

# Aplikace chytrých hodinek pro podporu sportovního tréninku a závodů

Pavel Dohnalík\*



## Abstrakt

Cílem této práce je navrhnout a realizovat novou aplikaci na chytré hodinky pro podporu sportovních tréninků a malých závodů ve sportovních klubech. Výsledná práce zahrnuje multiplatformní multijazyčnou mobilní aplikaci pro Android a iOS, aplikaci na chytré hodinky s Wear OS a webovou aplikaci. Aplikace pro chytré hodinky má obsahovat snadné párování na aplikaci v telefonu. Webová aplikace slouží k vytváření závodních tras. Pomocí hodinek lze přesně změřit sportovní aktivitu, kterou lze následně na telefonu analyzovat. Sportovní aktivitou, na kterou je aplikace zacílená je trénink a závody na in-line rychlobruslích. Na telefonu bude probíhat dopočet zpřesnění polohy k průjezdové bráně a času průjezdu. Stejná funkce měření sportovní aktivity bude k dispozici v telefonu. Závodník bude dostávat informace o své pozici v závodě a to pomocí jednoduchého uživatelského rozhraní. K vytvoření mobilní multiplatformní aplikace, aplikace pro chytré hodinky a webové aplikace je použit framework Flutter 2.0 spolu s jazykem Dart. Pro serverovou část je využívána Java Spring v kombinaci s MySQL databází. Celá serverová část je nasazována v Dockeru a komunikuje s mobilními zařízeními pomocí REST API. Výsledkem je serverová aplikace, mobilní multiplatformní aplikace, aplikace na chytré hodinky a webová aplikace.

**Klíčová slova:** Mobilní aplikace — SmartWatch — Flutter — iOS — Android — Wear OS — 5G — REST — Java — MySQL — Docker

**Příložené materiály:** N/A

\*[dohna39@stud.fit.vutbr.cz](mailto:dohna39@stud.fit.vutbr.cz), Faculty of Information Technology, Brno University of Technology

## 1. Úvod

Cílem této práce je navrhnout a implementovat aplikaci pro měření sportovních aktivit ve sportovních klubech. Mezi tyto sportovní aktivity se řadí kromě tréninků také kontrolní a výkonnostní závody. Měření kontrolních závodů v menších sportovních klubech, případně klubech, které se věnují méně častým sportům, jako je například in-line rychlobruslení, běžecké lyžování apod. probíhá ještě stále jenom za pomoci stopek a následného zápisu času na papír.

Nákup profesionální čipové časomíry, včetně hard-

warového a softwarového vybavení nutného pro správný chod, je pro tyto kluby finančně velmi náročný a pro provoz většinou nemají k dispozici člověka s dostatečnou odborností. Mobilní aplikace ve verzi zdarma většinou neposkytují důležité statistiky z pohledu konkrétního sportu. V krajním případě není ani konkrétní méně častý sport v nabídce aplikace. U aplikací na chytré hodinky je situace ještě komplikovanější v tom, že se nevyužívají senzory přímo v hodinkách. Hodinky ve většině případů jenom zobrazují údaje z telefonu na svém displeji.

12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22

23 Protože se osobně věnuji závodně rychlobruslení  
24 a to jak na kolečkách, tak na ledě, rozhodl jsem se  
25 vytvořit aplikaci, která s vhodným hardwarovým vy-  
26 bavením bude fungovat na chytrých hodinkách zcela  
27 autonomně, nezávisle na telefonu. Pro komunikaci  
28 bude využívána e-sim v hodinkách spolu s datovým  
29 tarifem, s výhledem na nástup mobilních sítí 5. ge-  
30 nerace (tzv. 5G) a využití jejich přenosových vlast-  
31 ností. Důraz bude kladen na dostatečnou přesnost GPS  
32 měření a následně matematické zpřesňování naměře-  
33 ných hodnot. Mobilní aplikace a aplikace na chytré  
34 hodinky by měla sloužit převážně malým a středním  
35 klubům jako spolehlivá a levná alternativa k nákladné  
36 čipové časomíře.



Obrázek 2. Jednorázový pasivní RFID čip.

## 37 2. Existující řešení

38 Mezi aktuálně dostupnými řešeními, která jsou digitální  
39 povahy, patří nákup a provoz čipové časomíry nebo je  
40 na výběr z několika aplikací, které jsou dostupné jak  
41 pro platformu Android tak pro iOS.



Obrázek 3. Anténa pro snímání čipů během závodu.

### 42 2.1 Čipová časomíra

43 Čipová časomíra je složena z pasivního RFID čipu,  
44 který má sportovec během aktivity u sebe. Jedná se  
45 buď o čip, který je opakovaně použitelný, jako je vidět  
46 na obrázku 1, nebo o levnější a méně spolehlivý jed-  
47 norázový čip, který je vidět na obrázku 2. Pro snímání  
48 se používá anténa, která je vidět na obrázku 3. Anténa  
49 je připojená k počítači s vhodným softwarem pomocí  
50 síťového rozhraní.



Obrázek 1. Opakovaně použitelný pasivní RFID čip.

51 Mezi výhody čipové časomíry patří její přesnost,  
52 která se pohybuje kolem 0,01 s [1]. Pro závodníka je  
53 výhoda že k měření sportovní aktivity je plně dostatečný  
54 jeden lehký čip. Mezi nedostatky se řadí nutnost  
55 hlídání si vracení RFID čipů, vyšší počáteční pořizovací  
56 náklady a z důvodu síťového propojování a konfig-  
57 urace i více antén s řídicím počítačem alespoň ele-  
58 mentární znalost z oblasti počítačových sítí. Mezi  
59 nevýhody bych řadil časovou náročnost výstavby a

konfigurace kontrolních bodů, kde se provádí měření a 60  
riziko poškození antény při nepříznivém počasí (silný 61  
poryv větru, vydatnější déšť). 62

### 2.2 Mobilní aplikace Endomondo 63

Aplikace Endomondo, která ve své době patřila k velmi 64  
populárním a používaným aplikacím, je v dnešní době 65  
již bez podpory a není dostupná ani na Google Play 66  
ani na App Store [2]. 67

Aplikace nicméně ve své době nabízela velmi sluš- 68  
né přehledy ze sportovních aktivit a to i ve verzi zdarma. 69  
Tím poskytla nejen sportovci ale i trenérovi dobrá data 70  
k následné analýze. Aplikace byla k dispozici pro 71  
chytré telefony s operačním systémem Android a iOS. 72  
Stáhnout šla také na chytré hodinky. Zde ale fungovala 73  
pouze jako zrcadlo údajů, které se naměřily pomocí 74  
telefonu. Endomondo disponovalo také webovým 75  
prostředím pro podrobnější zkoumání sportovní ak- 76  
tivity. 77

Pro změření a následné porovnání sportovních 78  
výkonů mezi závodníky bylo potřeba vytvořit tzv. výzvu 79  
a přizvat do ní závodníky. Výzvy ale fungují pouze 80  
na principu dosažení určitého cíle (např.: ujet 5 km), 81  
už se nekontroluje, zda všichni účastníci absolvovali 82  
stejnou trasu. 83

## 84 2.3 Strava

85 Další z aktuálně dostupných aplikací pro měření sporto-  
86 vních aktivit je Strava<sup>1</sup>. Strava disponuje aplikací pro  
87 iOS a Android a podporuje i aplikaci na chytré hodinky.  
88 Pro porovnání přímo mezi sportovci slouží tzv. Strava  
89 segmenty. Nevýhodou je, že segmenty v tento okamžik  
90 fungují pouze pro cyklistiku a běh. Ve verzi zdarma  
91 je také dostupnost statistických dat velmi omezena a  
92 uživatel navíc nemůže vytvářet vlastní segmenty.

## 93 3. Návrh aplikací

94 Aplikace pro chytré hodinky bude pro svou funkciona-  
95 litu potřebovat aplikační a databázový server pro ukládá-  
96 ní a synchronizaci dat, mobilní aplikaci pro prohlížení  
97 výsledků po dokončení sportovní aktivity a webovou  
98 aplikaci pro plánování závodních okruhů pomocí mapy.

### 99 3.1 Návrh aplikace pro chytré hodinky

100 Aplikace pro chytré hodinky poskytuje sportovci zákla-  
101 dní informace o sportovní aktivitě, která právě probíhá.  
102 Hodinky neustále komunikují se serverem, na který  
103 odesílají data o průjezdu bránou. Jako odpověď se  
104 vrací ze serveru aktuální pozice v závodě a ztráta na  
105 první místo.

#### 106 3.1.1 Identifikace uživatele

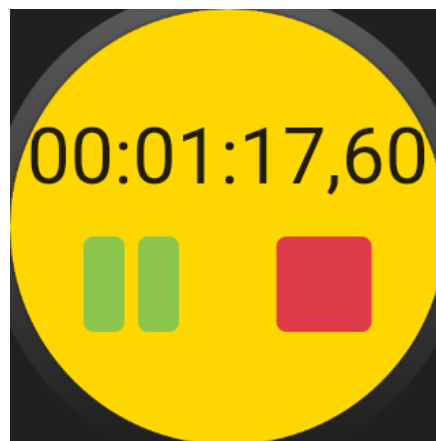
107 Běžným uživatelem aplikace je sportovec věnující  
108 se rychlobruslení na in-line bruslích. Tento uživatel  
109 využívá sportovní hodinky k měření své sportovní  
110 výkonnosti a nechce během tréninku nebo závodu s  
111 sebou vozit telefon, který by údaje potřebné pro vy-  
112 hodnocení zvládl změřit. Hodinky pro sportovce plní  
113 pouze úlohu měřicího zařízení a s ohledem na tuto  
114 požadovanou funkcionalitu je navržené grafické roz-  
115 hraní aplikace.

116 Uživatel aplikace na chytrých hodinkách chce, aby  
117 spuštění aplikace a zahájení měření bylo realizované  
118 na co nejméně uživatelských interakcích. Běžný čas, kdy  
119 uživatel (sportovec) sahá po měřicí aplikaci je ten, kdy  
120 je už připravený vyrazit na okruh – to je má oblečený  
121 dres, ochranné pomůcky (helma, rukavice, chrániče),  
122 nazuté brusle a na ruce chytré hodinky.

#### 123 3.1.2 Grafický návrh aplikace na chytré hodinky

124 Aplikace obsahuje 6 obrazovek, pomocí kterých může  
125 sportovec dostávat informace o právě prováděné sporto-  
126 vní aktivitě. Jedná se o obrazovku na přihlášení, re-  
127 spektive pro párování hodinek s účtem vytvořeném na  
128 telefonu, potvrzovací obrazovku, obrazovku s časem  
129 od startu, obrazovku se vzdáleností od startu, obra-  
130 zovku s pozicí v závodě a odhlašovací obrazovku.

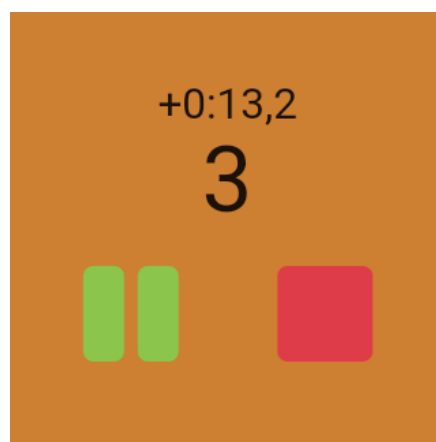
<sup>1</sup><https://apps.apple.com/cz/app/strava-run-ride-training/id426826309?l=cs>



Obrázek 4. Předání informace o pozici (1. místo) a času od startu na kulatém displeji hodinek.

Protože displeje hodinek jsou dost malé - rozměrově 131 mají průměr displeje 38mm, 40mm nebo 44mm, rozhodl 132 jsem se ovládat přesun mezi displeji pomocí gest. Po- 133 mocí táhnutí prstem nahoru a dolů se realizuje přepínání 134 mezi informačními displeji o sportovní aktivitě. Toto 135 přepínání tvoří nekonečnou smyčku. Táhnutí prstem 136 zprava doleva se přepne na obrazovku s možnostmi 137 odhlásit a ukončit aplikaci. 138

Obrazovky hodinek zobrazující informace o právě 139 probíhané sportovní aktivitě, mají jako hlavní prvek 140 ovládací tlačítko. Toto tlačítko umožní odstartování 141 sportovní aktivity, její pozastavení nebo ukončení a 142 je přístupné ze všech obrazovek. Kromě ovládacího 143 tlačítka je dalším dominantním prvkem informace o ak- 144 tivitě. Na obrazovce, která je na obrázku 4 to je čas od 145 startu aktivity. Na další obrazovce, je to vzdálenost, co 146 už sportovec během aktivity absolvoval a na poslední 147 obrazovce, která je na obrázku 5 je to pozice v závodě 148 se ztrátou na první místo. 149



Obrázek 5. Zobrazování informací na hranatém displeji hodinek. Na displeji jsou informace o pozici v závodě přenášeny pomocí barvy pozadí. Současně jsou k dispozici ovládací tlačítka, která pozastavují nebo stopují sportovní aktivitu.

150 Detekce průjezdu branou je spuštěná na pozadí  
151 všech obrazovek. To tedy znamená, že informace o  
152 pozici v závodě se aktualizují i když závodník zrovna  
153 není přepnutý na obrazovku s nimi. Hodinky si po  
154 celou dobu běhu ukládají všechna data o poloze. Po  
155 ukončení sportovní aktivity se odešlou tato data na  
156 server. Hodinky tedy neprovádějí finální dopočet, který  
157 pomůže zpřesnit časový údaj o průjezdu branou.

## 158 3.2 Návrh mobilní aplikace

159 Aplikace na telefon umožní sportovci sledovat stejné  
160 informace jako aplikace na chytré hodinky. Navíc  
161 ukáže závodníkovi informace o jeho aktivitě po jejím  
162 dokončení. Jedná se například o průměrnou rychlost,  
163 maximální rychlost a průměrný čas na kilometr. Navíc  
164 bude možné v mobilní aplikaci nahlédnout do mapy  
165 aby si sportovec mohl prohlédnout polohu bran, kterými  
166 musí projet.

### 167 3.2.1 Identifikace uživatele

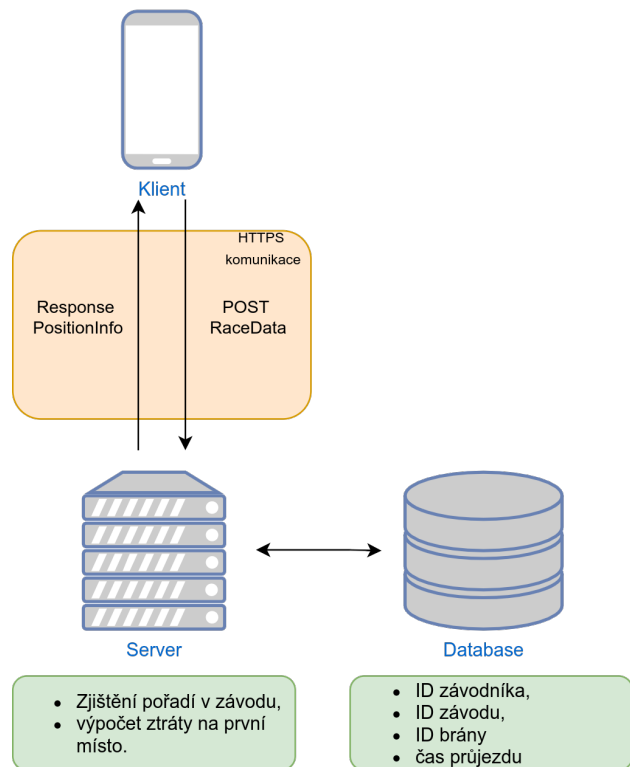
168 Běžný uživatel aplikace na telefon je sportovec (rychlo-  
169 bruslař na in-line bruslích), který z nějakého důvodu  
170 nemůže nebo nechce pro měření využít aplikaci na  
171 chytrých hodinkách. Jako důvody může být nekompat-  
172 ibilita mezi aplikací a operačním systémem hodinek  
173 nebo nebo nedostatečná hardwarová vybavenost hodi-  
174 nek.

175 Hlavním cílem aplikace na chytrém telefonu je  
176 umožnit měření sportovní aktivity a to stejně rychle,  
177 jako na hodinkách. To znamená, že uživatel spustí  
178 aplikaci a hned po jejím startu může zahájit měření  
179 sportovní aktivity. Sportovec na úvodní obrazovce  
180 (obrázek 7) hned vidí mapu, na kterou si přes filtr může  
181 zobrazit nejbližší závody. Má zvolený svůj hlavní  
182 sport a má k dispozici tlačítko 'start'. Po stisknutí start  
183 tlačítka je zahájeno měření a to buď ve formě volného  
184 tréninku nebo vybraného závodu.

### 185 3.2.2 Obrazovka s měřením sportovní aktivity

186 První, co závodník uvidí po přihlášení do aplikace,  
187 respektive po jejím opětovném spuštění, pokud se už  
188 jednou přihlásil, je obrazovka, která ihned umožní  
189 odstartovat sportovní aktivitu. Jedná se o sportovní ak-  
190 tivitou volný trénink, která umožní sportovci si změřit  
191 vzdálenost, kterou během aktivity urazil, a čas, jak  
192 dlouho mu to trvalo. Pohyb se vykresluje do mapy,  
193 která zabírá větší část displeje. Závodník si ještě  
194 před odstartováním volného tréninku může vybrat jaký  
195 sport bude provozovat. Toto může udělat pomocí ikony,  
196 která znázorňuje jim vybraný sport.

197 Pokud se chce závodník účastnit měřené sportovní  
198 aktivity, nebo závodu, je nutné aby si nejprve vybral  
199 sport, kterému se chce věnovat. Následně si pomocí



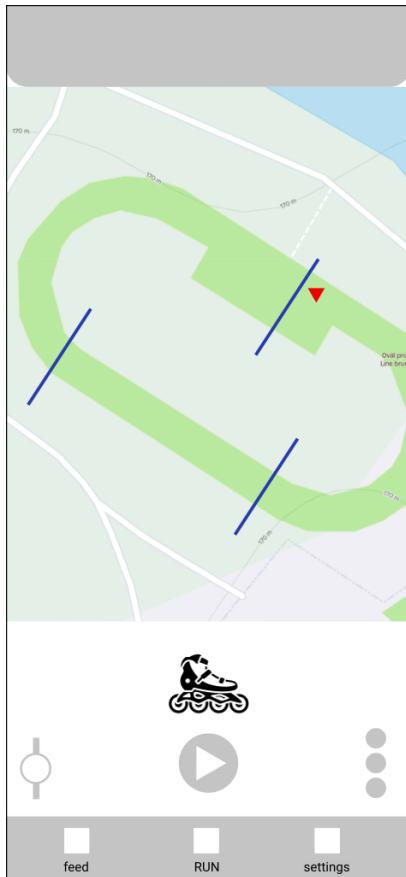
**Obrázek 6.** Schéma komunikace mezi klientem a serverem během závodu. Klient (aplikace puštěná na chytrých hodinkách nebo telefonu) odešle údaje identifikaci závodníka, závodu, brány a času průjezdu. V odpovědi ze serveru dostane pozici závodníka v závodě a jeho ztrátu na první místo.

200 filtru vpravo dole vybere závod, který se mu vykreslí  
201 do mapy pomocí bran. Takto vykreslený závod je  
202 vidět na obrazovce, která je na obrázku 7. Pokud má  
203 závodník spárované hodinky s aplikací v telefonu, tak  
204 se mu vlevo dole zobrazí ikona hodinek. Pomocí ní  
205 může závod odeslat do hodinek a měřit ho pomocí  
206 aplikace v chytrých hodinkách.

207 Měření sportovní aktivity na telefonu probíhá na  
208 pozadí a závodník může tedy přepínat mezi obrazov-  
209 kami. Je nutné aby během aktivity typu závod byl tele-  
210 fon neustále připojený k internetu. Po detekci průjezdu  
211 branou telefon odešle údaje identifikující závodníka,  
212 právě projitou bránu a čas průjezdu na server. Server v  
213 odpovědi vrátí pozici v závodě a ztrátu na první místo.  
214 Tato informace je následně předaná závodníkovi. Kom-  
215 unikace je vyobrazená na obrázku 6.

## 216 3.3 Návrh webové aplikace

217 Webová aplikace umožní sportovci prohlížet stejné in-  
218 formace jako v mobilní verzi. Navíc umožňuje vytvářet  
219 závody a do mapy zakreslovat průjezdové brány po-  
220 mocí klikání myši. Dále si zde závodník může prohlížet  
221 závody co v minulosti vytvořil. Závod, který ještě  
222 nebyl odstartován lze editovat a mazat. Závod co už



**Obrázek 7.** Hlavní obrazovka aplikace, kde je vybraný závod. Modře jsou vyznačené průjezdové brány. Vlevo dole je tlačítko pro odeslání závodu do hodinek. Uprostřed tlačítko pro start. Vpravo filtr pro výběr závodu.

223 je odstartován ale ještě není dokončen je uzamknutý  
 224 pro veškerou editaci. Závod co je již dokončený zle  
 225 kopírovat a kopii editovat v plném rozsahu pro jakékoliv  
 226 budoucí datum.

### 227 3.3.1 Obrazovka pro vytvoření závodu

228 Největší část obrazovky pro tvorbu závodů ve webové  
 229 aplikaci zabírá mapa. Mapu umožňuje závodníkovi  
 230 snadné plánování bran, kterými je nutné projet při  
 231 závodě. Mapa umožňuje provádět základní mapové  
 232 operace jako přiblížení/oddálení nebo posouvání v ní.

233 V pravé části obrazovky je možné vybrat si sport,  
 234 pro který je závod tvořen a datum jeho zahájení a ukončení.  
 235 Následně se pomocí velkého tlačítka plus, které je  
 236 vpravo dole přidávají průjezdové brány. V mapě se po-  
 237 pomocí myši vždy vybere jeden a pak druhý hraniční bod  
 238 brány. Následně se provede dokreslení nejkratší rovné  
 239 cesty mezi těmito body. Brány jsou barevně rozlišeny.  
 240 Zelená barva je pro startovací bránu, červená barva pro  
 241 koncovou bránu a černá barva pro průjezdové brány.

## 3.4 Návrh serverové aplikace 242

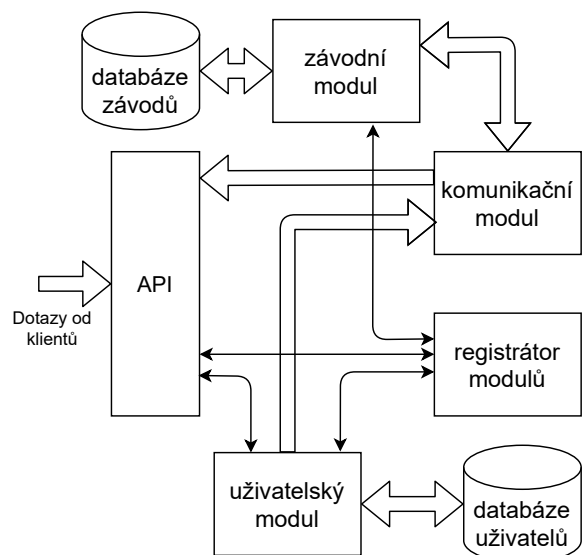
243 Serverová aplikace umožňuje ukládání uživatelských  
 244 dat, jejich synchronizaci a sdílení s ostatními uživateli  
 245 aplikace. Aplikační serverová část poskytuje komu-  
 246 nikační rozhraní pro klienty, komunikační rozhraní  
 247 pro další služby běžící na serveru a funkcionalitu pro  
 248 ukládání a čtení dat z databáze.

### 3.4.1 Server složený z modulů 249

250 Server je složený z několika modulů, jak je vidět na  
 251 schématu na obrázku 8. Komunikační modul API  
 252 přebírá dotazy od klientů (chytré hodinky, mobilní a  
 253 webová aplikace) a přeposílá je buď do uživatelského  
 254 modulu nebo jejich požadavek zkusí vyřešit, pokud  
 255 má uživatel dostatečná oprávnění.

256 Pomocí uživatelského modulu probíhá také ověření,  
 257 zda má uživatel přístup k požadované operaci a datům  
 258 na serveru. Uživatelský modul disponuje rozhraním  
 259 pro komunikaci s API. Má vlastní databázi uživatelů,  
 260 kterou si udržuje a poskytuje data z ní dalším službám  
 261 a to buď pomocí komunikace typu dotaz-odpověď  
 262 nebo publikováním do streamu zpráv. Uživatelský  
 263 modul se musí zvládnout registrovat do registračního  
 264 modulu.

265 Závodní modul má vlastní databázi, kde si udržuje  
 266 záznamy o závodech a výsledcích. V závodním mo-  
 267 dulu je taky implementován dopočet přesnosti pro  
 268 hodinky, které tuto funkcionalitu nemají. Závodní  
 269 modul publikuje data z databáze pomocí streamu zpráv.  
 270 Závodní modul se musí zvládnout registrovat do regis-  
 271 tračního modulu.



**Obrázek 8.** Návrh modulárního serveru pro aplikaci. Registrační modul umožní distribuci jednotlivých modulů na samostatná fyzická zařízení. Komunikační modul zajišťuje doručování zpráv uvnitř systému pomocí mechanismů producent-konzument.

272 Registrační modul v sobě udržuje informace o  
273 službách, co se k němu přihlásily. Na vyžádání tyto  
274 informace poskytuje ostatním službám. Jde o infor-  
275 mace o adrese a portu, kde služba běží. Tyto infor-  
276 mace uložené a zpracováváné v registračním mod-  
277 ulu umožňují mít jednotlivé moduly distribuovány po  
278 různých fyzických zařízeních. Vyhledávání je řešeno  
279 pomocí názvu služby.

280 Komunikační modul má na zodpovědnost distribuci  
281 zpráv pomocí streamů. Jde o zaslání zpráv v režimu  
282 producent-konzumenti, kdy uživatelský modul posky-  
283 tuje data o uživateli (je producent) a závodní modul  
284 je přijímá pro vyhodnocování výsledků (plní roli konzumu-  
285 menta). Závodní modul je producentem, když pub-  
286 likuje výsledky závodníka. V tento moment je konzumu-  
287 mentem modul API.

## 288 4. Implementace

### 289 4.1 Implementace pro chytré hodinky

290 Aplikace pro chytré hodinky obsahuje nejzákladnější  
291 funkcionalitu, která je implementovaná. Tato funkcio-  
292 nalita je následně rozšiřována v aplikaci pro telefony s  
293 Androidem a iOS. Další rozšíření je ve webové verzi.  
294 Celá aplikace na chytré hodinky je implementovaná  
295 v jazyce Dart<sup>2</sup> pomocí frameworku Flutter.

296 Aplikace na chytrých hodinkách obsahuje celkem  
297 6 obrazovek. Každá obrazovka je poděděná od třídy  
298 `StatefulWidget`. Každá obrazovka má svůj ob-  
299 jekt třídy `Model`, který v sobě uchovává data. V ap-  
300 likaci je implementováno 6 tříd, které operují nad  
301 daty a plní funkci služeb. V těchto třídách je im-  
302 plementovaná komunikace se serverem, služby pro  
303 zjišťování a zpřesňování polohy a služba pro měření  
304 času sportovní aktivity.

305 Všechny třídy v aplikaci, které plní roli služeb,  
306 jsou implementovány podle návrhového vzoru single-  
307 ton. Díky tomu je možné všechny tyto třídy využít ke  
308 sdílení dat napříč všemi obrazovkami aplikace. Pro  
309 implementaci třídy jako singleton je využitý flutterový  
310 plugin `get_it`<sup>3</sup>.

#### 311 4.1.1 Komunikace se serverem

312 Pro komunikaci se serverem je implementovaná třída  
313 `HttpService`. Třída v sobě má implementovanou  
314 metodu `String createUrl(List<String> params)`, která vezme list parametrů a načtené údaje  
315 z globální konfigurace a vytvoří odpovídající URL  
316 adresu serveru.

318 Pro komunikaci se serverem jsou zde implemento-  
319 vané metody pro HTTP dotazy typu GET, POST, PUT

a DELETE. Kde každá metoda je pro každý typ dotazu 320  
implementovaná dvakrát. První z nich vykoná dotaz 321  
na server bez položky `authorization` v hlavičce 322  
HTTP požadavku a odešle dotaz. Druhá z metod přidá 323  
do hlavičky HTTP požadavku autorizační token, pokud 324  
ho má z předchozí komunikace k dispozici. Všechny 325  
metody zajišťující HTTP komunikaci z aplikace na 326  
server jsou asynchronní a vyhodnocení se provádí v 327  
bloku `then`. 328

Tělo HTTP dotazu je tvořeno řetězcem v JSON 329  
formátu. Aby práce s objekty, které slouží jako model 330  
pro ukládání dat byla co nejjednodušší, je v aplikaci 331  
využitý plugin `json_serializable`<sup>4</sup>. Všechny 332  
třídy v modelu aplikace mají anotaci `@JsonSeriali-` 333  
`zable()`. Každá třída modelu je implementována 334  
s konstruktorem, který na vstupu má všechny třídní 335  
proměnné, do každé třídy je dopsaná metoda `from-` 336  
`Json(Map<String, dynamic> json)` a meto- 337  
da `toJson()`. Obě tyto metody pracují se zdrojovým 338  
kódem generovaným při kompilaci a realizují převod 339  
z objektu programátorem vytvořené třídy na objekt 340  
třídy `Map`, který je serializovatelný do řetězce v JSON 341  
formátu a zpětný převod z objektu třídy `Map` do ob- 342  
jektu třídy vytvořené programátorem. 343

#### 4.1.2 Zjišťování polohy a její zpřesnění 344

Další částí aplikace, co bylo potřeba implementovat, 345  
byla práce s aktuální polohou uživatele. Jak pro systém 346  
Android tak pro operační systém Wear OS je nutné, aby 347  
uživatel povolil oprávnění přistupovat k geolokačním 348  
službám. K zobrazení povolení je nutné upravit sou- 349  
bor `AndroidManifest.xml`, který obsahuje XML 350  
strukturu nesoucí v sobě informace o aplikaci včetně 351  
jejich oprávnění. Do kořenového uzlu této struktury 352  
byla přidána následující oprávnění: 353

- `ACCESS_FINE_LOCATION`, 354
- `ACCESS_BACKGROUND_LOCATION`, 355
- `ACCESS_COARSE_LOCATION`. 356

Celá logika pro práci s polohou uživatele je im- 357  
plementovaná ve třídě s názvem `GeoService`, kde 358  
její struktura je vidět na obrázku 9. Tato třída posky- 359  
tuje přístup k metodám pro zjišťování aktuální polohy, 360  
výpočtu vzdálenosti mezi body, detekci průjezdu branou a první zpřesnění času průjezdu branou. Pro 361  
získání aktuální polohy je využitý plugin `Geoloca-` 362  
`tor`<sup>5</sup>. Moje třída `GeoService` využívá její instanci 363  
pro čtení bodů souřadnicového systému. Data, která z 364  
pluginu `Geolocator` získána, ukládám do objektu třídy 365  
`Position`. Když je aplikace ve stavu měření závodu, 366  
367

<sup>2</sup><https://dart.dev/>

<sup>3</sup>[https://pub.dev/packages/get\\_it](https://pub.dev/packages/get_it)

<sup>4</sup>[https://pub.dev/packages/json\\_serializable](https://pub.dev/packages/json_serializable)

<sup>5</sup><https://pub.dev/packages/geolocator>

GeoService
positions: [Position] meters: double lastPosition: Position lastIntersection: Position
distanceFrom(position: Position): double calcIntersection(newPosition: Position, gateA: Position, gateB: Position): bool isLocationEnable(): bool distanceFromTo(old: Position, new: Position): double calcFixedGridIntersection(int detectionIndex): double

**Obrázek 9.** Struktura třídy GeoService, která poskytuje metody pro práci s polohou, detekci průjezdu branou a zpřesnění času průjezdu branou

368 jsou data o poloze poskytována pomocí metody `positionStream()`. Tato metoda má na vstupu dva argumenty. Jako parametry nastavuji argument přesnosti určení polohy `desiredAccuracy` na hodnotu nejlepší. To má za následek že zařízení se pokusí určit aktuální polohu co možná nejpřesněji. Jako druhý parametr nastavuji `timeLimit` na hodnotu 5 vteřin. Je to čas, po kterém dojde k `TimeoutException` z důvodu nenavázání GPS spojení a nezjištění aktuální polohy.

378 Zpřesnění času průjezdu branou pomocí metody `calcFixedGridIntersection()` je založené na předpokladu, že se uživatel na krátké vzdálenosti své trajektorie pohybuje konstantní rychlostí. Ve třídě GeoService jsou ukládány všechny hodnoty o poloze včetně času. Následně jde tedy modelovat situaci jako je na obrázku 10. Oranžová čára je trajektorie pohybu, kde v bodech  $T_x$  znám souřadnice a čas, kdy v nich byl uživatel. Body  $G_0$  a  $G_1$  jsou krajní body průjezdové brány. Hodnoty v  $S_0$  a  $S_1$  jsou vzdálenosti mezi průsečíkem brány s trajektorií pohybu a prvním bodem před branou a za ní. Naměřené údaje mi umožní využít následující vzorec:

$$t_I = (T_3 - T_2) \frac{S_0}{S_0 + S_1} + T_2 \quad (1)$$

391 kde výsledné  $t_I$  je zpřesněný čas průjezdu na základě  
392 prvního bodu před branou a prvního bodu za branou a  
393  $S_0$  s  $S_1$  jsou počítány následujícím způsobem:

$$R = 6372.795 \quad (2)$$

$$\delta lat = lat_2 - lat_1 \quad (3)$$

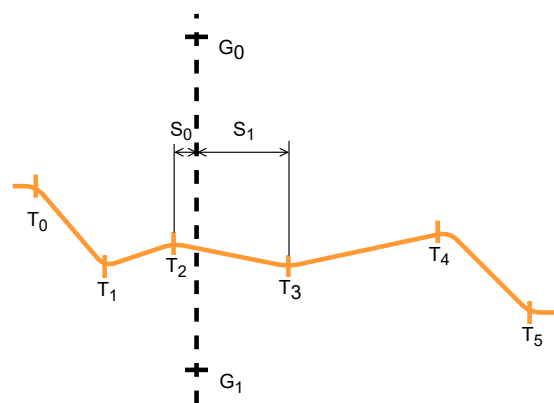
$$\delta lon = lon_2 - lon_1 \quad (4)$$

$$a = \sin^2\left(\frac{\delta lat}{2}\right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2\left(\frac{\delta lon}{2}\right) \quad (5)$$

$$c = 2 \cdot \arctan\left(\frac{\sqrt{a}}{\sqrt{1-a}}\right) \quad (6)$$

$$d = R \cdot c \quad (7)$$

kde  $R$  je poloměr zeměkoule v kilometrech,  $lat_1, lon_1$  jsou souřadnice prvního bodu a  $lat_2, lon_2$  jsou souřadnice druhého bodu [3]. Z takto získaných vzdáleností  $S_0$  a  $S_1$  se vypočítá poměr  $S_0$  ku celkové vzdálenosti mezi  $T_2$  a  $T_3$ . Vzdálenostní poměr mezi  $T_2$  a  $T_3$  by měl odpovídat poměru z rozdílu času mezi  $T_2$  a  $T_3$ . Tento poměr času se nakonec přičte k času v  $T_2$ .



**Obrázek 10.** Schéma dopočtu času průjezdu branou na základě dvou bodů a poměru vzdáleností mezi body a průsečíkem s branou. Oranžová čára je trajektorie pohybu, kde v bodech  $T_x$  znám souřadnice a čas, kdy v nich byl uživatel. Body  $G_0$  a  $G_1$  jsou krajní body průjezdové brány. Hodnoty v  $S_0$  a  $S_1$  jsou vzdálenosti mezi průsečíkem brány s trajektorií pohybu a prvním bodem před branou a za ní.

#### 4.1.3 Párování hodinek na uživatele 401

Obrazovka pro párování s mobilním zařízením se zobrazí pouze tehdy, když se jedná o první spuštění aplikace, nebo se uživatel z aplikace odhlásil. Při vykreslování obrazovky ukázané na obrázku 11 je v metodě `initState()` ve třídě `OpenScreen` vygenerovaný pomocí třídy `RandomNumService` a ověřený dotazem na server unikátní náhodný šestciferný kód. Po zadání do aplikace v telefonu a potvrzení úkonu na hodinkách dojde ke spárování hodinek s uživatelem.



**Obrázek 11.** Obrazovka pro párování s uživatelským účtem. Po uživatelském testování byl původní záměr zadávat kód na hodinky vyhodnocen jako pro uživatele nepříjemný a to z důvodu špatného ovládání ve sportovních rukavicích.

## 412 4.2 Implementace mobilní aplikace

413 Aplikace pro chytrý telefon rozšiřuje funkcionalitu z  
414 hodinek. Oproti aplikaci na chytré hodinky je zde  
415 navíc implementovaná podpora pro operační systém  
416 iOS, mapové služby využívající knihovnu pro Map-  
417 Box<sup>6</sup> mapy, obrazovka zobrazující detail sportovní  
418 aktivity a ‘nekonečný’ list přehledu sportovních ak-  
419 tivit.

420 Pro přepínání mezi obrazovkami je implemento-  
421 vaná v samostatném souboru `routing` metoda spolu  
422 s konstantami nutnými pro přepínání obrazovek s pod-  
423 porou animace. Pro vybranou komunikaci se serverovou  
424 aplikací je implementovaný websocket. Mobilní ap-  
425 likace navíc oproti hodinkám disponuje zpřesňováním  
426 času průjezdu branou pomocí lineární regrese.

### 427 4.2.1 Zpřesnění času průjezdu branou pomocí li- 428 neární regrese

429 Mobilní aplikace po dokončení sportovní aktivity pro-  
430 vádí dopočet pomocí matematické lineární regrese  
431 a to přímo v telefonu uživatele. Pro tyto účely je  
432 využitý plugin `ml_algo`<sup>7</sup>. Pro výpočet dvojce čísel  
433 představujících parametry přímky si vytvořím objekt ze  
434 třídy `DataFrame`. V konstruktoru této třídy jsou vs-  
435 tupnými parametry dvě pole, kde první z nich představu-  
436 je hodnoty pro osu X (zeměpisná šířka, respektive  
437 zeměpisná délka) a druhé pro osu Y (čas naměřený v  
438 bodě). Takto vytvořený objekt se použije jako parametr  
439 pro konstruktor třídy `LinearRegressor`. Z takto  
440 vytvořeného objektu čtu hodnotu proměnné `coeffi-`  
441 `icients`, kde jsou uložena čísla pro parametrický  
442 popis přímky.

<sup>6</sup><https://www.mapbox.com/>

<sup>7</sup>[https://pub.dev/packages/ml\\_algo](https://pub.dev/packages/ml_algo)

Pro zpřesnění se provede výpočet lineární regrese 443  
dvakrát a to v prvním případě na závislost mezi země- 444  
pisnou šířkou a časem v nich naměřených a podruhé 445  
na závislost mezi zeměpisnou délkou a časem v nich 446  
naměřených. Pro výpočet se použije 6 naměřených 447  
bodů před detekcí průjezdu branou a 6 naměřených 448  
bodů po průjezdu branou. Výsledkem jsou dvě dvojce 449  
čísel, které popisují parametricky přímku funkce času 450  
v závislosti k zeměpisné šířce, resp. zeměpisné délce. 451

Do parametrických rovnic přímky vytvořených po- 452  
mocí parametrů získaných z lineární regrese, dosadím 453  
zeměpisnou délku, respektive zeměpisnou šířku a vypo- 454  
čítám dva časy průjezdů, které následně zprůměruji. 455  
Tato funkcionalita je implementovaná ve třídě `Li-` 456  
`nearRegressionService`. 457

### 4.2.2 Detail závodu po jeho dojetí 458

Aplikace pro mobilní telefon umožňuje navíc oproti 459  
hodinkám zobrazit informace o absolvovaném závodě. 460  
Obrazovka (obrázek 12) je vytvořena z několika tříd, 461  
které dědí od `StatefulWidget` třídy. Pro získání 462  
dat o pozici a času na průjezdech jednotlivými branami 463  
se využije třída `HttpService`, která pomocí metody 464  
`getRequest()` získá ze serveru řetězec v JSON 465  
formátu. Ten je následně pomocí funkce `fromJson()` 466  
převeden na objekt třídy `RaceResult`. Objekt je 467  
předán příslušným třídám zajišťující zobrazení dat, 468  
které po jeho obdržení provolají metodu `setState()`. 469  
To má za následek zobrazení obdržенých dat na displej 470  
zařízení. 471

## 4.3 Implementace webové aplikace 472

Webová aplikace navazuje implementačně na mobilní 473  
aplikaci, ze které přebírá služby a objekty pro zo- 474  
brazení dat o závodě. Z implementace vypouští GPS 475  
služby, protože není předpokládáné jejich využívání 476  
přes webový prohlížeč. Cílem webové aplikace je 477  
poskytnout pohodlné klikací prostředí pro plánování 478  
závodů v mapě. 479

### 4.3.1 Mapové služby 480

Pro vykreslení map a jejich využívání jsem vytvořil 481  
třidu `MapService`. Tato třída využívá plugin `map-` 482  
`box_gl`<sup>8</sup>. Pro zobrazení map je potřeba se registrovat 483  
na stránkách <https://www.mapbox.com/>, kde 484  
se získá přístupový token. 485

V konstruktoru třídy `MapService` aplikuji získa- 486  
ný token. Následně nastavím kameru pomocí třídy 487  
`CameraPosition`, které předám získané souřadnice 488  
o aktuální poloze a nastavím počáteční hodnotu pro 489  
zoom. Při kliknutí do mapy, jsou z mapy získány 490  
souřadnice. Jakmile existuje dvojce souřadnic, je tato 491

<sup>8</sup>[https://pub.dev/packages/mapbox\\_gl](https://pub.dev/packages/mapbox_gl)





**Obrázek 12.** Detail dokončeného závodu. Uživatel obdrží informace o svém umístění v cíli, ztrátě na první místo, celkovém čase a vzdálenosti co absolvoval. Dále dostane informace o časech na jednotlivých kontrolních bodech, tzv bránách.

492 dvojice použítá jako jeden z argumentů pro konstruktor  
493 třídy `Polyline`. Ta provede zakreslení čáry před-  
494 stavující bránu do mapy.

#### 495 4.4 Implementace serverové aplikace

496 Serverová aplikace je implementovaná v programovacím  
497 jazyce Java pomocí frameworku Spring Boot. Ap-  
498 likace je rozdělená do balíků `model`, `controller`,  
499 `service` a `repository`.

500 Třídy v balíku `model`, jsou využívány jak pro  
501 zachycení databáze, tak pro tvorbu objektů, které jsou  
502 následně odesílány nebo přijímány přes interní a ex-  
503 terní komunikační rozhraní. Třídy v balíku `controller`  
504 tvoří komunikační body pro ostatní moduly a aplikace.  
505 Tyto třídy v sobě volají metody tříd z balíku `service`,  
506 kde je implementovaná práce s daty.

##### 507 4.4.1 Komunikační a registrační modul

508 Pro interní komunikaci mezi moduly serverové ap-  
509 likace se využívá REST API a program Apache Kafka

spolu s programem Apache ZooKeeper<sup>9</sup>. 510

Aby každý modul mohl běžet samostatně na fyz- 511  
ickém zařízení, je ho potřeba prve registrovat do Apache 512  
Zookeeper. Ten si vytvoří záznam obsahující IP adresu, 513  
port a název modulu a je schopný o něm informovat os- 514  
tatní moduly. Připojení jak uživatelského modulu, tak 515  
závodního modulu do programu Apache Zookeeper je 516  
definován v souboru `application.properties` 517  
pomocí následující proměnných: 518

- `spring.application.name`, 519
- `spring.cloud.zookeeper.` 520  
`connect-string`, 521
- `spring.cloud.zookeeper.` 522  
`discovery.enabled`. 523

První proměnná definuje jméno, pod kterým bude 524  
modul do registrátoru Apache ZooKeeper přihlášen. 525  
Druhá oznamuje modulu na jaké URL adrese a portu 526  
má program Apache ZooKeeper hledat. Poslední po- 527  
voluje zaslání informací na ostatní moduly registrované 528  
v programu Apache Zookeeper. 529

Pokud chce modul komunikovat s jiným po REST 530  
api, je potřeba se prve zeptat programu Apache Zoo- 531  
Keeper, kde modul je. To se provede pomocí mnou 532  
naprogramované metody `getModuleInfo()` ze třídy 533  
`DiscoverService`. Tato metoda vrací objekt ob- 534  
sahující IP adresu a port. Ty jsou následně spolu s daty 535  
předány do třídy `HttpService`, kde je dokončeno 536  
vytváření cílové URL adresy následně odeslání dat. 537

Pro interní komunikaci mezi moduly je využit 538  
také program Apache Kafka. Přes tento modul je v 539  
režimu producent-konzument zasíláno větší množství 540  
dat. Uživatelský modul má třídu `KafkaService`, 541  
která obsahuje metodu `userPublisher()`. Tato 542  
metoda zastává roli producenta a dává k dispozici 543  
všechny uživatele, co jsou aktuálně přihlášení do ap- 544  
likace. Tyto uživatele čte závodní modul pomocí me- 545  
tody `userConsument()` ve třídě `KafkaService`. 546  
Závodní modul je producentem zpráv obsahujících vý- 547  
sledky závodů. K jejich zasílání do programu Apache 548  
Kafka využívá metodu `resultPublisher()` ve 549  
třídě `KafkaService`. Konzumentem těchto zpráv je 550  
API modul, který je pomocí třídy `WebSocketSer-` 551  
`vice` převádí z programu Apache Kafka do web- 552  
socketu přes který je odesílá do právě připojených 553  
zařízení, které jsou identifikované jako mobilní tele- 554  
fony. 555

## 5. Testování 556

Testování aplikace probíhalo jak na telefonu, tak na 557  
chytrých hodinkách. Algoritmus pro zpřesňování času 558

<sup>9</sup><https://zookeeper.apache.org/>

559 průjezdu branou byl testovaná pomocí dvou mobilních  
560 telefonů, kde oba telefony současně a opakovaně ab-  
561 solvovaly stejnou trasu. V tomto testu byl předpoklá-  
562 daný výsledek ten, že telefon A i telefon B budou vyka-  
563 zovat časy průjezdu jen s mírným, až zanedbatelným  
564 časovým rozdílem.

565 Grafické uživatelské rozhraní hodinek bylo opako-  
566 vaně testováno na uživateli při sportování v reálných  
567 podmínkách jako jsou sprinty, delší výběhy, jízda na  
568 in-line bruslích i na silničním kole. Největší problém  
569 činilo zadávání kódu pro párování hodinek a aplikace  
570 pomocí virtuální klávesnice na hodinkách. Uživatel  
571 nebyl schopný napsat kód napoprvé správně bez rukavic.  
572 V celoprstových rukavicích, které se často používají  
573 jako bezpečnostní prvek při in-line bruslení nebo cyk-  
574 listice, byla tato operace téměř nemožná. Výsledkem  
575 tohoto testu bylo úplné odstranění klávesnice z ap-  
576 likace pro hodinky a přesunutí psaní kódu pro párování  
577 do telefonu.

## 578 6. Rozšíření a pokračování

579 Jako rozšíření aplikace pro chytré hodinky vznikla  
580 aplikace pro telefon, kde uživatel může pohodlně sle-  
581 dovat svoje výsledky a to ihned po dojezdu. Dále  
582 vznikla webová aplikace, kde uživatel může vkládat  
583 brány pro kontrolní body v závodech.

584 Díky tomu, že už teď je aplikace v hodinkách  
585 zcela nezávislá na čidlech, senzorech a komunikačních  
586 modulech telefonu, můžou hodinky, resp. aplikace  
587 využívat komunikaci na 5G síti a to téměř okamžitě,  
588 jak bude 5G síť k dispozici. Jedna z nejzajímavějších  
589 funkcionalit, co by měla v 5G síti být k dispozici je  
590 samotná architektura zaměřená na zařízení (Device-  
591 centric architectures). Chytré hodinky s aplikací by  
592 pak měly snadnější komunikaci mezi účastníky v závo-  
593 dě, protože komunikace by probíhala na přímo mezi  
594 nimi [4]. Dále se bude dát využít vyšší přenosová  
595 rychlost a odolnost proti výpadkům například pro SOS  
596 volání sportovce, kde oznámí svoji polohu a dostupné  
597 informace o zdravotním stavu přímo zdravotnímu dispe-  
598 činku. 5G modul slibuje lepší správu napájení a tím  
599 prodlouženou výdrž baterie [5].

## 600 7. Závěr

601 Cílem práce bylo navrhnout a implementovat aplikaci  
602 na chytré hodinky pro podporu sportovního tréninku a  
603 závodů.

604 Před zahájením vývoje aplikace na chytré hodinky  
605 bylo provedeno několik testovacích měření přesnosti  
606 GPS na mobilních telefonech. Stejně tak bylo prove-  
607 deno několik testů uživatelského rozhraní na hodinkách  
608 a to přímo se sportovci - převážně in-line rychlobruslaři

- kdy uživatelské rozhraní muselo vyhovět vysokým 609  
nárokům a to jak z pohledu ovládní, kdy závodník 610  
má na sobě většinou tenké rukavice a pohybuje se ve 611  
vysoké rychlosti, tak z pohledu předávání informace, 612  
kdy k předání je jen několik málo jednotek sekund. 613

Výsledkem je tedy jednoduchá multijazyčná ap- 614  
likace na chytré hodinky s operačním systémem Wear 615  
OS, která dokáže změřit sportovní výkon a předávat 616  
sportovci základní informace v reálném čase. Tato 617  
aplikace je doplněna o mobilní multiplatformní a mul- 618  
tijazyčnou aplikaci pro operační systém Android a 619  
iOS. V mobilní aplikaci jsou k dispozici přehledy pro 620  
sportovce. Pro snadné vytváření měřících bodů vznikla 621  
webová aplikace, kde je využit Mapbox pro tvorbu 622  
bran. Veškerá synchronizace a komunikace probíhá 623  
přes Docker s Java aplikačním serverem a databází. 624

Jako další rozšíření se nabízí přidání uživatele s 625  
rolí trenér, co by měl v reálném čase přehled o svém 626  
závodníkovi. Stejně tak následné testování na 5G síti – 627  
až na to bude k dispozici HW podpora v hodinkách. 628

## Poděkování 629

Rád bych poděkoval svému vedoucímu práce panu 630  
profesoru Ing. Adamu Heroutovi, Ph.D. za odborné 631  
vedení při tvorbě této práce. 632

## Literatura 633

- [1] amchronos. Chronos. online, Dec 2020. 634  
[http://www.amchrono.cz/cz/s3679/](http://www.amchrono.cz/cz/s3679/c2573-Technologie-a-SW) 635  
[c2573-Technologie-a-SW.](http://www.amchrono.cz/cz/s3679/c2573-Technologie-a-SW) 636
- [2] Endomondo. Endomondo is retired. on- 637  
line, March 2021. [https://support.](https://support.endomondo.com/hc/en-us/articles/360016251359-Endomondo-Is-Retired) 638  
[endomondo.com/hc/en-us/articles/](https://support.endomondo.com/hc/en-us/articles/360016251359-Endomondo-Is-Retired) 639  
[360016251359-Endomondo-Is-Retired.](https://support.endomondo.com/hc/en-us/articles/360016251359-Endomondo-Is-Retired) 640
- [3] Bc. Jakub Konečný. Měření vzdálenosti 641  
a plochy pomocí gps. Diplomová práce, 642  
March 2021. [https://dspace.vutbr.cz/](https://dspace.vutbr.cz/xmlui/handle/11012/12264) 643  
[xmlui/handle/11012/12264.](https://dspace.vutbr.cz/xmlui/handle/11012/12264) 644
- [4] Azremi Abdullah Al-Hadi Mohd Faizal Jamlos 645  
Guy A.E. Vandenbosch Dominique Schreurs Nur 646  
Farahiyah Mohamad Aun, Ping Jack Soh. Rev- 647  
olutionizing wearables for 5g: 5g technologies: 648  
Recent developments and future perspectives for 649  
wearable devices and antennas. online, March 650  
2021. [https://ieeexplore.ieee.org/](https://ieeexplore.ieee.org/abstract/document/7893084) 651  
[abstract/document/7893084.](https://ieeexplore.ieee.org/abstract/document/7893084) 652
- [5] Jukka Mäkelä Olli Apilo, Mikko Uitto. 5g 653  
test network: testing the mobile commu- 654  
nications for sports wearables and me- 655  
dia broadcasting. online, March 2021. 656

657 [https://sisr.swissinformatics.](https://sisr.swissinformatics.org/wp-content/uploads/sites/28/2019/05/EN117-web.pdf#page=12)  
658 [org/wp-content/uploads/sites/28/](https://sisr.swissinformatics.org/wp-content/uploads/sites/28/2019/05/EN117-web.pdf#page=12)  
659 [2019/05/EN117-web.pdf#page=12.](https://sisr.swissinformatics.org/wp-content/uploads/sites/28/2019/05/EN117-web.pdf#page=12)