

# Časoměrný systém pro nízkorozpočtové sportovní akce

## Bakalářská práce

*Studijní program:*

B2646 Informační technologie

*Studijní obor:*

Informační technologie

*Autor práce:*

**Lukáš Kořínek**

*Vedoucí práce:*

Ing. Jana Kolaja Ehlerová, Ph.D.

Ústav nových technologií a aplikované informatiky





## Zadání bakalářské práce

# Časoměrný systém pro nízkorozpočtové sportovní akce

*Jméno a příjmení:* **Lukáš Kořínek**  
*Osobní číslo:* M18000214  
*Studijní program:* B2646 Informační technologie  
*Studijní obor:* Informační technologie  
*Zadávací katedra:* Ústav nových technologií a aplikované informatiky  
*Akademický rok:* **2020/2021**

### Zásady pro vypracování:

1. Proveďte rešerši běžně dostupných časoměrných technik pro běžecké a cyklistické závody.
2. Navrhněte a implementujte aplikaci pro mobilní zařízení, která v sobě obsáhne inteligentní stopky, možnost importovat startovní listinu a exportovat výsledky.
3. Otestujte aplikaci v ostrém provozu.
4. Zamyslete se nad návrhy pro zlepšení uživatelského prožitku, zverifikujte a zkalibrujte výsledky vůči nezávislému měření.

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

dle potřeby dokumentace  
30 – 40 stran  
tištěná/elektronická  
Čeština



### **Seznam odborné literatury:**

- [1] Dokumenty IAAF k metodice měření, online text, <https://www.iaaf.org/about-iaaf/documents>.
- [2] Křivda, Martin: Analýza možností automatického měření běžeckých závodů, Praha, 2019, Bakalářská práce, Vysoká škola ekonomická, Katedra informačního a znalostního inženýrství.
- [3] GARNER, 2017, PikaTimer: An OpenSource Race Timing Application [online]. PikaTimer. Dostupné z: <https://github.com/PikaTimer/pikatimer>.

*Vedoucí práce:*

Ing. Jana Kolaja Ehlerová, Ph.D.  
Ústav nových technologií a aplikované informatiky

*Datum zadání práce:*

19. října 2020

*Předpokládaný termín odevzdání:*

17. května 2021

prof. Ing. Zdeněk Plíva, Ph.D.  
děkan

L.S.

Ing. Josef Novák, Ph.D.  
vedoucí ústavu

V Liberci dne 19. října 2020

## Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

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

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

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

16. května 2021

Lukáš Kořínek

# Abstrakt

Tato bakalářská práce se zabývá problematikou měření sportovních akcí, jako jsou například běžecké nebo cyklistické závody. V první části práce jsou popsány dostupné metody měření těchto závodů a následně přiblíženy některé služby, které jsou organizátorům schopné změřit celý závod.

Praktickou částí této práce je návrh a implementace mobilní aplikace pro operační systém Android, pomocí které lze měřit závody. Druhá část práce se tedy zabývá návrhem této aplikace. Pozornost je věnována požadavkům na funkcionalitu, možnostem využití, popisu aplikace, popisu použitých technologií a architektuře aplikace. Ve třetí části se práce zabývá samotnou implementací, jsou přiblíženy jednotlivé fragmenty aplikace a použité prvky. Při popisu použitých fragmentů a prvků jsou současně uváděny dostupné funkce a možnosti použití aplikace.

Následně je popsán průběh a výsledek testování vytvořené aplikace na tréninkovém běžeckém závodě. Součástí je zhodnocení výsledků a porovnání s jiným měřicím systémem. V poslední části jsou diskutovány možnosti rozšíření mobilní aplikace.

## **Klíčová slova:**

mobilní aplikace, Android, závod, měření závodů, Java, XML

# Abstract

This bachelor thesis deals with the issue of timekeeping sports events, such as running or cycling races. The first part of the thesis describes currently available methods of timekeeping races and then introduces some services that can time keep the whole race.

The practical part of this bachelor thesis is the design and implementation of a mobile application for the operating system Android. This application can be used to time keep races. So the second part of the thesis deals with the design of this application. Attention is paid to the requirements for functionality, possibilities of use, description of the application, description of used technologies, and architecture of the application. In the third part, the bachelor thesis deals with the implementation. There are descriptions of the individual fragments of the application and descriptions of used elements. When describing the used fragments and elements, the available functions and possibilities of using the application are listed at the same time.

Subsequently, the testing process and result of testing at a training running race are described. It includes evaluation of the results and comparison with another timekeeping system. The last part discusses the possibilities of extending the mobile application.

## **Keywords:**

mobile applications, Android, competition, race time measurement, Java, XML

## Poděkování

Rád bych poděkoval vedoucí mé bakalářské práce Ing. Janě Kolaje Ehlerové, Ph.D., a také Ing. Janu Kolajovi, Ph.D., za cenné rady, vynaložený čas a také organizaci tréninkového závodu pro testování mé vytvořené mobilní aplikace. Dále bych chtěl poděkovat také všem účastníkům tohoto závodu.

# Obsah

Úvod.....	12
<b>1 Současné nabízené služby měření závodů.....</b>	<b>13</b>
1.1 Měření pomocí čipové technologie .....	13
1.1.1 Ptime.cz .....	13
1.1.2 AM CHRONO.....	14
1.1.3 IRONTIME .....	15
1.2 Mobilní aplikace .....	16
1.2.1 Mobilní aplikace Race Timer .....	16
1.3 Měření pomocí kombinace čipové technologie a mobilní aplikace .....	19
1.3.1 Mobilní aplikace ORGSU Sport Timekeeping .....	19
1.4 Výsledek analýzy.....	19
<b>2 Návrh mobilní aplikace .....</b>	<b>20</b>
2.1 Požadavky.....	20
2.2 Možnosti využití.....	22
2.3 Popis aplikace.....	22
2.4 Název a grafická identita .....	22
2.5 Architektura aplikace.....	23
<b>3 Implementace aplikace .....</b>	<b>24</b>
3.1 Popis fragmentů aplikace .....	25
3.1.1 Čas .....	27
3.1.2 Všichni závodníci v cíli .....	32
3.1.3 Výsledky.....	33
3.1.4 Startovní listina.....	34
3.1.5 Přidat závodníka do startovní listiny .....	35
3.1.6 Zobrazení závodníka ve startovní listině.....	36
3.1.7 Upravit závodníka ve startovní listině.....	38
3.1.8 Nastavení .....	39
3.2 Komunikace mezi Fragmentem a Activity .....	41
3.3 Použité Android prvky .....	41
3.3.1 View a ViewGroup.....	41
3.3.2 TextView, EditText a Button .....	42
3.3.3 Chronometer .....	43
3.3.4 CountdownTimer.....	44



3.3.5	ListView .....	45
3.3.6	Adapter .....	45
3.3.7	Spinner.....	46
3.3.8	Toast .....	47
3.3.9	Dialog .....	48
3.3.10	Options menu.....	49
3.3.11	LinearLayout, ScrollView .....	49
3.4	Databáze .....	49
3.4.1	Popis databázových tabulek .....	50
3.5	Import startovní listiny .....	51
3.6	Export výsledků.....	52
3.7	Kategorie závodu.....	54
3.8	Návod k použití .....	56
3.8.1	Nastavení závodu .....	56
3.8.2	Nastavení vlastního závodu.....	56
3.8.3	Nastavení načteného závodu .....	57
3.9	Zveřejnění aplikace .....	59
<b>4</b>	<b>Testování aplikace.....</b>	<b>61</b>
4.1	Výsledek testování.....	62
4.2	UX, neboli uživatelský prožitek .....	64
<b>5</b>	<b>Možnosti rozšíření.....</b>	<b>65</b>
5.1	Návrhy na vylepšení aplikace.....	65
5.2	Webová aplikace .....	66
	<b>Závěr.....</b>	<b>69</b>
	<b>Literatura.....</b>	<b>70</b>
	<b>Přílohy .....</b>	<b>72</b>
	Příloha A: Aplikace na Google Play.....	72
	Příloha B: Zdrojové kódy.....	72
	Příloha C: Import startovní listiny – vzorový CSV soubor.....	72
	Příloha D: Export výsledků do CSV souboru – příklad.....	73
	Příloha E: Export výsledků do PDF souboru – příklad.....	75
	Příloha F: Export výsledků do HTML souboru – příklad.....	76

## Seznam zkratek

UI	<i>User Interface</i>
UX	<i>User Experience</i>
CSV	<i>Comma-separated values</i>
PDF	<i>Portable Document Format</i>
HTML	<i>Hyperext Markup Language</i>
SMS	<i>Short Message Shortcut</i>
RFID	<i>Radio frequency identification</i>
SI	<i>SportIdent</i>
XML	<i>Extensible Markup Language</i>
SDK	<i>Software Development Kit</i>

## Seznam obrázků

Obrázek 1: Webové stránky společnosti AM CHRONO .....	15
Obrázek 2: Aplikace Race Timer - základní obrazovka.....	17
Obrázek 3: Aplikace Race Timer - zobrazení závodníků v cíli.....	18
Obrázek 4: Ikona aplikace .....	23
Obrázek 5: Activity skládající se z více fragmentů [9] .....	24
Obrázek 6: Hlavní Activity aplikace .....	25
Obrázek 7: Fragment s rozkliknutým horním menu .....	26
Obrázek 8: Hlavní fragment aplikace před spuštěním závodu – start pomocí tlačítka ..	28
Obrázek 9: Hlavní fragment aplikace před spuštěním závodu - start v zadaný čas .....	29
Obrázek 10: Hlavní fragment aplikace v průběhu závodu .....	30
Obrázek 11: Úprava čísla doběhlého závodníka .....	31
Obrázek 12: Chybný zápis čísla závodníka.....	32
Obrázek 13: Fragment se všemi závodníky, kteří doběhli .....	33
Obrázek 14: Fragment výsledky.....	34
Obrázek 15: Fragment zobrazující startovní listinu závodu.....	35
Obrázek 16: Fragment, který se stará o přidání závodníka do startovní listiny .....	36
Obrázek 17: Fragment, který zobrazí informace o závodníkovi .....	37
Obrázek 18: Fragment, který zobrazí informace o závodníkovi - typ závodu časovka .	38
Obrázek 19: Fragment, který upraví závodníka .....	39
Obrázek 20: Fragment nastavení závodu .....	40
Obrázek 21: View a ViewGroup [10] .....	42
Obrázek 22: Tlačítka, jejichž pozadím je obrázek .....	43
Obrázek 23: Ukázka Spinnerů.....	47
Obrázek 24: Toast oznamující chybu při zadávání čísla závodníka.....	48
Obrázek 25: Entitně-relační model databáze.....	51

Obrázek 26: Možnosti importu startovní listiny.....	52
Obrázek 27: Export výsledků .....	53
Obrázek 28: Sdílení výsledků.....	54
Obrázek 29: Možnost filtrování startovní listiny podle kategorií .....	55
Obrázek 30: Výsledky závodu filtrované podle kategorie .....	55
Obrázek 31: Úspěšně načtený závod.....	58
Obrázek 32: Schéma možností nastavení závodu .....	59
Obrázek 33: Hlavní grafika aplikace PocketRaceTimer .....	60
Obrázek 34: Testování měření běžeckých závodů pomocí vytvořené mobilní aplikace	62
Obrázek 35: Graf rozdílů v měření.....	63
Obrázek 36: Návrh webové aplikace - hlavní stránka.....	67
Obrázek 37: Návrh webové aplikace – registrace závodu.....	68

# Úvod

V současné době existuje spousta služeb, které převezmou měření běžeckých či cyklistických závodů do své režie a pomocí čipové technologie dokáží profesionálně změřit téměř každý závod. Pro větší závody s dostatečným finančním rozpočtem je to ideální řešení a pravděpodobně po něm ihned sáhnou, pokud již sami nevlastní nějaké své RFID čipy a k nim příslušnou časomíru. Problém ovšem nastává u menších sportovních událostí, které si tyto nákladné služby dovolit nemohou. K měření proto často používají klasické stopky a papír, což s sebou v dnešní době plné technologií přináší spoustu práce.

Cílem této bakalářské práce je vyvinout mobilní aplikaci přesně pro tyto potřeby. Aplikace se zaměřuje na menší závody a snaží se jim poskytnout jednoduché měření pomocí mobilního telefonu. Pomocí ní lze měřit širší spektrum závodů – od těch úplně nejmenších, kdy si pár kamarádů uspořádá kratší běh a třeba ani nemají sepsanou startovní listinu, až po ty větší, které mají například pevně daný čas startu, několik desítek závodníků na startovní listině a celý závod se běží jako časovka, tedy každý závodník vyráží na trať v jiný okamžik. I přes toto množství funkcionalit je stále cíleno na jednoduchost použití, celá aplikace na první dojem působí jako chytré stopky.

Aplikace je vytvořena pro mobilní zařízení s operačním systémem Android, jelikož většina lidí má na svém telefonu právě tento operační systém, a to zejména pro snadnou dostupnost a dostatečnou bezpečnost. Jedná se o nativní aplikaci, která byla vytvořena v jazyce Java pomocí vývojového prostředí Android Studio.

Tato bakalářská práce ve své první části seznámí čtenáře s aktuálním stavem měření závodů a přiblíží některé konkrétní služby, které jsou schopné organizátorům závod změřit. Následně práce pojednává o samotné aplikaci, její struktuře, implementaci a návodu k použití. Dále je popsán průběh a výsledek testování na menším běžeckém závodě a také další možnosti rozšíření této práce.

# 1 Současné nabízené služby měření závodů

Jak již bylo naznačeno v úvodu, současní organizátoři závodu mají širokou škálu možností, jak si nechat změřit svůj závod. Všechny tyto možnosti se dají rozdělit na tři základní kategorie. První z nich je kategorie, kdy si pořadatel změří svůj závod tradiční metodou „bába-tužka-blok“, jednoduše vezme stopky a pár papírů a všechny časy si zapisuje společně s čísly závodníků. Je zřejmé, že tato možnost je časově náročná a opravdu pracná, leckdy se ještě několik hodin po závodě musí dávat dohromady přepis zápisů do elektronické formy. Z tohoto důvodu je vhodné zapojit moderní technologie a vyhnout se repetitivnímu zápisu. Značným vylepšením může být měření pomocí mobilního zařízení. Do této kategorie spadá aplikace, která je vytvořená v rámci této bakalářské práce. Třetí a zároveň nejmodernější kategorií je měření závodů pomocí čipové technologie. Tato metoda se v praxi používá v drtivé většině závodů, jelikož je pro organizátory nejvhodnější.

## 1.1 Měření pomocí čipové technologie

Na internetu lze najít mnoho služeb, které se za poplatek postarají o celé měření vašeho závodu, a to za pomoci čipové technologie. Ve většině případů se používají RFID čipy. Název RFID označuje technologii využívající rádiové vlny k bezkontaktní výměně dat k identifikaci prvků a samotná zkratka pochází z anglického spojení *radio-frequency identification*, což lze přeložit jako radiofrekvenční identifikace. [1] Tyto čipy se závodníkům připnou na startovní číslo, nebo mohou být umístěny na kotníku či zápěstí.

Zástupci těchto služeb mohou být například následující tři – ptime.cz, AM CHRONO a IRONTIME.

### 1.1.1 Ptime.cz

Webová stránka [ptime.cz](https://www.ptime.cz)<sup>1</sup> měří závody již od roku 2016 a má za sebou více jak 120 naměřených akcí. Jedná se o webovou stránku s jednoduchým a přehledným designem, kterou zastřešuje organizace Pteam sport, z.s. Jejich služby zahrnují změření závodu pomocí RFID čipů, online registraci závodníků a registraci závodníků na místě s přidělením startovních čísel. Co se týče online registrace, tak ta může být propojena přes společné weby poskytovatele a organizátora, záleží na domluvě. Dále pořadateli služba zveřejní závod

---

<sup>1</sup> <https://www.ptime.cz>

na svém facebookovém profilu a webových stránkách. Ptime také po závodě zpracuje všechny výsledky a zašle každému závodníkovi diplom ve formátu PDF, pokud si to organizátor přeje. Všechny tyto aktivity jsou zahrnuty v ceně služeb, ta ovšem není pevně dána. Na webu lze najít cenový konfigurátor, kde se organizátorovi po vyplnění všech informací o závodě zobrazí výsledná cena.

Ptime dále nabízí možnost dalšího pronájmu až čtyř reproduktorů (cena 600,- za kus), bezdrátového mikrofónu (500,-), čtyřmetrové nafukovací startovní brány (1790,-) nebo například pronájem LED displeje s časem (1500,-). [2]

Při organizování malého závodu s 50 účastníky, startem a cílem na jednom místě, bez mezičasu na jiném místě na trati a polohou závodu v okolí Liberce jejich cenový konfigurátor zobrazil přibližnou cenu 5800,-. Snad ve většině případů je vypočítaná přibližná cena vyšší než 4000,- a to se jedná o opravdu malé závody.

### 1.1.2 AM CHRONO

Další webovou stránkou, která nabízí měření závodů pomocí čipové technologie je [AM CHRONO](https://www.amchrono.cz)<sup>2</sup>. Tato společnost za dobu své existence naměřila již 91 sportovní akcí, které obsahovaly celkově 631 závodů.

AM CHRONO využívá pro své měření čipovou technologii ImpinJ, tato technologie lze využít jak s vodotěsnými vratnými čipy, které si závodník umístí na kotník, tak i s jednorázovými samolepicími čipy, které se přilepí na startovní číslo. Přesnost měření by se měla pohybovat okolo 0,01 sekundy. Společnost disponuje vlastním softwarovým vybavením, které by mělo pokrývat většinu požadavků pořadatelů, zajišťují například zobrazení výsledků online na stránkách této společnosti nebo propůjčení tabletu s výsledky komentátorovi. Dále služba nabízí možnost půjčení oboustranné LED tabule, která zobrazuje čas probíhajícího závodu. V neposlední řadě služba organizátorům vytiskne startovní i výsledkovou listinu závodu. Webové stránky společnosti jsou moderní a přehledné (viz Obrázek 1), službu lze využít na širokou škálu sportů - například běh, triatlon, horská kola, in-line brusle nebo klasické lyžování. [3] Nevýhodou této služby je, že nelze nezávazně zjistit cenu.

---

<sup>2</sup> <https://www.amchrono.cz>

The screenshot shows the AM CHRONO website. At the top left is the logo. The navigation menu includes 'Úvod', 'Kalendář akcí', 'Výsledky', 'Technologie a SW', 'Cenová nabídka', and 'Kontakt'. The main banner features a bicycle wheel and the text 'Čipová časomíra' with a subtext 'Zajistíme Vám změření závodů pomocí čipů.' and a 'Více informací' button. Below the banner are two tabs: 'Kalendář nejbližších akcí' and 'Aktuální výsledky'. The 'Kalendář nejbližších akcí' tab is active, displaying a table of events.

Start	Akce	Lokalita	Závodníků?	Typ sportu
11.06.2021	PŘIPRAVUJEME SE: DTB serie - Běh K5 po kunovických vrcholech 2021	Kunovice - Hasičské cvičiště	0	<a href="#">Zaregistruj se na akci</a>
12.06.2021	PŘIPRAVUJEME SE: Pohoda Run Únanov	Únanov	9	<a href="#">Zaregistruj se na akci</a>
02.10.2021	Kaskáda Race 2021	Vranov nad Dyjí	119	<a href="#">Zaregistruj se na akci</a>
02.10.2021	Kaskáda Race týmy a školy 2021	Vranov nad Dyjí	8	<a href="#">Zaregistruj se na akci</a>

Obrázek 1: Webové stránky společnosti AM CHRONO

### 1.1.3 IRONTIME

Další z mnoha služeb, které se dají najít na internetu je [IRONTIME](https://irontime.cz)<sup>3</sup>. IRONTIME se hodí pro závody, které počítají s velkým množstvím účastníků, protože víceméně převezme organizaci do vlastních rukou. Služba řeší vše od přihlašování závodníků do závodu pomocí elektronické přihlášky až po odesílání neoficiálních výsledků závodníkům pomocí SMS. Dále každý závodník obdrží oficiální výsledky do své e-mailové schránky, je ale také možné si je stáhnout z webu, kam se exportují průběžně během závodu, nebo až po skončení. Je zde možnost zkopírovat si je do excelové tabulky nebo stáhnout jako PDF. Před závodem se na stránkách generuje startovní listina, která je všem veřejná. Další výhodou pro organizátora je možnost editovat závodníky a přiřazovat k nim příchozí platby za startovné.

Měření probíhá za pomoci RFID UHC čipů. Na většinu závodů se používají jednorázové čipy Smartrac Dobgone, které jsou závodníkům přilepeny na startovní čísla. Pokud se ovšem jedná o triatlonový závod, jsou využity vratné čipy HuTag XC, které jsou pasivní a vodotěsné. Jsou umístěny na kotníku nebo zápěstí závodníka.

Za posledních 8 let služba naměřila více jak 600 závodů. Jak už bylo ale řečeno, více se hodí na závody většího charakteru, a to hlavně z důvodu vyšších cen. Ceny totiž začínají

<sup>3</sup> <https://irontime.cz>



na 11 000,– za dvouhodinový závod ve všední den a šplhají až na 49 000,– za motocrossové závody. K ceně je vždy nutné připočítat dopravné, cestovní čas, cenu za SMS zprávy a cenu za čipy. [4]

## 1.2 Mobilní aplikace

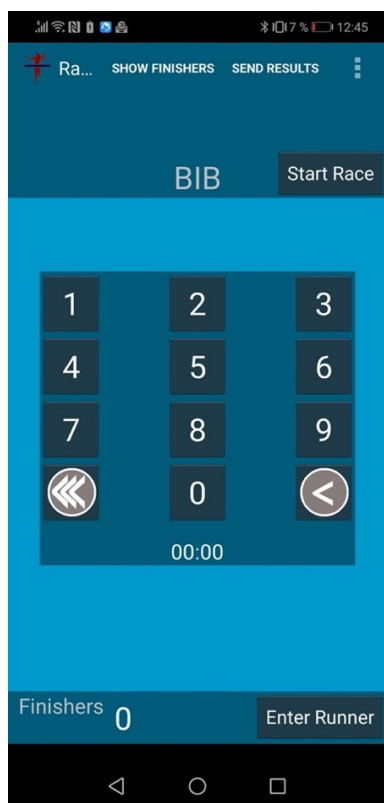
Závody se dají měřit také pomocí mobilních aplikací, na což se zaměřuje praktická část této bakalářské práce. Při průzkumu [Google Play](https://play.google.com/store)<sup>4</sup>, tedy oficiálního obchodu s aplikacemi pro platformu Android, lze nalézt pár mobilních aplikací, pomocí kterých by se z části dal změřit závod. Většina z nich jsou ale pouze lehce upravené klasické stopky. Jedna z mála, která se tváří, že lze být použita přímo na měření závodů, je aplikace s názvem *Race Timer*.

### 1.2.1 Mobilní aplikace Race Timer

Tato aplikace nabízí uživateli možnost odstartovat závod a poté zadávat čísla závodníků, kteří dobíhají do cíle, splňuje tedy základní princip měření závodu. Základní obrazovka aplikace je zobrazena na obrázku (Obrázek 2).

---

<sup>4</sup> <https://play.google.com/store>



Obrázek 2: Aplikace Race Timer - základní obrazovka

Obrazovka obsahuje číselný ciferník, pomocí kterého zadává časoměřič číslo závodníka. Dále je možné zobrazit všechny závodníky, kteří doběhli do cíle, tato část aplikace je vidět na dalším obrázku (Obrázek 3).

Place	Number	Time
1	67	1:03:03
2	88	1:03:06
3	77	1:03:08
4	12	1:03:10
5	15	1:03:12
6	66	1:03:14
7	99	1:03:16
8	44	1:03:19
9	8	1:03:21
10	10	1:03:23

Obrázek 3: Aplikace Race Timer - zobrazení závodníků v cíli

Aplikace tedy implementuje základní funkcionalitu chytrých stopek na měření závodu. Dále je zde možnost poslat výsledky závodu, k čemuž může být využita jakákoliv aplikace instalovaná v zařízení. Uživatel má také možnost manuálně nastavit aktuální čas závodu.

Přes toto všechno obsahuje aplikace spoustu nedostatků. Prvním a velmi významným nedostatkem je fakt, že pokud uživatel vypne aplikaci, už se nikdy nedostane k předšlému probíhajícímu závodu. Dalším mínusem je, že nijak nelze upravovat zadaná data. Pokud tedy časoměřič omylem zadá špatné číslo, už nelze provést změnu. Dále také aplikace vůbec nehlídá, zda nebylo zadáno číslo, které už předtím doběhlo. V praxi se tak může stát, že tentýž závodník obsadí v závodě více pozic. Je zde také možnost stisknout tlačítko „Enter runner“ aniž by bylo vyplněné startovní číslo dobíhajícího závodníka, což se na první pohled může zdát jako přínosná funkce – při měření závodu občas nastává situace, kdy dobíhá více závodníků najednou, v ten moment může časoměřič víckrát za sebou stisknout toto tlačítko a později doplnit k těmto časům čísla závodníků. Problémem je ale to, že tato pozdější úprava už není možná. Ve výsledkové listině tak zůstanou pouze časy bez čísel závodníků. V případě, že by uživatel potřeboval změřit závod, který nemá hromadný start, tak by mu tato aplikace vůbec neposloužila, jelikož nenabízí ani

žádné funkce, které by umožňovaly práci se startovní listinou, natož pak s časovkou. Mírným nedostatkem by také mohl být design aplikace, který na první pohled nemusí působit dobrým dojmem.

Aplikace Race Timer tedy nabízí základní funkcionalitu, ovšem není dotažena do zdárného konce. V praxi ji téměř jistě nelze použít k měření žádného závodu, a to už jen z toho důvodu, že si uživatel nemůže být jist, že se mu náhle neztratí data, která během používání nasbíral.

## 1.3 Měření pomocí kombinace čipové technologie a mobilní aplikace

### 1.3.1 Mobilní aplikace ORGSU Sport Timekeeping

S trochu odlišnou taktikou přišla společnost [ORGSU](https://www.orgsu.com/cs/)<sup>5</sup>. Tato společnost nabízí opravdu velmi mnoho služeb. V podstatě pokrývá snad všechny druhy organizace a měření závodu, na které si člověk může vzpomenout. ORGSU měří závody jak pomocí čipové technologie, tak také pomocí mobilní aplikace, která je vyvinutá pro platformy Android a iOS. Měřit závody umí i ručně, pomocí tužky a papíru. K dalším nabízeným službám patří e-shop, hromadný e-mailing, online registrace, sledování závodu živě na mapě a kalendář organizovaných závodů. [5]

Jedná se ovšem o placenou službu, jejíž cena není předem známá. Velkým nedostatkem této služby je nepřehlednost informací. Zákazník bude pravděpodobně velmi dlouho pročítat webové stránky, než vůbec přijde na to, co daná společnost nabízí a co vše je možné si zakoupit. Mobilní aplikaci není možné použít bez registrace. Jedná se o tedy o službu, která má za cíl pokrýt co nejširší spektrum služeb, ovšem postrádá jednoduchost a přehled.

## 1.4 Výsledek analýzy

Po této analýze současných dostupných metod měření závodů lze říci, že neexistuje žádná aplikace pro mobilní zařízení, kterou by si uživatel mohl snadno, rychle a zdarma stáhnout na Google Play a následně by s její pomocí mohl spolehlivě změřit svůj závod.

---

<sup>5</sup> <https://www.orgsu.com/cs/>

## 2 Návrh mobilní aplikace

Cílem tedy bylo uživatelům zprostředkovat aplikaci, která nemá být určena jako konkurence pro měření pomocí profesionální čipové technologie, ale má sloužit jako náhrada ručního měření závodu pomocí stopek a zapisování výsledků na papír.

Takováto aplikace byla v průběhu práce vytvořena. Je dostupná pro mobilní zařízení s operačním systémem Android. Volba padla na Android z toho důvodu, že je rozšířenějším operačním systémem, než operační systém iOS, který je vyvíjený společností Apple. Dle statistik na webu statista.com vypadal trh s mobilními telefony v České Republice v roce 2020 následovně: [6]

- 79 % Čechů používalo telefon právě s operačním systémem Android
- 20,69 % používalo mobilní zařízení s platformou iOS
- 0,31 % využívalo jiný operační systém

### 2.1 Požadavky

Před samotnou implementací, nebo v průběhů, byly sestavovány hlavní požadavky na vytvořenou aplikaci:

- Požadovaná aplikace měla být zdarma a veřejně dostupná, nejlépe tedy na oficiálním obchodu s aplikacemi pro platformu Android – obchod Google Play.
- Důležitým požadavkem bylo, aby aplikace umožňovala spolehlivě změřit sportovní akci, tedy aby v sobě měla zabudované stopky, neboli čas závodu, a následně, aby do ní bylo možné dopisovat závodníky, kteří zrovna dobíhají do cíle.
- Mělo být umožněno zobrazit všechny závodníky, kteří již doběhli a také provádět různé úpravy, jako třeba měnit číslo závodníka nebo odstranit závodníka z této výsledkové listiny.
- Dále bylo potřeba, aby si aplikace dokázala poradit s případem, kdy probíhá více závodníků současně, to znamená vyřešit situaci, kdy by nebylo možné stihnout napsat postupně čísla všech těchto závodníků a odeslat je v přesný moment, kdy protínají cílovou pásku. Z tohoto důvodu bylo žádoucí, aby existovala pro časoměřiče možnost stisknout víckrát tlačítko bez vyplněného čísla závodníka, aby se vytvořily pouze pozice s časovými údaji a následně tyto pozice upravit, neboli dopsat k nim čísla závodníků.

- Aplikace měla umožnit uživateli volbu, jakým způsobem chce odstartovat závod, s tím, že by měly existovat tyto dvě možnosti:
  - Odstartování závodu stisknutím tlačítka „Start“
  - Odstartování závodu automaticky v přesně stanovený čas. V tomto případě si uživatel v nastavení závodu zvolí datum a čas startu. Aplikace se poté o start postará sama, součástí toho by mělo být zobrazování informace uživateli o času startu a případný odpočet do startu závodu.
- Dalším požadavkem bylo, aby aplikace uměla spolupracovat se službou [Sportchallenge](https://www.sportchallenge.cz)<sup>6</sup>, což je opět služba, která poskytuje organizátorům čipovou časomíru. Služba umožňuje registrovat závody v jejich systému a následně pomůže s měřením těchto závodů za využití čipové technologie. Spolupráce aplikace a služby je jednoduchá – aplikace měla umět načíst konkrétní registrovaný závod a tento závod změřit za pomoci mobilního zařízení. Ze systému se tedy vždy měla načíst data, která obsahují čas startu závodu, typ závodu, název závodu a startovní listinu.
- Na předchozí požadavek navazuje další velmi významný fakt. Tím je, že aplikace by měla zvládat změřit také závody, kdy nespustí všichni závodníci zároveň, ale vyběhají v předem daných časových intervalech, tedy závody, kterým se v praxi říká „časovka“. Z tohoto důvodu bylo potřeba, aby aplikace uměla pracovat se startovní listinou závodu, aby bylo možné u závodníků ukládat jejich čas startu, v případě že se nejedná o závod s hromadným startem. Tuto startovní listinu mělo jít importovat z CSV souboru, případně ji načíst z již existujícího závodu, který je registrovaný na Sportchallenge. Aplikace měla umožnit uživateli s touto listinou dále pracovat, to znamená odstraňovat a upravovat již existující závodníky, případně přidávat závodníky nové.
- Posledním požadavkem na funkčnost byla možnost exportovat výsledky po ukončení závodu do různých souborových formátů.
- Důraz měl být ovšem kladen na jednoduchost. I přes tyto poněkud rozsáhlejší funkcionality měla aplikace působit hlavně dojmem „chytrých stopek“, tedy aby uživatel po spuštění ihned pochopil, jak aplikaci ovládat.

---

<sup>6</sup> <https://www.sportchallenge.cz>

## 2.2 Možnosti využití

Vytvořená aplikace by následně měla být využívána na menších sportovních akcích jako to nejjednodušší řešení pro organizátory. Záběr sportů v podstatě není omezený, takže je vhodná jak na běžecké závody, tak na cyklistiku, in-line bruslení, klasické lyžování, triatlon, případně i na nějaké malé motosport závody. Zjednodušeně lze říci, že se dá použít všude tam, kde by se dalo měřit pomocí klasických stopek. Je navržena tak, aby pojala širší potřeby organizátorů, ať už se jedná o krátký závod pěti kamarádů kolem koupaliště, nebo o několikahodinový závod s intervalovými starty. V tom druhém případě je ovšem nutné zmínit, že profesionálním čipovým technologiím se z hlediska přesnosti měření rovnat nemůže.

## 2.3 Popis aplikace

Pro vývoj aplikace bylo zvoleno vývojové prostředí Android Studio. Toto vývojové prostředí je optimální pro vývoj mobilních aplikací určených pro zařízení s operačním systémem Android. Jeho velkou výhodou je integrovaný Android emulátor, pomocí kterého lze aplikaci testovat na libovolném virtuálním Android zařízení, je možné zvolit si různé verze Androidu.

Pro vývoj mobilních aplikací pro Android je možné využít programovací jazyky Java nebo Kotlin. Díky předešlým zkušenostem s jazykem Java a neznalostí programovacího jazyka Kotlin padla volba na první možnost. Jedná se o univerzální objektově orientovaný programovací jazyk, který je široce využíván pro vývoj aplikací pro laptopy, mobilní zařízení, vědecké superpočítače, datová centra, herní konzole a další. Výhodou Javy je rychlost, bezpečnost, spolehlivost a přenositelnost. [7]

Aplikace je určena pro zařízení, které běží na Androidu verze 4.4 (KitKat) a vyšším. Tato minimální verze odpovídá SDK (Software Development Kitu) ve verzi 19.

## 2.4 Název a grafická identita

Velmi důležitými prvky aplikace jsou název a ikona. Jedná se o prvky, které zprostředkovávají uživateli první dojem. Při tvorbě bylo využíváno mnoho pracovních názvů, nakonec byl vybrán název „*PocketRaceTimer*“, což může být volně přeloženo jako „kapesní časomíra“. Tento název se snaží poukázat na hlavní výhodu aplikace, a sice to, že celý časoměrný systém se časoměřiči vejde do kapsy. Při tvorbě ikony aplikace byla

upřednostňována jednoduchost, ikona obsahuje symbol stopek na černém pozadí (viz Obrázek 4).



Obrázek 4: Ikona aplikace

## 2.5 Architektura aplikace

K vývoji aplikace byla využita MVC architektura. Lukáš Rychtecký ve své přednášce o MVC uvádí, že: „*Architektura MVC dělí aplikaci na 3 logické části tak, aby je šlo upravovat samostatně a dopad změn byl na ostatní části co nejmenší. Tyto tři části jsou Model, View a Controller. Model reprezentuje data a business logiku aplikace, View zobrazuje uživatelské rozhraní a Controller má na starosti tok událostí v aplikaci a obecně aplikační logiku.*“ [8]

V aplikaci odpovídá Controlleru prvek s názvem *Activity*, View je reprezentován prvkem *Fragment*<sup>7</sup> a Model zastupují vytvořené objekty *RacerModel* (reprezentuje závodníka) a *RaceModel* (reprezentuje závod).

---

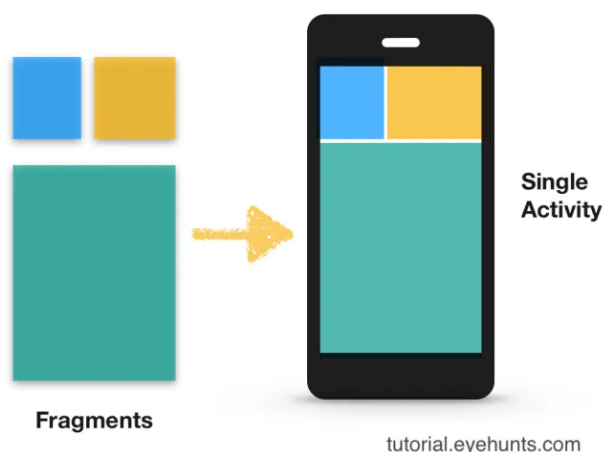
<sup>7</sup> *Activity* a *Fragment* budou popsány v následující kapitole



### 3 Implementace aplikace

Celá aplikace obsahuje pouze jednu hlavní *Activity*. Tento pojem se dá volně vysvětlit jako prezenční vrstva aplikace. Dá se říci, že je to tedy takový *Controller* z MVC architektury, který ovládá jeden pohled aplikace. Pod pojmem pohled si lze představit jednu stránku aplikace, která zobrazuje například tlačítko a nějaký text. Pokud má aplikace více *Activities*, tak to znamená, že má různé pohledy, které se mění podle činnosti uživatele v aplikaci. V závislosti na události se může vyvolat přepnutí těchto aktivit, což znamená, že se uživatel přepne na jinou stránku aplikace, která nabízí jiné ovládací prvky.

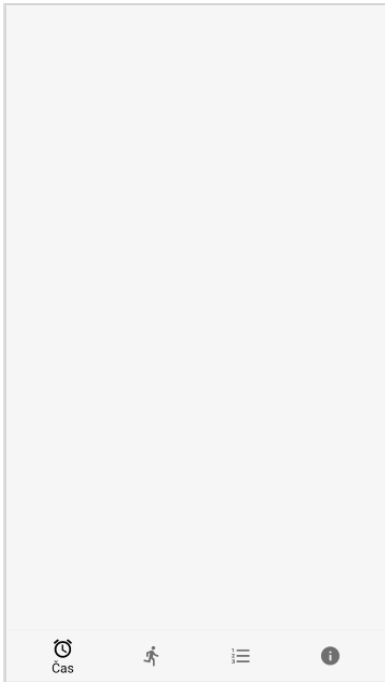
*PocketRaceTimer* aplikace je vytvořena tak, že má pouze jednu tuto hlavní *Activity*, přesto ale nabízí spoustu různých stránek neboli pohledů. A to díky takzvanému *Fragmentu*. *Fragment* si můžeme představit jako část uživatelského rozhraní, může to být skupina nějakých komponent, která je použitelná na více místech. Konkrétní příklad fragmentu může být třeba formulář. Takovýto *Fragment* by obsahoval nějaké pole pro zadání hodnot a jedno tlačítko s nápisem „Odeslat“. Stačilo by tedy pouze jednou sestavit tuto část UI a dále je možné tento formulář používat na více místech aplikace, stačí vždy pouze vložit fragment do zvolené *Activity*. Výhodou poté je, že se jedna *Activity* může skládat s různých fragmentů, tak jak je vidět na obrázku (Obrázek 5). Může je skrývat, zobrazovat nebo měnit v závislosti na vyvolaných událostech.



Obrázek 5: *Activity* skládající se z více fragmentů [9]

Přesně tato výhoda je využita ve vytvořené aplikaci. Celá aplikace je tvořena pouze jednou hlavní *Activity*, a ta obsahuje pouze hlavní menu, které se nachází dole na obrazovce (viz Obrázek 6). V závislosti na zvolené položce menu se zobrazí jiný fragment, tedy jiný

pohled. Důležité ale je, že ať už se zobrazí jakýkoliv fragment, hlavní menu je vždy přístupné.



Obrázek 6: Hlavní Activity aplikace

### 3.1 Popis fragmentů aplikace

Hlavní menu umožňuje uživateli přepínat mezi čtyřmi hlavními fragmenty, neboli stránkami, aplikace. Těmito fragmenty jsou:

- Čas
- Všichni závodníci v cíli
- Startovní listina
- Nastavení

To ovšem platí pouze v případě, že je závod v průběhu, nebo ještě nebyl odstartován. Pokud nastane situace, že byl závod již ukončen, změní se druhá položka hlavního menu ze „Závodníci“ na „Výsledky“ a po kliknutí se zobrazí fragment s názvem „Výsledky“. Přes hlavní menu se tedy lze dostat k pěti fragmentům, záleží na tom, v jaké fázi se závod nachází.

Dále jsou implementovány ještě další fragmenty, a to konkrétně fragment pro přidání závodníka do startovní listiny, zobrazení závodníka startovní listiny a úprava závodníka startovní listiny. Všechny tyto pohledy jsou přístupné ze startovní listiny.

Jak již bylo zmíněno, aplikace reaguje na stav závodu, kde mohou nastat tři možnosti - závod ještě nezačal, závod právě probíhá nebo závod byl ukončen. První přechod, tedy odstartování závodu se řeší v nastavení, kde si uživatel zvolí, jak se závod odstartuje a případně kdy. Druhý přechod, tedy ukončení závodu, lze provést z jakéhokoliv fragmentu aplikace, a to přes horní menu, které je vidět v rozkliknutém stavu na obrázku (Obrázek 7). Při zvolení položky „Ukončit závod“ se zastaví časomíra a změní se zobrazení všech závodníků, kteří již doběhli, na zobrazení výsledků, jak již bylo zmíněno. Pokud závod již ukončený je, tak je tato volba uživateli nepřístupná. Druhá položka tohoto menu má název „Nový závod“ a po jejím zvolení se smažou všechna dosavadní data a aplikace se vrátí do pomyslného bodu nula, kdy je možné nastavit nebo načíst nový závod. Jedná se tedy o přechod z jakéhokoliv stavu, do stavu, kdy závod ještě nezačal. Důležité je ještě zmínit, že po kliknutí na kteroukoliv položku z tohoto horního menu, se ještě zobrazí dialog neboli vyskakovací okno, kde uživatel musí potvrdit svou volbu, aby v případě nechtěného překliknutí nedošlo k nechtěným nevratným změnám.



Obrázek 7: Fragment s rozkliknutým horním menu

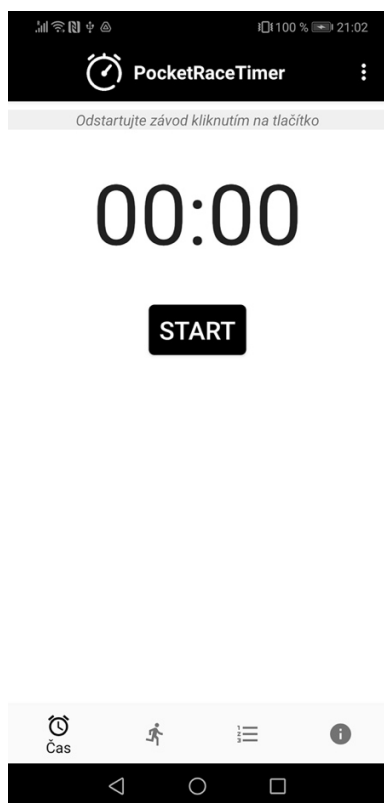
Jelikož toto menu obsahují všechny fragmenty aplikace, tak byl při implementaci vytvořen fragment s názvem *AppFragment*, který dědí od základní třídy *Fragment*. Tento

*AppFragment* obsahuje zmíněné horní menu a obsluhuje jeho funkcionalitu. Všechny ostatní fragmenty už jsou poté jeho potomkem, což znamená, že toto menu zobrazují.

Každý fragment ale po kliknutí na některou položku z tohoto menu a tedy ukončení nebo vyvolání nového závodu, reaguje jinak. Například fragment, který obsluhuje nastavení závodu musí uživateli po zvolení nového závodu znovu zpřístupnit všechna nastavení. Dále fragment, který obsahuje časomíru závodu musí po ukončení závodu tuto časomíru zastavit. Z tohoto důvodu obsahuje *AppFragment* dvě abstraktní metody s názvy *handleRaceStatus* a *raceEnded*. Abstraktní jsou proto, protože je neimplementuje samotná třída *AppFragment*, nýbrž musí být implementovány samotnými potomky této třídy. To znamená, že například po ukončení závodu zavolá *AppFragment* metodu *raceEnded*, kterou má každý fragment implementovanou podle sebe, což zajistí, že se stane přesně to, co se u daného fragmentu vyžaduje.

### 3.1.1 Čas

Fragment, který je v menu nazvaný jako „Čas“, je takovým hlavním fragmentem celé aplikace. Objeví se vždy jako první po spuštění a obsahuje samotné stopky, neboli čas závodu. Na této stránce probíhá zadávání čísel závodníků, kteří právě doběhli do cíle, dále tu také lze odstartovat samotný závod. Fragment je vidět na obrázku (Obrázek 8). Takto jsou rozloženy komponenty, pokud se závod startuje pomocí tlačítka „Start“.



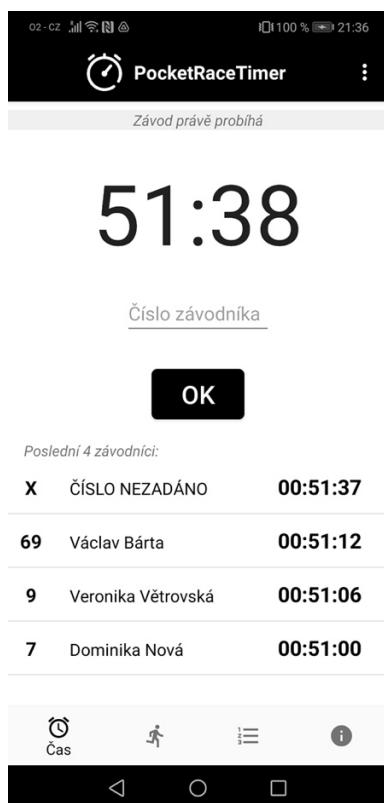
Obrázek 8: Hlavní fragment aplikace před spuštěním závodu – start pomocí tlačítka

Další možností startu je, že uživatel v nastavení zvolí přesný čas, kdy se závod má odstartovat. Tuto možnost volí organizátoři, kteří mají předem zvolený čas, v který se musí vyběhnout a ví, že tomu tak bude. Pokud je tedy tato možnost zvolena, fragment vypadá tak, jak je vidět na obrázku (Obrázek 9). Informační text nahoře s šedým pozadím uživateli zobrazí, kdy přesně, nebo za jak dlouho proběhne start závodu. Pokud start nenastane tentýž den, zobrazí se celé datum a čas startu. Je-li ale start naplánovaný ve stejný den, kdy je aplikace používána, ukazuje se v informační liště odpočet do startu závodu. Čím více se blíží start, tím je odpočet konkrétnější, To znamená, že nejdříve aplikace ukazuje pouze za kolik hodin začne závod, těsně před startem už zobrazí přesný odpočet vteřin.



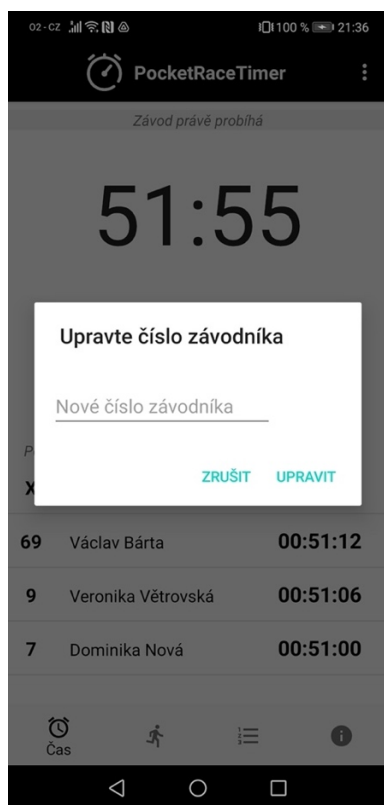
Obrázek 9: Hlavní fragment aplikace před spuštěním závodu - start v zadaný čas

Ať už se odstartuje pomocí tlačítka nebo automaticky na čas, tak se čas startu uloží do databáze a hlavní fragment vypadá v průběhu závodu tak, jak je vidět na dalším obrázku (Obrázek 10). Fragment obsahuje pole pro vyplnění čísla dobíhajícího závodníka a tlačítko „Ok“ pro zadání jeho cílového času. Pro lepší přehled se pod tímto tlačítkem nachází ještě seznam s posledními čtyřmi zadanými závodníky. Tento seznam je zde hlavně z toho důvodu, aby uživatel viděl, že aplikace zadaného závodníka opravdu zaregistrovala a úspěšně uložila jeho cílový čas.



Obrázek 10: Hlavní fragment aplikace v průběhu závodu

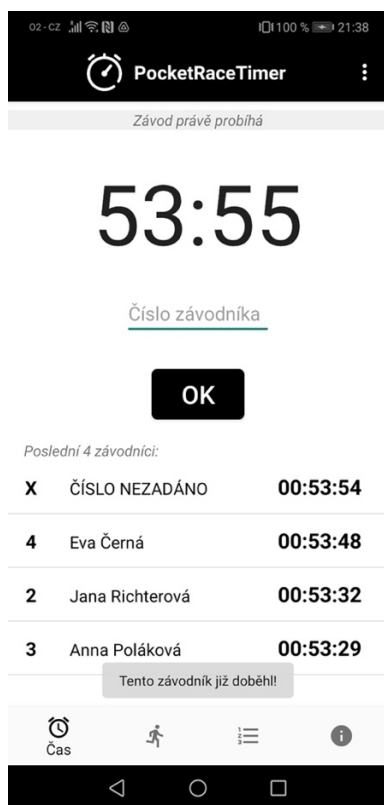
Poslední důležitou funkcionalitou je možnost přidat cílový čas, aniž by bylo zadáno startovní číslo závodníka. To se hodí zejména při těsném doběhu více závodníků najednou, kdy by uživatel nestihl přesně zadat všechna čísla a určit jejich časy. Stačí tedy pouze stisknout tlačítko „Ok“, bez zadaného startovního čísla, a aplikace se zachová tak, jak je možné zpozorovat na předešlém obrázku (Obrázek 10). Alternativou je klepnutí na čas závodu, to vyvolá stejnou událost – vytvoří se závodník bez čísla a jména, pouze s časovým údajem. Tohoto závodníka lze později jednoduše upravit – uživatel dlouze stiskne pole s tímto závodníkem a objeví se menu s položkami „Upravit“ a „Odstranit“, pokud tedy uživatel vybere položku pro úpravu závodníka, tak se objeví okno s možností zadat nové číslo (Obrázek 11). Stejně tak je možné upravovat a odstraňovat závodníky, kteří již mají přiřazené startovní číslo.



Obrázek 11: Úprava čísla doběhlého závodníka

Pokud by se stalo, že uživatel zadá číslo, které již figuruje v seznamu všech závodníků, kteří již doběhli do cíle, objeví se informace „Tento závodník již doběhl“ a neumožní to tak přepsat jeho cílový čas. Přesto se ale vytvoří nový závodník, který žádné číslo nemá, tedy stejně jako kdyby uživatel nezadal číslo žádné. Tato funkcionality je zde pro případ, že se časoměřič přehlédl nebo omylem zapsal špatně číslo, což znamená, že doběhl závodník s jiným číslem, které lze později doplnit pro tento vytvořený časový údaj. Tento případ znázorňuje obrázek (Obrázek 12).





Obrázek 12: Chybný zápis čísla závodníka

### 3.1.2 Všichni závodníci v cíli

V pořadí druhá položka v hlavním dolním menu nese název „Závodníci“ a po jejím zvolení se zobrazí fragment, který je na obrázku (Obrázek 13). Hlavní částí obsahu tohoto fragmentu je seznam všech závodníků, kteří byli v cíli zadání do aplikace jakožto závodníci, kteří doběhli. Závodníci jsou seřazeni sestupně podle času v cíli, tedy přesně tak, jak je časoměřič zadával. Tento fragment tedy slouží k tomu, aby si uživatel mohl prohlédnout všechny závodníky, které už zadal, dále aby si mohl předběžně zkontrolovat výsledné časy závodníků a v neposlední řadě je zde stejná možnost upravovat nebo odstraňovat závodníky. Toto lze provádět znovu přes menu, které se vyvolá dlouhým stisknutím pole s daným závodníkem, následně se zobrazí možnosti „Upravit“ a „Odstranit“.

Součástí tohoto fragmentu je také menší časomíra závodu, která je zde pouze pro informativní účel, slouží k lepšímu přehledu uživatele o stavu a času závodu.

02 - cz 100% 21:40

**PocketRaceTimer**

55:07

Počet závodníků v cíli: 21

102	Petra Černá	00:54:56
100	Eliška Stránská	00:54:41
5	Pavčina Vrchovska	00:54:12
6	Vendula Patecká	00:53:54
4	Eva Černá	00:53:48
2	Jana Richterová	00:53:32
3	Anna Poláková	00:53:29
66	Šimon Novák	00:53:06
62	Tomáš David	00:52:53
61	David Tok	00:52:52

Závodníci

Obrázek 13: Fragment se všemi závodníky, kteří doběhli

### 3.1.3 Výsledky

Pokud je závod ukončený, změní se druhá položka hlavního menu na „Výsledky“ a zobrazí se fragment, který je na následujícím obrázku (Obrázek 14).

1:02:33			
EXPORTOVAT VÝSLEDKY		Všechny kategorie	
1.	<b>Lukáš Kořínek</b>	67	00:50:09 Muži
2.	<b>Vojtěch Musil</b>	64	00:50:22 Muži
3.	<b>Adam Grancil</b>	60	00:50:38 Muži
4.	Prokop Němec	63	00:50:49 Muži
5.	Dominika Nová	7	00:51:00 Ženy
6.	Veronika Větrovská	9	00:51:06 Ženy
7.	Václav Bárta	69	00:51:12 Muži
8.	Tereza Stará	1	00:51:37 Ženy

Obrázek 14: Fragment výsledky

Fragment obsahuje list závodníků, kde je nejdříve zobrazena jejich pozice a následně jméno, číslo, výsledný čas a kategorie. Výsledky je možné filtrovat podle kategorií pomocí *spinneru*, který se nachází vpravo nahoře. Po zvolení některé z kategorií se zobrazí pouze závodníci dané kategorie, tentokrát s novým umístěním, které je počítané jako umístění v dané kategorii. Pokud některý ze závodníků závod nedokončil, v kolonce umístění se místo čísla pozice zobrazuje písmeno „N“.

Tento fragment znovu obsahuje menší časomíru závodu pro lepší přehled. Dále se zde ještě nachází důležité tlačítko „Exportovat výsledky“, které uživateli umožní sdílet výsledky závodu.

### 3.1.4 Startovní listina

Při zvolení třetí položky v hlavním menu s názvem „Startovka“ se zobrazí fragment, který uživateli poskytuje zobrazení a manipulaci se startovní listinou závodu. I v tomto fragmentu se nahoře zobrazuje malá časomíra závodu, dále je zde znovu možnost filtrovat startovní listinu podle kategorií pomocí *spinneru*. Vedle filtru se nachází tlačítko

pro přidání dalšího závodníka do startovní listiny, které zobrazí jiný fragment, který se postará o získání všech údajů a samotné přidání. Dále je zde tlačítko pro import startovní listiny

Pod těmito komponenty se už nachází samotná listina v podobě seznamu. Závodníci jsou seřazeni podle startovního čísla a samotný seznam zobrazí u každého závodníka jeho startovní číslo, jméno a příjmení, kategorii a jeho případný výsledný čas. Každého závodníka je možné upravit nebo odebrat a to za pomoci dlouhého stisknutí, které vyvolá menu s položkami „Upravit“ a „Odstranit“. Další možností pro uživatele je rozkliknout zvoleného závodníka a zobrazit tak všechny jeho informace. Celý fragment je vidět na obrázku (Obrázek 15)



Obrázek 15: Fragment zobrazující startovní listinu závodu

### 3.1.5 Přidat závodníka do startovní listiny

Pokud uživatel potřebuje přidat do startovní listiny dalšího závodníka, poslouží k tomu fragment, který je na obrázku (Obrázek 16).



Obrázek 16: Fragment, který se stará o přidání závodníka do startovní listiny

Zde je potřeba vyplnit všechny údaje, kromě data narození a názvu týmu. Dále se startovní číslo nesmí shodovat s číslem jiného závodníka. Při volbě kategorie, do které patří závodník, se zobrazí možnost „+ Přidat kategorii“, kde lze vytvořit novou kategorii závodu. Pokud závod zatím neobsahuje žádné kategorie, je zde místo *spinneru* textové pole, kam se rovnou zadá název kategorie. Zadá-li uživatel jakýkoliv údaj špatně (například zadá neplatné datum narození, nebo textový údaj do kolonky startovního čísla, nebo další ...) aplikace tuto chybu zachytí pomocí výjimek a následně vypíše chybovou hlášku s informací o chybě a nedovolí takového závodníka přidat do startovní listiny. K přidání závodníka slouží příslušné tlačítko. Dále je zde tlačítko zpět, které se nachází vždy vlevo nahoře a má ikonku šipky doleva. To lze využít, pokud uživatel nechce uložit změny a přidat tak nového závodníka.

### 3.1.6 Zobrazení závodníka ve startovní listině

Pokud uživatel ve startovní listině stiskne některého ze závodníků, zobrazí se nový fragment s informacemi o tomto závodníkovi. Tento fragment je na následujícím obrázku (Obrázek 17).



Obrázek 17: Fragment, který zobrazí informace o závodníkovi

Informace o závodníkovi zahrnují jeho jméno a příjmení, pohlaví, datum narození, kategorii, tým, číslo a případný výsledný čas. Fragment ovšem reaguje na typ organizovaného závodu – pokud se nejedná o závod s hromadným startem, nýbrž o časovku, zobrazí se ještě informace o času startu závodníka (viz Obrázek 18). Tento údaj říká, o kolik později startuje závodník s ohledem na celkový čas startu.

Fragment obsahuje v horní části ještě tři tlačítka, které jsou znázorněny ikonami. Jedná se o tlačítko zpět, tlačítko pro úpravu závodníka a tlačítko pro odstranění závodníka.



Obrázek 18: Fragment, který zobrazí informace o závodníkovi - typ závodu časovka

### 3.1.7 Upravit závodníka ve startovní listině

Při nutnosti upravit některého ze závodníků poslouží fragment, který zobrazuje obrázek (Obrázek 19). Svým rozložením se velmi podobá fragmentu, který se stará o přidání nového závodníka. Rozdíl je v tom, že jsou načteny údaje o závodníkovi a všechna pole jsou již předvyplněná, poté je lze upravit.

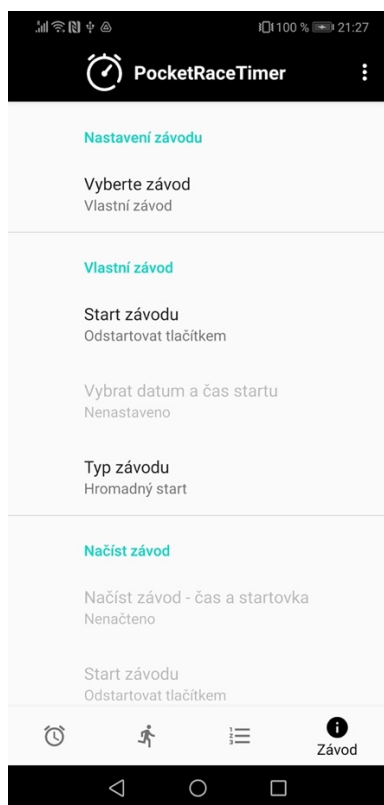


Obrázek 19: Fragment, který upraví závodníka

### 3.1.8 Nastavení

Poslední položkou hlavního menu je položka s názvem „Závod“, ve které se celý závod nastavuje. Tento fragment je na obrázku (Obrázek 20).





Obrázek 20: Fragment nastavení závodu

Fragment s nastavením závodu jako jediný nedědí ze třídy *AppFragment* a to z toho důvodu, že dědí od třídy *PreferenceFragmentCompat*, což je importovaná třída, která se v praxi používá, pokud je potřeba vytvořit nastavení aplikace. Tato třída nevyužívá klasické komponenty jako ostatní fragmenty, ale používá komponenty zvané *Preferences*, které jsou provázané se *SharedPreferences*. *SharedPreferences* je trvalé úložiště, které je schopné uložit data ve formátu klíč-hodnota. K uložení je tedy potřeba nějaký textový klíč, ke kterému je přiřazena hodnota libovolného datového typu. *Preferences* je spojen se *SharedPreferences* tím způsobem, že každý element (tedy položka v nastavení) obsahuje svůj vlastní klíč, který je uložený v úložišti. Při změně nějaké položky v nastavení se tedy v *SharedPreferences* vyhledá hodnota s klíčem této položky a následně se změní na hodnotu novou.

Samotný fragment tedy obsahuje několik elementů typu *Preference*, které mají každý svůj klíč a hodnotu, která je uložena v úložišti. K hodnotám se lze jednoduše dostat a při každé změně se změní i atributy závodu, které jsou uloženy v databázi. Pokud tedy uživatel v nastavení například změní typ závodu ze závodu s hromadným startem na časovku, aplikace si tuto změnu přepíše v databázi.

Důležitou funkcionalitou tohoto fragmentů je také znemožnění některých úprav. Pokud závod probíhá, nebo je již ukončený, všechny položky v nastavení jsou zablokovány, aby nebylo možné měnit parametry již probíhajícího závodu. Dále se také jednotlivé položky zablokovávají v závislosti na jiných zvolených hodnotách, například pokud si uživatel vybere, že chce odstartovat závod tlačítkem, je mu znemožněno nastavovat čas startu.

## 3.2 Komunikace mezi Fragmentem a Activity

Jelikož všechny události vyvolané uživatelem zpracovávají fragmenty, je potřeba předávat data z fragmentu do hlavní *Activity* a opačně. K této komunikaci slouží *interface*, neboli rozhraní. Každý fragment má vytvořený svůj *interface*, který se nazývá *fragment listener*. V tomto rozhraní jsou poté definované metody, které musí implementovat třída, která toto rozhraní implementuje. V praxi to poté vypadá tak, že hlavní *Activity* implementuje rozhraní všech fragmentů a obsahuje všechny příslušné metody. Tyto metody se často opakují, proto není potřeba pro každý fragment psát metody zvlášť, protože ve výsledku si spíše rozhraní fragmentu vybírá metody z hlavní *Activity*, které potřebuje. Pokud tedy poté fragment potřebuje předat nějaká data do *Activity*, zavolá metodu, kterou má definovanou ve svém rozhraní. Jelikož má *Activity* tuto metodu implementovanou, dostanou se potřebná data k ní pomocí argumentů této metody.

Je-li potřeba předat nějaká data z hlavní *Activity* do fragmentu, slouží k tomu návratové hodnoty těchto metod. Fragment si zavolá metodu ze svého rozhraní, která má definované návratové hodnoty dat. *Activity* mu tato data vrátí.

## 3.3 Použité Android prvky

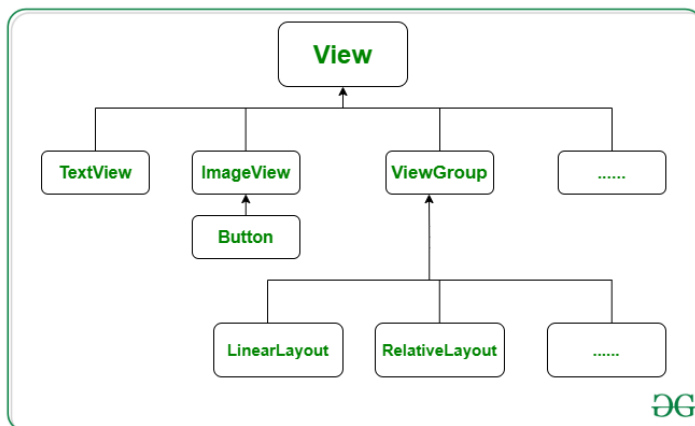
Při tvorbě Android aplikace je nutné kromě základních prvků jazyka Java použít také třídy a objekty, které jsou určeny pro správný chod Android aplikací. Základem jsou prvky *Activity* a *Fragment*, které již byly popsány. *Activity* je důležitým prvkem, bez ní by samotná aplikace vůbec nešla spustit. Stará se totiž o komunikaci s uživatelem. Tato komunikace je zprostředkována pomocí ovládacích prvků jako jsou různá tlačítka, textová pole a jiné. Tyto prvky jsou vždy potomkem tříd *View* nebo *ViewGroup*.

### 3.3.1 View a ViewGroup

Jedná se o třídy, které jsou, hned po *Activity*, základním stavebním kamenem aplikace. Pod pojmem *View* si lze představit každý ovládací prvek aplikace, jsou to tedy všechny

možné textové pole, zaškrtnuté pole, tlačítka, vyhledávací lišty, přepínače, obrázky, spinnery a jiné. Všechny tyto komponenty musí buď přímo, nebo vzdáleně, dědit od třídy *View*. Ta tedy zajišťuje správnou funkčnost těchto komponent. Každé *View* lze nastavit spouštěcí akce, neboli *listenery*. *Listener* je rozhraní, které se používá pro zachycení různých akcí. Příkladem může být *OnClickListener*, což je *listener*, který čeká, až uživatel klikne na danou *View*, a následně vykoná nějakou činnost. U *View* lze nastavovat mnoho dalších vlastností, jednou z nich je například *visibility* (neboli viditelnost), tedy zda bude daná komponenta uživateli zobrazena, nebo bude skryta.

Oproti tomu *ViewGroup* je třída, která je potomkem třídy *View* (Obrázek 21). Liší se v tom, že může obsahovat další jiné *Views*, což z ní dělá základní třídu pro různé rozložení a kontejnery. Třídy, které jsou potomky *ViewGroup* jsou následně používány v aplikaci jako jakési neviditelné kontejnery, které obsahují různé komponenty (potomky *View*). Jejich funkcí je určování vzájemné polohy těchto komponent. [10]



Obrázek 21: *View* a *ViewGroup* [10]

### 3.3.2 *TextView*, *EditText* a *Button*

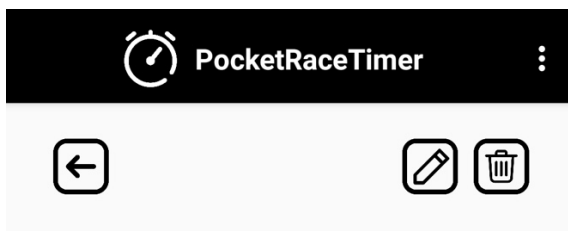
Základními prvky Android aplikací jsou prvky *TextView*, *EditText* a *Button*. Je tedy téměř samozřejmostí, že všechny tyto tři byly použity ve vytvořené aplikaci. *TextView* je komponenta, pomocí které lze vypisovat textové informace. Je využita všude tam, kde je třeba nějaký výpis na obrazovku. Textům lze nastavit různé parametry – velikost, styl, barva písma, a další.

Další komponentou, která stejně jako *TextView* pracuje s textovými řetězci, je *EditText*. Rozdíl mezi těmito dvěma komponentami je ten, že *EditText* reprezentuje pole, do kterého uživatel zadává nějakou textovou informaci. V aplikaci je tato komponenta využita

například na hlavním fragmentu při průběhu závodu. Uživatel do *EditTextu* zadává číslo závodníka, který zrovna dobíhá do cíle, a poté odešle jeho výsledný čas pomocí tlačítka. *EditTextu* je možné nastavovat různé další atributy, v tomto případě byl využit atribut *inputType*, jehož hodnota je nastavena na „number“. Toto nastavení zajistí, že uživatel do daného textové pole může zadat pouze čísla, nikoliv písmena. Dalším atributem, který byl použit, je *hint*, neboli nápověda. Jeho hodnotou je jakýkoliv textový řetězec, který se zobrazí v textovém poli v pozadí šedým písmem jakožto nápověda uživateli, aby věděl, co má do daného pole vyplnit.

Text v poli je možné již přednastavit, při zobrazení dané komponenty uživatel pouze upravuje danou hodnotu. To je využito při upravování závodníků ve startovní listině. Aplikace uživateli zobrazí všechny informace o závodníkovi – tyto informace jsou zobrazeny pomocí *EditTextů*, je tedy možné je upravit a následně odeslat úpravy pomocí tlačítka.

Další velmi často používanou komponentou je *Button*, neboli tlačítko. U tlačítek je vždy důležité nastavit *OnClickListener* a tím definovat, co se stane po jeho zmáčknutí. Tlačítka jsou v aplikaci využívána na mnoha místech. Kromě klasických tlačítek, kterým se nastaví text, styl a barva pozadí, jsou v aplikaci využita také tlačítka, jejichž pozadím je obrázek. Jedná se například o tlačítko „zpět“, které je znázorněno pomocí šipky doleva, dále také tlačítka „upravit“ a „odstranit“. Všechna tyto tlačítka se nacházejí na následujícím obrázku (Obrázek 22).



Obrázek 22: Tlačítka, jejichž pozadím je obrázek

### 3.3.3 Chronometer

Velmi důležitým prvkem v aplikaci je *Chronometer*. Jedná se o třídu, která implementuje jednoduchý časovač. Tento prvek slouží v aplikaci k zobrazení aktuálního času závodu. Kromě velké časomíry na hlavním fragmentu aplikace, se čas uživateli ukazuje i na dalších fragmentech, vždy v horní části displeje. *Chronometer* je možné v Android aplikacích využívat dvěma způsoby:

- První způsob je ten, že *Chronometer* pracuje jako časovač, tedy přijme časový údaj a od tohoto času odčítá směrem dolů, dokud nedojde do hodnoty 0.
- Druhá varianta připomíná klasické stopky – *Chronometer* přičítá čas směrem nahoru. Je možné na začátku předat této komponentě nějaký časový údaj, na kterém bude časomíra začínat, pokud se tak nestane, čas se přičítá od 0.

Mezi těmito variantami lze volit pomocí metody *setCountDown(boolean)*, kterou je nutné zavolat na instanci objektu *Chronometer*. Mobilní aplikace využívá druhou variantu, tedy funkčnost, která se podobá stopkám. Při startu závodu si aplikace uloží do databáze čas startu a spustí *Chronometer*. V případě, že uživatel aplikaci ukončí a později by se chtěl k probíhajícímu závodu znovu vrátit, při načítání aplikace se do *Chronometeru* nastaví aktuální čas závodu. Ten se získá jednoduchým způsobem – aplikace si z databáze načte čas startu závodu a následně tento čas odečte od aktuálního času. K nastavení počátečního času se používá metoda *setBase(long base)*, jejíž parametr je časový údaj typu *long*. Dalšími užitečnými metodami jsou metody *start()* a *stop()*, které spouštějí nebo pozastavují počítání. [11]

Jelikož je *Chronometer* potomkem třídy *View*, lze mu nastavit stejné atributy jako této třídě. Díky tomuto faktu umožňuje aplikace uživateli na hlavním fragmentu zadat probíhajícího závodníka kliknutím na čas závodu. *Chronometeru* je nastaven *onClickListener*, v případě kliknutí na čas se aplikace zachová tak, jakoby uživatel klikl na tlačítko „Ok“ bez vyplněného čísla závodníka, což může být výhodou při dobíhání více závodníků najednou. Nejedná se o žádnou zásadní funkcionalitu (je to pouze zdvojení již implementované funkce), přesto to může posloužit k většímu komfortu při používání aplikace.

### 3.3.4 CountdownTimer

*CountDownTimer* není komponenta, která by dědila od třídy *View*, jedná se pouze o objekt, pomocí kterého lze spustit odpočet a po skončení tohoto odpočtu vykonat nějakou událost. Stejně tak lze vykonávat nějakou funkci v každém kroku odpočtu. V aplikaci je tento objekt využit při odpočítávání času do startu závodu. Pokud uživatel v nastavení zvolí, že chce odstartovat závod automaticky v přesně určený čas, je třeba, aby aplikace hlídala, kdy nastane start. Při zvolení času startu se čas uloží do databáze, takže v pozadí si aplikace start pohlídá, pokud uživatel aplikaci vypne a spustí ji znovu až po startu, závod již bude běžet a bude obsahovat správný časový údaj probíhajícího závodu. *CountDownTimer* ovšem řeší situaci, kdy má uživatel aplikaci zapnutou a potřebuje vědět,

za jak dlouho závod začne. Při zobrazení některého z fragmentů, které obsahují *Chronometer* s časem závodu, se v pozadí vytvoří instance objektu *CountDownTimer*, která hlídá start závodu. Spustí se časovač, který končí přesně v čas startu, v ten moment je spuštěn *Chronometer* s časem závodu. To je docíleno pomocí metody *onFinish*, která se automaticky provede po dokončení odpočtu.

Na hlavním fragmentu aplikace je využita další metoda *CountDownTimeru*, která se nazývá *onTick*. Tato metoda provede zvolenou funkci v každém kroku odpočtu. V případě, že se start závodu blíží, metoda vypisuje na hlavním fragmentu do šedého informativního textu informaci o tom, kolik vteřin zbývá do startu. Informativní text s odpočtem do startu závodu je možné vidět na jednom z výše uvedených obrázků (Obrázek 9). [12]

### 3.3.5 ListView

Pro zobrazení více dat stejného typu slouží prvek zvaný *ListView*. Jedná se o komponent, který zobrazí všechna tyto data ve formě seznamu. Jedná se o potomka třídy *ViewGroup*. *ListView* je v aplikaci využit všude tam, kde jsou zobrazeny seznamy závodníků, to znamená startovní listina, výsledková listina a závodníci v cíli. Tento prvek vyžaduje pro svoji funkci komponent zvaný *Adapter*, který se nastavuje pomocí metody *setAdapter*.

### 3.3.6 Adapter

*Adapter* je komponenta, která se využívá pro zobrazení dat v seznamech, neboli v *ListView*. Jedná se o most mezi kontejnery, která zobrazují data (příkladem je již zmíněný *ListView*), a samotnými daty. *Adapter* definuje, která data se v kontejneru budou zobrazovat a jakým způsobem se budou data zobrazovat. Je tedy nutné předat mu nějaké pole nebo list dat a dále také nějaký *Layout*, neboli rozvržení dat. *Adapter* tyto data zobrazí požadovaným způsobem, každá položka je vnímána jako *View*. Je možné použít již vytvořené adaptéry, které dokáží zobrazit jednoduchá data, jedním z nich je *ArrayAdapter*. Tento adaptér zobrazí data tím způsobem, že na každé položce seznamu zavolá metodu *toString()*, vypíše tedy její textovou reprezentaci.

Pro zobrazení komplexnějších dat je nutné vytvářet si vlastní adaptéry, přesně tímto způsobem je tento prvek využit v aplikaci. Aplikace na mnoha místech zobrazuje uživateli seznam závodníků - buď závodníky v cíli nebo závodníky ve startovní listině.

V kontejneru je tedy potřeba zobrazit data, která jsou datového typu *RacerModel*, což je objekt reprezentující závodníka. Tento objekt je zobrazován více způsoby:

- Ve startovní listě se v seznamu zobrazuje startovní číslo, jméno a příjmení, kategorie a informace o tom, zda závodník již doběhl (případně výsledný čas)
- Ve výsledkové listině se v seznamu zobrazuje výsledná pozice závodníka, jeho jméno a příjmení, startovní číslo, výsledný čas a kategorie, ve které závodil
- Ve Fragmentu, kde se zobrazují všichni závodníci, kteří již doběhli do cíle, je potřeba zobrazit startovní číslo, jméno, příjmení a výsledný čas.
- Pokud je startovní listina závodu prázdná, místo jména, příjmení a čísla závodníka se zobrazí pouze text „Závodník číslo“ doplněný číslem závodníka.

Ve všech případech je tedy potřeba zobrazit data jiným způsobem, k tomu slouží vytváření různých vlastních adaptérů. Všechny tyto vlastní adaptéry musí být potomkem třídy *ArrayAdapter*. Pro implementaci je důležité přepsat metodu *getView*, která se stará o samotné zobrazení, tato metoda vrátí výsledné *View*, tedy data v požadované podobě.

### 3.3.7 Spinner

*Spinner* je komponenta, která je využívána v případě, kdy je potřeba, aby si uživatel vybral jednu hodnotu z pevně daného seznamu. Při kliknutí na *Spinner* se rozbálí menu a uživateli se ukáží všechny položky. Po vybrání jedné z nich je tato položka zobrazována *Spinnerem*.

Na obrázku (Obrázek 23) se nacházejí dva Spinnery, první z nich je nerozbalený u kolonky „Pohlaví:“, druhý je možné vidět v rozbalené variantě u kolonky „Kategorie:“.

The image shows a form with several input fields. At the top, there is a label 'Pohlaví:' followed by a spinner control showing 'Muž'. Below it is a label 'Datum:' followed by three empty input fields. Then, there is a label 'Kategorie:' followed by a spinner control showing a list of options: 'Děti', 'Muži', and 'Ženy', with a '+ Přidat kategorii' option at the bottom. Below the category spinner is a label 'Tým:' followed by an empty input field. At the bottom left is a label 'Číslo:' followed by an empty input field. At the bottom center is a black button with white text that says 'PŘIDAT ZÁVODNÍKA'.

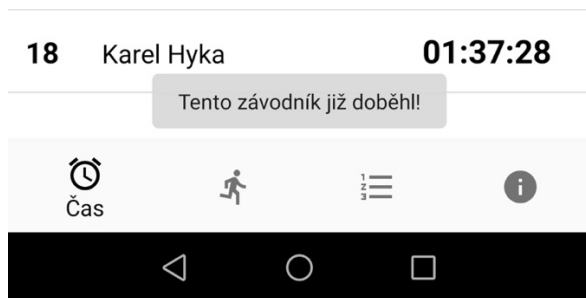
Obrázek 23: Ukázka Spinnerů

Spinnery jsou v aplikaci používány pro filtrování startovní listiny a výsledků podle kategorií, dále také pro vytváření a úpravu závodníků, kdy je třeba zvolit z předem určených kategorií nebo pohlaví. Při jejich implementaci je nutné využít interface *OnItemSelectedListener*, kde je důležitá metoda s názvem *onItemSelectedListener*. Tato metoda je zavolána po každé, když se změní hodnota spinnerů, je tedy možné zde implementovat různé funkce, které jsou požadované po zvolení hodnoty. [13]

### 3.3.8 Toast

*Toast* je komponenta, která je viditelná na obrazovce pouze na určitou dobu. Je často používána pro vypisování informačních nebo chybových zpráv uživateli. Aplikace tuto komponentu využívá pro vypisování chyb, například pokud se uživatel pokouší načíst závod, který neexistuje, nebo pokud zadá číslo závodníka, který už dávno předtím doběhl. Zobrazí se vždy v dolní části obrazovky, jak je vidět na obrázku (Obrázek 24). Je možné určit délku jeho zobrazení, často se využívají předem připravené konstanty „*Toast.LENGTH\_SHORT*“ a „*Toast.LENGTH\_LONG*“, kde první zobrazí *Toast* na kratší dobu (2 s) a druhá na delší dobu (3,5 s). [14]





Obrázek 24: Toast oznamující chybu při zadávání čísla závodníka

### 3.3.9 Dialog

Pod pojmem dialog si lze představit vyskakovací okno. Toto okno se uživateli zobrazí v určitou chvíli a vyžaduje, aby uživatel provedl nějakou událost. Aplikace například využívá tři druhy dialogů:

- Informativní dialogy – tyto dialogy pouze zobrazí uživateli určený text. Jsou využity v nastavení aplikace, kde lze rozkliknout informace o aplikaci, a také při importu startovní listiny, kde se nacházejí tlačítka, které po stisknutí zobrazí uživateli nápovědu, jakým způsobem lze úspěšně importovat data závodníků.
- Potvrzovací dialogy – jedná se o dialogy, kde aplikace vyžaduje, aby uživatel potvrdil svou předešlou volbu. Tyto dialogy jsou žádoucí ve chvíli, kdy volba nenávratně změní stav, nebo smaže některá data aplikace, například pokud si uživatel přeje ukončit závod (již nebude možné dále přidávat závodníky).
- Dialogy zprostředkovávající příjem dat – tyto dialogy obsahují další komponenty jako třeba tlačítka nebo *EditTexty*, vyžaduje se po uživateli, aby předal aplikaci další data. Příkladem je upravování čísla závodníka v cíli – pokud si uživatel přeje upravit číslo závodníka, zobrazí se dialog ve kterém je pole pro zadání čísla, do tohoto pole je nutné zadat nové číslo závodníka a potvrdit změnu.

Při vytváření dialogů je vždy použita třída *AppCompatActivity*, kdy vytvořený dialog je potomkem této třídy. K sestavení dialogu se používá objekt *AlertDialog* ze třídy *AlertDialog*. Tento objekt obsahuje již připravené metody pro nastavení nadpisu dialogu a tlačítek pro potvrzení nebo zrušení změn. Komunikace mezi dialogem a fragment je zajištěna stejným způsobem, jako mezi fragmentem a *Activity*, tedy za pomoci *interface*, neboli rozhraní.

### 3.3.10 Options menu

*Options menu* je menu, které se nachází na obrazovce vpravo nahoře, pod symbolem tři teček. V tomto menu je možné ukončit závod, nebo začít zcela nový závod, menu je vidět na obrázku (Obrázek 7). Obě volby mají zásadní dopad na aktuální stav aplikace. *Options menu* lze přidat do *Activity* nebo *Fragmentu* pomocí metody *setHasOptionsMenu(true)*, následně jsou pro práci s menu důležité tři metody:

- Metoda *onCreateOptionsMenu* – jedná se o metodu, která vytváří *Options menu*. Této metodě je v parametru předána instance objektu *MenuInflater*, které je poté nutné předat zdrojový XML soubor, ve kterém je definována podoba menu.
  - Metoda *onPrepareOptionsMenu* – zde se v aplikaci nastavuje, které položky menu jsou uživateli přístupné. Například pokud závod ještě nezačal, nebo je již ukončený, nelze kliknout na položku „Ukončit závod“.
  - Metoda *onOptionsItemSelected* – v této metodě je obstarána funkcionality jednotlivých voleb. Je zde definováno, co se stane po kliknutí na danou položku menu.
- [15]

### 3.3.11 LinearLayout, ScrollView

Dalšími komponentami, které jsou stejně jako *ListView* potomky *ViewGroup* jsou *LinearLayout* a *ScrollView*. Jedná se o prvky, které spadají do skupiny kontejnerů, obsahují tedy další *Views*. *LinearLayout* se stará o zarovnání všech *Views* v jednom směru – horizontálně, nebo vodorovně. *ScrollView* je prvek, který umožňuje uživateli scrollovat komponentou, která je umístěna uvnitř tohoto prvku. Komponenta tedy může mít jakoukoliv velikost, uživatel si ji může vždy prohlédnout celou. Do *ScrollView* lze vždy umístit pouze jedno *View*, může jím být ovšem některý *Layout* (například *LinearLayout*), který obsahuje těchto *Views* více. *LinearLayout* i *ScrollView* jsou velice často využívané pro určení pozic tlačítek, textů a jiných komponent.

## 3.4 Databáze

Aplikace využívá *SharedPreferences* k nastavování klíčových vlastností závodů. Tyto *SharedPreferences* ale nejsou jediným trvalým úložištěm, které je využíváno v aplikaci. Data všech závodníků, kategorií a samotná data závodu jsou uložena v SQLite databázi. Tato databáze je součástí operačního systému Android, je uložena přímo v paměti zařízení a velkou výhodou je, že přístup k jejím datům má vždy pouze aplikace, která tyto

data vytvořila. K implementaci databáze v rámci aplikace byla využita třída *SQLiteOpenHelper*, která byla vytvořena přesně pro tyto účely. Jedná se o pomocnou třídu pro vytváření a správu databáze SQLite. Třída *DAO*, která se v aplikaci stará o práci s databází, tedy dědí od třídy *SQLiteOpenHelper*. Třída *DAO* funguje následovně:

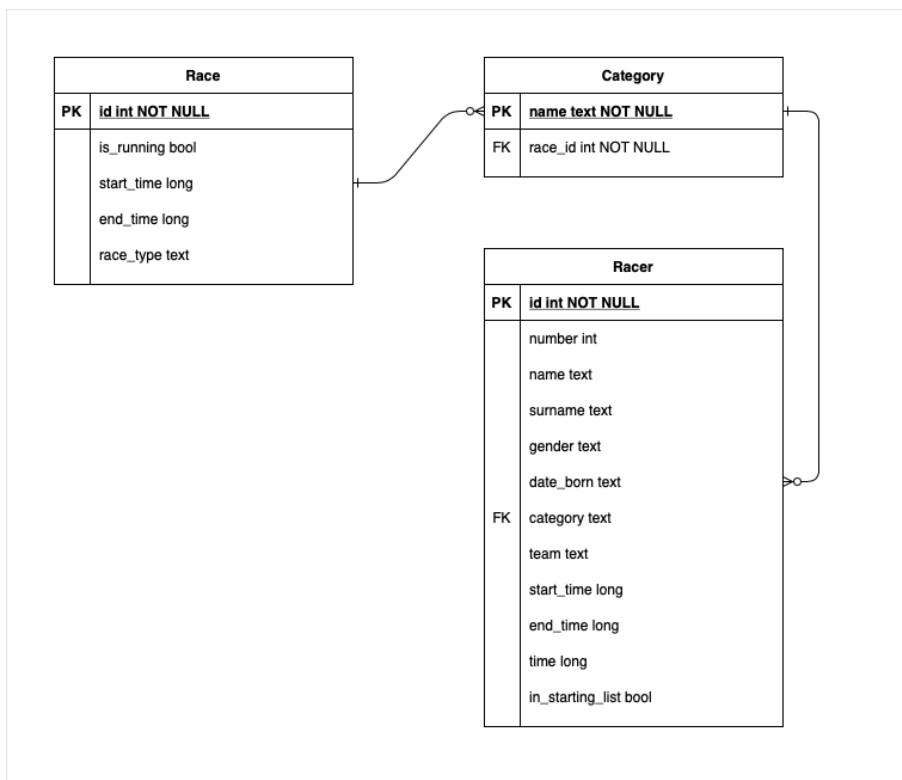
- Při vytvoření, tedy v konstruktoru zavolá konstruktor třídy *SQLiteOpenHelper* a předá mu kontext, jméno databáze a verzi databáze. Tím se inicializuje práce s touto databází
- Pokud otevíraná databáze ještě neexistuje, zavolá se metoda s názvem *onCreate*. Tato metoda je implementována v samotné třídě *DAO* a jsou zde příkazy pro vytvoření samotných databázových tabulek.
- Následně už třída *DAO* obsahuje různé potřebné metody pro úpravu dat v databázi, jako jsou například přidávání závodníků do databáze, úprava času startu závodu, a mnoho dalších. Všechny tyto metody jsou využívány při práci s daty a jsou volány z různých míst aplikace, vždy prostřednictvím hlavní *Activity*, která obsahuje instanci třídy *DAO*.

#### 3.4.1 Popis databázových tabulek

Databáze obsahuje následující entity:

- *Racer* – tato entita reprezentuje závodníka, jsou v ní uloženy všechny potřebné údaje o závodníkovi – id, startovní číslo, jméno a příjmení, pohlaví, datum narození, kategorie, tým, čas startu, čas doběhu, výsledný čas a údaj, zda se závodník nachází ve startovní listině
- *Race* – tato entita ukládá všechny důležitá data závodu, těmi jsou čas startu, čas konce závodu, typ závodu a údaj, zda závod právě probíhá
- *Category* – tato tabulka obsahuje názvy všech kategorií závodu

Na následujícím obrázku se nachází entitně-relační diagram (Obrázek 25).



Obrázek 25: Entitně-relační model databáze

### 3.5 Import startovní listiny

Jak již bylo několikrát zmíněno, do aplikace lze importovat startovní listinu ze serveru SportChallenge nebo z CSV souboru. Oba tyto importy lze provádět prostřednictvím fragmentu se startovní listinou, při stisknutí tlačítka „Importovat startovku“ se uživateli zobrazí dialog, který je vidět na následujícím obrázku (Obrázek 26).

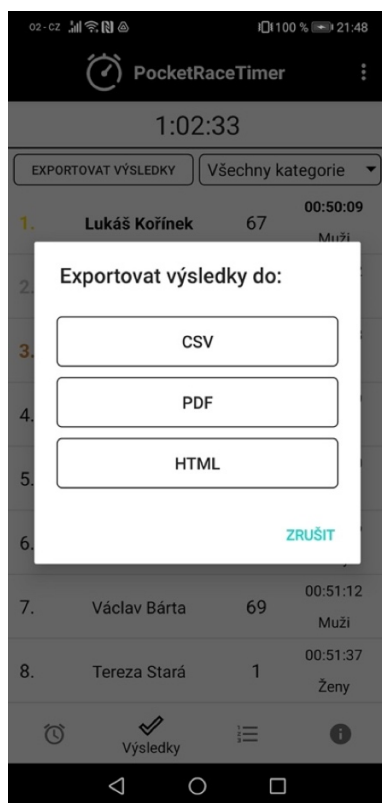


Obrázek 26: Možnosti importu startovní listiny

Kromě dvou možností importu jsou v dialogu také tlačítka se symbolem otazníku, ty slouží pro zobrazení nápovědy. Pokud tedy uživatel stiskne například tlačítko pro nápo- vědu u možnosti importu z CSV souboru, zobrazí se mu informace o tom, jak by měl soubor vypadat a která všechna data by měl obsahovat. Uživatel si poté může takovýto soubor připravit, uložit si ho kdekoli ve svém mobilním zařízení a následně importovat data do aplikace. Pro čtení souboru je využit objekt *BufferedReader*, kterému je pomocí argumentů předaný vstupní soubor. Tento soubor si vybere uživatel ze svého zařízení, k tomu je použitý objekt zvaný *Intent*, který se pro takovéto účely běžně používá. Obrá- zek vzorového CSV souboru určeného pro import se nachází v příloze (Příloha C: Import startovní listiny – vzorový CSV soubor).

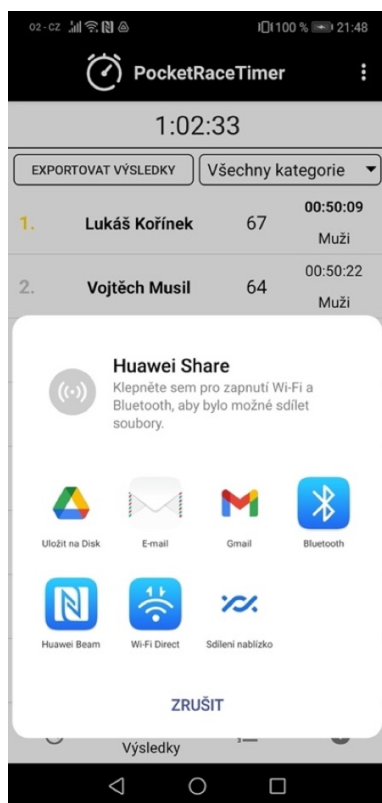
### 3.6 Export výsledků

Potřebuje-li uživatel sdílet výsledky závodu, stačí po ukončení závodu stisknout tlačítko „Exportovat výsledky“. Následně se mu zobrazí vyskakovací okno, které nabízí volbu formátu souboru s výsledky. Na výběr jsou formáty CSV, PDF a HTML (viz Obrázek 27).



Obrázek 27: Export výsledků

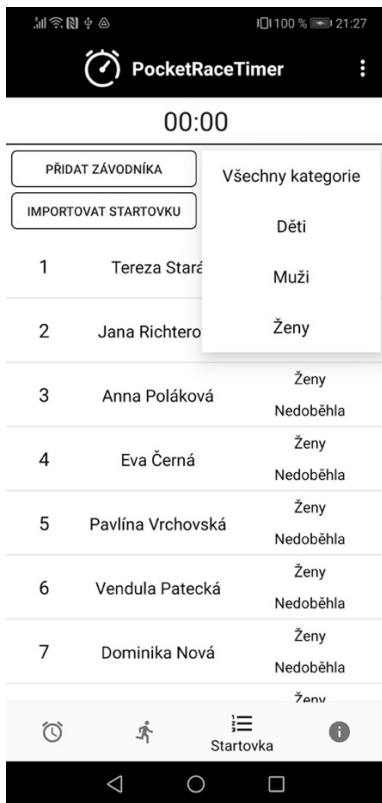
Soubory jsou vytvářené za pomoci objektu *FileOutputStream*, který se stará o vytvoření souboru a zápis dat. Po zvolení formátu je možné soubor sdílet s jinými aplikacemi v telefonu, které má uživatel nainstalované (viz Obrázek 28). Výsledky je tak možné například odeslat emailem, nebo uložit na Google Disk. Příklady vytvořených souborů se nacházejí v přílohách (Příloha D: Export výsledků do CSV souboru – příklad, Příloha E: Export výsledků do PDF souboru – příklad a Příloha F: Export výsledků do HTML souboru – příklad).



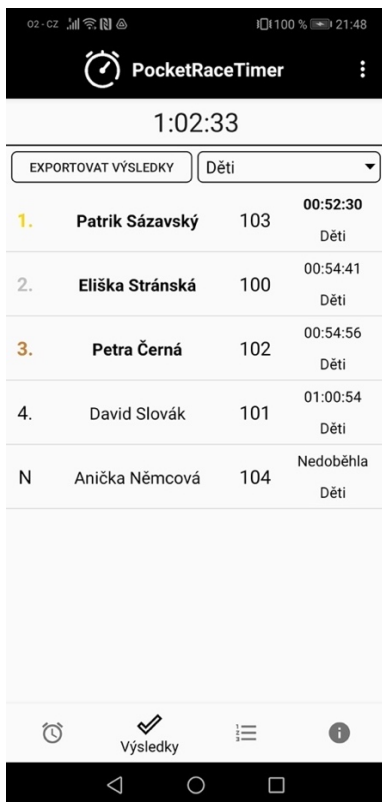
Obrázek 28: Sdílení výsledků

### 3.7 Kategorie závodu

Aplikace je schopna měřit pouze jeden závod, ovšem ten může obsahovat více kategorií. Kategorie se využívají při filtraci startovní listiny (viz Obrázek 29) a filtraci výsledků. Filtrované výsledky obsahují pouze závodníky z dané kategorie s aktuálním pořadím v kategorii (viz Obrázek 30).



Obrázek 29: Možnost filtrování startovní listiny podle kategorií



Obrázek 30: Výsledky závodu filtrované podle kategorie



Existují dva způsoby, jak do závodu přidat novou kategorii. První způsob je ten, že uživatel importuje startovní listinu, kde jsou u závodníků kategorie vyplněné. Pokud se u závodníka objeví kategorie, kterou daný závod ještě neobsahuje, automaticky se do závodu přidá.

Druhým způsobem, jak lze vytvořit novou kategorii, je přidat nového závodníka do startovní listiny přímo v aplikaci. Při přidávání závodníka je potřeba zvolit kategorii, které se nachází ve spinneru. Položkami tohoto spinneru jsou všechny existující kategorie závodu, na konec je ale vždy přidána položka „+ Přidat kategorii“ (viz Obrázek 23).

Po zvolení této položky se uživateli zobrazí vyskakovací okno, kde uživatel napíše název nové kategorie, která je závodníkovi přidělena.

Názvy kategorií už později v aplikaci nelze změnit. Jediným způsobem jak lze toto provést, je vytvořit kategorii novou a do ní přesunout všechny požadované závodníky. Stará kategorie se poté odstraní automaticky. Aplikace při každé změně kategorie u kteréhokoliv závodníka zkontroluje, zda se ve staré kategorii ještě nachází nějaký jiný závodník. Pokud zjistí, že se jednalo o jediného závodníka, který byl přiřazený do této kategorie, tak ji automaticky smaže. Uživatel tak toto vůbec nemusí řešit, aplikace se sama postará, aby závod neobsahoval prázdné a zbytečné kategorie.

## 3.8 Návod k použití

Použití aplikace je v mnoha případech jednoduché a intuitivní, velká část funkcionalit byla již popsána. Důležité je ovšem správné nastavení parametrů závodu.

### 3.8.1 Nastavení závodu

Po spuštění aplikace, je třeba nejprve nastavit závod, který chce uživatel měřit. Proto je třeba v hlavním menu ve spodní části obrazovky zvolit poslední položku „Závod“ a aplikace přesune uživatele do nastavení, které je vidět na jednom z předešlých obrázků (Obrázek 20). V nastavení je nejprve potřeba určit, zda se jedná o vlastní závod, který není nikde zaregistrovaný, nebo zdali chce uživatel naměřit závod, který je registrovaný na Sportchallenge.

### 3.8.2 Nastavení vlastního závodu

Pokud se jedná o vlastní závod, je nutné vybrat položku „Vlastní závod“. Poté se zpřístupní skupina nastavení vlastního závodu, naopak skupina nastavení načteného závodu

se uživateli zablokuje, aby bylo vidět, co je třeba dále nastavovat. Ve skupině nastavení vlastního závodu se nacházejí další tři položky nastavení závodu, jedná se o „Start závodu“, „Vybrat datum a čas startu“ a „Typ závodu“. Zjednodušeně lze říci, že tedy uživatel musí nastavit způsob startu a typ závodu.

U způsobu startu lze zvolit, zda bude závod odstartovaný tlačítkem „Start“ nebo zda bude start závodu v přesně zvolený čas a datum. Na této volbě závisí i možnost volby času startu. Pokud si uživatel přeje odstartovat závod tlačítkem, toto nastavení se zablokuje, naopak pokud chce odstartovat závod v přesně zvolený čas, je nutné následně provést nastavení dne a času.

Poslední položkou nastavení je typ závodu, kde se jedná o jednoduchou volbu. Závod může být buďto závod s hromadným startem, nebo časovka. Při hromadném startu vybíhají všichni závodníci v momentu startu závodu, to znamená že cílový čas závodníka je zároveň i jeho výsledným časem. Druhou možností typu závodu je časovka. Při tomto typu vybíhá každý závodník v jiný moment a má předem dané zpoždění na startu. Při doběhnutí závodníka je od jeho cílového času odečteno zpoždění jeho startu a tím se získá výsledný čas.

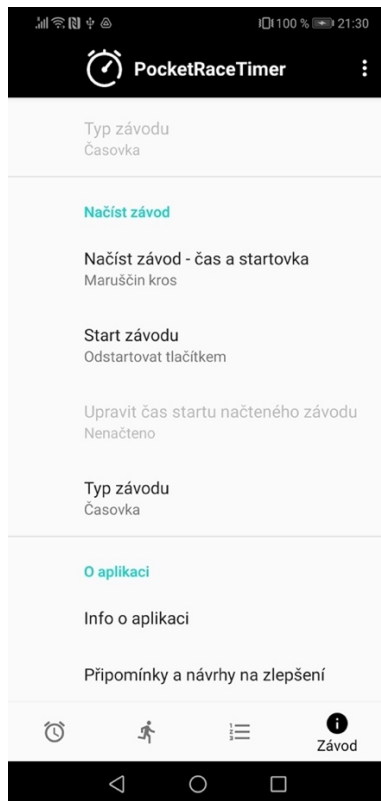
U časovky je tedy potřeba mít připravenou startovní listinu, kde je u každého závodníka zvolený čas jeho startu. Tuto listinu lze do aplikace importovat z CSV souboru a to tak, že uživatel přejde v menu na obrazovku startovní listiny a stiskne tlačítko „Importovat startovní listinu“. Další možností je přidat každého závodníka do startovní listiny v aplikaci zvlášť, skrze tlačítko „Přidat závodníka“. Závod bude fungovat i v případě, že bude startovní listina prázdná, v tomto případě ale nebude mít volba časovky smysl, jelikož u závodníků, kteří se nenachází ve startovní listině se počítá, jakoby vybíhali přesně v momentě začátku závodu, nemají tedy žádné zpoždění na startu. Ve výsledku se tedy stejně jedná o závod s hromadným startem.

### 3.8.3 Nastavení načteného závodu

V případě, že chce uživatel změřit závod, který je registrovaný na Sportchallenge, je třeba znát identifikační číslo paketu daného závodu. V kategorii nastavení „Načíst závod“ se toto číslo zadá do první položky nastavení. Následně se aplikace pokusí tento závod načíst. Při úspěšném načtení se zobrazí jméno závodu, jak je vidět na obrázku (Obrázek 31), a uloží se startovní listina tohoto závodu. Dále se načte typ závodu a čas startu, oba tyto údaje se projeví v nastavení závodu. V případě, že by čas startu načteného závodu byl

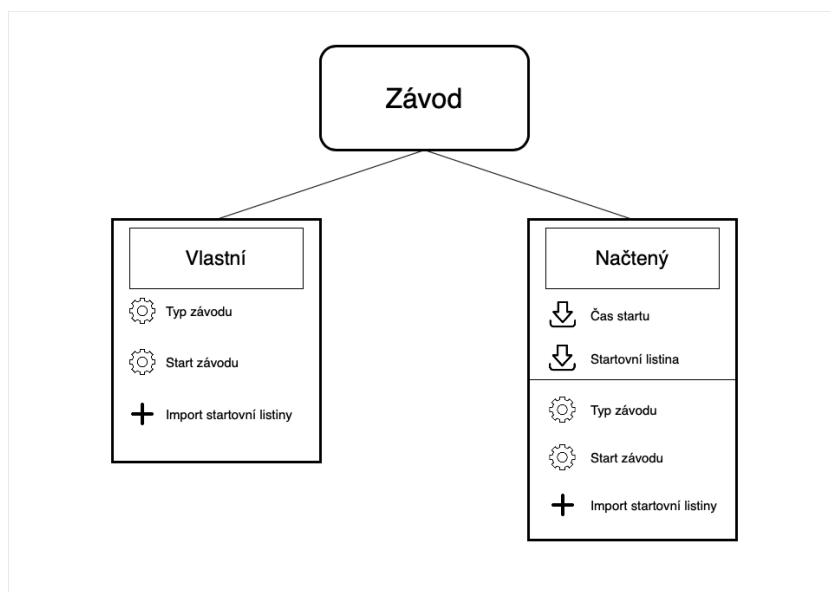
neplatný, nebo by se jednalo o závod, který podle data již měl proběhnout, tak se jako způsob startu nastaví odstartování pomocí tlačítka.

Po úspěšném načtení lze stále změnit typ závodu a způsob startu. Stejně tak lze přidávat a odebírat závodníky ze startovní listiny. Uživateli je také umožněno importovat další závodníky z CSV souboru a přidat je tak do startovní listiny ke stávajícím závodníkům.



Obrázek 31: Úspěšně načtený závod

Základní schéma nastavení závodu jsou znázorněny na následujícím obrázku (Obrázek 32).



Obrázek 32: Schéma možností nastavení závodu

### 3.9 Zveřejnění aplikace

Po úspěšném dokončení byla aplikace *PocketRaceTimer* zadána k publikaci na Google Play. Pro samotnou publikaci na této platformě je třeba zařídit několik věcí:

- Vývojář musí mít vytvořený vývojářský Google účet. Pro úspěšné vytvoření a následnou publikaci aplikací je nutné zaplatit jednorázový poplatek ve výši 25 dolarů.
- Při publikování aplikace je třeba vyplnit podrobnosti o aplikaci, mezi které patří název, krátký popis a úplný popis.
- Dále je třeba vytvořit grafiku aplikace, zde je třeba nahrát logo aplikace a následně také hlavní grafiku, což může být jakýkoliv propagující obrázek, který má velikost  $1024 \times 500$  pixelů.
- Je třeba zveřejnit 2 – 8 snímků obrazovky telefonu, na kterém je spuštěná aplikace.
- Je nutné definovat země a oblasti, ve kterých bude aplikace dostupná a také podporované jazyky.
- Je nutné vyplnit dotazníky týkající se obsahu aplikace, cílového publika a reklam.

Všechny tyto náležitosti byly splněny. Byla vytvořena hlavní grafika aplikace, která se nachází na následujícím obrázku (Obrázek 33). Jakožto země a oblasti, ve kterých je aplikace *PocketRaceTimer* dostupná byla vybrána Česká Republika a Slovensko,

podporovaným jazykem je pouze čeština. Dne 14. 05. 2021 bylo zahájeno plné vydání aplikace, následně je nutné, aby aplikace prošla všemi kontrolami společnosti Google. Tyto kontroly mohou trvat až 7 dnů. Všem uživatelům by tak mohla být aplikace dostupná přibližně 21. 05. 2021. [16]



*Obrázek 33: Hlavní grafika aplikace PocketRaceTimer*

## 4 Testování aplikace

Testování aplikace proběhlo 28. dubna 2021 pod hlavičkou tréninkového běžeckého závodu s názvem „Maruščin kros“. Tohoto závodu se zúčastnili běžci ze sportovních klubů Slavia Liberec Orienteering a AC Slovan Liberec. Celkově měl závod 22 účastníků, kteří měli na výběr ze dvou běžeckých tras:

- Dlouhá trať, která měřila 5,6 km
- Krátká trať, určená především pro děti do 12 let, délka 4,2 km

Delší trať si zvolilo 18 běžců, na menší se vydali 4 závodníci. Jednalo se o závod, kdy běžci vybíhali v různých časových intervalech. Měření proběhlo třemi způsoby. Oficiální měření zprostředkovali zástupci Sportchallenge – Ing. Jan Kolaja, Ph.D., a Ing. Jana Kolaja Ehlerová, Ph.D., za pomoci čipové časomíry. Použity byly čipy SIAC Air od značky SportIdent (zkráceně SI čipy). Dále byly využívány krabičky BS11, také od společnosti SportIdent. Měření probíhá následujícím způsobem:

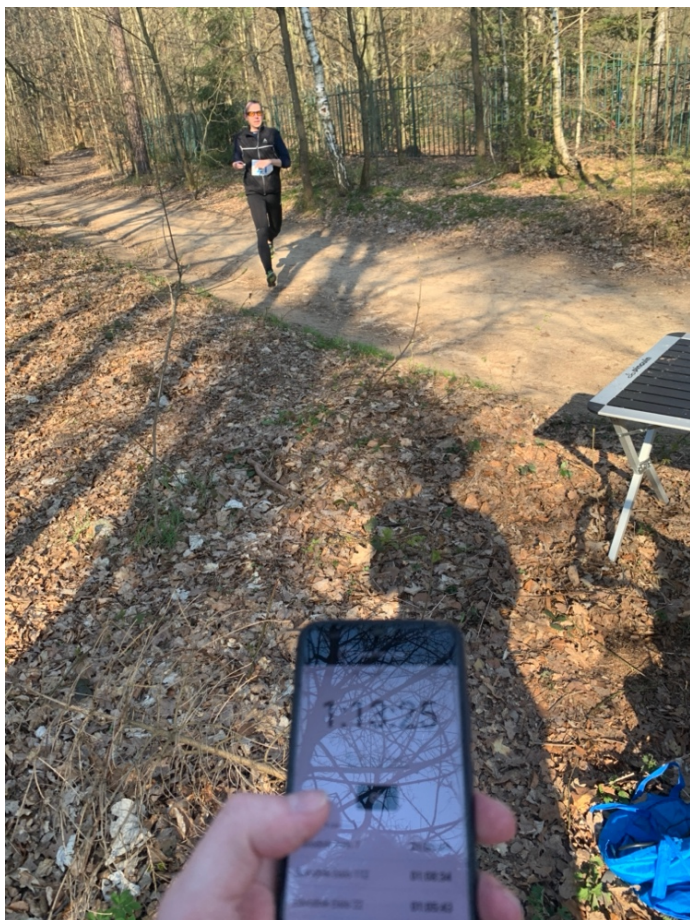
- Závodník si nese svůj SI čip, který má připnutý na prstu
- Jakmile se ocitne 3 metry od měřicí krabičky, čip zapípá a načte čas z krabičky
- Následně jsou po závodě z čipů přečteny a uloženy časové informace

Výměna dat mezi krabičkou a čipem trvá pouhých 60 ms, měření je tedy poměrně přesné. [18] V rámci tohoto měření byl běžcům změřený cílový čas a mezičas (ten byl měřený pouze na dlouhé trati).

Druhé měření závodu provedl Bc. Petr Novák v rámci testování své diplomové práce, jednalo se o měření za pomoci RFID čipů, které měli závodníci připnuté na startovním čísle. Tyto čipy byly poté v cíli zachyceny za pomoci RFID readeru a antény. V tomto typu měření nebyl závodníkům měřený mezičas, pouze výsledný čas.

Třetí způsob měření byl pomocí vytvořené mobilní aplikace. Tato aplikace běžela na dvou mobilních zařízeních současně, jedno zařízení měřilo v místě, kde byly zaznamenávány mezičasy, druhé přímo v cíli. Na obou byl nastavený automatický start závodu v 16:30, tím byla zajištěna kompatibilita časových údajů.

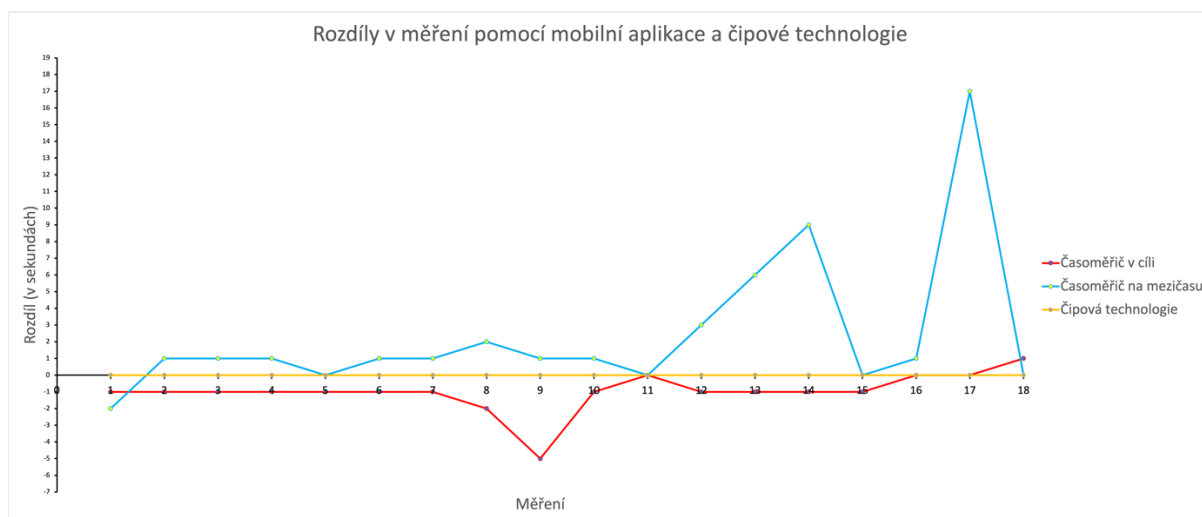
Startovní listina, a tedy i časy startů závodníků, byla načtena ze serveru Sportchallenge, kam byli závodníci v průběhu startů postupně přidáváni. Běžci vybíhali v různých časových intervalech, často i více závodníků najednou.



Obrázek 34: Testování měření běžeckých závodů pomocí vytvořené mobilní aplikace

#### 4.1 Výsledek testování

Měření pomocí mobilní aplikace proběhlo bez větších problémů, všichni závodníci byli zaznamenáni, ke všem byl na obou mobilních zařízeních přiřazen mezičas/cílový čas. Naměřené časy byly následně porovnávány s oficiálními časy, naměřenými pomocí SI čipů. Na obrázku (Obrázek 35) se nachází graf, který znázorňuje rozdíly v naměřených časech oběma časoměřiči, jedná se pouze o závodníky, kteří běželi dlouhou trať závodu.



Obrázek 35: Graf rozdílů v měření

Z grafu je zřetelné, že přesnost měření je závislá na osobě, která tuto aplikaci používá. Časoměřič, který měřil časy v cíli, má průměrnou odchylku od oficiálních časů 1,1 sekundy a většinou zadal závodníka do aplikace dříve, než proběhl kolem readeru. Druhý časoměřič, který měřil mezičasy závodníků, oproti tomu ve většině případů zadal závodníky do aplikace o chvíli později, než byli načtení čtečkou. Jeho průměrná odchylka je 2,6 sekundy, to je ovšem způsobeno dvěma velkými výkyvy – rozdíly 9 a 17 vteřin. Při vynechání těchto extrémních hodnot je jeho průměrná odchylka pouhých 1,3 sekundy. Nejčastější rozdíl v měření byl  $\pm 1$  vteřina. Testování prokázalo, že větší rozdíly od skutečného času závodníka způsobují dvě příčiny:

- nepozornost časoměřiče
- opomenutí odeslání výsledku po napsání čísla

Obě příčiny tedy nejsou způsobeny špatnou implementací aplikace.

Testování ovšem prokázalo také jednu výhodu měření závodu pomocí mobilní aplikace, nastala totiž situace, kdy závodnice, která běžela krátkou trať, doběhla do cíle, ovšem čtečka v momentě protnutí cílové čáry nezaregistrovala její SI čip. Bylo to způsobeno zmatením závodnice a nedodržením technické procedury. Zaregistrování závodnice čtečkou proběhlo až mnohem později, kdy už byla dávno v cíli. Její výsledný čas v mobilní aplikaci byl ovšem zadaný ve správný moment, a tak se lišil o celé 2 minuty a 26 vteřin. Čas závodnice v mobilní aplikaci tedy lze v tomto případě považovat za správný.



## 4.2 UX, neboli uživatelský prožitek

Při testování byly objeveny dva menší nedostatky týkající se uživatelského prožitku při interakci s aplikací. Prvním z nich byl fakt, že po vyplnění čísla probíhajícího závodníka a následném stisknutí tlačítka „Ok“ se kolonka pro číslo závodníka, která je vidět na jednom z předešlých obrázků (Obrázek 10), automaticky nesmazala a zůstalo v ní předchozí číslo. Bylo tedy nutné ji vždy smazat ručně, což není žádoucí, zvláště pokud je potřeba rychle zadávat probíhající běžce.

Druhým nedostatkem, nebo spíše návrhem na vylepšení, bylo, že časoměřič nikde v aplikaci neměl informaci o tom, kolik běžců již doběhlo. Jedinou možností bylo spočítat všechny závodníky, kteří měli v aplikaci cílový čas.

Obě tyto námitky byly vyřešeny, případně opraveny. Bylo doimplementováno automatické mazání kolonky a také byla přidána informace o počtu zadaných závodníků do aplikace, tato informace se nyní nachází na druhém fragmentu aplikace, nad všemi závodníky, kteří jsou již v cíli. Je možné ji spatřit na jednom z předešlých obrázků (Obrázek 13).

## 5 Možnosti rozšíření

Kromě drobných chyb a nedostatků aplikace, které byly všechny v průběhu práce opraveny, se naskytlo ještě pár dalších nápadů na možná rozšíření této bakalářské práce, která by případně vylepšovala aktuální aplikaci, nebo by ji nějakým způsobem doplňovala.

### 5.1 Návrhy na vylepšení aplikace

Co se týče aplikace samotné, tak zde se naskytuje možnost implementovat verzi pro Apple zařízení. Aplikace je dostupná pouze pro platformu Android, pokud by ovšem byla vytvořena a nabídnuta také vlastníkům mobilních zařízení s operačním systémem iOS, mohl by ji používat téměř každý, jelikož tyto dva operační systémy jsou nejrozšířenějšími v České Republice.

Dalším vylepšením by mohla být podpora více jazyků, v současné době aplikace pracuje pouze v českém jazyce. Na tuto událost je ovšem připravena, všechny textové řetězce, které se objevují ve výpisech, jsou uloženy v samostatném souboru s textovými řetězci s názvem *string.xml*. V tomto souboru jsou uloženy textové řetězce tím způsobem, že každý text má svůj atribut *name*. V případě, že by byla žádaná podpora více jazyků, stačí pouze do projektu přidat novou složku do složky *resources*, která obsahuje zdrojové texty a soubory aplikace. Nová složka vždy musí mít následující podobu názvu:

```
< resource type > -b+< language code > [+< country code >].
```

Pokud je tedy potřeba přidat nový soubor s textovými řetězci, které jsou například ve španělském jazyce, je potřeba přidat složku s názvem „*values-b+es*“ a do ní vložit nový soubor *string.xml*. Dále už je potřeba vzít všechny textové řetězce z původního souboru *string.xml* a následně je přeložit do španělského jazyka. Důležité ovšem je, aby zůstal stejný atribut *name*, protože pomocí tohoto atributu se k textovým řetězcům přistupuje z kódu. Po této úpravě už bude nový jazyk dostupný. Aplikace vždy použije jazyk, který je nastavený jako výchozí v mobilním zařízení. [17]

Jedním z dalších návrhů na vylepšení je podpora měření jednoho závodu pomocí více mobilních zařízení, tedy jejich propojení. Aplikace zvládne měřit závod pouze na jednom mobilním zařízení. Při závodech, které mají větší počet účastníků, kteří dohánějí často krátce za sebou, by se časoměřiči mohlo hodit, že by měl u sebe dalšího člověka, kterému by v mobilní aplikaci běžel stejný závod a mohl by také zadávat probíhající závodníky.

Takováto úprava by přispěla k větší přesnosti měření. Bylo by poté možné provádět různé druhy měření pomocí této aplikace, například by každý měřil jinou kategorii nebo by jeden časoměřič mohl klikat pouze časy přesně ve chvíli, kdy závodníci protínají cílovou pásku, druhý by poté k těmto časům přiřazoval čísla závodníků.

Poslední možnou úpravou je měření mezičasu. Aplikace neumožňuje měřit mezičasy, jediným způsobem jak toho momentálně dosáhnout, je nastavit si na dalším mobilním zařízení závod se stejnými parametry a případně i stejnou startovní listinou. Poté lze v místě mezičasu zadávat závodníky do aplikace tím způsobem, jako by se v daném místě nacházel cíl. Přesně tento způsob byl použit při testování aplikace. Nevýhodou je, že data nejsou sjednocena na jednom místě a také je potřeba více mobilních zařízení, tedy i více časoměřičů.

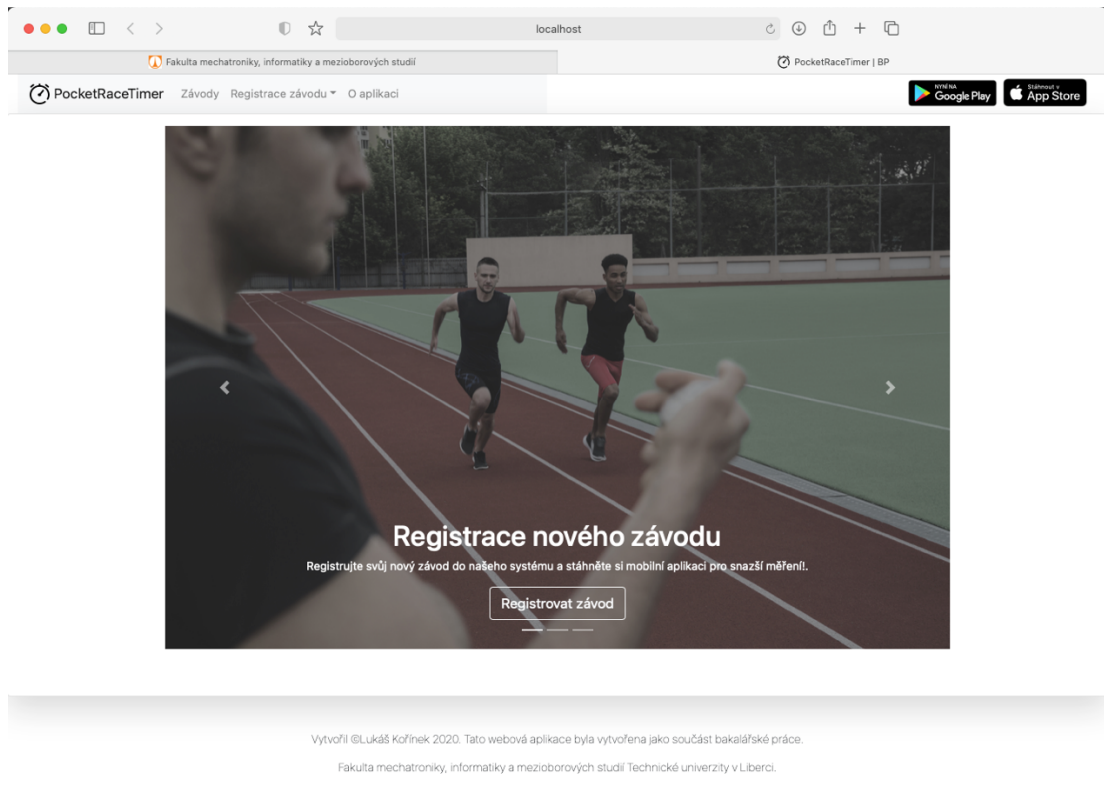
## 5.2 Webová aplikace

Mobilní aplikace má spoustu funkcí, nelze se v ní ovšem postarat o administraci závodu. Je vytvořena tak, že načte startovní listinu závodníků, a to buď z CSV souboru nebo ze serveru Sportchallenge. Pokud organizátor chce mít měření závodu plně ve vlastní režii, jedinou možností, jak do aplikace dostat startovní listinu, je vytvořit CSV soubor, který má všechny potřebné náležitosti, a ten poté nahrát do aplikace. Alternativou je přidávat závodníky do startovní listiny postupně po jednom přímo v aplikaci.

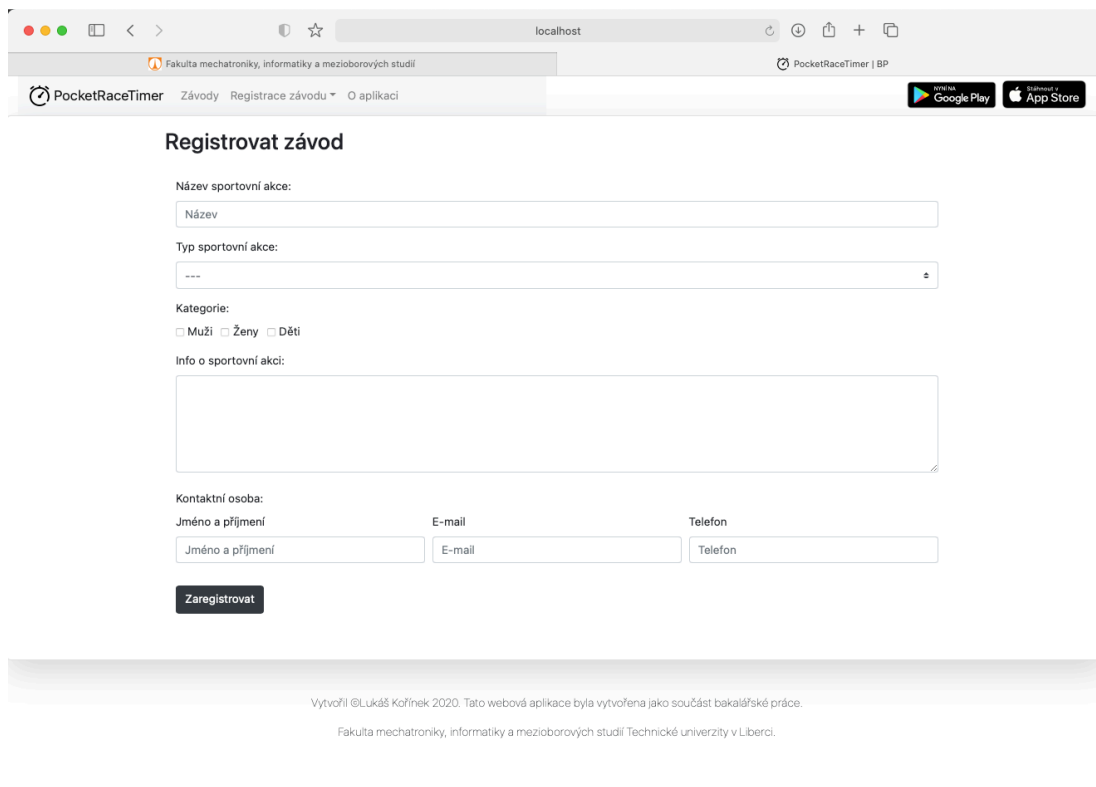
O sestavení startovní listiny se v obou případech musí postarat samotný organizátor. Nabízí se tedy možnost vytvořit jednoduchou webovou aplikaci, která by organizátorům umožňovala vytvořit si svůj vlastní závod, na který by se účastníci přihlašovali online. Všechny informace by se zprostředkovaly přes tuto webovou aplikaci. Závodníkům by byla přiřazena startovní čísla a organizátor by si z tohoto webu pouze před závodem načtl všechna data podobným způsobem, jako se načítají ze serveru Sportchallenge. Měl by tedy vytvořenou startovní listinu automaticky.

Takováto webová aplikace byla v průběhu práce vytvářena. Za pomoci serverového javascriptu Node.js a javascriptové knihovny React byla vytvořena základní aplikace, která zatím nesplňovala požadované funkcionality. Její implementace nakonec nebyla dokončena. Rozhodnutí nepokračovat v jejím vývoji bylo učiněno z toho důvodu, že není jisté, jestli by byla pro organizátory natolik přínosná, aby se vyplatilo udržovat její provoz. Proto bylo při práci vloženo více času a energie do implementace mobilní aplikace.

Ukázka vytvořené aplikace se nachází na následujících obrázcích (Obrázek 36 a Obrázek 37).



Obrázek 36: Návrh webové aplikace - hlavní stránka



Obrázek 37: Návrh webové aplikace – registrace závodu

## Závěr

V rámci této bakalářské práce byl proveden průzkum aktuální situace měření běžeckých závodů. Z této analýzy bylo zjištěno, že neexistuje žádná aplikace pro mobilní zařízení, kterou by si uživatel mohl snadno, rychle a zdarma stáhnout na Google Play a následně by s její pomocí mohl spolehlivě změřit svůj závod. Organizátorům zbývají dvě základní možnosti – čipová technologie a ruční měření. Měřit závod pomocí čipové technologie může být problém pro menší závody, protože nemusí mít potřebné vybavení. V takovém případě je třeba využít nákladné služby, které se o takovéto měření postarají. Druhá možnost, tedy měření závodu ručně, je často velmi zdlouhavá a pracná.

Pro tyto účely byla v rámci práce vytvořena mobilní aplikace, pomocí které lze změřit širší škálu sportovních akcí. Vhodná je zejména pro běžecké závody, lze ji ovšem použít na jiné sportovní události, například cyklistické závody. Velkou výhodou je, že zvládne změřit jak závody s hromadným startem, tak se starty jednotlivých závodníků v různých časových intervalech, zvládne tedy pracovat i se startovní listinou závodu. Důležitým faktorem je, že aplikace je organizátorům dostupná zcela zdarma. Při implementaci byl kladen důraz na jednoduchost a přehlednost aplikace, jelikož závody často organizují i netechnicky založení uživatelé, proto bylo třeba aby byla aplikace vhodná i pro ně.

Z důvodu situace ohledně pandemie koronaviru byla mobilní aplikace otestována pouze na menším tréninkovém závodě a porovnána s měřením pomocí čipové technologie. Při testování byly odhaleny pouze drobné nedostatky týkající se funkčnosti ovládacích prvků. Samotná aplikace obstála a v ostrém provozu zvládla změřit tento malý závod s uspokojivými výsledky. Průměrná odchylka od oficiálních výsledků byla v rozmezí 1–2 vteřin, nejčastěji se výsledky lišily o 1 vteřinu. Díky testování bylo zjištěno, že přesnost měření velmi závisí na pozornosti časoměřiče. Při velkém počtu účastníků by mohlo dojít k větším nepřesnostem a také nepravdivým výsledkům závodu.

V poslední části práce byly popsány různé možnosti vylepšení nebo rozšíření mobilní aplikace. Byla nastíněna možnost rozšíření o webovou aplikaci, která by se starala o administraci závodů, a také drobné vylepšení vytvořené mobilní aplikace, konkrétně možnost propojení více mobilních zařízení, přidání dalších jazyků, možnost měření mezičasu závodu a implementace pro zařízení s operačním systémem iOS.

## Literatura

- [1] POTREČZ, Vojtěch. *RFID čipy a průmysl jedenadvacátého století* [online]. [cit. 2021-04-22]. Dostupné z: <https://www.systemonline.cz/rizeni-vyroby/rfid-cipy-a-prumysl-jedenadvacateho-stoleti.htm>
- [2] ČIPOVÉ MĚŘENÍ ZÁVODŮ. *Ptime* [online]. [cit. 2021-4-24]. Dostupné z: <http://ptime.cz/cipove-mereni-zavodu/>
- [3] *AM CHRONO* [online]. [cit. 2021-4-25]. Dostupné z: <https://www.am-chrono.cz>
- [4] *IRONTIME* [online]. [cit. 2021-4-25]. Dostupné z: <https://irontime.cz>
- [5] *ORGSU* [online]. [cit. 2021-4-25]. Dostupné z: <https://www.orgsu.com/cs/>
- [6] Market share of mobile operating systems in Czechia from 2010 to 2020. *Statista* [online]. [cit. 2021-5-5]. Dostupné z: <https://www.statista.com/statistics/669587/market-share-mobile-operating-systems-czech-republic/>
- [7] What is Java? Definition, Meaning & Features of Java Platforms. *Guru99* [online]. [cit. 2021-5-5]. Dostupné z: <https://www.guru99.com/java-platform.html>
- [8] RYCHTECKÝ, Lukáš. MVC vs. MVP. *ČVUT FEL CourseWare Wiki* [online]. [cit. 2021-5-9]. Dostupné z: [https://cw.fel.cvut.cz/old/\\_media/courses/a7b39wa1/tutorials/08/mvc-mvp.pdf](https://cw.fel.cvut.cz/old/_media/courses/a7b39wa1/tutorials/08/mvc-mvp.pdf)
- [9] ROHIT. Android Fragment Example in Kotlin. *Tutorial By EyeHunts* [online]. [cit. 2021-4-24]. Dostupné z: <https://tutorial.eyehunts.com/android/android-fragment-example-kotlin/>
- [10] Difference Between View and ViewGroup in Android. *GeeksforGeeks* [online]. [cit. 2021-5-5]. Dostupné z: <https://www.geeksforgeeks.org/difference-between-view-and-viewgroup-in-android/>
- [11] Chronometer Tutorial With Example In Android Studio. *AbhiAndroid* [online]. [cit. 2021-5-5]. Dostupné z: <https://abhiandroid.com/ui/chronometer>
- [12] CountdownTimer. *Android Developers* [online]. [cit. 2021-5-5]. Dostupné z: <https://developer.android.com/reference/android/os/CountDownTimer>
- [13] Spinners. *Android Developers* [online]. [cit. 2021-5-5]. Dostupné z: <https://developer.android.com/guide/topics/ui/controls/spinner>
- [14] How to Show a Toast for a Specific Duration in Android. *Code Project* [online]. [cit. 2021-5-5]. Dostupné z: <https://www.codeproject.com/Articles/988256/How-to-Show-a-Toast-for-a-Specific-Duration-in-And>

- [15] Menus. Android Developers [online]. [cit. 2021-5-5]. Dostupné z: <https://developer.android.com/guide/topics/ui/menus>
- [16] Publikování aplikace. Nápověda Play Console [online]. [cit. 2021-5-16]. Dostupné z: <https://support.google.com/googleplay/android-developer/answer/9859751?hl=cs>
- [17] Support different languages and cultures. Android Developers [online]. [cit. 2021-5-5]. Dostupné z: <https://developer.android.com/training/basics/supporting-devices/languages>
- [18] Cards. SPORTident [online]. [cit. 2021-5-16]. Dostupné z: <https://www.sportident.com/products.html#cards>



# Přílohy

## Příloha A: Aplikace na Google Play

Mobilní aplikaci je po kontrole ze strany společnosti Google možné stáhnout na Google Play pod názvem „*PocketRaceTimer*“. Odkaz na aplikaci je následující:

- <https://play.google.com/store/apps/details?id=com.lukaskorinek.pocketracetimer>

## Příloha B: Zdrojové kódy

Zdrojové kódy mobilní aplikace je možné stáhnout z repositáře na [Githubu](https://github.com)<sup>8</sup>. Odkaz na projekt je následující:

- <https://github.com/LukasKorinek67/PocketRaceTimer>

## Příloha C: Import startovní listiny – vzorový CSV soubor

	A	B	C	D	E	F	G	H
1	Jméno	Příjmení	Číslo	Pohlaví	Datum naroz	Kategorie	Tým	
2	Tereza	Stará	1	Žena	02.08.1980	Ženy	AC Slovan Liberec	
3	Jana	Richterová	2	Žena	03.08.1977	Ženy	Slavia Liberec Orienteering	
4	Anna	Poláková	3	Žena	04.06.1994	Ženy	Slavia Liberec Orienteering	
5	Eva	Černá	4	Žena	16.11.1983	Ženy	AC Slovan Liberec	
6	Pavčina	Vrchovská	5	Žena	12.03.1989	Ženy		
7	Vendula	Patecká	6	Žena	03.04.1988	Ženy		
8	Dominika	Nová	7	Žena	21.10.1990	Ženy		
9	Karolína	Sázavská	8	Žena	23.05.1988	Ženy	AC Slovan Liberec	
10	Veronika	Větrovská	9	Žena	23.07.1993	Ženy		
11	Adam	Granc	60	Muž	07.07.1975	Muži	AC Slovan Liberec	
12	David	Tok	61	Muž	09.09.1980	Muži		
13	Tomáš	David	62	Muž	09.05.1992	Muži	Slavia Liberec Orienteering	
14	Prokop	Němec	63	Muž	03.03.1989	Muži	Slavia Liberec Orienteering	
15	Vojtěch	Musil	64	Muž	02.01.1997	Muži	AC Slovan Liberec	
16	Petr	Pavelka	65	Muž	12.12.1990	Muži		
17	Šimon	Novák	66	Muž	11.03.1982	Muži		
18	Lukáš	Kořínek	67	Muž	21.09.1998	Muži		
19	Robert	Gernát	68	Muž	22.08.2012	Muži	AC Slovan Liberec	
20	Václav	Bárta	69	Muž	11.04.1988	Muži	Slavia Liberec Orienteering	
21	Eliška	Stránská	100	Žena	10.10.2013	Děti		
22	David	Slovák	101	Muž	23.12.2011	Děti		
23	Petra	Černá	102	Žena	04.05.2010	Děti	AC Slovan Liberec	
24	Patrik	Sázavský	103	Muž	01.08.2010	Děti	AC Slovan Liberec	
25	Anička	Němcová	104	Žena	03.03.2014	Děti		
26								

<sup>8</sup> <https://github.com>

## Příloha D: Export výsledků do CSV souboru – příklad

	A	B	C	D	E	F
1	Děti					
2	Umístění	Číslo závodníka	Jméno	Čas	Tým	
3	1	103	Patrik Sázavský	0:52:30	AC Slovan Liberec	
4	2	100	Eliška Stránská	0:54:41		
5	3	102	Petra Čemá	0:54:56	AC Slovan Liberec	
6	4	101	David Slovák	1:00:54		
7	N	104	Anička Němcová	-		
8						
9						
10	Muži					
11	Umístění	Číslo závodníka	Jméno	Čas	Tým	
12	1	67	Lukáš Kořínek	0:50:09		
13	2	64	Vojtěch Musil	0:50:22	AC Slovan Liberec	
14	3	60	Adam Grancí	0:50:38	AC Slovan Liberec	
15	4	63	Prokop Němec	0:50:49	Slavia Liberec Orienteering	
16	5	69	Václav Bárta	0:51:12	Slavia Liberec Orienteering	
17	6	65	Petr Pavelka	0:52:40		
18	7	61	David Tok	0:52:52		
19	8	62	Tomáš David	0:52:53	Slavia Liberec Orienteering	
20	9	66	Šimon Novák	0:53:06		
21	N	68	Robert Gemát	-	AC Slovan Liberec	
22						
23						
24	Ženy					
25	Umístění	Číslo závodníka	Jméno	Čas	Tým	
26	1	7	Dominika Nová	0:51:00		
27	2	9	Veronika Větrovská	0:51:06		
28	3	1	Tereza Stará	0:51:37	AC Slovan Liberec	
29	4	8	Karolína Sázavská	0:52:34	AC Slovan Liberec	
30	5	3	Anna Poláková	0:53:29	Slavia Liberec Orienteering	
31	6	2	Jana Richtlová	0:53:32	Slavia Liberec Orienteering	
32	7	4	Eva Čemá	0:53:48	AC Slovan Liberec	
33	8	6	Vendula Patecká	0:53:54		
34	9	5	Pavína Vrchovská	0:54:12		
35						
36						
37	Všichni závodníci					
38	Umístění	Číslo závodníka	Jméno	Čas	Kategorie	Tým
39	1	67	Lukáš Kořínek	0:50:09	Muži	Tým neuveden
40	2	64	Vojtěch Musil	0:50:22	Muži	AC Slovan Liberec

34		9	5	Pavína Vrchovská	0:54:12		
35							
36							
37	Všichni závodníci						
38	Umístění	Číslo závodníka	Jméno	Čas	Kategorie	Tým	
39		1	67	Lukáš Kořínek	0:50:09	Muži	Tým neuveden
40		2	64	Vojtěch Musil	0:50:22	Muži	AC Slovan Liberec
41		3	60	Adam Grancí	0:50:38	Muži	AC Slovan Liberec
42		4	63	Prokop Němec	0:50:49	Muži	Slavia Liberec Orier
43		5	7	Dominika Nová	0:51:00	Ženy	Tým neuveden
44		6	9	Veronika Větrovská	0:51:06	Ženy	Tým neuveden
45		7	69	Václav Bárta	0:51:12	Muži	Slavia Liberec Orier
46		8	1	Tereza Stará	0:51:37	Ženy	AC Slovan Liberec
47		9	103	Patrik Sázavský	0:52:30	Děti	AC Slovan Liberec
48		10	8	Karolína Sázavská	0:52:34	Ženy	AC Slovan Liberec
49		11	65	Petr Pavelka	0:52:40	Muži	Tým neuveden
50		12	61	David Tok	0:52:52	Muži	Tým neuveden
51		13	62	Tomáš David	0:52:53	Muži	Slavia Liberec Orier
52		14	66	Šimon Novák	0:53:06	Muži	Tým neuveden
53		15	3	Anna Poláková	0:53:29	Ženy	Slavia Liberec Orier
54		16	2	Jana Richterová	0:53:32	Ženy	Slavia Liberec Orier
55		17	4	Eva Černá	0:53:48	Ženy	AC Slovan Liberec
56		18	6	Vendula Patecká	0:53:54	Ženy	Tým neuveden
57		19	5	Pavína Vrchovská	0:54:12	Ženy	Tým neuveden
58		20	100	Eliška Stránská	0:54:41	Děti	Tým neuveden
59		21	102	Petra Černá	0:54:56	Děti	AC Slovan Liberec
60		22	101	David Slovák	1:00:54	Děti	Tým neuveden
61	N		68	Robert Gemát	-	Muži	AC Slovan Liberec
62	N		104	Anička Němcová	-	Děti	Tým neuveden

## Příloha E: Export výsledků do PDF souboru – příklad

Výsledky závodu					
<b>Děti</b>					
1.	103	Patrik Sázavský	00:52:30	AC Slovan Liberec	
2.	100	Eliška Stránská	00:54:41	Tým neuveden	
3.	102	Petra Černá	00:54:56	AC Slovan Liberec	
4.	101	David Slovák	01:00:54	Tým neuveden	
N	104	Anička Němcová	-	Tým neuveden	
<b>Muži</b>					
1.	67	Lukáš Kořínek	00:50:09	Tým neuveden	
2.	64	Vojtěch Musil	00:50:22	AC Slovan Liberec	
3.	60	Adam Granc1	00:50:38	AC Slovan Liberec	
4.	63	Prokop Němec	00:50:49	Slavia Liberec Orienteering	
5.	69	Václav Bárta	00:51:12	Slavia Liberec Orienteering	
6.	65	Petr Pavelka	00:52:40	Tým neuveden	
7.	61	David Tok	00:52:52	Tým neuveden	
8.	62	Tomáš David	00:52:53	Slavia Liberec Orienteering	
9.	66	Šimon Novák	00:53:06	Tým neuveden	
N	68	Robert Gernát	-	AC Slovan Liberec	
<b>Ženy</b>					
1.	7	Dominika Nová	00:51:00	Tým neuveden	
2.	9	Veronika Větrovská	00:51:06	Tým neuveden	
3.	1	Tereza Stará	00:51:37	AC Slovan Liberec	
4.	8	Karolína Sázavská	00:52:34	AC Slovan Liberec	
5.	3	Anna Poláková	00:53:29	Slavia Liberec Orienteering	
6.	2	Jana Richterová	00:53:32	Slavia Liberec Orienteering	
7.	4	Eva Černá	00:53:48	AC Slovan Liberec	
8.	6	Vendula Patecká	00:53:54	Tým neuveden	
9.	5	Pavčina Vrchovská	00:54:12	Tým neuveden	
<b>Všichni závodníci</b>					
1.	67	Lukáš Kořínek	00:50:09	Muži	Tým neuveden
2.	64	Vojtěch Musil	00:50:22	Muži	AC Slovan Liberec
3.	60	Adam Granc1	00:50:38	Muži	AC Slovan Liberec
4.	63	Prokop Němec	00:50:49	Muži	Slavia Liberec Orienteering
5.	7	Dominika Nová	00:51:00	Ženy	Tým neuveden

## Příloha F: Export výsledků do HTML souboru – příklad

file:///Users/lukaskorinek/Downloads/Výsledky%20závodu.html

Můj disk - Disk Google

### Výsledky závodu

#### Děti

Umístění	Číslo závodníka	Jméno	Čas	Tým
1.	103	Patrik Sázavský	00:52:30	AC Slovan Liberec
2.	100	Eliška Stránská	00:54:41	Tým neuveden
3.	102	Petra Černá	00:54:56	AC Slovan Liberec
4.	101	David Slovák	01:00:54	Tým neuveden
N	104	Anička Němcová	-	Tým neuveden

#### Muži

Umístění	Číslo závodníka	Jméno	Čas	Tým
1.	67	Lukáš Kofínek	00:50:09	Tým neuveden
2.	64	Vojtěch Musil	00:50:22	AC Slovan Liberec
3.	60	Adam Grancl	00:50:38	AC Slovan Liberec
4.	63	Prokop Němec	00:50:49	Slavia Liberec Orienteering
5.	69	Václav Bárta	00:51:12	Slavia Liberec Orienteering
6.	65	Petr Pavelka	00:52:40	Tým neuveden
7.	61	David Tok	00:52:52	Tým neuveden
8.	62	Tomáš David	00:52:53	Slavia Liberec Orienteering
9.	66	Šimon Novák	00:53:06	Tým neuveden
N	68	Robert Gernát	-	AC Slovan Liberec

#### Ženy

Umístění	Číslo závodníka	Jméno	Čas	Tým
----------	-----------------	-------	-----	-----

file:///Users/lukaskorinek/Downloads/Výsledky%20závodu.html

Můj disk - Disk Google

### Výsledky závodu

#### Ženy

Umístění	Číslo závodníka	Jméno	Čas	Tým
1.	7	Dominika Nová	00:51:00	Tým neuveden
2.	9	Veronika Větrovská	00:51:06	Tým neuveden
3.	1	Tereza Stará	00:51:37	AC Slovan Liberec
4.	8	Karolína Sázavská	00:52:34	AC Slovan Liberec
5.	3	Anna Poláková	00:53:29	Slavia Liberec Orienteering
6.	2	Jana Richterová	00:53:32	Slavia Liberec Orienteering
7.	4	Eva Černá	00:53:48	AC Slovan Liberec
8.	6	Vendula Patecká	00:53:54	Tým neuveden
9.	5	Pavčina Vrchovska	00:54:12	Tým neuveden

#### Všichni závodníci

Umístění	Číslo závodníka	Jméno	Čas	Kategorie	Tým
1.	67	Lukáš Kofínek	00:50:09	Muži	Tým neuveden
2.	64	Vojtěch Musil	00:50:22	Muži	AC Slovan Liberec
3.	60	Adam Grancl	00:50:38	Muži	AC Slovan Liberec
4.	63	Prokop Němec	00:50:49	Muži	Slavia Liberec Orienteering
5.	7	Dominika Nová	00:51:00	Ženy	Tým neuveden
6.	9	Veronika Větrovská	00:51:06	Ženy	Tým neuveden
7.	69	Václav Bárta	00:51:12	Muži	Slavia Liberec Orienteering
8.	1	Tereza Stará	00:51:37	Ženy	AC Slovan Liberec
9.	103	Patrik Sázavský	00:52:30	Děti	AC Slovan Liberec
10.	8	Karolína Sázavská	00:52:34	Ženy	AC Slovan Liberec
11.	65	Petr Pavelka	00:52:40	Muži	Tým neuveden
12.	61	David Tok	00:52:52	Muži	Tým neuveden
13.	62	Tomáš David	00:52:53	Muži	Slavia Liberec Orienteering