



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Aplikace pro plánování sportovního tréninku

Bakalářská práce

Studijní program: B2646 – Informační technologie

Studijní obor: 1802R007 – Informační technologie

Autor práce: **Anzhelika Rysaeva**

Vedoucí práce: Mgr. Jiří Vraný, Ph.D.





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Application for sport training plans

Bachelor thesis

Study programme: B2646 – Information Technology

Study branch: 1802R007 – Information Technology

Author: **Anzhelika Rysaeva**

Supervisor: Mgr. Jiří Vraný, Ph.D.





Zadání bakalářské práce

Aplikace pro plánování sportovního tréninku

<i>Jméno a příjmení:</i>	Anzhelika Rysaeva
<i>Osobní číslo:</i>	M16000051
<i>Studijní program:</i>	B2646 Informační technologie
<i>Studijní obor:</i>	Informační technologie
<i>Zadávací katedra:</i>	Ústav nových technologií a aplikované informatiky
<i>Akademický rok:</i>	2018/2019

Zásady pro vypracování:

1. Seznamte se s problematikou programování webových služeb s podporou REST API, s problematikou ukládání většího množství provázaných dat a SW řešeními používanými pro tvorbu klientské a serverové části aplikace v jazyce JavaScript.
2. Navrhněte aplikaci pro plánování sportovního tréninku, která umožní trenérům připravovat tréninkové plány pro závodníky a závodníkům naopak tyto tréninkové plány číst a reportovat průběh tréninku. Při návrhu se zaměřte na bezpečnost systému, uživatelskou přístupnost a srozumitelnost.
3. Navrženou aplikaci implementujte a ověřte její praktickou použitelnost. Při implementaci využijte architektury REST, vytvořte serverovou i klientskou část.

Rozsah grafických prací:
Rozsah pracovní zprávy:
Forma zpracování práce:
Jazyk práce:

dle potřeby
30 – 40 stran
tištěná/elektronická
Čeština



Seznam odborné literatury:

- [1] RICHARDSON, Leonard a Michael AMUNDSEN. RESTful Web APIs. Beijing: O'Reilly, 2013. ISBN 978-1449358068.
- [2] SUBRAMANIAN, Vasam. Pro MERN stack: full stack web app development with Mongo, Express, React, and Node. Berkeley, California: Apress, 2017. Books for professionals by professionals. ISBN 978-1484226520.
- [3] BANKS, Alex a Eve PORCELLO. Learning React: functional web development with React and Redux. Sebastopol, CA: O'Reilly Media, 2017. ISBN 978-1491954621.

Vedoucí práce:

Mgr. Jiří Vraný, Ph.D.
Ústav nových technologií a aplikované informatiky

Datum zadání práce:

18. října 2018

Předpokládaný termín odevzdání:

30. dubna 2019

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

L.S.

Ing. Josef Novák, Ph.D.
vedoucí ústavu

V Liberci dne 18. října 2018

Prohlášení

Byla jsem seznámena s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracovala samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že texty tištěné verze práce a elektronické verze práce vložené do IS/STAG se shodují.

24. srpna 2019

Anzhelika Rysaeva

Poděkování

Ráda bych poděkovala vědoucímu své bakalářské práce panu Mgr. Jiřímu Vranému, Ph.D. za odborné vedení a pomoc při zpracování této práce.

Abstrakt

Tato bakalářská práce se zaměřuje na implementaci webové aplikace pro vytrvalostní sporty, která je určena pro cyklistiky, běžce, lyžaře atd. Aplikace nabízí trenérům možnost vytvářet tréninkové plány a sportovcům tyto tréninkové plány číst a reportovat průběh tréninku. Dosažení daného cíle v této práci probíhá zaprvé zkoumáním problematiky programování webových služeb a populárních webových řešení sportovních aplikací, a zadruhé zvolením vhodných technologií pro vývoj a následným návrhem vlastního řešení, které je podrobně popsáno v kapitole „Rešerše“.

Jedním z důležitých kroků pro dosažení cíle je implementace rozhraní, která je podrobněji popsána v kapitole „Webová aplikace“. V ní se soustředím na popis použitých technologií a jednotlivých implementovaných funkcí pro tuto webovou aplikaci. Webová aplikace využívá programovací jazyk TypeScript, JavaScriptovou knihovnu React, pomocné React nástroje Context a knihovnu Ant Design pro implementaci klientské části, a systém Node.js a framework Express.js pro realizaci serveru a také nerelační databáze MongoDB pro práci s daty.

Poslední část bakalářské práce je věnována zhodnocení dosažených výsledků a plánům rozšíření funkcí této webové aplikace do budoucna.

Klíčová slova: Webová aplikace, plánování sportovního tréninku, REST, MongoDB, TypeScript, React, Node.js

Abstract

This bachelor's thesis focuses on the implementation of an endurance sports web application which was developed for sportsmen such as cyclists, runners or skiers. Thanks to this application, coaches build training plans for athletes who can access them during their training session. Moreover, this application enables an easy addition of comments to any activity. The goals of this thesis are focused on programming of web services and popular web-based solutions for sport applications. Moreover, it focuses on the selection of appropriate technologies for the development of an own application. The above-mentioned process is described in detail in the chapter Research. This thesis is specifically concerned with the implementation of an interface which is discussed in the chapter Web Application. This chapter focuses on the description of technologies used and the features implemented. The web application uses a programming language TypeScript, JavaScript library React, React auxiliary tools Context and the Ant Design library for programming the client – side of the web application. Node.js and server framework Express.js were used to implement the backend part of the web application, data management is handled by the non-relational database MongoDB. The last chapter of the thesis evaluates the results achieved and explores the future development of the web application.

Keywords: Web application, sport training plans, REST, MongoDB, TypeScript, React, Node.js

Obsah

Seznam zkratek	12
1 Úvod	13
2 Rešerše	14
2.1 Problematika programování webových služeb	14
2.1.1 REST API	15
2.1.2 Ukládání většího množství provozovaných dat	16
2.1.3 Softwarové řešení pro tvorbu klientské a serverové části aplikace v jazyce JavaScript	17
2.2 Dostupné webové aplikace pro plánování sportovního tréninku	19
2.2.1 Trainingpeaks.com	20
2.2.2 Strava.com	21
2.2.3 Trainright.com	23
2.3 Volba funkcí, technologií webové aplikace a motivace k bakalářské práci	24
3 Webová aplikace	26
3.1 Fundament aplikace	26
3.1.1 Vícejazyčnost aplikace	26
3.1.2 Klientská část	27
3.1.3 Serverová část	29
3.2 Datové úložiště	29
3.3 Realizované rozhraní webové aplikace	31
3.3.1 Profil sportovce	33
3.3.2 Kalendář tréninků	34
3.3.3 Přidání tréninků	34
3.3.4 Analýza výsledků tréninků sportovce	35
3.3.5 Komunikace trenéra a sportovce	38

3.4	Bezpečnost webové aplikace	38
4	Závěr	41
	Seznam použité literatury	42

Seznam obrázků

2.1	Trainingpeaks.com: Kalendář závodníka	20
2.2	Trainingpeaks.com: seznam závodníků trenéra a jejich data	21
2.3	Trainingpeaks.com: analýza výsledných hodnot tréninku	22
2.4	Strava.com: praktická ukázka profilu webové aplikace	23
2.5	Strava.com: funkce Gobal Heatmap	24
2.6	Webová aplikace Trainright.com	25
3.1	Struktura React komponent webové aplikace	28
3.2	Model MongoDB	30
3.3	Přihlašovací formulář	31
3.4	Profil sportovce: osobní údaje a týdenní statistika trénování	33
3.5	Profil sportovce: naplánované tréninky	33
3.6	Profil sportovce: odtrénované tréninky	34
3.7	Kalendář tréninků: rozhraní pro trenéra	34
3.8	Kalendář tréninků: rozhraní pro sportovce	35
3.9	Přidání tréninků: editor tréninků	35
3.10	Přidání tréninků: knihovna tréninků	36
3.11	Rozhraní pro sportovce: pohled na trénink a funkce importu výsledků do aplikace	36
3.12	Převod dat z formátu XML do JSON	37
3.13	Graf výsledných hodnot tréninku	37
3.14	Funkce komunikace trenéra a sportovce	38
3.15	Záhlaví odpovědi HTTP	39

Seznam zkratek

REST	(Representational state transfer) je architektura komunikace server-klient, která byla navržena Royem Fieldingem
SOAP	(Simple Object Access Protocol) je protokol sloužící pro výměnu zpráv pomocí HTTP
React	je JavaScriptová knihovna určena pro vytváření uživatelských rozhraní
Node.js	je softwarový systém, který je založen na V8 JavaScript od společnosti Google
XML	(eXtensible Markup Language) je značkový jazyk vyvinutý W3C
CRUD	(Create, Read, Update, Delete) jsou čtyři operace, které se provádějí nad záznamem v trvalém úložišti
JSON	(JavaScript Object Notation) je datový formát
HTML	(HyperText Markup Language) značkovací jazyk používaný pro tvorbu webových stránek
CSS	(Cascading Style Sheets) jsou kaskádové styly
JSX	(JavaScript XML) je rozšíření syntaxe jazyka JavaScript
PHP	(Hypertext Preprocessor) je skriptovací programovací jazyk pro implementaci serverové části webových aplikací
SQL	(Structured Query Language) je dotazovací jazyk, který je používán pro manipulaci s daty v relačních databázích
MongoDB	je dokumentová databáze
CSRF	(Cross Site Request Forgery) je metoda útoku na webové stránky, která zneužívá akce uživatelů, kteří jsou během útoku přihlášení
XSS	(Cross-site scripting) je metoda útoku na webové aplikaci prostřednictvím chyb ve skriptech aplikace
API	(Application Programming Interface) je rozhraní, které je vytvořeno za účelem programování aplikací
UI	(User Interface) je uživatelské rozhraní
GPX	(GPS eXchange Format) je textový formát určený pro uchovávání dat GPS, který je založený na XML

1 Úvod

V současné době existuje velký výběr webových aplikací pro různé typy sportů, např. pro plánování fitness tréninku nebo měření počtu kroků a další. Tyto aplikace mají různá nastavení i funkce a pomáhají trenérům a sportovcům zlepšovat svůj výkon.

Tato práce je založena na splnění hlavního cíle — realizaci webového prostředí, které umožní trenérům připravovat tréninkové plány pro sportovce a sportovcům umožňuje tyto plány číst a reportovat průběh tréninku. Metody řešení daného cíle spočívají ve zkoumání dvou důležitých aspektů — problematiky programování webových služeb a existujících webových prostředí pro plánování tréninků.

Co se týče prvního aspektu, je vhodné se soustředit na problematiku vývoje webových služeb s podporou REST API, problematiku ukládání většího množství provozovaných dat a existující softwarová řešení pro tvorbu klienta a serveru v jazyce JavaScript. Tyto technologie byly požadavkem pro tuto práci a jsou také vhodné pro vývoj moderních webových služeb. Co se druhého aspektu týče, efektivním způsobem je zanalyzovat existující řešení sportovních webových aplikací, zjistit jaké funkce jsou pro trenéra i pro sportovce praktické, a na základě zjištěných informací navrhnout a implementovat webovou aplikaci, která bude přinášet něco užitečného nebo nového pro obě strany.

2 Rešerše

Tato kapitola popisuje problematiku programování webových služeb s podporou REST API, na příkladě porovnání architektury rozhraní REST a SOAP protokolu pro výměnu zpráv přes síť. Také seznamuje s problematikou ukládání většího množství provozovaných dat prostřednictvím analýzy relačních a nerelačních databází, na příkladě MySQL a MongoDB. Dále jsou zde popsány technologie pro tvorbu klientské a serverové části webové aplikace, což je programovací jazyk JavaScript, jeho knihovny a frameworky a také programovací jazyk PHP a platforma Node.js. Po rozboru problematiky programování webových služeb tato kapitola analyzuje dostupné webové aplikace pro plánování sportovního tréninku a na příkladě Trainingpeaks.com, Strava.com a Trainright.com zjišťuje praktické použitelné funkce sportovních webových aplikací. Na konci je uveden návrh funkcí vlastní webové aplikace a zvolené technologie pro programování této webové aplikace spolu s motivací pro tuto práci.

2.1 Problematika programování webových služeb

Webová služba je softwarové prostředí, které umožňuje komunikaci mezi klientskými a serverovými aplikacemi na webu. Služby mohou být vyhledány po síti a mohou být také vyvolány. Při vyvolání by měla webová služba poskytnout určité funkce klientovi, který ji vyvolal.[1]

Po přečtení této obecné ale velmi přesné definice je vidět, že není zde zmínka o síťovém protokolu, programovacím jazyku ani softwarové platformě. To znamená, že neexistují žádná omezení na použití technologie. To znamená, že tento softwarový systém umožňuje komunikaci a výměnu dat několika informačních systémů, nehledě na použité technologie, jazyky a protokoly komunikace.

Problematika programování webových služeb spočívá v tom, že je pro každý projekt ještě na začátku vývoje potřeba zvolit vhodné technologie pro rychlý a

efektivní vývoj. To jsou technologie pro programování klientské a serverové části aplikace, pak návrh datového úložiště a výběr architektury rozhraní nebo protokoly výměny zpráv.[1]

Následující podkapitoly popisují několik existujících řešení pro programování webových služeb a webových aplikací a pak volí vhodné technologie pro tuto práci.

2.1.1 REST API

Pro výměnu informací v decentralizovaném distribuovaném prostředí se používají protokoly a architektury. Příkladem takových protokolů a architektur pro webové služby jsou SOAP (Simple Object Access Protokol), REST (Representational State Transfer) a XML-RPC (XML Remote Procedure Call).

SOAP protokol je odvozen z XML-RPC a je jeho dalším krokem ve vývoji, zatímco REST je spíše architektura komunikace založená na manipulaci s objekty CRUD (Create Read Update Delete). Samozřejmě existují i jiné protokoly, ale protože se široce nepoužívají, zaměřím se na dvě hlavní technologie – SOAP a REST. Vzhledem k tomu, že XML-RPC je „zastaralý“ protokol, nebudu ho analyzovat detailněji.[2]

REST neboli Representational state transfer je architektura rozhraní, navržená pro systémy jako je např. WWW (world wide web), stavěný pro webové služby. Architektura REST je z hlediska použití velmi jednoduchá. Díky vzhledu příchozího požadavku můžeme okamžitě určit, co daný požadavek dělá bez nutnosti pochopení formátu (na rozdíl od SOAP, XML-RPC). REST je považován za méně náročný, protože data jsou přenášena bez použití dalších vrstev a není tedy nutno parsovat požadavek pro pochopení a není potřeba překládat data z jednoho formátu do druhého. Požadavky a odpovědi RESTful systémů mohou být přenášeny v ASCII, XML, JSON nebo v jakýchkoliv jiných formátech, které rozpozná klient a server. Formát JSON je jednodušší a pochopitelný pro čtení a psaní. Ve většině případů je ideální pro výměnu dat přes internet. [3]

SOAP protokol slouží pro výměnu zpráv a využívá XML ke kódování požadavku a odpovědi mezi klientem a serverem. Tento protokol velmi dlouho sloužil pro zaslání zpráv, používaly ho téměř všechny webové služby. SOAP se většinou používá pro podnikové webové služby vyžadující vysoký stupeň zabezpečení, například pro finanční a platební služby. Dalším argumentem pro uplatnění SOAP je podpora starších webových služeb, které mají hodně uživatelů, a ty se připojují ke svým

službám prostřednictvím rozhraní SOAP. [2]

Vzhledem k tomu, že vývojáři v dnešní době potřebují vytvářet lehké webové a mobilní aplikace, získala flexibilnější architektura REST rychle na popularitě. A proto je v této bakalářské práci při implementaci vlastního řešení webové aplikace, podle mého názoru, vhodné použití REST architektury komunikace klienta a serveru, což přináší flexibilitu a jednoduchost z hlediska užívání a s použitím výměny dat ve formátu JSON přináší lepší pochopitelnost pro čtení a psaní zpráv.

2.1.2 Ukládání většího množství provozovaných dat

Ještě před začátkem vývoje jakéhokoliv softwaru je pro vývojáře hlavní otázkou volba relační (SQL) nebo nerelační (NoSQL) databáze.

Relační databáze používají dotazovací jazyk SQL (Structure Query Language) pro definování a zpracování strukturovaných dat, což představuje velké možnosti pro softwarový vývoj. Důležitou vlastností relační databáze je to, že data jsou prezentována ve formě tabulek. MySQL, Oracle, PostgreSQL a Microsoft SQL server jsou systémy řízení báze dat podporující relační model.[4]

Co se nerelačních databází týče, existuje velké množství NoSQL technologií - dokumentové, objektové, grafové atp. Například dokumentová databáze prezentuje data ve formě dokumentů, které vývojář může během vývoje softwaru vytvářet, aniž by musel nastavovat jejich struktury, každý dokument přitom může mít svou vlastní strukturu. Pro práci s nerelační databází jsou vhodné systémy řízení báze dat podporující nerelační model, což jsou MongoDB, BigTable, Redis, RavenDB, Cassandra, HBase, CouchDB atd.[5]

Pro lepší pochopení výhod a nevýhod SQL a NoSQL systémů řízení báze dat a pro volbu nejvhodnějšího systému pro vlastní řešení webové aplikace dále porovnávám relační a nerelační model na příkladě MySQL a MongoDB.

MySQL je vysoce vyvinutý databázový systém, kolem kterého je vytvořena velká komunita vývojářů, obsahuje velké množství příkladů a je vysoce spolehlivý. Tato databáze je dostupná na všech hlavních platformách včetně Linuxu, Windows, BSD nebo Solarisu. Jedná se o open source databázi, která je volně dostupná. Co se týče databáze MongoDB, ta umožňuje pružně pracovat s datovým schématem, aniž by bylo nutné měnit samotná data, nabízí vysoký výkon při provádění jednoduchých dotazů a vývojář může přidávat pole nebo sloupce do této databáze bez poškození struktury dat a výkonu systému.[6]

V případě této práce jsou důležitá kritéria: jednoduše měnit strukturu databáze a přidávat nové datové typy, pak rozšíření funkcí webové aplikace během vývoje, technická podpora a zkušenosti vývojářů systému řízení báze dat.

MySQL je tou správnou volbou pro každý objekt, který se může spolehnout na předdefinovanou strukturu a specifikovaná schémata. Oproti tomu MongoDB je skvělou volbou pro rychlerostoucí projekty bez specifické struktury. Pokud vývojář není schopen na začátku projektu definovat strukturu databáze nebo se projekt neustále mění, je dobré zvolit MongoDB, což odpovídá požadavkům na databázi webové aplikace této bakalářské práce. A proto je MongoDB vyhovující volbou pro tuto webovou aplikaci.[6]

2.1.3 Softwarové řešení pro tvorbu klientské a serverové části aplikace v jazyce JavaScript

První softwarové řešení pro tvorbu klientské a serverové části aplikace v jazyce JavaScript, které budu analyzovat, je kombinace jazyků HTML, CSS a JavaScript pro klientskou část a taky PHP pro serverovou část. Toto řešení pro klientskou část vyžaduje použití značkovacího jazyka HTML pro vykreslení struktury webové stránky, kaskádové styly CSS pro popis způsobu zobrazení elementů na webové stránce a skriptovací jazyk JavaScript pro vytváření interaktivní webové stránky, což znamená programování web stránky nebo aplikace samostatně, bez pomocných technologií, které zjednoduší práci programátora. Takové řešení je vhodné pro jednoduché stránky nebo malé webové aplikace. S použitím knihoven nebo frameworků lze řešit celou řadu problémů implementace, zabezpečení a testování aplikace, což vývoj aplikace zjednoduší a zrychlí.

Použití skriptovacího jazyka PHP (Personal Home Page) pro serverovou část webové aplikace má své výhody a nevýhody. Jednou z výhod PHP je obrovská kódová databáze pro všechny druhy řešení. Na rozdíl od jazyků Java nebo Python a dalších univerzálních programovacích jazyků byl PHP navržen speciálně pro web, a z toho důvodu obsahuje potřebné funkce pro práci s HTML, servery či databázemi. Nevýhodou PHP je to, že PHP následuje klasický model klient-server, kde žádost o stránku spouští aplikaci, připojuje se k databázi, zpracovává a vykresluje HTML. Díky tomu je PHP poněkud pomalejší než například platforma Node.js.[7]

JavaScriptové knihovny a frameworky, například knihovna React nebo framework Angular, umožňují vytváření komponent, které dělí webovou stránku na kousky

kódu, jež se snadněji ladí a případně jsou znovu použitelné na jiných stránkách. Při použití JavaScriptových knihoven nebo frameworků se stále uplatňují HTML, CSS a JavaScript a k tomu se nabízí možnost vytvářet vlastní komponenty.[8]

Další softwarové řešení pro tvorbu klientské a serverové části aplikace v jazyce JavaScript je s využitím JavaScriptových frameworků nebo knihoven pro klientskou část plus softwarové platformy Node.js a frameworku Express.js pro serverovou část.

Angular je JavaScriptový framework, který je napsaný v jazyce TypeScript a byl vyvinut společností Google. React je JavaScriptová knihovna vyvinutá společností Facebook a také slouží pro vytváření uživatelských rozhraní. React a Angular mají mnoho podobných prvků, ale také mnoho rozdílů. Jeden z nich je ten, že Angular slouží jako plnohodnotný framework MVC, zatímco React dává programátorovi mnohem více svobody a poskytuje „view“ v MVC, což znamená, že Model a Controller si vývojář musí vyřešit sám. Z tohoto důvodu si vývojář může vybrat libovolné vlastní knihovny. Každý projekt React je odlišný a vyžaduje vlastní strukturu a architekturu projektu.[8]

React dále také poskytuje zajímavé nástroje, jakými jsou nástroje pro testování komponent, knihovny pro přidání funkčnosti, do nichž například patří React-route pro směrování, Redux nebo MobX pro správu stavu aplikace. React používá virtuální DOM a Angular má také vlastní implementaci DOMu - incremental DOM. Díky těmto technologiím jsou Angular a React nesmírně rychlé. Vývojáři JavaScriptové knihovny React se rozhodli zkombinovat šablony uživatelského rozhraní a logiku JavaScriptu a výsledkem je „JSX“, který používá jazyk podobný jazyku XML. JSX umožňuje psát značky přímo do kódu JavaScriptu a usnadňuje vývoj, protože vše je na jednom místě. React vyžaduje pouze znalosti JavaScriptu, zatímco při práci s frameworkem Angular musí vývojář znát i jeho specifickou syntax. Jak už bylo zmíněno, React a Angular jsou oba založené na komponentách. Komponenta obdrží vstup a po nějaké interní logice vrátí vykreslenou šablonu uživatelského rozhraní (například přihlašovací formulář nebo tabulku) jako výstup. Komponenty by měly být snadno použitelné v rámci jiných komponent nebo projektů.[9]

Pomocí platformy Node.js a frameworku Express.js lze vytvářet vysoce škálovatelné serverové aplikace. Node.js byl vytvořen v roce 2009 Ryanem Dahlem ve společnosti Google a slouží jako softwarová platforma, založená na JavaScriptu V8.[10]

Node.js umožňuje vytvářet aplikace, které mohou zpracovávat více požadavků najednou pomocí fronty událostí JavaScriptu, jež maximalizují využití jednoho procesoru a paměti počítače při zpracování více požadavků než běžné servery. Díky této

funkci je Node.js vynikající volbou pro aplikace v reálném čase nebo pro ty, které vyžadují mnoho I/O operací. Jednou z hlavních výhod Node.js je to, že mnoho populárních JavaScriptových knihoven, jako jsou React nebo Vue, je napsáno v JavaScriptu, který je jazykem všech moderních prohlížečů. Mít jeden jazyk v klientské a serverové části webové aplikace je pro podporu aplikace a pro koordinaci mezi členy týmu velmi dobré. Node.js má také své nevýhody, k nimž patří například jeho architektura, která má určitá omezení, jako je nízká efektivita při vysokém zatížení CPU, ačkoliv se Node.js v konkurenčním zpracování mnoha požadavků, stále se mu nedaří s takovými operacemi, jakými jsou tvorba grafiky nebo zpracování obrazu. Koncept a implementace Node.js jsou skvělé pro aplikace v reálném čase.[10]

Na základě analýzy existujících řešení pro tvorbu klientské a serverové části aplikace byla pro klientskou část zvolena JavaScriptová knihovna React a pro serverovou část platforma Node.js a framework Express.js z toho důvodu, že knihovna React je založena na komponentách, což umožňuje snadnější ladění a použití stejných komponent na několika místech ve webové aplikaci a použití stejného jazyka v serverové a klientské části při implementaci webové aplikace je velmi dobré pro její podporu.

2.2 Dostupné webové aplikace pro plánování sportovního tréninku

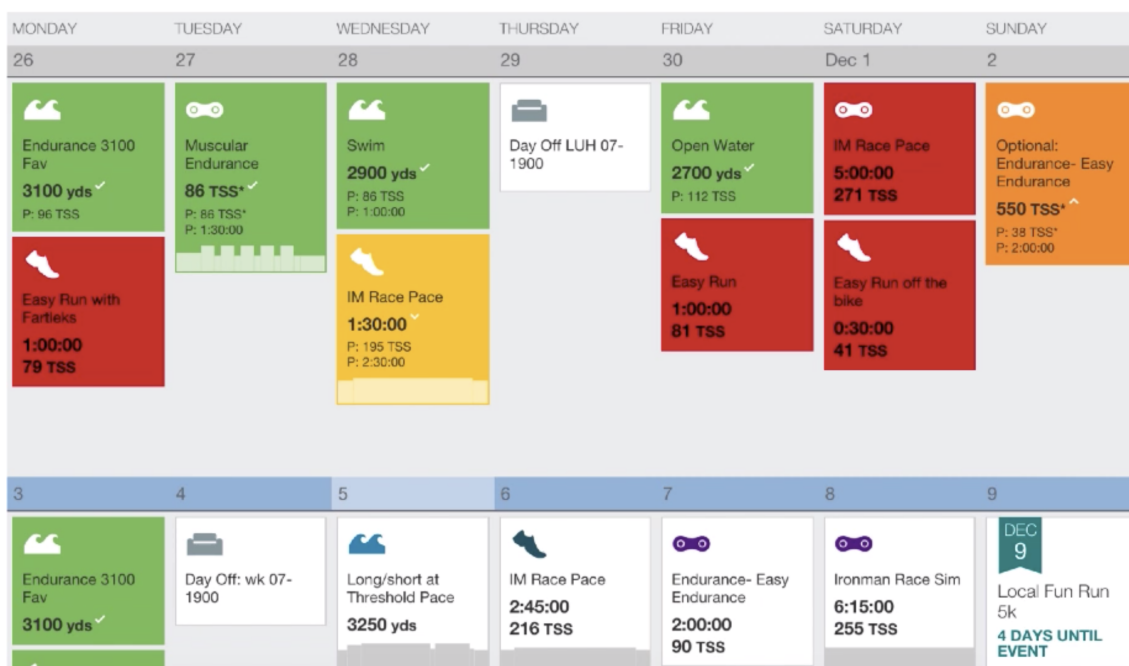
Většina lidí, která se zajímá o sport, v současné době používá pomocné softwary, z nichž každý má své vlastní požadavky k funkcím takového softwaru. Uživatelé těchto softwarů jsou velmi často profesionální závodníci a trenéři, kteří potřebují pohodlné a srozumitelné rozhraní pro sledování svých výkonů a pokroků, synchronizaci dat o trénincích z různých zařízení, plánování, sledování a analýzu tréninků a oboustrannou komunikaci. Mezi takové aplikace pro plánování sportovního tréninku se řadí například Trainingpeaks.com, Strava.com nebo Trainright.com.

Tato bakalářská práce je zaměřena na webovou aplikaci pro vytrvalostní sporty. Například aplikace pro cyklisty se dělí do několika kategorií. Jednou z nich jsou aplikace, které prostřednictvím navigace určují ujetou vzdálenost závodníka. Další takovou kategorií jsou aplikace, pomáhající plánovat trasu jízdy a obsahují příručky pro opravu kol. Tato kapitola na příkladu placených a neplacených aplikací popisuje podrobněji sportovní webové aplikace, kterými jsem se inspirovala.

2.2.1 Trainingpeaks.com

Na trhu existuje mnoho elektronických asistentů, které pomáhají trenérům a závodníkům vytvářet a reportovat tréninky, jako jsou webové a mobilní aplikace, pulsmetry, senzory aktivit atd. Vývojáři těchto softwarů používají vlastní algoritmy a v důsledku toho udržují své vlastní rozhraní pro analýzu sportovních dat.

Jedním z takových softwarů je Trainingpeaks.com, který nabízí mnoho zajímavých řešení jak pro závodníka, tak trenéra. TrainingPeaks.com je placená webová aplikace pro triatlon, cyklistiku, běh a jiné sporty. Každý závodník, který má stanovený určitý cíl, má díky této webové aplikaci rozhraní pro tyto účely. Závodník má možnost vybrat si hotový trénink nebo určitého trenéra, jenž mu připraví trénink na míru. Tyto tréninky se objeví v kalendáři závodníka a závodník má možnost komentovat průběh tréninku a také se svým trenérem komunikovat (obrázek 2.1). [11]



Obrázek 2.1: Trainingpeaks.com: Kalendář závodníka

Jednou z důležitých funkcí webové aplikace Trainingpeaks.com je to, že si závodník může synchronizovat, nahrávat nebo si přetahovat data tréninků z jiných webových nebo mobilních aplikací, které používá. Mezi hlavní výhody této webové aplikace oproti konkurenci patří, že podporuje import dat ze zařízení a programů od takových výrobců, jako jsou Garmin, Polar, Timex či Strava. Jak už jsem zmínila, webová aplikace Trainingpeaks.com obsahuje rozhraní i pro trenéra. Jednou z funkcí

trenérského rozhraní je, že trenér vidí seznam svých závodníků a jejich data na jednom místě (obrázek 2.2).[11]

Athlete	W	T	F	S	S	M	T	W	Alerts ?
Ricky Bobby	●	●	●	●	●	●	●	●	●
Cecil Cyclist	●	●	●	●	●	●	●	●	●
Max Power	●	●	●	●	●	●	●	●	●
Corredor Rápido	●	●	●	●	●	●	●	●	●
Emma WATTson	●	●	●	●	●	●	●	●	●

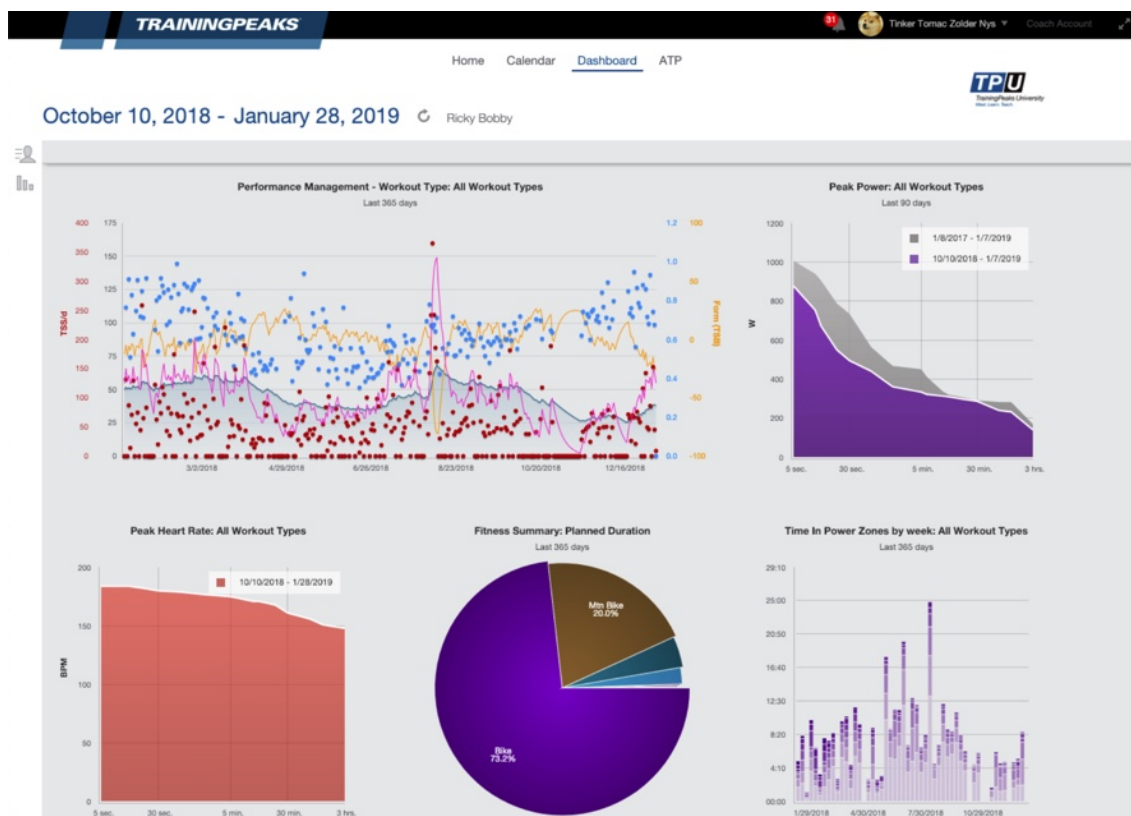
Obrázek 2.2: Trainingpeaks.com: seznam závodníků trenéra a jejich data

Tato funkce usnadňuje plánování a sledování průběhu tréninků a komunikaci se závodníky. Další zajímavou funkcí pro trenéry je možnost vytvářet strukturovaná cvičení, která se automaticky aktualizují do kalendáře určitého závodníka. Trenér si tato cvičení může uložit do seznamu cvičení a pak je snadno aplikovat do kalendáře každého závodníka. Funkce Performance management chart umožňuje ve formě grafu v průběhu cvičení monitorovat důležité hodnoty a zajistit, jestli závodník dosáhl svých cílů. Poslední avšak velice důležitou funkcí je, že webová aplikace Trainingpeaks.com analyzuje výsledné hodnoty tréninku (obrázek 2.3), jako jsou výkon, srdeční tep a další data. Tím pomáhá trenérovi vylepšit i trénink do budoucna.[11]

Měsíční předplatné této webové aplikace je pro závodníky 19,95 dolarů, čtvrtletní předplatné je za 16,33 dolarů měsíčně a celoroční předplatné za 9,92 dolarů měsíčně. Cena pro trenéra je stanovena podle počtu trénovaných závodníků. Do čtyř závodníků je cena 19 dolarů měsíčně plus jednorázový vstupní poplatek 99 dolarů. Při neomezeném počtu závodníků je cena 49 dolarů a taktéž vstupní poplatek 99 dolarů.[11]

2.2.2 Strava.com

Jedná se o americký projekt, který má jak webovou, tak mobilní verzi aplikace a je určený výhradně pro závodníky. Aplikace Strava je velmi oblíbená díky tomu, že kombinuje funkci tréninkového deníku, který zobrazuje všechny klíčové ukazatele, jakými jsou vzdálenost, rychlost, spotřebované kalorie, puls, intenzita a zátěž tréninku a také funkce GPS sledování a sociální sítě. Tato aplikace závodníkovi umožňuje zaznamenávat svou činnost, sdílet fotografie z průběhu tréninku a získávat hodnocení a komentáře od ostatních závodníků (obrázek 2.4).[12]

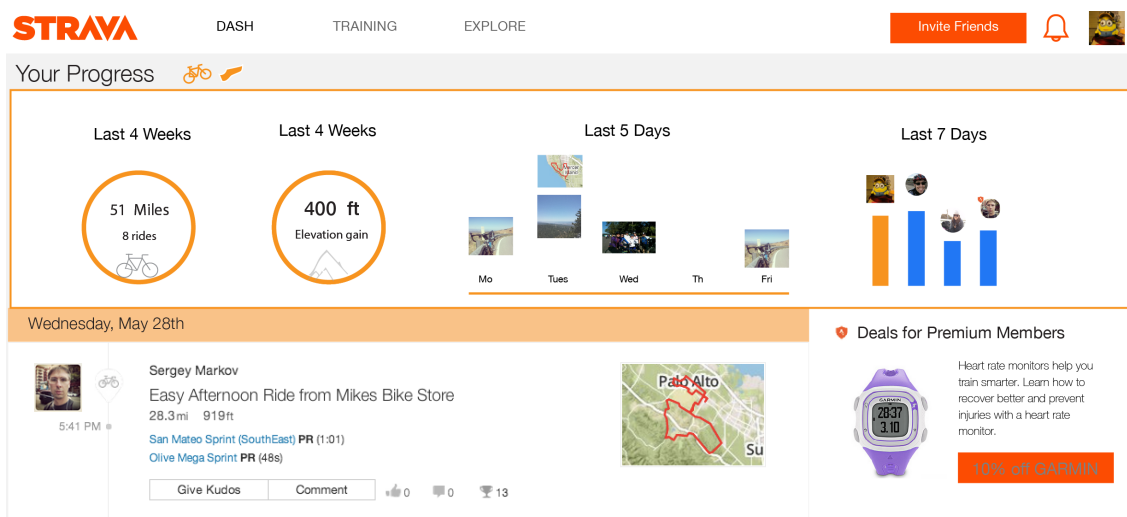


Obrázek 2.3: Trainingpeaks.com: analýza výsledných hodnot tréninku

Podle mého názoru je velkou konkurenční výhodou aplikace Strava to, že umožňuje synchronizaci dat o tréninku z jiných zařízení a obsahuje data z různých softwarů, například Zwift, Sunto, Amazfit, Fitbit, MIO, Polar, Tacx atd.

Další funkcí, kterou se Strava liší od konkurence, je Global Heatmap. Strava prostřednictvím své aplikace shromáždila informace o více než 100 milionech jízd na kole svých uživatelů. Tyto trasy jsou umístěny na celosvětové mapě, což umožňuje globální hodnocení sportu z různých zemí. Tato mapa živě ukazuje místa, kde žijí obyvatelé, kteří se věnují sportu. Na obrázku 2.5 je vidět mapa všech jízd na kole za období dvou let.[12]

Tato mapa se skládá ze 700 milionů činností, ze 7,7 bilionů rastrovaných pixelů, z 5 terabajtů vstupních dat, z celkové vzdálenosti 16 miliard kilometrů a z celkově zaznamenané aktivity činící 100000 let. Jedinou nevýhodou, kterou jsem v aplikaci Strava objevila, bylo to, že aplikace nenabízí žádné rozhraní pro trenéra a trenér nemá žádnou možnost vytvářet, analyzovat a ani jakkoliv řídit trénink. Co se týká cenové politiky, závodník může využít bezplatné verze, v nichž je většina funkcí vypnuta. Pokud závodník bude chtít používat veškeré funkce, které aplikace



Obrázek 2.4: Strava.com: praktická ukázka profilu webové aplikace

Strava nabízí, musí si zaplatit měsíční předplatné, pohybující se v řádech jednotek dolarů.[12]

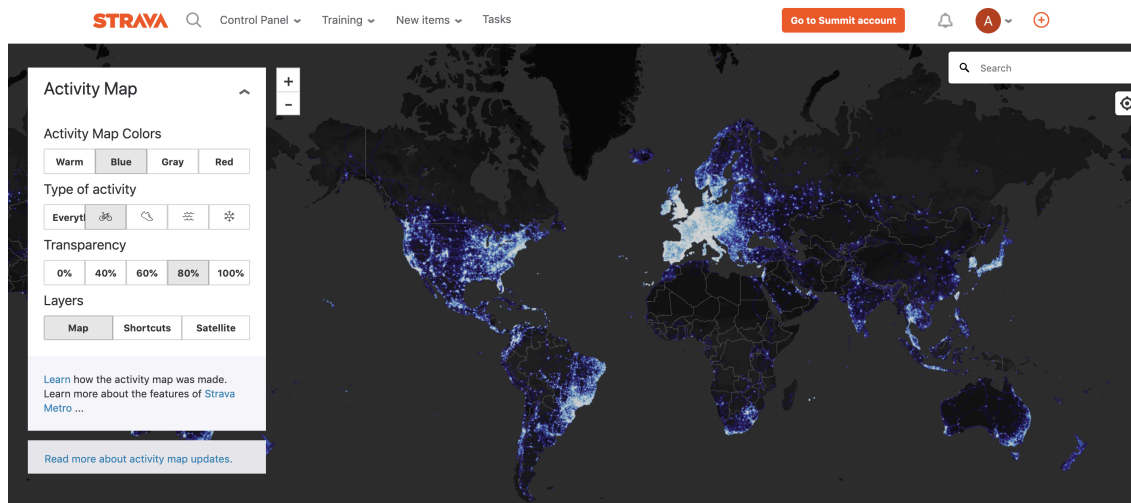
Podle mého názoru je aplikace Strava nejvíce propracovaná ze všech dostupných aplikací a zároveň pro závodníka obsahuje větší množství užitečných funkcí.

2.2.3 Trainright.com

Další aplikací pro plánování sportovního tréninku, kterou jsem se inspirovala, je Trainright.com, jež slouží jako informační portál pro závodníky.

Trainright.com nenabízí rozhraní pro závodníka ani trenéra, ale využívá rozhraní aplikace Trainingpeaks.com, kterou jsem zmiňovala již dříve. Jelikož je tato webová aplikace spíše informačního charakteru, trenéři a závodníci tak mají možnost se dozvědět hodně zajímavých informací o cyklistice, o různých sportovních akcích a táborech a také mají prostřednictvím této aplikace možnost si na tyto akce koupit vstup (obrázek 2.6).[13]

Její největší nevýhodou je podle mého názoru finanční náročnost a to, že pro účely plánování a analýzy tréninků využívá cizí rozhraní. Naopak největší výhodou oproti konkurenčním řešením je to, že Trainright.com funguje jako sociální síť, která sdružuje cyklistickou komunitu na základě pořádání různých sportovních akcí.



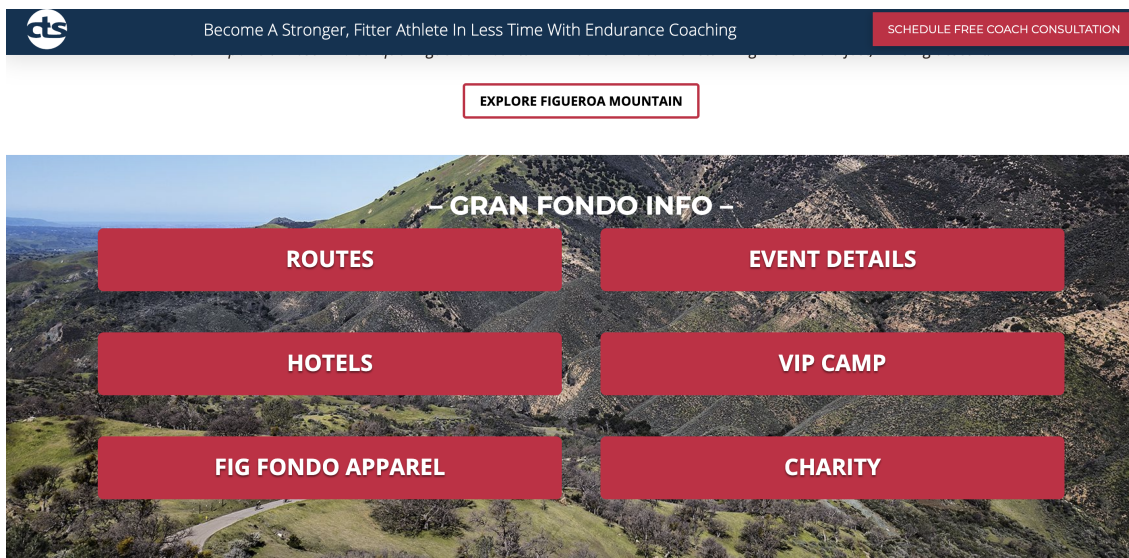
Obrázek 2.5: Strava.com: funkce Global Heatmap

2.3 Volba funkcí, technologií webové aplikace a motivace k bakalářské práci

Motivací pro tuto bakalářskou práci bylo tedy vytvořit aplikaci, která sportovcům a trenérům poskytne základní užitečné funkce a příjemné rozhraní pro práci, a která se případně stane základem sportovní open source aplikace.

Funkce webové aplikace této bakalářské práce byly navrženy na základě analýzy existujících sportovních webových aplikací a také na základě komunikace s konečným uživatelem sportovních aplikací, což je nejlepší způsob, jak zjistit praktické použitelné funkce pro jakýkoliv software. Dále jsou uvedeny funkce vlastního řešení webové aplikace, popis implementace těchto funkcí se nachází v kapitole „Webová aplikace“ této bakalářské práce. Trenér a sportovec mají v této webové aplikaci vlastní rozhraní na základě svých požadavků. Trenér má k dispozici Dashboard pro sledování informací o každém ze svých sportovců, to znamená: osobní údaje, statistiku trénování, komentáře od sportovce a výsledky každého tréninku. Také je pro něj připravený kalendář každého sportovce, kam si trenér může přidávat nové tréninky a sledovat výsledky tréninků ve formě grafu.

Rozhraní pro sportovce obsahuje Dashboard s informacemi sportovce o trénincích, statistice tréninků a detailnější popis každého tréninku, který byl naplánován určitým trenérem. Jednou z praktických, užitečných funkcí webové aplikace pro sportovce je synchronizace dat z aplikace Strava.com prostřednictvím importu dat ve formátu



Obrázek 2.6: Webová aplikace Trainright.com

gpx. Tato importovaná data webová aplikace analyzuje ve formě grafu pro lepší zjištění výkonnosti tréninku. Rozhraní pro sportovce také nabízí kalendář s tréninky pro jejich sledování a reportování.

Na základě zkoumání existujících technologií pro tvorbu webových služeb byla zvolena pro komunikace klienta a serveru – architektura REST, pro tvorbu klientské části webové aplikace – JavaScriptová knihovna React, pro tvorbu serverové části webové aplikace – platforma Node.js a framework Express.js, pro ukládání většího množství provozovaných dat – nerelační databáze MongoDB.

3 Webová aplikace

Tato kapitola detailně popisuje rozhraní pro trenéra a sportovce a dále technologie použité během vývoje, implementace a testování klientské a serverové části. Webová aplikace pro plánování sportovního tréninku je napsaná v programovacím jazyce TypeScript, který doplňuje JavaScriptová knihovna React a nástroj Context pro jednoduché uchování aktuálního stavu webové aplikace. Serverová část webové aplikace byla implementovaná prostřednictvím platformy Node.js, frameworku Express.js a pro manipulaci s daty byla použita nerelační databáze MongoDB.

3.1 Fundament aplikace

3.1.1 Vícejazyčnost aplikace

Webová aplikace této bakalářské práce uživateli nabízí rozhraní ve dvou jazycích, v angličtině a češtině. Tato funkce je implementována pomocí knihovny React Intl, která poskytuje React komponenty a API pro zpracovávání překladů dat, čísel a řetězců. Do projektu byla nainstalována knihovna React Intl, pak byl v komponentě importován nástroj IntlProvider z react-intl a soubory en.json a cz.json, které obsahují klíč a hodnotu pro každé slovo nebo větu, jež je potřeba přeložit. Pak je nezbytné vnořit komponentu, kterou je potřeba přeložit, do komponenty IntlProvider a pro každé slovo nebo větu určit klíč, podle něhož se bude hledat jeho překlad.

Aplikace se snadno rozšiřuje do dalších jazykových verzí, například pro španělskou verzi je nutné vytvořit soubor sp.jsou, který bude obsahovat klíč a překlad pro každé slovo v aplikaci, a poté je potřeba importovat tento soubor do hlavní komponenty aplikace App.tsx. Tímto způsobem je možné přidávat do aplikace další jazykové verze.

3.1.2 Klientská část

Webová aplikace je naprogramovaná v jazyce TypeScript.

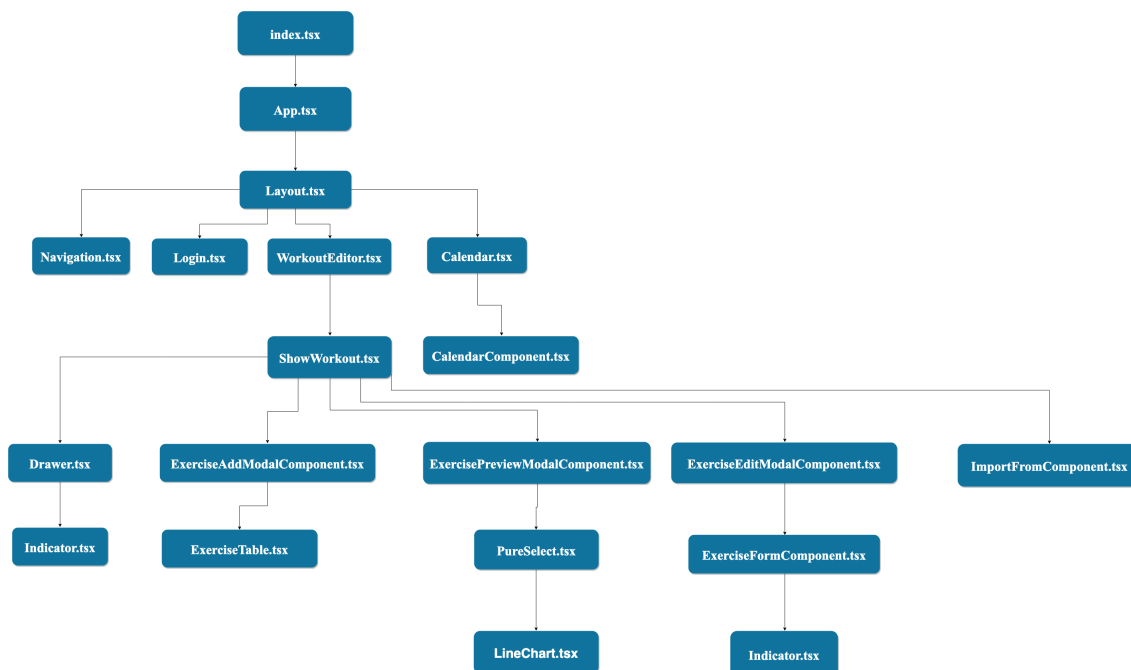
TypeScript je nadmnožinou JavaScriptu, která podporuje statické typování, třídy a interfacy. Výhodou použití JavaScriptu je to, že IDE nabízí větší možnosti pro vychytání chyb během napsání kódu. TypeScript je moderní JavaScript. Jedná se o včasné vychytávání chyb, což dělá vývoj efektivnější zároveň s použitím komunity JavaScriptu. TypeScript se velmi dobře integruje s knihovnamy JavaScript, čímž pomáhá k pohodlnějšímu programování.[14]

Pro implementaci webové aplikace byl použit editor Visual Studio Code, k němuž byla potřeba instalace různých balíčků a závislostí pro programovací jazyky, knihovny a databáze. Visual Studio Code podporuje programovací jazyk TypeScript, ale neobsahuje kompilátor TypeScript. V rámci implementace webové aplikace byl nainstalován kompilátor TypeScript lokálně do projektu prostřednictvím správce balíčků yarn. Po instalaci kompilátoru TypeScript je důležité vytvoření souboru tsconfig.json, který definuje nastavení JavaScriptu a možnosti kompilátoru. Tento soubor obsahuje „compilerOptions“, což jsou pravidla kompilace, a také „include“, který obsahuje informace o všech souborech, jež se kompilují do programovacího jazyka JavaScript. Dalším důležitým souborem, který byl vygenerován, je tslint.json. Tento soubor obsahuje sadu pravidel, jež při každém spuštění projektu nahlásí chybné použití rozšíření JavaScriptu, jako je například chybné použití statického typování nebo atributů objektově orientovaného programování. Pro zobrazení inteligentního doplnění kódu a různých informací jsem použila IntelliSense, což je nástroj editoru Visual Studio Code. Tento nástroj mi pomohl psát kód rychleji a správně.

Při implementaci klientské části webové aplikace byla použita JavaScriptová knihovna React. Knihovna React je založena na komponentách a každá komponenta vykreslí do prohlížeče pomocí metody render šablonu uživatelského rozhraní jako výstup. Některé komponenty, implementované v klientské části této webové aplikace, byly použity v rámci jiných komponent, což usnadňuje implementaci uživatelských rozhraní webové aplikace.[9]

Na obrázku 3.1 je ukázka struktury komponent JavaScriptové knihovny React, která byla implementovaná na základě požadavků této bakalářské práce.

Editor Visual Studio Code, jenž byl použit pro programování webové aplikace, podporuje JSX a jedná se o rozšíření syntaxe programovacího jazyka JavaScript. Komponenty knihovny React, které byly implementovány v klientské části webové



Obrázek 3.1: Struktura React komponent webové aplikace

aplikace, obsahují JSX. Při práci s tímto rozšířením bylo nutné změnit příponu souborů *.ts na příponu *.tsx.

Design webové aplikace je implementován prostřednictvím React UI knihovny Ant design, které obsahuje velkou sadu hotových komponent nebo jejich částí pro vytváření interaktivních uživatelských rozhraní. Tato knihovna je napsána v jazyce TypeScript[15], což je jednou z výhod pro webovou aplikaci této bakalářské práce.

Dalšími výhodami knihovny jsou její popularita – na github má kolem 50000 hvězd a také to, že komponenty, podle mého názoru, jsou pro konečného uživatele sympatické a pohodlné k použití. React UI Knihovna Ant design nabízí široký výběr komponent pro vytváření uživatelského rozhraní. V této práci byly použity další komponenty nebo jejich části: menu, tabulky, selecty pro jazyk, data, pak přihlašovací formulář, kalendář, tlačítka atd. Práce s React UI knihovnou Ant byla snadná. Bylo zapotřebí instalovat UI knihovnu do projektu a pak pro použití komponent UI knihovny v hlavičce každé vlastní komponenty importovat použitý element komponenty React UI knihovny Ant.

Pro uchování aktuálního stavu webové aplikace byl použit React nástroj Context, který poskytuje způsob předávání dat přes strom komponent bez ručního předávání Props na každé úrovni. Z důvodu možnosti skládání React komponent do sebe se používá Props, což je prostředek pro předávání dat mezi kompen-

tami, který funguje jen jednosměrně, směrem shora dolů od komponenty rodič ke komponentě potomka. Tato metoda je občas příliš náročná pro Props, přenášených do mnoha komponent ve webové aplikaci. Tento problém právě řeší React nástroj Context, použitý v této práci. Pro použití nástroje Context ve webové aplikaci bylo potřeba nainstalovat Context do projektu a pak v komponentě, jež obsahuje užitečné Props pro jinou komponentu, byla vnořena komponenta Context.Provider, která definuje, jaká data bude předávat a jaké komponentě. Pak je v komponentě, v níž potřebujeme použít Props, definována komponenta Context.Consumer, která přijímá Props a pak s nimi pracuje.[9]

3.1.3 Serverová část

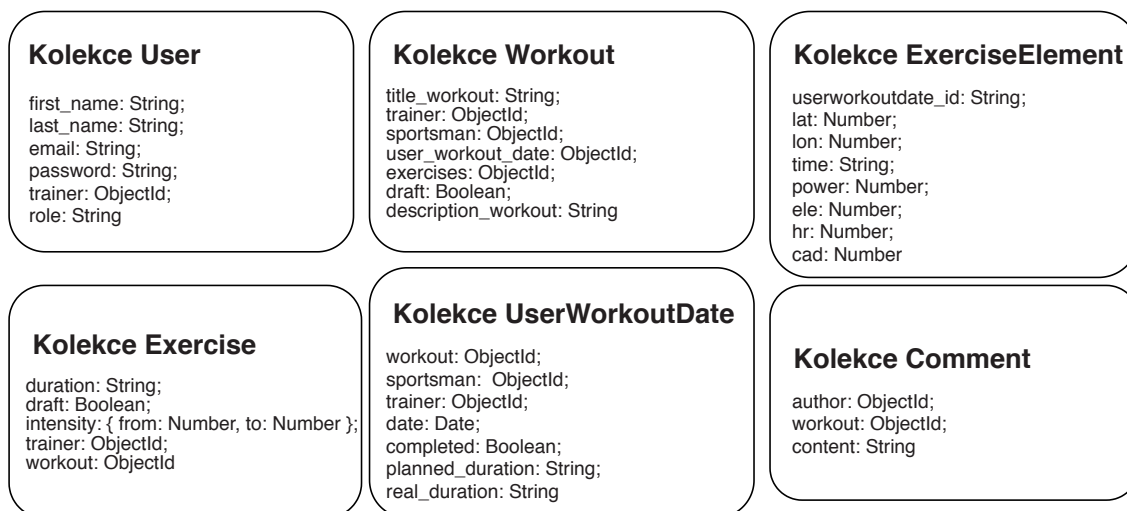
Implementace serveru webové aplikace proběhlo v softwarovém prostředí Node.js a skládá se z node modules — potřebná složka, která obsahuje všechny moduly použité během práce s Node.js, pak z middleware — součást serveru, která kontroluje oprávnění uživatele a obsahuje logiku práce s tokenem. Algoritmus autorizace uživatele pomocí tokenu je popsán v podkapitole „Testování a bezpečnost webové aplikace“, z models — soubory určené pro modely kolekcí dokumentů databáze MongoDB. S modely pracuje MongoDB pomocí modulu Mongoose a taky z routes — obsahuje soubory s logikou chování serveru, což jsou metody komunikace serveru s klientskou částí webové aplikace.

3.2 Datové úložiště

Pro uložení dat byla použita nerelační databáze MongoDB. Důvodem k jejímu použití byl fakt, že během implementace webové aplikace se neustále měnila struktura kolekcí databáze a při dalším rozšiřování funkcí této webové aplikace pak bude snadné přidat nové datové typy, kolekce a dokumenty MongoDB.

MongoDB umožňuje ukládat a přidávat dokumenty do kolekcí bez změny celé struktury modelu databáze, což dělá práci s daty pro programátora pohodlnější. Pro jednodušší přístup k objektům v MongoDB v této bakalářské práci byla použita JavaScriptová knihovna Mongoose, která podporuje nástroje pro CRUD operace.[6]

Model MongoDB této webové aplikace zahrnuje kolekce, do nichž se ukládají data o trenérovi, sportovci, naplánovaných a odtrénovaných trénincích včetně jejich výsledků (obrázek 3.2).



Obrázek 3.2: Model MongoDB

Struktura dokumentu kolekce User obsahuje atributy jméno a příjmení uživatele, emailovou adresu a šifrované heslo, používané pro ověřování přístupu uživatele do webové aplikace, dále pak atribut role pro rozpoznání uživatele závodníka či trenéra, pak id trenéra – tento atribut je určen pro vyhledávání seznamů sportovců určitého trenéra.

Jednou z pomocných funkcí pro trenéra je zde možnost vytváření vlastní knihovny tréninků. Tato funkce je implementována pomocí kolekce Workout. Struktura dokumentu této kolekce obsahuje atribut draft datového typu Boolean. Atribut draft se nastaví na hodnotu true, pokud je entita určená pro knihovnu tréninků, nebo naopak na hodnotu false, pokud entita je reálný trénink pro sportovce určený trenérem.

Struktura dokumentu kolekce Exercise se používá pro manipulaci se cvičeními, ze kterých se trénink skládá, a obsahuje následující atributy: doba cvičení, intenzita cvičení (hodnota intenzity od a do), id trenéra a id tréninku.

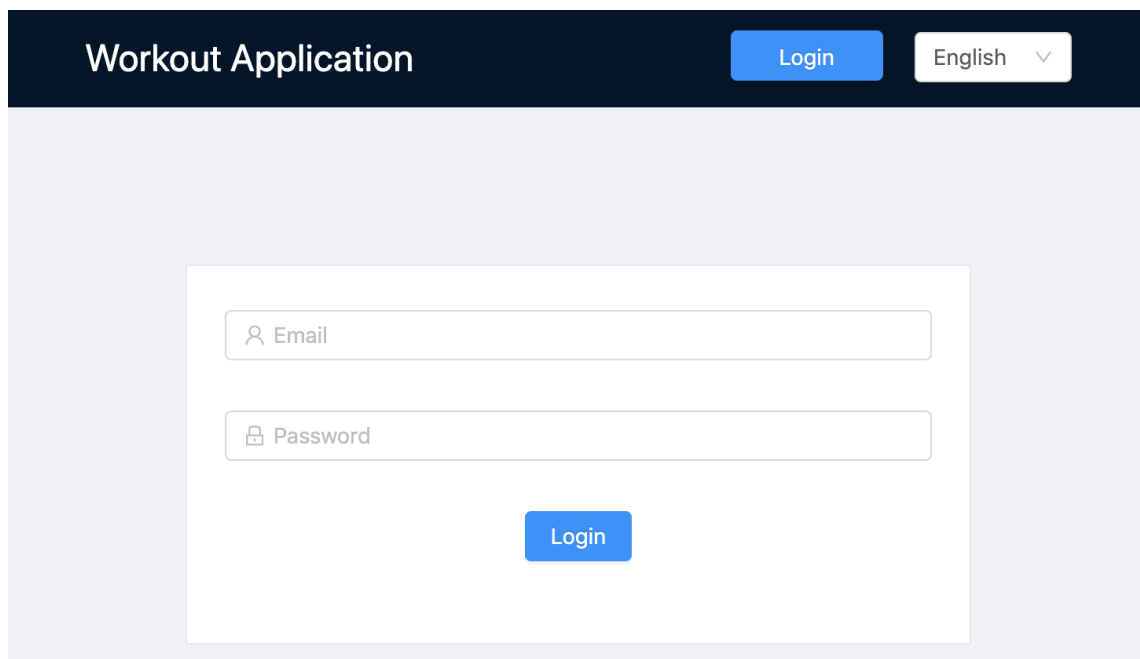
Kolekce Workout se užívá pro manipulaci s tréninky a její struktura obsahuje další atributy: název tréninku, id trenéra, který ten trénink naplánoval, id sportovce, jenž ten trénink bude trénovat, pole cvičení, ze kterých se skládá trénink, draft, popis – poznámku od trenéra pro sportovce. Atribut id userworkoutdate je určen pro funkce mazání tréninku z kalendáře trenéra či sportovce.

Pro jednodušší manipulaci s tréninky v kalendáři trenéra a sportovce je určená kolekce UserWorkoutDate. Struktura dokumentu této kolekce obsahuje atributy id tréninku, id sportovce, id trenéra a datum tréninku.

Webová aplikace také nabízí funkce importu souborů ve formátu gpx. Pro tuto

funkci je v databázi webové aplikace určená kolekce ExerciseElement, do které se ukládají dokumenty s atributy: lat, lon, time, power, ele, hr, cad. To jsou výsledné hodnoty tréninku sportovce, z nichž se skládá gpx soubor. Atribut userworkoutdate-id v této kolekci je určen pro spojení těchto výsledných dat tréninku s určitým tréninkem v kalendáře trenéra a sportovce.

Poslední kolekce MongoDB webové aplikace je Comment, která je určena pro uchovávání zpráv sportovce a trenéra ohledně tréninku. Dokumenty této kolekce obsahují atributy id tréninku, id autora zprávy a zprávu.

The image shows a screenshot of a web application's login page. At the top, there is a dark blue header with the text "Workout Application" on the left, a blue "Login" button in the center, and a white "English" dropdown menu on the right. Below the header is a light gray background. In the center, there is a white rectangular form with a rounded border. Inside the form, there are two input fields: the top one is labeled "Email" with a magnifying glass icon, and the bottom one is labeled "Password" with a lock icon. Below these fields is a blue "Login" button.

Obrázek 3.3: Přihlašovací formulář

3.3 Realizované rozhraní webové aplikace

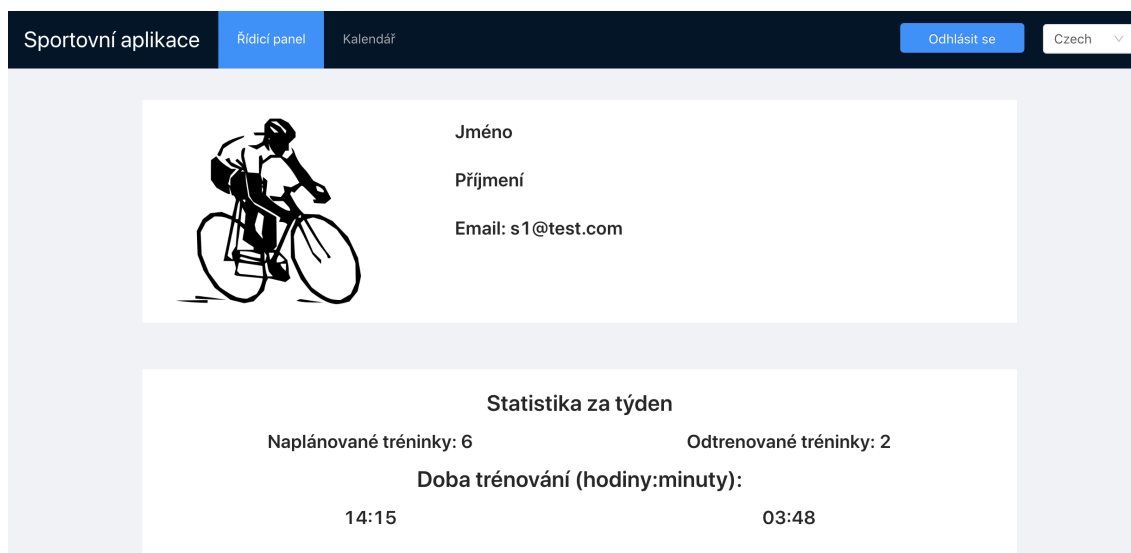
Uživatelé této webové aplikace se dělí na trenéry a sportovce. Každý z uživatelů má přístup k určitým funkcím a stránkám webové aplikace. Cílem práce bylo navrhnout a implementovat praktickou webovou aplikaci pro plánování sportovního tréninku, kde mají trenéři i sportovci k dispozici rozhraní podle svých potřeb. První stránka, která se uživateli po spuštění aplikace zobrazí, je přihlašovací formulář (obrázek 3.3). Ten umožňuje přístup do webové aplikace prostřednictvím zadání emailové adresy a hesla.

V případě shody uživatelských údajů vygeneruje server JSON Web Token, což je standard, který definuje způsob bezpečného přenosu informací mezi klientem a serverem ve formátu objektů JSON[16]. Spolu s informacemi o uživateli pošle server klientovi zprávu. Pokud dojde k přesměrování uživatelem na jinou stránku, aplikace pochopí, že se jedná o autorizovaného uživatele a nevyžaduje po něm opětovné přihlášení. Klient pošle serveru dotaz, který obsahuje JSON Web Token uložený do prohlížeče. Server dešifruje data JSON Web Tokenu, porovná je s daty existujících uživatelů v databázi aplikace a pak klientovi pošle odpověď s informacemi o tom, že takový uživatel již existuje a nasměruje ho na jím požadovanou stránku. Pokud uživatel zadá emailovou adresu či heslo, které nejsou v databázi, webová aplikace nahlásí uživateli, že emailová adresa nebo heslo nejsou správné. V opačném případě, pokud přihlášení proběhlo úspěšně, webová aplikace nahlásí uživateli, že emailová adresa a heslo jsou správné a povolí přístup do webové aplikace.

Podle zvolené role uživatele — trenér nebo sportovec — vygeneruje webová aplikace funkce, které bude nabízet.

Trenérovi nabízí aplikace funkci „Řídicí panel“, která poskytuje trenérovi seznam všech jeho sportovců, s možností prohlížení kalendáře jednotlivých sportovců s naplánovanými a odtrénovanými tréninky, a také s možností prohlížení profilu s osobními údaji sportovců, statistikou trénování za týden, naplánované a odtrénované tréninky. Do kalendáře určitého sportovce má trenér možnost přidávat trénink na libovolné datum prostřednictvím vytvořené knihovny tréninků nebo vytvořením nového tréninku. Pomocí chatu může také se sportovcem komunikovat. Další zajímavá funkce je „Kalendář“. Trenér může v jednom kalendáři sledovat tréninky všech sportovců.

Sportovcům nabízí aplikace funkci „Řídicí panel“. Prostřednictvím této funkce sportovec má možnost sledovat svou vlastní týdenní statistiku trénování, může si prohlížet odtrénované a naplánované tréninky. Funkce „Kalendář“ nabízí sportovci rozhraní, ve kterém může sledovat své trenérem naplánované tréninky a po rozkliknutí tréninku se mu objeví stránka s podrobnými informacemi o tréninku a možnosti přidání výsledků tréninku pomocí funkce importu souboru ve formátu gpx. I sportovec zde má k dispozici nástroj pro komunikaci s trenérem, a prostor, kam si může zapsat své pocity a poznámky z tréninku. Následující podkapitoly popisují detailně funkce realizované ve webové aplikaci.



Obrázek 3.4: Profil sportovce: osobní údaje a týdenní statistika trénování

Naplánované tréninky:

Název	Popis	Datum tréninku	Akce
Cycling training 25 km	From Liverec to Frydland	August 22, 2019	Přehled tréninku Vymazat
Cycling training 40 km	Competition preparation	August 23, 2019	Přehled tréninku Vymazat
Cycling training 5 km	Mountain riding	August 24, 2019	Přehled tréninku Vymazat

Obrázek 3.5: Profil sportovce: naplánované tréninky

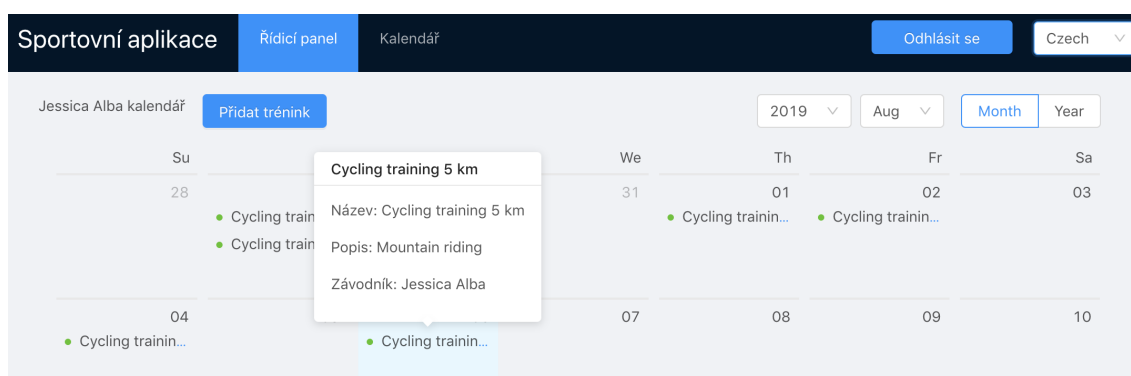
3.3.1 Profil sportovce

Trenér má k dispozici profil každého sportovce (obrázek 3.4), v něm vidí osobní údaje sportovce — jméno, příjmení, emailovou adresu, dále týdenní statistiku trénování, tj. dobu trénování a počet naplánovaných a odtrénovaných tréninků. Pro lepší sledování výsledků sportovce zde aplikace trenérovi nabízí tabulku odtrénovaných tréninků (obrázek 3.6) s možností detailního prohlížení výsledků formou grafů a komentářů od sportovce, na které má trenér možnost odpovědět. Dále profil sportovce nabízí trenérovi možnost sledovat naplánované tréninky v tabulce „Naplánované tréninky“ (obrázek 3.5).

Odtrenované tréninky:

Název	Popis	Datum tréninku	Akce
Cycling training 10 km	In Liberec	August 20, 2019	Přehled tréninku Vymazat
Cycling training 5 km	Mountain riding	August 19, 2019	Přehled tréninku Vymazat
Cycling training 10 km	In Liberec	August 14, 2019	Přehled tréninku Vymazat

Obrázek 3.6: Profil sportovce: odtrenované tréninky



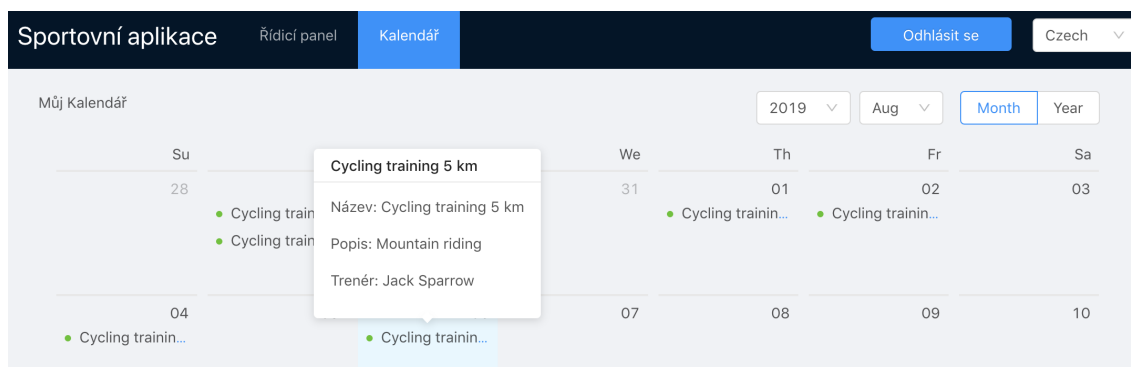
Obrázek 3.7: Kalendář tréninků: rozhraní pro trenéra

3.3.2 Kalendář tréninků

Funkce „Kalendář tréninků“ je k dispozici pro oba typy uživatelů (obrázek 3.7, obrázek 3.8). Trenér má k dispozici dva typy kalendáře, prvním je kalendář se všemi tréninky všech sportovců pro jednoduchý přehled všech akcí, které se budou konat v určitém časovém intervalu (např. tento týden nebo měsíc) a druhým je kalendář určitého sportovce, do kterého má trenér možnost přidávat tréninky a kontrolovat jejich výsledky.

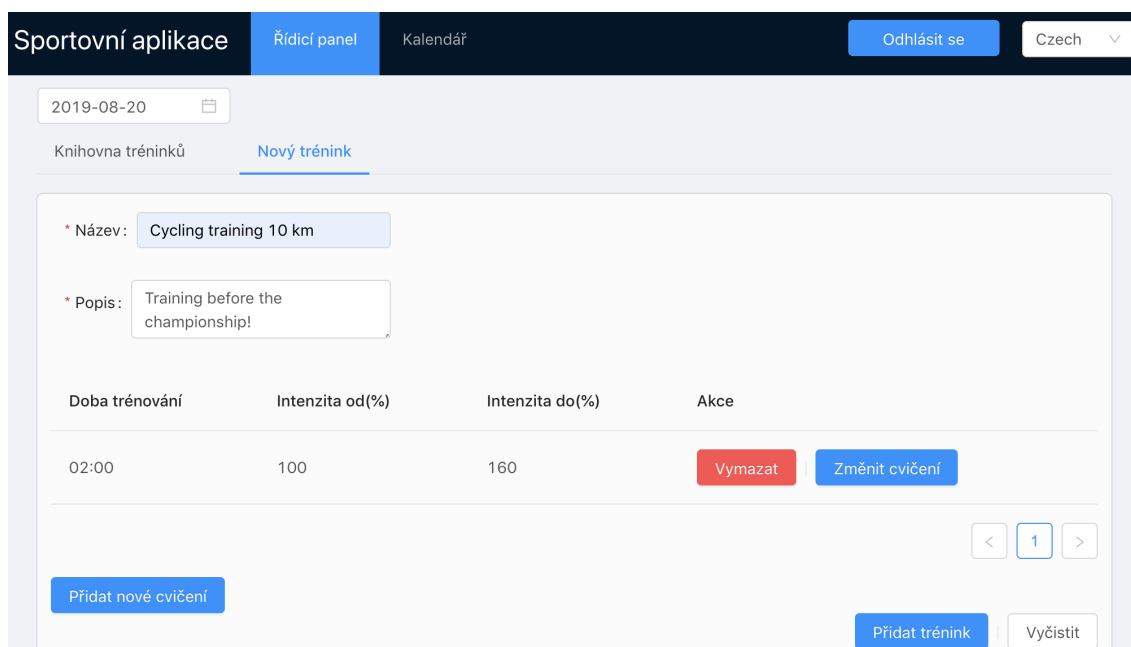
3.3.3 Přidání tréninků

Funkce plánování sportovních tréninků je v této webové aplikaci implementována dvěma způsoby. První možností je vytvoření nového tréninku prostřednictvím editoru (obrázek 3.9). V tomto případě trenér v kalendáři daného sportovce musí zvolit datum, název, popis tréninku a vytvořit sadu cvičení, ze kterých se trénink skládá.



Obrázek 3.8: Kalendář tréninků: rozhraní pro sportovce

Cvičení obsahuje dobu trénování a intenzitu cvičení, což je hodnota udávaná v procentech. Po zadání všech informací o tréninku a stisknutím tlačítka „přidat trénink“, se nový trénink objeví v kalendáři sportovce.

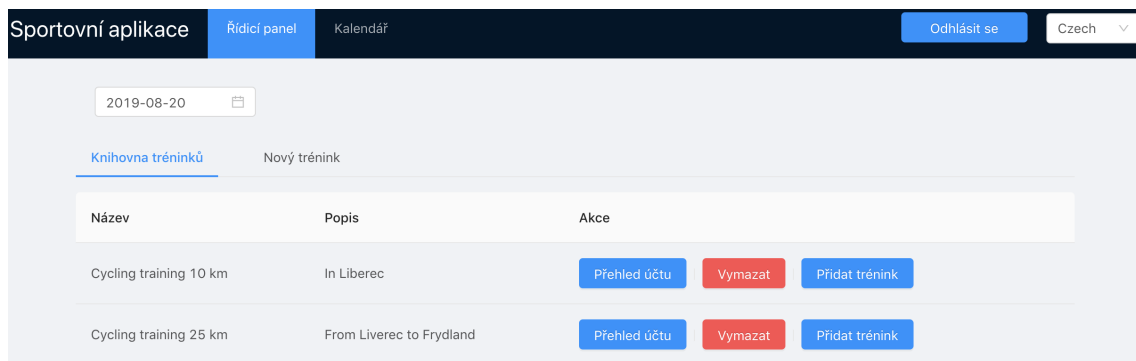


Obrázek 3.9: Přidání tréninků: editor tréninků

Druhou možností je vytvoření vlastní knihovny tréninků trenérem (obrázek 3.10). Vytvořené tréninky pak přidává do kalendářů jednotlivých sportovců.

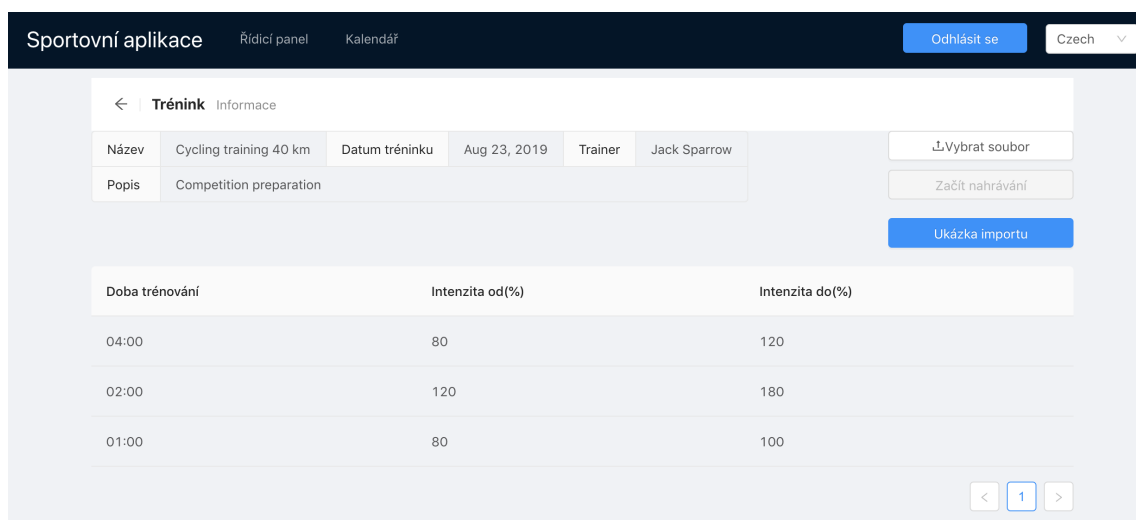
3.3.4 Analýza výsledků tréninků sportovce

Funkce analýzy výsledků tréninků sportovce v této webové aplikaci byla implementovaná importem gpx souborů (obrázek 3.11) se zaznamenanými hodnotami z průběhu



Obrázek 3.10: Přidání tréninků: knihovna tréninků

jeho tréninku.




Obrázek 3.11: Rozhraní pro sportovce: pohled na trénink a funkce importu výsledků do aplikace

Při implementaci importu souborů byla použita knihovna Xml.js. Tato knihovna slouží pro převedení dat z formátu XML do formátu JSON a také k pasování těchto dat (obrázek 3.12).

Gpx je soubor, který je pro tuto práci vygenerovaný prostřednictvím webové aplikace Strava.com a obsahuje výsledná data cvičení, z nichž se skládá trénink. Knihovna Xml.js transformuje tato data do formátu JSON a metoda parseGPX provede rozbor těchto dat.

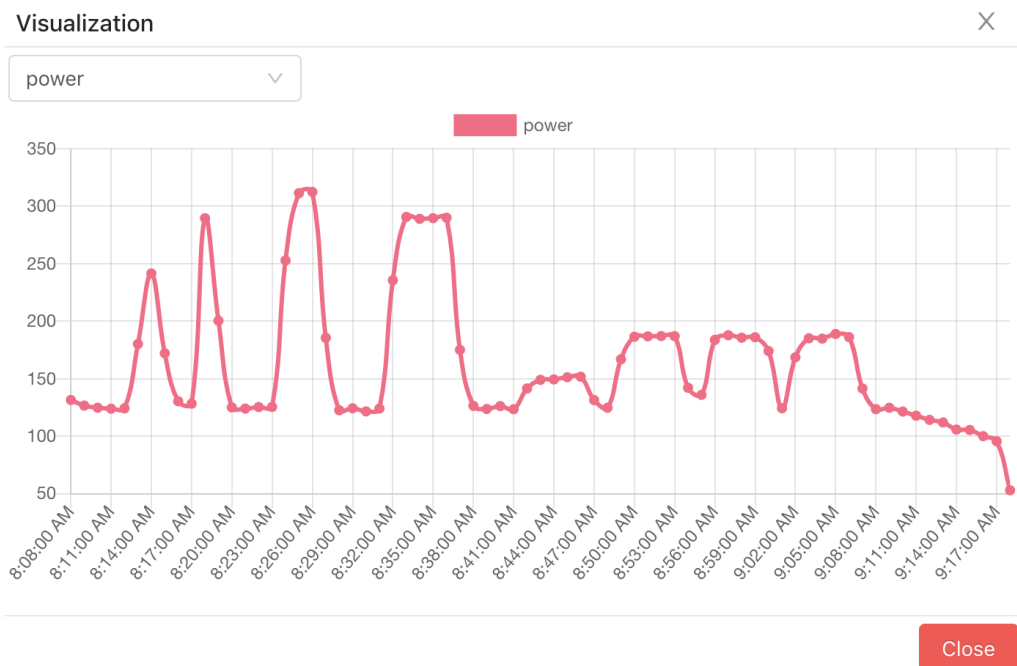
Gpx soubor obsahuje data z každé vteřiny tréninku. Pro analýzu výsledků tréninku je potřeba spočítat střední hodnotu pro každou minutu tréninku. Tato hodnota se vypočítá sečtením každých 60 vteřin tréninku a vydělí se číslem 60, což je počet

<p>Data ve formátu XML souboru gpx</p> <pre> <trkpt lat="-11.6369800" lon="166.9495940"> <ele>1.4</ele> <time>2019-03-02T08:08:16Z</time> <extensions> <power>163</power> <gpctx:TrackPointExtension> <gpctx:hr>87</gpctx:hr> <gpctx:cad>37</gpctx:cad> </gpctx:TrackPointExtension> </extensions> </trkpt> </pre>		<p>Data ve formátu JSON po parsování</p> <pre> { "_id": "ObjectId('5d5053735497d341e50832f4')", "userworkoutdate_id": "5d50403228f8803db51e90c0", "time": "March 2, 2019 9:08 AM", "power": 131.545454545455, "lat": -11.6361532272727, "lon": 166.950464977273, "ele": 1.52727272727273, "hr": 103.340909090909, "cad": 69.6818181818182, "_v": 0 } </pre>
---	---	--

Obrázek 3.12: Převod dat z formátu XML do JSON

těchto hodnot za minutu tréninku. Tímto způsobem jsou počítány střední hodnoty každé minuty tréninku, které se pak zobrazí na grafu.

Graf výsledných hodnot tréninku je realizován pomocí šablony Line knihovny Chart.js. JavaScriptová knihovna Chart.js je nástroj umožňující snadno vytvářet různé typy grafů. Šablona Line generuje spojovací čárový graf (obrázek 3.13), který uživatelům při přejetí myši na určitou hodnotu umožňuje pozorovat každou hodnotu tohoto grafu zvlášť a také umožňuje pozorovat křivku změny těchto hodnot.



Obrázek 3.13: Graf výsledných hodnot tréninku

Klientská část webové aplikace obsahuje React komponentu LineChart.tsx pro vykreslení grafů a taky komponentu PureSelect.tsx pro zvolení elementu (tagu),



Obrázek 3.14: Funkce komunikace trenéra a sportovce

podle něhož se pak generuje graf určitých hodnot gpx souboru.

3.3.5 Komunikace trenéra a sportovce

Reportování průběhu tréninku, zapisování poznámek a pocitů sportovcem je důležitou součástí úspěšného tréninku. Tato funkce může pomoci trenérovi sportovce lépe poznávat a plánovat jim tréninky na míru. Proto byla v této práci implementována funkce komunikace trenéra a sportovce (obrázek 3.14). Funkce je implementována pomocí React komponenty `CommentComponent.tsx`. Zprávy se ukládají do kolekce `Comment` databáze MongoDB.

3.4 Bezpečnost webové aplikace

Jedním z důležitých kroků při implementaci webové aplikace bylo soustředit se na bezpečnost systému. Efektivním způsobem ochrany proti útoku CSRF je token. Token je šifrovaná sada bajtů a ochrana spočívá v ověření tokenu, který server vygeneroval, a tokenu, odeslaného uživatelem. Webová aplikace používá protokol JSON Web Token k ověření uživatele při vstupu do aplikace prostřednictvím přihlášení. Trenér nebo sportovec prostřednictvím přihlašovacího formuláře odešle na server webové aplikace soukromé údaje, jako jsou emailová adresa a heslo. Server pak v případě shody emailové adresy a hesla s údaji v databázi aplikace vygeneruje JSON Web Token protokol, který obsahuje údaje uživatele. Tento protokol server pošle

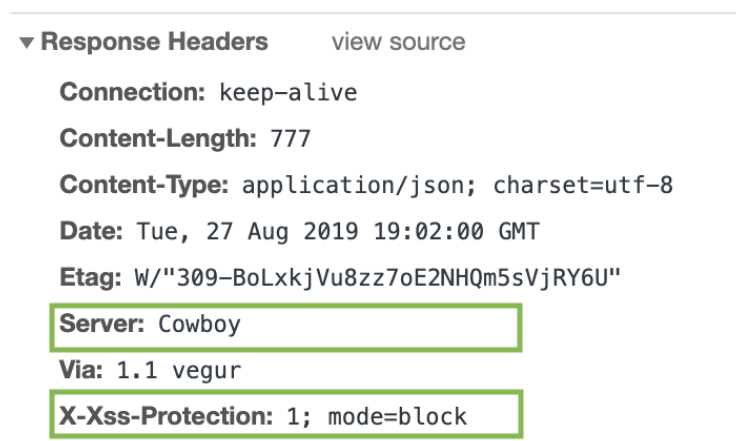
klientovi. Při směrování na jinou stránku aplikace, pomocí protokolu JSON Web Token, který uživatel posílá serveru, pochopí, že se jedná o autorizovaného uživatele a nevyžaduje po něm opětovné autorizování a server pak uživateli povolí přejít na požadovanou stránku.

Pro uchovávání JSON Web Tokenu ve webové aplikaci byl použit modul `express-session`. Tento modul uchovává data JSON Web Tokenu a cookies v serverové části aplikace vytvořením nové kolekce v MongoDB – `session`. V prohlížeči se tak ukládá pouze id dokumentu MongoDB, čímž se vylučuje možnost krádeže informací z cookies a JSON Web Tokenu. Navíc je doba platnosti cookies omezena na 15 minut, což snižuje riziko krádeže cookies a útoků na aplikaci skrze ukradená data.

Zajímavou alternativou k modulu pro obsluhování cookies dat je modul `cookies-session`. Tento modul lze použít pouze v případě, že cookies data jsou relativně malá a lze je převést na elementární hodnoty snadněji, než na objekty. Je důležité si uvědomit, že soubory cookie jsou pro klienta viditelné, a z toho důvodu by měly být v této webové aplikaci chráněny nebo skryty. Proto byl zvolen modul `express-session`.

Na ochranu webové aplikace proti útokům byl použit nástroj Helma, který před některými známými typy útoků pomáhá chránit pomocí konfigurace hlavičky HTTP.

Jako ochrana proti XSS byla použita funkce `xssFiltr` nástroje Helmet, která aktivuje funkce `X-XSS-Protection` ve většině moderních prohlížečů a také funkce `hidePoweredBy`, která ukrývá informaci o tom, že server aplikace je napsaný prostřednictvím frameworku `Express.js`. To vylučuje možnost určitých útoků přímo na tento framework. Na obrázku je uveden příklad HTTP odpovědi s implementovanými funkcemi Helmet (obrázek 3.15).



Obrázek 3.15: Záhlaví odpovědi HTTP

Tento nástroj má i další zajímavé funkce určené proti útokům na webové aplikace, jako např. funkce hpkp, která přidává v záhlaví Public Key Pinning, aby zabránil útokům pomocí falešných certifikátů, hsts nastavuje záhlaví, Strict-Transport-Security nastavuje ochranu připojení k serveru (přes HTTP pomocí SSL/TLS), další funkce noCache nastavuje záhlaví Cache-Control a také záhlaví Pragma tak, aby zakázalo na straně klienta ukládání do cache, a v neposlední řadě také funkce frameguard, která nastavuje záhlaví X-Frame-Options na ochranu proti clickjackingu, což je typ útoku na aplikaci, při kterém uživatel provede nějakou činnost na stránce a spustí akci, kterou nepředpokládal.

4 Závěr

Cílem této práce bylo realizovat praktické webové prostředí, které umožňuje trenérům pohodlné plánování tréninků a zároveň umožňuje sportovcům číst a reportovat své tréninky. Podle mého názoru je tento cíl splněn. Kromě základních funkcí, které byly požadované na začátku práce, nabízí webová aplikace další užitečné a zajímavé prostředí pro trenéra a sportovce.

V aplikaci je implementován editor pro vytváření tréninků, navíc si trenér může vytvořit vlastní knihovnu tréninků, což mu pomáhá svou práci zefektivnit. Sportovec má možnost reportovat a psát si poznámky ke svému tréninku, a i trenér může prostřednictvím chatu sdílet svou zpětnou vazbu. Webová aplikace nabízí i další zajímavé možnosti, například analýzu výsledků tréninku ve formě grafu nebo týdenní statistiku trénování. Tyto funkce jsou uvedeny v podkapitole „Realizované rozhraní webové aplikace“.

Jedním z důležitých kroků napsání této práce byla analytická část, která proběhla ještě před samotným vývojem webové aplikace a je popsána v kapitole „Rešerše“. Předmětem analytické části bylo zkoumání problematiky programování webových služeb, zvolení nejvhodnějších technologií pro vývoj webové aplikace, a také zkoumání existujících webových řešení, což pomohlo navrhnout vlastní řešení webové aplikace.

Do budoucna se plánuje přidat nové funkce jako jsou: např. automatická synchronizace dat s populárními aplikacemi, které používají sportovci, dále pak zlepšení funkce analýzy výsledků tréninků a také komunikace mezi sportovci pro vytvoření přátelské sportovní komunity v rámci této webové aplikace.

Seznam použité literatury

- [1] M. Kalin. *Java Web Services: Up and Running*. 1005 Gravenstein Highway North, Sebastopol, CA: O'Reilly Media, Inc., 2009. ISBN: 9780596521127.
- [2] R. Anderson. *SoapUI Cookbook*. Birmingham B3 2PB, UK.: Packt Publishing Ltd., 2015. ISBN: 9781784394219.
- [3] L. Richardson, S. Ruby a M. Amundsen. *RESTful Web APIs*. O'Reilly Media, Inc., 2013. ISBN: 9781449358068.
- [4] C. J. Date. *The new relational database dictionary*. 2016. ISBN: 9781491951736.
- [5] G. Harrison. *Next Generation Databases NoSQL, NewSQL, and Big Data*. 2015. ISBN: 9781484213308.
- [6] MongoDB. *MongoDB and MySQL Compared*. URL: <https://www.mongodb.com/compare/mongodb-mysql>. (cit. 2019-15-07).
- [7] D. Sklar. *Learning PHP*. O'Reilly Media, Inc., 2016. ISBN: 9781491933572.
- [8] S. Seshadri. *Angular: Up and Running*. O'Reilly Media, Inc, 2018. ISBN: 9781491999837.
- [9] A. Banks a E. Porcello. *Learning React: functional web development with React and Redux*. Sebastopol, CA: O'Reilly Media, 2017. ISBN: 9781491954621.
- [10] V. Subramanian. *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*. Berkeley, California: Apress, 2017. ISBN: 9781484226520.
- [11] Trainingpeaks. *Be Prepared on Race Day*. URL: <https://www.trainingpeaks.com/athlete-features/>. (cit. 2019-17-07).
- [12] Strava. *Building the home for your active life*. URL: <https://www.strava.com/about>. (cit. 2019-17-07).
- [13] Trainright. *Become A Stronger, Fitter Athlete In Less Time With Endurance Coaching*. URL: <https://trainright.com/>. (cit. 2019-17-07).

- [14] G. Lim. *Learning TypeScript*. Packt Publishing Ltd., 2015. ISBN: 9781783985548.
- [15] Antdesign. *Ant Design of React Documentation*. URL: <https://ant.design/docs/react/introduce>. (cit. 2019-25-07).
- [16] JWT. *Introduction to JSON Web Tokens*. URL: <https://jwt.io/introduction/>. (cit. 2019-17-07).