

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Webová aplikace pro pořádání závodů v orientačním běhu**

Místo této strany bude  
zadání práce.

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 26. června 2017

Jan Palcút

# Poděkování

Tímto bych chtěl poděkovat vedoucímu bakalářské práce panu Ing. Tomášovi Hercigovi za cenné rady, připomínky a odborné vedení této práce.

## **Abstract**

The topic of this bachelor thesis is the creation of a web application for organizing orienteering races including user registration, creation of races, registration of users to races, management of teams and contestants, and results evaluation based on selected criteria. Section 2 deals with the description of Java web frameworks. Section 3 describes the Spring modules and the reasons why this framework was chosen for implementation. Section 4 includes used technologies. The Section 5 describes the functionalities of the website and database. Section 6 describes the creation of the application in the framework Spring, the structure of the project and selected sections of the code. Section 7 includes race simulation, stress test, web browsers compatibility, and Selenium tests.

## **Abstrakt**

Předmětem mé bakalářské práce je vytvoření webové aplikace pro pořádání závodů v orientačním běhu umožňující registraci uživatelů, vytváření závodů, registraci uživatelů do závodu, spravování seznamu týmů včetně účastníků a následné vyhodnocení zadaných výsledků podle zvolených kritérií. Kapitola 2 se zabývá popisem webových frameworků jazyka Java. Kapitola 3 popisuje moduly Springu a důvody proč byl tento framework vybrán pro implementaci. Kapitola 4 obsahuje použité technologie. Kapitola 5 popisuje funkcionality webu a databázi. Kapitola 6 popisuje tvorbu aplikace ve frameworku Spring, strukturu projektu a vybrané části kódu. Kapitola 7 obsahuje simulaci závodu, zátěžový test aplikace, zkoušku kompatibility webových prohlížečů a Selenium testy.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Rešerše frameworků</b>	<b>2</b>
2.1	Apache Struts 2 . . . . .	2
2.2	JavaServer Faces . . . . .	2
2.3	Apache Wicket . . . . .	3
2.4	Google Web Toolkit . . . . .	3
2.5	Apache Tapestry . . . . .	4
2.6	OpenXava . . . . .	4
2.7	RIFE . . . . .	5
2.8	Seam . . . . .	5
2.9	Srovnání frameworků . . . . .	6
<b>3</b>	<b>Framework Spring</b>	<b>7</b>
3.1	Historie . . . . .	7
3.2	Moduly . . . . .	7
<b>4</b>	<b>Použité technologie</b>	<b>10</b>
4.1	HyperText Markup Language . . . . .	10
4.2	Cascading Style Sheets . . . . .	11
4.3	Bootstrap . . . . .	11
4.4	Javascript a jQuery . . . . .	12
4.5	Hibernate . . . . .	12
4.6	DataTables . . . . .	12
<b>5</b>	<b>Návrh aplikace</b>	<b>14</b>
5.1	Požadavky na aplikaci . . . . .	14
5.2	Třídy uživatelů . . . . .	15
5.3	Funkcionality webu . . . . .	16
5.4	Návrh databáze . . . . .	21
<b>6</b>	<b>Implementace</b>	<b>28</b>
6.1	Struktura projektu . . . . .	28
6.1.1	Komponenty . . . . .	28
6.1.2	Obsah složky webapp . . . . .	33
6.2	Ukázky implementace . . . . .	35

<b>7 Testování</b>	<b>37</b>
7.1 Simulace závodu . . . . .	37
7.2 Zátěžový test . . . . .	39
7.3 Kompatibilita prohlížečů . . . . .	40
7.4 Selenium testy . . . . .	41
<b>8 Závěr</b>	<b>43</b>
<b>Literatura</b>	<b>44</b>
<b>Přílohy</b>	<b>47</b>
A Databáze . . . . .	48
B Dokumentace . . . . .	49
C Obsah DVD . . . . .	65

# 1 Úvod

Cílem mé bakalářské práce je vytvoření webové aplikace pro pořádání orientačních běhů. Tato webová aplikace umožní spravovat závody, u kterých půjde při vytváření definovat vlastní kategorie a počet členů v týmu. Součástí závodu bude registrace přes formuláře, které umožní uživateli přidat do závodu jeho tým nebo registrovat sám sebe jako uživatele bez týmu. Z registrovaných týmů bude vytvořen seznam, který bude sloužit jako prezentace účastníků závodů. Správci závodu bude umožněno přidávat, mazat týmy a editovat jednotlivé údaje o týmech i jejich členech. Součástí seznamu registrovaných týmů bude umožněno správci závodu importovat a exportovat týmy ve formátu XLSX.

Dále bude součástí webové aplikace správa a prezentace výsledků. Správa bude umožňovat zadávání startovního času, času dokončení, získaných bodů a určení délky závodu pro výpočet penalizace. Správa i vizualizace výsledků bude obsahovat filtry, řazení hodnot v tabulce a vyhledávání nejen pro efektivnější zadávání výsledků, ale i například umožnění zobrazit výsledky pro příslušnou kategorii. U zobrazených výsledků bude pro správce závodu umožněno vyfiltrované údaje exportovat ve formátu XLSX a CSV.

Prvním bodem zadání mé bakalářské práce je provést rešerši nejpoužívanějších frameworků pro tvorbu webových aplikací v jazyce Java, ve které bude o každém z nich napsáno pár vět. Součástí rešerše bude i následný výběr frameworku, který bude použit pro implementaci návrhu řešení.

Druhým bodem mého řešení je návrh webové aplikace, který bude vycházet ze zadaných požadavků na aplikaci. Součástí návrhu bude popis výběru jednotlivých technologií pro implementaci, popis funkcionalit webu a návrh řešení ukládání dat pro potřeby webu.

Po návrhu následuje implementace. V této části dojde k vytvoření webové aplikace ve frameworku Spring. Na tento bod navazuje testování, ve kterém bude provedena simulace imaginárního závodu. Cílem tohoto testu bude odhalit nedostatky aplikace. Dalším testem bude zátěžový test, který definuje rozsah velikosti závodu, pro který bude tato aplikace použitelná. Součástí testování bude zkouška kompatibility nejpoužívanějších prohlížečů a vytvoření Selenium testů pro ověření správnosti funkcionalit aplikace. Na závěr bude vytvořena uživatelská dokumentace, která bude součástí přílohy.



## 2 Rešerše frameworků

V této kapitole je popsána rešerše osmi frameworků pro tvorbu webových aplikací v jazyce Java. Tato kapitola neobsahuje Spring, protože jemu je věnována celá kapitola 3.

### 2.1 Apache Struts 2

Struts 2 je populární webový framework, který je založen na MVC architektuře. Vznikl kompletním přepsáním původní architektury jeho předchůdce, který nesl označení Struts. K přepsání došlo především proto, že nová verze měla být lépe použitelná a rozšiřitelná pro vývojáře.

Při vzniku nové verze se odstoupilo od původních akčních formulářů. Se Struts 2 se přešlo k používání POJO (Plain Old Java Object) tříd pro vstup z formulářů. Tento framework nám poskytuje vlastní značky, které ušetří psaní kódu pro vývojáře. Součástí těchto značek je i možnost využití technologie Ajax.

Jako další výhoda se udává jednoduchá rozšiřitelnost o další frameworky. Lze snadno integrovat frameworky jako je Spring, Tiles nebo SiteMash. Pro rozšiřitelnost jádra frameworku lze použít i další zásuvné moduly, kterých tento framework nabízí nespočet [1].

### 2.2 JavaServer Faces

JavaServer Faces (JSF) je stejně jako Struts webový framework, který podporuje použití MVC architektury. Cílem tohoto frameworku je zjednodušení konstrukce uživatelského rozhraní.

K usnadnění tvorby uživatelského rozhraní dochází skládáním jednotlivých šablon pomocí komponentů, které jdou jednoduše vložit a vyžadují minimální konfiguraci. Tento framework poskytuje API, které popisuje komponenty uživatelského rozhraní, spravuje jejich stav, zpracovává události a ověřuje správnost dat. Tento framework nabízí řadu základních knihoven, které obsahují komponenty pro použití. V případě nevyhovujících základ-

ních komponentů těchto knihoven nám JSF poskytuje API pro jejich vlastní vývoj a rozšíření.

Stejně jako Struts 2 nám nabízí možnost integrace různých frameworků jako je Spring nebo Hibernate. I tento framework podporuje vlastní značky, které lze použít při práci se šablonami [2].

## 2.3 Apache Wicket

Wicket je komponentově založený framework pro tvorbu webových stránek, který je vyvíjen pod licencí Apache License 2.0 neziskovou organizací Apache Software Foundation.

Framework nám umožňuje vytvořit znovu použitelné komponenty. U jednotlivých komponentů je potřeba definovat, jak bude každý komponent reagovat na vstup od uživatele. Pro přehlednost mezi kódem Java a značkovacím jazykem lze použít pro vkládání kódu napsaného v jazyce Java do HTML značky JSP (JavaServer Pages). Framework Wicket poskytuje i vlastní značky, které nabízí řadu možností.

Stejně jako ostatní výše uvedené frameworky i Wicket umožňuje integraci jiných frameworků (jako je Spring) a podporuje Ajax technologii. Od verze 6 dokonce tento framework začal podporovat nejpoužívanější JavaScriptovou knihovnu jQuery [3, 4].

## 2.4 Google Web Toolkit

Google Web Toolkit (GWT) je vývojový nástroj pro tvorbu RIA (Rich Internet Application). Tento nástroj nám umožňuje psát klientskou stranu v jazyce Java.

Jedinečnost tohoto nástroje spočívá v tom, že kompiluje klientský kód napsaný v jazyce Java do skriptovacího jazyka JavaScript. Kromě výhody psaní klientské strany v jazyce Java je nám také umožňuje plně debugovat celou aplikaci. Součástí Google Web Toolkit je integrace frameworku JUnit a nástroje Maven.

Google Web Toolkit poskytuje Widgets knihovnu, která obsahuje základní nástroje jako je Button, TextBox, TextArea, DatePicker nebo Rich-TextArea. V případě nenalezení vhodného nástroje umožňuje GWT rozšíření o vlastní nástroj [5].

## 2.5 Apache Tapestry

Apache Tapestry je komponentově založený framework pro tvorbu webových stránek jako je například Wicket. Už z názvu vyplývá, že se jedná o framework adoptovaný organizací Apache Software Foundation.

Tapestry na svých stránkách uvádí, že pro psaní webových stránek s použitím toho frameworku kromě jazyka Java lze využít i JVM jazyků jako je Scala nebo Groovy. Tapestry nám umožňuje snadné vytváření nových komponentů. Jednotlivé šablony aplikace lze skládat z různých komponentů. I u Tapestry dochází k používání POJOs a podpoře Ajaxu.

Od verze 5 poskytuje framework možnost automatického načtení změn u tříd a šablon. Dříve muselo dojít k opětovnému přesunutí WAR souboru do servlet kontejneru. Tato vlastnost nám umožní příjemnější a rychlejší vývoj aplikace. Proces neustálého znovuzavádění WAR souboru do servlet kontejneru je totiž nepříjemný a zdržuje vývoj aplikace [6, 7].

## 2.6 OpenXava

OpenXava je framework pro rychlý vývoj podnikových aplikací využívajících jazyk Java. Hodí se především pro aplikace, ve kterých je potřeba 4 základních vlastností, kterými jsou vytváření, načtení, upravování a smazání.

Tento framework nám umožňuje generovat uživatelská rozhraní pro webové stránky, mobilní zařízení a tablety. Programátor jen vytvoří Java třídy, kde použije anotaci OpenXava a JPA pro definování modelu. Poté dochází k využití nástrojů zjednodušeného modelu MDD (Model-Driven Development), který používá OpenXava. Využití tohoto zjednodušeného modelu nám umožňuje generovat funkcionality v čase, což je značná výhoda oproti klasickému MDD modelu, kde dochází k generování všech funkcionalit předem.

Tento framework zajišťuje jednoduché vytváření funkcí jako je export do PDF, snadné přidávání, upravování, zobrazení a smazání dat nebo použití filtrů pro vizualaci dat. Rozhraní využívá technologie Ajax, která nám zaručí efektivní práci bez obnovování stránky [8].

## 2.7 RIFE

V předchozích příkladech jsme měli framework Struts 2, který je požadavkově založený a Wicket, který je naopak komponentově založený. RIFE je specifický tím, že mísí tyto dva způsoby do sebe v podobě jejich nejlepších vlastností. RIFE slouží pro vývoj webových stránek, ale v tomto případě bez použití Java Enterprise Edition.

Tento framework se skládá z vrstev. Tyto vrstvy obsahují funkce pro vývoj webových aplikací. Každá vrstva byla navržena tak, aby byla integrovatelná s ostatními vrstvami, ale dokázala fungovat i samostatně. Při používání tohoto frameworku nejsme odkázáni na to, abychom využívali jen tyto vrstvy, i když teoreticky obsahují veškeré nutné základní funkcionality pro vývoj webu. RIFE podporuje integraci jiných frameworků jako je Spring, Hibernate nebo Velocity.

Dále nám tento framework nabízí detekci modifikací souborů, kterou poskytuje i Apache Tapestry. Tato detekce nám minimalizuje počet zavádění WAR souborů do servlet kontejneru [9].

## 2.8 Seam

Seam je framework pro vývoj webových aplikací, který se řadí do kategorie Rich Internet Application a Java Enterprise Edition. Tento framework byl vytvořen skupinou lidí, která vyvinula Hibernate.

Tento framework poskytuje komponentový model EJB (Enterprise Java Beans) 3.0. Tento model obsahuje kontejner, který řídí životní cyklus všech objektů uvnitř něho, spravuje transakce (např. commit, rollback), zajišťuje bezpečnost aplikace (ověření uživatele) a poskytuje další funkcionality. Objekt každé třídy lze vložit do tohoto kontejneru. Tento framework nám za každou cenu nevnucuje použití tohoto modelu. Ovšem je potřeba říci, že

tento model zajistí řadu potřebných funkcionalit.

Seam framework poskytuje podporu pro Java 5 anotace a efektivní tvorbu základních funkcí pro ukládání, čtení, úpravu a smazání dat s využitím Hibernate a JPA. Dále poskytuje knihovny ICEfaces a Ajax4JSF, které vznikly integrací JavaServer Faces a Ajaxu. Tyto knihovny nám umožní pracovat s Ajaxem bez nutnosti znalosti JavaScriptu [10, 11].

## 2.9 Srovnání frameworků

Na závěr této kapitoly jsem se vytvořil tabulku 2.1, ve které jsou frameworky zařazeny do různých kategorií. Porovnat jednotlivé frameworky nebylo jednoduché. Většina z nich podporuje značnou řadu stejných nebo účelově podobných nástrojů. V případě, že není potřebný nástroj v základu frameworku, tak lze většinou framework o tento nástroj rozšířit. V tabulce je kategorie komponentově založený framework, požadavkově založený framework a hybridní. Poslední z těchto tří pojmů je potřeba chápat jako kombinaci dvou předchozích typů frameworků. Dále jsou v tabulce dvě kategorie licencí. První kategorie je Apache License 2.0 a druhá GNU LGPL. V předposlední kategorii jsou obsaženy frameworky, které využívají Java Enterprise Edition platformy. Na závěr je kategorie podpora Ajaxu. Do této kategorie jsem zařadil všechny frameworky, protože každý z nich podporuje tuto technologii ať už v podobě knihovny jQuery nebo jiného nástroje.

Kategorie	Frameworky
Komponentově založený framework	Wicket, Tapestry, JSF
Požadavkově založený framework	Spring, Struts 2, OpenXava
Hybridní framework	RIFE
Apache License 2.0	Spring, Struts 2, Wicket, Tapestry
GNU LGPL license	Seam, OpenXava, RIFE
MVC architektura	Spring, Struts 2, Seam, RIFE, JSF, Tapestry
Java Enterprise Edition	Spring, Seam, Struts 2, JSF, Tapestry, GWT, OpenXava, Wicket
Podpora Ajaxu	Všechny frameworky

Tabulka 2.1: Zařazení frameworků do kategorií

## 3 Framework Spring

Spring je open-source framework pro vývoj aplikací v rámci Java EE (Java Enterprise Edition) platformy. Jako nejdůležitější vlastnost Springu se uvádí jeho modulárnost. Umožňuje nám využít jen ty moduly, které potřebujeme. Spring podporuje vytváření webových, podnikových a desktopových aplikací.

Při výběru frameworku jsem se rozhodl pro Spring ve verzi 4.3.1, která momentálně patří mezi jednu z posledních verzí. Jedním z důvodů proč jsem se rozhodl pro Spring, je velikost jeho komunity, která se právem řadí mezi jedny z největších v oblasti programování webových aplikací v jazyce Java. Dalším důvodem, proč jsem si vybral Spring, je kvalita zpracování jeho dokumentace, která je rozsáhlá a popisuje využití jednotlivých modulů. Kromě popisu modulů obsahuje dokumentace i ukázky kódů a postupy potřebné pro vývoj aplikace.

S využitím Springu dochází nejen k ušetření práce, ale i ke kvalitnímu zabezpečení aplikace. K ušetření práce při psaní aplikace dochází například integrací jednotlivých frameworků do Springu (třeba integrace Hibernate pro ORM) a pro zabezpečení aplikace lze využít framework Spring Security.

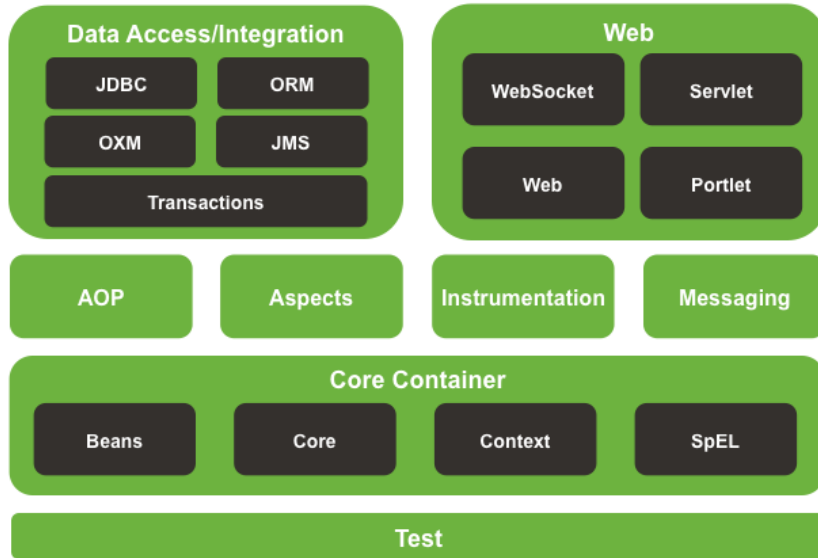
### 3.1 Historie

Původní myšlenka Springu vznikla v roce 2002, kdy Rod Johnson publikoval knihu *Expert One-on-one J2EE Design and Development*. Kniha popisovala na 750 stránkách framework Interface21, který měl usnadnit vývoj Java Enterprise Edition aplikací. Za spolupráce Jürgena Hüllera a Yanna Caroffa došlo k rozšíření původní verze. První jmenovaný se značně podílel na rozšíření kódu frameworku. Caroff se zapsal do historie především tím, že vymyslel název Spring pro framework. V červnu 2003 se oficiálně dostal Spring na veřejnost [12].

### 3.2 Moduly

Framework poskytuje okolo 20 modulů, které se používají podle potřeby dané aplikace. Tyto moduly jsou seskupeny do 8 částí (Data Access/Integration,

Web, AOP, Aspects, Instrumentation, Messaging, Core Container a Test). Na obrázku 3.1 lze vidět seskupení modulů. Pro popis jednotlivých modulů bylo využito zdroje [13].



Obrázek 3.1: Moduly Springu [13]

## Core Container

- Core - Modul, který poskytuje základní části frameworku včetně Dependency Injection funkcí a IoC (Inversion of Control) kontejneru, který slouží pro správu objektů.
- Beans - Poskytuje implementaci továrního vzoru, který je nazván BeanFactory.
- Context - Založený na základě modulů Core a Beans. Poskytuje přístup k definovaným a nakonfigurovaným objektům. Obsahuje Application-Context rozhraní pro přístup třetích stran.
- SpEL - Modul poskytující výrazový jazyk pro dotazování a manipulaci s objekty.

## Data Access/Integration

- JDBC - Obsahuje JDBC (Java Database Connectivity) abstraktní vrstvu, která slouží pro práci s databázemi.

- ORM - Modul poskytující integrační vrstvu pro mapování relačních objektů.
- OXM - Poskytuje abstraktní vrstvu, která podporuje Objekt/XML mapování.
- JMS - Obsahuje funkce pro produkování a zpracování zpráv. Od verze Springu 4.1 poskytuje integraci s modulem Messaging.
- Transactions - Modul pro podporu řízení transakcí.

## Web

- Web - Modul poskytuje základní webově orientované integrační funkce, jako je multipart file upload a inicializace IoC (Inversion of Control) kontejneru.
- WebSocket - Poskytuje podporu pro obousměrnou komunikaci mezi klientem a serverem.
- Servlet - Obsahuje implementaci MVC architektury pro webové aplikace.
- Portlet - Modul umožňující realizaci MVC. Dochází zde k použití funkcionality Web-Servlet modulu.

## Ostatní moduly

- AOP - Modul poskytující implementaci aspektově orientovaného programování.
- Aspects - Tento modul slouží pro propojení s AspectJ, které se využívá při aspektově orientovaném programování.
- Instrumentation - Poskytuje podporu pro dynamické načítání tříd do JVM (Java Virtual Machine).
- Messaging - Slouží pro podporu síťové komunikace.
- Test - Modul pro podporu testování JUnit a TestNG.



## 4 Použité technologie

Pro tvorbu webové aplikace bude využito MVC architektury, která rozděluje aplikaci do 3 logických částí. Písmeno M (Model) reprezentuje data a byznys logiku, písmeno V (View) odpovídá jednotlivým šablonám a na závěr písmeno C (Controller) slouží jako prostředník mezi klientem a modelem.

Jak už jsem zmínil v kapitole popisující Spring, rozhodl jsem se použít tento framework jako základ pro tvorbu mé webové aplikace. Pro zajištění bezpečnosti použiji Spring Security (ve verzi 3.2.8.). Dále pak pro vkládání kódu napsaného v jazyce Java do HTML5 značek bude využito technologie JSP (JavaServer Pages). Tato technologie poskytuje vlastní značky, které mi umožní efektivně vytvářet obsah jednotlivých webových stránek. Pro lepší prezentaci stránek jsem se rozhodl použít CSS až s podporou verze 3, která mi umožní zlepšit vizualizaci dat oproti použití samotného HTML5.

Pro zpříjemnění práce s webovou aplikací na straně klienta využiji skriptovacího jazyka JavaScript a knihovny jQuery (ve verzi 1.12.4). Použití této knihovny a skriptovacího jazyka umožní například dynamicky vytvářet textové elementy formuláře nebo posílat požadavek na server, zpracovat ho a vrátit odpověď bez obnovení stránky. Dále pak především ke zkvalitnění vzhledu webových stránek použiji framework Bootstrap (ve verzi 3.3.7), který nabízí škálu CSS stylů a řadu JavaScript pluginů.

Jako další nástroj jsem vybral Hibernate ORM ve verzi 4.3.11 pro konverzi dat mezi relační databází a objektově orientovaným programovacím jazykem. Na závěr použiji plugin DataTables (ve verzi 1.10.13), který umožní efektivní práci se získanými výsledky závodu.

### 4.1 HyperText Markup Language

HyperText Markup Language (HTML) je značkový jazyk, který se používá pro tvorbu webových stránek. Slovo HyperText vyznačuje metodu, která umožňuje pohyb po webu klikáním na hypertextové odkazy. Kromě přesunu v kontextu našeho webu umožňuje hypertextový odkaz přesun kamkoli v rámci rozsahu systému World Wide Web.

Slovo Language nám říká, že se jedná o jazyk. To nám udává, že stejně

jako ostatní jazyky má i tento definovaný soupis pravidel, který ho popisují a umožňují práci s ním. Slovo Markup pak popisuje co dělají HTML značky s textem uvnitř nich (jak bude text v prohlížeči zobrazen). Značky pak lze podle jedné ze základních definic rozdělit na párové a nepárové. Párové musejí mít odpovídající ukončovací značku k sobě. Oproti tomu nepárové nemají odpovídající ukončovací značku [14].

## 4.2 Cascading Style Sheets

Kaskádové styly se podílejí na prezentaci elementů napsaných ve značkovacím jazyce. CSS vzniklo především proto, aby oddělilo základní strukturu danou značkovacím jazykem a obsah od vzhledu, a také umožnilo rozsáhlejší možnosti zobrazení dokumentu, než umožňují značkovací jazyky.

Jednotlivé styly pak lze psát různými způsoby. Prvním způsobem je vepsání stylů na konkrétní stránce do párové značky style. Jako další způsob lze CSS do atributu style vepsat u příslušného elementu. Posledním způsobem je psaní stylů do odděleného souboru, který lze importovat pomocí značky `link` nebo využitím `@import`. Soubor obsahující styly lze importovat z lokálního i externího úložiště [15].

## 4.3 Bootstrap

Bootstrap je open-source front-end framework vyvinutý společností Twitter. Framework obsahuje šablony návrhu založené na HTML a CSS pro formuláře, tlačítka, tabulky, navigaci, modální okna a další. Součástí tohoto frameworku jsou JavaScriptové pluginy. S využitím Bootstrapu můžeme snadno definovat responzivní design webových stránek, který se přizpůsobí konkrétnímu zařízením [16].

Bootstrap není jediným frameworkem svého druhu. Existuje spousta front-end frameworků. Dalším příkladem lze pak uvést Material Design Lite (MDL) nebo Semantic-UI. Při výběru jsem se rozhodl pro Bootstrap především proto, že už jsem s ním měl osobní zkušenost. Dalším důvodem je kvalitně zpracovaná dokumentace a obrovská komunita lidí, kterou má tento framework.

## 4.4 Javascript a jQuery

JavaScript je objektově orientovaný skriptovací jazyk. V JavaScriptu oproti ostatním jazykům jako je C, C++ nebo Java nedochází ke specifikaci typu proměnné. JavaScriptový kód nebo import skriptu na straně klienta najdeme v zobrazené šabloně. Skriptům vloženým v těchto dokumentech je umožněno využít DOM (Document Object Model) k modifikaci daných dokumentů a poté následně zobrazit provedené změny v prohlížeči. DOM poskytuje API, které definuje, jak přistoupit k jednotlivým objektům, které tvoří dohromady dokument [17].

Pro zjednodušení psaní skriptů lze využít jQuery knihovnu, která poskytuje víceúčelovou abstraktní vrstvu. S použitím této knihovny můžeme snáze přistupovat k elementům dokumentu, upravovat CSS styly a měnit obsah dokumentu. Dále můžeme pomocí technologie Ajax posílat požadavek na server, zpracovat ho a vrátit odpověď bez obnovení stránky [18].

## 4.5 Hibernate

Hibernate je open-source framework umožňující objektově-relační mapování (ORM). Tento framework poskytuje spoustu důležitých rozhraní. Avšak jako dvě základní pro tvorbu dotazů bych uvedl rozhraní `Query` a `Criteria`. První jmenované rozhraní nám umožní provádět dotazy a kontrolovat jejich vyhodnocení. Kontrolou vyhodnocení je myšleno například nastavení limitu omezující počet vrácených objektů (hodnot). Jednotlivé dotazy mohou být napsány buď v HQL (Hibernate Query Language) nebo v klasickém SQL. Druhé jmenované rozhraní nám umožní definovat kritéria pro vytvoření a vyhodnocení dotazu [19].

## 4.6 DataTables

DataTables API umožňuje provádět operace nad inicializovanou tabulkou. K inicializaci DataTables tabulky může dojít například pomocí jQuery příkazem `$("#identifikátorTabulky").DataTable()`. Využití API umožňuje používat funkce, které nám například umožní měnit hodnoty v jednotlivých řádcích [20].

Vizualizace a možnosti tabulky se liší podle námi definovaných hodnot. DataTables umožňuje vyhledávat v tabulce, řadit jednotlivé hodnoty sloupců abecedně pro řetězce nebo podle číselné hodnoty u čísel. Další funkcionalitou může být výběr počtu zobrazených záznamů v tabulce nebo export dat například ve formátu PDF, XLSX a CSV.

# 5 Návrh aplikace

Tato kapitola obsahuje požadavky, které jsou kladené na vytvoření webové aplikace. Druhou podkapitolou je rozdělení uživatelů do tříd a popis jejich vztahu k aplikaci. Poté následuje návrh funkcionalit webu. V závěru této kapitoly je návrh databáze.

## 5.1 Požadavky na aplikaci

Na aplikaci jsou kladeny různé požadavky. Na tyto požadavky lze pohlížet například z hlediska bezpečnosti nebo funkcionalit webu. Soupis těchto nároků jsem rozdělil do dvou skupin na funkční a mimofunkční požadavky.

### Funkční požadavky

- Vytváření závodu - U vytváření závodu je kladen požadavek pro definování vlastních kategorií, které nám umožní rozdělit týmy nebo závodníky do jednotlivých kategorií. Rozdělení týmů do kategorií nám umožní identifikovat skupiny, které mohou mít například určenou jinou trasu závodu nebo určený jiný čas pro dokončení závodu.
- Registrace - Aplikace musí umožnit registraci uživatelů už před závodem přes webové stránky. Účelem tohoto je mít většinu závodníků zaregistrovaných už před závodem.
- Správa týmu - Tento požadavek by měl umožnit upravovat změnu kategorie a název týmu. Dalším požadavkem tohoto bodu je exportování a importování týmů ve formátu XLSX.
- Správa výsledků - Správa musí obsahovat filtry, které nám umožní rychleji se zorientovat při zadávání výsledků, času pro start a dokončení závodu týmu.
- Zobrazení výsledků - Vizualizace výsledků musí umožnit filtrování podle kategorií, seřazení jednotlivých hodnot v sloupcích a vyhledávání v tabulce. Dalším požadavkem je umožnění exportování dat pro správce závodu a jeho vlastníka ve formátu XLSX a CSV.

- Seznam závodníků - Tento seznam musí umožnit měnit údaje o závodnících. K použití toho seznamu typicky dochází při placení startovného před závodem, kde se mění hodnota údaje neplatil na platil.

### **Mimofunkční požadavky**

- Stabilita aplikace - Tento nárok je důležitý pro chod aplikace. Počítá s použitím této aplikace na místě pro správu závodu. Nemůže se stát, že by aplikace při zadávání dat spadla, stala se nepoužitelnou a neumožnila by nám dále spravovat závod.
- Přehlednost - Systém musí být přehledný, protože je kladen důraz na rychlost zadávání výsledků.
- Bezpečnost - Webová aplikace musí být zabezpečena proti vkládání skriptů a HTML značek, které by mohly poškodit obsah webu nebo dokonce zničit záznamy v datovém úložišti. Dále musí být zabezpečeny konkrétní funkce odpovídající správcům a vlastníkům závodů, aby se tam nedostal neoprávněný uživatel.

## **5.2 Třídy uživatelů**

Pro potřeby webové aplikace jsem se rozhodl rozdělit uživatele do čtyř tříd. První třídou jsou běžní návštěvníci webu, kteří nejsou přihlášení do webové aplikace. Druhou třídou jsou registrovaní uživatelé, na které bude při návrhu pohlíženo jako na webu přihlášené uživatele. Třetí třídou jsou uživatelé, kteří vytvořili závod. Při popisu návrhu na ně bude pohlíženo jako na vlastníky závodů. Poslední třídou uživatelů jsou správci závodu. Správcem závodu se stane uživatel po přiřazení vlastníkem závodu do seznamu správců.



Obrázek 5.1: Třídy uživatelů - Use Case Diagram

## 5.3 Funkcionality webu

### Registrace a přihlášení uživatele

První základní funkcionalitou webu bude registrace uživatele a přihlášení. Při registraci bude uživatel zadávat unikátní přihlašovací jméno, email a telefonní číslo. Dalšími zadávanými údaji bude křestní jméno, příjmení a nakonec dvakrát heslo pro ověření jeho správnosti. Minimální délka hesla bude omezena na osm znaků.

Přihlášení uživatele do aplikace bude nutnou podmínkou pro vytvoření závodu a jeho správu. Další důležitou funkcionalitou pro přihlášeného uživatele bude umožnění registrace do závodů.

### Vytváření závodů

Jak už bylo zmíněno při popisu registrace a přihlášení uživatele, tak přihlášení bude nutnou podmínkou, která umožní uživateli vytvořit závod. Při

vytváření závodu se bude zadávat unikátní název závodu a velikost týmu, která bude omezena na 1 až 5 závodníků.

Součástí vytvoření závodu bude definování podkategorií pro jednotlivé závodníky a týmy. Definování podkategorií týmů (závodníků) půjde rozdělit do třech možností:

- Definování bez podkategorií
- Výběr z předpřipravených kategorií - Každá tato kategorie bude obsahovat podkategorie. Jednotlivé předpřipravené kategorie a jejich podkategorie se budou lišit u závodníků i týmů.
- Vytvoření vlastních podkategorií - Při této volbě půjde vytvořit podkategorie s názvem o velikosti 1-20 znaků. Maximální počet vlastních definovaných podkategorií bude omezen na počet dvacet.

## **Seznam závodů**

Pro potřeby webové aplikace budou závody rozděleny na dva seznamy. První seznam bude obsahovat závody, které se budou konat. V tom seznamu bude vždy odkaz na závod, na jeho registrační formuláře, a také zobrazeno, jestli je umožněna registrace do závodu nebo není. Druhý seznam bude obsahovat už vyhodnocené závody, u kterých bude odkaz na závod a na jeho výsledky. Toto zobrazení seznamů bude pro všechny třídy uživatelů stejné.

Z pohledu uživatelských účtů bude součástí každého účtu seznam, který bude jmenovat moje závody. Tento seznam bude obsahovat konkrétní závody vytvořené uživatelem, a také závody, ve kterých je uživateli přiřazena funkce správce závodu. Seznam bude obsahovat odkazy na tyto závody.

## **Registrace závodníků a týmů**

Registrace bude rozdělena do dvou sekcí. První sekce bude určena pro vlastníka závodu a správce závodu. Druhá sekce bude pro běžně přihlášené uživatele, kteří se chtějí přihlásit do závodu. Množství formulářů v jednotlivých sekcích se bude lišit podle nastavení velikosti týmů při registraci. Pro nepřihlášené uživatele tato sekce nebude dostupná.



Pro jednočlenné týmy bude administrátorská sekce obsahovat jeden formulář, ve kterém půjde vyplnit křestní jméno, příjmení, telefonní číslo, email a zaplacení startovného. Ani na jeden ze základních atributů závodníka nebude nahlíženo jako na unikátní údaje. Dalšími údaji, které půjde vybrat bude kategorie týmu a závodníka<sup>1</sup>.

Při definovaném větším počtu závodníků v týmu než jedna bude k dispozici podobný formulář jako u formuláře o velikosti jedna. Tento formulář se bude lišit ve výchozím stavu počtem závodníků pro vyplnění, který bude určen podle definovaného počtu závodníků v týmu. Pro potřebu zvětšení počtu závodníků v týmu budou přidána tlačítka, které umožní rozšířit tým až o dva členy. Další změnou oproti předchozímu formuláři je přidání názvu týmu, který nebude nutné při vyplňování formuláře zadat. Největší změnou při velikosti týmu větší než jedna bude druhý formulář, který umožní přidat závodníka bez týmu. Tento formulář bude obsahovat stejné atributy závodníka jako předchozí formulář.

Druhá sekce pro přihlášené uživatele bude při velikosti týmu jedna obsahovat jeden formulář, ve kterém se bude vybírat podkategorie pro tým i závodníka<sup>1</sup>. Ostatní potřebné údaje o závodníkovi budou vzaty z jeho uživatelského účtu.

U velikosti týmu větší než jedna bude formulář srovnatelný s administrátorským formulářem pro registraci týmu s počtem závodníků větším než jedna. Rozdíl ve formulářích bude v tom, že u prvního závodníka budou osobní data vzaty z jeho uživatelského účtu a bude neměnný počet členů v týmu. Formulář navíc nebude u závodníků obsahovat atribut platby. Při velikosti větší než jedna bude tato sekce obsahovat ještě druhý formulář, který umožní uživatele zaregistrovat jako závodníka bez týmu. Tento formulář je totožný formuláři v sekci přihlášený uživatel při velikosti jedna. Rozdíl v nich bude v tom, že v tomto případě nepůjde závodníkovi přiřadit týmovou kategorii.

## **Správa seznamu týmů**

Na seznam týmů budeme moci pohlížet dvěma pohledy. První pohled je ze strany přihlášených uživatelů a nepřihlášených uživatelů do systému. Tento pohled je pak potřeba rozdělit podle výchozího nastavení velikosti týmu

---

<sup>1</sup>v případě jejich definování

u závodu. Při definování velikosti týmu jedna budou zobrazeny údaje o jednotlivých závodnících včetně přiřazené podkategorie závodníka i týmu<sup>2</sup>. Při tomto pohledu nebude vidět email a ani telefonní číslo závodníka. Pro případ velikosti týmu větší než jedna bude navíc vidět název týmů a místo jednoho závodníka bude zobrazen seznam členů týmu.

Druhý pohled bude sloužit pro správce závodů a vlastníky závodu. Při definované velikosti týmu jedna bude každá řádka seznamu odpovídat jednomu závodníkovi. V tomto seznamu bude možné upravovat osobní údaje závodníků včetně změny podkategorií a atributu platil. Součástí každé řádky bude ještě ID pro identifikaci závodníka a tlačítko pro smazání. Seznam bude obsahovat tlačítka pro exportování a importování závodníků ve formátu XLSX.

Při definované velikosti týmu větší než jedna bude v seznamu zobrazeno ID týmu, název týmu, jeho členové a týmová podkategorie<sup>2</sup>. Součástí toho seznamu bude editace podkategorie<sup>2</sup>, názvu týmu a tlačítko pro smazání týmu. Po zmáčknutí tohoto tlačítka se zobrazí modální okno, ve kterém půjde vybrat, jestli chceme smazat tým i s jeho členy. Součástí toho seznamu bude i importování a exportování týmů ve formátu XLSX.

## **Správa seznamu všech závodníků**

Sekce obsahující seznam všech účastníků bude dostupná jen pro vlastníky závodu a správce závodu. Ostatní uživatelé zde nebudou mít přístup. Tato sekce bude sloužit pro úpravy osobních údajů závodníků včetně nastavení kategorie závodníka a úpravy atributu platby. Součástí každého záznamu bude navíc ID týmu. Tato sekce bude sloužit jen pro závody s definovanou velikostí týmu větší než jedna.

## **Tvorba týmu**

Tato sekce bude dostupná jen pro závody, které budou mít definovanou velikost týmu větší než jedna. Pro běžné uživatele bude v této sekci zobrazen pouze seznam závodníků, kteří nemají tým. Pro tvůrce závodu zde bude podobný seznam, který bude rozšířený o větší množství informací o závodnících, a také zde bude umožněno pracovat s tímto seznamem. Prvním rozšířením bude vytvoření týmu výběrem závodníků ze seznamu. Při výběru

---

<sup>2</sup>v případě definování podkategorií pro závod

závodníků půjde doplnit název týmu. V případě definovaných týmových kategorií bude nutností vybrat příslušnou kategorii. Kromě tvorby týmu bude umožněno i mazat jednotlivé závodníky ze seznamu.

## **Zobrazení výsledků**

Při zobrazování výsledků závodu bude využito pluginu DataTables, který poskytuje řadu vhodných funkcí pro toto zobrazení. Tento plugin nám umožní řadit hodnoty sloupců sestupně a vzestupně. Další funkcí, která bude využita je vyhledávání přes všechny sloupce, která podle zadané hodnoty vyfiltruje příslušné odpovídající řádky.

V zobrazené tabulce budou hodnoty seřazeny podle výpočtu celkových bodů týmů (závodníků). Tento výpočet bude zahrnovat součet získaných bodů a bonusových bodů, od kterých bude odečtena hodnota penalizace. Hodnota penalizačních bodů bude mít přiřazený jeden bod za každou započatou minutu, která přesáhne určený čas pro závod. Součástí této tabulky bude umístění, název týmu s členy, body, bonusové body, penalizace, startovní čas, celkové body, čas dobehnutí a čas délky běhu určené pro tým. V případě, že bude mít závod přiřazené týmové podkategorie, tak i ty budou zobrazeny v tabulce u jednotlivých týmů. Při definované velikosti týmu jedna bude sloupec s názvem týmu a jeho členy odpovídat jednomu závodníkovi.

Všechny výše popsané vlastnosti budou zobrazeny a umožněny pro všechny třídy uživatelů. Pro vlastníka závodu a správce závodu bude navíc přidána možnost exportu výsledků ve formátu XLSX a CSV. Export dat se bude vždy lišit podle aktuálně vyfiltrovaných dat v tabulce.

## **Správa výsledků**

Tato sekce bude určena pro vlastníka závodu a správce závodu. Ostatní třídy uživatelů zde nebudou mít přístup. V této sekci se budou nastavovat výsledky pro jednotlivé týmy (závodníky). Tabulka bude stejně jako při zobrazování výsledků implementována pluginem DataTables s možností řazení sloupců a vyhledáváním. U této sekce budou navíc přidány tři tlačítka pro rychle filtrování. První tlačítko zobrazí všechny týmy (závodníky), druhé zobrazí pouze řádky, u kterých není vyplněn čas dokončení závodu. Poslední tla-

čítka filtrování zobrazí řádky, kde bude vyplněn čas dokončení závodu. Tyto tlačítka budou implementována především pro lepší orientaci při zadávání výsledků.

Tabulka bude obsahovat jméno týmu s jeho členy, týmovou kategorii<sup>3</sup>, body, bonusové body, čas dokončení, startovní čas a poslední atribut bude zvolený čas, do kterého by měl tým (závodník) dokončit závod. U každé řádky půjde ručně měnit hodnoty bodů, startovního času, bonusových bodů a času dokončení závodu. Každá řádka navíc bude obsahovat tlačítko, které po jeho zmáčknutí přiřadí aktuální čas jako čas dokončení pro příslušnou řádku.

Pro lepší práci s přiřazování startovního času bude přidáno modální okno, které se zobrazí po zmáčknutí na tlačítko. V tomto okně půjde přiřadit startovní čas pro všechny týmy (závodníky), pro dalších deset bez startovního času nebo pro týmy s příslušnou kategorií. Druhým vyskakovacím oknem bude formulář pro nastavení délky závodu. Čas pro délku závodu se bude zadávat v minutách. Hodnotu pak půjde přiřadit pro všechny týmy nebo pro konkrétní týmy s kategorií.

## Konfigurace závodu

Jedná se o sekci, která bude určena pro vlastníky závodu. V této sekci půjde blokovat a povolovat registraci do závodu pro přihlášené uživatele. Dále zde bude možné zařadit závod do skupiny vyhodnocených závodů. V této sekci také bude možno přidat a odebrat správce závodu. Přidávání uživatelů mezi správce bude prováděno přes unikátní přihlašovací jméno. Poslední možností bude smazání závodu, u kterého bude při provádění této akce potřeba zadat heslo uživatele.

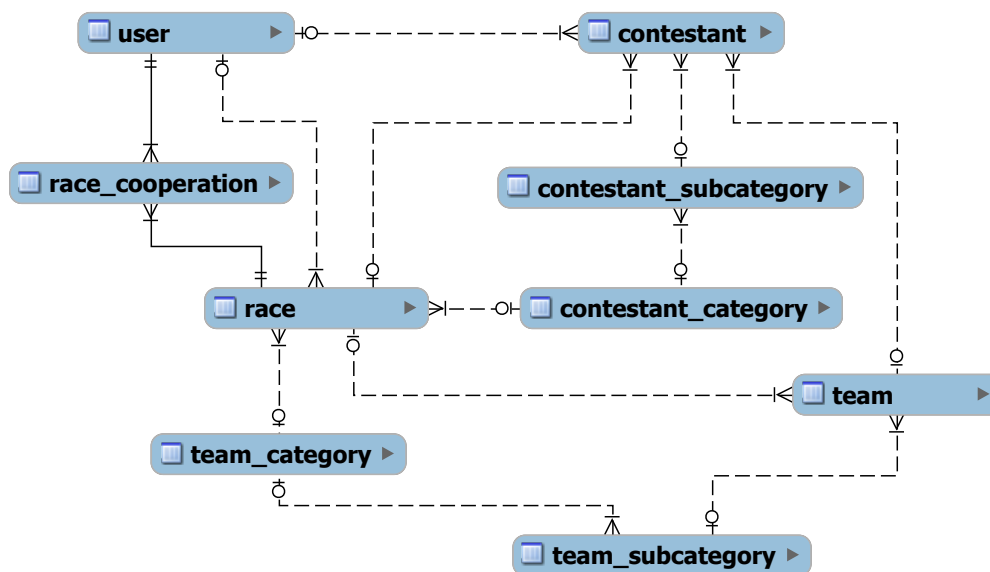
## 5.4 Návrh databáze

Při návrhu databáze jsem vytvořil ER diagram (obrázek 5.2) s devíti tabulkami, které obsahují třináct relací. Tento návrh nám zaručí možnost ukládání veškerých potřebných dat pro potřeby webové aplikace. ER diagram (obrázek 5.2) pouze popisuje relace mezi jednotlivými tabulkami. Podrobnější

---

<sup>3</sup>v případě definování týmových podkategorií závodu

popis tabulek v podobě ERA digramu je přiložen v příloze.



Obrázek 5.2: Návrh databáze - ER diagram

## Entita user

Tabulka `user` slouží pro registraci uživatele do systému. Tato entita se zdá částečně podobná tabulce `contestant`. Zmíněná entita oproti tabulce `user` obsahuje záznamy závodníků pro konkrétní závody. Entita je v relaci 1:N s třemi tabulkami, kterými jsou tabulky `contestant`, `race_cooperation` a `race`.

- `id` - Atribut typu `int(11)` sloužící jako jednoznačná identifikace registrovaného uživatele. K vytvoření tohoto čísla dochází podle příslušné hodnoty (`AUTO_INCREMENT`) automaticky při vkládání záznamu do databáze.
- `login` - Tento atribut slouží pro přihlašování uživatelů do webové aplikace, a proto je unikátní. Atribut má dán typ `varchar(32)`, který omezuje maximální velikost loginu na hodnotu 32 znaků.
- `email` - Stejně jako u atributu `login` se jedná o unikátní atribut. Typ toho atributu je `varchar(32)`.

- phone - Telefonní číslo je jediný atribut u tabulky `user`, který může být mít hodnotu NULL. Typ toho atributu je `varchar(13)`.
- firstname - Atribut pro zadání křestního jména uživatele. Typ atributu je `varchar(32)`.
- surname - Jedná se o atribut odpovídající příjmení uživatele. Typ atributu je `varchar(32)`.
- password - Heslo uživatele, které je omezeno velikostí 256 znaků.

### Entita `race_cooperation`

Tato entita umožňuje přidat libovolný počet správců k příslušnému závodě. Hlavním účelem této tabulky je pomoc vlastníkově závodů s registrací účastníků a případným zadáváním výsledků týmů. Primární klíčem byla určena skupina atributů `user_id` a `race_id`, kde jejich kombinace zaručuje jedinečnost. Tabulka je ve 2 relacích typu N:1 s tabulkami `user` a `race`.

- `user_id` - Cizí klíč, který odkazuje na primární klíč tabulky `user`.
- `race_id` - Jedná se také o cizí klíč jako u atributu `user_id`, akorát v tomto případě odkazuje na primární klíč tabulky `race`.

### Entita `race`

Tabulka `race` obsahuje záznamy odpovídající závodům. Entita má 2 vazby typu N:1 s tabulkami `team_category` a `user`. Kromě 2 relací typu N:1 má tabulka 3 relace typu 1:N s tabulkami `race_cooperation`, `team` a `contestant`.

- `id` - Atribut `id` je primární klíč tabulky typu `int(11)`.
- `name` - Název závodu, který může mít maximální velikost 32 znaků.
- `team_size` - Udává základní velikost týmu, která je určena pro registraci týmů do závodu. Tento atribut je typu `int(1)`.
- `registration` - Jedná se o atribut typu boolean. Tento atribut povoluje a zakazuje registraci do závodu. V případě hodnoty `true` je registrace povolena. Pro hodnotu `false` je zakázána.

- `evaluation` - Atribut typu `boolean`, který umožňuje zařadit závod do seznamu vyhodnocených závodů. Hodnota `true` ho označí jako vyhodnocený a hodnota `false` jako nevyhodnocený.
- `con_category_id` - Cizí klíč, který odkazuje na primární klíč tabulky `contestant_category`. Tento atribut umožňuje přiřadit závodu kategorii, která umožní při registraci charakterizovat účastníky kategorií z tabulky `contestant_subcategory`. Typickým příkladem lze určit kategorii pohlaví, která bude mít podkategorii muž a žena.
- `team_category_id` - Jako v případě `con_category_id` se jedná o cizí klíč fungující na podobném principu. V tomto případě je závodu přiřazena kategorie z tabulky `team_category`, která umožní charakterizovat týmy podkategoriemi z tabulky `team_subcategory`.
- `user_id` - Cizí klíč odkazující na primární klíč tabulky `user`. Slouží pro identifikaci uživatele, který vytvořil závod.

### Entita `team_category`

Tabulka, která umožňuje při vytváření závodu definovat vlastní podkategorie pro týmy nebo přiřadit týmu výchozí dostupnou kategorii. Tabulka je v relaci 1:N s tabulkou `race`, což nám umožní přiřazení kategorie závodu. Entita je v relaci 1:N s tabulkou `team_subcategory`, kde tato relace záznamy z tabulky `team_subcategory` přiřazuje ke konkrétnímu záznamu z tabulky `team_category`.

- `id` - Atribut `id` je primární klíč typu `int(11)`.
- `name` - Název kategorie, který nabývá při definování vlastní skupiny podkategorií hodnoty `NULL`. Atribut je typu `varchar(20)`.
- `default_category` - Atribut je typu `boolean`. Hodnota `true` pro výchozí kategorie pro přiřazení k závodu. V opačném případě je hodnota `false`.

### Entita `team_subcategory`

Tabulka obsahující podkategorie přiřazené kategorii z entity `team_category`. Relace s touto tabulkou je N:1. Dále je entita ve vztahu 1:N s tabulkou `team`, což umožní přiřazení konkrétní podkategorie příslušnému týmu.

- id - Atribut id je primární klíč typu int(11).
- name - Název podkategorie, který je typu varchar(20).
- category\_id - Odkazuje na primární klíč tabulky `team_category`.

## Entita `team`

Entita `team` slouží pro vytvoření týmu. Tabulka je vztahu 1:N s entitou `contestant`, což nám umožní přiřazení závodníka do týmu. Tabuobsahuje další dvě relace typu N:1, které slouží pro přiřazení týmu k závodu a přiřazení hodnoty týmové podkategorie.

- id - Atribut id je primární klíč typu int(11).
- name - Název týmu, který je typu varchar(32). Atribut nabývá hodnoty NULL při definované velikosti týmu jedna.
- points - Udává počet bodů, který tým získal. Atribut je typu int(11). Výchozí hodnota je 0.
- bonus - Atribut typu int(11) a udává počet získaných bonusových bodů. Výchozí hodnota je 0.
- start\_time - Startovní čas týmu, který je typu datetime. Výchozí hodnota je NULL po vytvoření týmu.
- finish\_time - Atribut typu datetime, který udává čas dokončení závodu týmu. Výchozí hodnota je NULL.
- deadline\_time - Čas zadaný v minutách, který udává časové omezení závodu. Slouží pro výpočet penalizace. Atribut je typu int(11).
- team\_subcat\_id - Cizí klíč, který odkazuje na primární klíč tabulky `team_subcategory`. Tento atribut slouží pro přiřazení podkategorie týmu. Atribut může mít hodnotu NULL pro případ, že by závod neměl definovanou týmovou kategorii.
- race\_id - Sekundární klíč odkazující na entitu `race`. Přiřazuje tým ke konkrétnímu závodu.



## **Entita `contestant_category`**

Entita odpovídající funkčnosti tabulky `team_category`. Cílem této entity je rozšířit možnost informací o závodníkovi v podobě umožnění definovat kategorií pro závod obsahující podkategorie. Tabulka je v relaci 1:N s tabulkami `race` a `contestant_subcategory`.

- `id` - Atribut `id` je primární klíč typu `int(11)`.
- `name` - Název kategorie, který nabývá při definování vlastních podkategorií hodnoty `NULL`. Atribut je typu `varchar(20)`.
- `default_category` - Atribut je typu `boolean`. Hodnota `true` pro výchozí kategorie k použití. V opačném případě je hodnota `false`.

## **Entita `contestant_subcategory`**

Tabulka obsahující podkategorie závodníků. Je ve vztahu 1:N s tabulkou `contestant` a N:1 s tabulkou `contestant_category`.

- `id` - Atribut je primární klíč typu `int(11)`.
- `name` - Název podkategorie, který je typu `varchar(20)`.
- `category_id` - Odkazuje na primární klíč tabulky `team_category`.

## **Entita `contestant`**

Entita obsahující záznamy o závodnících. Je ve vztahu N:1 s tabulkou `user`, `race`, `contestant_subcategory` a `team`.

- `id` - Je primární klíč typu `int(11)`.
- `firstname` - Atribut odpovídající jménu závodníka. Je typu `varchar(32)`.
- `lastname` - Příjmení závodníka, které je typu `varchar(32)`.
- `email` - Email, který má omezení 32 znaků.

- phone - Telefonní číslo, které je typu varchar(13).
- paid - Atribut typu boolean. Hodnota true když závodník zaplatil. V opačném případě hodnota false.
- user\_id - Cizí klíč tabulky user. Odpovídá uživateli, který zaregistroval účastníka.
- race\_id - Sekundární klíč přiřazující závodníka k závodu.
- con\_subcat\_id - Cizí klíč zařazující závodníka do jedné z podkategorií podle definování závodu. Hodnota NULL v případě, že závod nemá definovanou kategorii.
- team\_id - Sekundární klíč zařazující závodníka do týmu. Hodnota NULL pro případ, že se uživatel zaregistroval jako osoba bez týmu.

# 6 Implementace

Tato kapitola popisuje implementaci webové aplikace. Součástí implementace je popis struktury projektu. Dále jsou součástí této kapitoly části kódu a důležité algoritmy pro chod webové aplikace.

## 6.1 Struktura projektu

Struktura projektu byla ovlivněna především použitím architektury MVC. Dalším krokem organizace projektu bylo uspořádání do rozšířitelné a přehledné struktury. Pro potřebu webové aplikace bylo vytvořeno 7 komponent (balíčků) obsahujících dohromady 63 tříd. Dále je obsahem projektu složka nazvaná webapp, která obsahuje podsložky Public, WEB-INF a www. Adresář Public dále obsahuje složky css, images a js. Uvnitř adresáře WEB-INF jsou základní konfigurační soubory pro webovou aplikaci. Dále obsahuje složku tags, ve které jsou definovány značky pro importování šablon, u kterých dochází k opakující se potřebě u jiných šablon. Složka www se dále rozděluje na další podadresáře, které obsahují JSP soubory.

### 6.1.1 Komponenty

#### Controller

- AuthenticationController - Pro přihlášení a odhlášení do systému.
- AvailableRacesController - Odpovídá seznamu dostupných závodů pro registraci závodníků.
- ContestantController - Pro zobrazení všech závodníků v závodu. Obsahuje metody pro změnu údajů o závodníkovi.
- CreateRaceController - Odpovídá pohledu pro vytvoření závodu a obsahuje metody pro zpracování tohoto formuláře.
- ErrorController - Obsahuje metodu pro zobrazení HTTP chybových zpráv.
- EvaluatedRacesController - Slouží k zobrazení seznamu vyhodnocených závodů.

- HomeController - Obsahuje metodu pro zobrazení úvodní stránky webu.
- MyRacesController - Pro zobrazení seznamu vlastních závodů uživatele včetně zobrazení závodů, ve kterých je uživatel přiřazen jako správce.
- RaceController - K zobrazení stránky pro konfiguraci závodu. Obsahuje metody umožňující změnit nastavení závodu a upravit seznam správců závodu.
- RaceRegistrationController - Pro registraci závodníků a týmů do závodu.
- RaceResultController - K zobrazení a správě výsledků týmů. Obsahuje metody pro zadání startovního času, bodů, času dokončení závodu a nastavení délky závodu v minutách.
- SoloContestantController - Pro vytvoření týmu ze závodníků, kteří se registrovali samostatně bez týmu. Obsahuje i metodu pro mazání těchto závodníků.
- TeamController - Obsahuje metodu pro zobrazení seznamu týmů. Dále obsahuje metody pro úpravu názvu týmu, týmové kategorie nebo smazání týmu.
- UserController - K zobrazení profilu uživatele.
- UserRegistrationController - Pro registraci uživatele do systému.

## Form

Tato komponenta obsahuje třídy, které slouží pro zabalení získaných dat z formulářů na serveru. Objekty této třídy můžeme nalézt v parametrech metod pro zpracování dat z formulářů. Tyto objekty bývají označeny anotací `@ModelAttribute` nebo `@RequestBody`. Druhá jmenovaná anotace se používá v případě, že jsou data posílaná od klienta ve formě JSONu. Využití těchto kontejnerů nám umožní efektivně pracovat se získanými daty. Při použití první anotace lze využít objektu třídy `BindingResult`. Tento objekt musí být jako parametr v metodě za zmiňovanou anotací. Důvodem použití objektu této třídy je především metoda `hasErrors()`, která nám zachytí například chybu přiřazení řetězce do objektu `Integer`. Výpis 6.1 slouží k pochopení předchozích vět. Jedná se o zjednodušenou ukázkou metody pro registraci týmu, která je v třídě `RaceRegistrationController`.

```

1 @RequestMapping(value = "/race/{id}/teamRegistration",
    method = RequestMethod.POST)
2 public @ResponseBody String teamRegistration(
    @ModelAttribute("teamRegForm") TeamRegForm
    teamRegForm, BindingResult bindingResult,
    @PathVariable("id") int race_id) {
3     //zachyti prirazeni retezce objektu Integer
4     if (bindingResult.hasErrors()) {
5         //vratit retezec
6     }
7     //vratit retezec
8 }

```

Výpis 6.1: Zjednodušená ukázka metody pro registraci týmu

## Container

Slouží stejně jako v případě komponentu Form jako kontejner pro data. Obsahem jsou například třídy `ContestantImportExcel` a `TeamImportExcel`. Těchto kontejnerů bylo využito při zpracování dat ze souboru ve formátu XLSX. Dále lze v této komponentě nalézt i třídy, kterých bylo využito pro odpověď na požadavek při použití Ajaxu.

## Comparator

Tato komponenta obsahuje třídu `FinalPointsCompare`, která implementuje rozhraní `Comparator<Team>`. Uvnitř této třídy je metoda pro porovnání skóre dvou týmů `compare(Team teamOne, TeamTwo)`, která má návratovou hodnotu typu `int`. Metoda vrací hodnotu -1, jestliže má první argument metody větší skóre než druhý. V opačném případě vrací hodnotu 1. Když se budou porovnávané body obou týmů rovnat, tak vrácená hodnota bude 0. Této metody je v programu využito při řazení spojového seznamu pro výpis výsledků týmů. Uvnitř této komponenty je také třída `SupportedRacesCompare`. Metoda `compare()` této třídy je použita pro seřazení listu kooperací závodu sestupně podle ID.

## Model

Každá třída této komponenty odpovídá právě jedné tabulce v databázi. U tříd je využito anotace `@Table(value = "")`, kde parametr `value` odpovídá názvu entity vytvořené databáze. Tyto třídy mají proměnné, které jsou označeny anotací `@Column`. Hodnota parametru `name` u této anotace odpovídá příslušnému atributu entity databáze. Proměnné, které nemají s atributy entity nic společného jsou označeny anotací `@Transient`. Této anotaci jsem využil pro potřeby vizualizace výsledků u třídy `Team`, kde jsem takto označil proměnné pro výpočet finálních bodů, penalizace a času doběhnutí týmu.

Pro mapování relací mezi jednotlivými entitami je u konkrétních objektů v těchto třídách využito anotace `@JoinColumn` (viz výpis 6.2). Parametr `name` této anotace obsahuje název atributu cizího klíče entity, s kterou je entita v relaci. U objektu třídy je kromě anotace `@JoinColumn` využito i anotace pro popis vztahu mezi entitami (například `@ManyToOne`). Tříd této komponenty je využito při práci s nástrojem Hibernate pro ukládání, upravování, vybírání a mazání záznamů z databáze. Třídy této komponenty obsahují metody pro získání a nastavení hodnot (instancí objektů).

```
1  @Entity
2  @Table(name = "team") //entita team
3  public class Team implements Serializable{
4
5      @Column(name = "name")
6      private String name; //nazev tymu
7
8      @Transient
9      private String resultTime; //neni soucasti databaze
10
11     @ManyToOne
12     @JoinColumn(name = "race_id") //propojeni s tridou
13         Race
14         race;
15
16         //Trida obsahuje metody pro nastaveni a ziskani
17         //hodnot (instanci)
18     }
```

Výpis 6.2: Zjednodušená ukázka třídy `Team`

## Service a DAO

DAO obsahuje třídy, kde jedna třída z této komponenty odpovídá právě jedné třídě z komponenty Model. Pro potřeby práce s třídami této komponenty bylo využito rozhraní frameworku Hibernate ORM, kterými jsou `SessionFactory`, `Criteria`, `Query` a `Session`. Tyto rozhraní nám umožní přidávat, upravovat, mazat a vybírat data z databáze. Funkčnost těch operací zajišťuje anotace u tříd komponenty Model, která popisuje, jakým způsobem se mají data objektu transformovat do databáze. Jednotlivé metody tříd komponenty Dao se liší podle potřeby webové aplikace.

Service slouží jako vrstva, přes kterou se přistupuje k jednotlivým třídám z DAO. Komponenta obsahuje třídy, které mají metody označené anotací `@Transactional`. Tato anotace umožňuje přístup k databázi přes tyto metody označit jako transakci (viz výpis 6.3). Parametry této anotace lze určit, jestli se jedná o transakci, která je určena pro čtení nebo pro zápis. Použití této anotace nám umožní v případě chyby vyvolat navrácení prováděných změn v databázi zpět do původního stavu před zahájením transakce. Bez použití této anotace by jsme museli ručně napsat příkazy pro začátek transakce, potvrzení transakce nebo navrácení změny zpět při odchycení vzniklé chyby [21].

```
1 @Service
2 public class RaceService {
3     @Autowired
4     RaceDAO raceDAO;
5
6     @Transactional
7     public void save(Race race){raceDAO.save(race);}
8 }
9 @Repository
10 public class RaceDAO {
11     @Autowired
12     SessionFactory sessionFactory;
13
14     public void save(Race race){
15         sessionFactory.getCurrentSession().save(race);
16     }
17 }
```

Výpis 6.3: Ukázka uložení závodu do databáze

## 6.1.2 Obsah složky webapp

### Složka Public

Tato složka dále obsahuje podsložky css, images a js.

- Složka css - Obsahuje soubor style.css, ve kterém jsou mnou napsané vlastní kaskádové styly.
- Složka images - Obsahuje dva obrázky background\_home.png a logo.png, které jsou využity v aplikaci. První jmenovaný obrázek je zobrazen na úvodní stránce webu. Druhý je obsahem hlavičky webu sloužící jako logo stránky.
- Složka js - Uvnitř složky je deset JavaScriptových souborů obsahující funkce využití u jednotlivých pohledů stránek pro přidávání a odebrání elementů, editaci textových polí, editaci funkcí u tlačítek nebo pro modifikaci modálních oken. Dále obsahuje funkce pro přenos dat s využitím technologie Ajax, které jsem nejvíce využil při správě výsledků závodu, kde bylo zapotřebí posílat požadavky na server, zpracovat je a vracet odpovědi bez obnovení stránky.

### Složka WEB-INF

Obsahem této složky je základní konfigurační soubor webové aplikace, soubor pro inicializaci kontextu servletu a složka tags. Obsahem složky tags jsou definované základní šablony, u kterých dochází k využití v ostatních šablonách.

Pro potřeby popisu konfiguračního souboru je potřeba definovat Java Servlet. Tento pojem říká, že se jedná o třídu, která rozšiřuje schopnosti serveru a reaguje na příchozí požadavky. Servlet je dále nazýván jako program, který běží na serveru, aby tvořil dynamickou webovou stránku [22].

- web.xml - Jedná se o základní konfigurační soubor webové aplikace, který definuje, jaké servlety budou zavedeny do servlet kontejneru serveru a jaké mapovací vzory k těmto servletům odpovídají. Pro potřebu mé webové aplikace jsem použil výchozí servlet frameworku Spring MVC org.springframework.web.servlet.DispatcherServlet, pro



který byl použit vzor mapování lomítka. Dále tento soubor obsahuje odkaz na inicializační kontext servletu (`mvc-dispatcher-servlet.xml`), nastavení kódování UTF-8 pro webovou aplikaci a definování přesměrování na chybové hlášky odpovídající nevhodnému požadavku klienta na server a další potřebná nastavení.

- `mvc-dispatcher-servlet.xml` - Součástí toho souboru jsou veškeré potřebné konfigurace pro framework Spring MVC. Dále tento soubor definuje přístupové cesty v podobě URL k metodám Controllerů, kde je využito dvou parametrů `access` a `method`. První jmenovaný parametr umožňuje například definovat přístup k metodě typu bez možnosti přístupu, pro přihlášené uživatele do systému, přístup pro všechny nebo pro konkrétní uživatele s přidělenou rolí. Druhý parametr pak umožňuje definovat, jestli mají být přijatá data součástí URL nebo uvnitř HTTP požadavku. Dále tento soubor obsahuje objekty pro databázi, Hibernate, Spring Security a další potřebná nastavení.
- Složka `tags` - Obsahuje šablonu `template.tag` a `race_menu.tag`, které jsou definovány jako vlastní značky. Tyto značky jsou využité v ostatních JSP souborech. První jmenovaná šablona je importována do všech ostatních. Import této šablony do ostatních nám umožní mít hlavičku a základní menu ve všech JSP souborech (viz výpis 6.4). Druhá jmenovaná šablona obsahuje menu závodu, které je využito u pohledů týkajících se závodu.

```
1 <%@taglib prefix="t" tagdir="/WEB-INF/tags" %>
2 <t:template>
3     <jsp:body>
4         //Sem se vklada obsah sablony , ktery bude
5         obalen zakladni sablonou
6     </jsp:body>
7 </t:template>
```

Výpis 6.4: Vložení `template.tag` do ostatních šablon

## Složka `www`

Tato složka obsahuje podsložky, které seskupují jednotlivé JSP soubory podle jejich využití.

- Složka `error` - Obsahuje chybové šablony.
- Složka `others` - Tato složka zahrnuje šablonu, která je použita pro úvodní stránku aplikace. Dále se uvnitř nalézají JSP soubory, které slouží k zobrazení seznamu vyhodnocených závodů a dostupných závodů pro registraci.
- Složka `race` - Obsahuje pohledy týkající se závodu. Těmito pohledy jsou například zobrazení informací o závodě, výsledky závodu, registrace závodníků a další.
- Složka `user` - Uvnitř jsou šablony pro přihlášení, registraci, uživatelskou stránku, zobrazení seznamu závodů uživatele a šablona pro odepření přístupu.

## 6.2 Ukázky implementace

Tato sekce obsahuje ukázky kódů a popisy algoritmů, kterých jsem využil při programování webové aplikace.

### Výpočet bodů týmu

Pro výpočet finálních bodů slouží příkaz (viz výpis 6.5), který je uvnitř třídy `RaceResultController` v metodě `countFinalPoints()`. Tato metoda má jako parametr spojový seznam objektů třídy `Team`. Příkaz sečte body týmu s bonusovými a odečte od nich penalizaci, která se vypočítá uvnitř metody `countPenalizationPoints()`. Tato metoda přímá jako parametr objekt třídy `Team`. V této metodě je využito třídy `Duration` a `DateTime`, které patří do knihovny `Joda-Time`. Při vytváření instance třídy `Duration` jako první parametr je začátek závodu (instance třídy `DateTime`), který přičítá pomocí metody z této třídy čas pro dokončení závodu v minutách. Druhým parametrem je datum a čas (instance třídy `DateTime`) dokončení závodu týmem. U třídy `Duration` je dále využito metody `getStandardMinutes()`, která vypočítá o kolik minut tým přesáhl určený čas pro závod. Každá minuta navíc odpovídá jednomu bodu penalizace. Součástí této metody je proveden výpočet pro zjištění započaté minuty, který pro započatou minutu přičte jedničku k penalizačním bodům. Tato metoda vrací číslo 0 v případě,

že tým dokončil závod v definovaném rozmezí nebo číslo větší než nula, které bude odpovídat bodům penalizace.

```
1 penalization = countPenalizationPoints(team);
2 int finalPoints = team.getPoints() + team.getBonus() -
    penalization;
```

Výpis 6.5: Výpočet finálních bodů týmu

## Ověření zadaného data a času

Při zadávání startu týmu a času dokončení jsou tyto dvě hodnoty zadávány ve formátu `yyyy-MM-dd HH:mm:ss`. U tohoto formátu se může stát, že se člověk jednoduše splete, a proto je potřeba tento formát ověřit. Pro ověření zadaného data a času jsem využil třídy `DateTimeFormat` (viz výpis 6.6) a `DateTimeFormatter`, které jsou obsahem knihovny Joda-Time. První jmenovaná třída má statickou metodu `forPattern()`, která přímá jako parametr řetězec, ve kterém jsem definoval požadovaný formát data. Návrátová hodnota této statické metody přiřadí instanci objektu třídy `DateTimeFormatter`. Tato třída obsahuje metodu `parseDateTime()`, která z přijatého řetězce vytvoří novou instanci objektu třídy `DateTime`, která je také součástí Joda-Time knihovny. Všechny tyto příkazy jsou v `try-catch` bloku pro zachycení nevhodného formátu data.

```
1 DateTime newDate;
2 DateTimeFormatter format = DateTimeFormat.
3     forPattern("yyyy-MM-dd_HH:mm:ss");
4 try {
5     newDate = format.parseDateTime(datetimeTeamForm.
6         getDate());
7     return "wrong_format";
8 }
```

Výpis 6.6: Ověření zadaného data

# 7 Testování

Tato kapitola popisuje testování aplikace na simulaci imaginárního závodu, zátěžový test, zkoušku kompatibility webových prohlížečů a Selenium testy.

## 7.1 Simulace závodu

Testování webové aplikace proběhlo na simulaci imaginárního závodu. Cílem tohoto testování byla snaha pokusit se najít možné nedostatky aplikace a odhalit, jestli je aplikace použitelná pro reálnou situaci. Na tomto testování se podílel vedoucí bakalářské práce Ing. Tomáš Hercig, který má zkušenosti s pořádáním závodů v orientačním běhu.

Webová aplikace byla otestována ručním zadáváním údajů přes formuláře pro registraci týmů. Přes tyto formuláře bylo přidáno 71 týmů, kde každý tým obsahoval právě dva závodníky. Součástí toho testování bylo měření času trvání jednotlivých akcí. Registrace všech týmů přes registrační formuláře trvala jednu hodinu. Zadávání výsledků všech týmů zabralo hodinu a půl. Z toho testování vznikl seznam požadavků, které by bylo vhodné v aplikaci změnit nebo doplnit.

Seznam požadavků plynoucí z testování:

- Výchozí kategorie - Umožnění nastavit v konfiguraci závodu, která kategorie bude při registraci vybrána jako výchozí.
- Seznam týmu - Tento seznam neobsahuje žádné vyhledávání. V Mozilla Firefox tato funkcionality také není k dispozici, protože tento prohlížeč neumožňuje vyhledávat v zablokovaných textových polích.
- Modální okna - Při úspěšné editaci týmu nebo závodníka vyskočí modální okno, které informuje o provedení akce. Neustále zavírání tohoto oznámení je nepříjemné.
- Správa výsledků - Bylo by dobré, aby si aplikace zapamatovala vybrané filtry pro zadávání výsledků. Po obnovení webové stránky zadavatel výsledků o tyto filtry přijde.
- Startovní čas - Bylo by dobré mít možnost uložit startovní čas v textovém poli, aby si ho aplikace pamatovala i po obnovení stránky.

- Zadávání výsledků více lidmi - Při zadávání výsledků více lidmi se nepromítají změny v seznamu ostatních zadavatelů. To znamená, že každý zadavatel má svůj jedinečný seznam, ve kterém do doby než obnoví stránku vidí jen vlastní změny. Při této situaci hrozí, že dojde k přepsání hodnot jiného zadavatele.
- Export výsledků - Pro exportování výsledků není k dispozici filtrování podle kategorií. Jediný způsobem, kterým lze filtrovat řádky s kategoriemi je přes všechny sloupce. Tento způsob není vhodný, protože může dojít k zobrazení neodpovídajících řádků.
- Zadávání hodnot - Po zobrazení modálního okna pro zadávání výsledků se musí kliknout do textového pole. Toto klikání do textových polích zdržuje. Zadávání bodů také nepodporuje záporné hodnoty.
- Stejně body týmů - Při stejných finálních bodech více týmů není zohledněn lepší čas dokončení závodu.

Z výše uvedeného seznamu jsem se rozhodl některé body v programu dodělat (upravit). Při zobrazení modálního okna, které oznamuje změnu údajů týmu nebo závodníka se toto okno po uplynutí krátkého časové okamžiku samo zavře. Pro možnost lepšího filtrování a umožnění exportu dat podle kategorií jsem přidal seznam, který vyfiltruje hodnoty tabulky podle zvolené příslušné kategorie. Při otevírání jednotlivých modálních oken pro efektivnější zadávání výsledků přeskočí kurzor do textového pole a umožní nám hned psát. Upravil jsem také funkci pro výpočet výsledků, která při stejném počtu bodů upřednostní tým, který dokončil závod rychleji. U zadávání výsledků jsem dále umožnil zadávat záporné hodnoty. Seznam týmů jsem změnil do struktury tabulky, která umožňuje lepší zobrazení dat a zároveň řeší problém s vyhledáváním údajů v prohlížeči Mozilla Firefox.

Nejdůležitějším bodem výše uvedeného seznamu je umožnění zadávání výsledků více lidmi najednou a promítání změn mezi ostatní zadavatele. Tento problém jsem vyřešil posláním žádosti o data tabulky s využitím technologie Ajax. Pro kontrolu shody nově získaných dat jsem napsal JavaScriptový kód s využitím knihovny jQuery, který v tabulce údaje aktualizuje, přidá nové a smaže neexistující. Interval posílání žádosti o nová data jsem nastavil na každých 5 sekund. Toto řeší nám zaručí aktuálnost údajů v tabulce. Pro zamezení případné kolize při zadávání výsledků se s nově zadávanými daty posílají i původní, které se kontrolují s hodnotami v databázi.

Simulace závodu na fiktivních datech ukázala, že je aplikace schopná zvládat závod obsahující 140 závodníků. Z toho testování také vyplynulo, že webová aplikace obsahuje všechny základní potřebné funkcionality pro správu závodu. Na nesplněné požadavky lze nahlížet jako na možné budoucí rozšíření aplikace. Nebylo doděláno ukládání filtru při správě výsledků, ukládání globálního startovního času a výběr výchozí kategorie týmů (závodníků) pro zobrazení při registraci týmů (závodníků).

## 7.2 Zátěžový test

Při zátěžovém testu byla aplikace postupně testována na větším množství dat do doby než se ukázala jako nepoužitelnou pro správu závodu. Pro potřeby testování byl vytvořen závod, který umožňoval registrovat do týmu dva závodníky a přiřadit kategorii závodníkovi i týmu. Aplikace byla testována postupným přidáváním týmů, které byly importovány do aplikace z Excelu.

Získané testovací hodnoty:

- 100 týmů - Aplikace fungovala bez problémů. Seznam týmů načítal 403 ms, seznam závodníků 669 ms a zobrazení výsledků 544 ms.
- 200 týmů - Webová aplikace stále nevykazovala žádné známky problému. Seznam týmů načítal 448 ms, seznam závodníků 1,19 sekundy a zobrazení výsledků 611 ms.
- 300 týmů - Přidání sto týmů nevykazovalo změnu chování aplikace. Seznam týmů načítal 580 ms, seznam závodníků 1,6 sekundy a zobrazení výsledků 704 ms.
- 400 týmů - Při tomto počtu byl seznam týmů i zobrazení výsledků v pořádku. Avšak tabulka závodníků už se začínala lehce sekát. Seznam týmů načítal 693 ms, seznam závodníků 2,2 sekundy a zobrazení výsledků 765 ms.
- 500 týmů - Seznam týmů už začal vykazovat známky sekání se při změně kategorie. Stav tabulky závodníků se zase o něco zhoršil. Seznam týmů načítal 808 ms, seznam závodníků 3,2 sekundy a zobrazení výsledků 798 ms.

- 600 týmů - Při tomto počtu už se tabulka týmů i závodníků hůře ovládala, protože docházelo k sekání. Seznam týmů načítal 872 ms, seznam závodníků 4 sekundy a zobrazení výsledků 900 ms.
- 700 týmů - Stav tabulky týmů i závodníků se značně zhoršil. Seznam týmů načítal 1 s, seznam závodníků 4,92 sekundy a zobrazení výsledků 1,12 s.
- 1000 týmů - Seznam týmů i závodníků už dávno ukázal při předchozích hodnotách, že při větším množství týmů není příliš použitelný. Avšak zobrazení výsledků při tomto počtu týmů vykazovalo minimální známky sekání se. Seznam týmů načítal 1,58 s, seznam závodníků 8,32 sekundy a zobrazení výsledků 1,39 s. V tomto místě jsem ukončil zářezový test aplikace.

Test byl proveden na notebooku, kde jeho součástí byl i server s webovou aplikací. Tento notebook měl 2 jádrový procesor i7-5500U s frekvencí 2.4 GHz a 8GB operační paměti. Tento test nám ukázal, že webová aplikace je v pořádku použitelná pro 300 týmů o velikosti dvou závodníků. Jako strop pro použití této aplikace bych zvolil 400 týmů. Při tomto počtu byla aplikace stále plně funkční. Akorát u seznamu závodníků se začaly projevovat menší problémy, které byly stále přijatelné pro chod aplikace.

## 7.3 Kompatibilita prohlížečů

Funkcionality webové aplikace byly vyzkoušeny na pěti základních nejpoužívanějších prohlížečích, kterými jsou Google Chrome, Mozilla Firefox, Internet Explorer, Microsoft Edge a Opera. Při zkoušce prohlížečů nebyla nalezená žádná chyba až na jednu malou při vyhledávání u Firefoxu v zablokovaných textových polích. Tento problém byl opraven. Scénář zkoušky kompatibility prohlížečů byl následující:

- Registrace uživatelů
- Vytvoření dvou závodů s rozdílnou velikostí týmů
- Registrace týmů (závodníků) přes všechny typy formulářů
- Editace záznamů týmů (závodníků) včetně exportu a mazání dat

- Správa výsledků (Zadání hodnot, řazení, filtrování, export záznamů)
- Přidání správce do závodu a testování z jeho pohledu
- Procházení záznamů z pohledu nepřihlášeného a přihlášeného uživatele

Verze zkoušených prohlížečů:

- Google Chrome 57.0.2987.133
- Mozilla Firefox 47.0.2
- Internet Explorer 11.1066.14393.0
- Microsoft Edge 38.14393.1066.0
- Opera 43.0.2442.1144

## 7.4 Selenium testy

Webová aplikace byla otestována v prohlížeči Mozilla Firefox použitím pluginu Selenium IDE (verze 2.9.1). Přes tento doplněk bylo vytvořeno 15 testů, při jejichž psaní bylo dbáno na to, aby bylo pokryto co nejvíce možných variant vstupů u formulářů. Vzhledem k tomu, že je většina formulářů vytvářena dynamicky, tak bylo dosti obtížné vyzkoušet všechny varianty. Z tohoto důvodu jsem se soustředil především na formuláře, které šlo testovat v cyklu bez nutnosti přepsání parametrů testu.

Na obrázku 7.1 je ukázka Selenium testu pro registraci uživatele. V tomto testu bylo pomocí JavaScriptu generováno uživatelské jméno, příjmení, křestní jméno, telefonní číslo a email. U tohoto testu jsem se především soustředil na výpis chybových hlášek pro už existující přihlašovací jméno a email. Do JavaScriptu pro generování hodnot byla vložena podmínka, která ověřuje, jestli je náhodně vygenerované číslo v rozmezí 0-3 rovno nule. Jestliže je rovno nule, tak do formuláře bude vložen prázdný řetězec. U tohoto formuláře musí být každý atribut vyplněn až na telefonní číslo. Pro nevyplněné požadované atributy je očekáváno chybové oznámení. U tohoto testu jsem se nesoustředil na délku zadaných hodnot a ani na shodu hesel uživatele. Testování těchto případů proběhlo v dalších dvou testech, které u tohoto formuláře byly provedeny.



user_registration3		
clickAndWait	link=Registration	
type	id=user.login	javascript{if(Math.floor(Math.random()*4) ==0) {""} else {"test"+Math.floor(Math.random()*111)} ;}
type	id=user.firstname	javascript{if(Math.floor(Math.random()*4) ==0) {""} else {"test"+Math.floor(Math.random()*111)} ;}
type	id=user.surname	javascript{if(Math.floor(Math.random()*4) ==0 ) {""} else {"test"+Math.floor(Math.random()*111)} ;}
type	id=user.phone	javascript{if(Math.floor(Math.random()*4) ==0) {""} else {Math.floor(Math.random()*999999999+111111111)} ;}
type	id=user.email	javascript{if(Math.floor(Math.random()*4) ==0) {""} else {"test" + Math.floor(Math.random()*111) + "@seznam.cz"} ;}
type	id=user.password	123456789
type	id=passwordAgain	123456789
clickAndWait	name=submit	

Obrázek 7.1: Selenium test - registrace uživatele

## 8 Závěr

V rámci bakalářské práce byla vytvořena webová aplikace pro pořádání závodů v orientačním běhu, která umožní registrovat uživatele do systému, vytvářet závody včetně jejich správy a registrace týmů (závodníků) do závodu správci i běžnými uživateli. Další úkolem bylo umožnění spravovat seznam týmů (závodníků) včetně úpravy i mazání údajů. Seznam závodníků měl navíc umožňovat import i export týmů ve formátu XLSX. Posledním úkolem bylo zobrazení výsledků včetně jejich správy umožňující vyhledávat v těchto datech, řadit jednotlivé sloupce sestupně i vzestupně, filtrovat podle kategorií a exportovat data ve formátu XLSX a CSV.

V první části práce byla provedena rešerše webových frameworků v jazyce Java, jejíž součástí bylo i vybrání jednoho z nich pro implementaci. Nakonec byl vybrán Spring. Tomuto frameworku byla věnována samostatná kapitola, ve které byly popsány důvody výběru Springu, jeho dostupné moduly a krátký popis historie.

Při návrhu aplikace byly popsány požadavky na aplikaci, které byly rozděleny pro potřebu aplikace na funkční a mimofunkční. Součástí této části byl vytvořen ERA diagram včetně popisu atributů a relací všech tabulek. Pro potřeby aplikace byli rozděleni uživatelé do tříd. Jednotlivé funkcionality webu pak byly v návrhu popsány z pohledu přístupu těchto tříd uživatelů.

V implementaci byla popsána struktura projektu. Tento popis byl především soustředěn na komponent Controller, u kterého byly popsány k čemu slouží jednotlivé třídy. Další komponenty už byly popsány obecněji. Součástí implementace byly také popsány konfigurační soubory, rozdělení šablon do složek podle využití a další adresáře obsahující kaskádové styly, JavaScriptové soubory nebo obrázky.

Aplikace byla otestována na simulačním závodu. Většina požadavků na vylepšení z toho testování byla dodělána. Zátěžový test ukázal, že je aplikace schopná zvládnout v pořádku až 300 týmů při definování počtu členů v týmu dva. Součástí testování byla i zkouška prohlížečů a provedení Selenium testů.

Na závěr byla vytvořena uživatelská dokumentace. Součástí této dokumentace byly popsány jednotlivé kroky od registrace uživatele až po správu výsledků týmů. Touto poslední částí byly splněny všechny body zadání včetně požadavků na vytvoření aplikace.

# Literatura

- [1] *Struts 2 Overview* [online]. Tutorial Point, 2015. [cit. 2016/11/25].  
Dostupné z: [https://www.tutorialspoint.com/struts\\_2/struts\\_overview.htm](https://www.tutorialspoint.com/struts_2/struts_overview.htm).
- [2] *What is JSF?* [online]. Tutorial Point, 2017. [cit. 2017/06/09]. Dostupné z: [https://www.tutorialspoint.com/jsf/jsf\\_overview.htm](https://www.tutorialspoint.com/jsf/jsf_overview.htm).
- [3] *Apache Wicket Index* [online]. Apache Software Foundation, 2016. [cit. 2016/11/27]. Dostupné z: <https://cwiki.apache.org/confluence/display/WICKET/Index>.
- [4] *Comparing Wicket and Struts* [online]. Apache Software Foundation, 2006. [cit. 2016/11/27]. Dostupné z: <https://cwiki.apache.org/confluence/display/WICKET/Struts>.
- [5] *GWT - Overview* [online]. Tutorial Point, 2015. [cit. 2016/11/28].  
Dostupné z: [https://www.tutorialspoint.com/gwt/gwt\\_overview.htm](https://www.tutorialspoint.com/gwt/gwt_overview.htm).
- [6] *Introduction* [online]. Apache Software Foundation, . [cit. 2016/11/29].  
Dostupné z: <http://tapestry.apache.org/introduction.html>.
- [7] *Apache Tapestry* [online]. Apache Software Foundation, . [cit. 2016/11/29].  
Dostupné z: <http://tapestry.apache.org/>.
- [8] *Appendix A: Architecture & philosophy* [online]. [cit. 2016/12/05].  
Dostupné z: [https://openjava.wikispaces.com/philosophy\\_en](https://openjava.wikispaces.com/philosophy_en).
- [9] *Welcome to RIFE* [online]. RIFE tým. [cit. 2016/12/06]. Dostupné z: <http://www.rifers.org>.
- [10] *Ten Good Reasons To Use Seam* [online]. Red Hat. [cit. 2016/12/08].  
Dostupné z: <http://seamframework.org/Home/WhySeam.html>.
- [11] *Understanding Enterprise JavaBeans* [online]. Oracle, 2008. [cit. 2017/06/25]. Dostupné z: [https://docs.oracle.com/cd/E13222\\_01/wls/docs100/ejb/understanding.html#wp1141710](https://docs.oracle.com/cd/E13222_01/wls/docs100/ejb/understanding.html#wp1141710).
- [12] JOHNSON, R. *Spring Framework: The Origins of a Project and a Name* [online]. Pivotal Software, 2006. [cit. 2016/11/15]. Dostupné z: <https://spring.io/blog/2006/11/09/spring-framework-the-origins-of-a-project-and-a-name>.

- [13] *Part I. Overview of Spring Framework* [online]. Pivotal Software, Inc, 2016. [cit. 2016/11/12]. Dostupné z: <http://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/overview.html>.
- [14] SHANNON, R. *What is HTML* [online]. 2012. [cit. 2016/04/15]. Dostupné z: <http://www.yourhtmlsource.com/starthere/whatishtml.html>.
- [15] BOS, H. W. L. . B. *Cascading Style Sheets: Designing for the Web (3rd Edition)*. Addison-Wesley Professional, 2005. ISBN 0-321-19312-1.
- [16] *Bootstrap Get Started* [online]. Refsnes Data. [cit. 2016/04/05]. Dostupné z: [https://www.w3schools.com/bootstrap/bootstrap\\_get\\_started.asp](https://www.w3schools.com/bootstrap/bootstrap_get_started.asp).
- [17] FLANAGAN, D. *JavaScript: The Definitive Guide, 5th Edition*. O'Reilly Media, 2006. ISBN 0-596-10199-6.
- [18] JONATHAN CHAFFER, K. S. *Mistrovství v jQuery*. Computer Press, 2013. ISBN 978-80-251-4103-8.
- [19] CHRISTIAN BAUER, G. K. *Hibernate in action*. Manning Greenwich CT, 2005. ISBN 1932394-15-X.
- [20] *Reference* [online]. SpryMedia. [cit. 2016/04/06]. Dostupné z: <https://datatables.net/reference/>.
- [21] *Part V. Data Access* [online]. Pivotal Software, 2016. [cit. 2016/04/08]. Dostupné z: <http://docs.spring.io/spring-framework/docs/4.3.x/spring-framework-reference/html/transaction.html>.
- [22] *Servlet Tutorial* [online]. SSS IT. [cit. 2016/04/20]. Dostupné z: <https://www.javatpoint.com/servlet-tutorial>.

# Zkratky

AJAX - Asynchronous JavaScript and XML

AOP - Aspect Oriented Programming

API - Application Programming Interface

CSS - Cascading Style Sheets

DAO - Data Access Objekt

DOM - Document Object Model

EJB - Enterprise Java Beans

GWT - Google Web Toolkit

HQL - Hibernate Query Language

HTML - HyperText Markup Language

IoC - Inversion of Control

J2EE - Java 2 Enterprise Edition

JDBC - Java Database Connectivity

JMS - Java Messaging Service

JPA - Java Persistence API

JSF - JavaServer Faces

JSON - JavaScript Object Notation

JSP - Java Server Pages

JVM - Java Virtual Machine

MDD - Model-Driven Development

MDL - Material Design Lite

MVC - Model-View-Controller

ORM - Object-Relational Mapping

OXM - Object/XML Marshalling

POJO - Plain Old Java Object

RIA - Rich Internet Application

SpEL - Spring Expression Language

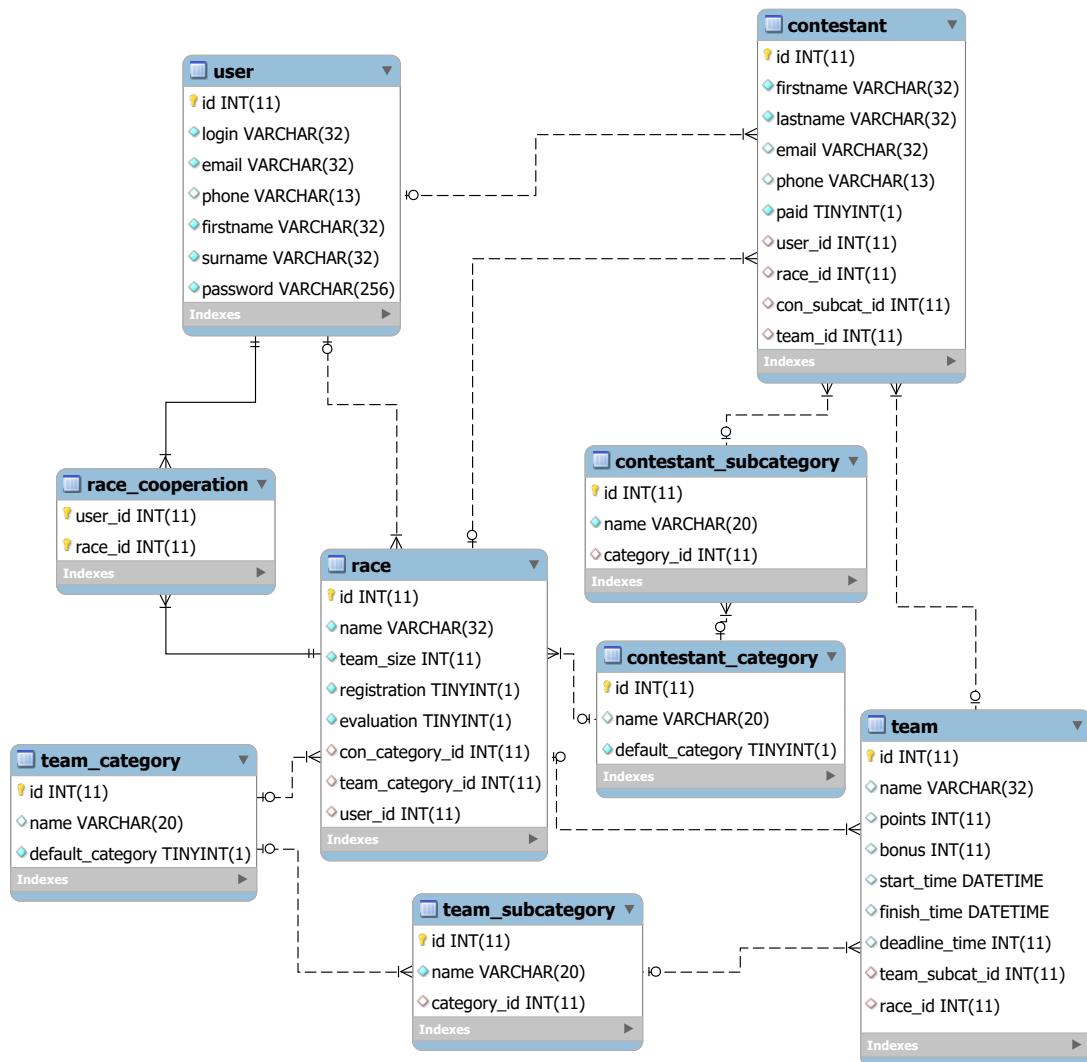
SQL - Structured Query Language

UI - User interface

WAR - Web Application ARchive

# Přílohy

## A Databáze



Obrázek 1: ERA diagram databáze

## B Dokumentace

### Registrace uživatele

Při registraci (obrázek 2) se musí zadat textová pole označená červenou hvězdičkou. Počet zadaných znaků musí vždy odpovídat rozsahu v závorce u názvu textového pole. Pro odeslání formuláře se musí kliknout na modré tlačítko s nápisem Registration.

- Username - unikátní uživatelské jméno (3 - 32 znaků)
- Firstname - křesní jméno uživatele (3 - 32 znaků)
- Surname - příjmení uživatele (3 - 32 znaků)
- Phone - telefonní číslo ve formátu 123456789 nebo +420123456789
- Email - unikátní email (6 - 32 znaků)
- Password - heslo (8 - 256 znaků)
- Password again - zadání stejného hesla znovu

Username\* (3 - 32 length):

Firstname\* (3 - 32 length):

Surname\* (3 - 32 length):

Phone (123456789, +420123456789):

Email\* (6 - 32 length):

Password\* (8 - 256 length):

Password again\* (8 - 256 length):

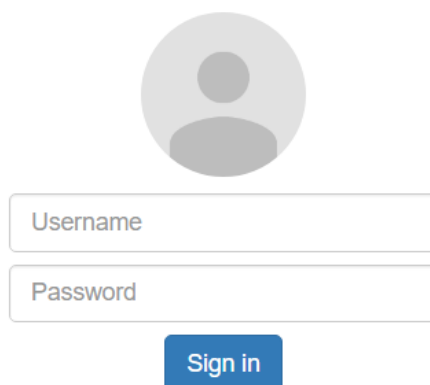
Registration

Obrázek 2: Registrace uživatele



## Přihlášení uživatele

Pro přihlášení uživatele (obrázek 3) se musí zadat uživatelské jméno do pole s názvem Username a heslo do pole s názvem Password.

The image shows a user login interface. At the top center is a circular placeholder for a user profile picture, rendered in shades of gray. Below this are two stacked text input fields. The top field is labeled 'Username' and the bottom field is labeled 'Password'. Both fields have a light gray border and a subtle shadow. Below the password field is a blue rectangular button with the text 'Sign in' in white, centered on the button.

Obrázek 3: Přihlášení uživatele

## Vytvoření závodu

Při vytváření závodu (obrázek 4) se musí zadat název závodu s počtem znaků 3 až 32 a definovat počet závodníků v týmu. Tento počet se vybírá z předpřipraveného seznamu, kde lze určit velikost týmů v rozmezí 1 až 5 závodníků. Dále se musí zvolit kategorie pro závodníky i týmy. Při výběru kategorií závodníků jsou k dispozici tři přepínače. První přepínač je None. Při výběru tohoto přepínače nebude definována žádná kategorie pro závodníky. Druhým přepínačem je Use default category. Při výběru tohoto přepínače musíme zvolit kategorii z předpřipraveného listu. Poslední volbou je přepínač Create own categories. Při tomto výběru můžeme definovat 2 až 20 vlastních kategorií. Počet textových polí pro definování kategorií se přidává zeleným tlačítkem plus a odebírá červeným tlačítkem mínus. Definování týmových kategorií probíhá stejným způsobem. Po vyplnění všech potřebných údajů se musí kliknout na modré tlačítko Create race. Po úspěšném vytvoření závodu dojde k přesměrování na jeho nastavení.

Race name\* (3 - 32 length):

Size of team\*:

Define contestant categories

None

Use default category

Create own categories

Category 1 (1 - 20 length):

Category 2 (1 - 20 length):

Obrázek 4: Vytvoření závodu

## Úvodní stránka závodu

Tato stránka je rozdělena do čtyř základních sekcí. První sekce nazvána About the race (obrázek 5) obsahuje název závodu a výchozí velikost týmů. Tato sekce je dostupná pro všechny uživatele. Druhou sekcí je Race settings (obrázek 6). Tato sekce obsahuje dvě tlačítka. Horním tlačítkem lze zablokovat přihlašování závodníků do závodu. Při hodnotě Enabled je registrace uvolněna a při hodnotě Disabled je zablokována. Spodní tlačítko označuje závod jako vyhodnocený. Při hodnotě Evaluated je závod vyhodnocený a při hodnotě Not evaluated není. Měnit nastavení závodu smí pouze jeho vlastník.

ABOUT THE RACE

Race name:

Size of team:

Obrázek 5: Informace o závodě

RACE SETTINGS

Registration	Enabled
<hr/>	
Results	Not evaluated

Obrázek 6: Nastavení závodu

Třetí sekce slouží pro přidávání správců závodu (obrázek 7). Pro přidání správce závodu se musí vložit jeho uživatelské jméno do pole nazvané Copoperator login. Pro potvrzení toho formuláře se musí zmáčknout na modré tlačítko Add copoperator. Dále tato sekce obsahuje seznam správců, u kterého lze příslušného správce odebrat tlačítkem vedle jeho jména. Tato sekce je dostupná jen pro vlastníka závodu.

Poslední sekce umožňuje majiteli smazat jeho závod (obrázek 8). Pro smazání závodu musí majitel zadat svoje přihlašovací hesla a zmáčknout červené tlačítko Delete race.

The image shows two parts of a web interface. The top part is a light gray box with the text 'ADD RACE COOPERATOR' centered. Below it, the text 'Cooperator login:' is followed by a white input field. To the right of the input field is a blue button with the text 'Add cooperator'. The bottom part is another light gray box with the text 'LIST OF COOPERATORS' centered. Below it, the text 'TestUserA' is followed by a red button with the text 'Delete cooperator'.

Obrázek 7: Seznam správců závodu a jejich přidávání

The image shows a light gray box with the text 'DELETE THE RACE' centered. Below it, the text 'Password:' is followed by a white input field. To the right of the input field is a red button with the text 'Delete race'.

Obrázek 8: Smazání závodu

## Registrace do závodu

Registrace je rozdělena do dvou základních sekcí. První sekce slouží pro potřeby správců závodu. V této sekci je formulář pro zadání týmu nazvaný Admin team registration (obrázek 9). Tento formulář obsahuje název týmu, který se nemusí vyplnit. Při jeho vyplnění musí být název v rozsahu 3 až 32 znaků. Dále se tu definuje kategorie týmu ze seznamu kategorií přiřazených pro tento závod. Poté se definují jednotliví závodníci týmu. Počet závodníků oproti výchozí hodnotě lze zvětšit až o dva závodníky navíc. Pro tyto potřeby je přidán zelené tlačítko plus, které zobrazí pole pro zadání dalšího závodníka a červené tlačítko mínus, které naopak ubere závodníka.

U všech závodníků týmu musí být zadané křestní jméno a příjmení s počtem znaků 3 až 32. V případě definovaných kategorií pro závodníky se musí vybrat kategorie ze seznamu. U závodníků lze dále zadat email, telefon a zaškrtnout, jestli zaplatil. Pro umožnění registrace závodníka bez týmu je tu formulář nazvaný Admin solo registration (obrázek 10). Při registraci závodníka se definují stejné údaje jako u předchozího zmíněného formuláře. U toho formuláře oproti předchozímu nelze definovat název týmu, vybrat týmovou kategorii a ani nejde rozšířit počet závodníků.

**ADMIN TEAM REGISTRATION**

Team name (3 - 32 length): <input type="text"/>	Team category*: <input type="text" value="A"/>
----------------------------------------------------	---------------------------------------------------

Teammate 1:

Firstname* (3 - 32 length): <input type="text"/>	Lastname* (3 - 32 length): <input type="text"/>	Phone (123456789, +420123456789): <input type="text"/>
Email (6 - 32 length): <input type="text"/>	Contestant category*: <input type="text" value="Male"/>	<input type="checkbox"/> Paid

---

Teammate 2:

Firstname* (3 - 32 length): <input type="text"/>	Lastname* (3 - 32 length): <input type="text"/>	Phone (123456789, +420123456789): <input type="text"/>
Email (6 - 32 length): <input type="text"/>	Contestant category*: <input type="text" value="Male"/>	<input type="checkbox"/> Paid

Obrázek 9: Registrace týmu správcem

ADMIN SOLO REGISTRATION

Firstname* (3 - 32 length): <input style="width: 95%;" type="text"/>	Lastname* (3 - 32 length): <input style="width: 95%;" type="text"/>	Phone (123456789, +420123456789): <input style="width: 95%;" type="text"/>
Email (6 - 32 length): <input style="width: 95%;" type="text"/>	Contestant category*: <div style="border: 1px solid #ccc; padding: 2px; display: flex; align-items: center;"><span style="margin-right: 5px;">Male</span><span style="font-size: 0.8em;">▼</span></div>	<input type="checkbox"/> Paid

[Solo registration](#)

Obrázek 10: Registrace závodníka bez týmu správcem

Druhá sekce je pro přihlášené uživatele. Tato sekce obsahuje také dva formuláře. U Solo registration formuláře (obrázek 11) uživatel vybírá pouze kategorii závodníka. Zbytek potřebných údajů je vzato z jeho účtu. Druhý formulář (obrázek 12) odpovídá formuláři Admin team registration. Jedinou změnou oproti tomuto formuláři je absence zaškrtačacího tlačítka platby a nevyplňují se údaje prvního závodníka kromě jeho kategorie. Údaje o prvním závodníkovi budou vzaty z jeho účtu.

SOLO REGISTRATION

Contestant category\*:

Male▼

Informations about you will be take from your account.

[Solo registration](#)

Obrázek 11: Registrace závodníka bez týmu

TEAM REGISTRATION

Team name (3 - 32 length):

Team category\*:

You:

Contestant category\*:

Informations about you will be take from your account.

Teammate 1:

Firstname* (3 - 32 length): <input style="width: 100%;" type="text"/>	Lastname* (3 - 32 length): <input style="width: 100%;" type="text"/>	Phone (123456789, +420123456789): <input style="width: 100%;" type="text"/>
Email(6 - 32 length): <input style="width: 100%;" type="text"/>	Contestant category*: <input style="width: 100%;" type="text" value="Male"/>	

Obrázek 12: Registrace týmu

## Sestavení týmů

Při sestavování týmů ze závodníků (obrázek 13), kteří se registrovali bez týmu se vybírají členové kliknutím na zelené tlačítko plus vedle jejich údajů. Červené tlačítko mínus pak slouží pro odznačení z výběru. Při vytváření týmů lze definovat název týmu o velikost 3 až 32 znak včetně výběru týmové kategorie. Po vyplnění příslušných hodnot a zvolení členů se formulář odešle zmáčknutím na modré tlačítko Registration. Pro smazání závodníků slouží červené tlačítko Delete vedle jejich jména. Sestavování týmů je k dispozici jen pro správce a vlastníka závodu.

Team name:  Team category:

	Firstname	Lastname	Email	Phone	Category	Delete
<input type="button" value="+"/>	Jan	Palcút			Male	<input type="button" value="Delete"/>
<input type="button" value="+"/>	Tomáš	Novák			Male	<input type="button" value="Delete"/>
<input type="button" value="-"/>	Karel	Nový	test@test.cz		Male	<input type="button" value="Delete"/>

Obrázek 13: Sestavení týmů ze závodníků bez týmu

## Seznam týmů

Pro běžné uživatele tento seznam zobrazuje jen název týmu včetně jejich členů a podkategorii týmu (obrázek 14). Další funkcionality už jsou jen pro správce závodu a vlastníka závodu. Součástí seznamu je navíc zobrazeno ID týmu a umožněno editovat název i podkategorii týmu. Dále je přidáno tlačítko Delete pro smazání týmu. Po zmáčknutí tlačítka se zobrazí modální okno (obrázek 15), ve kterém lze vybrat možnost smazání týmu včetně závodníků nebo bez nich. Nad seznamem týmů je tlačítko Export teams, které vyexportuje týmy ve formátu XLSX. Vedle toho tlačítka je tlačítko Import teams, na které když zmáčkne, tak se otevře modální okno pro importování týmů ve formátu XLSX (obrázek 16). Po zmáčknutí na tlačítko Import XLSX file vybereme soubor pro importování a potvrdíme tuto akci modrým tlačítkem.



Import teams Export teams

ID	Team	Members	Team category	Save	Delete
278	TEAM A	Evžen Novotný, Roman Novák	A ▾	Save	Delete
280	Tým roku	Zdeněk Svátý, Michal Běžný	A ▾	Save	Delete
281	<input type="text"/>	Michal Novotný, Kamil Střecha	A ▾	Save	Delete

Obrázek 14: Seznam týmů

Delete team ×

---

Would you like to delete the team?

---

DELETE TEAM

DELETE WITH CONTESTANTS

Close

Obrázek 15: Smazání týmů

Import teams from EXCEL ×

---

Import XLSX file

---

Import
Close

Obrázek 16: Import týmů z Excelu

## Seznamy závodů

K dispozici jsou dva základní seznamy závodů. První obsahuje závody, které jsou k dispozici k registraci (obrázek 17). Po zmáčknutí na název závodu

budeme přesměrováni na základní informace o závodě. Když zmáčkneme na tlačítko Registration, tak budeme přesměrováni na registrační formulář. U každého závodu je napsáno, jestli je momentálně registrace povolena nebo zakázána.

Race name	Registration
test456	Enabled <a href="#">Registration</a>
TeamSizeOne	Enabled <a href="#">Registration</a>

Obrázek 17: Seznam závodů pro registraci

Druhý seznam (obrázek 18) obsahuje závody, které byly označeny jako vyhodnocené vlastníkem závodu. Při kliknutí na název závodu budeme přesměrováni na informace o závodě. Když klikneme na tlačítko Results, tak budeme přesměrováni na výsledky závodu.

Race name
Test aplikace 2 <a href="#">Results</a>
Test aplikace <a href="#">Results</a>

Obrázek 18: Vyhodnocený seznam závodů

## Správa seznamu závodníků

Tento seznam (obrázek 19) je určený pro správce a vlastníka závodů. Každá řádka obsahuje ID týmu, ke kterému je závodník přiřazen. Dále je součástí každého záznamu křestní jméno, příjmení a podkategorie. Tyto údaje lze po přepsání změnit zmáčknutím na tlačítko Save. Pro změnu atributu platil je k dispozici tlačítko buď zelené barvy (platil) nebo červené barvy (neplatil).

Team	Firstname	Lastname	Category	Paid	Save
278	Evžen	Novotný	Male ▼	YES	Save
278	Roman	Novák	Male ▼	NO	Save
281	Michal	Novotný	Male ▼	YES	Save

Obrázek 19: Seznam závodníků bez telefonních čísel a emailů

## Zobrazení výsledků

Zobrazení výsledků (obrázek 20) je dostupné pro všechny uživatele. Součástí každého řádku je název týmů včetně jeho členů, umístění, podkategorie týmu, body, bonusové body, finální body, penalizace, startovní čas, finální čas a délka běhu závodu. Údaje lze filtrovat podle kategorií nebo použít filtrování přes všechny sloupce a řádky. Jednotlivé sloupce lze řadit sestupně a vzestupně. Pro správce a vlastníky závodu je zde navíc zobrazeno ID týmu a umožněno exportovat aktuálně vyfiltrovaná data ve formátu XLSX a CSV.

Rank ▲	Team	Id	Team category	Points	Bonus	Penalization	Final Points	Start time	Finish time	Result time
1	TEAM A (Evžen Novotný, Roman Novák)	278	A	55	10	0	65	2017-05-02 15:26:30	2017-05-02 16:26:30	01:00:00
2	Tým roku (Zdeněk Svatý, Michal Běžný)	280	A	40	0	0	40	2017-05-02 15:26:30	2017-05-02 16:15:30	00:49:00
3	Michal Novotný, Kamil Střecha	281	A	10	0	0	10	2017-05-02 15:26:30	2017-05-02 17:26:30	02:00:00

Obrázek 20: Zobrazení výsledků

## Správa výsledků

Správa výsledků (obrázek 21) je dostupná jen pro správce a vlastníky závodu. Pro zadávání startovního času více týmům je určeno textové pole vedle tlačítka Set start time. Po zadání startovního času a zmáčknutí tohoto tlačítka se otevře modální okno (obrázek 22). V tomto okně lze přiřadit startovní čas všem týmům, dalším deseti nebo týmům podle vybrané kategorie.

Id	Team	Team category	Points	Bonus	Start time	Finish time	Deadline time	Action
278	TEAM A (Evžen Novotný, Roman Novák)	A	55	10	2017-05-02 15:26:30	2017-05-02 16:26:30	120	Finish
280	Tým roku (Zdeněk Svatý, Michal Běžný)	A	40	0	2017-05-02 15:26:30	2017-05-02 16:15:30	120	Finish

Obrázek 21: Správa výsledků

Set start time

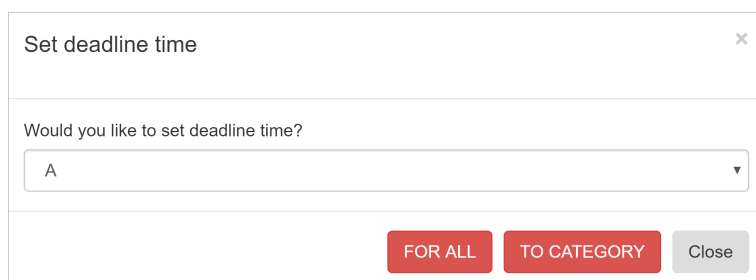
Would you like to set a time for?

A

FOR ALL Next 10 TO CATEGORY Close

Obrázek 22: Nastavení startovního času pro více týmů

Vedle tlačítka Deadline time je nalevo textové pole pro zadání času určeného pro dokončení závodu v minutách. Po zadání hodnoty a zmáčknutí tohoto tlačítka se otevře modální okno (obrázek 23), ve kterém lze zvolit přiřazení času pro dokončení závodu všem týmům nebo podle vybrané kategorie.



Obrázek 23: Nastavení času pro dokončení závodu

Při zadávání bodů týmu se musí na příslušné řádce kliknout do sloupce s body. Po tomto kliknutí se zobrazí modální okno (obrázek 24), ve kterém lze přiřadit body. Zadávání bodů se neliší od zadávání bonusových bodů. Vždy se otevře stejné okno s dvěma textovými poli pro zadání obou typů bodů.



Obrázek 24: Nastavení bodů týmu

Při zadávání startovního a finálního času se vždy musí kliknout v řádku na příslušnou pozici. U zadávání startovního času se zobrazí po kliknutí modální okno (obrázek 25), kde lze zadat startovní čas a potvrdit ho tlačítkem Set time. K dispozici je i druhé tlačítko, které se jmenuje Set global time. Po kliknutí na toto tlačítko se nastaví jako startovní čas týmu hodnota z textového pole ležící nalevo od tlačítka Set start time na obrázku 21. Při zadávání finálního času do modálního okna se zadání hodnoty potvrdí tlačítkem Set (obrázek 26).

Obrázek 25: Nastavení startovního času

Obrázek 26: Nastavení času dokončení závodu

## Import ve formátu XLSX

Součástí seznamu týmů je možnost importování týmů ve formátu XLSX. Na obrázku 27 je vidět struktura tabulky v Excelu pro importování týmů o definované výchozí velikosti týmu jedna. Tato struktura je neměnná a musí se dodržovat. List v tomto souboru musí mít název Contestants.

	A	B	C	D	E	F	G
1	FIRSTNAME	LASTNAME	EMAIL	PHONE	PAID	CON_CATEGORY	TEAM_CATEGORY
2	Jan	Palcút			YES	A	FIRST
3	Jakub	Palcút			YES	B	SECOND
4	Tomáš	Palcút			YES	A	FIRST

Obrázek 27: Struktury tabulky pro importování jednočlenného týmu

Při definované výchozí velikosti týmu větší než jedna je struktura složitější. Ukázka je vidět na obrázku 28. Struktura v Excelu se bude lišit právě podle definované velikosti týmů, která v tomto případě musí být větší než jedna. Atributy začínající číslem pak představují index konkrétního závodníka týmu. Závodníci v týmu se indexují od 0 až do definované velikosti týmu mínus jedna. Pro definovanou výchozí velikost týmu dva bude podle

obrázku 28 nutné přidat atributy s indexem jedna, které budou hned za atributy s indexem nula. Pro případ importování závodníka bez týmu musí být vepsána hodnota YES do atributu SOLO a musejí být vyplněné jen údaje závodníka s indexem nula. Seznam týmů umožňuje exportovat předpřipravenou strukturu ve formátu XLSX. List v tomto souboru musí mít název Teams.

	A	B	C	D	E	F	G	H	I
1	SOLO	TEAM	TEAM_CATEGORY	OFIRSTNAME	OLASTNAME	OEMAIL	OPHONE	OPAID	OCATEGORY
2	NO	AAA	Smrtelník	Strom	sstromovič			YES	FAV
3	NO	BBB	Smrtelník	Jan	Palcút			NO	FAV
4	YES			Jan	Palcút			NO	FAV

Obrázek 28: Struktura tabulky v Excelu pro importování týmů

## C Obsah DVD

Součástí práce je DVD, které obsahuje:

- `readme.txt` - Textový soubor, který popisuje kroky pro zprovoznění webové aplikace.
- `bptext` - Složka, která obsahuje zdrojové soubory práce psané v  $\text{\LaTeX}$ u včetně obrázků ve složce `img`.
- `bp.pdf` - Soubor ve formátu PDF, který obsahuje text práce.
- `selenium` - Složka, která obsahuje Selenium testy.
- `webapp` - Tento adresář obsahuje složku `src`, ve které jsou zdrojové soubory aplikace. Uvnitř složky `target` je WAR soubor s názvem `orienteering_races`. Složka `webapp` také obsahuje soubor `pom.xml` pro vytvoření WAR souboru nástroj Apache Maven.
- `database.sql` - Soubor, který slouží pro vytvoření databáze.