



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

ZADÁNÍ DIPLOMOVÉ PRÁCE

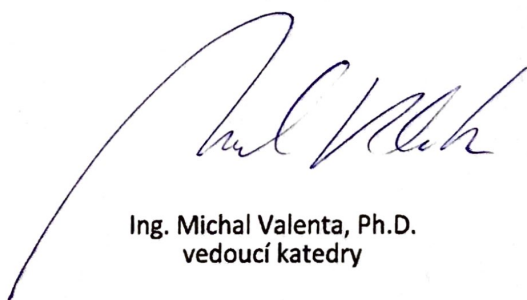
Název: Informační systém pro podporu trénování skoků na trampolíně
Student: Bc. Naim Ashhab
Vedoucí: Ing. Peter Macejko
Studijní program: Informatika
Studijní obor: Webové a softwarové inženýrství
Katedra: Katedra softwarového inženýrství
Platnost zadání: Do konce zimního semestru 2019/20

Pokyny pro vypracování

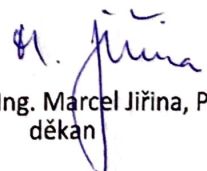
1. Analyzujte existující iOS aplikaci Trampigo, která slouží jako sportovní deník pro skokany na trampolíně.
2. Navrhněte rozšíření pro aplikaci Trampigo s architekturou klient server, které umožní:
 - správu uživatelských účtů
 - tvorbu tréninkových plánů
 - sběr informací o trénincích
 - prezentaci statistik výkonů jednotlivých skokanů.
3. Diskutujte a zvolte vhodnou implementační platformu.
4. Navržený systém implementujte, řádně otestujte a zdokumentujte.

Seznam odborné literatury

- [1] HAUSER, Dominik. Test-Driven iOS Development with Swift 3. Packt Publishing, 2016. ISBN 1787125033.
[2] NADAREISHVILI, Irakli. Microservice architecture: aligning principles, practices, and culture. Sebastopol, CA: O'Reilly Media, 2016. ISBN1491956259.



Ing. Michal Valenta, Ph.D.
vedoucí katedry



doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 20. února 2018

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

Informační systém pro podporu trénování skoků na trampolíně

Bc. Naim Ashhab

Vedoucí práce: Ing. Peter Macejko

9. května 2018

Poděkování

Chtěl bych poděkovat za mnoho konzultací a pomoc při vytváření diplomové práce Ing. Miloši Kozákovi, Ph.D., dále svému vedoucímu Ing. Peteru Macejkovi. V neposlední řadě lidem, kteří se účastnili uživatelského testování, jmenovitě Adamu Súltovi, Karolíně Sikorové, Šárce Korbelové, Jakubovi Vodičkovi a Filipu Fenclovi.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 9. května 2018

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2018 Naim Ashhab. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Ashhab, Naim. *Informační systém pro podporu trénování skoků na trampolíně*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Práce je pokračováním autorovy bakalářské práce, ve které byl vytvořen základ systému pro podporu tréninků ve skocích na trampolíně. Práce zkoumá průběh tréninku na trampolíně, v rámci kterého studuje i systém závodů, jakožto hlavní cíl trénování. Hlavní cíle, které mají původní práci rozšířit, jsou Rychlé a jednoduché zapisování tréninku a Spojení skokan trenér. Z nich analýzou vznikají jednotlivé funkcionality, které jsou navrženy a implementovány do systému. Analýza i návrh navazují na původní práci, narozdíl od implementace. Mobilní aplikace jsou podrobeny testování uživatelského rozhraní pomocí metod heuristika a sledování uživatelů. Výstup práce je funkční a otestovaný systém pro podporu tréninků ve skocích na trampolíně.

Klíčová slova trampolína, trénink, analytický systém, iOS, mobilní aplikace

Abstract

The work is continuation of authors bachelor thesis, where a foundation of a system for a trampoline training support was built. Within the work were researched training sessions and a competition system as a main goals of a training activity. The system main goals are Quick and easy training session

filling and Coach-gymnast connection. These are extensions of the original work. Main goals were analysed and mapped on a set of functionalities, which were later designed and implemented into the system. Only analytic and design phases follow the original work. Mobile application user interfaces were tested via heuristics and observed users. The work outcome is a functional and tested system supporting trampoline trainings.

Keywords trampoline, training, analytic system, iOS, mobile application

Obsah

Úvod	1
1 Cíl práce	3
1.1 Základní pojmy	3
2 Analýza	5
2.1 Závody	5
2.2 Trénink na trampolíně	7
2.3 Uživatelé	9
2.4 Existující řešení	9
2.5 Netechnická analýza	11
2.6 Platformy pro jednotlivé role	12
2.7 Technická analýza	12
3 Návrh	19
3.1 Server	19
3.2 Mobilní aplikace	23
3.3 Uživatelské rozhraní	24
3.4 Navázání spojení skokan - trenér	29
3.5 Wireframe	29
4 Implementace	31
4.1 Způsob vývoje	31
4.2 Mobilní aplikace	37
4.3 Server	37
5 Uživatelské testování	43
5.1 Heuristická analýza	43
5.2 Testování použitelnosti	44

6	Možnosti rozšíření	51
6.1	Další typy událostí	51
6.2	Tréninkové plány	52
6.3	Filtrování a skupiny	52
6.4	Senzor pro automatické zapisování	53
6.5	Kurzor při vytváření kola	53
6.6	Role oddíl	53
	Závěr	55
	Literatura	57
A	Seznam použitých zkratk	59
B	Obsah příloženého CD	61
C	Technické požadavky	63
D	Heuristická analýza uživatelského rozhraní	67

Seznam obrázků

2.1	Funkční požadavky pro Vytváření tréninků z bakalářské práce . . .	13
2.2	Rozšířené funkční požadavky pro Vytváření tréninků	13
2.3	Funkční požadavky pro Správu klávesnice prvků	14
2.4	Funkční požadavky pro Správu seznamu vazeb	14
2.5	Nefunkční požadavky systému	16
2.6	Model tříd z bakalářské práce	16
2.7	Aktualizovaný Model tříd	17
3.1	Sekvenční diagram přihlášení pomocí kombinace Realm a Firebase	23
3.2	Graf obrazovek a aktivit mobilních klientů	25
3.3	Wireframe obrazovky kalendáře se seznamem tréninků	28
3.4	Wireframe obrazovky prázdného tréninku	28
3.5	Wireframe obrazovky vytváření nového kola	28
4.1	Obrázek vývoje produktu pomocí tradiční metodiky a pomocí agilní iterativní metodiky (zdroj: https://www.digitalogy.co/single-post/2017/09/18/Minimum-Viable-Product-MVP---Its-Importance-Pitfalls)	34
4.2	Screenshot skokanského kalendáře se seznamem tréninků	39
4.3	Screenshot trenérského kalendáře se seznamem tréninků	39
4.4	Screenshot obrazovky detailu probíhajícího tréninku	39
4.5	Screenshot obrazovky měsíčních analytik	39
4.6	Screenshot obrazovky vytváření nového kola	39
4.7	Screenshot obrazovky vybírání polohy prvku z klávesnice	39
C.1	Funkční požadavky na přihlášení	63
C.2	Funkční požadavky na vytváření tréninku	64
C.3	Funkční požadavky na spojování uživatelů	64
C.4	Funkční požadavky na zobrazení probíhajících tréninků v reálném čase	64
C.5	Funkční požadavky na zobrazení všech uložených tréninků	65

C.6	Funkční požadavky na analýzu tréninků	65
C.7	Funkční požadavky na klávesnici prvků	65
C.8	Funkční požadavky na seznam vazeb	66
C.9	Nefunkční požadavky na systém	66

Seznam tabulek

2.1	Tabulka parametrů Analýzy tréninků	18
3.1	Tabulka porovnání Realm a Firebase	22
3.2	Tabulka srovnání akcí obrazovky tréninku v různých stavech . . .	27
4.1	Tabulka srovnání strukturovaných a agilních metodik	32
4.2	Tabulka iterací skokanské mobilní aplikace	38
4.3	Tabulka iterací trenérské mobilní aplikace	40
5.1	Tabulka nalezených chyb z uživatelského testování heuristickou analýzou	45

Úvod

Práce popisuje průběh vývoje systému pro podporu trénování skoků na trampolíně. Navazuje na autorovu bakalářskou práci „Mobilní aplikace Trampingo“ [1], ve které byl vytvořen základ pro systém na podporu skoků na trampolíně. V rámci původní práce byly vytvořeny dvě mobilní aplikace, jedna pro roli skokana a druhá pro roli trenéra a server, přes který spolu obě mobilní aplikace komunikovaly. Aplikace měly splňovat minimální potřeby skokanů na trampolíně pro zapisování a sledování výkonnosti. Jedním z cílů bylo Rychlé a jednoduché vyplňování tréninku. Nakonec se cíl, kvůli velké náročnosti, nepodařilo realizovat. Proto si práce klade za hlavní cíl právě Rychlé a jednoduché vyplňování tréninku.

K dosažení cíle bude v práci zanalyzován trénink skokana na trampolíně a závodní systém skoků na trampolíně. Protože skokan trénuje hlavně kvůli účasti na závodech, bude závod nastavovat měřítka výkonnosti, které v tréninku bude možné sledovat. Další analýzou vzniknou požadavky technické i netechnické a množina funkcionalit, které budou dále navrženy, implementovány a otestovány.

V této práci se bude analýza i návrh soustředit primárně na nové funkcionality. Protože práce navazuje na původní práci v analýze a návrhu, bude se v některých částech na původní analýzu a návrh odkazovat. Hlavně v případě funkcionalit, které už v původní práci analyzovány a navrženy byly, i když v systému budou obsaženy také.

Implementace bude probíhat od začátku v iteracích a nebude použit žádný kód z původní práce.

Obě aplikace budou uživatelsky otestovány pomocí heuristiky, ve které aplikaci prochází odborník a kontroluje, zda aplikace splňuje určené body. A dále pomocí metody sledování uživatelů, kteří aplikaci používají. Výstup z testování bude do systému zapracován.

Autor je dlouholetým závodníkem a reprezentantem ve skocích na trampolíně. Stává se tak potřebným expertem při analýze tréninků a závodního systému. V dnešní době stále neexistuje dostatečně specializovaný systém pro

ÚVOD

podporu trénování skoků na trampolíně, který by byl používán ve světovém nebo regionálním měřítku. Proto je autorova motivace takový systém vytvořit a využívat jej i pro vlastní tréninkové potřeby.

Cíl práce

Cíl práce je rozšířit a doplnit existující aplikaci, která byla výstupem autorovy bakalářské práce. Jako konkrétní cíle jsou pak zadefinovány ty požadavky, které se v bakalářské práci kvůli náročnosti vynechaly.

Rychlé a jednoduché zapisování tréninku Aplikace by měla být mobilní tak, aby si ji každý skokan mohl stáhnout a používat v průběhu tréninku. Důraz se klade na rychlost a jednoduchost vyplňování tréninku. Aplikace nesmí skokana od tréninku zdržovat. Zadávání tréninku musí být navrženo tak, aby skokan vyplňoval nebo počítal co nejméně hodnot. Aplikace za skokana odvede většinu práce.

Tento cíl musel být, z důvodů náročnosti z bakalářské práce vyjmut. Proto je to hlavní cíl této práce. Cílem je dát uživateli možnost vyplnit celý trénink jen pomocí prvků, které skočil a ušetřit uživatele předpracovávání tréninkových dat.

Spojení skokan trenér Aplikace by měla nabízet jednoduchý způsob vytváření spojení mezi skokanem a trenérem. Trenér pak vidí všechny uložené tréninky svých spojených skokanů.

1.1 Základní pojmy

V textu se vyskytují pojmy z trampolínové terminologie, které mohou být čtenáři neznámé.

Prvek Prvky jsou ve skocích na trampolíně všechny skoky kromě obyčejného výskoku. Tedy na to, aby se skok mohl označit za prvek, musí skokan provést minimálně čtvrt salta, půl vrut nebo polohu.

Obtížnost prvku Každý prvek na trampolíně má přiřazenou obtížnost. Tato obtížnost je vypočítaná z počtu čtvrt salt, půl vrutů v prvku a dalších bonusových bodů.

1. CÍL PRÁCE

Pokus Pokusem se myslí nepřerušená posloupnost prvků v kole. Skokan může v kole zaskákat několik pokusů. Pokus může obsahovat i jeden prvek.

Kolo Skokan na tréninku neskáče na trampolíně nepřetržitě, ale opakovaně nastupuje a slézá z trampolíny. Pro představu se dá kolo přirovnat ke sprinterovi, který doběhne do cíle a krokem se vrací na start, aby znovu běžel do cíle. Jeden takovýto cyklus se nazývá jedno kolo. Kolo je složeno z pokusů.

Poloha Poloha skoku je postavení těla v průběhu skoku definované pravidly. Polohy jsou ve skocích na trampolíně tři. Poloha schylmo, skrčmo a toporně. Každý prvek musí mít přiřazenou polohu ve které je proveden. Každá poloha přidává různé bonusové body k obtížnosti prvku.

Salto Saltem se myslí otočení o 360° kolem horizontální osy. Hlava se v průběhu salta dostane pod úroveň těla.

Vrut Vrutem se myslí otočení o 360° kolem vertikální osy. Hlava se v průběhu vrutu nedostane pod úroveň těla, pokud není kombinován se saltem.

Jednoduché salto Jednoduché salto je prvek, ve kterém se skokan otočí právě o 360° okolo horizontální osy, tedy skočí právě jedno salto.

Dvojnité salto Dvojnité salto je takové salto, kde se skokan otočí dvakrát přes hlavu. Celkově tedy o 720° .

Analýza

2.1 Závody

Tato sekce popisuje průběh závodů ve skocích na trampolíně, základní pravidla a způsob hodnocení. Závod je hlavní cíl tréninku, proto se očekává, že si skokan bude chtít v tréninku sledovat hodnoty, které se v závodě hodnotí.

2.1.1 Sestavy

Základem závodění ve skocích na trampolíně jsou sestavy. Sestava se definuje jako nepřerušovaná posloupnost deseti různých prvků [2]. V závodním systému je více typů sestav podle obtížnosti. Každý typ má svoje dodatečné podmínky a každý pořadatel závodu si může stanovit minimální typ sestavy, s kterou se mohou závodníci na závod přihlásit.

2.1.2 Závodní systém

Ve skocích na trampolíně se závodí podobně jako v gymnastice. Nejdříve je společná kvalifikace, ze které postupuje prvních 8 skokanů do finále. V případě mistrovství světa a mistrovství evropy je mezi kvalifikací a finálem navíc semifinále, do kterého postupuje 24 skokanů.

V kvalifikaci jsou skákány 2 sestavy, tradičně se označují jako první a druhá. První sestava se od druhé liší pouze tím, že se v ní nepočítá obtížnost všech prvků, ale pouze čtyř vybraných, které se pak nesmějí opakovat ve druhé sestavě ve stejné kvalifikaci. Jak se počítá obtížnost sestavy bude popsáno v sekci 2.1.3.2.

Ve finále, případně semifinále se skáče už jen druhé sestavy, ve kterých je dovoleno opakovat jakýkoli skok z kvalifikace.

Z pravidel vychází, že každý skokan má připravené tři typy sestav. První, druhou a finálovou. Některé mohou být stejné, to záleží na preferenci konkrét-

2. ANALÝZA

ního skokana, ale podle pravidel jsou to tyto tři sestavy, které se v závodech skáčou.

2.1.3 Způsob hodnocení

Skokani dostávají za svoje sestavy bodové ohodnocení, které je složeno z několika částí. Provedení, obtížnot, pohyb na trampolíně a čas letu. Provedení se narozdíl od ostatních částí hodnotí subjektivně za pomoci rozhodčích, kteří dávají, resp. strhávají body za každý skočený prvek. Ostatní části se měří pomocí přístrojů, díky tomu je známka objektivní.

2.1.3.1 Provedení

Provedení hodnotí čtyři rozhodčí, kteří pozorují skokana při sestavě. Skokan dostane za každý skočený prvek 1 bod a rozhodčí mu pak mohou srazit až půl bodu v závislosti na tom, jak byl prvek skočený. Jak se prvky mají podle pravidel skákat a jak se hodnotí, není pro fungování systému podstatné. Více informací o bodování provedení jsou k dispozici v oficiálních pravidlech [2].

Postup pro výpočet výsledné známky je následující. Pro každý prvek se spočte srážka jako medián všech srážek k tomuto skoku od všech rozhodčích. Srážka se odečte od jednoho bodu, tím se získá bodové ohodnocení každého prvku. Zámka za provedení je pak součet všech ohodnocení za skočené prvky. Může se stát, že závodník sestavu nedokončí. Protože ve skocích na trampolíně, narozdíl od gymnastiky, neexistuje možnost sestavu dokončit, závodníkovi se počítají jen ty prvky, které do přerušení skočil.

2.1.3.2 Obtížnost

Každý prvek má podle pravidel svou obtížnost. U první sestavy se vyberou 4 prvky, u druhé jsou to pak všechny. Jejich obtížnost se sečte a výsledek je přímo součástí celkové známky. Vybrané prvky z první sestavy nesmějí být opakovány v druhé sestavě, jinak se jejich obtížnost nepočítá.

Hodnota obtížnosti prvku je založena na počtu čtvrt salt a půl vrutů. Navíc se získávají bonusy za skočené dvojnásalto nebo polohu, ve které byl prvek skočen.

Může se stát, že skokan skočí prvek tak, že není jasné jaká poloha se má počítat. Proto jsou na závodech dva rozhodčí, kteří kontrolují co bylo skočeno, počítají obtížnosti a v případě nejasností rozhodnou, jaký prvek byl skočen.

2.1.3.3 Pohyb na trampolíně

Skokan by měl po celou dobu sestavy skákat co nejvíce na středu trampolíny. Pokud se od středu odchýlí, dostává strážky podobné těm za provedení. Podle pravidel je trampolína rozdělena do sektorů. Ke každému sektoru je přiřazena srážka, kterou skokan dostane, pokud do onoho sektoru doskočí.

Pohyb po trampolíně se měří pomocí tlakových podložek pod nohama trampolíný a je potřeba jen jeden rozhodčí, který obsluhuje přístroj.

2.1.3.4 Čas letu

Ve skocích na trampolíně je snaha hodnotit i výšku sestavy. Protože by bylo velice náročné a nepřesné měřit reálnou výšku skoku, je výška reprezentována časem letu a to tak, že se počítá pouze čas strávený mimo trampolínu. Tím se vyrovnají rozdíly ve váze a výšce skokana. Celkový čas je pak přímo součástí známky za sestavu.

Čas letu je měřen pomocí přístrojů, které rozpoznají, zda je skokan v trampolíně nebo ne. Je zapotřebí jeden rozhodčí, který obsluhuje přístroj.

2.2 Trénink na trampolíně

Tato sekce analyzuje trénink na trampolíně. Identifikuje fáze tréninku a popisuje, jak se skokan na tréninku chová.

2.2.1 Fáze tréninku

Typický trampolínový trénink je rozdělen do několika fází.

1. Rozcvičení
2. Skákání
3. Posilování
4. Protahání

Systém se zaměřuje pouze na fázi skákání. Ostatní fáze mohou být součástí možných rozšíření z kapitoly 6. Pokud se v textu objeví termín trénink, je tím myšlena jen fáze skákání.

Na tréninku je typicky více než jeden skokan. I když je to sport individuální, na tréninku je obvyklé, že jednu trampolínů sdílí několik skokanů najednou. Je to z toho důvodu, že na trampolíně nelze skákat na maximum moc dlouho. Svoji nročností se velice podobá sprintu, kdy sprinter nemůže běžet stejnou rychlostí například závod dlouhý 10 kilometrů. Aby se tedy neztrácel čas, tak se na jedné trampolíně střídají tři až pět skokanů.

Jednomu protočení skokanů se říká kolo.

2.2.2 Kolo na trampolíně

V jednom kole se všichni skokani vystřídají na trampolíně, u chytací matrace a nebo odpočívají.

2.2.2.1 Odpočívání

To je nejvhodnější chvíle pro zápis skočeného kola do systému. Skokan má čas na vydechnutí a přípravu na další kolo. Pokud bude skokan vyplňovat informace právě v tomto čase, nijak ho to nezdržuje od tréninku.

2.2.2.2 Chytání

Skoky na trampolíně mohou být velice nebezpečné, pokud se skokan ve vzduchu ztratí. K tomu dochází i u těch nejlepších skokanů. Kromě začátku a úplného konce nemá skokan vůbec žádný oční kontakt s trampolínou a může se spolehnout jen na svoji orientaci v prostoru. Kvůli tomu je vedle trampolíny ještě jeden člověk, v tréninku je to obvykle skokan, který nastupuje další v pořadí nebo trenér, který dohlíží na právě skákajícího skokana. V případě, že by skokan ztratil orientaci v prostoru, chytač mu podhodí matraci, která celý dopad ztlumí a radikálně sníží šanci na zranění. Matrace se dá podhazovat i v případě, kdy skokan letí moc blízko rámu trampolíny a hrozilo by, že může propadnout nohama do pružin nebo skočit přímo na rám trampolíny. V obou případech pak matrace slouží jako měkká podložka pro bezpečný dopad.

2.2.2.3 Na trampolíně

V části na trampolíně skokan skáče prvky ve vazbách. Toto vyplývá ze závodního systému, ve kterém se závodí v sestavách. Skokanovi tedy nestačí skákat prvky po jednom. To by měl odrážet i tréninkový systém.

Není podmínkou, že by musel skokan skočit vždy jednu vazbu a skončit. Může se stát, že skokan skáče vazbu několikrát za kolo. V tomto případě by měl systém podporovat vnášení informací o kole tak, aby bylo jasné, jestli skokan skočil vazbu nebo jednotlivé prvky.

Příkladem může být trénování nového prvku, kdy skokan skočí vždy jen jeden prvek. V případě, že by systém nepodporoval rozdělování kola na jednotlivé vazby, tato informace by se ztratila a mohlo by se zdát, že skokan skočil několik prvků za sebou. To by ale byla mylná informace. Proto je potřeba, aby systém podporoval takzvané pokusy v kole. Jedním pokusem se pak myslí nepřerušená vazba prvků.

Protože je v závodní známce přímo započítaná výška reprezentovaná časem letu, je běžné, že trenér svým skokanům měří všechny nebo alespoň vybrané pokusy. Ať už jsou to závodní sestavy nebo jen části, je pro skokana i trenéra důležité vědět, jak „vysoko“ byla vazba skočena. Proto by v systému měla být možnost zadat čas k jednotlivým pokusům. Protože je pro zúčastněné náročné měřit dobu letu za pomoci přístrojů, často se tento čas zaměňuje za celkový čas, ve kterém je i doba strávená v trampolíně. Ten nese stejnou informaci o tom, zda se skokanovy sestavy zvyšují nebo snižují, ale jeho měření lze pohodlně zvládnout i za pomoci obyčejných stopek.

Maximálky

Pro skokany na trampolíně je typický ještě jeden typ kola na trampolíně nazývaný „maximálky“. V tomto nástupu na trampolínu je skokanovi měřen čas pouze deseti výskoků. Tento čas pak slouží pro sledování, jak vysoko je skokan schopný skákat bez předvádění jakýchkoli prvků.

Tento typ kola by měl systém také podporovat. Je časté, že se maximálky skáčí alespoň jednou za trénink.

2.3 Uživatelé

Z analýzy vzešly role skokan a trenér. V této sekci budou analyzovány potřeby jednotlivých rolí, na které bude v návrhu brán větší zřetel.

2.3.1 Skokan

Skokan je hlavní role celého systému, která vytváří data, se kterými může systém počítat. Skokan má v tréninku dvě možnosti, kdy si trénink vyplňovat. Jedna je v době odpočinku, kdy skokan vyplňuje trénink postupně, a druhá na konci tréninku, kdy skokan vyplní trénink naráz.

První možnost je preferovanější, protože skokan si pamatuje jak kolo, které odskákal, tak čas, který mu naměřil trenér. Není to však podmínka použití systému.

Skokan, jakožto tvůrce dat, bude nejvíce používat vytváření kol pomocí jednotlivých prvků. Každé zrychlení v této části, ušetří skokanům mnoho času. Špatně navržené zadávání kol by mohlo mít velmi negativní dopad na použití celé aplikace.

2.3.2 Trenér

Trenér je neméně důležitá role při tréninku, ale z pohledu systému je to role sekundární. Trenér už jen pracuje s daty, které skokan vytvořil.

Trenér potřebuje sledovat, jak se skokan vyvíjí. Je důležité, aby měl možnost zobrazit detail tréninku, ale protože je ve většině případů na tréninku osobně, jsou pro trenéra důležitější tréninková analytika, která zobrazí vývoj skokana. Díky analytikám může trenér lépe plánovat budoucí kroky, sledovat aplikované strategie a hledat chyby.

2.4 Existující řešení

Tato sekce popisuje existující řešení pro sledování tréninků ve skocích na trampolíně. Popisuje technicky nejjednodušší způsob zapisování, možnosti použití nesespecializovaných aplikací a využití bakalářské práce autora.

2.4.1 Tužka a papír

Jako první a stále často používaný způsob sledování a vyplňování tréninků na trampolíně je tužka a papír. Podle potřeby trenéra si skokan zapisuje veškeré informace, které následně ručně zpracuje.

Výhodou tužky a papíru je možnost vyplnit si cokoli. Není omezená žádnými pravidly a ani není nutné zapisovat všechny informace, pokud je není nutné sledovat. Jednoduše se dají zapisovat například cíle, komentáře nebo poznámky a není nutné čekat na implementaci obdobné funkcionality v systému.

Nevýhodou je nutnost všechny informace zpracovávat ručně. Papír trenérovi nemůže zpracovat data do přehledných tabulek nebo grafů tak, aby byla vidět tendence skokanovy výkonnosti. V případě, že by chtěl skokan znát nejčastěji skákaný prvek, měl by s tím mnoho práce.

2.4.2 FXE

Služba FXE¹ je určena pro malé trampolíny, na kterých se nezávodí, ale využívají se ke zlepšení kondice. I když služba nemůže být zvažována jako alternativa pro tréninkový systém, je zajímavá svými funkcionalitami.

FXE vyrábí zařízení, které si uživatel připevní na botu a pomocí aplikace může sledovat, jak si vede při běhu a skákání. Aplikace uživateli zobrazuje data o jeho výkonu, která sbírá ze zmiňovaného senzoru na botě. Bohužel je aplikace použitelná pouze se senzorem a tak není možné zjistit více o praktickém fungování aplikace.

Je to však velice zajímavé téma, které je v kapitole 6 Možnosti rozšíření dále rozvíjeno.

2.4.3 Nеспециализované aplikace

Kvůli absenci specializovaného systému se mnoho týmu uchyluje k použití nспециализovaných aplikací, do kterých se informace vyplňují předzpracované.

2.4.3.1 DRIVN

Aplikace DRIVN² dává sportovnímu týmu možnost zadat informace do mobilní aplikace a pak sledovat, jak se v čase mění. Tyto informace jsou, až na výjimky, v textovém formátu. Služba podporuje role sportovce a trenéra, a tak je možné aplikaci využít i pro sdílení informací trenérovi.

Nevýhoda je, že se data musí před zadáním předzpracovat. Nelze vyplnit jednotlivá kola tréninku, ale jen součty dílčích částí. Například v této aplikaci není možné vyplnit, jaké byly obtížnosti jednotlivých kol, ale pouze celkovou obtížnost jako součet všech obtížností za kolo, které je nutné předpočítat.

¹<http://www.fxe-gear.com>

²<https://drivn.today>

2.4.4 Trampingo

Vhodný kandidát může být výstup autorovy bakalářské práce, pojmenován jako Trampingo.

Trampingo mělo za cíl co nejvíce zjednodušit a zrychlit vyplňování tréninků ve skocích na trampolíně. Cíl se podařilo naplnit jen z části. Kvůli náročnosti, která nebyla na začátku práce správně odhadnuta, bylo nutné slevit z cíle a nebyl vytvořen plně specializovaný systém. Trampingo mělo funkcionality pro vyplnění tréninku po kole, ale nikoli po prvcích. Proto i při použití Trampinga musel skokan informace o počtu skoků a obtížnosti kola předpočítat.

Z tohoto důvodu není ani Trampingo vhodné řešení pro dnešní potřeby skokanů.

2.5 Netechnická analýza

Díky analýze závodů a tréninků je možné lépe specifikovat dílčí části pro dosažení zadaných cílů.

Možnost zadat kolo po prvcích Hlavní funkcí by měla být možnost zadávat kola, respektive pokusy, po jednotlivých prvcích. Pak by skokanovi stačilo vyplnit kolo, respektive vazbu, a vše ostatní za něj může spočítat systém.

Nastavitelná klávesnice prvků K tomu, aby mohl skokan vyplňovat kola po prvcích, je potřeba, aby měl vlastní klávesnici prvků, z které bude prvky vybírat. Do té by měl mít možnost prvky přidávat a odebírat podle svojí úrovně a vývoje.

Možnost zadat čas letu k pokusu Čas letu je součástí závodní známky. Proto je pochopitelné, že skokana zajímá, jak si na tom v tréninku vede. Systém by měl umožňovat přidat čas letu k jednotlivým pokusům.

Podpora zápisu více druhů sestav Protože skokan závodí se třemi sestavami, je důležité, aby v systému byl schopný tyto tři sestavy zadat a pak sledovat kolikrát je skočil. Čím více je skokan v tréninku skáče, tím vzrůstá jistota a tím i všechny ostatní hodnocené části.

Seznam připravených vazeb Aby měl skokan zadávání sestav co nejjednodušší, měl by mu systém nabídnout seznam vazeb, které si zadefinuje. V rámci těchto vazeb nemusejí být pouze sestavy, ale i často skákané vazby pro zrychlení zapisování kol.

Maximálky Protože je tento typ kola tak často skákaný, systém by měl podporovat jeho vyplnění v tréninku.

2.6 Platformy pro jednotlivé role

Skokan potřebuje, aby mohl zařízení, na kterém běží systém, rychle vzít, zaplat kolo a zase odložit. K tomu se nejvíce hodí telefon. Jako mobilní aplikace se skokanovi podaří velice rychle a opakovaně systém otevřít a kolo vyplnit.

Pro trenéra, který nepotřebuje tak často se systémem interagovat, ale hodila by se mu větší plocha pro přehlednost, je vhodný tablet. Je mobilní, takže si ho může brát na trénink, ale zároveň je dostatečně velký, aby na něm mohl přehledně vidět trenérská analytika. Pro skokana by mohl být tablet až příliš nepraktický svojí velikostí.

2.7 Technická analýza

V rámci technické analýzy bude rozšířena už existující analýza z bakalářské práce.

2.7.1 Požadavky

Tato část je zaměřena na technické požadavky cílů této práce. Zbylé požadavky na funce typu „Přihlášení“, „Registrace“ atd. naleznete v bakalářské práci [1] a nebudou dále rozváděny.

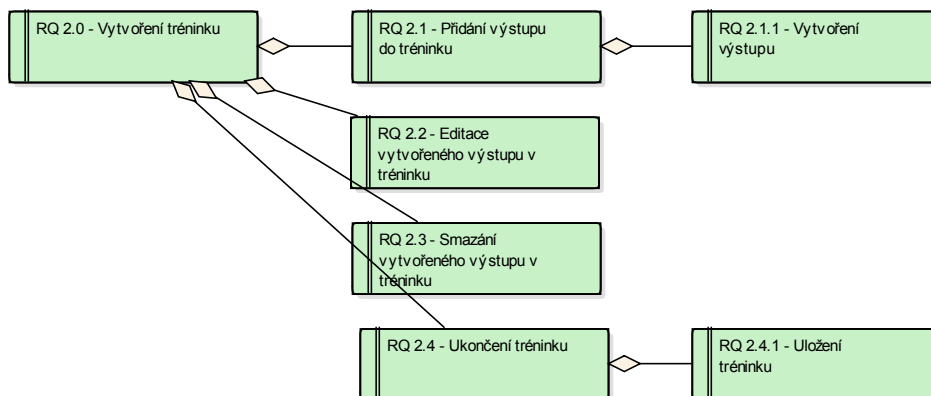
2.7.1.1 Funkční požadavky

První cíl ovlivňuje hlavně způsob vyplňování tréninku v systému, proto budou požadavky spojené s tímto cílem vycházet z požadavku „Vytváření tréninku“. Diagram tohoto požadavku je, pro připomenutí, na obrázku 2.1. V diagramu jsou zmiňovány výstupy, kterými se myslí, v kontextu této práce, kola. Výstup byl v této práci přejmenován na kolo pro lepší pochopení a reflektování skutečnosti.

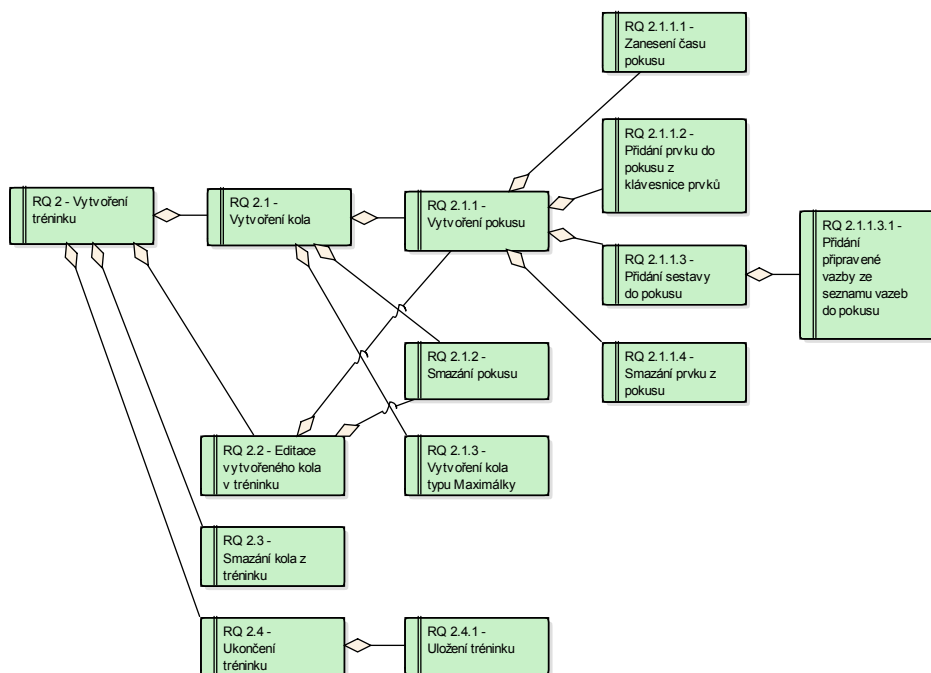
Po rozšíření starého požadavku o všechny nové funkce vypadá diagram tak jak na obrázku 2.2.

V novém diagramu se oproti starému přejmenoval výstup na kolo, aby lépe zapadal do názvosloví této práce. Pro zjednodušení se spojil požadavek na přidání a vytvoření kola, neboť to v reálu je ta samá věc. Objevilo se mnoho požadavků právě za vytvářením kola, které reflektují potřebu skokana vyplňovat trénink po prvcích. Kolo je navíc rozděleno na pokusy tak, aby skokan mohl zadat ke každému pokusu čas. Dále se objevily požadavky na vytvoření kola typu maximálky.

Zajímavé jsou požadavky „RQ 2.1.1.2 - Přidání prvku do pokusu z klávesnice prvků“ a „RQ 2.1.1.3.1 - Přidání připravené vazby ze seznamu vazeb do pokusu“. Tyto dva požadavky naznačují, že bude potřeba vytvořit dvě nové struktury, které se budou o prvky a vazby starat. Požadavky na nové struktury jsou na obrázcích 2.3 a 2.4.

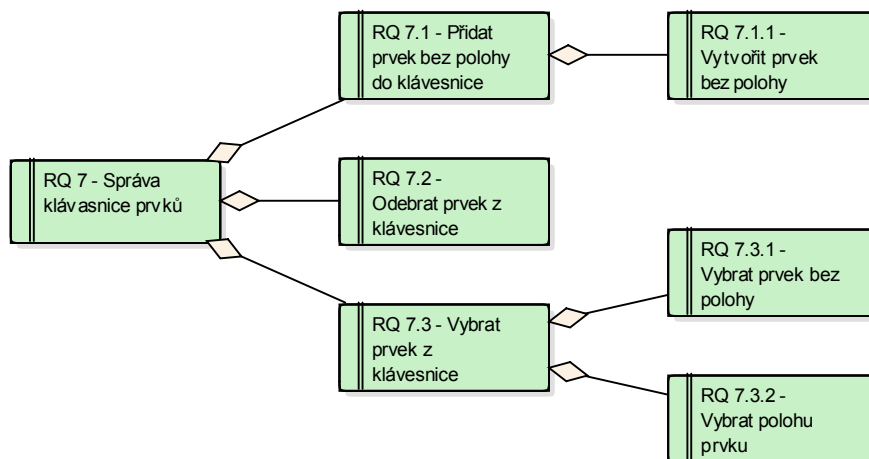


Obrázek 2.1: Funkční požadavky pro Vytváření tréninků z bakalářské práce

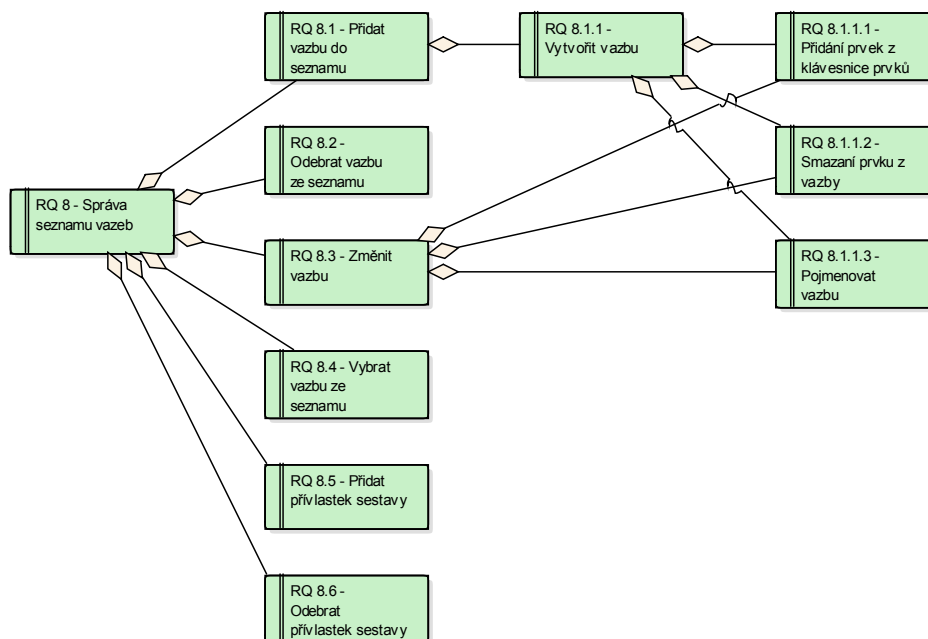


Obrázek 2.2: Rozšířené funkční požadavky pro Vytváření tréninků

2. ANALÝZA



Obrázek 2.3: Funkční požadavky pro Správu klávesnice prvků



Obrázek 2.4: Funkční požadavky pro Správu seznamu vazeb

Každý prvek může být skočen ve třech různých polohách, přičemž každá poloha z toho udělá trochu jiný prvek. Ve své podstatě jsou ale všechny tři varianty tohoto prvku, co do počtu salt a vrutů, stejné. Proto je v požadavcích na klávesnici prvků zaveden pojem „prvek bez polohy“, s kterým celá klávesnice operuje. Tento prvek bez polohy se používá proto, aby se jeden a ten samý prvek neopakoval v klávesnici zbytečně třikrát. Ve chvíli, kdy si chce skokan vybrat prvek, tak si nejdříve vybere prvek bez polohy a až v dalším kroku polohu. Tím se v klávesnici ušetří mnoho místa a celá klávesnice se zpřehlední.

Seznam vazeb nemá žádné překvapivé požadavky. Je vidět, že uživatel může vazbu při vytváření pojmenovat, aby se mu pak lépe hledala v seznamu, editovat a mazat. Navíc může k vazbě přidat přívlástek, zda je vybraná vazba sestava nebo není. Pokud pak v tréninku vyplní tuto vazbu, respektive sestavu, bude se mu započítávat do celkových skočených sestav.

Z požadavků „RQ 2.4.1 - Uložení tréninku“ nebo starého požadavku „RQ 3 - Navázání spojení mezi trenérem a skokanem“ (diagramy všech požadavků naleznete v příloze C) je znát potřeba mít data na serveru. Ten funguje jako prostředník pro komunikaci mezi skokanem a trenérem a úložiště skokanských tréninků. Všechny požadavky na server už jsou zmíněny v jiných požadavcích, například dříve zmíněné požadavky na ukládání tréninku a spojení uživatelů, proto není potřeba je dále rozebírat. Jedinou změnou je pak přidání serveru do nefunkčních požadavků vedle mobilních aplikací pro jednotlivé role.

2.7.1.2 Nefunkční požadavky

Nefunkční požadavky se oproti bakalářské práci výrazně nezměnily. Pořád platí, že obě role budou mít svoji mobilní aplikaci a navíc byl přidán požadavek na server. Nefunkční požadavky jsou zobrazeny na obrázku 2.5.

2.7.2 Rozšíření modelových tříd

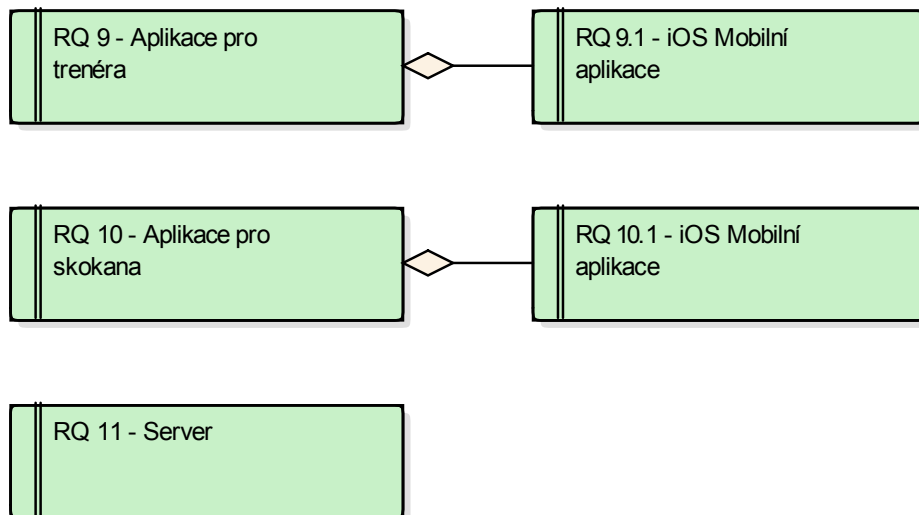
S aktualizovanými požadavky je potřeba zaktualizovat i modelové třídy. Pro připomenutí je model tříd z bakalářské práce na obrázku 2.6.

Nový model tříd je zachycen na obrázku 2.7. Stejně jako u požadavků, i zde se hlavně rozšiřovala třída „Trénink“. Znovu se kvůli udržení konzistence názvosloví přejmenovala třída „Výstup“ na „Kolo“.

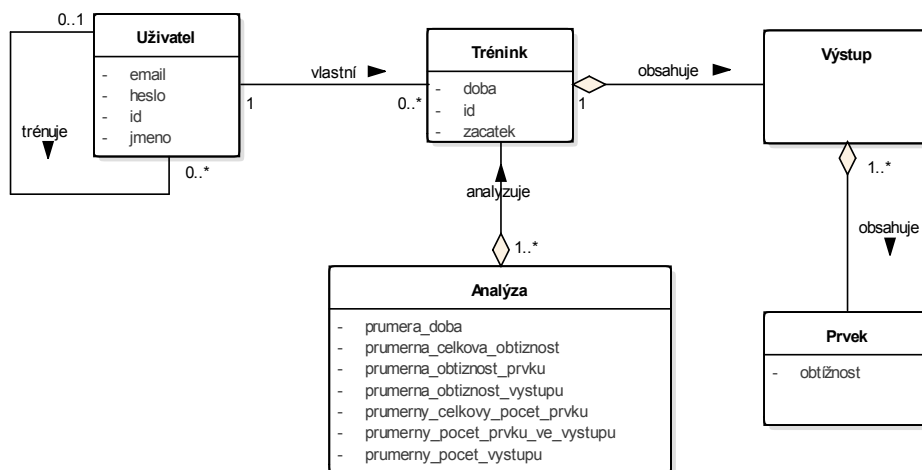
I když ve starém modelu už je třída „Prvek“, nese v sobě informaci pouze o obtížnosti. K potřebám této práce je zapotřebí, aby se stará třída „Prvek“ rozdělila na dvě třídy „Prvek bez polohy“, který v sobě nese informaci o počtu salt a počtu vrutů, a „Prvek“, který obohacuje „Prvek bez polohy“ právě o polohu. Z prvku už se dá spočítat obtížnost, která je v modelu také zachycena.

Dále je potřeba, aby se mezi třídy „Kolo“ a „Prvek“ přidala další třída „Pokus“. Ta s sebou nese informaci o čase pokusu a přívlástek, zda to je

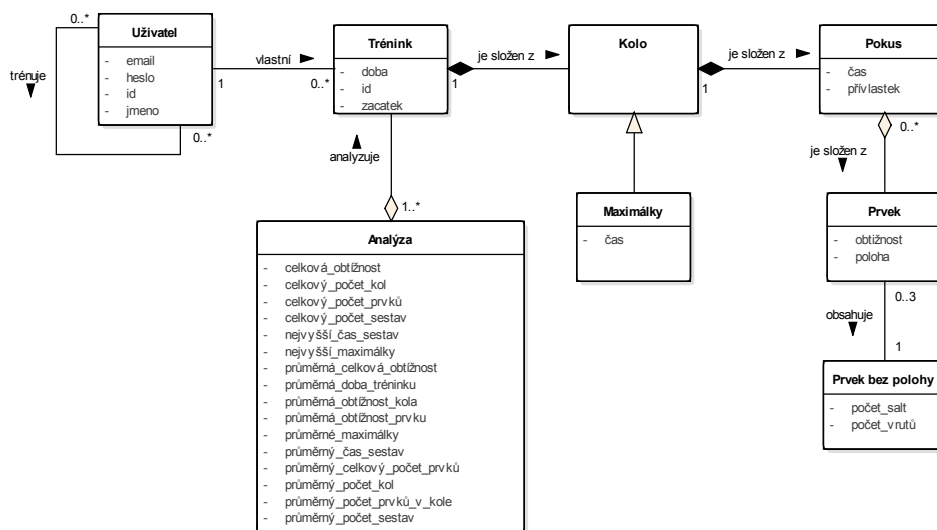
2. ANALÝZA



Obrázek 2.5: Nefunkční požadavky systému



Obrázek 2.6: Model tříd z bakalářské práce



Obrázek 2.7: Aktualizovaný Model tříd

nebo není sestava. Kolo je tedy složeno z několika pokusů a pokus je složen z posloupnosti prvků, přesně tak jak dříve vyšlo najevo z netechnické analýzy.

Do modelu byla přidána třída „Maximálky“. Ta přímo dědí ze třídy „Kolo“ a může být v systému za kolo nahrazena. Nese informaci jen o čase. Nic víc pro tuto třídu není potřeba.

Musela se rozšířit i třída „Analýza“, aby zahrnovala nově přidané třídy a vlastnosti systému. V tabulce 2.1 jsou sepsány všechny parametry analýzy s poznámkou zda jsou nové nebo staré. Jména parametrů jsou sama o sobě dostatečně popisná, proto v tabulce nebudou nijak rozváděna.

Jako poslední změnou je změna mohutnosti v případě vztahu skokana s trenérem. Ve staré práci byl model nastavený tak, že skokan mohl mít jednoho trenéra a trenér mohl mít více skokanů. V této práci se od tohoto omezení upustilo a i skokan může mít více trenérů. Je to z toho důvodu, že se může velice jednoduše stát, že skokan reálně trénuje s více trenéry. U malých dědí se mohou trenéři střídát s tím, že každý se zaměřuje na něco jiného. První se skokanem trénuje odrazy, druhý nové prvky a třetí provedení. Na druhou stranu u dospělých závodníků jeden může být osobní trenér a druhý reprezentační. V každém případě toto rozšíření dovolí skokanům i trenérům lépe spolupracovat.

2. ANALÝZA

Jméno parametru	Nový/Starý (N/S)
celková_obtížnost	N
celkový_počet_kol	N
celkový_počet_prvků	N
celkový_počet_sestav	N
nejvyšší_čas_sestavy	N
nejvyšší_maximálky	N
průměrná_celková_obtížnost	S
průměrná_doba_tréninku	S
průměrná_obtížnost_kola	S
průměrná_obtížnost_prvku	S
průměrné_maximálky	N
průměrný_čas_sestavy	N
průměrný_celkový_počet_prvků	S
průměrný_počet_kol	S
průměrný_počet_prvků_v_kole	S
průměrný_počet_sestav	N

Tabulka 2.1: Tabulka parametrů Analýzy tréninků

Návrh

3.1 Server

V bakalářské práci byl server navržen a naimplementován úplně od začátku. Bylo potřeba navrhnout databázi a aplikační server, na kterém bylo dále navrženo REST API, WebSocket komunikace a zpracovávání zpráv. Při implementaci vyvstávalo mnoho problémů jak technických, bezpečnostních nebo výkonnostních, které jsou v dnešních serverech už dávno vyřešeny. To celé bylo pro správné pochopení fungování serveru, a co vše jeho implementace obnáší, více než přínosné.

Na druhou stranu ve srovnání s dnešními požadavky na moderní server byl server z minulé práce v praxi nepoužitelný. Server trpěl několika bezpečnostními a optimalizačními problémy, které by reálnému použití bránily.

Výstup této práce má být nástroj, systém, který bude reálně použitelný. Z tohoto důvodu bylo zvoleno již existující řešení a nebyla snaha znovu vymýšlet kolo.

V informačních technologiích je tento způsob přemýšlení velice obvyklý. Pro většinu projektů je výhodnější použít subsystém, komponentu nebo knihovnu, na které už dříve pracovalo mnoho lidí a která má vyřešené všechny možné problémy, hlavně z pohledu bezpečnosti a výkonu.

Ačkoli požadavky na server a jeho funkce zůstanou stejné jako v bakalářské práci, jeho implementace bude jiná. Je tedy snaha najít službu nebo aplikaci, která by v tomto systému server nahradila. Při hledání vhodného řešení bylo potřeba specifikovat, co by takové řešení mělo splňovat. V analýze funkčních požadavků v sekci 2.7.1.1 jsou všechny požadavky na server zanalyzovány, ale roztroušené podle jiných logických celků. Pro lepší představu jsou popsány v seznamu níže.

Úložiště Pro ukládání a sdílení tréninků je potřeba, aby byl server hlavní úložiště celého systému. Uživatel pak není fixován pouze na jedno zařízení a nemůže dojít ke ztrátě dat v případě ztráty zařízení.

3. NÁVRH

Správa uživatelů V systému jsou přihlášení uživatelé, kteří se musejí nejdříve registrovat. Registrační údaje je potřeba někde schraňovat a to na serveru. Navíc by měl server implementovat procesy pro zapomenutá hesla a změnu přihlašovacích údajů.

Uživatelská práva Aby mohl trenér vidět tréninky svých spojených skokanů, je potřeba, aby server podporoval rozdělování práv na čtení obsahu. V případě, že se trenér a skokan spojí, trenér tím dostane právo na čtení skokanových tréninků.

Podpora real-time komunikace V požadavcích na systém je i real-time zobrazení právě probíhajících tréninků. K tomu je zapotřebí, aby server uměl navázat obousměrnou komunikaci s mobilními klienty, přes kterou bude propagovat změny v tréninku v reálném čase.

Podpora mobilních zařízení Protože je systém složen z mobilních aplikací, bylo by více než vhodné, aby použité řešení nabízelo i mobilní SDK pro danou platformu, přes kterou by aplikace se serverem komunikovaly a zajišťoval by například i synchronizaci v případě výpadku spojení. Tento požadavek není povinný, ale velice žádaný.

3.1.1 Realm

Jedno z možných řešení, které se v době hledání vhodné alternativy nabízelo, je Realm³. Realm vyvíjí a prodává databázové řešení pro mobilní aplikace (Realm Database) a implementaci aplikačního serveru, na který se databáze synchronizuje (Realm Object Server).

Realm Database je lehký objektový kontejner, který se chová jako klasická databáze s pár odlišnostmi. Stejně jako klasická databáze umožňuje data řetěžit, filtrovat, propojovat a ukládat. Ale oproti databázím data reprezentuje jako objekty, které jsou plně reaktivní a umožňují tak hladkou synchronizaci [3, p. 4].

Realm Object Server funguje jako middleware nebo samostatné úložiště, které přebírá schéma z připojené Realm Database a zajišťuje synchronizaci dat mezi klientem a serverem. Realm Object Server je nutno hostovat na vlastním serveru.

3.1.1.1 Výhody

Velkou výhodou tohoto řešení je existence lokální mobilní databáze, z které Realm automaticky propaguje a synchronizuje všechna data na server. Stejně to platí i obráceně. Pokud se něco změní v datech na serveru, například skokan přidá další kolo do tréninku, data se v mobilní aplikaci automaticky změní a všichni zaregistrovaní posluchači jsou o tom informováni.

³<https://realm.io>

Lokální databáze slouží i jako skvělý způsob zvládnání výpadků spojení. V případě výpadku může uživatel používat systém dále, tak jak je zvyklý, všechna data se ukládají lokálně a ve chvíli kdy se znovu spojí se serverem, jsou všechny změny propagovány. Navíc se přes síť posílají jen změny a nikoliv celé objekty, proto je komunikace rychlejší a méně náročná na přenos dat [3, p. 5].

S použitím lokální databáze odpadá nutnost serializovat a deserializovat data pro přenos a mapovat je na doménové třídy.

Tím, že celou komunikaci zajišťuje Realm pomocí synchronizace Realm Database na Realm Object Server, je zaručena bezpečná komunikace a optimální množství přenášených dat.

Realm, v produktu Realm Object Server nabízí i správu uživatelů a metody pro udělování a odebírání práv na data, která uživatel vlastní. Navíc podporuje více typů přihlášení. Hlavní je pomocí emailu a hesla, ale nechybí ani přihlášení pomocí Facebooku, Google+ nebo vlastního poskytovatele.

3.1.1.2 Nevýhody

I když má Realm omezení, jako například nepodporuje ukládání jednoduchých polí, kaskádové mazání, generické typy nebo fakt, že každý uživatel má své schéma synchronizované databáze a při větší změně datového modelu může dojít k nekonzistenci při pokusu o čtení dat uživatele, který svoji aplikaci neaktualizoval, je jako alternativa k vlastnímu serveru velice slibná.

Nakonec největším problémem je chybějící proces pro zapomenuté heslo, tedy možnost resetovat heslo i když uživatel není přihlášený. Tato možnost není dostupná ani pro administrátora. Vzhledem k tomu, jak moc podstatná funkce to je, nemohl být Realm použit jako aplikační server pro tento systém.

3.1.2 Firebase

Jako další možné řešení se nabízelo použít službu Firebase⁴. Firebase je SaaS (viz Seznam použitých zkratk), kterou v roce 2014 koupil Google [4] a nabízí mnoho zajímavých produktů od vývoje a testování, přes analytika, až po reklamu.

Nejvhodnějším produktem pro nahrazení vlastního serveru je jednoznačně Firebase Realtime Database (v době rešerše ještě neexistoval Cloud Firestore, který by byl vhodnějším řešením). Firebase Realtime Database je NoSQL databáze, která umožňuje ukládat a synchronizovat data v reálném čase [5].

3.1.2.1 Výhody

Velkou výhodou, oproti Realmu, je skvěle zpracovaná správa uživatelů. I zde je možno registrovat uživatele podle emailu a hesla, ale i Facebooku, Twit-

⁴<https://firebase.google.com>

3. NÁVRH

Požadavek	Realm	Firestore
Úložiště	A	N
Správa uživatelů	N	A
Uživatelská práva	A	N
Podpora real-time komunikace	A	A
Podpora mobilních zařízení	A	A

Tabulka 3.1: Tabulka porovnání Realm a Firestore

teru, Google+, telefonního čísla nebo dokonce zde existuje anonymní účet. Ale oproti Realmu jde zde proces, ve kterém si uživatel může nechat zaslat na email odkaz pro vytvoření nového hesla. Navíc se zde dají účty sjednocovat. Jinými slovy jeden uživatel se může přihlásit jak pomocí Twitteru, tak pomocí hesla, což v Realmu také nešlo.

Výhodou pro začínající projekt je i fakt, že Firestore je SaaS, který je hostován přímo Googlem. Tím odpadá potřeba spravovat a platit vlastní server, na kterém by aplikace běžela.

I Firestore Realtime Database má svoje mobilní SDK, přes které mobilní klienti komunikují se serverem. Navíc SDK nabízí metody pro sledování změn na serveru a tím se dostává funkcionality real-time propagace.

3.1.2.2 Nevýhody

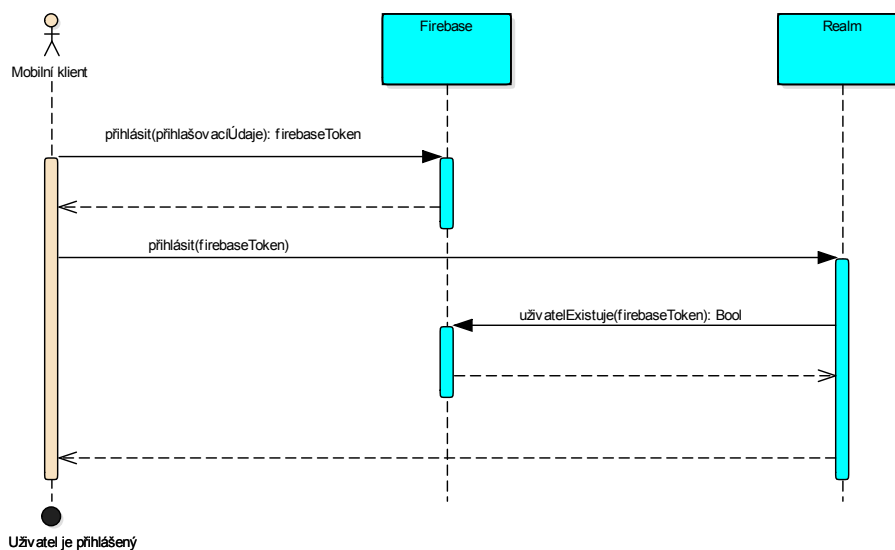
Jedna z hlavních nevýhod je reprezentace dat v databázi pouze ve formátu JSON. To znamená nejen to, že mobilní klienti musí všechna data nejdříve serializovat na JSON a až poté je mohou posílat na server, ale je potřeba jinak přemýšlet nad datovým modelem. To s sebou samo o sobě nese mnoho dalších komplikací.

Protože jsou data ukládána pouze ve formátu JSON, nemají defaultě žádné schéma. Tento nedostatek Firestore Realtime Database nahrazuje takzvanými „Rules“. V podstatě jde o jeden JSON soubor uložený na serveru, který definuje formát dat a pravidla pro jejich čtení a zápis.

3.1.3 Kombinace Realm + Firestore

V tabulce 3.1 je vidět, jak si které řešení vedlo ve stanovených požadavcích na server (v tabulce je A - splňuje, N - nesplňuje). Firestore skvěle splňuje požadavky na správu uživatelů, která Realmu chybí kvůli chybějícímu procesu pro obnovu zapomenutého hesla. Na druhou stranu Realm splňuje požadavky na úložiště a uživatelská práva, která jsou ve Firestore nesplněná.

Díky tomu, že Realm podporuje registraci, případně přihlášení přes externího poskytovatele, nabízí se řešení využít Realm jako hlavní úložiště dat a používat Firestore Realtime Database pouze pro správu uživatelů a ukládání



Obrázek 3.1: Sekvenční diagram přihlášení pomocí kombinace Realm a Firebase

uživatelských osobních dat. Celý proces přihlášení je zachycen v diagramu na obrázku 3.1.

Mobilní klient přihlásí uživatele přes Firebase SDK pomocí přihlašovacích údajů. Firebase vrátí token, pod kterým je ve službě Firebase uživatel jednoznačně identifikován. Tímto tokenem se, pomocí Realm SDK, uživatel přihlásí do Realmu. Ten si na serveru zkontroluje u služby Firebase, zda token patří k existujícímu uživateli. Pokud vše proběhne v pořádku, Realm najde uživatele podle tokenu a nastaví ho jako přihlášeného. Pro uživatelská data se pak mobilní klient dotazuje už jen na Realm.

3.2 Mobilní aplikace

Další krok v návrhu mobilních aplikací je rozdělit požadavky, které byly definovány v analýze, do jednotlivých obrazovek.

Jako hlavní a první obrazovka bude ta, na které se bude zobrazovat seznam tréninků. Skokan zde uvidí všechny svoje tréninky a trenér tréninky všech spojených skokanů. Z této obrazovky si uživatel zobrazí detail tréninku, přejde do nastavení a skokan zahájí nový trénink. Akcemi zobrazení detailu nebo zahájení tréninku se uživatel dostane na obrazovku detailu tréninku.

Obrazovka detailu tréninku slouží pro vyplňování tréninku, jeho editaci a zobrazení. Trenér nemůže tréninky nikdy editovat ani vyplňovat, takovou

3. NÁVRH

možnost má pouze skokan. Na této obrazovce je většina funkcí spojených s tréninkem. Přidat kolo, smazat kolo, přesunout kolo, editovat kolo, uložit trénink, smazat trénink a změnit datum tréninku. Kolem se myslí jak kolo složené z pokusů a prvků, tak i maximálky. Akce smazat nebo uložit trénink dostanou uživatele znovu na první obrazovku. Akce přidat kolo nebo editovat kolo otevřou obrazovku s detailem kola.

Detail kola je určen pro vytváření a vyplňování kol. Na této obrazovce se do kola vkládají jak pokusy, tak prvky. Ke každému pokusu lze vyplnit čas a prvky, které obsahuje. Prvky se vkládají výběrem prvku bez polohy z klávesnice prvků a následně polohy. Akce jsou tedy přidat pokus, smazat pokus, zadat čas pokusu, přidat prvek, smazat prvek, přidat vazbu. Akcí potvrdit kolo se pak uživatel, v tomto případě pouze skokan, který se na tuto obrazovku dostane, vrátí na obrazovku detailu tréninku.

Zpět k obrazovce seznamu tréninků. Z této obrazovky se uživatel dostane do nastavení. Jak název obrazovky napovídá, na této obrazovce si uživatel nastavuje všemožné nastavení aplikace. Od klávesnice prvků, přes způsob zobrazování prvků, až po nastavení spojení. Mimo jiné je to obrazovka, ze které lze kontaktovat podporu. Obrazovky, které jsou z nastavení dostupné, jsou přihlášení, registrace, navázání spojení, nastavení klávesnice prvků a nastavení připravených vazeb.

Pro lepší představu a vizuální reprezentaci všech obrazovek a jejich akcí je na obrázku 3.2 zachycen graf obrazovek a aktivit. V grafu jsou aktivity znázorněny zelenou barvou a obdélníky se zaoblenými rohy a obrazovky modrou barvou s ostrými rohy. Protože je aplikace pro trenéra technicky jen osekáná aplikace pro skokana, je pro obě aplikace vytvořen jen jeden graf. Aktivity nebo obrazovky, které se vztahují pro obě role jsou s plným ohraničením a ty, co se vztahují pouze pro skokana, jsou s čárkovaným ohraničením. Nakonec první obrazovka, na kterou se uživatel dostane po otevření aplikace je znázorněna zvýrazněným ohraničením.

Pro přehlednost jsou některé aktivity vynechány. Například maximálky by podle grafu mohly být reprezentovány novým kolem s deseti výskoky a časem. Nebo navázání spojení zde není dále rozebíráno a v grafu končí akcí „Navázání spojení mezi skokanem a trenérem“.

3.3 Uživatelské rozhraní

S navrženým rozložením požadavků, tedy aktivit, do obrazovek je dále možné navrhnout rozhraní aplikace.

3.3.1 Kalendář a seznam tréninků

Hlavní a první obrazovka, kterou uživatel po spuštění aplikace uvidí, je obrazovka se seznamem tréninků. Jedna možnost jak tuto obrazovku zpracovat je nechat jeden dlouhý seznam tréninků seřazený podle data zahájení. Tento

seznam by byl po chvíli používání velice dlouhý a nepřehledný. Je tedy potřeba tréninky seskupovat.

Jako nejvhodnější možnost se jeví využít kalendář, ve kterém si uživatel vybere den a zobrazí se mu seznam všech tréninků ve vybraném dni seřazený podle času zahájení. I kdyby skokan skákal 3x denně, seznam nebude nikdy tak velký, aby se stal nepřehledným. Pohyb v kalendáři je pro uživatele také velice intuitivní. Kalendář by navíc zobrazovat značky u dní, ve kterých je vyplněný trénink. Tuto informaci pak uživatel vidí na první pohled.

Jedna položka tréninku v seznamu k vybranému dni by měla obsahovat základní informace o tréninku. Uživatel už podle vybraného dne ví, k jakému datu je trénink vyplněn. Dalé by trénink měl obsahovat čas zahájení, celkový počet kol, prvků a celkovou obtížnost. Tyto informace jsou pro seznam tréninků dostačující a jsou to ty informace, které skokany a trenéry typicky zajímají.

Vybráním tréninku ze seznamu se otevře detail tréninku. Akce zahájení nového tréninku musí mít vlastní tlačítko, které bude zobrazeno pouze skokanovi. Stejně tak odkaz do nastavení. Tlačítko nového tréninku bude s největší pravděpodobností využíváno o mnoho více než tlačítko nastavení.

V roce 2013 byl na 1333-ti lidech pozorováno, jak drží a používají svůj mobilní telefon. 85% z nich používalo pro interakci s telefonem pouze jednu ruku [6]. V době, kdy se velikost telefonů každým rokem zvětšuje, je pro uživatele, používající telefon pouze jednou rukou, velice nepohodlné dosahovat na prvky, které jsou v horní části obrazovky. Proto je trend dávat důležité interaktivní prvky co nejniž, aby na ně uživatelé na velkých obrazovkách pořád pohodlně dosáhli.

Při aplikaci předchozího poznání je vhodné dát tlačítko zahájení tréninku co nejmíce dolů. Tlačítko pro nastavení může zůstat nahore v navigační liště, neboť nebude používáno tak často.

Na obrázku 3.3 je wireframe obrazovky kalendáře se seznamem tréninků.

3.3.2 Trénink

Na obrazovce tréninku bude skokan nejmíce používat akci vložení nového kola, proto bude tlačítko pro tuto akci co nejmíce na dosah, tedy co nejniž. Na obrazovce je potřeba mít dále tlačítka pro uložení a zrušení tréninku. Ty se použijí pouze jednou za trénink. Není třeba je dávat na rychle dostupné pozice a postačí, když budou v navigační liště. Tato obrazovka se zobrazí i když si bude uživatel zobrazovat detail tréninku nebo ho editovat. Jediné změny jsou pak v akcích v navigační liště a zde je tlačítko pro přidání nového kola zobrazené nebo ne. V tabulce 3.2 jsou tyto stavy přehledně zaneseny.

Z tabulky je například vidět, jak může uživatel trénink smazat. Zobrazí si detail tréninku, začne ho editovat a jako levá akce se mu pak zobrazí možnost trénink smazat. U pravé akce ve stavu detailu je * neboť pokud si detail

Stav tréninku	Levá akce	Pravá akce	Tlačítko nového kola
Probíhající	Zrušit	Uložit	Zobrazené
Detail	Zrušit	Editovat*	Nezobrazené
Editace	Smazat	Hotovo	Zobrazené

Tabulka 3.2: Tabulka srovnání akcí obrazovky tréninku v různých stavech

zobrazuje trenér, tak se mu na místě této akce nezobrazí nic. Trenér nemůže nijak upravovat data svých skokanů.

Obrazovka musí zobrazovat informace o tréninku a seznam kol. Informace o tréninku, které by uživatel potřeboval vědět jsou tyto: datum, čas zahájení, doba, počet kol, počet prvků, celková obtížnost, počet jednotlivých typů sestav. Tyto informace nejsou interaktivní, uživatel na ně tedy nemusí dosáhnout, a proto budou umístěny co nejvýše.

Jedno kolo v seznamu by mělo zobrazovat všechny pokusy s časy a jejich prvky. Navíc, pro přehlednost a zjednodušení, může kolo zobrazovat obtížnost a počet prvků daného kola. Akce pro editaci, smazání nebo opakování, budou schované v menu, které se zobrazí u konkrétního kola jeho přetažením vpravo. Protože počet pokusů a prvků není omezen, seznam se bude zobrazovat jako horizontálně posouvatelný seznam v řádku kola. Seznam kol se bude posouvat vertikálně, proto nedojde ke kolizi gest pro posun seznamu kol a seznamu prvků v kole.

Na obrázku 3.4 je wireframe obrazovky probíhajícího tréninku.

3.3.3 Kolo

Hlavní úkol obrazovky pro vytváření kol je rychlé zadávání prvků. Bude obsahovat klávesnici prvků, seznam předvyplněných vazeb a právě skládané kolo.

Na obrazovce jsou dvě akce, které posunou uživatele zpět do tréninku. Jedna pro potvrzení kola a druhá pro zrušení zadávání kola. První bude jednoznačně používanější, proto bude umístěna ve spodní části obrazovky. Druhá akce je pak nahoře v navigační liště. Navíc je vlevo, kde je typicky umístěno tlačítko zpět. Podobný význam tady má i tlačítko zrušit.

Hlavní komponenta obrazovky je zmíněná klávesnice prvků. Ta je také ve spodní části obrazovky, jakožto element, se kterým bude skokan nejvíce interagovat. Klávesnice zobrazuje prvky bez poloh jako tlačítka na klasické počítačové klávesnici. V tomto případě jsou klávesy vždy čtyři vedle sebe a celá klávesnice se dá posouvat vertikálně. To proto, že není omezení na maximální počet prvků v klávesnici. Jako poslední tlačítko je vždy tlačítko „+“, které otevře obrazovku, ve které si uživatel prvek manuálně vyplní a následně se mu vloží do klávesnice a automaticky vybere pro další krok, kterým je výběr polohy. Po výběru prvku bez polohy z klávesnice se přes celou klávesnici

3. NÁVRH



Obrázek 3.3: Wireframe obrazovky kalendáře se seznamem tréninků

Obrázek 3.4: Wireframe obrazovky prázdného tréninku

Obrázek 3.5: Wireframe obrazovky vytváření nového kola

zobrazí 3 možné polohy prvku, ze kterých si skokan vybere. Vybraný prvek s polohou se pak zobrazí jako poslední prvek v posledním pokusu.

Seznam předvyplněných vazeb se zobrazí posunutím klávesnice prvků vlevo. Tento seznam je vertikálně pohyblivý výčet jmen všech vazeb. Každá vazba zabírá celý řádek a při výběru se v kole zobrazí jako nový pokus. Pokud má vazba přívlástek sestavy, přidáním tohoto kola se v detailu tréninku přičte k celkovému počtu sestav daného typu.

Samotné zobrazení kola je pak kolekce prvků, kde každý pokus začíná časem, který lze pro daný pokus nastavit. Všechny prvky jsou vždy vidět a pokud se nevejdou na jeden řádek, jsou zalamovány na nový. Narozdíl od kola v tréninku, zde potřebuje skokan vidět co už má nebo nemá vyplněné. Kdyby i zde byl horizontálně pohyblivý seznam prvků, skokan by při dlouhé vazbě neviděl začátek a nebyl by si jistý, zda ve vazbě nemá chybu nebo na něco nezapoměl.

Na obrazovce je ještě akce, která smaže poslední prvek nebo poslední pokus, pokud už v sobě nemá žádný prvek. Navíc jsou zde pomocné akce pro vytvoření nového pokusu a opakování již vyplněných prvků v pokusu.

Na obrázku 3.5 je wireframe obrazovky vytváření kola.

3.4 Navázání spojení skokan - trenér

Po navržení uživatelského rozhraní a popsání tří hlavních obrazovek, stojí za zmínku ještě jeden proces, který tu nebyl zmíněn. Proces navázání spojení je v aplikacích navržen tak, že trenér má vlastní vygenerovaný QR kód, který skokan svojí aplikací načte a tím tenérovi nastaví čtecí práva na skokanovy data. Vzhledem k tomu, že se trenér a skokan v reálném životě potkávají, nedává smysl nutit skokana, aby trenéra nejdříve vyhledával a poté mu posílal žádost o přátelství, na kterou by později trenér odpověděl. Tím, že jsou oba na jendom místě, je řešení pomocí načítání QR kódů rychlejší. Navíc zabráňuje žádostem o spojení, které jsou pro trenéra nesmyslné a mohou vycházet například ze známosti jeho osoby ve sportu.

3.5 Wireframe

V rámci práce byl vypracován i interaktivní wireframe hlavních obrazovek mobilních aplikací a přechodů mezi nimi. Důraz byl kladen na potvrzovací hlášení a nastínění fungování aplikace. Wireframe měl odhalit nedostatky návrhu ještě dříve, než měly být aplikace implementovány. Nakonec se nepřišlo na nic zásadního. Obrázky 3.3 až 3.5 jsou převzaty z tohoto wireframu.

Wireframe byl vypracován ve webovém nástroji NinjaMock⁵, který nabízí i verzi zdarma, ve které přidává do exportovaných PDF vodoznak. Proto jsou i na obrázcích z wireframu tyto vodoznaky vidět. Wireframe je klikatelné PDF a je součástí příloženého CD.

⁵<https://ninjamock.com>

Implementace

4.1 Způsob vývoje

V dnešní době je více než 1000 metodik, které popisují způsob vývoje softwaru, kde některé jsou neveřejné v rámci firmy a jejího know-how [7, p. 119]. Tolik různých metodik poukazuje na fakt, že v dnešní době neexistuje jeden správný způsob, jak přistupovat k vývoji softwaru. Kdyby takový bezchybný a univerzální způsob existoval, všichni by jej používali a nevznikalo by takové množství metodik.

I když je metodik celá řada, všechny se shodnou na základních dílčích částech, kterými projekt musí projít. Těmi jsou: sběr požadavků, analýza, návrh, implementace, testování, nasazení a údržba.

Způsoby vývoje můžeme rozdělit do tří nejobecnějších skupin:

Ad Hoc Tento způsob vývoje se nedrží žádného definovaného procesu.

Strukturované Všechny dílčí části se plní sériově. Jedna část začíná až ve chvíli, kdy skončí část před ní. Do této skupiny spadá i známý waterfall.

Agilní Podstatou metodik v této skupině je průběžné dodávání výsledků v iteracích. Zákazník je zde zapojován daleko více a má větší šanci zasahovat do vývoje projektu. Metodiky z této skupiny jsou více flexibilní.

Protože Ad Hoc není žádná metodika, ale způsob vývoje bez metodiky, nebude v tomto textu dále rozvíjen.

Strukturované a agilní metody vývoje se liší v mnoha důležitých faktorech vývoje. Od počtu členů týmu, přes zapojení klienta, až po pojejí testování. Pro přehlednost jsou vybrané rozdíly mezi agilními a strukturovanými metodikami zaneseny v tabulce 4.1.

V roce 2014 byl firmou Ambyssoft proveden průzkum použitých metodik ve vývojových týmech. 52 % týmů uvedlo, že používají agilní metodiku pro vývoj softwaru. Navíc průzkum ukazuje, že tými, které se agilní metodiky

4. IMPLEMENTACE

	Strukturované	Agilní
Fáze	Projekt je rozdělen do fází podle typu činností.	Projekt je rozdělen do „sprintů“, podle funkcionalit.
Tým	Velký tým vývojářů, kteří dostanou úkol na vyřešení.	Malý tým složený z různých expertů na problematiku projektu. Nemusí být jen vývojáři.
Čas	Od začátku projektu je sestaven pevný časový harmonogram, který se plní a je dobře znám konec vývoje.	Není jasné jak dlouho projekt potrvá.
Klient	Je zapojen na začátku a poté až na konci vývoje.	Je zapojován po celou dobu vývoje.
Výstupy	Jeden výstup na konci projektu.	Průběžné výstupy, na základě kterých tým dostává zpětnou vazbu od zákazníka.
Testování	Jedna z fází projektu, realizuje se až na konci implementace.	Považuje se za součást implementace a je s ním souběžná.
Požadavky	Na začátku projektu musejí být pevně dané a dále je není možné měnit.	Na základě zpětné vazby z mezi výstupu je možné požadavky měnit.

Tabulka 4.1: Tabulka srovnání strukturovaných a agilních metodik

držely, měly lepší výsledky v různých faktorech, než ostatní. Například kvalitu dodaného produktu ohodnotily týmy s agilním přístupem 4.4 body z deseti a týmy s tradičním přístupem pouze 3.7 body. U času dodání je to ještě větší rozdíl mezi metodikami. Agilní týmy hodnotily splnění času dodání 2.6 body a tradiční týmy jen 0.9 body. Spokojenost zákazníků byla u agilních týmů hodnocena 4.8 body a u tradičních pouze 3.2 body [8].

Firma Ambyssoft na začátku roku 2018 provedla další průzkum úspěšnosti IT projektů vedenými různými metodikami. Průzkum ukázal, že agilní metodiky mají úspěšnost 55 %, kdežto tradiční metodiky (strukturované) dosahují úspěšnosti pouze v 29% případech. To je rozdíl 26 % mezi metodikami [9].

Jako nejméně úspěšná skupina metodik dnešní doby, má agilní způsob vývoje další výhodu a tou je, že není potřeba znát všechny požadavky na začátku projektu. Tento projekt se snaží vytvořit produkt, který ještě neexistuje a není jasné, jaké požadavky na něj budou kladeny. Je potřeba, aby se projekt vyvíjel v iteracích a aby se postupně zapracovávaly zpětné vazby uživatelů. Z tohoto důvodu se mohou požadavky nebo priority projektu několikrát za vývoj změnit nebo doplnit. To by v tradičním způsobu vývoje nebylo možné.

4.1.1 Agilní vývoj

Agilní metodiky jsou založené na iterativním a inkrementálním vývoji. Umožňují rychlý vývoj softwaru a zároveň dokáží reagovat na změnu požadavků v průběhu vývojového cyklu. Podle těchto metodik se správnost systému ověří jedině pomocí rychlého vývoje, předložení zákazníkovi a následných úprav dle zpětné vazby. Na obrázku 4.1 je zachycen rozdíl v procesu vývoje tradiční metodiky a agilní iterativní metodiky.

Agilní metodiky sdílí takzvaný „Manifest Agilního vývoje softwaru“ [10], který definuje čtyři hodnoty, kterých by se měl agilní vývoj držet. Těmi jsou:

1. **Jednotlivci a interakce** před procesy a nástroji
2. **Fungující software** před vyčerpávající dokumentací
3. **Spolupráce se zákazníkem** před vyjednáváním o smlouvě
4. **Reagování na změny** před dodržováním plánu

Z tohoto manifestu vychází dvanáct principů agilního vývoje [11].

4.1.1.1 SCRUM

Typický a nejrozšířenější agilní metodika je SCRUM. Jedna iterace se nazývá „sprint“ a trvá přibližně jeden až dva týdny. Každý produkt má svůj takzvaný backlog, který funguje jako seznam všech požadavků na funkcionality, které budou v projektu potřeba.

První fází je naplánování sprintu, ve kterém se vyberou požadavky z projektového backlogu a přesunou se do backlogu daného sprintu. Jak se požadavky vybírají je na daném týmu.

Tým postupně vyvíjí jednotlivé požadavky ze sprintového backlogu. Aby se udržela maximální informovanost týmu, jsou na každý den naplánovány informativní schůzky celého týmu, na kterých se probírá co bylo předešlý den uděláno a co je v plánu na následující den.

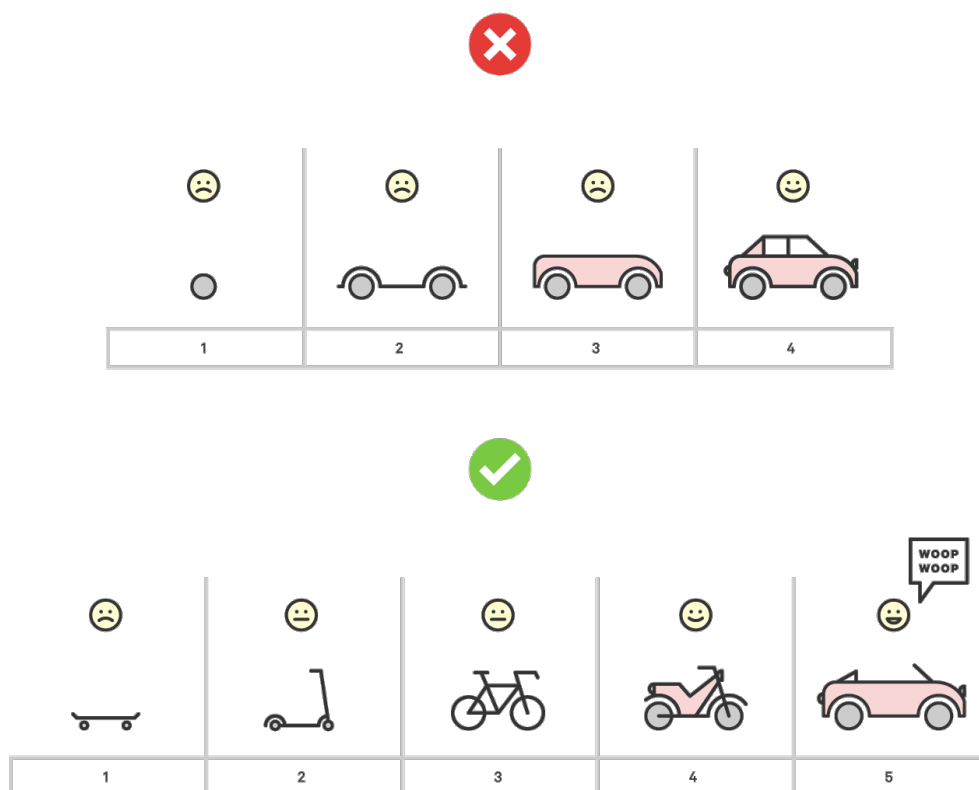
Po ukončení sprintu následuje fáze zhodnocení, ve které tým hodnotí proběhlý sprint. Z této fáze může dojít ke změně v plánování dalšího sprintu nebo přidání dalších požadavků do projektového backlogu.

Jako poslední fáze je vydání výstupu proběhlého sprintu jako další verze produktu, kterou mohou uživatelé testovat a jejich zpětná vazba je zapracována do požadavků v projektovém backlogu.

4.1.2 Lean development

Lean development je aplikace metody lean manufacturing, kterou ve společnosti Toyota vyvinuli Taiichi Óno a Šigeo Šingó [12, p. 24], do domény softwarového vývoje. Řadí se do skupiny agilních metodik. Hlavními pilíři této metodiky je sedm pravidel, podle kterých se má vývoj řídit. Těmi jsou:

4. IMPLEMENTACE



Obrázek 4.1: Obrázek vývoje produktu pomocí tradiční metodiky a pomocí agilní iterativní metodiky (zdroj: <https://www.digitalogy.co/single-post/2017/09/18/Minimum-Viable-Product-MVP---Its-Importance-Pitfalls>)

Odstranit vše co nepřináší hodnotu Lean development definuje hodnotu jako něco, za co bude uživatel ochoten zaplatit. Vše ostatní je považováno za odpad, kterého se musí projekt zbavit. V doméně softwarového vývoje se často hledání odpadu nahrazuje za vytváření a dodávání hodnot.

Učit se v průběhu Uživatelská zpětná vazba a krátké iterace zakončené aktualizací produktu jsou klíčové procesy při vývoji správných funkcionalit. Pro umožnění co nejkratších iterací je potřeba mít automatické testy, které by zajišťovali kvalitu.

Rozhodovat se co nejpozději Vývoj softwaru je vždy spojen s jistou nejistotou a neznalostí. Proto je pro lepší rozhodování dobré, odkládat rozhodnutí co možná nejdále a rozhodovat se podle pozdějších vědomostí, než zakládat důležitá rozhodnutí na doměnkách. Iterace jsou v tomto

směru také podstatné. Umožňují sbírat nové vědomosti a v některých případech úplně odstranit nejistotu, která byla na začátku projektu.

Dodávat co nejrychleji V dnešním světě rychlého technického pokroku na trhu nevítězí nejkvalitnější, ale nejrychlejší. Rychlé dodávání znamená dřívejší sběr zpětné vazby. Schopnost rychlého dodání navíc oddaluje nutnost se rozhodovat. V případě, že je potřeba důležitého rozhodnutí uprostřed iterace a jednu iteraci je možné stihnou za týden, je možné se rozhodnout jistě později, než kdyby iterace trvala tři měsíce.

Zodpovědnost a důvěra v týmu V některých organizacích je stále zažité pravidlo, že vedoucí týmu má říkat svým podřízeným, jak mají dělat svoji práci. Pravidlo o zodpovědnosti a důvěře v týmu říká, že tým je složen z expertů, kteří by měli mít důvěru vedení a odpovědnost za svoje rozhodnutí.

Zaměřit se na celkový dojem Stejně tak jako je důležitý produkt, je důležitá i jeho prezentace, reklama a dostupnost. Z pohledu softwarového vývoje ale toto pravidlo hovoří o konzistenci mezi flexibilitou, udržitelností, efektivností a odezvou. V průběhu vývoje se přidávají nové funkcionality. Je důležité, aby si systém udržel konzistenci a snižoval komplexnost po celou dobu vývoje. K tomu je použit refaktoring.

Zaměřit se na výsledek Vždy je potřeba hledět na výsledný produkt. To se odvíjí i při měření efektivity. Při špatném nastavení měřítek výkonnosti dochází k sub-optimalizaci, kdy se členové týmu soustředí na tyto sub-optimalizace více, než na celkový produkt, který tím zpravidla degraduje. V průběhu vývoje se v systému kumulují chyby, které je potřeba opravit. Není vhodné opravovat pouze důsledek chyby, což se ve většině případů děje, ale její příčinu.

4.1.3 Způsob vývoje

Aplikováním principů Lean developmentu na tento projekt, vzniká list doporučení při vývoji.

1. Jednoduchý návrh
2. Vyvíjet v iteracích
3. Zapracovávat zpětnou vazbu
4. Kód doplňovat o automatické testy
5. Refaktorovat

Několik pravidel, které Lean development definuje, nejsou v tomto kontextu relevantní. Například pravidlo 5, zodpovědnost a důvera v týmu. Tato práce je výstupem jednoho člověka a tak nemá smysl mluvit o týmu. Dále je zbytečné mluvit o neefektivní komunikaci nebo schvalování postupu.

4.1.4 Test Driven Development

Potřeba doplňovat kód o automatické testy a refaktorovat vede k použití takzvaného test driven developmentu (TDD). Prvním krokem je definice funkcionality a následné napsání testu, který tuto funkcionalitu ověřuje. Poté přichází na řadu psaní kódu a nakonec úprava tohoto kódu.

Vývojový cyklus je pak složen z pěti bodů.

1. Napsat test
2. Spustit testy a ujistit se, že všechny neprojdou
3. Napsat vlastní kód
4. Kód automatickými testy prochází
5. Refaktorce

Napsat test První krok je napsání testů, které ověřují, zda kód, který je implementován, dělá to, co se od něj očekává. Toto první napsání testů je vlastně definicí požadavků na onu funkcionalitu. Tento krok je odlišný od přístupů, kde se píše testy až na konci, protože nutí vývojáře zaměřit se na požadavky už před psaním kódu.

Spustit testy a ujistit se, že všechny neprojdou Poté co jsou nové testy napsány, jsou spuštěny a všechny musí končit neúspěchem. Kdyby totiž v tomto kroku prošly jako splněné, znamenalo by to, že jsou buď testy špatně napsané nebo že budou splněny vždy a ztrácejí tím svůj smysl.

Napsat vlastní kód Po neúspěšném dokončení všech nově napsaných testů přichází na řadu psaní kódu. Smyslem je pouze zajistit to, aby při příštím spuštění nové testy prošly. Neklade se zde za cíl, napsat hned nejefektivnější a nejlegantnější kód.

Kód automatickými testy prochází Pokud kód napsaný v minulém kroce zajistí úspěšné splnění testů, znamená to také, že plní definované požadavky.

Refaktorce Posledním krokem je refaktorce kódu. Efektivita a elegance napsaného kódu je předmětem až tohoto kroku. Odstraňují se duplicity v kódu a kód se celkově upravuje do co možná nepřijatelnější podoby.

Opětovné spouštění automatických testů v tomto kroku umožňuje neustále ověřovat, že zásahy do dříve napsaného kódu nezpůsobily nikde žádné chyby ve funkcionalitě a že se refaktorace nestane spíše kontra-produktivní.

4.2 Mobilní aplikace

Z bakalářské práce nebyl při implementaci převzat žádný kód. Oproti analýze a nářhu, implementace nenavazovala na práci z bakalářské práce.

Jedna iterace vývoje mobilních aplikací obsahuje výběr funkcionalit, které budou implementovány, analýzu, návrh jak technický, tak rozhraní, implementaci s průběžným psaním automatických testů, refaktoring a nakonec vydání nové verze aplikace.

Aplikace jsou vydávány na Apple AppStore, kde si je může kdokoli zdarma stáhnout. V aplikaci je kontaktní email, na který je možné psát svojí zpětnou vazbu, stejně tak jako odkaz na projektový Facebook, který funguje jako informační web. Většina zpětné vazby chodila přes Facebook, který může být považován do jisté míry i jako marketingový nástroj. V rámci marketignu byl vytvořen i projektový účet na Instagramu. I když se projekt na marketing tolik nesoustředil, díky malé aktivitě na sociálních sítích se několik nových uživatelů, tedy potencialních poskytovatelů zpětné vazby, našlo.

Skokanská mobilní aplikace prošla po dobu vývoje jedenácti iteracemi. Trenérská pak jen šesti. Tabulku iterací s verzí a popisem naleznete v tabulce 4.2 pro skokana a v tabulce 4.3 pro trenéra. Funkcionality, které vzešly přímo nebo nepřímo z uživatelské zpětné vazby jsou v tabulce zvýrazněny tučným fontem.

Na obrázcích 4.2 až 4.7 jsou screenshoty z poslední verze (1.4.2) aplikace.

4.3 Server

Pro fungování serveru bylo potřeba pronajmou hosting, na kterém bude nasažen Realm a nakonfigurovat externí autentizaci pomocí Firebase. Firebase je služba, kterou není potřeba nikam nasazovat, pouze nakonfigurovat ve webové administraci.

Jako zprostředkovatel cloudu byl vybrán FORPSI Cloud⁶, který funguje jako IaaS a nabízí linuxové servery. FORPSI Cloud konkuruje svoji cenou a je pro účely tohoto projektu dostačující. Jako operační systém byl instalován CentOS 7. Na serveru je nainstalován manažer procesů PM2, který zajišťuje plynulý chod aplikace.

Jak je popsáno v sekci 3.1.3 na obrázku 3.1, Realm se musí umět dotázat Firebase na existenci uživatelského tokenu. Firebase nabízí SDK nejen pro

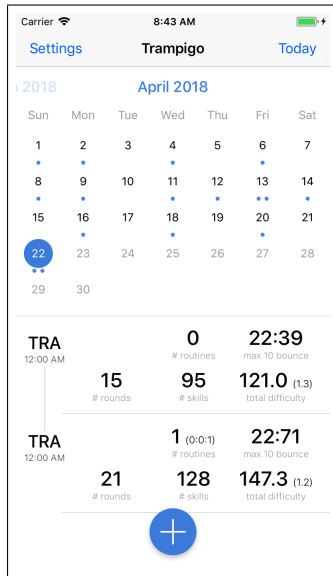
⁶<https://www.forpsicloud.cz>

4. IMPLEMENTACE

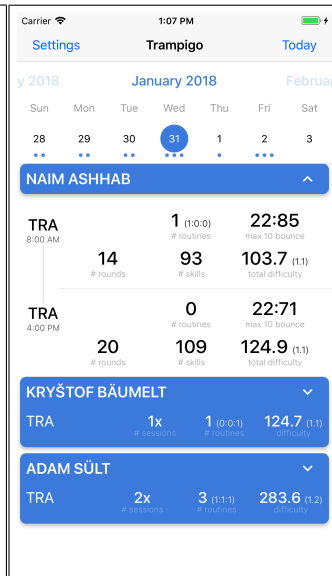
Iterace	Verze	Popis
1.	1.0	MVP se základními funkcionalitami. Kompletní vyplnění tréninku s koly, pokusy a prvky. Klávesnice prvků. Seznam předvyplněných vazeb. Seznam tréninků prezentován v kalendáři. Řádek tréninku obsahuje počet prvků, počet kol a celkovou obtížnost. Vše lokálně, bez synchronizace na server.
2.	1.0.1	Přidaná možnost nastavení způsobu zobrazení prvků.
3.	1.1.0	Přidán čas pokusu.
4.	1.2.0	Možnost přidání příznaku sestavy k předvyplněné vazbě. Ta se pak v tréninku započítá do počtu skočených sestav.
5.	1.2.1	Výskoky přestávají být počítány jako skoky. Mění se způsob zobrazení prvků bez salt a vrutů.
6.	1.3.0	Přidání uživatelských účtů. Synchronizace dat na server. Přidání možnosti vytvářet spojení mezi skokanem a trenérem. Přidán proces navázání spojení. Vydaná trenérská aplikace.
7.	1.3.1	Přidány měsíční analytika tréninků.
8.	1.3.2	Změna zobrazení času z XX,XX na XX:XX.
9.	1.4.0	Přidána možnost měnit kalendář z měsíčního na týdenní a obráceně. Vzniká více místa pro seznam tréninků. Do tréninkové buňky se přidávají nejvyšší maximálky za trénink a počet skočených sestav.
10.	1.4.1	Upravený data model pro zrychlení načítání.
11.	1.4.2	Přidaná obrazovka o nových funkcionalitách, která se zobrazí vždy jednou po aktualizaci. Uživatel rovnou vidí, co se v poslední verzi změnilo. Vedle celkové obtížnosti je přidána i průměrná obtížnost na prvek.

Tabulka 4.2: Tabulka iterací skokanské mobilní aplikace

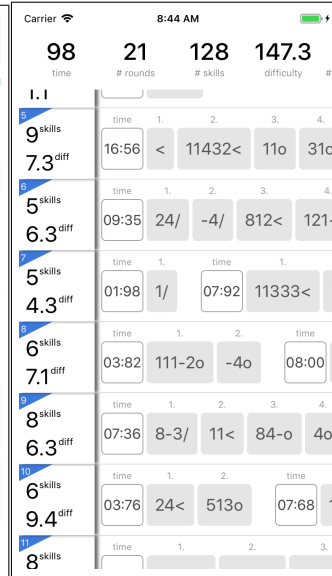
4.3. Server



Obrázek 4.2: Screenshot skokanského kalendáře se seznamem tréninků



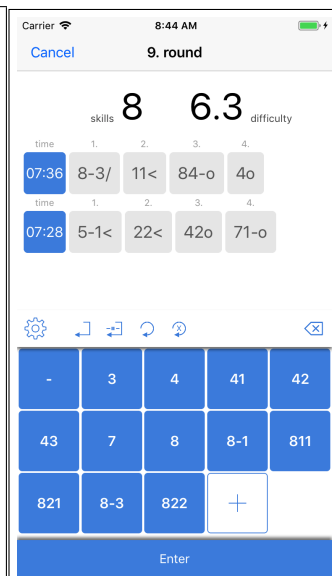
Obrázek 4.3: Screenshot trenérského kalendáře se seznamem tréninků



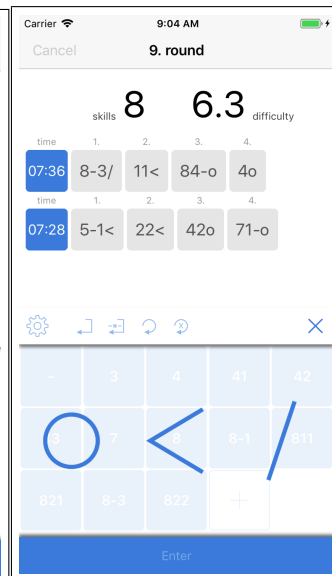
Obrázek 4.4: Screenshot obrazovky detailu probíhajícího tréninku



Obrázek 4.5: Screenshot obrazovky měsíčních analytik



Obrázek 4.6: Screenshot obrazovky vytváření nového kola



Obrázek 4.7: Screenshot obrazovky vybírání polohy prvku z klávesnice

4. IMPLEMENTACE

Iterace	Verze	Popis
1.	1.0	MVP se základními funkcionalitami. Seznam tréninků prezentován v kalendáři. Řádek tréninku obsahuje počet prvků, počet kol, celkovou obtížnost a jméno skokana. Uživatelké účty. Synchronizace na server. Proces spojování mezi skokanem a trenérem.
2.	1.1	Opravy chyb.
3.	1.3.2	Změna zobrazení času z XX,XX na XX:XX. Číslo verze synchronizované se skokanskou aplikací.
4.	1.4.0	Přidaná možnost měnit kalendář z měsíčního na týdenní a obráceně. Vzniká více místa pro seznam tréninků. Do tréninkové buňky se přidávají nejvyšší maximálky za trénink a počet skočených sestav. Seznam tréninků je agregovaný podél skokanů. Nejdříve je zobrazený seznam skokanů s tréninkem ve vybraný den, po prokliknutí seznam tréninků vybraného skokana. Prvek v seznamu zobrazuje počet tréninků, maximálky a počet sestav.
5.	1.4.1	Přidány měsíční analytika tréninků všech spojených skokanů.
6.	1.4.2	Přidaná obrazovka o nových funkcionalitách, která se zobrazí vždy jednou po aktualizaci. Uživatel rovnou vidí, co se v poslední verzi změnilo. Vedle celkové obtížnosti je přidána i průměrná obtížnost na prvek. Buňka v seznamu skokanů ve vybraný den zobrazuje místo maximálek obtížnost.

Tabulka 4.3: Tabulka iterací trenérské mobilní aplikace

mobilní klienty, ale i SDK psané v jazyce JavaScript, které je používáno v aplikačních serverech. Firebase JavaScript SDK se bude používat pro dotazování na existenci uživatele. V konfiguraci Realmu je možnost nastavit instanci, která bude zajišťovat přihlašovací a registrační logiku. Vlastní třída zajišťující logiku pak bude implementovat protokol, který definuje hlavní metodu „authenticateOrCreateUser“. V té se příchozí token zkontroluje s Firebase a v případě, že ve Firebase takový uživatel existuje, hledá se podle stejného tokenu i v Realmu. Pokud takový uživatel v Realmu není, je zaregistrován.

Jak Realm tak Firebase mají produkční a vývojářské prostředí.

Mezi mobilními klienty a serverem je nastavena bezpečná komunikace pomocí protokolu SSL. Jako certifikační autorita je v systému „Let’s Encrypt“⁷,

⁷<https://letsencrypt.org>

kteřá vydává certifikáty zdarma. Na serveru je nastavená automatická kontrola vypršení certifikátů a proces, který je obnovuje. Automatickou kontrolu spouští nástroj „cronab“. Cronab řádek pro kontrolu certifikátů:

```
@daily /usr/bin/certbot renew --quiet --renew-hook "pm2
restart trampigo; pm2 restart trampigo-dev"
```

Cronab je použit i pro automatické zálohy. Realm nabízí možnost manuální zálohy, která se pomocí cronabu denně generuje a posílá na AWS S3 uložště, ze kterého by byl v případě pádu nebo ztráty dat Realm obnoven. Cronab řádek pro zálohování dat z produkčního prostředí:

```
@daily aws s3 sync ~/trampigo/data s3://ros.trampigo.com
/trampigo-backup/ --exclude "realms/*" --exclude "
sync/migration/*" --exclude "*.lock" --exclude "*.
management/*" --delete
```

Finálně je pro dosažení serverové funkcionality bez implementace vlastního serveru potřeba služeb Firebase a AWS, aplikačního serveru Realm a cloudového poskytovatele FORPSI Cloud.

Uživatelské testování

Poslední verze skokanské mobilní aplikace byla podrobena důkladnějšímu uživatelskému testování. Testování se skládalo z heuristické analýzy a testování použitelnosti.

5.1 Heuristická analýza

Jedna z metod testování použitelnosti uživatelského rozhraní spočívá v procházení a hodnocení stránek odborníky. Odborníci zkoumají, jestli stránky splňují jednotlivá doporučení ohledně použitelnosti. Na konci je vytvořen seznam problémů a nedostatků řazený podle důležitosti [13].

5.1.1 Teorie

Odborník prochází všechny vyznačené a dostupné stránky aplikace. U všech vyhodnocuje zda splňují body, které jsou popsány v příloze D. Pokud expert najde rozpor v aplikaci a bodech, hned to poznamená do protokolu se zdůvodněním proč. Toto zdůvodnění se nesmí opírat o subjektivní názor a osobní preference experta.

5.1.2 Výstup analýzy odborníkem

1. Viditelnost stavu systému

- a) Vše v pořádku

2. Shoda mezi systémem a realitou

- a) Pro zadávání se v klávesnici prvků používají ikony typické pro daný sport
- b) Pouze ikony pro zjednodušení opakovaného zadání prvků nejsou na první pohled zřejmé (není nutné používat)

5. UŽIVATELSKÉ TESTOVÁNÍ

3. Minimální zodpovědnost (a stres)

- a) Při ukládání a mazání je vždy zobrazeno varovné upozornění
- b) Všechny údaje jdou upravit, či smazat

4. Shoda s použitou platformou a obecnými standardy

- a) Aplikace využívá standardů pro danou platformu (iOS)

5. Prevence chyb

- a) Výběr z klávesnice prvků, tj. nelze udělat chybu při zadávání
- b) Potvrzování akcí je v pořádku

6. Kouknu a vidím

- a) Žádné zbytečné akce viditelné nejsou

7. Flexibilita a efektivita

- a) Aplikace je dost konkrétně zaměřená
- b) Neobsahuje žádný pokročilý mód/nastavení

8. Minimalita (Klapky na očích)

- a) Velmi čistý design, je vždy vidět jen to potřebné

9. Smysluplné chybové hlášky

- a) Chybové hlášení jsou vždy dobře vysvětleny

10. Help a dokumentace

- a) Aplikace je jednoduchá a dobře pochopitelná. Nicméně některé prvky by si zasloužily nápovědu
- b) Pro někoho kdo nezná danou problematiku by se hodilo vysvětlení klávesnice prvků
- c) Způsob vypočítávání obtížnosti je nevysvětlen

V tabulce 5.1 jsou sepsány všechny nalezené problémy s jejich závažností. Závažnost 1 značí zanedbatelný a naopak 5 kritický problém, který uživatele omezuje.

5.2 Testování použitelnosti

Narozdíl od heuristické analýzy odborníkem, testování použitelnosti využívá reálných uživatelů k vyhodnocení kvality rozhraní. Uživatelům jsou předloženy úkoly, které v rámci testu plní. Testování uživatelé jsou po celou dobu sledováni a pozoruje se i jejich reakce, případně pozornost.

Problém	Pravidlo	Závažnost
Pouze ikony pro zjednodušení opakovaného zadání prvků nejsou na první pohled zřejmé (není nutné používat).	2	3
Přidání kola maximálek je nevýrazné a samotné zadání času by mělo být řešeno jinak.	5	4
Lze zadat i nereálné hodnoty doby maximálek	5	1
Neobsahuje žádný pokročilý mód/nastavení	7	2
Některé prvky by si zasloužily nápovědu. Pro někoho kdo nezná danou problematiku by pomohlo vysvětlení klavesnice prvku.	10	2
Způsob vypočítávání obtížnosti není vysvětlen	10	1

Tabulka 5.1: Tabulka nalezených chyb z uživatelského testování heuristickou analýzou

5.2.1 Nastavení testování

Testování bylo prováděno na zařízení iPhone 6s s operačním systémem iOS 11. Testující uživatelé dostali za úkol plnit připravené úkoly.

Testujícím uživatelům byla aplikace představena takto: *Aplikace je podpůrný systém pro skokany na trampolíně a slouží jako sportovní denník, do kterého si skokani vypisují přesně skoky, které na tréninku skočili.*

Cílová skupina uživatelů jsou skokani na trampolíně, proto bylo vybráno 5 skokanů z oddílu Sokol Kampa Praha tak, aby co nejlépe reprezentovali všechny závodní úrovně. U každého uživatele budou uvedeny informace o věku, úrovni skákání, profesi a zda je nebo není jako sportovec podporován organizací.

Testovací úkoly byly vytvořeny tak, aby otestovaly nejdůležitější funkce aplikace.

1. Zahájit trénink a vyplnit první 4 kola, kde první kolo jsou maximálky.
2. Upravit si klávesnici prvků tak, aby neobsahovala skoky, které skokan neskáče.
3. Smazat poslední vyplněné kolo.
4. Trénink uložit a změnit jeho začátek.
5. Vložit trénink na 18:00 ke veřejšímu datu.
6. Smazat vyplněný trénink.

Aby měli všichni testovaní uživatelé stejný počáteční stav, byla aplikace po každém testování obnovena do základního stavu. V tomto stavu nejsou žádné uložené tréninky a klávesnice prvků obsahuje pouze základní sadu prvků.

5.2.2 Adam Sült

Věk	23 let
Úroveň	Reprezentant
Ogranizace	Ministerstvo vnitra
Profese	Student VŠE v Praze

Úkoly

1. Nemá problém najít způsob jak zahájit trénink. Při zahájení tréninku se hned zobrazí vyplňování prvního kola. To musí zavřít, aby mohl jako první kolo vyplnit maximálky. Akci prakticky hned pochopí a vyplňování kola zavírá. Na obrazovce nachází text „Maximálky“ a po kliknutí vyplňuje čas. Na první pohled vypadá, že ví co dělá. Zbytek kol vyplňuje kliknutím na tlačítko „Nové kolo“. Po několika pohledech na klávesnici prvků mu dochází, co každá klávesa znamená a kolo vyplňuje velice rychle.
2. Uživatel je na obrazovce s tréninkem a hledá, kde se mění klávesnice. Pokouší se najít možnost na této obrazovce. Po chvíli zkouší přejít na obrazovku s klávesnicí a tam chybně kliká na možnost přidat nový prvek. Tu zavírá a po chvíli vidí ikonu nastavení a přes tu se dostane na žádanou obrazovku. Na této obrazovce už úkol dokáže vyřešit jednoduše.
3. Již dříve narazil na možnost vysunout řádek s kolem a tak neměl problém najít možnost pro smazání kola.
4. Bez problému trénink ukládá. Začíná editovat trénink a mění jeho začátek. Detail pak zavírá.
5. Bez problému vybírá den, na který chce trénink vložit, vybírá čas a vymyslí si data. Trénink pak ukládá.
6. Při editaci vyděl, že tlačítko smazat se zobrazí v editaci tréninku, proto jde rovnou do této obrazovky a trénink maže.

Souhrn

Aplikaci na první pokus dokázal velice dobře ovládat. Mátlo ho automatické otevření vyplňování prvního kola po zahájení tréninku, tato funkce by tam být nemusela. Jinak bez problému. Chválil rychlost vyplňování prvků a vnesl nápad, že by aplikace mohla mít připravené vazby, které by si skokan rovnou vyplnil. Tato funkce tam již je, ale není na první pohled zřejmá.

5.2.3 Karolína Sikorová

Věk	20 let
Úroveň	Bývalý závodník
Organizace	Žádná
Profese	Student ČZU v Praze

Úkoly

1. Trénink zahájila bez problémů. Dlouhou dobu strávila na automaticky otevřené obrazovce s vyplňováním kola. Dokonce byla potřeba nápověda pro pokračování. Po nápovědě však zbytek úkolu dokázala bez problému zvládnout. S klávesnicí prvků se ztotožnila velice rychle.
2. Již při plnění prvního úkolu našla tlačítko nastavení nad klávesnicí a tak věděla, kam má jít. V nastavení pak chvíli hledala možnost smazat, ale i tu nakonec bez pomoci našla.
3. Tento úkol se snažila vyřešit klikem na bílý čtverec, který jí napověděl, že řádek s kolem jde posunout a zobrazí se další možnosti. Jedna z nich i smazat kolo.
4. Uložení tréninku nebyl problém. Ale při změně času zahájení dlouho hledala tuto možnost na obrazovce s kalendářem a seznamem tréninků. Po několika nechápavých pohledech na způsob přišla sama.
5. Úkol splnila bez problémů.
6. Poučená, že na seznamu tréninků nejde udělat žádná jiná akce než zobrazit detail a zahájit trénink, šla rovnou do detailu a trénink smazala.

Souhrn

S vyplňováním tréninků neměla problém, způsob vyplňování si chválila. Vytkla neintuitivní místo změny data a smazání tréninku a vyskočení automatické obrazovky s vyplňováním prvního kola.

5.2.4 Šarka Korbelová

Věk	15 let
Úroveň	Reprezentant
Organizace	Centrum pro podporu talentované mládeže
Profese	Student gymnázia

Úkoly

1. Stejný problém jako u uživatele 5.2.3. Automatické zobrazení obrazovky ji spíše mátló, než zjednodušilo práci. Zbytek úkolu zvládla bez problémů.
2. Chvilí trvalo, než jí napadlo jít na obrazovku vytváření kola, kde se klávesnice realně nachází a kde je možnost ji nastavit. Ale i po tom nedokázala najít tlačítko pro nastavení. Muselo ji být napovězeno.
3. Úkol splnila bez problémů.
4. Intuitivně hledala možnost změnit čas zahájení v seznamu tréninků. Nakonec na to přišla sama.
5. Úkol splnila bez problémů.
6. Po předešlé zkušenosti, věděla kam má jít.

Souhrn

Stejně jako uživatel 5.2.3 jí dělala problém automatická vyskakovací obrazovka stejně tak nalezení možnosti nastavovat klávesnici. Klávesnice prvků ji připadala dobrá, ale navrhovala vyplňování rozdělit do více kroků. V prvním kroku by si skokan vybral kolik 1/4 salt prvek má, poté vyplnil počty 1/2 vrutů v připravených políčkách a nakonec polohu skoku. Klávesnice prvků by pak prakticky nebyla ani potřeba.

5.2.5 Jakub Vodička

Věk	12 let
Úroveň	Závodník
Ogranizace	Žádná
Profese	Student základní školy

Úkoly

1. Uživatel hned věděl jak zahájit trénink, ale tam skončil. Nevěděl jak má pokračovat a kde má maximálky vyplnit. Po nápovědě dokázal maximálky vyplnit. Další problém nastal, když měl vyplnit prvky v kole. I když je to závodník a několikrát si vyplňoval závodní kartu, stále nemá osvojené značky prvků tak dobře, aby je mohl rychle vyplňovat. Po vysvětlení byl schopen pomalu vyplnit požadovaný počet kol.
2. Tento úkol splnil s velkým překvapením velice rychle a sám.
3. Úkol splnil bez problémů.

4. Uložení nebyl problém. Se změnou zezачátku bojoval, pak ji ale našel sám.
5. Úkol splnil bez problémů.
6. Dlouho možnost hledal v seznamu tréninků. Po nějaké době to vzdal a po nápovědě způsob našel.

Souhrn

I když je uživatel závodník a ví jaké prvky skáče, je stále moc mladý na osvojení značení prvků a sám by na to nepřišel. Bylo by vhodné přidat vysvětlení, jak se prvky značí, aby byly ostatní mladší uživatelé schopni se rychleji značení naučit. Problém byl i ten, že byl uživatel velice mladý a až moc často se otáčel pro rady. To celý proces spomalovalo.

5.2.6 Filip Fencel

Věk	19 let
Úroveň	Rekreační skokan
Organizace	Žádná
Profese	Student gymnázia

Úkoly

1. Trénink zahájil bez problémů. Možnost vyplnit maximálky hledal na automaticky se otevírající obrazovce, kde ji nenašel a potřeboval radu. Poté dokázal maximálky vyplnit. I když to není závodník, značení prvků si osvojil a klávesnici dokázal používat bez nápovědy. Zbytek úkolu splnil bez problémů.
2. Úkol splnil bez problémů.
3. Úkol splnil bez problémů.
4. Změnu začátku jako většina testerů hledal v seznamu tréninků. Nakonec dokázal splnit bez nápovědy.
5. Úkol splnil bez problémů.
6. Úkol splnil bez problémů.

Souhrn

Znovu byla obrazovka s prvním kolem pouze matoucí. Uživatel znal značení prvků na trampolíně dobře a tak mu vyplňování s pomocí klávesnice nedělalo problém. Pro zadání tréninku na jiný den zkoušel gestem long press na daný den v kalendáři, které nakonec doporučil, jako dobrou alternativu.

5.2.7 Výstup testování použitelnosti

Z testování použitelnosti vzešlo několik problémů, které se uživatelům opakovaly.

Automatické zobrazení obrazovky s prvním kolem Testeři se nejvíce potýkali s problémem vyplnění maximálek jako prvního kola. Aplikace při zahájení tréninku automaticky otevře obrazovku s prvním kolem, která má ušetřit jedno kliknutí, ale místo toho celý proces spíše komplikuje. Může to být i tím, že testeři měli jako první kolo zadat právě maximálky a nikoli kolo s prvky.

Neintuitivní umístění mazání tréninku Uživatelé hledali mazání tréninku a editaci jeho zahájení v seznamu tréninků. Možné řešení je přidat možnost posunout řádek, kde budou tyto možnosti zobrazeny.

Hledání nastavení klávesnice prvků Je možné, že tento problém nastal kvůli špatně navržené posloupnosti úkolů. Uživatel dostal za úkol změnit nastavení klávesnice z úplně jiné obrazovky, než z které by typicky tento úkol dělal.

Nejasný způsob zapisování značek prvků Tento problém nastává hlavně u mladších skokanů, kteří si systém značek neosvojili. Pro ně by bylo vhodné přidat na obrazovku, kde se nachází klávesnice prvků, vysvětlení, jak se prvky značí. Nakonec by to i ušetřilo práci trenérům, kteří by to jinak skokanům museli vysvětlovat sami.

Možnosti rozšíření

6.1 Další typy událostí

Systém se nemusí omezovat pouze na vyplňování tréninků na trampolíně, ale může fungovat jako náhrada kalendáře pro sportovce. V tom případě by měl systém podporovat i jiné události.

6.1.1 Doplnkové události

Systém je založen na vyplňování tréninků po prvcích. Ale může nabýdnout i jiné události, jen pro informovanost trenéra. Informace, kterou by mohli potřebovat je například regenerace. Stačilo by, aby se u takové události objevovaly informace o času zahájení, době trvání a textového popisu. Trenér skoků na trampolíně nepotřebuje vědět, co přesně skokan při regeneraci dělal, ale stačí mu ta samotná informace, že skokan někou dobu regeneroval. V případě potřeby pro větší specifičnost bude k dispozici i textový popis.

Další události, ve kterých stačí vyplnit jen minimum informací, ale dávají trenérovi velkou přidanou hodnotu, jsou například běh, plavání, cyklistika, posilovna, jóga a pilátes. Všechny tyto události nejsou hlavní náplní skokanských tréninků, ale velkou mírou pomáhají skokanovi s jeho fyzickou přípravou.

Další vhodná událost je jídlo. Skokan má možnost vypsát co a kdy jedl. V této chvíli by se navíc od informací z doplnkových událostí objevila kategorie, zda se jedná o snídani, oběd, večeři nebo svačinu.

6.1.2 Disciplíny podobné trampolíně

S trampolínou se spojují další dvě sportovní disciplíny. Tumbling a Double-mini trampolína. Často se stává, že skokan trénuje všechny tři disciplíny, případně dvě ze tří zároveň. Každá disciplína má své odlišnosti, ale všechny tři disciplíny sdílí základní strukturu tréninku a prvky, které jsou skákány.

Mezi skoky na trampolíně a double-mini trampolínou je rozdíl pouze ve vypočítávání obtížnosti skoků. Značky prvků, jejich polohy i rozdělení tréninku je pak stejné.

Tumbling, oproti trampolíně a double-mini trampolíně, má také vlastní počítání obtížnosti prvků a navíc přidává jiný směr rotace. Jak trampolína, tak double-mini trampolína mají směry rotace vpřed nebo vzad. Tumbling má, podle pravidel, další směr a to do strany. I značka prvku je jiná. V trampolíně a double-mini trampolíně je první číslo počet čtvrt salt a každé další číslo značí počet půl vrutů v daném saltu. Značka „811“ tedy značí osm čtvrt salt, jeden půl vrut v prvním saltě a jeden půl vrutu v druhém saltě. Tumbling by stejný prvek zapsal takto „11“. V tumblingu se vynechává první číslo označující počet čtvrt salt a automaticky se počítá s tím, že skokan bude předvádět skoky pouze s celými salty. Počet salt se pak určuje podle počtu čísel ve značce.

Tumbling má navíc ještě několik prvků, které nelze zapsat v tradičním formátu s počty čtvrt salt a půl vrutů. Těmi jsou rondát, temposalto, přemet vpřed a přemet vzad.

Přidáním zbylých dvou disciplín by se naplnili potřeby těch závodníků, kteří trénují nejen skoky na trampolíně a zvětšila by se potencionální základna uživatelů, kteří trénují pouze jednu nebo obě zbylé disciplíny.

6.2 Tréninkové plány

Velmi žádanou funkcionalitou mezi trenéry je možnost vytvářet tréninkové plány, sdílet je se skokany, kteří by podle nich trénovali. Skokan se nejen že nemusí ptát trenéra, co má skákat následující kolo, ale hlavně se může na tréninkový plán připravit. Navíc může vidět plány na celý týden a být tak motivovaný.

Přidáním tréninkových plánů by v systému mohla vzniknout, vedle analýzy, i statistika. Ta by zobrazovala, jak je skokan procentuálně úspěšný v plnění tréninkových plánů. Trenér by měl další informace nejen o tom, co skokan skočil, ale co v daném kole skočit měl.

6.3 Filtrování a skupiny

Při velkém počtu spojených skokanů může být trenérská aplikace nepřehledná a seznam, i jen ve vybraném dni, velice dlouhý. Proto je filtrace a vytváření tréninkových skupin užitečná funkcionalita.

Trenér by měl více možností filtrace. Filtrovat jednoho skokana, filtrovat pouze určité typy událostí nebo filtrovat skupiny skokanů. Trenér může mít více tréninkových skupin. Systém by mu přidáním možnosti filtrace skupiny skokanů dal možnost sledovat pouze jednu skupinu a zároveň by mu skokaní z ostatních skupin nepřekáželi.

6.4 Senzor pro automatické zapisování

Při hledání existujících řešení z kapitoly 2.4 byla nalezena služba FXE, která pomocí senzoru na noze počítala skóre při skákání na trampolíně. Skóre je počítáno výhradně z frekvence skoků.

Využitím neuronových sítí by se dalo docílit naučení modelu na jednotlivé prvky a to jen z dat z akcelerometru a gyroskopu. Model by v reálném čase posílal rozpoznané prvky do aplikace a vyplňoval tak skokanův trénink. Přibyly by tím i informace o čase jednotlivých skoků, z které může trenér sledovat, kde skokan v sestavě snižuje a kde zvyšuje.

6.5 Kurzor při vytváření kola

V poslední verzi systému je možné přidávat nebo mazat prvky jen od konce. Například systém neumožňuje přidat prvek doprostřed pokusu. Není to funkce, bez které by byl systém nepoužitelný, ale může to být pro skokana nepříjemnost.

Proto by jako další rozšíření systému mohlo být přidání kurzoru do obrazovky s vytvářením kola. Ten by umožnil přidat nový prvek kamkoli do již vyplněného kola.

6.6 Role oddíl

Každý skokan, potažmo trenér, jsou součástí oddílu, který reprezentují na tuzemských závodech. Oddíl zajišťuje, že budou mít skokan a trenér prostor, náradí a čas na trénování. Proto by i oddíl měl mít možnost vidět, jak si skokani vedou.

Oddíl by viděl podobné informace jako trenér, ale od všech skoknů v oddíle. Navíc by u skokanů mohl sledovat informace o jejich zdravotních prohlídkách, mezinárodních licencích nebo je automaticky přihlašovat na závody.

Pod oddílem může vystupovat i organizace nebo federace, která podporuje vybrané skokany. Ať už je to podpora finanční nebo materiální, každá organizace chce vědět, zda se její investice zhodnocuje.

Závěr

Všechny cíle, které byly na začátku práce stanoveny, byly splněny. Podařilo se navrhnout a implementovat efektivní zadávání kola po prvcích s využitím klávesnice prvků, ve které jsou na výběr prvky bez poloh. Po výběru konkrétního prvku bez polohy skokan vybere polohu a prvek se zapíše do pokusu. Tím se zredukoval počet prvků v klávesnici na třetinu. Navíc byl vytvořen seznam připravených vazeb, pro zrychlení zadávání častých pokusů.

V analýze závodů byly identifikovány informace, které skokan na tréninku typicky sleduje. Těmi jsou čas pokusu a skočené sestavy. Funkcionality pro zanesení obou informací se do systému úspěšně implementovaly. Sledování sestav je možné díky označení připravené vazby přívlástkem typu sestavy.

V technické analýze se rozšířily modelové třídy a doplnily funkční a nefunkční požadavky tak, aby splňovaly potřeby pro rychlé a jednoduché zadávání kola po prvcích.

Pro dosažení všech funkcionalit, které systém potřebuje od serveru, byly využity služby Firebase a aplikační server Realm. Firebase slouží pro správu uživatelů a zajišťuje registraci, přihlášení, autorizaci, změnu přihlašovacích údajů a proces pro zapomenuté heslo. Realm slouží jako úložiště dat a zajišťuje synchronizaci, propagaci a autentizaci.

Uživatelské rozhraní obou mobilních aplikací bylo navrženo od začátku. Hlavní obrazovky jsou kalendář, detail tréninku a vytváření kola. Pro předčasné nalezení chyb v návrhu uživatelského rozhraní byl vytvořen wireframe skokanské mobilní aplikace. Všechny hlavní obrazovky byly otestovány heuristikou metodou za pomoci experta a sledováním uživatelů při plnění úkolů.

Implementace probíhala v iteracích a řídila se metodikou Lean developmentu. Celkově proběhlo jedenáct iterací pro skokanskou a šest iterací pro trenérskou aplikaci. Implementace byla nejvíce rozdílná oproti původní práci, neboť na bakalářskou práci nenavazovala.

Výstupem práce je funkční a otestovaný systém pro podporu trénování skoků na trampolíně, který je ke stažení na Apple AppStore pod jmény Trampigo Gymnast a Trampigo Coach. Ke dni psaní práce mají obě aplikace přes

ZÁVĚR

250 uživatelů z celého světa. Práce slouží jak první verze dostatečně specializovaného systému, který v trampolínovém sportu tak dlouho chyběl.

Literatura

- [1] Ashhab, N.: *Mobilní aplikace Trampingo*. Bakalářská práce (bc.), České Vysoké Učení Technické v Praze, 2016.
- [2] Fédération Internationale de Gymnastique: *2017 - 2020 Code of Points, Trampoline Gymnastics*. Listopad 2017, <http://www.fig-gymnastics.com/site/rules/disciplines/tra> [cit. 4.4.2018].
- [3] Realm Inc.: *BUILD BETTER APPS, FASTER WITH REALM*. 2017, <https://www2.realm.io/whitepaper/realm-overview-registration> [cit. 11.4.2018].
- [4] Lardinois, F.: Google Acquires Firebase To Help Developers Build Better Real-Time Apps. Říjen 2014, [cit. 11.4.2018]. Dostupné z: <https://techcrunch.com/2014/10/21/google-acquires-firebase-to-help-developers-build-better-realtime-apps/>
- [5] Firebase: Real time syncing for JSON data. [cit. 11.4.2018]. Dostupné z: <https://firebase.google.com/products/realtime-database/>
- [6] Hooper, S.: How Do Users Really Hold Mobile Devices? Únor 2013, [cit. 14.4.2018]. Dostupné z: <https://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php>
- [7] Herwig Mannaert, P. D. B., Jan Verelst: *Normalized Systems Theory*. Kopaa Digitale Media, 2016.
- [8] Ambysoft Inc.: 2014 Software Development at Scale Survey Results. Únor 2014, [cit. 19.4.2018]. Dostupné z: <http://www.ambysoft.com/surveys/stateOfITUnion2014Q2.html>
- [9] Ambysoft Inc.: 2018 IT Project Success Rates Survey Results. Únor 2018, [cit. 18.4.2018]. Dostupné z: <http://www.ambysoft.com/surveys/success2018.html>

LITERATURA

- [10] Manifest Agilního vývoje software. 2001, [cit. 19.4.2018]. Dostupné z: <http://agilemanifesto.org/iso/cs/manifesto.html>
- [11] Principy stojící za Agilním Manifestem. 2001, [cit. 19.4.2018]. Dostupné z: <http://agilemanifesto.org/iso/cs/principles.html>
- [12] Ries, E.: *Lean Startup*. BizBooks, 2015.
- [13] Pavla Lichnovská, E. K.: Heuristická analýza. 2009, [cit. 24.4.2018]. Dostupné z: <https://human-computer-interaction.webnode.cz/testovani-a-hodnoceni-rozhrani/metody-testovani/heuristicka-analyza/>

Seznam použitých zkratk

REST Representational State Transfer

API Application Programming Interface

SDK Software Development Kit

NoSQL Databázový koncept, ve kterém datové úložiště i zpracování dat používají jiné prostředky než tabulková schémata tradiční relační databáze.

PDF Portable Document Format

SaaS Software as a Service je model nasazení softwaru, kdy dochází k hostování aplikace provozovatelem služby.

IaaS Infrastructure as a service

RUP Rational Unified Process

MVP Minimum viable product

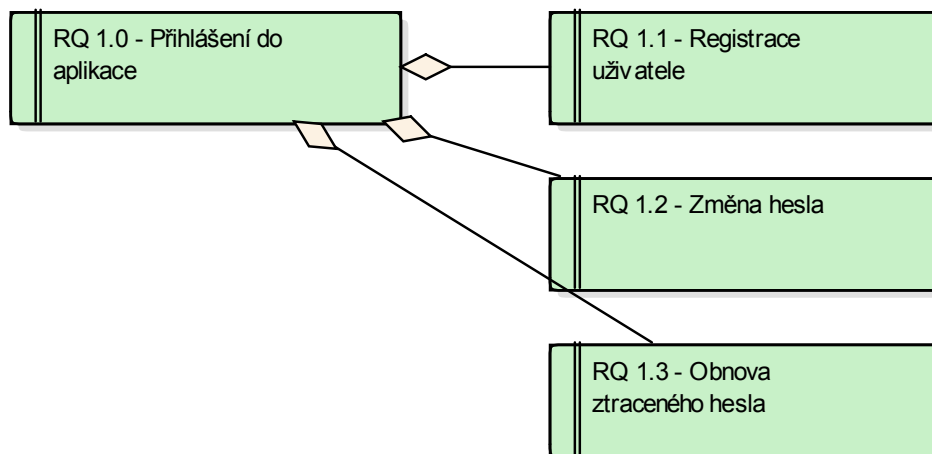
AWS Amazon Web Services

TDD Test Driven Development

Obsah přiloženého CD

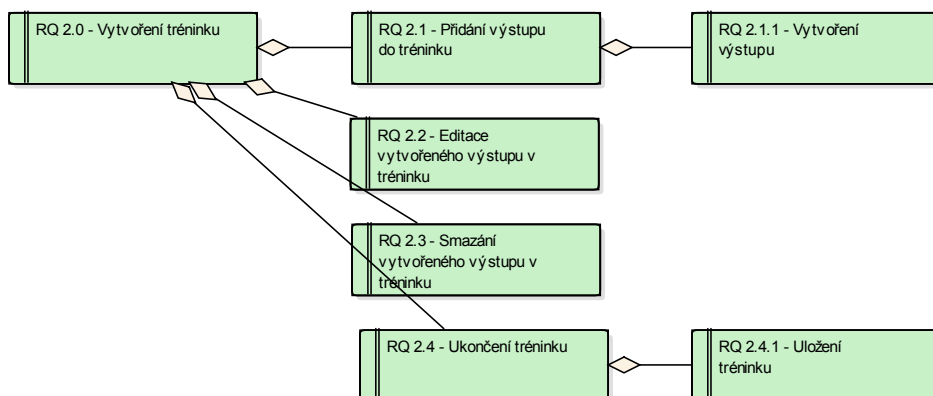
readme.txt.....	stručný popis obsahu CD
exe	adresář se spustitelnou formou implementace
├─ gymnast.ipa.....	skokanská mobilní aplikace
├─ coach.ipa.....	trenérská mobilní aplikace
src	
├─ impl.....	zdrojové kódy implementace
│ ├─ ios.....	zdrojové kódy mobilních aplikací
│ ├─ realm.....	zdrojové kódy aplikačního serveru
├─ DP_Ashhab_Naim_2018.zip...	zdrojová forma práce ve formátu L ^A T _E X
├─ gymnast_app_wireframe.pdf	wireframe skokanské aplikace
text	text práce
├─ DP_Ashhab_Naim_2018.pdf	text práce ve formátu PDF

Technické požadavky

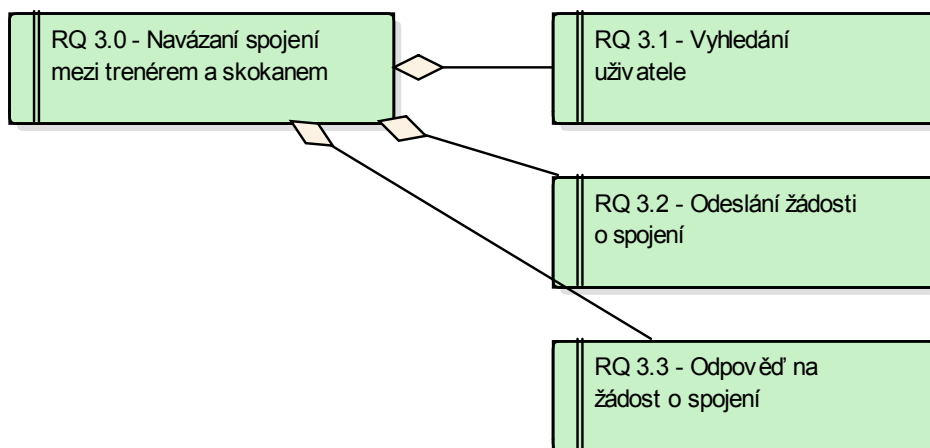


Obrázek C.1: Funkční požadavky na přihlášení

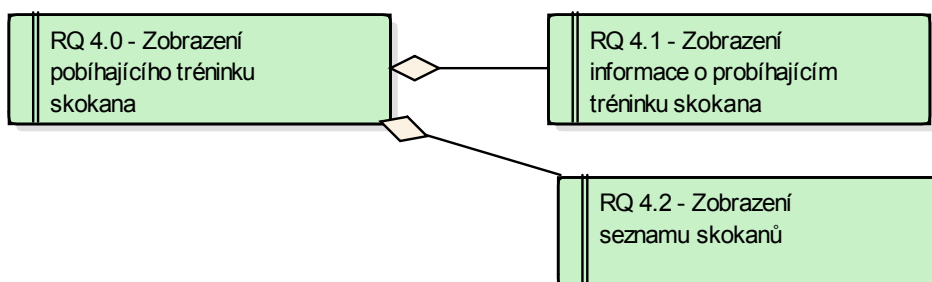
C. TECHNICKÉ POŽADAVKY



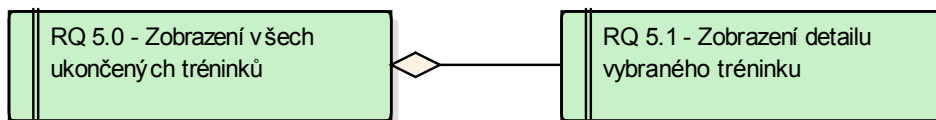
Obrázek C.2: Funkční požadavky na vytváření tréninku



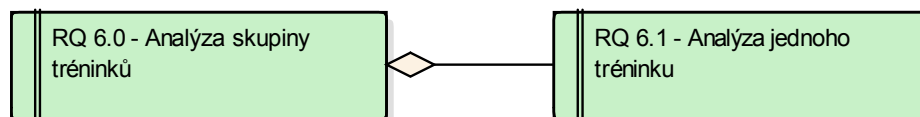
Obrázek C.3: Funkční požadavky na spojování uživatelů



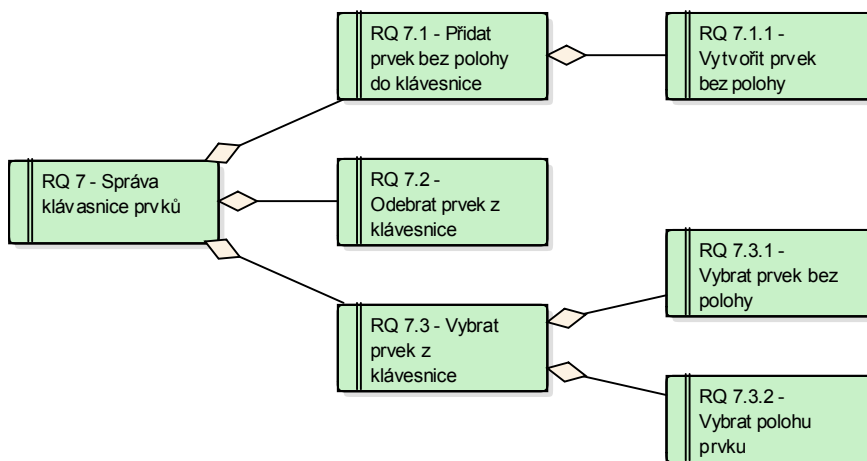
Obrázek C.4: Funkční požadavky na zobrazení probíhajících tréninků v reálném čase



Obrázek C.5: Funkční požadavky na zobrazení všech uložených tréninků

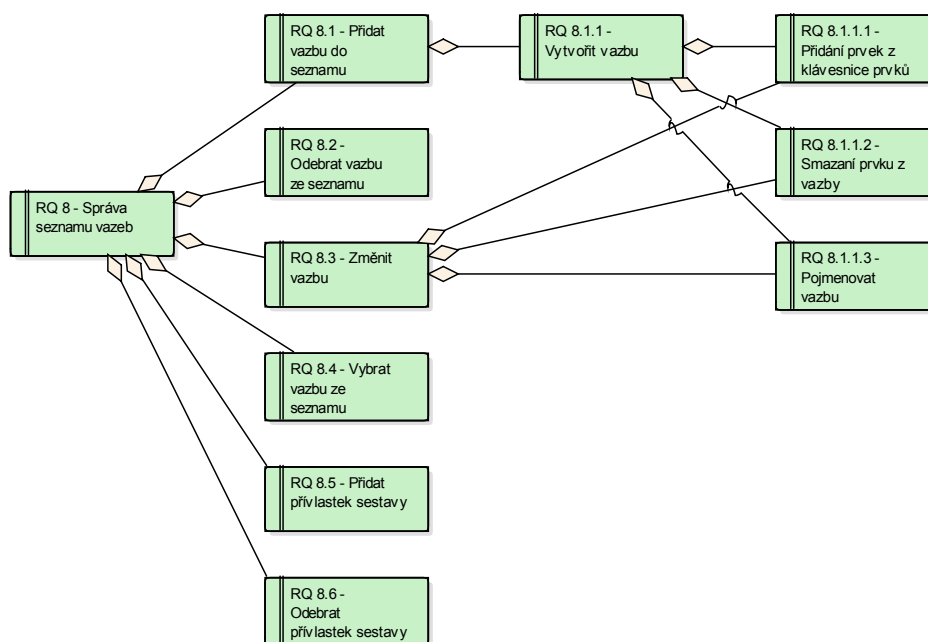


Obrázek C.6: Funkční požadavky na analýzu tréninků

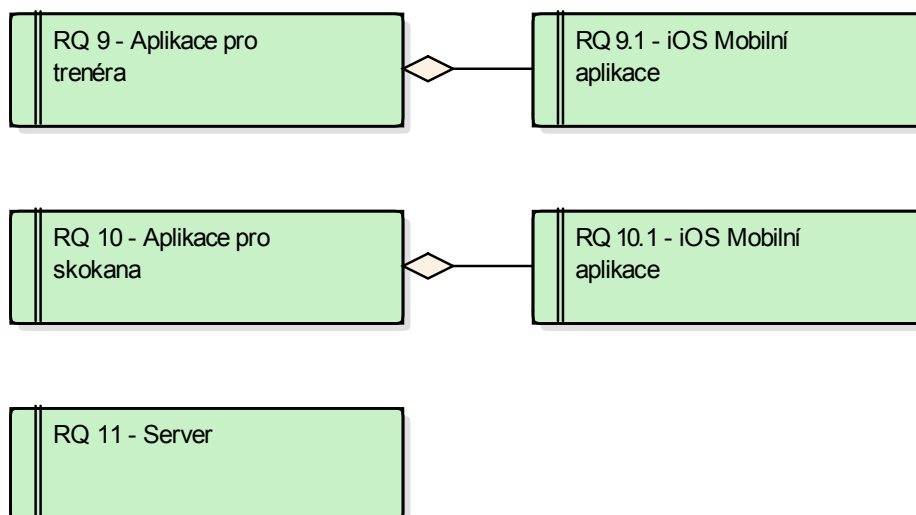


Obrázek C.7: Funkční požadavky na klávesnici prvků

C. TECHNICKÉ POŽADAVKY



Obrázek C.8: Funkční požadavky na seznam vazeb



Obrázek C.9: Nefunkční požadavky na systém

Heuristická analýza uživatelského rozhraní

1. Viditelnost stavu systému

- a) System nesmí zůstat zamrzlý a nereagovat na uživatelské vstupy
- b) Zobrazit progress bar
- c) Uživatel musí být informován o tom, co systém dělá

2. Shoda mezi systémem a realitou

- a) Zachování konvencí (a metafor) reálného světa
- b) Ikony (a metafor) se musí chovat jako to, co zobrazují / na co odkazují
- c) Do koše mohu věci nejen vyhodit, ale lze ho i vysypat, případně ho prohledat a vytáhnout už jednou vyhozenou věc
- d) Pozor na překlady

3. Minimální zodpovědnost (a stres)

- a) „Nic se nemůže pokazit“
- b) Vždy je možnost vrátit se zpět do předchozího stavu (undo)
- c) Nejlépe univerzální undo (historie)
- d) Uživatel ovládá systém, nikoli naopak
- e) Zrušení dlouho trvajících operací (rollback)
- f) Potvrzování akcí
- g) Varování před provedením nevratné akce

4. Shoda s použitou platformou a obecnými standardy

- a) Program pod Windows vypadá a chová se jako pod Windows. Nijak jinak.
- b) Pokud to jde, použít standardní systémové komponenty
- c) Systemové barvy a fonty
- d) Používat stejné termíny
- e) Vysvětlit zkratky

5. **Prevence chyb**

- a) Uživatel by neměl mít možnost zadat špatnou hodnotu
- b) Pokud je způsob zadávání z principu složitější, je třeba to na místě vysvětlit
- c) Zvýraznit povinné položky formulářů
- d) Potvrzování akcí

6. **Kouknu a vidím**

- a) Nezatěžovat uživatelskou paměť
- b) Akce, které uživatel může momentálně provést by měli být viditelné a snadno dosažitelné
- c) Stejně tak informace
- d) Nepotřebné kontrolky a informace nejsou potřeba
- e) Pozice uživatele
- f) Krok ve wizardu
- g) Pozice ve stromové struktúře

7. **Flexibilita a efektivita**

- a) Zkušený vs. běžný uživatelé
- b) To co zkušený uživatel pravidelně potřebuje, běžný kolikrát ani nepotřebuje vědět
- c) Pokročilý mód
- d) Klávesové zkratky / funkční klávesy
- e) Makra

8. **Miminalita (Klapky na očích)**

- a) Zobrazovat pouze informaci, která aktuálně k něčemu opravdu je
- b) Čím méně možností uživatel má, tím rychleji koná
- c) Grafika by neměla zastiňovat ovládání a účel

-
- d) Cokoli je zobrazeno soutěží o uživatelskou pozornost

9. Smysluplné chybové hlášky

- a) Nejlepší je nedojít do stavu, kdy je třeba chybového hlášení
- b) Chybové hlášení v běžném jazyce (žádné kódy)
- c) Chybové hlášení by mělo popsat co se stalo špatně, jak se to stalo a jak tomu příště předejít
- d) Případně možná řešení doporučit
- e) Mělo by poučit (vzdělat) uživatele

10. Help a dokumentace

- a) System by měl být použitelný bez jakékoli nápovědy, nicméně...
- b) Nápověda musí být
- c) Musí podporovat funkci vyhledávání
- d) Spíš než popisem by se mělam zabývat příklady
- e) Kontextová nápověda