



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Mobilní aplikace pro sledování běžecké aktivity na platformě Android
Student:	Hana Fukalová
Vedoucí:	Ing. Michal Valenta, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2020/21

Pokyny pro vypracování

Cílem práce je návrh a implementace mobilní aplikace pro platformu Android, která bude sledovat běžeckou aktivitu uživatele a umožní mu propojení s ostatními uživateli. Uživatel bude moci porovnávat své vlastní výkony a soutěžit s ostatními uživateli.

Postupujte v těchto krocích:

1. Analyzujte požadavky uživatelů pro tento typ aplikace.
2. Analyzujte konkurenční aplikace.
3. Na základě analýzy požadavků navrhnete vhodný koncept aplikace, definujte funkční a nefunkční požadavky a zvolte vhodnou implementační platformu.
4. Implementujte prototyp aplikace a řádně ho otestujte.
5. Zhodnoťte výsledek a navrhnete možná vylepšení.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

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

V Praze dne 6. února 2020



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

Bakalářská práce

Mobilní aplikace pro sledování běžecké aktivity na platformě Android

Hana Fukalová

Katedra softwarového inženýrství

Vedoucí práce: Ing. Michal Valenta, Ph.D.,

30. července 2020

Poděkování

Ráda bych poděkovala Ing. Michalu Valentovi, Ph.D., za vedení bakalářské práce, Bc. Marku Kodrovi za technické rady v oblasti vývoje Android aplikací a své rodině a přátelům za podporu při studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při 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. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 30. července 2020

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2020 Hana Fukalová. 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 a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Fukalová, Hana. *Mobilní aplikace pro sledování běžecké aktivity na platformě Android*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Bakalářská práce se zabývá návrhem a implementací mobilní aplikace na platformě Android, která slouží k sledování běžecké aktivity uživatele. Uživatel může zaznamenat své běžecké výkony a soupeřit s ostatními uživateli o body. V analytické části jsou specifikovány funkční a nefunkční požadavky na aplikaci a jsou zde zanalyzovány konkurenční aplikace. V kapitole týkající se návrhu jsou uvedeny vhodné technologie pro implementaci, návrh architektury, databázového modelu a uživatelského rozhraní. Realizace prototypu a jeho testování je poté popsána v kapitolách implementace a testování. V závěru jsou zhodnoceny výsledky práce a návrhy na případná vylepšení.

Klíčová slova mobilní aplikace, sledování polohy, záznam běhu, fitness aplikace, gamifikace, Android, Kotlin, Firebase

Abstract

Aim of this bachelor thesis is to design and implement mobile application on Android platform, which would track user's running activity. User would be able to save his running records and compare himself with other users by his point score. The functional and non-functional requirements and comparison with similar application is the topic of analysis chapter. Next section is devoted to application design. There is a summary of suitable technologies, application architecture, database model and user interface design. The implementation of prototype of the application and its testing is described in the implementation and the test part of the thesis. In the end the work results are evaluated and possible improvements are proposed.

Keywords mobile application, location tracking, run record, fitness application, gamification, Android, Kotlin, Firebase

Obsah

Úvod	1
Cíl práce	3
1 Analýza	5
1.1 Koncept aplikace a systém hry	5
1.2 Konkurenční aplikace	5
1.2.1 Zombies, Run!	6
1.2.2 Charity Miles	6
1.2.3 Strava	7
1.2.4 Zhodnocení	7
1.3 Specifikace požadavků	8
1.3.1 Funkční požadavky	8
1.3.2 Nefunkční požadavky	9
1.4 Případy užití	9
1.4.1 Vytvoření a správa uživatelského účtu	9
1.4.2 Zaznamenávání běžecké aktivity	10
1.4.3 Hledání protivníků a plnění výzev	10
2 Návrh	13
2.1 Architektura aplikace	13
2.2 Struktura databáze	14
2.3 Uživatelské rozhraní	14
2.4 Výběr technologií	16
2.4.1 Android platforma	16
2.4.1.1 Minimální verze API	16
2.4.1.2 Implementační jazyk	17
2.4.2 Firebase	17
2.4.2.1 Autentizace	17
2.4.2.2 Cloud Firestore	17

2.4.2.3	Firestore Storage	17
2.4.3	Google Maps	17
2.4.4	Nahrávání a zpracování obrázků	18
2.4.5	Tvorba grafů	18
3	Implementace	19
3.1	Základní komponenty	19
3.2	Struktura aplikace	20
3.2.1	LoginActivity	20
3.2.2	MainActivity	21
3.2.2.1	Uživatelský profil	21
3.2.2.2	Záznam běhu	22
3.2.2.3	Přehled záznamů aktivit	22
3.2.2.4	Přehled uživatelů	22
3.2.2.5	Výzvy	23
3.2.2.6	Navigace mezi obrazovkami	25
3.3	Komunikace mezi vrstvami	26
3.3.1	Observer pattern	26
3.3.2	Asynchronní operace	27
3.4	Sledování polohy	28
3.4.1	LocationService	28
3.4.2	Notifikace	28
3.4.3	Zpracování polohy	28
4	Testování	29
4.1	Testování autorem	29
4.2	Uživatelské testy	29
4.2.1	Test 1	30
4.2.2	Test 2	31
4.2.3	Test 3	31
4.2.4	Test 4	31
4.2.5	Test 5	32
4.2.6	Test 6	32
4.2.7	Shrnutí	32
5	Budoucí vylepšení	35
	Závěr	37
	Bibliografie	39
A	Seznam použitých zkratk	41
B	Obsah příloženého CD	43

Seznam obrázků

1.1	Snímky obrazovek z aplikace <i>Zombies, Run!</i> [1].	6
1.2	Snímky obrazovek z aplikace <i>Charity Miles</i> [2].	7
1.3	Snímky obrazovek z aplikace <i>Strava</i> [3].	8
1.4	Use Case diagram vytvoření uživatelského profilu a jeho správy . .	10
1.5	Use Case diagram zaznamenávání uživatelovy běžecké aktivity a plnění výzev	11
1.6	Diagram aktivity znázorňující vytvoření a plnění výzvy	12
2.1	Zobrazení rozložení zodpovědnosti na jednotlivých vrstvách architektury Model-View-ViewModel	13
2.2	Relační model struktury databáze	14
2.3	Návrhy obrazovek seznamu uživatelů, uživatelského profilu a běžeckého záznamu.	15
2.4	Obrazovka zobrazení běžeckých záznamů.	15
2.5	Obrazovky přehledu žádostí o výzvy, aktivních výzev a splněných výzev.	16
3.1	Diagram životního cyklu aktivity [20]	20
3.2	Obrazovky přihlášení a registrace	21
3.3	Obrazovka uživatelského profilu	22
3.4	Zobrazení běžeckého záznamu	23
3.5	Obrazovka přehledu uživatelských běžeckých záznamů	24
3.6	Obrazovka zobrazení seznamu nepřátel	24
3.7	Obrazovky přehledu žádostí o výzvy, aktivních výzev a ukončených výzev	25
3.8	Stavový diagram znázorňující přechody mezi jednotlivými stavy výzvy	26

Seznam tabulek

1.1	Pokrytí funkcionality konkurenčními aplikacemi	7
1.2	Pokrytí funkčních požadavků případy užití 1-6	11
1.3	Pokrytí funkčních požadavků případy užití 7-13	11
4.1	Přehled testerů	30

Úvod

Jedním z faktorů vzniku mnoha civilizačních chorob, jako jsou cukrovka, obezita, deprese či syndrom vyhoření, bývá uváděn i nedostatek přirozeného fyzického pohybu. S pokrokem doby obnáší mnoho povolání celodenní sezení v kanceláři. I přes nedostatek fyzické aktivity bývá paradoxně po takovém dni organismus často velmi unavený a je obtížné donutit se k jakémukoli namáhavější fyzické aktivitě. Přitom i pouhá půlhodina cvičení denně může potenciálně přispět k prevenci proti různým nemocem. Je proto potřeba nalézt vhodný způsob, jak zajistit a udržet potřebnou motivaci, než se jí stane strach o vlastní zdraví.

Takový způsob mohou představovat fitness aplikace, které měří a zaznamenávají uživatelskou fyzickou aktivitu. Přehled o výkonech a možném zlepšení může samo o sobě uživatele dostatečně motivovat, často ale po určité době zájem opadá, proto novější aplikace přinášejí různé zpestřující prvky, jako je např. vzájemné porovnávání s ostatními uživateli či plnění běžeckých výzev. Tento způsob vnesení herních prvků do původně neherních oblastí se označuje jako gamifikace a z pohledu marketingu a zvýšení zájmu uživatele představuje velký potenciál. Ve své práci se chci proto věnovat tvorbě fitness aplikace, která se bude snažit motivovat uživatele k fyzické aktivitě soupeřením mezi sebou, což by mělo vnést do aplikace jistý prvek gamifikace.

Cíl práce

Cílem této bakalářské práce je osvědčenými praktikami softwarového inženýrství navrhnout a realizovat prototyp Android aplikace, která bude zaznamenávat běžickou aktivitu uživatelů a umožní vzájemné soupeření uživatelů formou hry.

Prvním krokem bude určení uživatelských požadavků na daný typ aplikace a zanalyzování konkurenčních aplikací podobného typu. Na základě získaných poznatků budou specifikovány funkční a nefunkční požadavky na vytvářený prototyp. Po provedení analýzy bude proveden návrh prototypu, na jehož základě proběhne samotná implementace aplikace, která bude následně řádně otestována. Na závěr budou zhodnoceny výsledky práce a navržena možná budoucí vylepšení.

Analýza

Analýza slouží k specifikaci základních požadavků na aplikaci a definování podstatné funkcionality. Poznatky z této kapitoly budou využity při návrhu prototypu aplikace a výběru vhodných technologií.

Součástí této kapitoly je popis základního konceptu aplikace a analýza konkurenčních aplikací. Na jejich základě jsou potom určeny uživatelské požadavky na aplikaci.

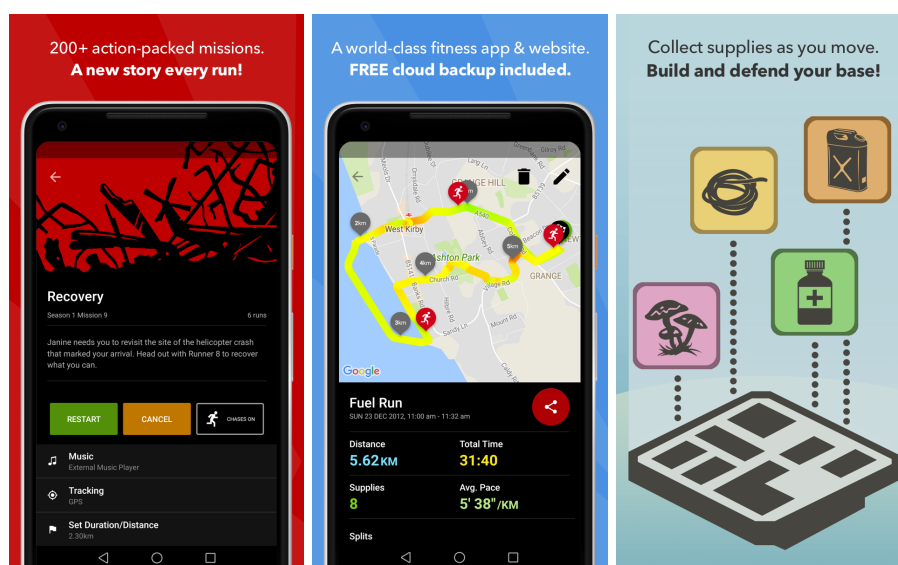
1.1 Koncept aplikace a systém hry

Základním konceptem aplikace je umožnění sledování běžecké aktivity uživatele a jeho propojení s komunitou ostatních běžců. Interakce mezi uživateli bude probíhat formou hry. Uživatel A může vyzvat uživatele B na běžeckou výzvu. Uživatel A zaběhne určitou vzdálenost, která ale nebude uživateli B známa, což by mělo uživatele B motivovat k co možná nejlepšímu výkonu. Uživatel B se bude poté snažit překonat vzdálenost zaběhnutou uživatelem A. Vítěz získává body a poražený je ztrácí. Oba uživatelé musí tak zvážit běžecké schopnosti svého soupeře a správně odhadnout, zda-li ho mohou porazit. Svůj odhad budou moci založit na základním přehledu běžeckých statistik v profilu uživatelů.

1.2 Konkurenční aplikace

Fitness aplikací sloužících k zaznamenávání běžecké aktivity uživatele existuje dnes na trhu již velké množství, jak pro Android OS, tak i pro jiné platformy. V této části práce jsem vybrala tři aplikace, které mimo sledování fyzické aktivity nabízejí uživateli i určitý ozvlášťující prvek, který má pomoci udržet uživatelovu motivaci k výkonu. Tyto aplikace jsem vybrala z toho důvodu, že stejný koncept bude využit v mé vlastní práci. Analyzovala jsem pouze

1. ANALÝZA



Obrázek 1.1: Snímky obrazovek z aplikace *Zombies, Run!* [1].

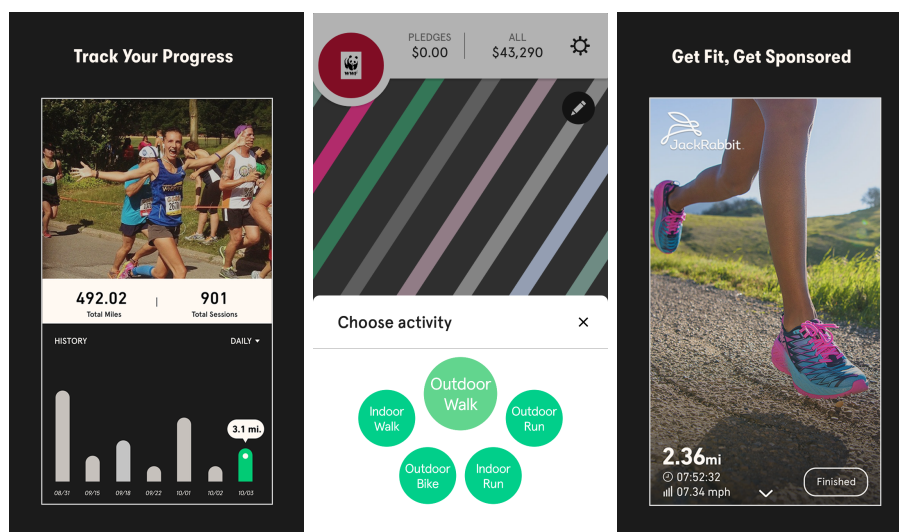
aplikace dostupné pro Android OS, jelikož realizace mého návrhu proběhne prozatím pouze pro tuto platformu.

1.2.1 Zombies, Run!

Zombies, Run! přináší originální způsob zaměstnání hlavy při fyzické aktivitě. Uživatel se při každém svém běhu stává aktérem v post-apokalyptickém světě sužovaném zombies. Do sluchátek je mu vyprávěn příběh pohledem obyvatel základny, kteří bojují každý den o přežití. V průběhu misí jsou sbírány různé předměty, které mohou být v další části aplikace využity na rozšíření virtuálního modelu základny. Aplikace dále poskytuje přehled běžeckých aktivit a statistik běhů, vytvoření tréninkového plánu či propojení s komunitou aplikace.

1.2.2 Charity Miles

Tato aplikace slibuje za každou naběhanou míli darovat určitou částku na vybranou charitu. Uživatel si při registraci vybere charitu, na kterou chce věnovat své naběhané míle, a poté už jen měří svou aktivitu. Ve svém profilu si může poté zobrazit přehled svých záznamů. Sponzorem samotných běhů jsou různé organizace a nebo může uživatel pozvat své přátele, aby jeho běhy sponzorovali. Uživatelé se také mohou sdružovat do skupin a běhat za společný zájem.

Obrázek 1.2: Snímky obrazovek z aplikace *Charity Miles* [2].

funkcionalita	Zombies, Run!	Charity Miles	Strava
Záznam běhu	ANO	ANO	ANO
Běžecké statistiky	ANO	NE	ANO
Přehled záznamů	ANO	ANO	ANO
Herní prvek	ANO	NE	NE
Soupeření mezi uživateli	NE	NE	ANO

Tabulka 1.1: Pokrytí funkcionality konkurenčními aplikacemi

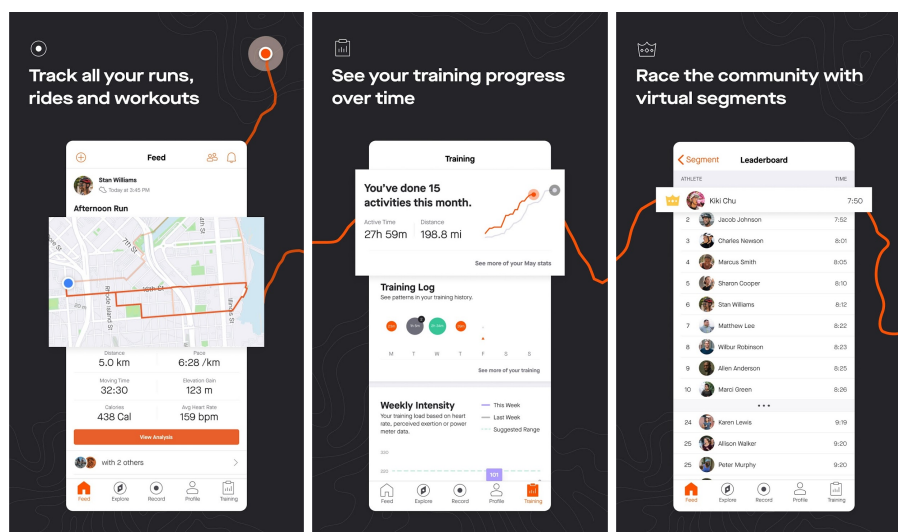
1.2.3 Strava

Strava je mobilní aplikace pro webový portál strava.com, který sdružuje nadšence do sportovních aktivit. Ačkoliv se může zdát, že se jedná o české jméno, ve skutečnosti je název ve švédštině, kde slovíčko *strava* lze přeložit do češtiny jako sloveso namáhat se, snažit se, což je zjevně i mottem celé komunity využívající portál. Strava má velkou uživatelskou základnu (okolo 50 mil. uživatelů) a umožňuje uživateli mobilní aplikace sledování nejen běžecké aktivity, ale i dalších forem pohybu, jako je jízda na kole, turistika, lyžování, plavání apod. Aplikace poskytuje uživateli detailní přehled o jeho aktivitách a umožňuje propojení s komunitou.

1.2.4 Zhodnocení

Při analýze konkurenčních aplikací jsem se zaměřila na pokrytí funkcionality, která je odvozena ze základního konceptu vyvíjené aplikace [1.1]. Žádná z analyzovaných aplikací nepokrývá všechny funkcionality, které jsou potřeba, jak je možné vidět v tabulce [1.1].

1. ANALÝZA



Obrázek 1.3: Snímky obrazovek z aplikace *Strava* [3].

1.3 Specifikace požadavků

Požadavky na systém jsou odvozeny z popisu konceptu aplikace a analýzy konkurenčních aplikací. Slouží k určení základní funkcionality aplikace.

1.3.1 Funkční požadavky

F1 Vytvoření uživatelského účtu

Aplikace bude přístupná pouze přihlášeným uživatelům. Každý uživatel se registruje pomocí emailové adresy a hesla, což budou také jeho přihlašovací údaje. Uživatel se bude moci také odhlásit z aplikace.

F2 Zobrazení profilu a správa údajů

Uživatel si bude moci zobrazit uživatelský profil se svými údaji a změnit profilovou fotku.

F3 Vytvoření běžeckých statistik

Uživatel si bude moci ve svém profilu zobrazit měsíční souhrn naběhaných kilometrů a celkový souhrn naběhané vzdálenosti a času a také nejdelší běh.

F4 Zaznamenávání běžecké aktivity

Uživatel bude moci spustit záznam běhu, který bude sledovat jeho polohu a měřit čas. Po ukončení aktivity uživatel rozhodne, zda záznam uloží nebo ne.

F5 Zobrazení běžeckých záznamů

Každý uložený běžecký záznam si bude uživatel moci v aplikaci zobrazit a

případně také smazat.

F6 Porovnání s ostatními uživateli

Podle počtu získaných bodů se budou řadit uživatelé do žebříčku a každý uživatel se bude moct podívat, jak si vede oproti ostatním uživatelům, a zobrazit si profil ostatních uživatelů.

F7 Vyzývání uživatelů na běžecké výzvy

Aplikace umožní uživateli vyzvat jiného uživatele na běžeckou výzvu, za kterou bude vítěz moci získat body.

F8 Zobrazení přehledu výzev

Uživatel bude mít přehled o svých splněných i prozatím nesplněných výzvách.

1.3.2 Nefunkční požadavky

N1 Mobilní aplikace na platformě Android

Aplikace bude vytvořena na Android platformě.

N2 Synchronizace s datovým serverem

Aplikace bude propojena s cloud databází Firestore, která umožňuje synchronizaci dat v reálném čase.

N3 Jednoduchý intuitivní design

Design uživatelského rozhraní by měl být intuitivní a přehledný pro snadnou orientaci v aplikaci.

N4 Vysoká míra podpory na zařízeních

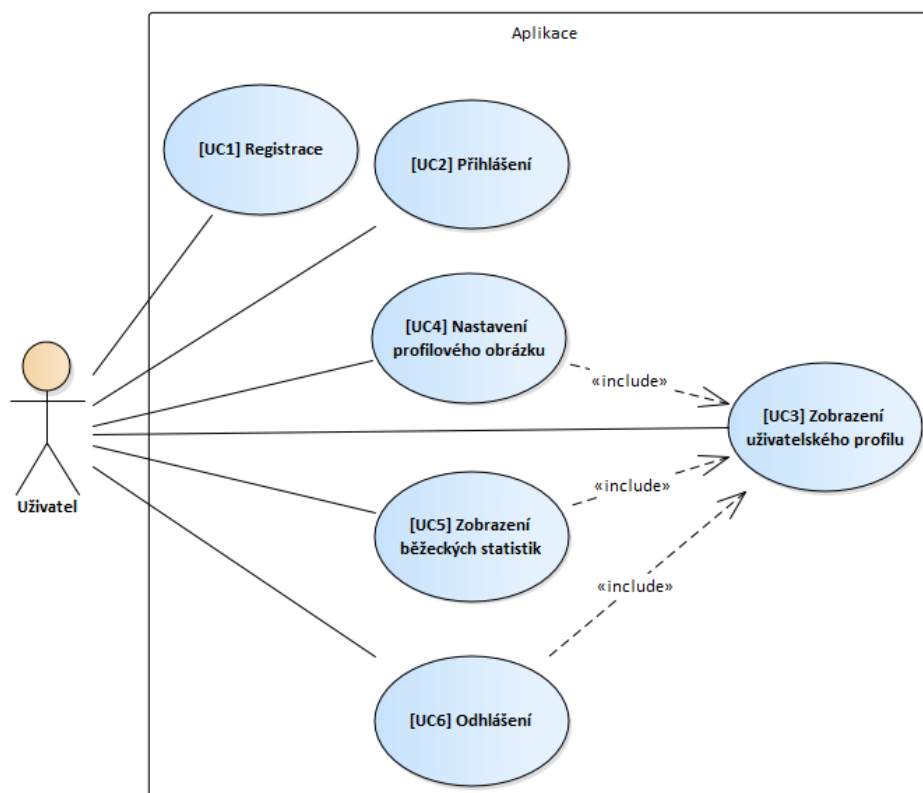
Aplikace by měla být kompatibilní s dostatečným množstvím aktivních zařízení.

1.4 Případy užití

V této sekci jsou uvedeny případy užití aplikace založené na funkčních požadavcích specifikovaných v sekci [1.3.1].

1.4.1 Vytvoření a správa uživatelského účtu

Aplikace bude přístupná pouze registrovaným uživatelům. Při registraci vyplní uživatel základní údaje, včetně emailové adresy a hesla, které budou sloužit k přihlášení. V aplikaci si bude uživatel moci zobrazit svůj profil, zobrazit si přehled svých běžeckých statistik nebo změnit profilový obrázek, případně se odhlásit z aplikace. Use case diagram k těmto případům je zobrazen na obrázku [1.4].



Obrázek 1.4: Use Case diagram vytvoření uživatelského profilu a jeho správy

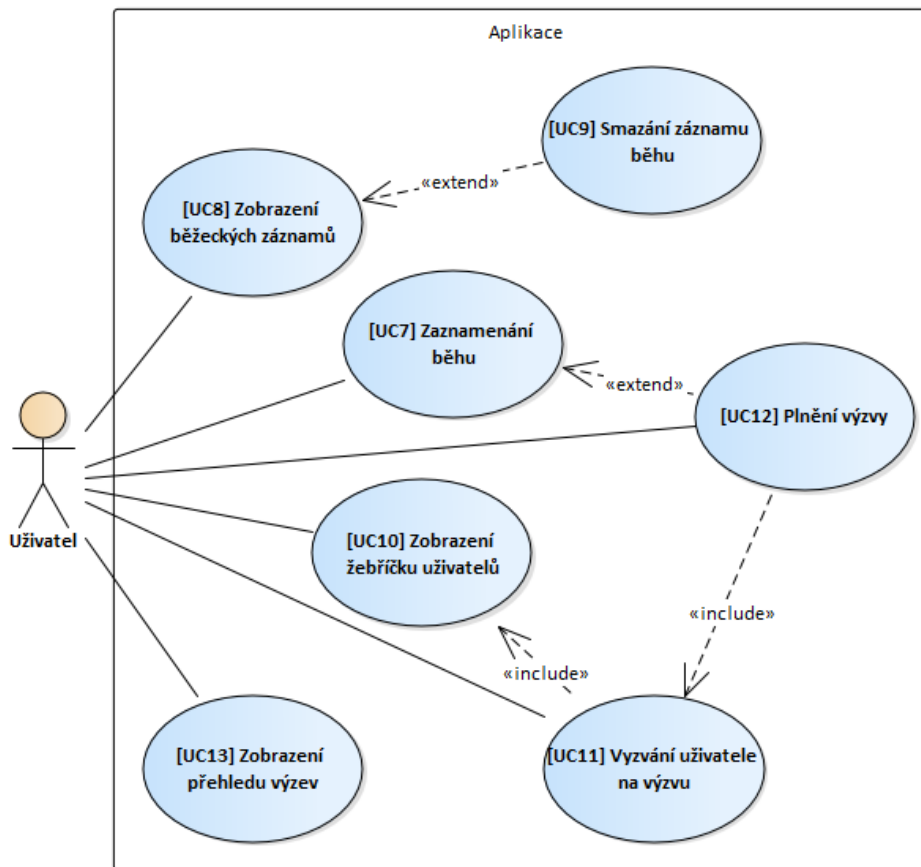
1.4.2 Zaznamenávání běžecké aktivity

Uživatel spustí zaznamenávání běhu a po skončení aktivity bude záznam uložen. Poté si bude moci uživatel zobrazit přehled svých běžeckých výkonů a statistiky jeho aktivity. Jednotlivé záznamy bude možné smazat.

1.4.3 Hledání protivníků a plnění výzev

Uživatel si bude moci prohlédnout své protivníky - ostatní uživatele a vyzvat je na běžeckou výzvu či plnit výzvy, na které byl vyzván jinými uživateli. Dále si bude moci zobrazit přehled svých výzev — aktivní i již splněné. Diagram aktivity na obrázku [1.6] popisuje proces vytvoření a plnění výzvy. Znázornění tohoto případu užití spolu se zaznamenáváním běžecké aktivity je v diagramu na obrázku [1.5].

Znázornění pokrytí funkčních požadavků případy užití je vidět v tabulkách [1.2] a [1.3].



Obrázek 1.5: Use Case diagram zaznamenávání uživatelské běžecí aktivity a plnění výzev

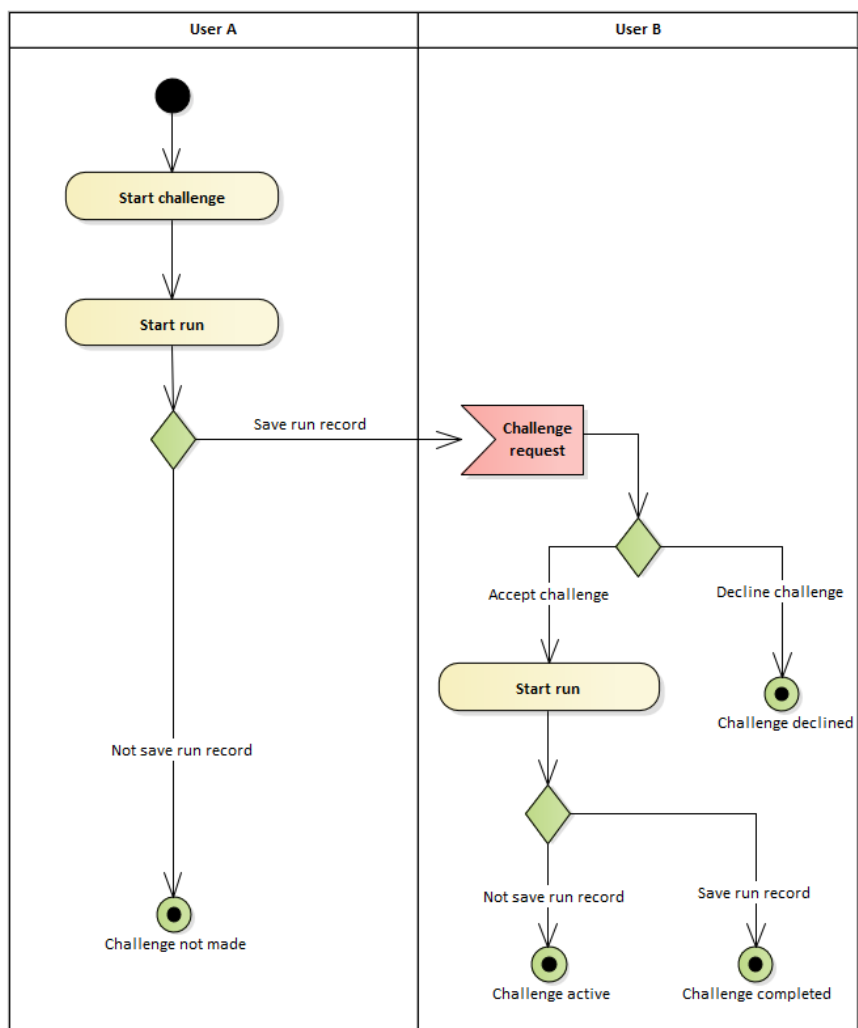
	UC1	UC2	UC3	UC4	UC5	UC6
F1	x	x				x
F2			x	x		
F3					x	

Tabulka 1.2: Pokrytí funkčních požadavků případy užití 1-6

	UC7	UC8	UC9	UC10	UC11	UC12	UC13
F4	x						
F5		x	x				
F6				x			
F7					x	x	
F8							x

Tabulka 1.3: Pokrytí funkčních požadavků případy užití 7-13

1. ANALÝZA



Obrázek 1.6: Diagram aktivity znázorňující vytvoření a plnění výzvy

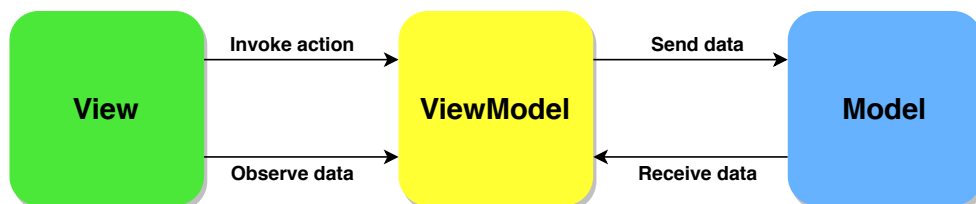
Návrh

Tato kapitola je věnována návrhu prototypu aplikace. Poznatky z analytické části práce jsou využity při výběru architektury aplikace, návrhu struktury databáze a uživatelského rozhraní a zvolení vhodných technologií.

2.1 Architektura aplikace

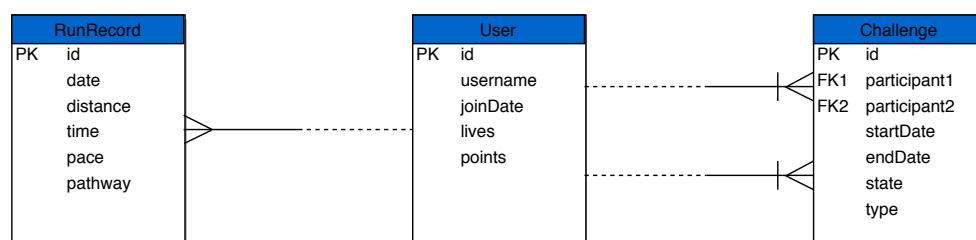
Pro rozdělení zodpovědnosti jednotlivých komponent se využívá v Android aplikacích standardně třívrstvá architektura. Princip oddělení uživatelského rozhraní, business logiky a komunikace s datovým úložištěm ulehčuje testování, rozšiřování a údržbu aplikace.

Ve své práci jsem se rozhodla využít vzor *Model-View-ViewModel*. Tento návrhový vzor se snaží o maximální oddělení zodpovědnosti a provázanosti jednotlivých vrstev. Funkcí Model vrstvy je zajištění propojení s datovým úložištěm. View zprostředkovává komunikaci s uživatelem, tedy implementuje uživatelské rozhraní, a ViewModel umožňuje komunikaci mezi těmito dvěma vrstvami a zpracovává data [4]. Diagram znázorňující komunikaci mezi vrstvami lze vidět na obrázku [2.1].



Obrázek 2.1: Zobrazení rozložení zodpovědnosti na jednotlivých vrstvách architektury Model-View-ViewModel

2. NÁVRH



Obrázek 2.2: Relační model struktury databáze

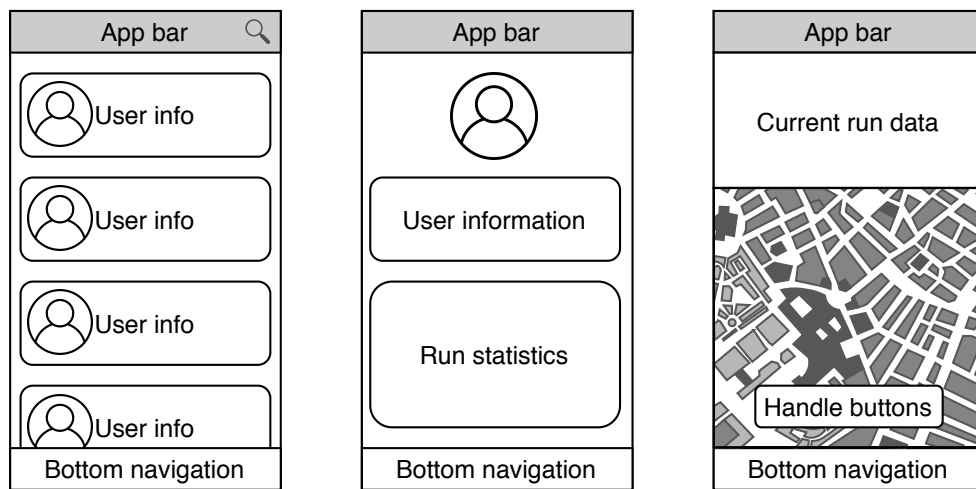
2.2 Struktura databáze

Základní entity a vztahy mezi nimi jsou popsány v konceptuálním schématu databáze na obrázku [2.2]. Databáze obsahuje uživatele, záznam běhu a výzvu. Každý uživatel může mít více záznamů běhu, stejně tak výzev. Výzva patří vždy právě ke dvěma uživatelům, které tím propojuje. Databázi jsem se rozhodla implementovat v konceptu NoSQL, proto tento model slouží pouze pro zjednodušení pochopení struktury databáze. NoSQL databáze jsou implementované jako sady strukturovaných souborů. Umožňují rychlou manipulaci s velkým objemem dat a jednodušší provádění změn [5].

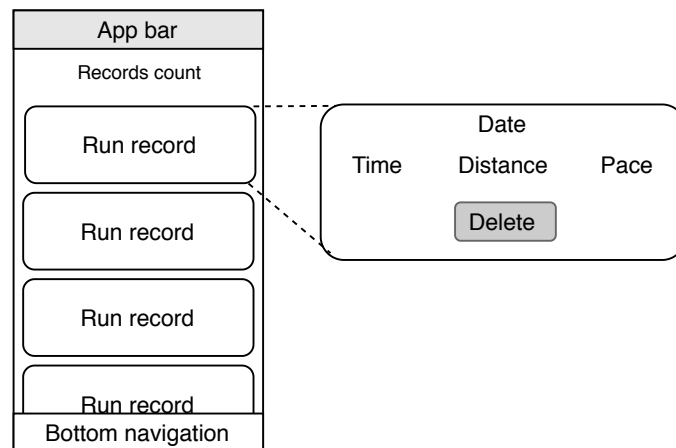
2.3 Uživatelské rozhraní

Při návrhu uživatelského rozhraní jsem dbala především na jednoduchost designu a přehlednost pro uživatele. Hlavní část aplikace bude obsahovat 5 obrazovek - profil, záznam běhu, přehled záznamů, seznam uživatelů a přehled výzev. Mezi jednotlivými obrazovkami se bude přecházet pomocí menu ve spodní části obrazovky.

- **Profil**
Na obrazovce profilu budou obsaženy uživatelské údaje, přehled běžecích statistik a tlačítko pro odhlášení.
- **Záznam běhu**
Obrazovka běhu zobrazí mapový segment, kde se bude vykreslovat trasa aktuálního záznamu, tlačítka pro spuštění a zastavení záznamu a přehled o aktuální vzdálenosti, čase a průměrném tempu běhu.
- **Seznam protivníků**
Zde bude zobrazen seznam nepřátel seřazených podle bodového ohodnocení od uživatele s nejvíce body po uživatele s nejméně body. Rozvržení této obrazovky spolu s obrazovkou profilu a běhu lze vidět na obrázku [2.3].



Obrázek 2.3: Návrhy obrazovek seznamu uživatelů, uživatelského profilu a běžeckého záznamu.



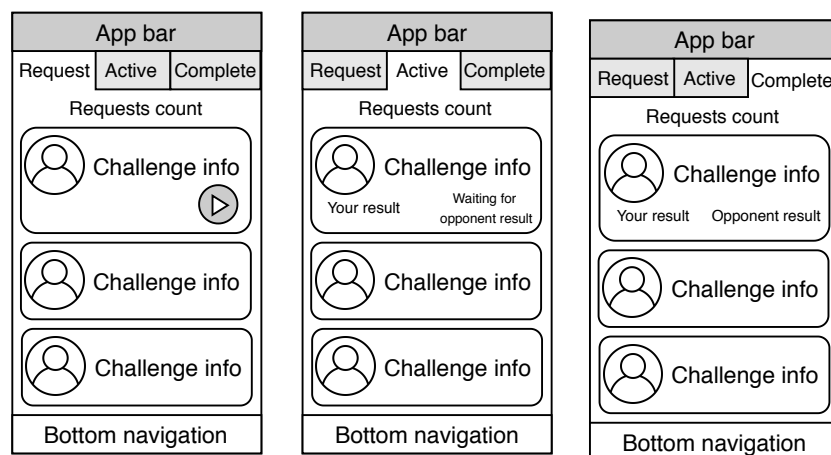
Obrázek 2.4: Obrazovka zobrazení běžeckých záznamů.

- **Přehled běžeckých záznamů**

V záložce aktivit bude přehled uživatelových běžeckých záznamů s informacemi o vzdálenosti, času a tempu běhu a s tlačítkem umožňujícím smazání záznamu [2.4].

- **Přehled výzev**

Zde uživatel nalezne seznam svých aktivních a již splněných výzev [2.5].



Obrázek 2.5: Obrazovky přehledu žádostí o výzvy, aktivních výzev a splněných výzev.

2.4 Výběr technologií

Tato kapitola obsahuje shrnutí základních technologií, které budou při vývoji aplikace využity.

2.4.1 Android platforma

Aplikace bude vytvořena na platformě Android. Aktuální podíl (k březnu 2020) Android OS na trhu činí 70,68 %, čímž se řadí mezi nejrozšířenější mobilní operační systémy na světě [6].

2.4.1.1 Minimální verze API

Od prvotního představení Android OS společností Google v roce 2008 bylo vydáno již mnoho verzí tohoto operačního systému. Aktuálně (ke dni 20. 6. 2020) je k dispozici jako nejnovější Beta verze Android 11 (API 30)[7]. Každá nově vydaná verze přináší vylepšení či rozšíření funkcionality systému a SDK pro vývojáře. Při vývoji Android aplikace je potřeba zvolit minimální verzi API, která bude ještě podporována. V zájmu vývojáře je vybrat takovou verzi, která bude kompromisem mezi množstvím nabízených funkcionalit a procentuálním pokrytím podporovaných zařízení. Pro svou aplikaci jsem vybrala minimální verzi API 21 (Android 5 Lollipop), která je podporována na 94,1 % zařízení [8]. V této verzi byl také poprvé představen Material design, který sjednocuje prvky uživatelského rozhraní Android aplikací [7].

2.4.1.2 Implementační jazyk

Jako implementační jazyk své práce jsem vybrala Kotlin. Google podporuje Kotlin jako hlavní vývojový jazyk pro Android od roku 2017 [9]. Kotlin je kompatibilní s Javou, oproti které je stručnější a přehlednější. Představuje *nullable* datové typy, které poskytují *nullpointer safety*¹. Poskytuje podporu pro lambda výrazy² a coroutines — knihovnu řešící asynchronní operace [[10]].

2.4.2 Firebase

Firebase je platforma poskytující služby usnadňující vývoj webových a mobilních aplikací. Umožňuje ukládání dat do cloud databází, nabízí autentizaci uživatelů, analýzu a testování aplikací a mnoho dalších [11]. Díky těmto funkcionalitám a také skutečnosti, že Firebase nabízí své služby zdarma pro menší, např. školní projekty, jsem si ho vybrala k implementaci backendové části své aplikace.

2.4.2.1 Autentizace

Firebase Authentication poskytuje možnost vytváření uživatelských účtů. Nabízí různé způsoby implementování autentizace – pomocí emailu a hesla, mobilního čísla, přihlášením účtů třetích stran jako je Google, Facebook, Twitter apod. a mnoho dalších. K účtu lze přidat další informace, např. uživatelské jméno či fotku. Účty jsou kompatibilní s dalšími Firebase službami [12].

2.4.2.2 Cloud Firestore

Firestore je cloudová databáze poskytující ukládání dat v konceptu NoSQL. Data nejsou ukládána do tabulek jako v relační databázi, ale jsou strukturovaná do kolekcí dokumentů s hodnotami namapovanými na klíče. Hodnoty mohou být jednoduché datové typy, komplexní objekty nebo pod-kolekce [13]. Výhodou NoSQL je jeho rychlost. Pokud je potřeba načíst data dokumentu obsahujícího pod-kolekce, načtou se pouze data daného dokumentu bez dat pod-kolekce, pokud to není specificky vyžádáno [5].

2.4.2.3 Firebase Storage

Firebase Storage slouží jako úložiště pro mediální soubory, jako jsou fotky a videa [14]. V aplikaci bude využit pro ukládání profilových obrázků uživatelů.

2.4.3 Google Maps

Google poskytuje Maps SDK pro Android, který umožňuje přidávání Google map do aplikací. Mapy jsou interaktivní a je možné provádět s nimi různé

¹Pouze datové typy s přidaným `?` za typem mohou nabývat null hodnoty.

²V Java až od verze 8

operace, jako je přidávání lokalizačních bodů a vykreslování trasy [15]. Tento SDK bude využit pro implementaci zobrazení aktuální polohy uživatele na mapě a vykreslování trasy při záznamu běhu.

2.4.4 Nahrávání a zpracování obrázků

Uživatelé si budou moci zvolit svůj profilový obrázek. Pro výběr a ořezání fotky jsem zvolila knihovnu ImagePicker, která poskytuje jednoduché rozhraní pro přístup do galerie zařízení či vyfocení fotky vestavěnou kamerou a zároveň umožňuje ořezání obrázku do požadovaného tvaru [16]. Obrázky budou ukládány do Firebase Storage a jejich nahrání do požadovaného prvku view bude zajištěno knihovnou Glide, která zprostředkovává nahrávání a zobrazování obrázků a videí [17].

2.4.5 Tvorba grafů

V profilu uživatele bude k dispozici graf s měsíčním souhrnem naběhaných kilometrů. K implementaci tohoto prvku využiji knihovnu MPAndroidChart, která umožňuje snadnou tvorbu komplexních interaktivních grafů v Android Aplikacích [18].

Implementace

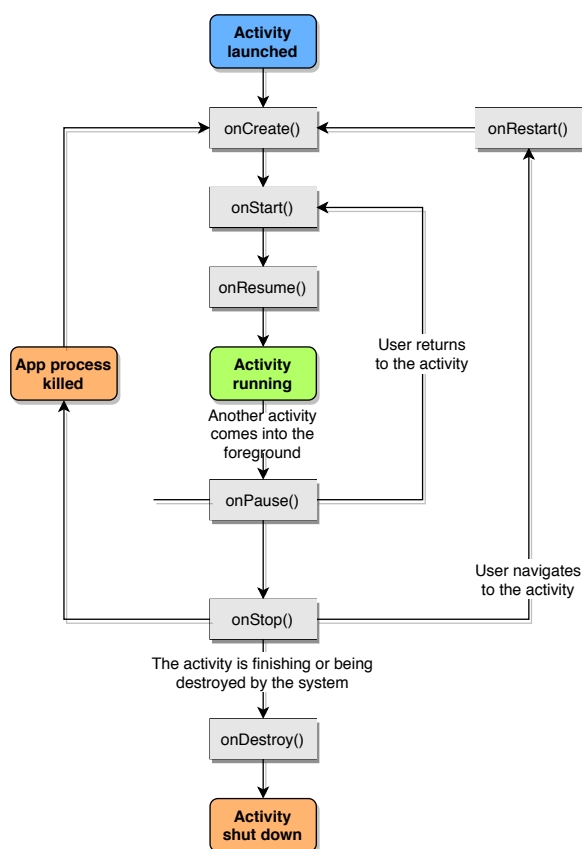
3.1 Základní komponenty

Android aplikace jsou tvořeny jednotlivými komponentami, které mezi sebou komunikují.

- *Activity* je hlavní stavební Android aplikací. Jednu aktivitu si lze představit jako jednu uživatelskou obrazovku. Aktivity slouží jako vstupní body do aplikace [19].
- *Fragment* představuje část uživatelského rozhraní, které je zobrazené v aktivitě. Pokud je obsah obrazovky rozdělen do více částí, je možné jednotlivé části implementovat právě pomocí fragmentu.
- *Service* slouží k zajištění běhu dlouhodobých operací na pozadí, i když aplikace není viditelná uživateli. Typický případ využití service je např. pro přehrávání hudby či sledování polohy uživatele.

Lifecycle

Většina komponent v Android aplikacích má svůj lifecycle, neboli životní cyklus. Komponenty mohou přepisovat jednotlivé metody reagující na události životního cyklu a tím spravovat např. vytvoření obrazovky pro danou komponentu či uvolnění jejích prostředků při jejím zániku. Na obrázku [3.1] je příklad životního cyklu aktivity.



Obrázek 3.1: Diagram životního cyklu aktivity [20]

3.2 Struktura aplikace

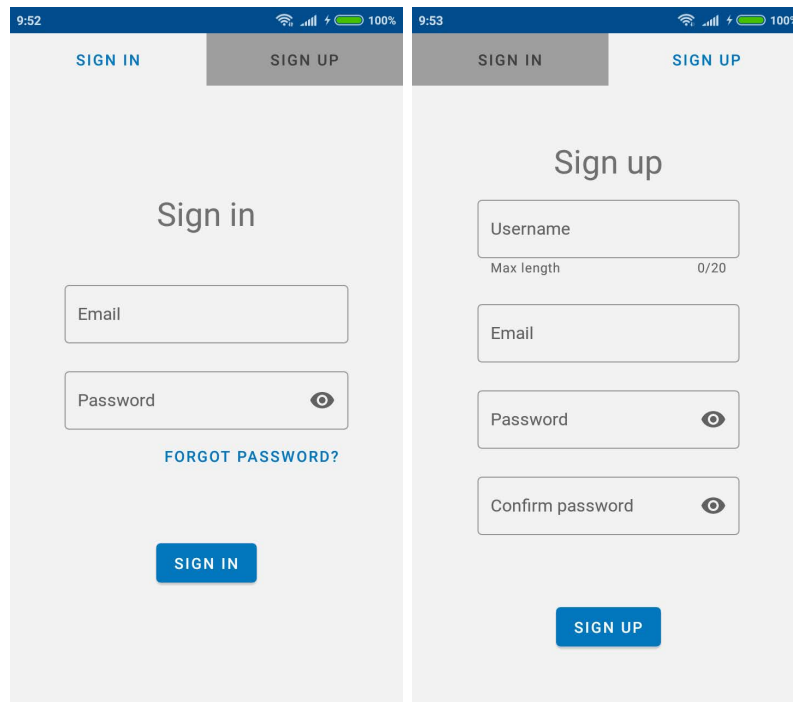
Aplikace se skládá ze dvou hlavních aktivit — *LoginActivity* a *MainActivity*.

3.2.1 LoginActivity

LoginActivity obsahuje dva fragmenty:

- *SignInFragment* představuje přihlašovací obrazovku.
- *SignUpFragment* slouží k zobrazení registrace uživatele.

Pokud není uživatel v aplikaci přihlášen, je přesměrován na login obrazovku. Uživatel se registruje pomocí emailové adresy a hesla, které slouží jako přihlašovací údaje do aplikace. Po úspěšné registraci nebo přihlášení je uživatel navigován do *MainActivity*, která obsahuje hlavní funkcionalitu aplikace. Obrazovku přihlášení a registrace je možné vidět na obrázku [3.2].



Obrázek 3.2: Obrazovky přihlášení a registrace

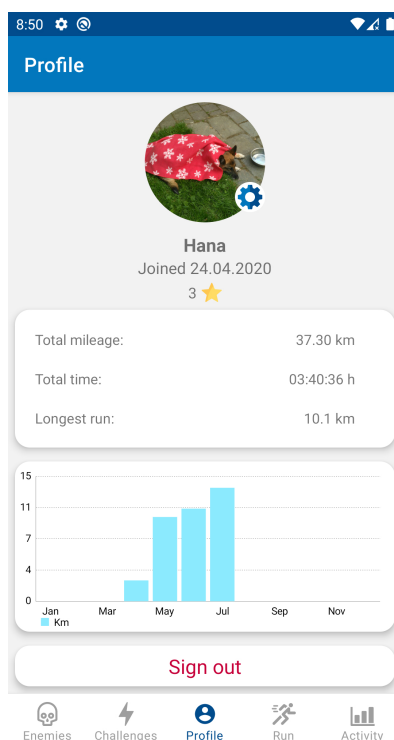
3.2.2 MainActivity

MainActivity představuje hlavní vstupní bod aplikace. Slouží jako kontejner pro jednotlivé fragmenty, které představují dílčí obrazovky, mezi kterými se naviguje pomocí menu. Uživatel se dostane zpět do *LoginActivity* odhlášením z uživatelského účtu.

3.2.2.1 Uživatelský profil

ProfileFragment zobrazuje uživatelský profil se základními informacemi, jako jsou uživatelské jméno, profilový obrázek, datum registrace, počet získaných bodů a přehled běžeckých statistik. Tyto informace jsou viditelné i ostatním uživatelům, kteří si mohou zobrazit profily svých protivníků. Ukázka uživatelského profilu je na obrázku [3.3]. Měsíční přehled naběhaných kilometrů je vytvořený pomocí knihovny MPAndroidChart [2.4.5].

Uživatel má možnost si nastavit si svůj profilový obrázek. Výběr a ořezání obrázku do správného formátu je umožněno pomocí knihovny ImagePicker [2.4.4]. Obrázek je poté uložen pod uživatelským ID do Firebase Storage [2.4.2.3]. Odtud jsou obrázky získávány pomocí Glide z Url [2.4.4] — frameworku pro práci s mediálními soubory.



Obrázek 3.3: Obrazovka uživatelského profilu

3.2.2.2 Záznam běhu

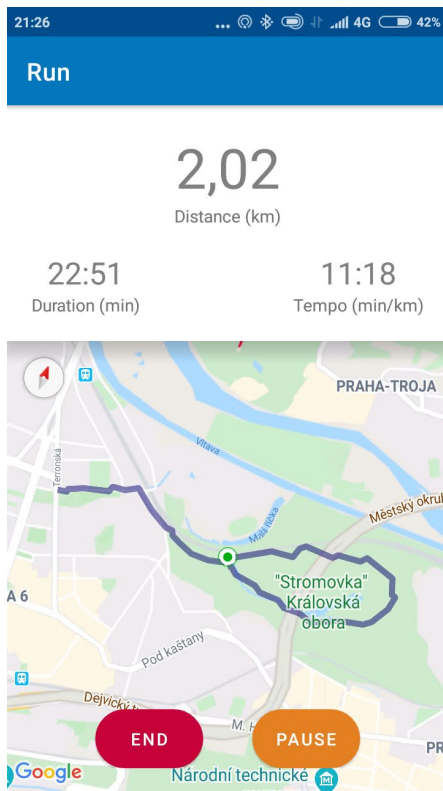
RunFragment představuje část aplikace, kde uživatel spouští záznam svého běhu a může sledovat svůj pohyb na mapě a načítání vzdálenosti, času a svého průměrného tempa, jak je zobrazeno na obrázku [3.4]. Po spuštění může být záznam pozastaven a opět spuštěn. Po ukončení záznamu má uživatel na výběr, zda-li chce záznam uložit. Při spuštěném záznamu běží zároveň notifikace ukazující čas a vzdálenost běhu

3.2.2.3 Přehled záznamů aktivit

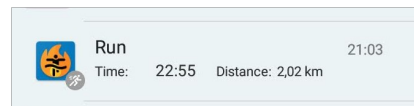
RunRecordsFragment slouží k zobrazení přehledu uživatelových běžeckých záznamů, které jsou seřazeny podle aktuálnosti. Uživatel vidí datum běhu, celkově uběhnutou vzdálenost a čas a průměrné tempo běhu. Po kliknutí na konkrétní záznam se karta záznamu rozbalí a nabídne se tlačítko pro smazání záznamu, jak je vidět na obrázku [3.5].

3.2.2.4 Přehled uživatelů

EnemiesFragment zobrazuje seznam všech protivníků, tedy ostatních uživatelů, seřazených podle počtu bodů. Po kliknutí na jednotlivé položky se rozbalí



a) Obrazovka běhu



b) Notifikace spuštěného běhu

Obrázek 3.4: Zobrazení běžeckého záznamu

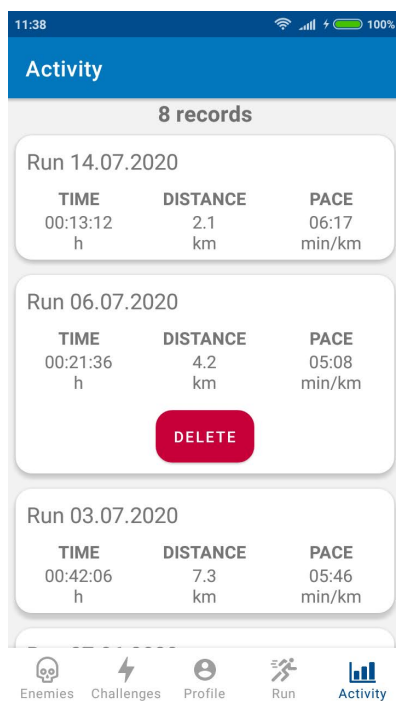
tlačítka na zobrazení profilu daného uživatele a na vyzvání uživatele na výzvu. Obrázek [3.6] ukazuje seznam s jednou rozbalenou kartou.

3.2.2.5 Výzvy

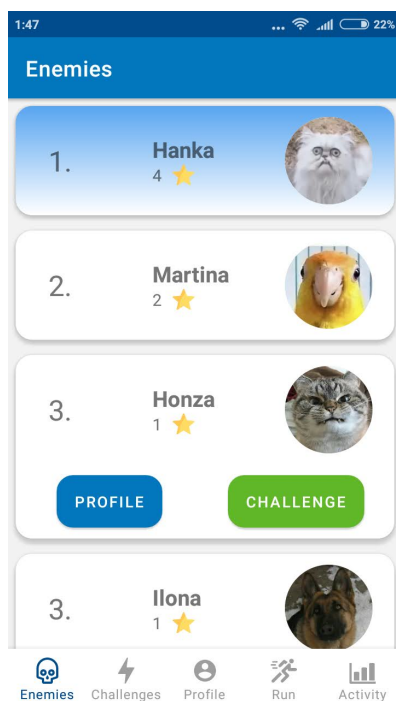
ChallengesFragment obsahuje přehled uživatelských výzev. Výzvy jsou rozděleny na *Requested*, *Active* a *Completed*.

- Záložka *Requested* obsahuje přehled výzev, na které byl uživatel vyzván a musí je teprve začít plnit.
- V *Active* jsou výzvy, na které uživatel vyzval jiné uživatele a čeká na jejich splnění druhou stranou.
- V *Completed* se zobrazují výzvy, které byly již ukončeny a ukazují, kdo je vítězem a kdo poraženým, případně pokud nastala remíza. Na obrázku [3.7] jsou zobrazeny všechny tři záložky s ukázkou výzev.

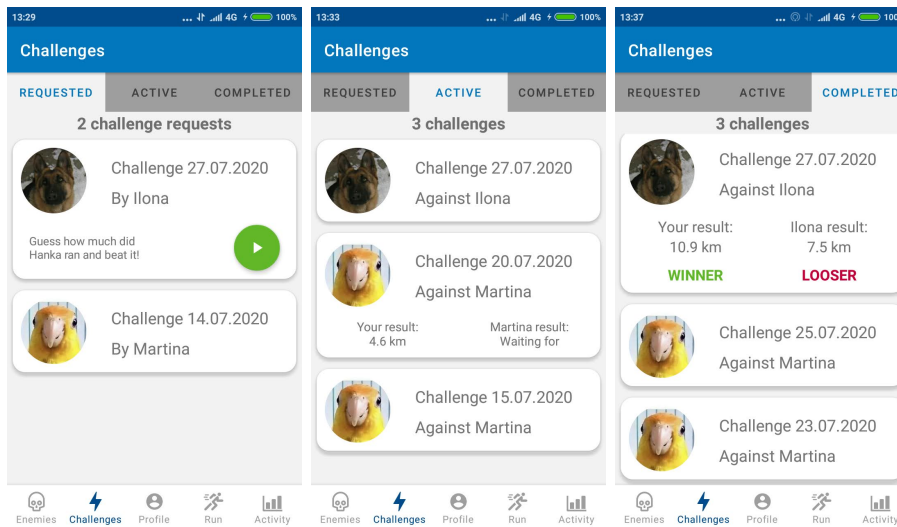
3. IMPLEMENTACE



Obrázek 3.5: Obrazovka přehledu uživatelských běžeckých záznamů



Obrázek 3.6: Obrazovka zobrazení seznamu nepřátel

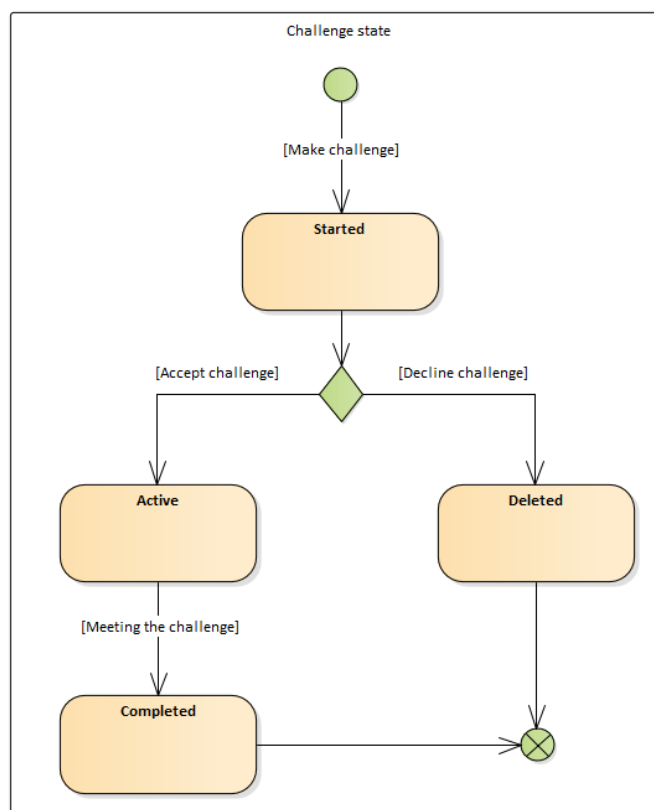


Obrázek 3.7: Obrazovky přehledu žádostí o výzvy, aktivních výzev a ukončených výzev

Diagram průběhu vytvoření a plnění výzvy je znázorněn na obrázku [1.6]. Uživatel vyzve protivníka na výzvu kliknutím na tlačítko *Challenge* na protivnickově kartě v přehledu nepřátel. Zobrazí se dialog, zda-li si uživatel opravdu přeje začít výzvu a pokud ano, je přesměrován na obrazovku se záznamem běhu, kde může svou výzvu započít. Pokud po ukončení běhu uloží záznam, je výzva vytvořena, jinak ne. Uživatel může ale spouštět záznamy běhu, i když nechce zrovna plnit žádnou výzvu nebo nevytváří novou výzvu, ale plní nějakou, na kterou byl vyzván. Pro rozlišení těchto případů a předání argumentů, jako je id nepřítele potřebné pro vytvoření výzvy, jsou využity *arguments* a *actions*, které jsou definovány v navigační komponentě popsané v sekci [3.2.2.6]. Znázornění jednotlivých stavů výzvy je možné vidět na diagramu [3.8].

3.2.2.6 Navigace mezi obrazovkami

Navigation Component umožňuje navigaci mezi jednotlivými komponentami aplikace. Aplikace obsahuje host fragment, v kterém se zobrazují jednotlivé fragmenty definované v *Navigation Graph*. Ten obsahuje všechny informace o jednotlivých přechodech v aplikaci, které navigační komponenta využívá.



Obrázek 3.8: Stavový diagram znázorňující přechody mezi jednotlivými stavy výzvy

3.3 Komunikace mezi vrstvami

Architektura aplikace je rozdělena do tří vrstev podle MVVM vzoru uvedeném v sekci [2.1]. Při komunikaci mezi jednotlivými komponentami je vhodné dodržovat nízkou provázanost komponent.

3.3.1 Observer pattern

Prezenční vrstva aplikace je tvořena aktivitami a fragmenty. Ty se starají primárně o vykreslování uživatelského rozhraní a interakci s uživatelem. ViewModel je třída ze souboru knihoven Android Jetpack, určená ke zpracovávání a ukládání dat UI. ViewModel si nedrží referenci na instanci aktivity či fragmentu, která je s ním spojená. Komunikaci ViewModel a View jsem implementovala pomocí *LiveData*, což je observable datová třída, která respektuje životní cyklus ostatních komponent. Komponenty registrují pozorovatele na těchto proměnných. Při jakékoliv změně je *Observer* upozorněn. Využití *LiveData* je znázorněno v ukázce kódu [1].

```
class ProfileViewModel: ViewModel() {
    val userStatistics: MutableLiveData<ActivityStatistics>
    by lazy { MutableLiveData<ActivityStatistics>() }
    ...
}
class ProfileFragment: Fragment() {
    override fun onCreateView( view: View,
                               savedInstanceState: Bundle?) {
        profileViewModel
            .userStatistics
            .observe(viewLifecycleOwner, Observer {
                setUserStatistics(it)
            })
    }
    ...
}
```

Listing 1: Ukázka využití LiveData

3.3.2 Asynchronní operace

Při komunikaci s datovým úložištěm bylo potřeba implementovat asynchronní operace. Kotlin Coroutines poskytují rozhraní pro jednoduchou práci s asynchronními operacemi, které probíhají v kontextu určitého *CoroutineScope*. ViewModel má nadefinovaný vlastní *viewModelScope* [2], v kterém je možné spouštět coroutines [2.4.1.2].

```
fun getUserData(userId: String) {
    viewModelScope.launch {
        userData.postValue(userFacade.getUser(userId))
    }
}
```

Listing 2: Ukázka využití Kotlin coroutines v Android komponentě

3.4 Sledování polohy

Zapnutí záznamu běhu je implementováno v *RunFragment*. Pokud není záznam spuštěný, pouze se na mapě aktualizuje poloha uživatele. Při spuštěném záznamu je potřeba umožnit sledování polohy i v případě, že je aplikace na pozadí a není uživateli aktuálně viditelná. K tomu slouží service, která zajišťuje běh aplikačních procesů na pozadí.

3.4.1 LocationService

Získávání polohy, i když je aplikace v pozadí, jsem implementovala pomocí *LocationService*. Využila jsem Fused Location Provider API, jenž poskytuje rozhraní pro získávání polohy nadefinované přesnosti, na jejímž základě vybere zdroj lokalizačního signálu. Service komunikuje s fragmentem přes aktivitu, ke které je vázaná, a poskytuje aktuální polohu přes sdílenou instanci *LiveData* v *LocationTrackingRecord*. Pro každý spuštěný záznam se zde načítá vzdálenost a seznam souřadnic.

3.4.2 Notifikace

LocationService je spuštěna foreground, což znamená, že proces této service nebude ukončen systémem při nedostatku prostředků zařízení. Tento typ service se využívá při spouštění procesů na pozadí, které jsou ale pro uživatele částečně viditelné, jako je např. přehrávání hudby či právě aktualizování polohy. Foreground services musí implementovat notifikace. Notifikace se aktualizují podle dat sdílené proměnné *LiveData* v *LocationTrackingRecord*.

3.4.3 Zpracování polohy

LocationTrackingRecord poskytuje seznam souřadnic aktuálního běhu a stejně tak vzdálenost mezi nimi, jelikož ta je potřebná jak pro záznam běhu, tak i notifikace o záznamu. K dalšímu zpracování záznamu, jako je čas a tempo běhu, dochází ve ViewModelu obrazovky běhu, který zajišťuje i následné uložení záznamu.

Testování

Testování jsem prováděla průběžně během vývoje aplikace. Jako autorovi mi byla práce usnadněna přístupem do databáze, kde bylo možné simulovat aktivitu fiktivních uživatelů. Poté, co byla aplikace dokončena a připravena na vydání, bylo provedeno uživatelské testování skupinou testerů, mezi které byla aplikace distribuována pomocí pomocí Firebase App Distribution.

4.1 Testování autorem

K testování aplikace jsem kromě vlastního fyzického zařízení využila emulátor v Android Studiu, který simuluje různé verze Android OS na virtuálních zařízeních s odlišnými parametry, jako je např. velikost obrazovky a její rozlišení.

Pro otestování správné manipulace aplikace s pamětí zařízení jsem využila knihovnu Leak Canary, která pomáhá detekovat memory leaks způsobené aplikací. K těm docházelo převážně v začátcích vývoje, kdy byly časté pády aplikace z různých důvodů.

4.2 Uživatelské testy

Pro uživatelské testování aplikace jsem vybrala tři osoby s různými zkušenostmi s Android aplikacemi a s technologiemi obecně. Jejich přehled je možné vidět v tabulce [4.1]. Testeři měli za úkol provést všechny úkony z následujícího seznamu.

Test 1 Zaregistrujte se do aplikace a změňte si profilový obrázek.

Test 2 Spusťte záznam běhu a po jeho skončení jej uložte.

Test 3 Zobrazte si přehled svých běžeckých záznamů a jeden z nich smažte.

4. TESTOVÁNÍ

	Technická zdatnost uživatele	Zařízení	Verze OS
Tester 1	Nemá mnoho zkušeností s technologiemi, je pouze běžný uživatel chytrého mobilního telefonu.	Samsung Galaxy Edge s7	7.0
Tester 2	Je zdatný v oblasti technologií, ale s Android aplikacemi nemá moc zkušeností.	Samsung Galaxy s9	9.0
Tester 3	Studuje technologickou školu. Má zkušenosti s vývojem Android aplikací.	Xiaomi Redmi 3	5.1.1

Tabulka 4.1: Přehled testerů

Test 4 Vyberte si protivníka a vyzvěte ho na výzvu.

Test 5 Zjistěte, zda-li máte nějaké žádosti o výzvu, a pokud ano, přijměte ji a splňte.

Test 6 Odhlaste se z aplikace.

4.2.1 Test 1

Testovací scénář

1. Uživatel spustí aplikaci, do které není přihlášený.
2. Klikne na záložku *Sign up*.
3. Vyplní požadované údaje a klikne na tlačítko *Sign up*.
4. Na obrazovce profilu klikne na ikonku nastavení u profilového obrázku.
5. Vybere z galerie fotku a ořízne ji a uloží výběr.

Tester 1 začal vyplňovat údaje v kartě přihlášení místo v kartě registrace, po uvědomění chyby pokračoval dále správně.

Tester 2 prošel testovací případ bez problému.

Tester 3 prošel testovací případ bez problému.

4.2.2 Test 2

Testovací scénář

1. Uživatel přejde na obrazovku běhu a stiskne tlačítko *Start*.
2. Pro ukončení záznamu zmáčkne *End*.
3. V dialogu o uložení záznamu zvolí *Yes*.

Tester 1 prošel testovací případ bez problému.

Tester 2 prošel testovací případ bez problému.

Tester 3 prošel testovací případ bez problému.

4.2.3 Test 3

Testovací scénář

1. Uživatel klikne na položku *Activity* v menu.
2. Na obrazovce záznamů klikne na jeden ze záznamů a zmáčkne tlačítko *Delete*.

Tester 1 prošel testovací případ bez problému.

Tester 2 chvíli hledal, kde najít své běžecké záznamy, po nalezení smazal záznam bez problému.

Tester 3 prošel testovací případ bez problému.

4.2.4 Test 4

Testovací scénář

1. Kliknutím na položku *Enemies* v menu se zobrazí uživateli seznam protivníků.
2. Klikne na jednoho z protivníků a zmáčkne rozbalené tlačítko *Challenge*.
3. V zobrazeném dialogu vybere možnost *Yes*, což jej přesměruje na obrazovku běhu.
4. Spustí běžecký záznam a po jeho skončení jej uloží.

Tester 1 hledal chvíli, kde najít seznam protivníků, dále pokračoval bez problémů.

Tester 2 se snažil najít vyzvání přes záložku *Challenges*, poté již zvládl najít protivníka a vyzvat jej.

Tester 3 prošel testovací případ bez problému.

4.2.5 Test 5

Testovací scénář

1. Uživatel přejde na obrazovku *Challenges*.
2. Pokud má v kartě *Requested* zobrazeny nějaké žádosti, klikne na jednu z nich.
3. Karta žádosti se rozbalí a uživatel klikne na zelené tlačítko.
4. V zobrazeném dialogu vybere uživatel možnost *Accept*.
5. Po přesměrování na obrazovku běhu uživatel spustí záznam a po jeho skončení jej uloží.

Tester 1 měl chvíli problém najít, kde se zobrazují žádosti o výzvu od ostatních uživatelů, dále pokračoval bez problému.

Tester 2 měl podobné problémy jako tester 1, po nalezení žádostí pokračoval bez problému.

Tester 3 prošel testovací případ bez problému.

4.2.6 Test 6

Testovací scénář

1. Uživatel přejde na obrazovku profilu.
2. Sescrolluje dolů a klikne na tlačítko *Sign out*.

Tester 1 prošel testovací případ bez problému.

Tester 2 hledal delší dobu, kde nalézt tlačítko na odhlášení. Nakonec mu muselo být poraděno, že je profilová obrazovka scrollovací a tlačítko se nachází dole.

Tester 3 prošel testovací případ bez problému.

4.2.7 Shrnutí

Všichni testeři se pohybovali v aplikaci celkem bez problému. Malé zmatení způsobilo pojmenování přihlašovací karty *Sign in*. Zde přichází ke zvážení přejmenování karty na *Log in*, jelikož s tímto pojmenováním se vyskytuje v aplikacích častěji. Největší problémy působil koncept vyzývání protivníků na výzvy. Uživatelé potřebovali chvíli na zorientování mezi tím, jaký je rozdíl mezi záložkou *Requested* a *Active*. Během uživatelského testování byly odhaleny drobné chyby, jako bylo nechtěné přesměrování na login obrazovku po kliknutí na zpětné tlačítko, pokud byl uživatel na obrazovce profilu, nebo chybějící

povolání manipulace s externím úložištěm zařízení na verzích Androidu vyšších než 9 při měnění profilového obrázku. Všechny chyby byly opraveny. Závěry testování budou zanalyzovány pro vylepšení budoucích verzí aplikace.

Budoucí vylepšení

Prototyp aplikace splňuje všechny specifikované funkční i nefunkční požadavky. Potenciální vylepšení do budoucna se nabízí jak z hlediska optimalizace a architektury aplikace, tak i v podobě nových funkcionalit. Z potenciálních vylepšení se nabízí následující:

- Přihlášení účtem třetích stran jako je Google, Facebook apod.
- Rozšíření herních prvků v aplikaci — komplexnější výzvy, případně hromadné společné výzvy, na které by se mohli uživatelé přihlašovat.
- Vyhledávání a filtrování uživatelů.
- Vyhledávání a filtrování běžeckých záznamů. Zobrazení trasy u záznamu.
- Notifikace při žádosti o výzvu od jiného uživatele.

Závěr

Dle zadání práce byl proveden návrh a realizace prototypu aplikace, jejíž hlavní funkci představuje zaznamenávání uživatelovy běžecské aktivity a propojení běžecské komunity aplikací formou hry.

Na počátku práce jsem popsala základní koncept aplikace a provedla analýzu tří aplikací podobného typu. Na základě popisu konceptu aplikace a analýzy konkurenčních aplikací jsem určila funkční a nefunkční požadavky na aplikaci, jejichž využití jsem poté popsala na případech užití.

Zvolená architektura aplikace se osvědčila při implementaci, jelikož bylo snadné provádět změny jednotlivých komponent bez přílišné nutnosti měnit ostatní komponenty. Návrh uživatelského rozhraní byl založen na jednoduchosti a přehlednosti, což se ukázalo být úspěšné při pozdějším uživatelském testování, kdy uživatelé neměli téměř žádný problém se v aplikaci vyznat.

Využití Firebase platformy pro implementaci backendové části aplikace se ukázalo jako velmi efektivní. Kromě poskytnutí datového úložiště byla platforma využita i pro implementaci autentizace uživatelů a po dokončení aplikace pro distribuci aplikace mezi testery.

Aplikace nabízí prostor pro potenciální budoucí vylepšení. Výsledný prototyp splňuje všechny specifikované požadavky a jeho funkčnost byla řádně otestována.

Bibliografie

1. *Zombies, Run!* [online] [cit. 2020-05-15]. Dostupné z: <https://play.google.com/store/apps/details?id=com.sixtostart.zombiesrunclient>.
2. *Charity Miles* [online] [cit. 2020-05-15]. Dostupné z: <https://play.google.com/store/apps/details?id=com.charitymilescom.android>.
3. *Strava* [online] [cit. 2020-05-15]. Dostupné z: <https://play.google.com/store/apps/details?id=com.strava>.
4. MAXWELL, Eric. The MVC, MVP, and MVVM Smackdown [online] [cit. 2020-05-11]. Dostupné z: <https://academy.realm.io/posts/eric-maxwell-mvc-mvp-and-mvvm-on-android/>.
5. Advantages of NoSQL Databases [online] [cit. 2020-07-16]. Dostupné z: <https://www.mongodb.com/nosql-explained/advantages>.
6. [online] [cit. 2020-05-28]. Dostupné z: <https://gs.statcounter.com/os-market-share/mobile/worldwide>.
7. RAPHAEL, JR. Android versions: A living history from 1.0 to 11 [online] [cit. 2020-07-15]. Dostupné z: <https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html?page=2>.
8. GOOGLE. *Android Studio 4.0.1* [software] [cit. 2020-07-30].
9. MILLER, Paul. Google is adding Kotlin as an official programming language for Android development [online] [cit. 2020-07-15]. Dostupné z: <https://www.theverge.com/2017/5/17/15654988/google-jet-brains-kotlin-programming-language-android-development-io-2017>.
10. SAMOJLO, Gabriel. Pros and Cons of the Kotlin Programming Language for Developing Android Apps. *netguru.com* [online] [cit. 2020-07-15]. Dostupné z: <https://https://www.netguru.com/blog/kotlin-pros-and-cons>.

BIBLIOGRAFIE

11. ESPLIN, Chris. What is Firebase? [online] [cit. 2020-07-16]. Dostupné z: <https://howtofirebase.com/what-is-firebase-fcb8614ba442>.
12. Firebase Authentication [online] [cit. 2020-07-16]. Dostupné z: <https://firebase.google.com/docs/auth>.
13. Cloud Firestore [online] [cit. 2020-07-16]. Dostupné z: <https://firebase.google.com/docs/firestore>.
14. Cloud Storage [online] [cit. 2020-07-16]. Dostupné z: <https://firebase.google.com/docs/storage>.
15. Maps SDK for Android: Overview [online] [cit. 2020-07-16]. Dostupné z: <https://developers.google.com/maps/documentation/android-sdk/overview>.
16. PATEL, Dhaval. *mage Picker Library for Android* [online] [cit. 2020-07-20].
17. Glide [online] [cit. 2020-07-20]. Dostupné z: <https://github.com/bumptech/glide>.
18. JAHODA, Philipp. MPAndroidChart [online] [cit. 2020-07-15]. Dostupné z: <https://github.com/PhilJay/MPAndroidChart>.
19. Introduction to Activities [online] [cit. 2020-07-10]. Dostupné z: <https://developer.android.com/guide/components/activities/intro-activities>.
20. Android Developer Guide: Understand the Activity Lifecycle [online] [cit. 2020-06-01]. Dostupné z: <https://developer.android.com/guide/components/activities/activity-lifecycle>.

Seznam použitých zkratk

API Application interface

SDK Software development kit

UI User interface

XML Extensible markup language

Obsah přiloženého CD

	readme.txt	stručný popis obsahu CD
	apk	adresář se spustitelnou formou implementace
	src	
	impl	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu \LaTeX
	text	text práce
	thesis.pdf	text práce ve formátu PDF