

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Webová aplikace pro organizaci automobilových závodů

Lukáš Přáda

Vedoucí: Ing. Božena Mannová, PhD.
Obor: Softwarové inženýrství a technologie
Květen 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Přáda** Jméno: **Lukáš** Osobní číslo: **475690**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Webová aplikace pro organizaci automobilových závodů

Název bakalářské práce anglicky:

Web application for organizing car races

Pokyny pro vypracování:

Seznamte se s problematikou organizace a evidence automobilových závodů a dostupnými aplikacemi pro organizaci takových závodů

Na základě provedené analýzy specifikujte základní požadavky na funkcionality systému.

Navrhněte architekturu systému.

Seznamte se s technologiemi potřebnými pro vytvoření aplikace. Vyberte a Porovnejte tyto technologie a vyberte ty, které mohou být pro aplikaci použity. Svá rozhodnutí zdůvodněte.

Vhodně zvolenými nástroji aplikaci implementujte.

Otestujte aplikaci včetně uživatelských testů a výsledky vyhodnoťte.

Při zpracování využívejte prostředky Softwarového inženýrství.

Seznam doporučené literatury:

1. Roger S. Pressmann Bruce Maxim: Software Engineering: A Practitioner's Approach , ISBN-10: 9780078022128
2. <https://grandprix-software-central.com/index.php/20-software/gprm/1-gprm-info>
3. What's the Difference Between SQL Server and MySQL?[online], <https://www.codefirst.co.uk/blog/difference-sqlserver-mysql/>.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Božena Mannová, Ph.D. kabinet výuky informatiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **11.02.2022**

Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Božena Mannová, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Chtěl bych poděkovat hlavně vedoucí této bakalářské práce paní Ing. Boženě Mannové, PhD. za konzultace, průběžnou kontrolu a rady, díky kterým byla práce usnadněna. Rád bych také poděkoval všem účastníkům testování za nalezené problémy a poznámky k aplikaci.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci zpracoval samostatně za pomoci odborných článků a literatury uvedené v dokumentu.

V Praze, dne 20.5.2022

Abstrakt

Tento projekt se zabývá analýzou, návrhem a implementací webové aplikace, která umožní organizaci automobilových závodů. Kromě základního popisu aplikace a příkladového procesu, obsahuje analýzu existujících řešení pro lepší pochopení problému. Další část je zaměřena na specifikaci aplikace. Tato kapitola obsahuje výpis všech požadavků, use-case diagram a návrh uživatelského rozhraní. Práce poté porovnává technologie, které mohou být použity pro implementaci a zdůvodňuje ty, které jsou vybrány. Dále je zde popsána samotná implementace, její postup a další její aspekty. Ukázka aplikace zobrazuje jak aplikace vypadá a funguje. Poslední dvě kapitoly jsou testování a závěr.

Klíčová slova: webová aplikace, java, javascript, organizace, motorsport, automobil, závod, spring, react

Vedoucí: Ing. Božena Mannová, PhD.

Abstract

This project is focused on the analysis, design and implementation of a web application for organizing car racing. Besides the basic overview of the application and exemplary process, this project contains an analysis of the existing solutions for a better understanding of the problem and definition of the application requirements. The next chapter specifies the application. This chapter list the necessary requirements, use-case diagram and wireframes. Further, it compares possible implementation solutions and defines reasons for choosing. Next is the implementation, its process and other aspects. The demonstration of the application shows the design and function. The last two chapters are testing and conclusion.

Keywords: web application, java, javascript, organization, motorsport, racecar, car, race, spring, react

Title translation: Web application for organizing car races

Obsah

Zadání práce	iii	3.4 Use case diagram	7
1 Úvod	1	3.5 Návrh uživatelského rozhraní	8
1.1 Cíl práce	1	4 Porovnání jednotlivých technologií pro implementaci	11
1.2 Popis aplikace	1	4.1 Front-end	11
1.3 Příklad procesu	2	4.1.1 React	11
2 Analýza existujících řešení	3	4.1.2 Angular	12
2.1 Existující řešení	3	4.1.3 Rozdíly mezi React a Angular	12
2.1.1 RaceDay	3	4.2 Back-end	13
2.1.2 Race Manager	4	4.2.1 Spring	13
2.1.3 ORGSU	4	4.2.2 Django	13
2.1.4 RaceRoster	4	4.3 Databáze	13
2.2 Shrnutí existujících řešení	4	4.3.1 PostgreSQL	14
3 Návrh a specifikace aplikace	5	4.3.2 MySQL	14
3.1 Funkční požadavky	5	4.3.3 Microsoft SQL	14
3.2 Nefunkční požadavky	6	4.3.4 Oracle	14
3.3 Uživatelské role	6	4.4 Shrnutí	15

5 Návrh implementace	17		
5.1 Vybrané technologie	17		
5.2 Databázový systém	17		
5.3 Architektura	18		
6 Implementace řešení	19		
6.1 Postup při implementaci	19		
6.2 Vývojové prostředí	20		
6.3 Architektura	20		
6.3.1 Sekvenční diagramy	20		
6.4 Databázový model	22		
6.5 Java	22		
6.6 Spring	23		
6.7 Spring Data JPA	23		
6.8 Spring Security	24		
6.9 Apache Maven	24		
6.10 Připojení k databázi	24		
6.11 React	25		
6.12 Struktura aplikace	25		
		6.12.1 Back-end	26
		6.12.2 Front-end	27
		7 Ukázka aplikace	29
		7.1 Přihlášení a Registrace	29
		7.2 Zobrazení uživatelského profilu	30
		7.3 Vytvoření závodu	31
		7.4 Výpis nadcházejících závodů	32
		7.5 Registrace a odhlášení ze závodu	33
		7.6 Moje závody	34
		7.7 Garáž	34
		7.8 Závodní týmy	35
		7.9 Export a import výsledků	37
		8 Testování	39
		8.1 Scénář pro uživatelské testy	40
		8.2 Vyhodnocení dotazníku k uživatelským testům	40
		9 Závěr	43
		9.1 Výstup práce	43

9.2 Shrnutí	43
A Struktura přiložených souborů	45
B Literatura	47

Obrázky

3.1 Případy užití	7	7.8 Odeslaní žádosti	36
3.2 Návrh stránky pro nadcházející závod	8	7.9 Zpracování žádosti	36
3.3 Návrh stránky pro zobrazení uživatelského profilu	9	A.1 Stromová struktura přiložených souborů	46
3.4 Návrh stránky pro zobrazení výsledků	9		
6.1 Sekvenční diagram vytvoření závodu	21		
6.2 Sekvenční diagram registrace do závodu	21		
6.3 Databázový model	22		
7.1 Přihlašovací obrazovka	30		
7.2 Uživatelský profil	31		
7.3 Domovská stránka	32		
7.4 Nadcházející závody	33		
7.5 Registrace a odhlášení ze závodu	33		
7.6 Závody uživatele	34		
7.7 Vozidla uživatele	35		

Kapitola 1

Úvod

1.1 Cíl práce

Cílem této práce je analyzovat, navrhnout a implementovat webovou aplikaci, která umožní organizaci automobilových závodů. Analýza zahrnuje pohled na existující řešení a jejich shrnutí. Návrh aplikace staví na analýze existujících řešení a dostupných technologií a specifikuje podobu aplikace. Výstupem samotné implementace je funkční otestovaná aplikace, která splňuje požadavky definované v návrhu.

1.2 Popis aplikace

Předmětem práce je webová aplikace, implementována pomocí jazyka Java, pro organizaci a evidenci automobilových závodů. Cílem je zpřístupnit vytváření a organizaci závodů pro organizátory a přihlášení do závodů pro jezdce. Aplikace bude nabízet funkce jako výpis výsledků závodů nebo závodních vozů. Dále například vytváření nového týmu, import a export výsledků. To vše přes intuitivní a přehledné webové rozhraní. Data budou ukládána do databáze běžící na PostgreSQL.

■ 1.3 Příklad procesu

Organizátor se může přihlásit, vytvořit nový závod. K závodu může přiřadit trať, datum, počet závodníků a kategorie vozů. Závodník se po přihlášení může zaregistrovat do závodu se správným vozem. Po závodě si organizátor exportuje seznam jezdců a vyplní pořadí na začátku a na konci závodu a zpět importuje do systému. Závodník se může podívat na výsledky a zkontrolovat správnost.

Kapitola 2

Analýza existujících řešení

2.1 Existující řešení

Na trhu existuje několik aplikací pro organizaci sportovních akcí a závodů. Dva z takových programů, které se starají přímo o organizaci automobilových závodů a nejvíce se podobají problematice této práce jsou RaceDay a Race Manger. Dále je na trhu ale spousta softwarů a aplikací, které se zaměřují na závody v běhu, cyklistice a dalších sportovních aktivitách. Společně tyto programy dodávají dobrý přehled o tom co je pro organizaci závodů důležité a co by měla webová aplikace pro organizaci závodů obsahovat a které funkce by neměly chybět.

2.1.1 RaceDay

Program od společnosti Digital Opera. Tento software se nejvíce podobá aplikaci, která je předmětem této práce. Umožňuje správu závodů, včetně tříd závodních aut, bodových systémů nebo rozdělení závodníků do skupin. Stará se o registraci závodníků do závodu, výsledky závodů i platby za vstupné a členství. [12]

■ 2.1.2 Race Manager

Software od firmy Track Ready Products pro organizaci automobilových, motokárových, motocyklových a jiných traťových závodů. Tento systém byl vydán v roce 1995, proto je už relativně zastaralý, hlavně co se týče uživatelského rozhraní. Funkcionality ale nepostrádá. Tento software umí založit závod nebo jinou akci, které mimo jména přidá i poplatky pro účastníky, bodovací systém od prvního do desátého místa nebo třídu závodních aut. Nechybí ani bohatá evidence automobilů a závodníků. [11]

■ 2.1.3 ORGSU

Webová aplikace od české společnosti Organizers support s.r.o. pro organizaci a evidenci závodů, závodníků s podporou pro marketing. Je to aplikace zaměřená na běh. Mezi její funkce patří databáze atletů, správa akcí, email marketing, profil atleta, měření výsledků, sledování poloh, e-shop, online registrace a další. [13]

■ 2.1.4 RaceRoster

Aplikace pro organizátory závodů v běhu, cyklistice, plavání nebo vlastního typu, vytvořená skupinou nadšenců sportu z Velké Británie. Podporuje registrace, platby, funkce pro časování závodníků, výsledky a další. Organizátor vytvoří závod zadáním atributů jako je typ, jméno, datum, místo konání a poplatek za účast v závodě. [14]

■ 2.2 Shrnutí existujících řešení

Zmíněné aplikace mají mnoho společného ale zároveň mají nějaké funkce navíc oproti ostatním nebo jim chybí. Aplikace RaceDay je nelepší ukázkou toho jak by měla aplikace vypadat. Pokrývá celou vizi tohoto projektu a mnoho věcí navíc. Software RaceRoster nabízí moderní a čistý design, ze kterého je dobré vzít si příklad při vytváření uživatelského rozhraní.

Kapitola 3

Návrh a specifikace aplikace

V této kapitole se zaměřuji na specifikaci aplikace, její funkční a nefunkční požadavky, případy využití a návrh uživatelského rozhraní. Tyto specifikace jsou definovány na základě analýzy existujících řešení a vlastních vědomostí a zkušeností v oblasti motorsportu.

Z existujících řešení lze odvodit několik funkcionalit, které by měla aplikace obsahovat. Mezi ty hlavní patří možnost vytvoření a specifikace závodu, správa výsledků včetně exportu a importu a databáze závodníků. Specifikace závodu bude spočívat v názvu, trati, maximálním počtu závodníků a dalších parametřů. Dále bude aplikace obsahovat možnost přiřadit závodu konkrétní třídu vozů, které budou do závodu vpuštěny aby se zachovala vyrovnanost závodu. U závodníků budou evidovány základní osobní informace pro případný kontakt a identifikaci. Závodníci musí při registraci do závodu vybrat i vozidlo, se kterým budou závodit, proto je potřeba i evidence vozidel.

3.1 Funkční požadavky

Funkční požadavky, které popisují všechny dostupné funkce aplikace. Tyto požadavky vznikly na základě analýzy existujících řešení a vlastních vědomostí v oblasti motorsportu.

1. Přihlášení a registrace nového uživatele

2. Přidání a smazání závodu
3. Registrace a odhlášení ze závodu
4. Vytvoření a smazání závodního týmu
5. Zobrazení výsledků závodů
6. Export a import závodních výsledků
7. Registrace nového vozu závodníka
8. Odeslání žádosti o vstup do závodního týmu
9. Akceptace/Odmítnutí žádosti o vstup do týmu
10. Přidání závodního okruhu

3.2 Nefunkční požadavky

1. Přístup přes webové rozhraní
2. Podpora prohlížeče Google Chrome
3. Možnost uživatelům přiřadit role, od kterých se odvíjí jejich práva a přístup k informacím o závodech.

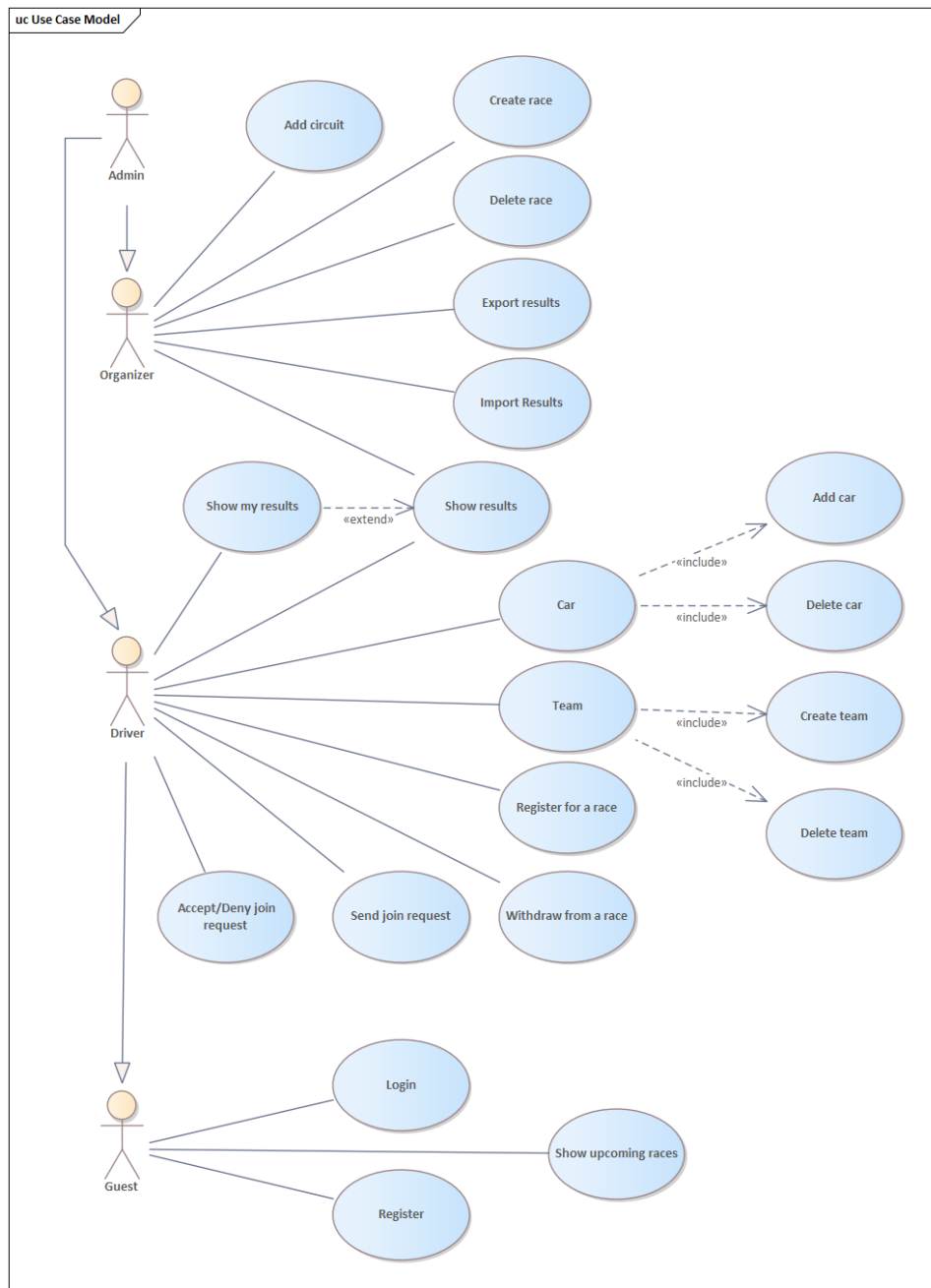
3.3 Uživatelské role

Aplikace rozděluje uživatele na čtyři role. Tyto role slouží k rozdělení uživatelů a jejich přístupu k jednotlivým funkcím systému. Následuje seznam rolí a popis funkcí, ke kterým mají přístup.

1. GUEST - Uživatel s rolí Guest se může přihlásit a zobrazit nadcházející závody, včetně výsledků.
2. DRIVER - Uživatel s rolí DRIVER může vytvořit tým, přidat vůz a zaregistrovat se do závodu.
3. ORGANIZOR - Uživatel s rolí Organizor může vytvářet závody, mazat je a přidávat výsledky. Dále může přidat nový závodní okruh.
4. ADMIN - Uživatel admin má přístup ke všem funkcím aplikace.

3.4 Use case diagram

Use case diagram neboli diagram případů užití zobrazuje aktéry neboli uživatele systému a funkce, ke kterým mají přístup.

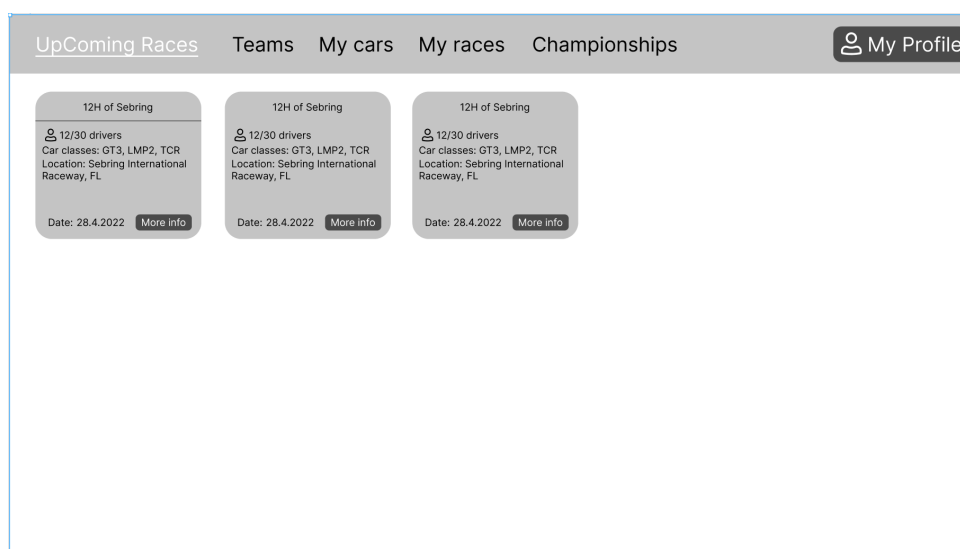


Obrázek 3.1: Případy užití

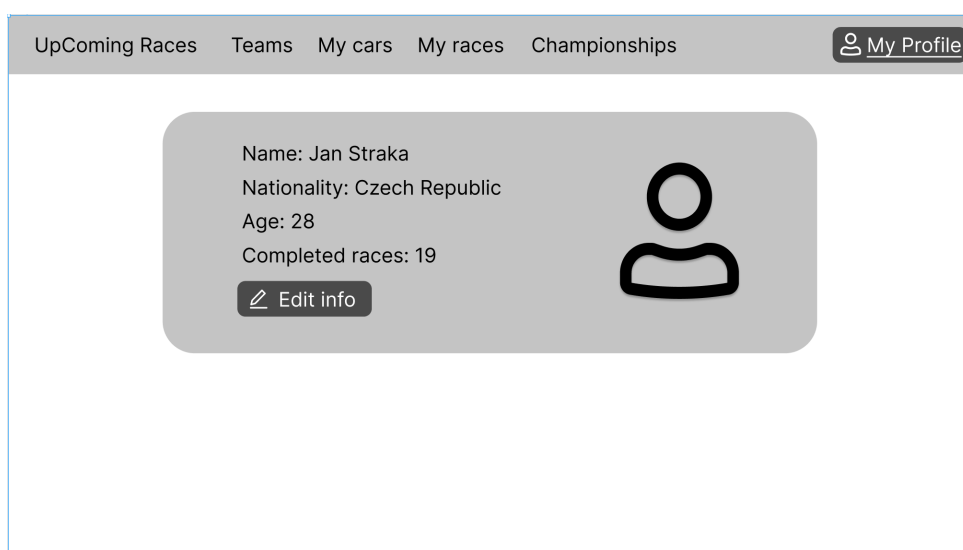
3.5 Návrh uživatelského rozhraní

Před samotnou implementací uživatelského rozhraní jsem si v programu Figma vytvořil několik stránek aplikace abych měl lepší představu o tom, co bude uživatelské rozhraní obsahovat a jak by mělo fungovat. V programu Figma jsem již pracoval v předchozích předmětech studia a to konkrétně v předmětu Informační systémy. Zde jsem se s programem relativně dobře naučil a proto jsem ho využil i k vytvoření wireframů do tohoto projektu.

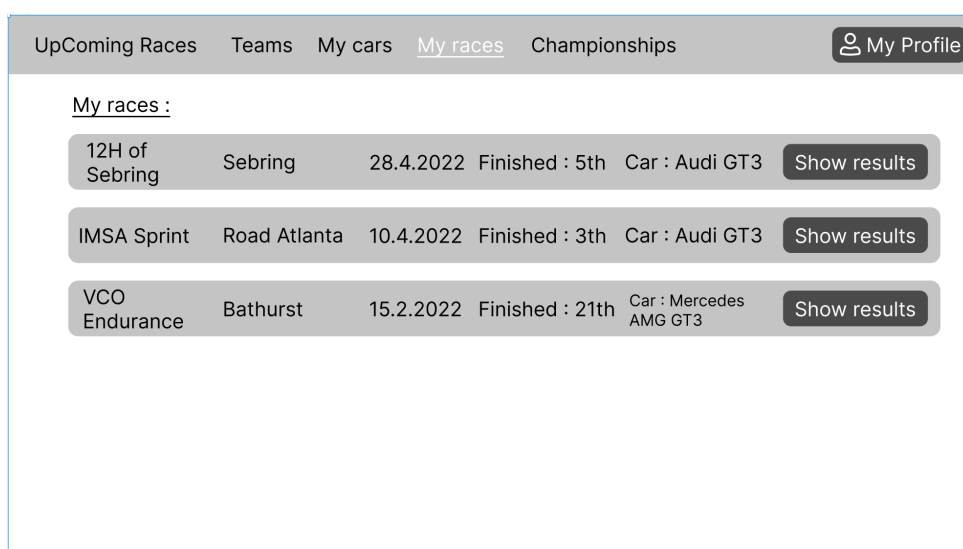
Zde je zobrazeno několik vytvořených návrhů pro uživatelské rozhraní:



Obrázek 3.2: Návrh stránky pro nadcházející závody



Obrázek 3.3: Návrh stránky pro zobrazení uživatelského profilu



Obrázek 3.4: Návrh stránky pro zobrazení výsledků

Kapitola 4

Porovnání jednotlivých technologií pro implementaci

V této kapitole porovnávám dostupné technologie a způsoby pro implementaci jednotlivých částí aplikace. Zaměřuji se spíše na nejpoužívanější a nejrozšířenější z těchto technologií nebo na ty, se kterými jsem se seznámil během studia a proto jsem zvažoval jejich použití při implementaci tohoto projektu.

4.1 Front-end

Dva z nejpoužívanějších javascript nástrojů pro vytváření uživatelského rozhraní jsou knihovna React a framework Angular. [18] Oba spolu nesou výhody i nevýhody oproti druhému. Jejich technologie a rozdíly jsou podrobně popsány v následujících sekcích.

4.1.1 React

Java knihovna od firmy Facebook. Vhodný pro aplikace, které využívají často měnící data. Využívá takzvaného virtuálního DOM. Každý kus uživatelského rozhraní je komponenta s vlastním stavem. Při změně stavu komponenty, React upraví virtuální DOM a porovná s předchozím. Po porovnání upraví

■ 4.2 Back-end

Pro implementaci back-end části aplikace je možné použít jeden z několika frameworků pro vývoj enterprise aplikací. Enterprise aplikace jsou podnikové aplikace, které využívají API a dalších komponent pro zajištění bezpečnosti, škálovatelnosti a stability. [15] Framework je podpůrná sbírka tříd, funkcí a objektů, které usnadňují práci při vývoji aplikace. [19]

■ 4.2.1 Spring

Spring je MVC framework používaný pro vytváření Enterprise aplikací v jazyce Java. Disponuje rozšířením Spring boot, které výrazně zjednodušuje nastavení aplikace. Díky JPA, tabulky databáze mohou být tvořeny jednoduchými java třídami. Závislosti jsou pak implementovány pomocí dependency injection, neboli návrhovým vzorem, který provádí objekty a dovolí tak jednomu objektu používat druhý. JPA je standard pro ORM. ORM(Object-Relational Mapping) je technika pro konverzi dat mezi relační databází a objektově orientovaným jazykem jako je Java. [21]

■ 4.2.2 Django

Django je framework pro tvorbu webových aplikací v jazyce Python. Stejně jako Spring je to open-source framework. Statisticky je to až dvacetkrát používanější framework než Spring. [17] Je používaný na tvorbu sociálních sítí a management systémů. Firmy jako Instagram, Spotify nebo Pinterest používají Django. Je vhodný na rozsáhlé aplikace. Spring je vhodnější na menší aplikace a služby. Popularita Django spočívá v tom, že poskytuje vše potřebné pro vývoj plně funkční webové aplikace již od samotného začátku. Jsou to funkce jako admin interface nebo autentifikační funkce. [16] [17]

■ 4.3 Databáze

Čtyři nejrozšířenější databázové systémy jsou Oracle, MySQL, PostgreSQL a Microsoft SQL server.[5] Všechny tyto systémy jsou systémy pro relační

■ 4.4 Shrnutí

Po analýze možných implementačních řešení byly vybrány technologie a řešení pro každou z částí aplikace. Tyto technologie a důvody, proč byly vybrány, jsou popsány v další kapitole.

Kapitola 5

Návrh implementace

Zde jsou popsány důvody pro vybrané technologie použité k implementaci konkrétních částí aplikace.

5.1 Vybrané technologie

Aplikace bude vytvořena pomocí jazyka Java. Bude použit framework Spring s rozšířením Spring boot, Spring JPA a Spring MVC. Spring byl vybrán, protože s ním mám zkušenosti, získané během studia a část aplikace je v něm již implementována. Podrobnější popis této technologie viz 6.6. Front-end aplikace je řešen knihovnou React. React byl vybrán pro využití virtuálního DOM, který je rychlejší než reálný DOM, používaný frameworkem Angular a protože je lepší pro začínající front-end programátory.

5.2 Databázový systém

Data jsou ukládány do databázového systému PostgreSQL. Daný databázový systém jsem zvolil kvůli zkušenostem získaným během studia a dostupnosti. Také díky aplikaci PgAdmin, která umožňuje snadný přístup a správu databáze. Databázový model je možné vidět na obrázku 6.3.

5.3 Architektura

Zvolená architektura je tří vrstvá architektura. Tato architektura se jeví jako nejlepší možnost pro takovýto druh aplikace. Důvodem implementace této architektury je separace částí aplikace tak, že každá je samostatně funkční. Jelikož byla část back-endu řešena už v předchozím předmětu, je snadné doplnit prezentační vrstvu a spojit tyto části. Zároveň je možné doplnit funkcionality do každé vrstvy bez ohledu na ostatní. Tyto tři části mají následující funkce:

-**Datová vrstva** reprezentuje data, jejich uspořádání a provázanost v aplikaci.
-**Prezentační vrstva** zastupuje tu část aplikace, která popisuje vzhled a interakci uživatele s aplikací. Předává vstup od uživatele aplikační vrstvě a zobrazuje mu data.

-**Aplikační vrstva** se stará o propojení prezentační a datové vrstvy. Řídí komunikaci mezi těmito dvěma vrstvami a přidává logiku samotné aplikace. Všechny tyto části jsou navzájem propojené a komunikují přes jejich společná rozhraní.

Komunikace mezi prezentační a aplikační vrstvou bude probíhat přes REST API.

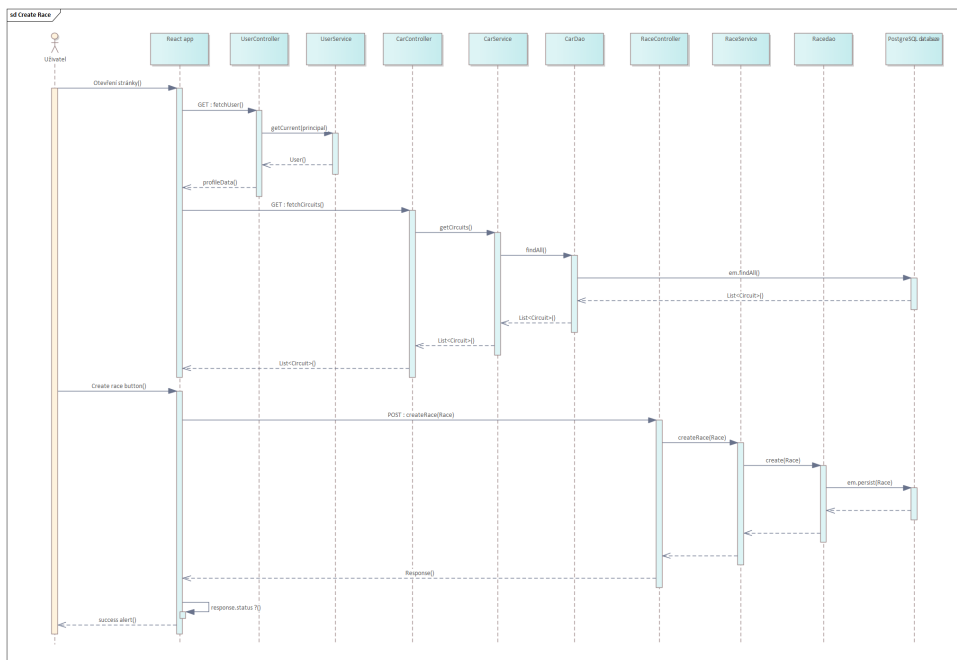
Kapitola 6

Implementace řešení

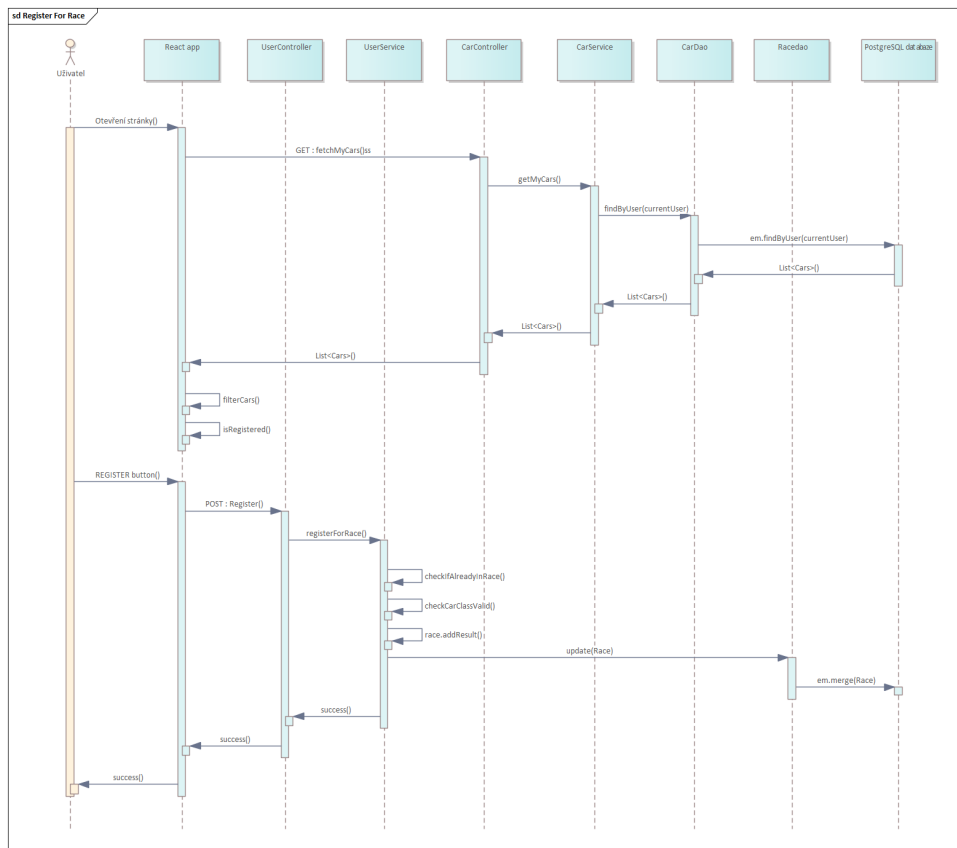
Tato část popisuje samotnou implementaci, která navazuje na návrh z předchozí kapitoly. Je zde obsažen postup při implementaci, použité programy pro implementaci a finální diagramy popisující funkčnost a podobu aplikace.

6.1 Postup při implementaci

Při implementaci jsem začal s back-end částí aplikace a to už v předmětu EAR, kde jsem tuto část vytvořil v rámci semestrální práce. Tato semestrální práce zahrnovala i testování aplikace, pomocí programu postman. Finální verze back-endu není identická té, která byla výsledkem semestrální práce v předmětu EAR. Aplikace byla z většiny pozměněna aby lépe vyhovovala tomuto projektu. Dále jsem vytvořil návrh pro uživatelské rozhraní v programu figma abych měl lepší představu o tom co by mělo uživatelské rozhraní obsahovat a mohl tak začít se samotnou implementací za pomoci knihovny React. V Reactu jsem nikdy předtím nepracoval, proto jsem asi týden studoval, jak tato knihovna funguje. Poté jsem tedy implementoval front-end část. Celý tento proces provázelo průběžné testování a oprava chyb, na které jsem při něm narazil. I finální testování, konkrétně uživatelské testy, objevilo několik nedostatků, které byly vyřešeny.



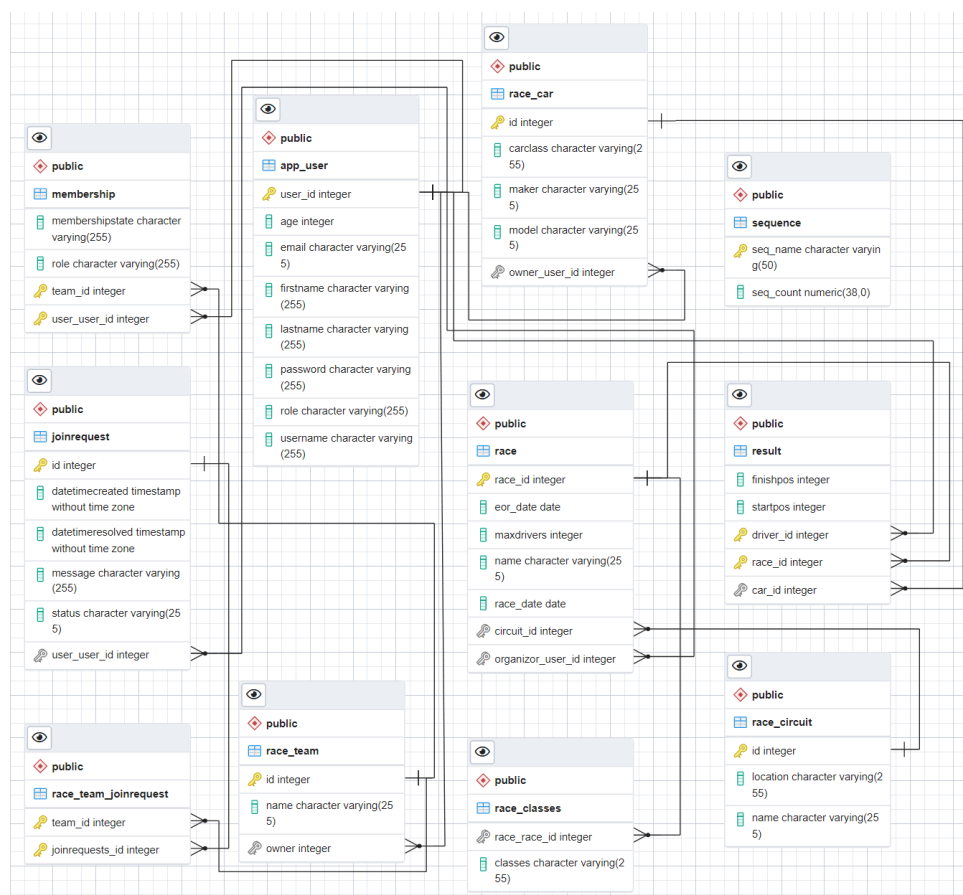
Obrázek 6.1: Sekvenční diagram vytvoření závodu



Obrázek 6.2: Sekvenční diagram registrace do závodu

6.4 Databázový model

Zde je zobrazen finální databázový model, který je výsledkem implementace ve spring aplikaci. Ve Spring aplikaci byly definovány hlavní entity a jejich relace. Následně se vytvořila databáze i s pomocnými vazebnými tabulkami, které jsou na obrázku 6.3.



Obrázek 6.3: Databázový model

6.5 Java

V návrhu aplikace byl pro implementaci back-end části aplikace zvolen programovací jazyk Java. S tímto jazykem jsem se seznámil již na střední škole a v porovnání s jinými jazyky s ním mám mnohem více zkušeností. Jazyk Java byl vytvořen v roce 1995 a je používán pro vytváření široké škály aplikací od

mobilních přes webové až po počítačové hry. Je to jazyk, který podporuje spousta platforem. [20]

6.6 Spring

Framework Spring je základní stavební kostkou jádra této aplikace. Velkou výhodou použití tohoto frameworku je zjednodušené nastavení aplikace a využití takzvané Inversion of Control (IoC). IoC, také známo jako Dependency Injection (DI) je proces, při kterém jsou definovány vztahy mezi objekty v aplikaci, pomocí argumentů v konstruktorech objektů, argumentů metod nebo vlastností, které jsou nastaveny objektu po jeho vytvoření. Objekt ve Spring aplikaci nazýváme Bean. Bean jsou objekty tříd, se kterými aplikace pracuje. Všechny námi implementované třídy se při vytvoření objektu stávají Bean, se kterou následně Spring pracuje a dokáže mezi nimi navazovat vztahy potřebné pro správnou funkčnost aplikace. [21]

6.7 Spring Data JPA

Tento modul je součástí větší rodiny Spring Data. Umožňuje snadnou konfiguraci a implementaci JPA repositářů a databázových tabulek. JPA repositáře vytváří rozhraní mezi databází a aplikací pro práci s daty. Umožňují vkládat, číst, mazat a aktualizovat data v databázi. Databázové tabulky jsou definovány anotací `@Table` a `@Entity`. Repositáře jsou označeny anotací `@Repository`. [22]

Listing 6.1: Ukázka využití Spring Data JPA anotací

```

@Entity
@Table(name = "RACE_CAR")
@NamedQueries({
    @NamedQuery(name = "Car.getByUser", query = "SELECT c from
        Car c WHERE c.owner = :user")
})
public class Car extends AbstractEntity{

@Repository
public class CarDao extends BaseDao<Car> {

```

6.8 Spring Security

Spring security je modul umožňující zabezpečit aplikaci pomocí autentizačních i autorizačních pravidel pro lepší bezpečnost a kontrolu nad aplikací. V této aplikaci je konfigurace tohoto modulu v souboru SecurityConfig. Dále jsou použity anotace @PreAuthorize, @PermitAll pro omezení přístupu k API rozhraní aplikace. [21]

Listing 6.2: Ukázka využitá Spring Security anotací

```
@PreAuthorize("hasAnyRole('ROLE_ADMIN', 'ROLE_DRIVER')")
@PostMapping(consumes = MediaType.APPLICATION_JSON_VALUE)
public ResponseEntity<Void> createCar(@RequestBody Car car,
    Principal principal) {
```

6.9 Apache Maven

Projekt Apache Maven je open-source nástroj, který usnadňuje práci se sestavováním java aplikací. Využívá project object modelu (POM) popsaném v souboru pom.xml, kde jsou popsána pravidla a další rozšíření potřebné ke správnému sestavení aplikace. Tento soubor leží v kořenovém adresáři projektu. Řeší závislosti mezi projektem, externími knihovnami a balíčky použitých pro implementaci aplikace. [23]

6.10 Připojení k databázi

Spring aplikace je připojena k PostgreSQL databázi pomocí PostgreSQL driveru, který je přidán do dependencí v souboru pom.xml. Dále je připojení definováno v souboru application.properties.

Listing 6.3: Nastavení připojení k databázi

```
# Database connection setup
spring.datasource.driver-class-name=org.postgresql.Driver
spring.datasource.url=jdbc:postgresql://localhost:5432/postgres
spring.datasource.username=postgres
spring.datasource.password=admin

# Database platform configuration for Eclipselink
spring.jpa.database-platform=org.eclipse.persistence.platform.database.PostgreSQLPlatform
# Additional database-access specifics
spring.jpa.generate-ddl=true
spring.jpa.show-sql=false
```

6.11 React

Pro vytvoření webového uživatelského rozhraní byla použita knihovna React, která staví na jazyce javascript. Každá část rozhraní má svoji komponentu, která ji reprezentuje. Komponenta má stav, ve kterém jsou uloženy data, se kterými pracuje. Pro načítání a posílání dat do aplikace jsou volány http requesty. Ty se vykonávají buďto při vytvoření komponenty nebo po akci vyvolané uživatelem. Data se mohou předávat mezi komponentami. Hlavní navigační komponentou aplikace je Navbar. Navbar obsahuje menu a Router, který se stará o směrování na konkrétní stránky. Podle přihlášeného uživatele se zobrazují dané komponenty v závislosti na tom, ke kterým funkcím má uživatel přístup nebo zda se jedná o ním vytvořený závod, tým a další. To znamená, že uživatel, který vytvořil závod, vidí tlačítko na smazání závodu a podobně.

6.12 Struktura aplikace

Zde jsou popsány struktury všech částí aplikace. Program je rozdělen do dvou hlavních složek. První složka obsahuje front-end část, kterou je ReactJS aplikace. Druhá složka obsahuje back-end část, kterou je java aplikace ve frameworku Spring. Každá část má dále svoji vlastní strukturu složek a souborů.

REST rozhraní. V této vrstvě probíhá jak validace tak případná komunikace s databází pomocí DAO objektů.

■ REST

V této složce jsou obsaženy soubory představující REST rozhraní aplikace. Každý objekt aplikace má svůj controller, který se stará o zpracování požadavků na aplikaci. Pomocí anotací je zde řešena podoba požadavku i autorizace pro každý typ požadavku. Požadavky jsou směrovány na service část aplikace, kde jsou data validovány a následně uloženy do databáze nebo zahozeny pokud nejsou v souladu s logikou aplikace.

■ Security

Obsahuje soubory pro nastavení a práci s autentizací. Jsou zde metody pro autentizaci při přihlašování a odhlašování uživatele. Dále jsou zde metody k získání právě přihlášeného uživatele a další konstanty.

■ 6.12.2 Front-end

Struktura souborů front-end části aplikace je rozdělena na složky pages, components a styles. Soubory využívají knihovny React a reprezentují komponenty. Tyto komponenty jsou implementovány jako třída nebo funkce.

- Složka pages obsahuje soubory reprezentující stránky, které mají definovanou cestu v Routeru.
- Ve složce components jsou komponenty, které jsou součástí stránek.
- Složka styles obsahuje css soubory, které nastavují vzhled aplikace.



Kapitola 7

Ukázka aplikace

V této kapitole jsou zobrazeny ukázky pro všechny stránky v aplikaci. Ke každé ukázce je přidán stručný popis toho jak je možné s uživatelským rozhraním pracovat a jak aplikaci používat. Při tvorbě uživatelského rozhraní byl dbán důraz na čistý a jednoduchý design, který umožňuje rychlou a snadnou orientaci při používání aplikace.



7.1 Přihlášení a Registrace

Při prvním vstupu do aplikace se uživatel musí registrovat. To provede na stránce Login/SignUp. Zde zadá potřebné údaje a pokud splňují pravidla validace, uživatel je informován o vytvoření profilu. Následně se uživatel může přihlásit a začít tak používat funkce aplikace, které jeho role umožňuje.

Race Organization

[Home](#) [Upcoming Races](#) [Teams](#) [Login/SignUp](#)

Login

Username :

Password :

Register

First name :

Last name :

Email :

Age :

Username :

Password :

Role :

Obrázek 7.1: Přihlašovací obrazovka

7.2 Zobrazení uživatelského profilu

Po přihlášení se může uživatel podívat na údaje o svém profilu na stránce My Profile. Pokud potřebuje, může je i upravit kliknutím na tlačítko EDIT. Po upravení údajů se stiskem tlačítka SAVE údaje uloží.

Race Organization

[Home](#)[Upcoming Races](#)[My Races](#)[Teams](#)[Garage](#)[My Profile](#)

My Profile

First name : Last name : Email : Username : Age : [EDIT](#)[LOGOUT](#)

Obrázek 7.2: Uživatelský profil

7.3 Vytvoření závodu

Pokud je uživatel organizátor nebo admin, může vytvořit nový závod vyplněním tohoto formuláře na domovské stránce. Pokud jsou zadané údaje o závodě v souladu s validací, uživatel je informován a závod vytvořen. Závod se následně zobrazí v seznamu nadcházejících závodů.

Race Organization

[Home](#) [Upcoming Races](#) [My Races](#) [Teams](#) [Garage](#) [My Profile](#)

Create a new race

Name :

Circuit :

Date :

End of registrations :

Max drivers :

Select classes : GT3 GT4 LMP2 LMP1

[Create race](#)

Add a new circuit

Name :

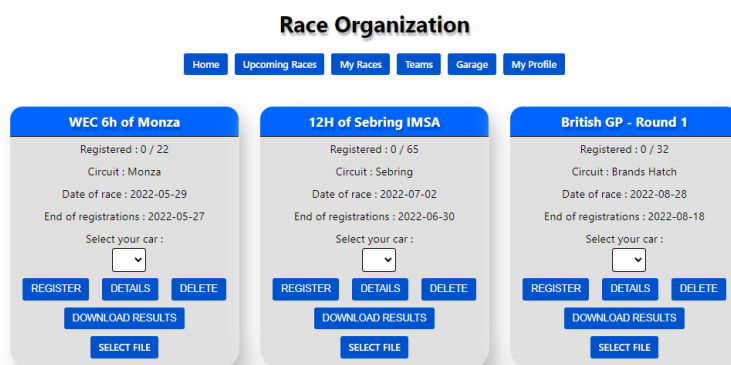
Location :

[Add circuit](#)

Obrázek 7.3: Domovská stránka

7.4 Výpis nadcházejících závodů

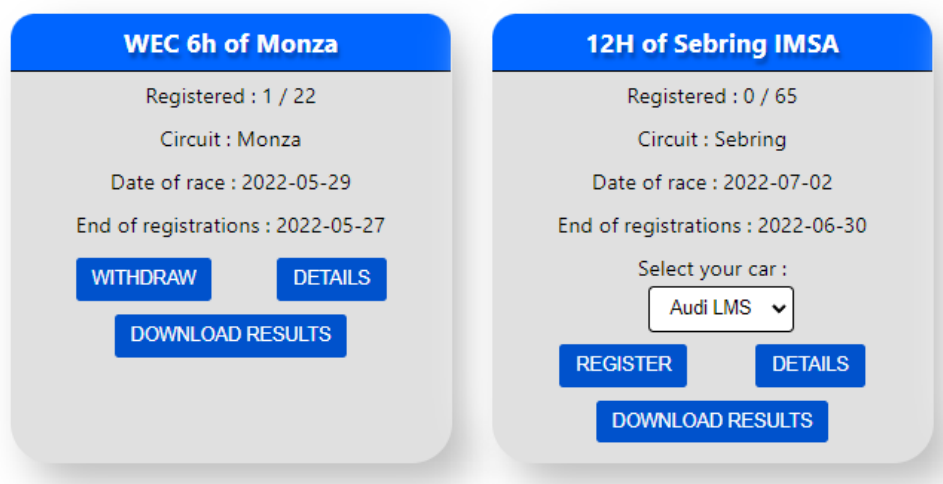
Stránka Upcoming Races zobrazuje všechny nadcházející závody společně s jejich detaily. Pokud je přihlášen uživatel, který nějaký ze závodů vytvořil a je tedy jeho organizátor, může daný závod smazat tlačítkem DELETE. Tlačítko DETAILS zobrazí více informací o závodě.



Obrázek 7.4: Nadcházející závody

7.5 Registrace a odhlášení ze závodu

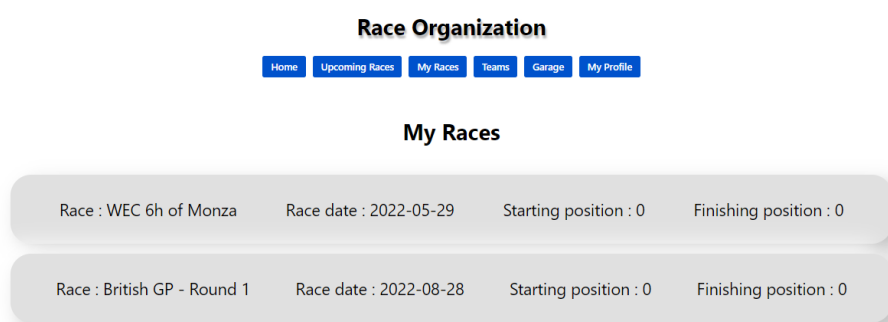
Uživateli s rolí driver je dovoleno se registrovat do konkrétního závodu. To provede vybráním vozu a kliknutím na tlačítko REGISTER. Pokud je uživatel již registrován, u tohoto závodu je zobrazeno tlačítko WITHDRAW, kterým zruší jeho registraci v závodě.



Obrázek 7.5: Registrace a odhlášení ze závodu

7.6 Moje závody

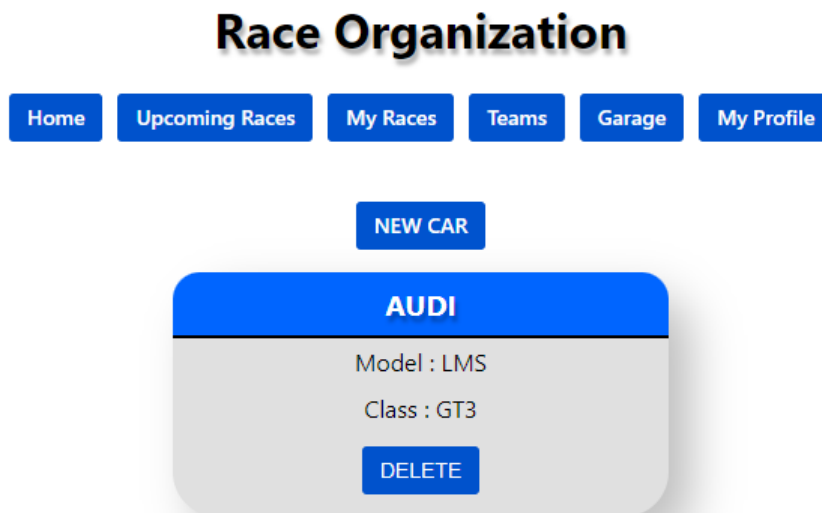
Na stránce My Races jsou zobrazeny výsledky závodů, ve kterých je uživatel zaregistrován. Je zde vidět závod, jeho datum, vůz, se kterým se závodů účastnil, pozice na startu a na konci závodu.



Obrázek 7.6: Závody uživatele

7.7 Garáž

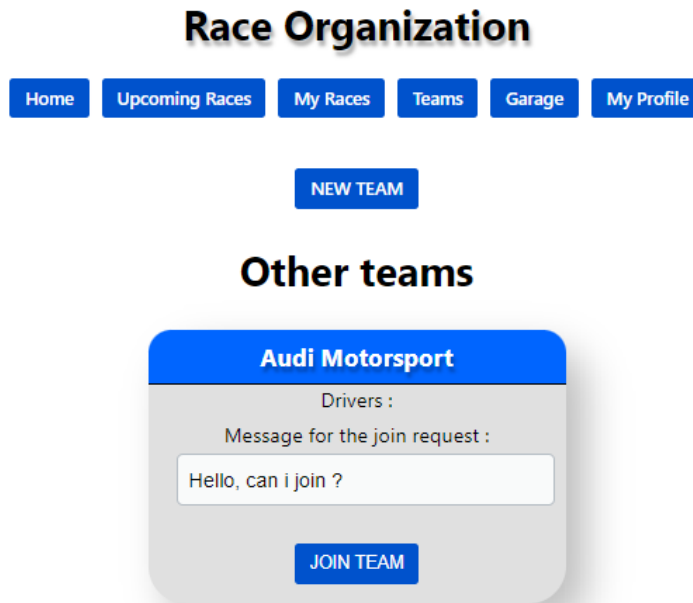
Stránka Garage umožňuje zobrazení aut, které má uživatel k dispozici a zároveň přidání nového závodního auta. To uživatel přidá tlačítkem NEW CAR a po zadání údajů se automobil zobrazí v garáži.



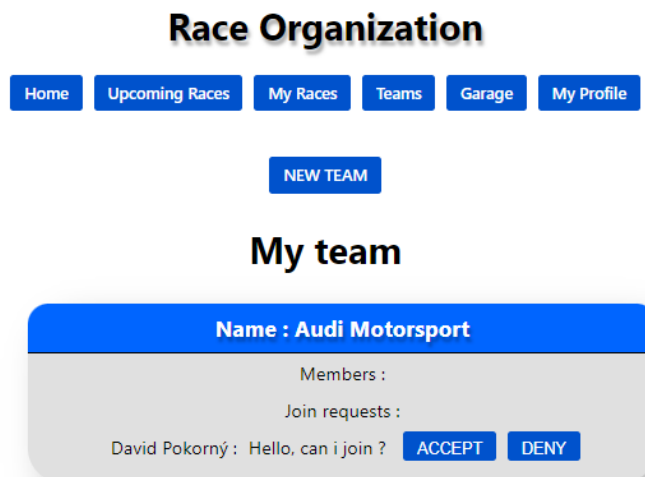
Obrázek 7.7: Vozidla uživatele

7.8 Závodní týmy

Na této stránce může uživatel s rolí admin nebo driver vytvořit nový závodní tým, nebo se přidat do jednoho ze zobrazených týmů. Zároveň zde tvůrce týmu vidí svůj tým a všechny žádosti a přidání do týmu od ostatních uživatelů. Tyto žádosti může přijmout nebo odmítnout.



Obrázek 7.8: Odeslaní žádosti



Obrázek 7.9: Zpracování žádosti

■ 7.9 Export a import výsledků

Organizátor si u každého svého závodu může stáhnout soubor s výsledky, který upraví podle konečného pořadí v závodě a následně importuje. Soubor stáhne kliknutím na tlačítko DOWNLOAD RESULTS a importuje vybráním souboru po kliknutí na SELECT FILE. Po stažení souboru ho organizátor otevře a upraví pozice na startu a na konci závodu. Tento soubor uloží a může ho importovat zpět do systému.



Kapitola 8

Testování

Testování aplikace je důležitou součástí při procesu implementace softwarového projektu. Bez testování nejsou odhaleny potencionální chyby, které vznikly při implementaci a při nasazení se tak aplikace může stát částečně nebo i zcela nefunkční.

Aplikace byla průběžně s každou přidanou funkcionalitou testována pomocí programu Postman, který umožňuje odesílat požadavky na REST api rozhraní aplikace bez nutnosti uživatelského rozhraní. Tato forma testování odhalila nejvíce chyb, které při implementaci vznikly. Všechny nalezené chyby byly opraveny a v aplikaci již nefigurují.

Dále byly napsány JUnit testy, které testují logiku aplikace. Tyto testy kontrolují zda metody v servisní vrstvě pracují správně a porovnává objekty, se kterými tyto metody pracují, zda bylo vše provedeno v souladu s logikou aplikace.

Pro uživatelské testy byl vytvořen formulář, který byl následně spolu s odkazem na nasazenou aplikaci odeslán několika osobám. Tito uživatelé následně vyplnili dotazník podle průběhu jejich práce s aplikací.

8.1 Scénář pro uživatelské testy

1. Zaregistrujte nového uživatele s rolí organizátor.
2. Přihlašte se jako organizátor.
3. Přidejte nový závodní okruh.
4. Vytvořte 2 závody. Závody jsou vidět na stránce Nadcházejících závodů.
5. Smažte jeden závod. Závod již není vidět.
6. Zaregistrujte nového uživatele s rolí závodník. Přihlašte se jako závodník a přidejte vůz. Uživatelské údaje si zapište.
7. Zaregistrujte se do závodu se správným vozem. Vůz musí mít třídu, která figuruje v závodě.
8. Zobrazte svoje závody. Měli byste vidět závod, do kterého jste se registrovali.
9. Vytvořte tým.
10. Odhlašte se.
11. Zaregistrujte a přihlašte se jako jiný závodník a pošlete žádost do týmu, který jste vytvořili.
12. Přihlašte se zpět na tvůrce týmu a přijměte žádost. V týmu je vidět nový člen.
13. Změňte si údaje o svém profilu a zkontrolujte, že se změnili.

Další otázky v dotazníku :

- Jak dobře jste se orientovali v aplikaci ?
- Máte nějaké další poznámky nebo připomínky?

8.2 Vyhodnocení dotazníku k uživatelským testům

Z uživatelských testů byly shromážděny následující odpovědi, které popisují problémy při používání aplikace:

- Přihlášení bych rád potvrzoval tlačítkem Enter, to nelze.
- Sekce Other teams má zavádějící název. Nachazejí se zde i moje týmy.
- Výber auta by mal dovoliť vyberať len z dovolených kategórií, nie všetkých áut.
- Logout tlačítko skryté.
- Pomohlo by ďalšie rozdelenie kategórií, pre lepšiu orientáciu.
- Změnil jsem věk, po překliknutí na jinou stránku je věk stále stejný jako před změnou.

Většina z uživatelů neměla s používáním aplikace větší problémy. Funkční problémy, které uživatelé zmínili, byly opraveny. Dále se našly menší nedostatky, které nemají vliv na funkčnost aplikace, ale bylo by dobré je adresovat a opravit, pokud by se aplikace začala využívat pro reálné účely.



Kapitola 9

Závěr



9.1 Výstup práce

Výstupem práce je prototyp aplikace se všemi základními funkcemi pro organizaci automobilových závodů, definovanými v části o specifikaci aplikace. Aplikace byla otestována a nasazena na cloudovou platformu Heroku. Aplikaci je možné vyzkoušet na adrese <https://race-org-app.herokuapp.com/>.



9.2 Shrnutí

Při práci na tomto projektu jsem se naučil a seznámil se několika novými technologiemi jako React nebo Heroku.

Při analýze existujících řešení jsem měl problémy najít správné typy programů a aplikací, které se zaměřují na tuto problematiku. Tato oblast není velice rozšířená, ale nakonec se mi povedlo najít řešení, která dobře pokrývají funkce, pro organizaci automobilových nebo jiných závodů.

Práce na projektu nebyla bez problémů. Jak už to u softwarových projektů bývá, provázelo mě spousta problémů, ke kterým bylo potřeba delšího vyhledávání po všech různých diskuzních fórech a dalších stránkách. Nakonec jsem

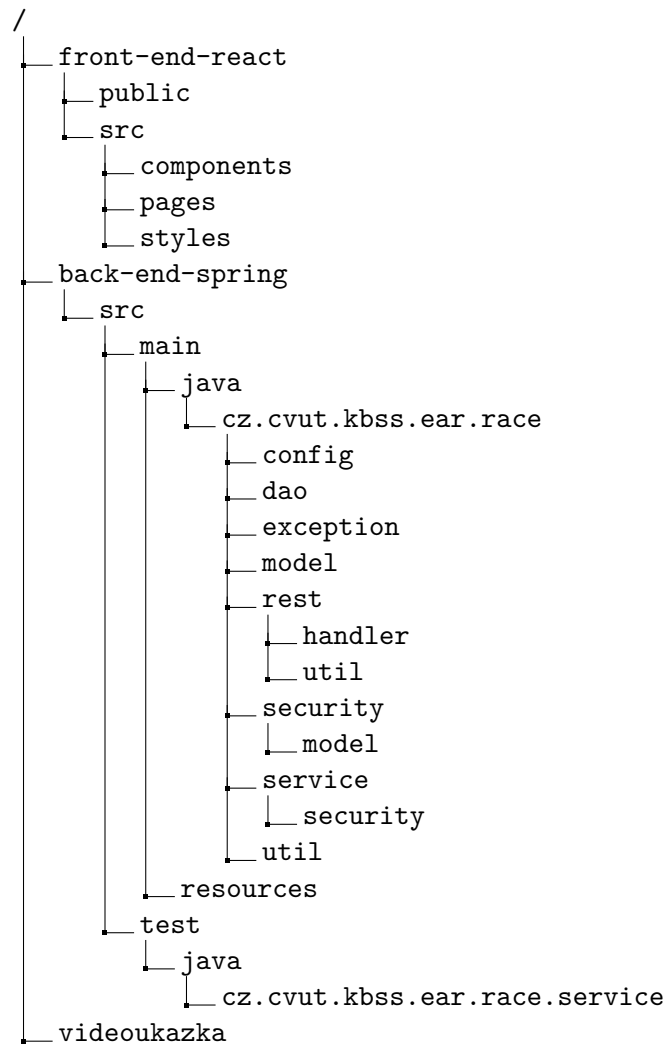
ale s aplikací a celkovou prací na projektu spokojen.

Aplikace je plně připravena na další rozšíření díky zvolené architektuře. Mezi možná rozšíření bych zařadil správu šampionátu jakožto kolekci několika závodů nebo bodový systém.



Příloha A

Struktura přiložených souborů



Obrázek A.1: Stromová struktura přiložených souborů

Příloha B

Literatura

- [1] Mark Smallcombe *PostgreSQL vs MySQL: The Critical Differences*[online] [cit. 2020-26-5] , <https://www.xplenty.com/blog/postgresql-vs-mysql-which-one-is-better-for-your-use-case/>.
- [2] *What's the Difference Between SQL Server and MySQL?*[online] [cit. 2018-23-9] , <https://www.codefirst.co.uk/blog/difference-sql-server-mysql/>.
- [3] *The Advantages and Disadvantages of Microsoft SQL Server*[online] , <https://www.rothrobot.com/the-advantages-and-disadvantages-of-microsoft-sql-server/>.
- [4] Michael Boroznets *Why use PostgreSQL as a Database for my Next Project in 2021*[online] [cit. 2021-21-1] , <https://fulcrum.rocks/blog/why-use-postgresql-database/>.
- [5] *Ranking of the most popular database management systems worldwide, as of January 2022*[online], <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems/>.
- [6] Hemendra Singh *Oracle Database Advantages, Disadvantages and Features*[online] [cit. 2022-5-5], <https://theninehertz.com/blog/advantages-of-using-oracle-database/>.
- [7] Ilana Brudo *Angular vs React: 5 Key Differences*[online] [cit. 2020-5-7] , <https://www.tabnine.com/blog/angular-vs-react/>.

