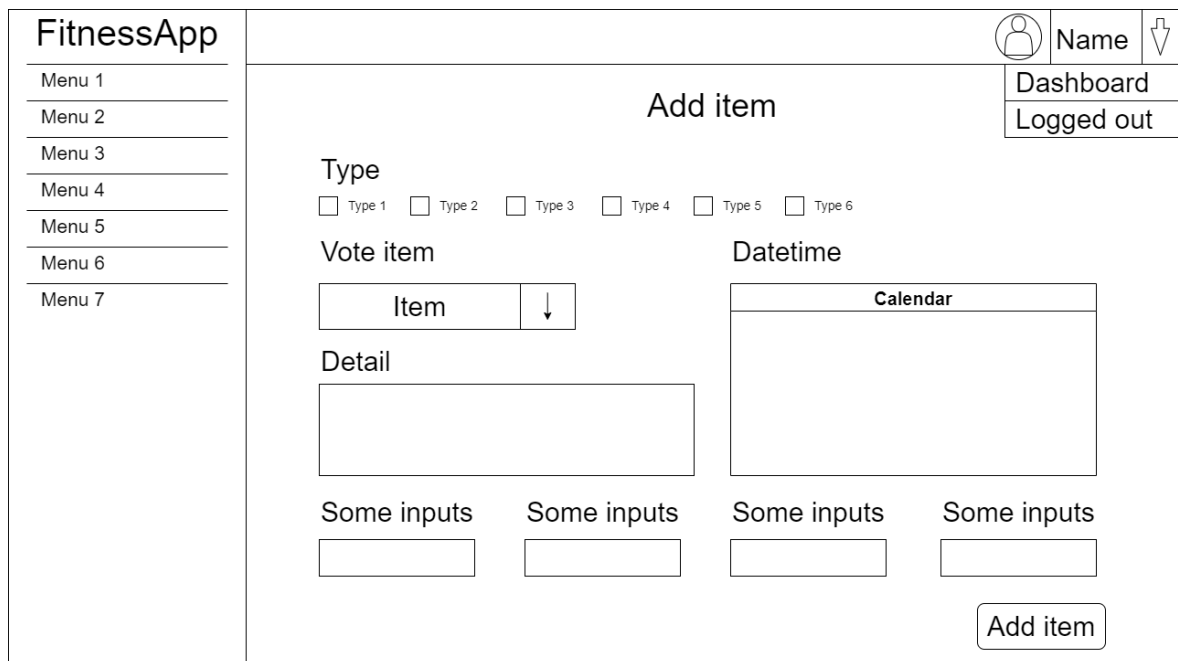
Obrázek 3. Wireframe – základní dashboard se statistikou

Čtvrtý návrh je zaměřený na zachycení zobrazení a jeho filtrace. Tyto prvky jsou součástí každé stránky webové aplikace, která využívá CRUD operace.



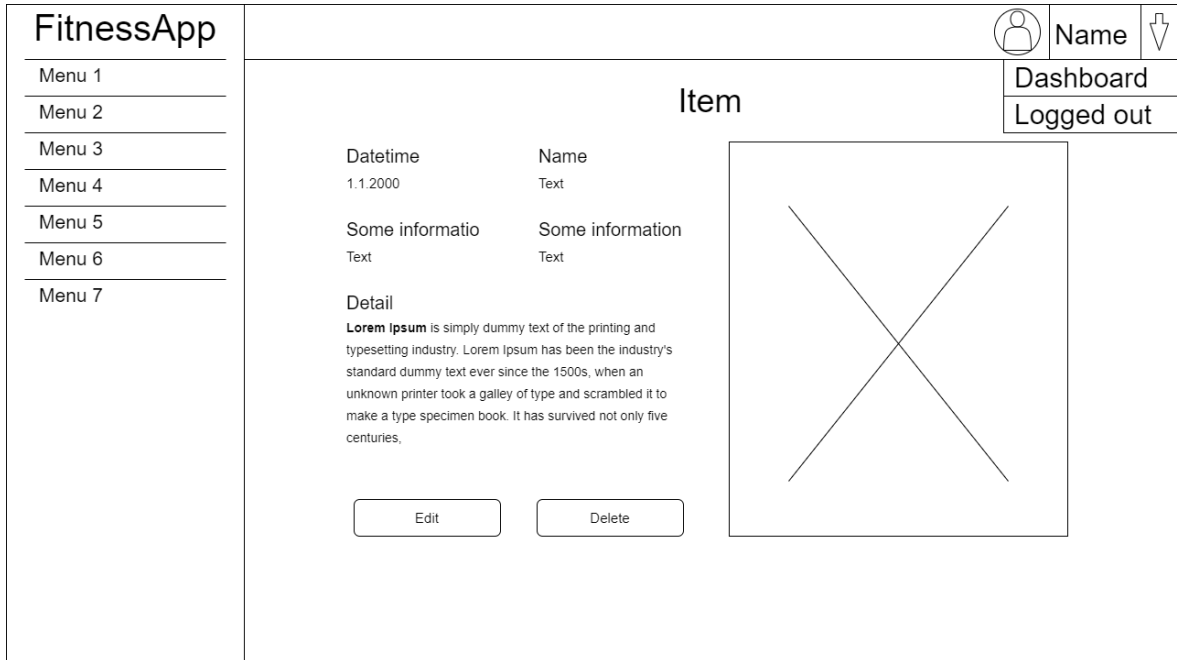
Obrázek 4. Wireframe – zobrazení s filtracemi

Návrh formuláře pro přidání prvku do databáze. Snaha o zachycení jednoduchosti a přehlednosti obecného formuláře. Formulář a jeho vstupy se v závislosti na typu UC mění.



Obrázek 5. Wireframe – formulář pro přidání prvku

Poslední wireframe zachycuje detail samotného prvku. Prioritou je zobrazení veškerých detailů, které nejsou zachyceny v základní tabulce. Důležitou součástí jsou také tlačítka pro editaci a smazání prvku.



Obrázek 6. Wireframe – zobrazení detailu o prvku

5 NÁVRH MOBILNÍ APLIKACE

Postup návrhu mobilní aplikace je složený ze stejných kroků jako návrh webové aplikace. Jejich hlavním rozdílem je rozdělení do dvou uživatelských rolí pro neregistrované a běžné uživatele.

5.1 Sběr a analýza požadavků mobilní aplikace

Pro běžné uživatele je vytvořena mobilní část aplikace. V aplikaci se nerozlišuje role administrátora, manažera, ale pouze neregistrovaného a registrovaného uživatele. Neregistrovaní uživatelé mají možnost vyzkoušet si aplikaci se základní funkcionalitou. Běžný registrovaný uživatel má k dispozici daleko více funkcí, jako ukládání svých tréninků, zobrazení svých statistik a ukládání oblíbených tréninků jiných uživatelů.

5.1.1 Neregistrovaný uživatel

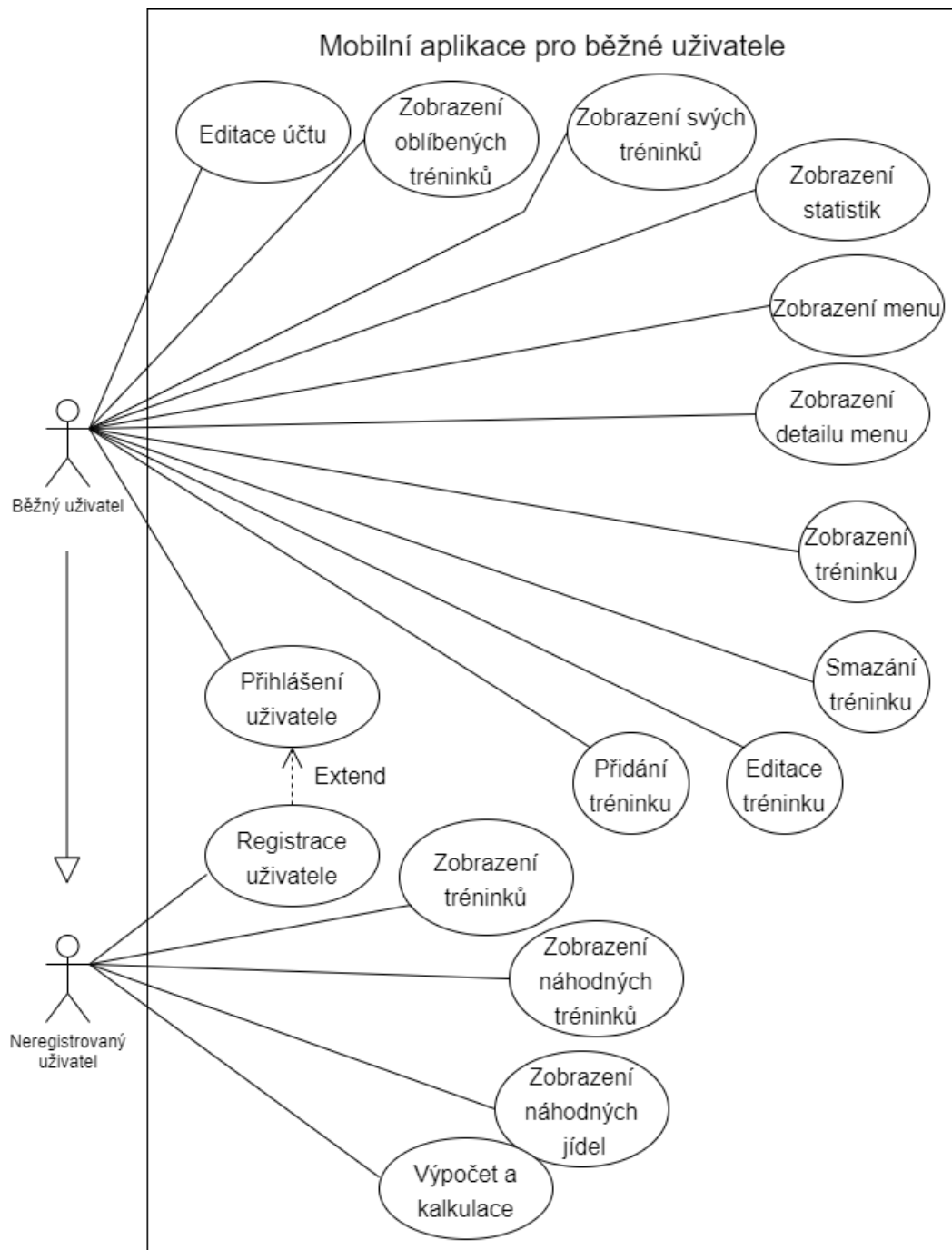
- Má možnost registrace a přihlášení do systému
- Zobrazení náhodných jídel
- Zobrazení náhodných tréninků
- Zobrazení všech tréninků
- Kalkulace BMI, ideální váhy

5.1.2 Registrovaný uživatel

- Může provádět vše co neregistrovaný uživatel
- Má možnost zobrazení, přidání, editace a mazání svých vlastních tréninků
- Má možnost zobrazení menu z jídel a jejich detailů
- Může editovat svoje profilové údaje
- Zobrazení oblíbených a svých tréninků
- Zobrazení osobních statistik

5.2 UC diagram mobilní aplikace

Mobilní aplikace obsahuje dva aktéry. Mezi tyto aktéry spadá neregistrovaný a běžný uživatel. UC diagram je zobrazen níže.



Obrázek 7. UC diagram mobilní aplikace

5.3 Scénáře

Scénáře nám zajišťují propojení systému s aktéry. V případě mobilních aplikací není popsán scénář přihlášení uživatele z důvodu, že tento scénář je totožný s webovou aplikací.

5.3.1 Registrace uživatele

Každý uživatel, kterých má zájem o využívání mobilní aplikace, musí být zaregistrovaný a uložený v databázi. V případě, že uživatel není zaregistrovaný, bude mít možnost registrace pomocí formuláře složeného z inputů v podobě křestního jména, příjmení, přezdívky, emailu, hesla a potvrzovacího hesla. U mobilní aplikace není vyžadována nutnost registrace, ale bez registrace není možné využívat veškeré funkce.

Tabulka 16. Scénář – registrace uživatele

Název scénáře	Registrace uživatele
Aktéři	Neregistrovaný uživatel
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko zaregistrovat se od uživatele 2. Systém zobrazí uživateli formulář pro registraci 3. Uživatel vyplní formulář 4. Uživatel odešle formulář 5. Systém zkontroluje správnost dat 6. Systém zaznamená data do databáze
Alternativní tok	5a. Data byla zadána špatně
Podmínky pro dokončení	Zobrazení dialogu o úspěšné registraci uživatele

5.3.2 Výpočet a kalkulace

Jedná se o UC, sloužící k získání hodnot BMI ideální váhy na základě vložených dat do formuláře. Po zadání dat dojde k přepočítání pomocí vzorce a aplikace vrátí výslednou hodnotu.

Tabulka 17. Scénář – výpočet a kalkulace

Název scénáře	Výpočet a kalkulace
Aktéři	Neregistrovaný uživatel
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko kalkulačky 2. Systém zobrazí uživateli formulář pro potřebná data

	<ol style="list-style-type: none"> 3. Uživatel vyplní a odešle formulář 4. Systém zkontroluje správnost dat 5. Systém vrátí výsledek
Alternativní tok	4a. Data byla zadána špatně
Podmínky pro dokončení	Zobrazení vypočtených dat

5.3.3 Zobrazení náhodných jídel

Zobrazení náhodných jídel zajišťuje vyzkoušení aplikace pro neregistrované uživatele

Tabulka 18. Scénář – zobrazení náhodných jídel

Název scénáře	Zobrazení náhodných jídel
Aktéři	Neregistrovaný uživatel
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po otevření aplikace 2. Systém vygeneruje náhodná jídla z databáze 3. Systém zobrazí náhodný seznam jídel
Alternativní tok	2a. Databáze neobsahuje žádná jídla

5.3.4 Zobrazení náhodných tréninků

UC sloužící k zobrazení náhodných tréninků cizích uživatelů. Neregistrovaný uživatel se může podívat na trénink včetně jeho detailů.

Tabulka 19. Scénář – zobrazení náhodných tréninků

Název scénáře	Zobrazení náhodných tréninků
Aktéři	Neregistrovaný uživatel
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po otevření aplikace 2. Systém vygeneruje náhodné tréninky z databáze 3. Systém zobrazí náhodné tréninky
Alternativní tok	2a. Databáze neobsahuje žádné tréninky

5.3.5 Zobrazení tréninků

Neregistrovaný uživatel má možnost prohlížení všech tréninků, které jsou vytvořeny.

Tabulka 20. Scénář – zobrazení tréninků

Název scénáře	Zobrazení tréninků
Aktéři	Neregistrovaný uživatel
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko všechny tréninky 2. Systém zobrazí všechny tréninky z databáze
Alternativní tok	2a. Databáze neobsahuje žádné tréninky

5.3.6 Zobrazení tréninků a operace s nimi

Každý registrovaný uživatel má možnost zobrazit si své vlastní tréninky. Tyto tréninky může poté libovolně přidávat, upravovat nebo mazat. Uživatel má také možnost zobrazení detailu po kliknutí na konkrétní trénink.

Tabulka 21. Scénář – zobrazení svých tréninků

Název scénáře	Zobrazení svých tréninků
Aktéři	Běžný uživatel
Podmínky pro spuštění	<p>Uživatel musí být přihlášený</p> <p>Uživatel musí mít vytvořený alespoň jeden záznam</p>
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko zobrazit mé tréninky 2. Systém zobrazí tabulku s tréninky uživatele
Alternativní tok	2a. Uživatel nemá vytvořený žádný trénink

Tabulka 22. Scénář – přidání tréninku

Název scénáře	Přidání tréninku
Aktéři	Běžný uživatel

Podmínky pro spuštění	Uživatel musí být přihlášený
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko přidat trénink 2. Systém zobrazí formulář pro vytvoření tréninku 3. Uživatel vyplní formulář a odešle ho 4. Systém zkontroluje správnost dat 5. Systém zaznamená data do databáze
Alternativní tok	4a. Data nebyla správně zadána
Podmínky pro dokončení	Zobrazení dialogu o úspěšném přidání tréninku

Tabulka 23. Scénář – editace tréninku

Název scénáře	Editace tréninku
Aktéři	Běžný uživatel
Podmínky pro spuštění	Uživatel musí být přihlášený
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko editovat trénink 2. Systém zobrazí formulář s předvyplněnými daty z databáze 3. Uživatel vyplní formulář a odešle ho 4. Systém zkontroluje správnost dat 5. Systém upraví data v databázi
Alternativní tok	<p>4a. Data nebyla správně zadána</p> <p>4b. Povinná data byla odstraněna</p>
Podmínky pro dokončení	Zobrazení dialogu o úspěšné změně tréninku

Tabulka 24. Scénář – smazání tréninku

Název scénáře	Smazání tréninku
Aktéři	Běžný uživatel
Podmínky pro spuštění	Uživatel musí být přihlášený

Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko smazat trénink 2. Systém zobrazí dialog s dotazem, zda má být trénink opravdu smazán 3. Uživatel potvrdí smazání 4. Systém vymaže trénink a jeho data z databáze
Alternativní tok	2a. Uživatel zruší smazání
Podmínky pro dokončení	Zobrazení dialogu o úspěšném odstranění tréninku

Tabulka 25. Scénář – zobrazení detailu tréninku

Název scénáře	Zobrazení detailu tréninku
Aktéři	Běžný uživatel
Podmínky pro spuštění	Uživatel musí být přihlášený
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na odkaz s tréninkem 2. Systém zobrazí uživateli stránku s detaily o zvoleném tréninku

5.3.7 Zobrazení menu

Uživatel, který je přihlášený do aplikace, má možnost zobrazení jídelníčku. Systém uživateli zobrazí jídla z databáze.

Tabulka 26. Scénář – zobrazení menu

Název scénáře	Zobrazení menu
Aktéři	Běžný uživatel
Podmínky pro spuštění	Uživatel musí být přihlášený
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko zobrazit jídla 2. Systém zobrazí uživateli jídla
Alternativní tok	4a. Databáze nemá žádná jídla
Podmínky pro dokončení	Zobrazení jídelníčku

Tabulka 27. Scénář – zobrazení detailu jídla

Název scénáře	Zobrazení detailu jídla
Aktéři	Běžný uživatel
Podmínky pro spuštění	Uživatel musí být přihlášený
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na jídlo 2. Systém zobrazí uživateli detaily o jídle

5.3.8 Zobrazení osobních statistik

Každý přihlášený uživatel má vedené statistiky. Jedná se o přehled nejdůležitějších informací, které se dají kdykoliv zobrazit jako počet tréninků za poslední týden.

Tabulka 28. Scénář – zobrazení osobních statistik

Název scénáře	Zobrazení osobních statistik
Aktéři	Běžný uživatel
Podmínky pro spuštění	Uživatel musí být přihlášený
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko zobrazit statistiky 2. Systém zobrazí osobní statistiky
Alternativní tok	2a. Uživatel má prázdnou databázi tréninků
Podmínky pro dokončení	Zobrazení statistik

5.3.9 Zobrazení oblíbených tréninků

Jakýkoliv trénink, který je vytvořený lze ohodnotit tlačítkem, které zastupuje oblíbenost. Tyto oblíbené tréninky si poté může přihlášený uživatel zpětně dohledat.

Tabulka 29. Scénář – zobrazení oblíbených tréninků

Název scénáře	Zobrazení oblíbených tréninků
Aktéři	Běžný uživatel
Podmínky pro spuštění	Uživatel musí být přihlášený

Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko zobrazit oblíbené tréninky 2. Systém zobrazí oblíbené tréninky
Alternativní tok	2a. Uživatel nemá žádný oblíbený trénink

5.3.10 Editace účtu

Přihlášení uživatelé mají možnost editace účtu. Editace je prováděna pomocí formuláře, který načítá data z databáze.

Tabulka 30. Scénář – editace účtu

Název scénáře	Editace účtu
Aktéři	Běžný uživatel
Podmínky pro spuštění	Uživatel musí být přihlášený
Základní tok	<ol style="list-style-type: none"> 1. UC začíná po kliknutí na tlačítko editovat účet 2. Systém zobrazí uživateli formulář s vlastními předvyplněnými daty 3. Uživatel upraví vstupy ve formuláři a odešle jej 4. Systém zkontroluje správnost dat 5. Systém upraví údaje o uživateli v databázi
Alternativní tok	<p>5a. Data byla zadána špatně</p> <p>5b. Povinná data byla odstraněna</p>
Podmínky pro dokončení	Zobrazení dialogu o úspěšné změně dat uživatele

5.4 Navržení mobilního UI

Cílem navržení mobilního UI je zaujmout uživatele už při prvním otevření aplikace. Základní rozložení aplikace, včetně jeho stránek, je tvořeno tak, aby bylo snadné se v něm orientovat a vše pochopit během chvíle.

Součástí hlavní stránky, po prvním otevření aplikace bez registrace, je ukázka náhodných tréninků, náhodných jídel, ale především velké tlačítko, které vyzývá k přihlášení.

U přihlášeného uživatele jsou navíc zobrazena data včetně detailu o posledním tréninku, a také statistika.

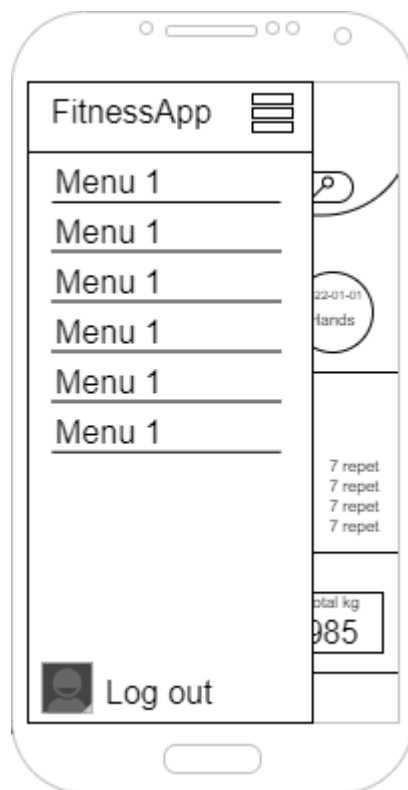


Obrázek 8. Wireframe – hlavní stránka bez registrace



Obrázek 9. Wireframe – hlavní stránka s přihlášením

V levém horním rohu aplikace se vždy nachází tlačítko, které obstarává zobrazení navigačního menu. Jehož součástí je i následné odhlášení a možnost editace profilu.



Obrázek 10. Wireframe – menu

Poslední wireframe, který je vytvořený se týká formuláře pro přidávání tréninku. Tento formulář je složený z několika inputů, tlačítka pro přidání dalšího cviku a tlačítka pro přidání celého tréninku do databáze.

The wireframe shows a mobile application interface for adding a training session. At the top, there is a header with a hamburger menu icon, the title "FitnessApp", and the subtitle "Add training". Below the header is a section titled "Body parts" containing six checkboxes, each labeled "Hands". Underneath is a dropdown menu with "Biceps push" selected. There are three input fields: "weight" with the value "25 kg", "repetition" with the value "5", and "series" with the value "8". A circular button with a "+" sign is located below the input fields. At the bottom of the screen is a large rounded button labeled "Add training".

Obrázek 11. Wireframe – přidání tréninku

6 NAPROGRAMOVÁNÍ APLIKACÍ A JEJICH POPIS

Vývoj aplikací byl rozdělen do tří fází. V první fázi došlo k vytvoření serverové části, která zajišťuje především práci s databází. Tato část byla programována v prostředí Node.js s využitím frameworku Express.js a databáze MongoDB. Následně byla vytvořena webová aplikace a její design pomocí knihovny React. Poslední fází bylo vytvoření mobilní aplikace prostřednictvím React Native.

6.1 Server a tvorba API

Součástí serverové části je oddělení logických celků. Mezi tyto celky spadá modely, controllery, middleware, routy a utils. V utils jsou pouze pomocné metody zajišťující vrácení error hlášky a odeslání emailu při obnově hesla. Hlavní částí, která zajišťuje spuštění aplikace, je soubor server.js. Díky němu dochází k připojení databáze, vytvoření API pro jednotlivé routy a nastavení portu serveru.

6.1.1 Vytvoření modelů

Aplikace využívá pět základních modelů. Pro vytváření modelových schémat je využíván balíček mongoose.

První schéma nese název User a obsahuje v sobě atributy pro uživatelská data. Mezi povinné atributy při vytváření spadá křestní jméno, příjmení, uživatelské jméno, email, heslo a role. Do nepovinných byly zařazeny váha, výška a den vytvoření, který je automaticky nastavený na aktuální čas. Při vytváření záznamu je automaticky generováno ID. Před uložením entity do databáze dochází k vytvoření hashe hesla. Pro tyto účely byl vložen balíček bcryptjs. Atribut uživatelské jméno i email musí být unikátní, aby nedocházelo ke kolizím při přihlášení, obnově hesel a ukládání tréninků. Součástí modelu jsou i přidané metody pro porovnání správnosti hesel, získání přihlašovacího tokenu a vytvoření tokenu pro obnovení hesla.

```
_id: ObjectId("62587249920f5bf1684e411c")
firstname: "Manazer"
lastname: "User"
username: "Manazer"
email: "manazer@randomgmail.com"
password: "$2a$10$38hfnFJEg0fraD5C00UJo.0RTKkXli1z1kPwYIcIbKf8.FWzJ0tD5"
weight: 185
width: 100
role: "manager"
created: 2022-04-14T19:13:13.295+00:00
__v: 0
```

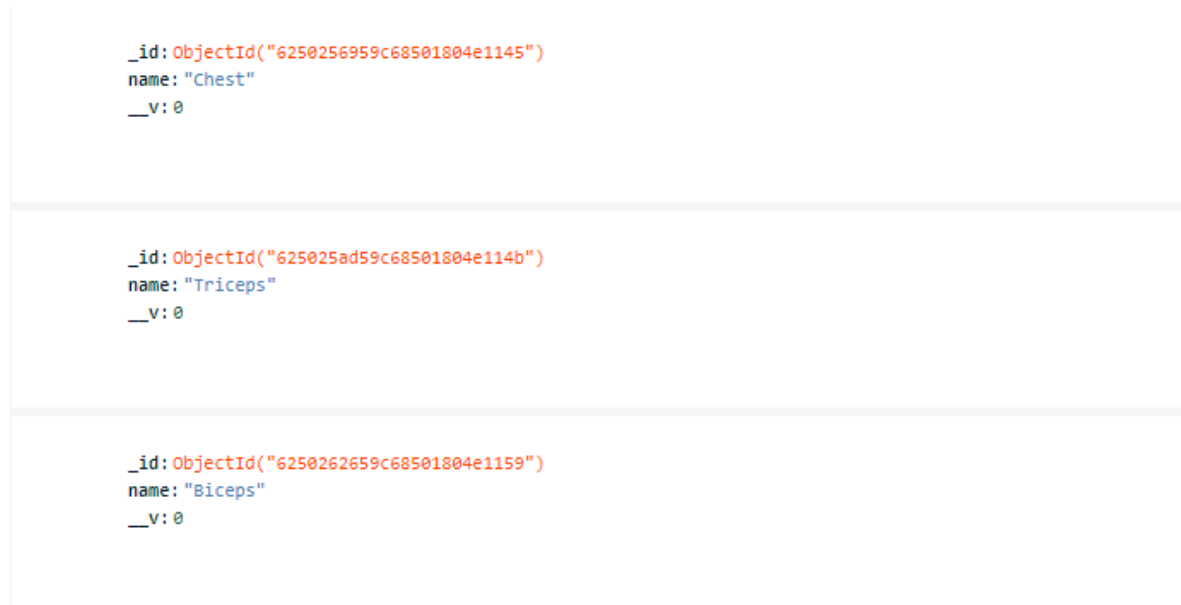
Obrázek 12. Entita uživatele v databázi

Meal je schéma sloužící pro zaznamenávání jídel s recepty. Jeho součástí jsou atributy jméno, část dne, kalorie, bílkoviny, cukry, tuky, recept, čas vytvoření a ingredience. Ingredience jsou navíc složeny z názvu, počtu a jednotky. Všechny tyto atributy jsou povinné.

```
>
_id: ObjectId("625845beb6449dc1a5dac60a")
name: "Steak"
day_part: "Lunch"
calories: 220
protein: 50
sugar: 6
fat: 20
recept: "Steak on grill 5 min"
ingredients: Array
  0: Object
    name: "Meal"
    amount: 200
    unit: "g"
    _id: ObjectId("6275a4f148984d8fb496de9d")
  1: Object
    name: "Peper"
    amount: 5
    unit: "g"
    _id: ObjectId("6275a4f148984d8fb496de9e")
  2: Object
  3: Object
created: 2022-04-14T16:03:10.402+00:00
__v: 6
```

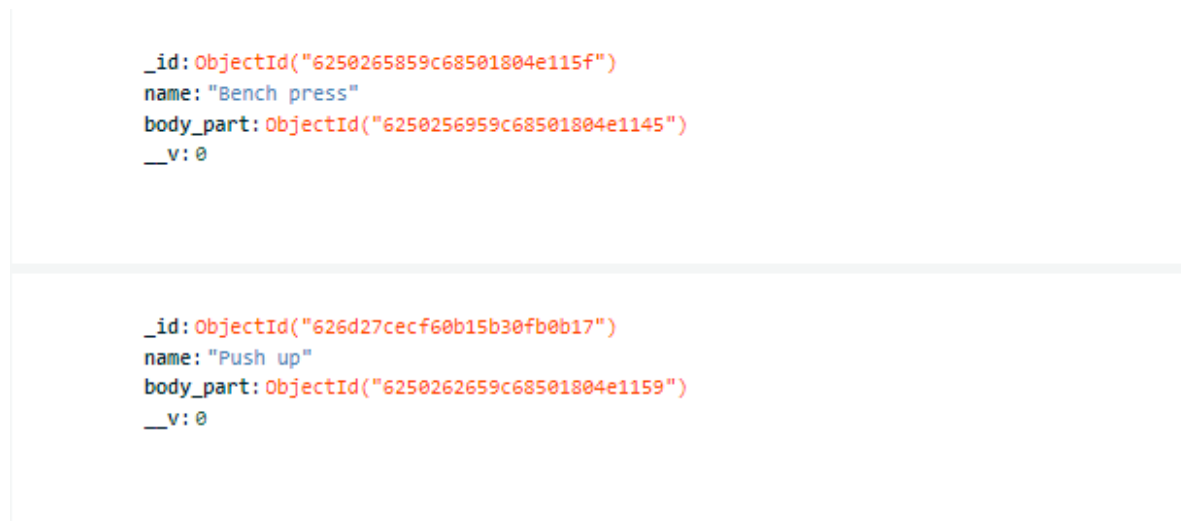
Obrázek 13. Entita jídla v databázi

Poslední tři schémata spolu souvisí. Jedná se o BodyPart, Workout a Training. BodyPart obsahuje pouze jeden atribut, kterým je název části těla. Tento atribut je povinný a musí být unikátní.



Obrázek 14. Entita části těla v databázi

Navazující entita je Workout, skládající se z ID BodyPart a svého názvu. Tato entita nám zobrazuje konkrétní cvik.



Obrázek 15. Entita cviku v databázi

Nejčastěji využívaným schématem celé aplikace je Training. Jedna entita zobrazuje jeden konkrétní trénink. Je složený z povinných atributů, mezi které spadá název tréninku a uživatelské jméno. Mezi nepovinné atributy byly zařazeny detail a čas vytvoření, který se generuje automaticky. Model také obsahuje pole cviků a pole uživatelů, kteří si trénink oblíbili. Každý prvek pole tréninků je složen z Workout ID, počtu sérií, opakování a váhy. Pole s oblíbenými uživateli je složeno z uživatelského jména.

```
  _id: ObjectId("62725d1a2e7a67b7318408fc")
  name: "My chest training"
  workouts: Array
    0: Object
      workout: ObjectId("626d27cecf60b15b30fb0b17")
      series: 5
      repetition: 8
      weight: 10
      _id: ObjectId("627596e116f50f24a22f9a7d")
    1: Object
      workout: ObjectId("6250265859c68501804e115f")
      series: 10
      repetition: 5
      weight: 20
      _id: ObjectId("627596e116f50f24a22f9a7e")
    2: Object
    3: Object
  detail: "Very good trainings for amaters..."
  username: "Zdenek123"
  created: 2022-05-04T11:01:46.310+00:00
  likes: Array
    0: Object
      username: "Zdenek123"
      _id: ObjectId("627596ff16f50f24a22f9a99")
  __v: 5
```

Obrázek 16. Entita tréninku v databázi

6.1.2 Controllery

Aplikace je složená ze sedmi controllerů. Jejich účelem je reagování aplikace na vstupy od uživatelů při snaze přístoupení k API.

6.1.2.1 Auth controller

Auth controller zajišťuje autentizace aplikací. Jeho součástí jsou metody register, login, forgotpassword a resetpassword.

Metoda pro registraci uživatele na začátku získá z uživatelského požadavku povinná data a doplní je o nastavení základní výšky, váhy a roli běžného uživatele. Pokud jsou veškerá data zadána a jsou validní dojde k vytvoření nové entity uživatele v databázi, vytvoření

uživatelského tokenu pomocí JWT a zaslání zprávy o úspěšném provedení registrace. V situaci, kdy by se uživatel pokusil vytvořit účet, který by neměl unikátní email nebo uživatelské jméno, dojde k přerušení registrace a zaslání odpovědi ze serveru, o nutnosti unikátní hodnoty atributu.

Přihlášení je metoda, která z uživatelského požadavku získá email a heslo. V případě, že jsou data vyplněny dojde k vyhledání uživatele v databázi a porovnání správnosti hesla. Pokud dojde ke shodě je vytvořen JWT token a zaslána odpověď o úspěšném přihlášení. Při situaci, kdy některé ověření není platné dojde k vrácení chybové hlášky.

Metoda pro zapomenutí hesla získá z požadavku email uživatele. Následuje zjištění, zda je uživatel v databázi a pokud ano dojde k vygenerování tokenu pro obnovu hesla a zaslání odkazu pro reset hesla na příslušný email. Po úspěšném odeslání je vrácena odpověď, že byl email odeslán.

Poslední metoda je obnovení hesla, která je provedena po validním odeslání požadavku od uživatele, kdy pomocí obnovovacího tokenu je nalezen uživatel a následně změněno jeho heslo. Současně dojde k odstranění tokenu pro obnovení hesla, uložení entity a vrácení odpovědi v podobě nově vygenerovaného JWT tokenu.

6.1.2.2 *Private controller*

Private controller je složený z jedné metody, sloužící k vrácení odpovědi v případě, kdy má uživatel právo na přístup do určité části aplikace.

6.1.2.3 *User, Bodypart, Workout, Training, Meal controller*

Jedná se o controllery sloužící pro práci s CRUD operacemi. Každý z nich obsahuje metody, jako získání seznamu, získání detailu, vytvoření, editaci nebo smazání entity. Controller Training je navíc rozšířený o metody pro přidání mezi oblíbené tréninky a odebrání z oblíbených tréninků.

Metody `getUserList`, `getBodyPartList`, `getWorkoutList`, `getTrainingList` a `getMealList` zajišťuje získání záznamu všech entit daného dokumentu v databázi. V případě, že dokument neobsahuje žádnou entitu je vrácena odpověď, že neexistuje žádný záznam v databázi.

`CreateUser`, `createBodyPart`, `createWorkout`, `createTraining` a `createMeal` jsou metody sloužící pro vytvoření nové entity v databázi. Controller získá data z requestu a pokud jsou

všechna data zadána validně dojde k vytvoření nového záznamu v patřičném dokumentu databáze. Po vytvoření je odeslána odpověď o vytvoření záznamu.

Další metody obstarávají získání konkrétní entity databáze pomocí hodnoty ID. Jestliže je v databázi nalezen záznam, controller ho vrátí společně s úspěšnou odpovědí. Mezi tyto metody spadá `getUser`, `getBodyPart`, `getWorkout`, `getTraining` a `getMeal`.

V případě, kdy chceme upravovat již vytvořenou entitu musíme použít metody `updateUser`, `updateBodyPart`, `updateWorkout`, `updateTraining` nebo `updateMeal`. Na začátku metody dojde k nalezení entity pomocí ID a kontroly, zda skutečně existuje. Poté atributy entity zaměněny za hodnoty získané z požadavku a dojde k uložení záznamu. Jakmile dojde k úspěšném uložení je odeslána odpověď o úspěšné editaci záznamu.

Posledními metodami, z kategorie běžných CRUD operací, jsou metody pro smazání záznamu z databáze. V případě metod `deleteMeal` a `deleteTraining` dochází pouze k vyhledání entity pomocí ID a jejího odstranění. U odstranění uživatele, pomocí `deleteUser`, dochází současně k odstranění všech tréninků, které si uživatel vytvořil. Poslední dvě metody jsou `deleteWorkout` a `deleteBodyPart`. Jejich součástí je omezení odstranění záznam za určitých podmínek. U `deleteWorkout` dochází ke kontrole, zda entita není obsažena v některém tréninku. Pokud ano, nesmí dojít k odstranění a je vrácena odpověď, že je součástí tréninku a nelze jej odstranit. Poslední metodou je `deleteBodyPart`, která nesmí být odstraněna, pokud je součástí libovolné entity v databázovém dokumentu `workout`.

Controller `Training` nabízí možnost přidání tréninku mezi oblíbené. Na začátku dojde k získání uživatelského jména z požadavku a ke kontrole, zda uživatel již nemá trénink mezi oblíbenými. V případě, že nemá, je přidán do pole oblíbených současně s odesláním odpovědi o úspěšném přidání. Uživatel má také možnost odebrání oblíbeného tréninku. Metoda získá z požadavku uživatelské jméno a odstraní záznam. Na závěr odešle odpověď, že byl záznam byl odstraněn z pole.

6.1.3 Middleware

Serverová část, která zajišťuje zabezpečení přístupu k určitým částím aplikace. Zabezpečení je rozděleno do tří základních metod. Jejich účelem je zajištění kontroly a oddělení uživatelských rolí. Druhou funkcí je vrácení odpovědi prostřednictvím serveru v případě neočekávaných chyb. To je zajištěno pomocí metody `errorHandler`.

Metoda `auth` obstarává zjištění, zda je uživatel přihlášený do systému. Na začátku dojde k získání autentizačního tokenu z požadavku. Následuje kontrola pomocí dekodování tokenu. Výsledkem dekodování je ID uživatele a v případě, kdy je ID součástí dokumentu databáze, dojde k úspěšnému potvrzení autentizace. V opačném případě je vrácena chybová hláška a uživateli je odepřen přístup k aplikaci.

Při používání webové aplikace musí uživatel využívat kromě autentizace i autorizaci. Ošetření autorizace nám zajišťují metody `protect` a `admin`. Obě mají podobný průběh jako metoda `auth`, s tím rozdílem, že kontrolují i uživatelské role. Metoda `protect` odepře přístup v případě, kdy uživatel není v roli manažera nebo admina. Poslední metodou je `admin`, která povoluje využití aplikace pouze administrátorovi.

6.1.4 Routes

Routy na straně serveru zajišťují možnost komunikace s API. Tyto routy jsou součástí `express.js`. V případě, kdy je daná ruta použita s konkrétní metodou, dochází také k použití určité metody `controlleru`. V některých případech je rozšířena také o autentizaci nebo autorizaci.

6.2 Popis webové aplikace

Základní rozdělení stránky je na `routing`, `pages` a `components`. Součástí `routingu` jsou veškeré možné podstránky, které lze na webu navštívit. Pro pohyb mezi stránkami byla implementována knihovna `react-router`. `Pages` a `components` poté udávají celkový vzhled stránky společně s načtenými daty. Veškerá komunikace s API je zprostředkována pomocí balíčku `axios`.

6.2.1 Přihlášení a obnova hesla

Jediné stránky, které si může neregistrovaný uživatel zobrazit, jsou přihlašovací, zapomenuté heslo a obnovení hesla. Součástí každé stránky je formulář, který odesílá data na server pomocí API. V případě chybného zadání údajů, dojde k výpisu chybové hlášky.

FitnessApp

Login

Email:

Password:

[Forgot password?](#)

Login

Obrázek 17. Webová aplikace – login

6.2.2 Neoprávněný přístup

Pro případ, kdy se přihlásí běžný uživatel nebo jde o snahu manažera bez administrátorské role dostat do administrace, byla vytvořena stránka s hláškou o nepovoleném přístupu a tlačítkem pro odhlášení.

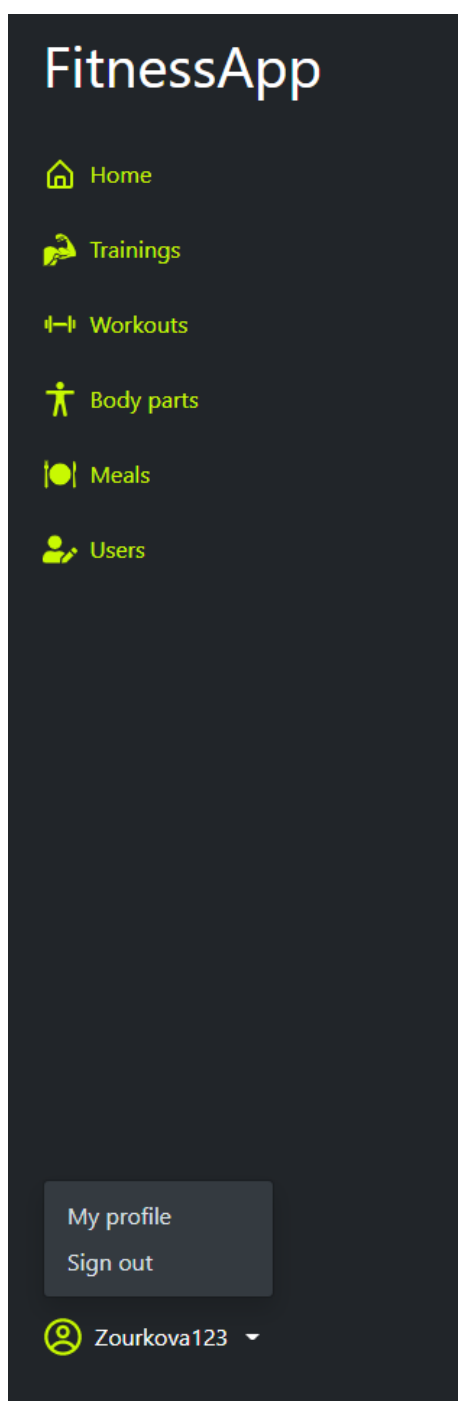
You do not have permission on this

Logout

Obrázek 18. Webová stránka – nepovolený přístup

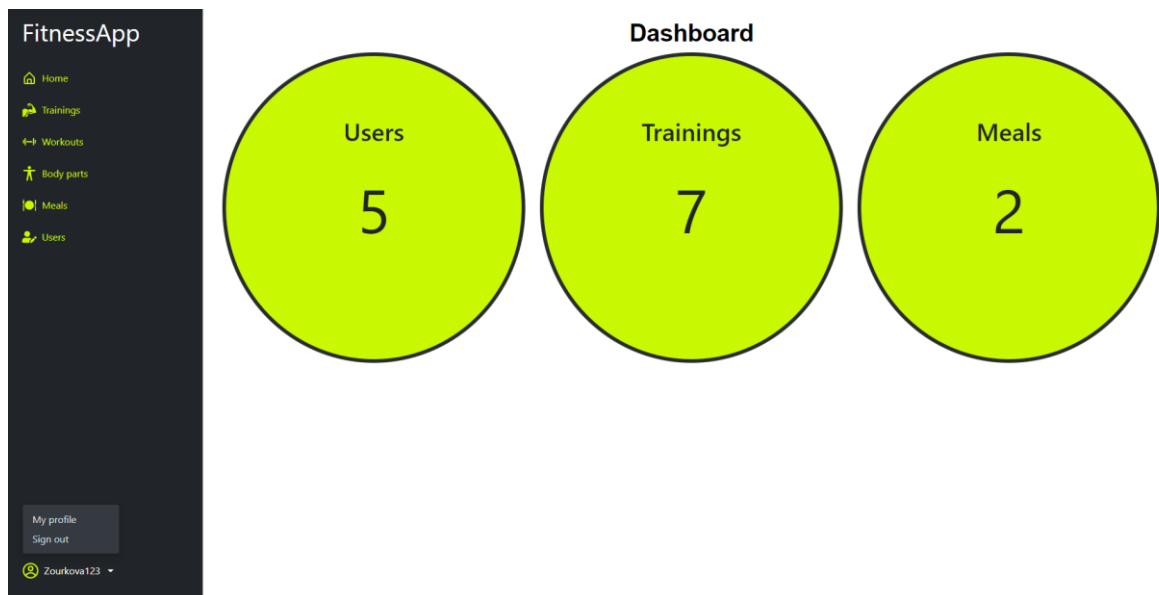
6.2.3 Dashboard a menu

Po přihlášení a úspěšné autorizaci je uživateli zobrazen dashboard a menu, které je součástí každé stránky. Menu je složené z tréninků, cviků, částí těla, jídel a uživatelů. V jeho spodní části má poté uživatel možnost odhlášení a editace svého profilu. Menu je tvořené pomocí knihovny react-bootstrap. Zobrazení ikon je zajištěno knihovnou react-icons.



Obrázek 19. Webová aplikace – detail menu

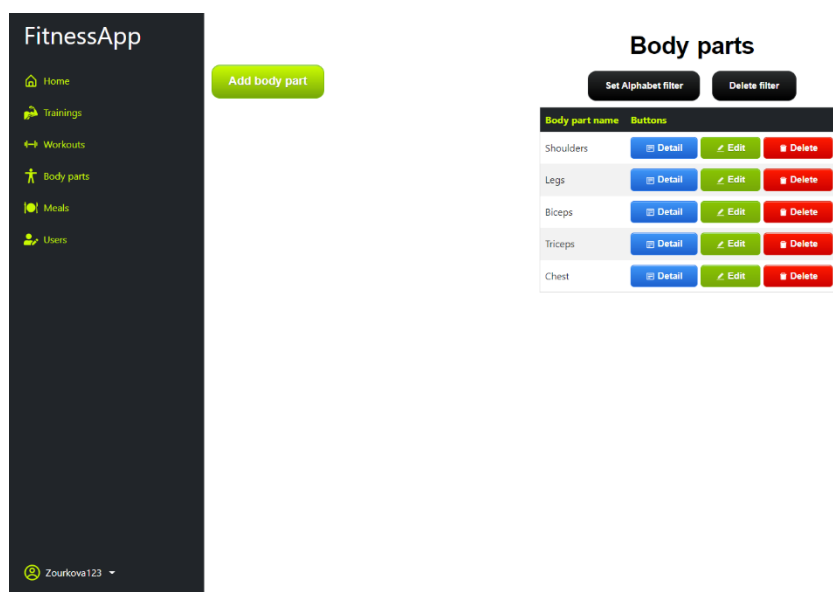
Dashboard zachycuje celkový počet uživatelů, tréninků a jídel, které jsou součástí databázových dokumentů.



Obrázek 20. Webová stránka – dashboard

6.2.4 Stránky částí těla

Při kliknutí na částí těla, v sekci menu, je zobrazena tabulka s přehledem entit uložených v dokumentu databáze. Součástí této tabulky je možnost zobrazení detailu, editace nebo smazání záznamu. V levém horním rohu je tlačítko, které umožňuje vytvořit novou entitu. Uživatel má možnost vyfiltrovat záznamy dle abecedy a také tento filtr zrušit.



Obrázek 21. Webová aplikace – částí těla

Body parts

Set Alphabet filter Delete filter

Body part name	Buttons		
Biceps	Detail	Edit	Delete
Chest	Detail	Edit	Delete
Legs	Detail	Edit	Delete
Shoulders	Detail	Edit	Delete
Triceps	Detail	Edit	Delete

Obrázek 22. Webová aplikace – detail tabulky částí těla

Jakmile uživatel klikne na tlačítko přidat trénink nebo editovat dojde k zobrazení formuláře s jedním inputem. V případě přidání tréninku je formulář bez předvyplněných dat. Uživatel má také možnost zobrazení stránky s detailem, kde se nachází název části těla společně s tlačítky pro editaci a smazání entity. Poslední tlačítko, které je součástí tabulky, zajišťuje smazání záznamu, kdy před odstraněním dojde k zobrazení potvrzujícího okna s dotazem, zda chceme část těla opravdu smazat.

Add body part form

Body part name:

Add body part

Obrázek 23. Webová aplikace – přidání části těla

6.2.5 Stránky cviků

Druhou částí menu jsou cviky. Jakmile se načte stránka se cviky, je nám zobrazena tabulka s názvy cviků a částí těla, kterých se dané cviky týkají. Zbylé prvky stránky jsou obdobné jako u částí těla.

Workouts

Set Alphabet filter Delete filter

Workout	Body part	Buttons
Leg press	Legs	Detail Edit Delete
Push up	Chest	Detail Edit Delete
Push up	Biceps	Detail Edit Delete
Bench press	Chest	Detail Edit Delete

Obrázek 24. Webová aplikace – detail tabulky cviků

Přidání a editace cviků jsou zprostředkovávány formulářem, který obsahuje jeden input pro název cviku a jeden selectbox pro volbu části těla.

Add workout form

Workout name:

Body part name:

Biceps ▼

Biceps

Chest

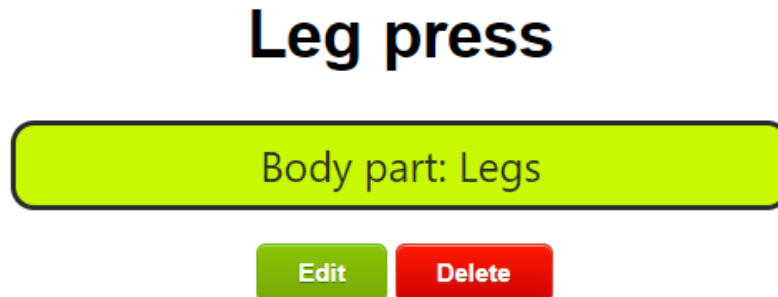
Legs

Shoulders

Triceps

Obrázek 25. Webová aplikace – přidání cviku

Detail cviku nám zobrazuje informace o názvu cviku a jeho části těla, kterou má entita uloženu. Pro smazání cviku musí být potvrzeno zobrazující okno s dotazem, zda si opravdu přejeme odstranit cvik.



Obrázek 26. Webová aplikace – detail cviku

6.2.6 Stránky pro tréninky

Stránka s tréninky obstarává zobrazení všech záznamů tréninků, které vytvořili libovolní uživatelé. Součástí každého řádku tabulky je název tréninku, uživatelské jméno autora, datum vytvoření, počet uživatelů, kterým se trénink líbil a seznam částí těla, kterých se trénink týká.

Name	Training body parts	Author	Date	Likes	Buttons
Grznar training	Biceps	Zdenek123	2022-05-04	0	Detail Edit Delete
Up body	Biceps Chest	Zdenek123	2022-05-04	0	Detail Edit Delete
Chest train	Chest	Zourkova123	2022-04-23	2	Detail Edit Delete
Chest day	Chest	Zourkova123	2022-04-21	1	Detail Edit Delete
New bench	Chest Biceps	Zourkova123	2022-04-21	0	Detail Edit Delete
Bench	Chest	Zourkova123	2022-04-21	0	Detail Edit Delete
First train	Chest	Zdenek123	2022-04-15	2	Detail Edit Delete

Obrázek 27. Webová aplikace – detail tabulky tréninků

Přidání tréninku v sobě obsahuje formulář, který je složený ze dvou hlavních inputů a tlačítka pro přidání cviku do tréninku. První input obstarává zadání názvu tréninku a druhý pro vyplnění detailu. Po stisknutí tlačítka, přidávajícího cvik, jsou zobrazeny další dva selectboxy a tři inputy. První selectbox určuje výběr části těla. Po jejím zvolení se automaticky v druhém selectboxu načte seznam cviků, které má konkrétní část těla vytvořené. Zbylé tři inputy přijímají pouze čísla a zajišťují zaznamenání počtu sérií, opakování v sérii a velikosti závaží v kilogramech. Při opětovném stisknutí tlačítka přidat cvik se navíc na konci řádku zobrazí možnost odstranění celého cviku. Tato možnost není zobrazena u prvního cviku, protože každý trénink musí obsahovat alespoň jeden cvik. Stejným způsobem funguje i editace tréninků, která nám do inputů a selectboxů již vkládá jednotlivé atributy z databáze.

Add training form

Training name:	Training detail:				
<input type="text" value="Nový trénink"/>	<input type="text" value="Enter training detail"/>				
Body part name:	Workout name:	Number of series:	Number of repetition:	Number of weight:	
<input type="text" value="Legs"/>	<input type="text" value="Leg press"/>	<input type="text" value="5"/>	<input type="text" value="5"/>	<input type="text" value="50"/>	
Body part name:	Workout name:	Number of series:	Number of repetition:	Number of weight:	<input type="button" value="Remove"/>
<input type="text" value="Biceps"/>	<input type="text" value="Push up"/>	<input type="text" value="8"/>	<input type="text" value="4"/>	<input type="text" value="10"/>	
Body part name:	Workout name:	Number of series:	Number of repetition:	Number of weight:	<input type="button" value="Remove"/>
<input type="text" value="Chest"/>	<input type="text" value="Bench press"/>	<input type="text" value="Enter number of series"/>	<input type="text" value="Enter number of repetition"/>	<input type="text" value="Enter number of weight"/>	
<input type="button" value="New workout"/>	<input type="text" value="Bench press"/>				<input type="button" value="Add training"/>
	<input type="text" value="Push up"/>				

Obrázek 28. Webová aplikace – přidání tréninku

Stránka detailu tréninku je rozdělena na více částí. Na vrchní části je zobrazeno jméno tréninku v podobě nadpisu. Zbytek stránky je rozdělený na dva sloupce. Levý sloupec obsahuje obecné informace jako jméno autora, datum vytvoření, detail tréninku a počet lidí, kteří si trénink oblíbili. V pravém sloupci se nachází tabulka, ve které jeden řádek zobrazuje jeden cvik konkrétního tréninku. Pod tabulkou jsou poté tlačítka pro editaci, smazání a přidání tréninku mezi oblíbené. Smazání probíhá s ověřením pomocí alertu.

New bench

User

Zourkova123

Date

2022-04-21

Likes

1

Detail

Some detail about training...

Body part	Workout	Series	Repetition	Weight (kg)
Chest	Bench press	10	101	10 kg
Biceps	Push up	5	5	5 kg
Chest	Push up	10	8	7 kg

Edit
Delete
Unlike

Obrázek 29. Webová aplikace – detail tréninku

6.2.7 Stránky pro jídla

Manažer i admin mají možnost vytvářet jídla. Pro tyto účely byla vytvořena stránka s tabulkou obsahující všechny jídla z databáze. Součástí tabulky je název jídla, kalorie, bílkoviny, tuky, cukry, datum vytvoření a část dne, kdy je doporučeno jídlo konzumovat. Každý záznam může být zobrazen, editován nebo odstraněn. Uživatelé s autorizací mohou také vytvářet nové záznamy.

Meals

Set Alphabet filter
Delete filter

Name	Day part	Calories	Protein	Sugar	Fat	Date	Buttons
Protein	Dinner	50	30	5	2	2022-05-05	Detail Edit Delete
Steak	Lunch	220	50	6	20	2022-04-14	Detail Edit Delete
Cake	Dinner	214	32	21	45	2022-04-14	Detail Edit Delete

Obrázek 30. Webová aplikace – detail tabulky jídel

Editace a vytvoření nového jídla je složené z šesti inputů, jednoho selectboxu a jednoho tlačítka pro přidání ingrediencí. Mezi inputy, které jsou všechny povinné, spadá název jídla, kalorie, bílkoviny, cukry, tuky a recept. Selectbox zajišťuje zvolení doporučené části dne pro konzumaci jídla. Po stisknutí tlačítka nové ingredience jsou zobrazeny další tři inputy pro název ingredience, počet a jednotku. Jedna ingredience je povinná a při vytvoření druhé lze zrušit celou ingredienci pomocí tlačítka odstranit.

Edit meal form

Meal name: Day part: Calories: Protein: Sugar: Fat:

Recept:

Ingredient name: Amount: Unit:

Ingredient name: Amount: Unit:

Ingredient name: Amount: Unit:

Obrázek 31. Webová aplikace – editace jídla

U detailu jídla lze nalézt veškeré informace, které jídlo má. Mezi ty základní spadá název, datum vytvoření, kalorie, bílkoviny, cukry, tuky a doporučená část dne pro konzumaci. Tyto informace jsou poté doplněny o recept ve formě textu a tabulku s potřebnými ingrediencemi. Poslední částí detailu jsou tlačítka pro editaci a smazání.

Steak

Day part: Date:

Calories: Proteins (g):

Sugars (g): Fat (g):

Recept:

Name	Amount	Unit
Meal	200	g
Peper	5	g
Salt	5	g

Obrázek 32. Webová aplikace – detail jídla

6.2.8 Uživatelské stránky

Poslední položkou, kterou lze nalézt v menu, je stránka s uživateli. Tuto stránku si může zobrazit manažer i admin, ale manažer nemá povolení jakýmkoliv způsobem měnit nebo zobrazovat detaily uživatelských dat. Součástí stránky je tabulka, která je rozdílná dle role. Manažer má k dispozici pouze zobrazení uživatelských jmen a datum vytvoření uživatelského účtu. Admin má tuto tabulku rozšířenou o křestní jméno, příjmení, email a roli. Administrátor může jako jediná role vytvářet, editovat nebo smazat libovolného uživatele.

Users

User name filter
Delete filter

User name	Created
userT1	2022-04-29
TestUser	2022-04-15
Zourkova123	2022-04-15
Salinka	2022-04-14
Zdenek123	2022-04-08

Obrázek 33. Webová aplikace – detail tabulky uživatelů v roli manažera

Users

User name filter
Delete filter

User name	First name	Last name	Email	Role	Created	Buttons
userT1	tester	test	tester1@email.cz	manager	2022-04-29	Detail Edit Delete
TestUser	Test	User	test@email.ct	normal	2022-04-15	Detail Edit Delete
Zourkova123	Adela	Zourkova	zoura@gmail.com	admin	2022-04-15	Detail Edit Delete
Salinka	Ondrej	Salek	salek@gmail.com	manager	2022-04-14	Detail Edit Delete
Zdenek123	Zdenek	Karlik	z_karlik@utb.cz	normal	2022-04-08	Detail Edit Delete

Obrázek 34. Webová aplikace – detail tabulky uživatelů v roli admina

Stránka pro přidání uživatele obsahuje formulář složený z osmi inputů a jednoho selectboxu. Mezi inputy spadá křestní jméno, příjmení, uživatelské jméno, email, heslo, potvrzení hesla, výška a váha. Selectbox poté obsahuje možnosti uživatelských rolí. V případě editačního formuláře se již nezobrazují inputy pro heslo a potvrzení hesla.

Add user form

First name: <input type="text" value="Enter first name"/>	Last name: <input type="text" value="Enter last name"/>	User name: <input type="text" value="Enter user name"/>	Email: <input type="text" value="Enter email"/>
Password: <input type="password" value="Enter password"/>	Confirm password: <input type="password" value="Enter confirm password"/>		
Weight: <input type="text" value="Enter weight"/>	Height: <input type="text" value="Enter height"/>		
Role: <div style="border: 1px solid #ccc; padding: 2px;"><div style="background-color: #f0f0f0; padding: 2px;">admin</div><div style="background-color: #007bff; color: white; padding: 2px;">admin</div><div style="padding: 2px;">manager</div><div style="padding: 2px;">normal</div></div>			

Obrázek 35. Webová aplikace – přidání uživatele

Mimo křestní jméno, příjmení, uživatelské jméno, roli a data o založení účtu, lze zobrazit i informace o váze a výšce. K tomuto účelu byl vytvořen detail uživatele, jehož součástí jsou i tlačítka pro editaci a smazání.

Zdenek123

First name Zdenek	Last name Karlik	Email z_karlik@utb.cz	
Role normal	Date 2022-04-08	Weight 180	Height 85

Obrázek 36. Webová aplikace – detail uživatele

6.3 Popis mobilní aplikace

Mobilní aplikace je rozdělena na pět hlavních částí. První je soubor App, kde je vložena navigace pro pohyb mezi jednotlivými obrazovkami. Pro fungování navigace byla vložena knihovna react-navigation. Uchování JWT tokenu je zajištěno pomocí importované knihovny react-native-async-storage. Druhá část je adresář se samotnou navigací na jednotlivé části aplikace. Tyto části jsou závislé na tom, zda je uživatel přihlášený nebo ne. Nepřihlášený uživatel má k dispozici odkaz na přihlášení a registraci, zatímco přihlášený má více funkcí. Třetí částí jsou komponenty aplikace, kde se nachází části konkrétních navigací. Čtvrtým prvkem jsou samotné obrazovky aplikace. Každá daná část aplikace je rozdělena do podadresáře s vlastním stylováním. Všechny podadresáře jsou poté uloženy v hlavním adresáři s názvem views. Poslední část je soubor nastavení API URL, které je uloženo v adresáři config.

6.3.1 Navigační menu a hlavní obrazovka

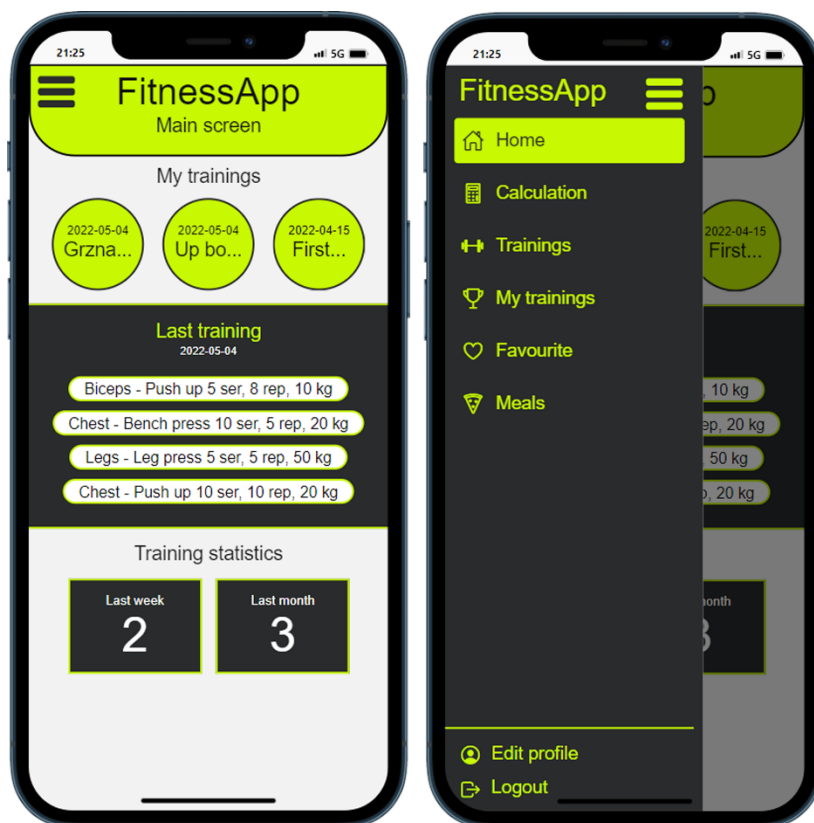
Součástí každého oddílu aplikace je vrchní hlavička složená z nadpisu, podnadpisu a tlačítka pro zobrazení menu. Při kliknutí na tlačítko menu je uživateli zobrazen postranní panel, na kterém může vidět všechny odkazy na obrazovky, které má k dispozici.

Jakmile nepřihlášený uživatel spustí aplikaci je mu načtena hlavní část, složená ze tří hlavních prvků, mezi které spadá zobrazení tří náhodných tréninků, sekce pro výzvu k přihlášení nebo registraci uživatele, jejíž součástí jsou i tlačítka, a ukázka čtyř náhodných receptů. Každý recept a trénink lze rozkliknout a načíst jeho detailní část. Po kliknutí na tlačítko menu je uživateli zobrazena navigace pro domovskou, kalkulační, tréninkovou, přihlašovací a registrační obrazovku.

V případě, kdy je uživatel již přihlášený, je mu zobrazena hlavní stránka s jinými prvky. Náhodné generace je nahrazena jeho posledními třemi tréninky, které si sám vytvořil. V prostřední části obrazovky je k vidění detailní popis posledního provedeného tréninku. Spodní sekce, zaměřená na osobní statistiku, zobrazuje uživateli jeho výkon v posledním týdnu a měsíci. Při zobrazení menu má přihlášený uživatel k dispozici navíc přehled svých tréninků, oblíbených tréninků a také veškerých jídel. Ve spodní části navigace se nachází odkazy pro editaci profilu a odhlášení z aplikace.



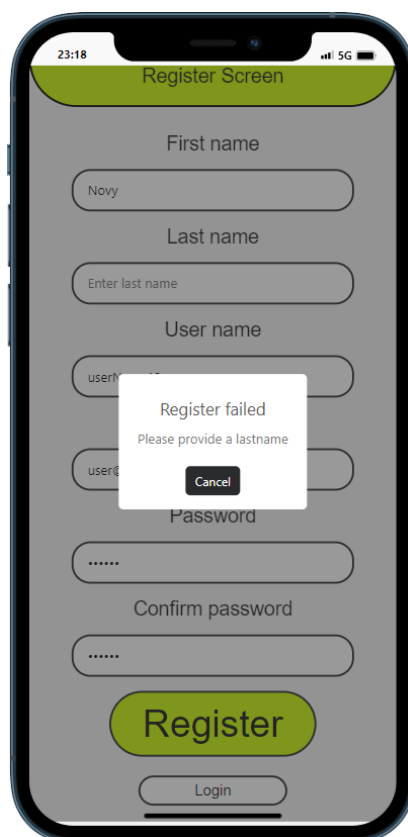
Obrázek 37. Mobilní aplikace – domovská obrazovka nepřihlášeného uživatele



Obrázek 38. Mobilní aplikace – domovská obrazovka přihlášeného uživatele

6.3.2 Přihlášení, registrace a zapomenuté heslo

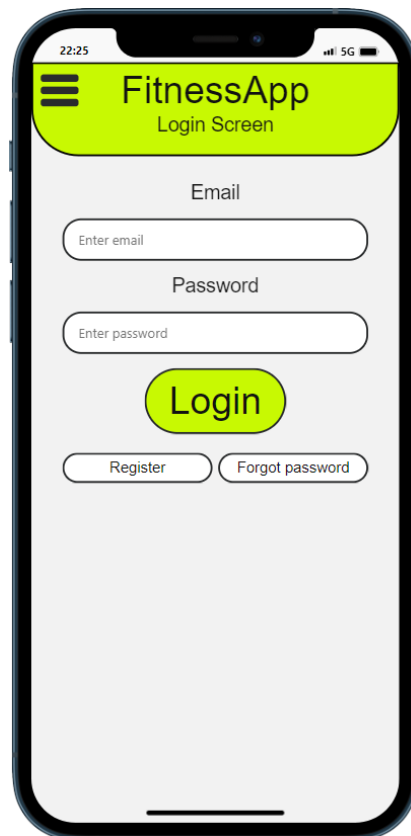
V případě, kdy si neregistrovaný uživatel chce založit svůj účet, musí přejít na obrazovku registrace, kde mu bude zobrazeno šest textových inputů pro zadání křestního jména, příjmení, uživatelského jména, emailu, hesla a potvrzujícího hesla. Pod inputy se nachází tlačítko pro potvrzení registrace a odkaz na přihlášení. Jestliže uživatel nevyplní data validně, je mu zobrazeno okno s hláškou o chybě.



Obrázek 39. Mobilní aplikace – registrační obrazovka s chybou

Uživatel má také možnost obnovy hesla. V případě, kdy si ji vyžádá, je uživateli zobrazen formulář s jedním inputem pro zadání emailu, na který mu bude zaslán odkaz pro obnovení hesla.

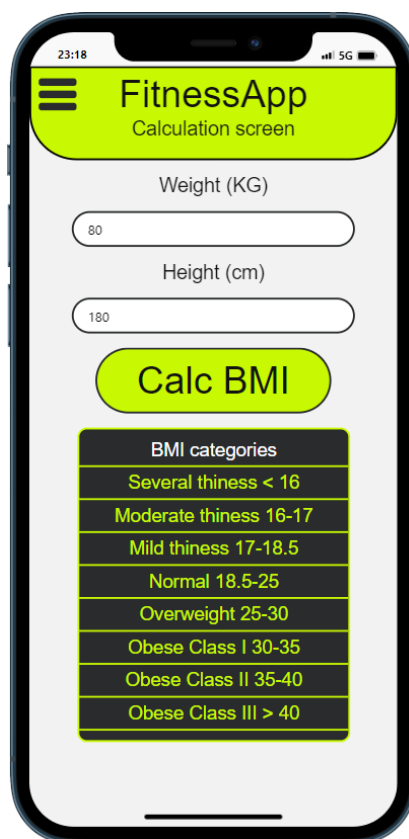
Přihlášený uživatel má v aplikaci rozšířenou funkcionalitu. Z tohoto důvodu byla vytvořena obrazovka, která je složená ze dvou inputů pro zadání emailu a hesla, sloužící pro přihlášení uživatele do systému a odeslání požadavku pro vygenerování JWT tokenu. Jestliže je autentizace úspěšná, je token uložen do paměti a uživateli je zobrazena domovská obrazovka.



Obrázek 40. Mobilní aplikace – obrazovka přihlášení

6.3.3 Kalkulační obrazovka

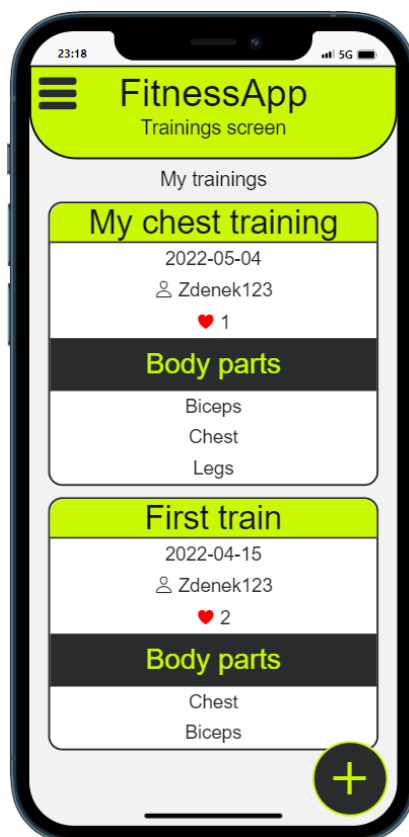
Všichni uživatelé mají možnost zjistit svou aktuální hodnotu BMI indexu. Součástí obrazovky je formulář se dvěma inputy, kde uživatel zadá svou výšku a váhu. Po vyplnění je uživateli vrácena aktuální hodnota BMI společně s kategorií, do které uživatel zapadá. Pod formulářem je k dispozici seznam všech kategorií. [25]



Obrázek 41. Mobilní aplikace – obrazovka kalkulace BMI

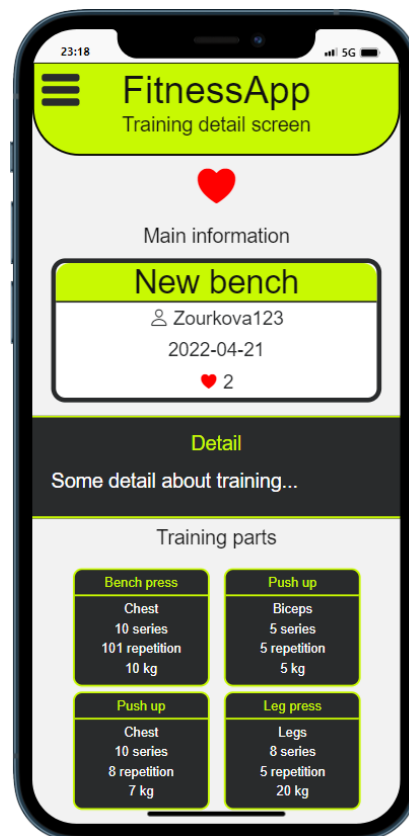
6.3.4 Zobrazení tréninků a operace s nimi

Každý uživatel má k dispozici obrazovku se seznamem všech vytvořených tréninků, které jsou řazeny dle data vytvoření od nejnovějšího. Součástí každé entity v je název, datum vytvoření, autorovo jméno, seznam částí těla, kterých se záznam týká, a počet uživatelů, kteří si trénink oblíbili. Obrazovka mé tréninky, které zobrazuje pouze entity vytvořené daným uživatelem, má stejný vzhled s rozšířením o tlačítko pro přidání tréninku. V případě oblíbených jsou uživateli zobrazeny entity, které již přidal do svého seznamu oblíbených.



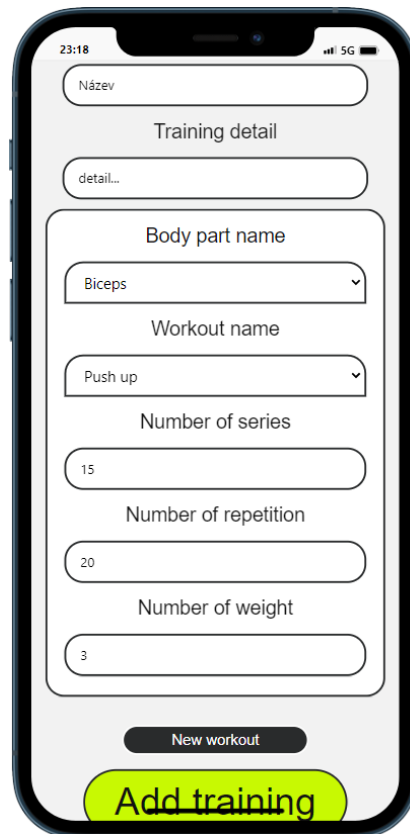
Obrázek 42. Mobilní aplikace – obrazovka zobrazení vlastních tréninků

Po rozkliknutí na detailní obrazovku je uživateli zobrazen navíc detailní popis entity, tlačítko pro možnost přidání nebo odebrání tréninku z oblíbených a výpis všech cviků společně s jejich počtem sérií, opakování a závaží. Pokud zobrazený tréninku vytvořil přihlášený uživatel, je na spodní části obrazovky navíc uživateli zobrazeno tlačítko pro editaci a smazání záznamu.



Obrázek 43. Mobilní aplikace – obrazovka detailu cizího tréninku

Přidání a editace tréninku, jsou obrazovky složené z formuláře, který obsahuje dva inputy pro jméno a detail. Tento formulář může být rozšířen o libovolné množství cviků, které se přidávají pomocí tlačítka nový cvik. Každý cvik je složený z názvu a názvu částí těla, které jsou vybírány pomocí selectboxů. Tyto názvy jsou doplněny o inputy pro počet sérií, opakování a závaží. V případě editace jsou všechny povinné části již předvyplněny. Obě operace mají k dispozici pouze přihlášení uživatelé.



The image shows a mobile application interface for adding a training session. The screen displays a form with the following fields and controls:

- Time: 23:18, Signal: 5G, Battery icon.
- Input field: "Název" (Name).
- Section: "Training detail".
- Input field: "detail..."
- Section: "Body part name".
- Select box: "Biceps" (dropdown arrow).
- Section: "Workout name".
- Select box: "Push up" (dropdown arrow).
- Section: "Number of series".
- Input field: "15".
- Section: "Number of repetition".
- Input field: "20".
- Section: "Number of weight".
- Input field: "3".
- Button: "New workout" (dark grey).
- Button: "Add training" (yellow).

Obrázek 44. Mobilní aplikace – obrazovka přidání tréninku

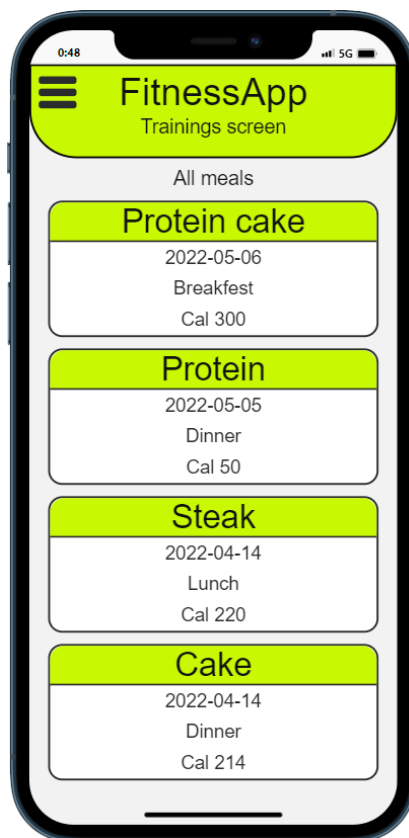
Pro smazání záznamu je nutné provést potvrzení v okně, které se zobrazí po stisknutí tlačítka odstranit. Funkci smazání tréninku má k dispozici pouze jeho přihlášený autor.



Obrázek 45. Mobilní aplikace – obrazovka odstranění tréninku

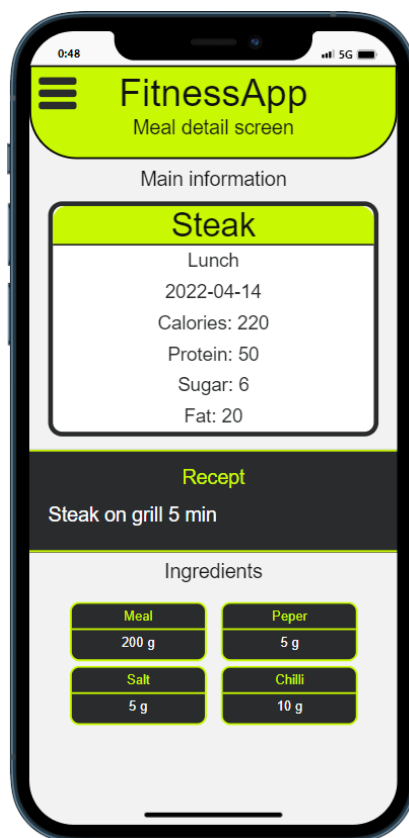
6.3.5 Zobrazení jídel a jejich detailu

Po přejití na obrazovku všech jídel, které má k dispozici pouze přihlášený uživatel, dochází k zobrazení seznamu vytvořených entit. Součástí každé entity jsou atributy názvu, data vytvoření, počtu kalorií a doporučené denní doby ke konzumaci.



Obrázek 46. Mobilní aplikace – obrazovka zobrazení všech jídel

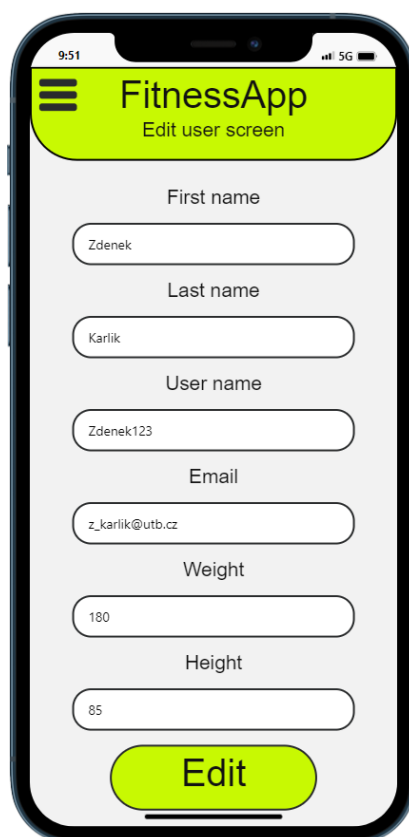
Uživatel má po kliknutí na dané jídlo možnost vyobrazit počet kalorií, bílkovin, cukrů, tuků, detail receptu a seznam všech ingrediencí složený z názvu, množství a jednotky.



Obrázek 47. Mobilní aplikace – obrazovka detail jídla

6.3.6 Editace profilu

Přihlášený uživatel má možnost upravit svá profilová data pomocí odkazu pro editaci profilu. Poté co je načtena obrazovka, jsou vyplněna data o uživateli do konkrétních inputů uvnitř formuláře. Uživatel je může libovolně zaměnit, ale musí splnit podmínky validity. Mezi inputy spadá křestní jméno, příjmení, uživatelské jméno, email, výška a váha.



The image shows a mobile application interface for editing a user profile. The screen has a light green header with the app name 'FitnessApp' and the subtitle 'Edit user screen'. Below the header, there are six input fields, each with a label above it and a sample value inside the field:

- First name: Zdenek
- Last name: Karlik
- User name: Zdenek123
- Email: z_karlik@utb.cz
- Weight: 180
- Height: 85

At the bottom of the form is a large green button labeled 'Edit'.

Obrázek 48. Mobilní aplikace – obrazovka editace profilu

ZÁVĚR

Hlavním cílem bakalářské práce bylo vytvoření webové a mobilní fitness aplikace s pokrytím veškerých požadavků funkční analýzy.

V teoretické části byla provedena literární rešerše, zaměřená na webové a mobilní technologie, využívané v praktické části práce. U webové části aplikace dojde k seznámení s tématy HTML, CSS, Javascript, a především balíčkem MERN, který se skládá z technologií MongoDB, Express.js, React a Node.js. V případě mobilních technologií dochází k objasnění frameworku React Native.

Součástí praktické části je zachycen návrh aplikací, který je složený z analýzy požadavků, modelu případů užití, konkrétních scénářů případu užití a navržení základního UI. Na návrhy navazuje představení hlavních částí serverové stránky aplikace, kde jsou zobrazeny hlavní metody zajišťující propojení serveru s databází, autentizace s autorizací, komunikaci s API a validaci dat na straně serveru. Poslední oddíl bakalářské práce se týká popisu jednotlivých aplikací včetně názorných ukázek.

Mobilní aplikace zajišťuje evidenci uživatelských tréninků. Přihlášený uživatel má možnost přidávat a odebírat tréninky z oblíbených, zobrazovat veškerá vytvořená jídla a zobrazovat si svou statistiku z posledního týdne a měsíce. Pro neregistrované uživatele je vytvořena aplikace s omezenými funkcemi jako je zobrazení náhodných tréninků nebo výpočet BMI.

Webová aplikace zajišťuje evidenci částí těla, cviků, tréninků, jídel a uživatelů. Pro přístup k webové aplikaci je nutné mít vytvořený účet s autorizací manažera nebo admina.

SEZNAM POUŽITÉ LITERATURY

- [1] What is Web Application. Javatpoint [online]. 2021 [cit. 2022-02-08]. Dostupné z: <https://www.javatpoint.com/web-application>
- [2] FELKE-MORRIS, Terry. Web development and design foundations with HTML5. 9th edition. New York, NY: Pearson, [2019]. ISBN 978-0134801148.
- [3] Proč se zajímat o HTML5. IT Network [online]. 2022 [cit. 2022-02-09]. Dostupné z: <https://www.itnetwork.cz/html-css/html5>
- [4] Úvod do CSS. IT Network [online]. 2022 [cit. 2022-02-09]. Dostupné z: <https://www.itnetwork.cz/html-css/webove-stranky/jak-psat-moderni-web-html-tutorial-uvod-do-css>
- [5] Javascript frameworks. Technostacks [online]. 2021 [cit. 2022-02-09]. Dostupné z: <https://technostacks.com/blog/best-javascript-frameworks>
- [6] SUBRAMANIAN, Vasam. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node. Berkeley, CA: Apress, 2017. ISBN 978-1484226520.
- [7] Úvod do MongoDB. IT Network [online]. 2022 [cit. 2022-02-23]. Dostupné z: <https://www.itnetwork.cz/javascript/nodejs/uvod-do-mongodb>
- [8] HOQUE, Shama. Full-Stack React Projects: Learn MERN stack development by building modern web apps using MongoDB, Express, React, and Node.js. 2nd Edition. Birmingham: Packt, 2018. ISBN 978-1839215414.
- [9] Úvod do Node.js. IT Network [online]. 2022 [cit. 2022-02-23]. Dostupné z: <https://www.itnetwork.cz/javascript/nodejs/uvod-do-nodejs>
- [10] Basic routing. Expressjs [online]. 2017 [cit. 2022-02-23]. Dostupné z: <https://expressjs.com/en/starter/basic-routing.html>
- [11] HTTP Request Methods. W3schools [online]. 2022 [cit. 2022-02-23]. Dostupné z: https://www.w3schools.com/tags/ref_httpmethods.asp
- [12] BODUCH, Adam. React and React Native. 3rd Edition. Birmingham: Packt, 2020. ISBN 978-1839211140.
- [13] React Components. W3schools [online]. 2022 [cit. 2022-02-23]. Dostupné z: https://www.w3schools.com/react/react_components.asp

- [14] Atomic Design Methodology. Atomicdesign [online]. 2022 [cit. 2022-02-24]. Dostupné z: <https://atomicdesign.bradfrost.com/chapter-2/>
- [15] How to Style Your React App. Freecodecamp [online]. 2021 [cit. 2022-02-24]. Dostupné z: <https://www.freecodecamp.org/news/how-to-style-react-apps-with-css/>
- [16] Mobile Application. Techopedia [online]. 2022 [cit. 2022-02-24]. Dostupné z: <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>
- [17] Core Components and Native Components. Reactnative [online]. 2022 [cit. 2022-02-24]. Dostupné z: <https://reactnative.dev/docs/intro-react-native-components>
- [18] Jak rychle a kvalitně vytvořit (nejen) webovou aplikaci. IT Network [online]. 2022 [cit. 2022-02-24]. Dostupné z: <https://www.itnetwork.cz/navrh/best-practices-pro-vyvoj-softwaru/jak-rychle-a-kvalitne-vytvorit-webovou-aplikaci>
- [19] Úvod do UML. IT Network [online]. 2022 [cit. 2022-03-02]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-uvod-historie-vyznam-a-diagramy>
- [20] SOMMERVILLE, Ian. Software Engineering. 10th Edition. Edinburgh Gate: Pearson, 2016. ISBN 978-1292096131.
- [21] Use-Case Model. Javatpoint [online]. 2021 [cit. 2022-03-09]. Dostupné z: <https://www.javatpoint.com/use-case-model>
- [22] Use Case Specifikace. IT Network [online]. 2022 [cit. 2022-03-10]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-specifikace-diagram>
- [23] Wireframing in UI/UX Design. Medium [online]. 2020 [cit. 2022-03-10]. Dostupné z: <https://medium.com/detaux/what-is-ui-ux-wireframe-designerss-46dac9c8a153>
- [24] About Git. GitHub [online]. 2022 [cit. 2022-03-10]. Dostupné z: <https://docs.github.com/en/get-started/using-git/about-git>
- [25] BMI Calculator. Calculator [online]. 2022 [cit. 2022-04-30]. Dostupné z: <https://www.calculator.net/bmi-calculator.html>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

API	Aplikační programovací rozhraní
BMI	Index tělesné hmotnosti
BSON	Binární JavaScriptový objektový zápis
CRUD	Vytvořit, číst, editovat, smazat
CSS	Kaskádové styly
DB	Databáze
DOM	Objektový model dokumentu
HTML	Hypertextový značkovací jazyk
ID	Identita
JSON	JavaScriptový objektový zápis
JSX	Rozšíření syntaxe Javascriptu
JWT	JSON web token
LAMP	Linux, Apache, MySQL a PHP
MEAN	MongoDB, Express, Angular.js a Node.js
MERN	MongoDB, Express, React a Node.js
NoSQL	Not only SQL
ORM	Objektově relační mapování
SASS	Syntactically Awesome Style Sheets
SCSS	Syntactically Cascading Style Sheets
SQL	Standardizovaný strukturovaný dotazovací jazyk
UC	Případ užití
UI	Uživatelské rozhraní
UML	Unifikovaný sjednocený jazyk
URI	Jednotný identifikátor zdroje

URL Jednotný lokátor zdroje

SEZNAM OBRÁZKŮ

Obrázek 1. Model případu užití webové aplikace	26
Obrázek 2. Wireframe – formulář pro přihlášení	35
Obrázek 3. Wireframe – základní dashboard se statistikou.....	35
Obrázek 4. Wireframe – zobrazení s filtracemi	36
Obrázek 5. Wireframe – formulář pro přidání prvku.....	36
Obrázek 6. Wireframe – zobrazení detailu o prvku.....	37
Obrázek 7. UC diagram mobilní aplikace	39
Obrázek 8. Wireframe – hlavní stránka bez registrace	47
Obrázek 9. Wireframe – hlavní stránka s přihlášením.....	47
Obrázek 10. Wireframe – menu.....	48
Obrázek 11. Wireframe – přidání tréninku	49
Obrázek 12. Entita uživatele v databázi.....	51
Obrázek 13. Entita jídla v databázi	51
Obrázek 14. Entita části těla v databázi	52
Obrázek 15. Entita cviku v databázi	52
Obrázek 16. Entita tréninku v databázi.....	53
Obrázek 17. Webová aplikace – login	57
Obrázek 18. Webová stránka – nepovolený přístup	58
Obrázek 19. Webová aplikace – detail menu	59
Obrázek 20. Webová stránka – dashboard.....	60
Obrázek 21. Webová aplikace – části těla	60
Obrázek 22. Webová aplikace – detail tabulky částí těla	61
Obrázek 23. Webová aplikace – přidání části těla.....	61
Obrázek 24. Webová aplikace – detail tabulky cviků	62
Obrázek 25. Webová aplikace – přidání cviku	62
Obrázek 26. Webová aplikace – detail cviku	63
Obrázek 27. Webová aplikace – detail tabulky tréninků	63
Obrázek 28. Webová aplikace – přidání tréninku.....	64
Obrázek 29. Webová aplikace – detail tréninku	65
Obrázek 30. Webová aplikace – detail tabulky jídel	65
Obrázek 31. Webová aplikace – editace jídla.....	66
Obrázek 32. Webová aplikace – detail jídla	66

Obrázek 33. Webová aplikace – detail tabulky uživatelů v roli manažera.....	67
Obrázek 34. Webová aplikace – detail tabulky uživatelů v roli admina	67
Obrázek 35. Webová aplikace – přidání uživatele	68
Obrázek 36. Webová aplikace – detail uživatele	68
Obrázek 37. Mobilní aplikace – domovská obrazovka nepřihlášeného uživatele	70
Obrázek 38. Mobilní aplikace – domovská obrazovka přihlášeného uživatele.....	70
Obrázek 39. Mobilní aplikace – registrační obrazovka s chybou.....	71
Obrázek 40. Mobilní aplikace – obrazovka přihlášení	72
Obrázek 41. Mobilní aplikace – obrazovka kalkulace BMI	73
Obrázek 42. Mobilní aplikace – obrazovka zobrazení vlastních tréninků.....	74
Obrázek 43. Mobilní aplikace – obrazovka detailu cizího tréninku	75
Obrázek 44. Mobilní aplikace – obrazovka přidání tréninku	76
Obrázek 45. Mobilní aplikace – obrazovka odstranění tréninku	77
Obrázek 46. Mobilní aplikace – obrazovka zobrazení všech jídel	78
Obrázek 47. Mobilní aplikace – obrazovka detail jídla	79
Obrázek 48. Mobilní aplikace – obrazovka editace profilu.....	80

SEZNAM TABULEK

Tabulka 1. Scénář – přihlášení uživatele	27
Tabulka 2. Scénář – editace účtu	28
Tabulka 3. Scénář – zobrazení všech záznamů.....	28
Tabulka 4. Scénář – přidání záznamu	29
Tabulka 5. Scénář – editace záznamu	29
Tabulka 6. Scénář – smazání záznamu	30
Tabulka 7. Scénář – zobrazení detailu záznamu.....	30
Tabulka 8. Scénář – zobrazení registrovaných uživatelů.....	31
Tabulka 9. Scénář – zobrazení statistik	31
Tabulka 10. Scénář – zobrazení filtrace.....	32
Tabulka 11. Scénář – zobrazení uživatelů a rolí	32
Tabulka 12. Scénář – vytvoření uživatele.....	33
Tabulka 13. Scénář – editace uživatelů	33
Tabulka 14. Scénář – smazání uživatele	34
Tabulka 15. Scénář – zobrazení uživatele	34
Tabulka 16. Scénář – registrace uživatele	40
Tabulka 17. Scénář – výpočet a kalkulace.....	40
Tabulka 18. Scénář – zobrazení náhodných jídel	41
Tabulka 19. Scénář – zobrazení náhodných tréninků	41
Tabulka 20. Scénář – zobrazení tréninků.....	42
Tabulka 21. Scénář – zobrazení svých tréninků	42
Tabulka 22. Scénář – přidání tréninku.....	42
Tabulka 23. Scénář – editace tréninku.....	43
Tabulka 24. Scénář – smazání tréninku	43
Tabulka 25. Scénář – zobrazení detailu tréninku.....	44
Tabulka 26. Scénář – zobrazení menu	44
Tabulka 27. Scénář – zobrazení detailu jídla	45
Tabulka 28. Scénář – zobrazení osobních statistik	45
Tabulka 29. Scénář – zobrazení oblíbených tréninků.....	45
Tabulka 30. Scénář – editace účtu	46

SEZNAM PŘÍLOH

Příloha P I: Obsah všech CD

PŘÍLOHA P I: OBSAH VŠECH CD

Součástí přiložených CD je zdrojový kód aplikací