

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačů

## Mobilní aplikace pro sdílení výzev

**Štěpán Bendl**

Vedoucí: Ing. David Kukačka  
Obor: Softwarové inženýrství a technologie  
Květen 2021



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Bendl** Jméno: **Štěpán** Osobní číslo: **483816**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra počítačů**  
Studijní program: **Softwarové inženýrství a technologie**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Mobilní aplikace pro sdílení výzev**

Název bakalářské práce anglicky:

**Mobile application for sharing challenges**

Pokyny pro vypracování:

Definujte záměr aplikace pro sdílení výzev v komunitách. Výzva je chápána jako sázka mezi jednotlivými členy komunity. Popište podobné aplikace dostupné na trhu, případně podobné principy v aplikacích s jiným zaměřením. Definujte cílovou skupinu uživatelů a navrhnete business model této aplikace.

Popište funkcionalitu výsledné aplikace, uživatel bude moci mimo jiné: zakládat výzvy, aktualizovat stav své výzvy, sledovat výzvy svých přátel, v rámci výzev sdílet fotografie, videa, komentáře a další vhodné funkcionality. Práci vhodně doplňte o vybrané diagramy pro popis aplikací, používajících se v softwarovém inženýrství (například Use Case diagram, doménový model atd.).

Navrhnete UI rozhraní aplikace pro mobilní telefon a otestujete je na definované cílové skupině. Následně provedte návrh a implementujte front-end a back-end aplikace ve vybraných technologiích. Výslednou aplikaci řádně zdokumentujte a otestujte.

Seznam doporučené literatury:

1. Stephen M. Garcia, Avishalom Tor & Tyrone M. Schiff, The Psychology of Competition: A Social Comparison Perspective, 8 Persp on Psych Sci 634 (2013). Available at: [https://scholarship.law.nd.edu/law\\_faculty\\_scholarship/941](https://scholarship.law.nd.edu/law_faculty_scholarship/941)
2. Mendoza, A. (2014). Mobile user experience: Patterns to make sense of it all. Amsterdam, Netherlands: Morgan Kaufmann, an imprint of Elsevier., ISBN: 9780124114906
3. Norman, D. A. (2013). The design of everyday things. Cambridge, MA: MIT Press, ISBN: 9780262525671
4. "Developer Guides : Android Developers." Android Developers, [developer.android.com/guide](http://developer.android.com/guide).

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. David Kukačka, Centrum znalostního managementu FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **11.02.2021**

Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **30.09.2022**

Ing. David Kukačka  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

### III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací.  
Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## Poděkování

Nejprve bych rád poděkoval vedoucímu mé závěrečné práce, Ing. Davidu Kukačkovi, za jeho cenné rady, nápady a trpělivost při tvorbě práce. Další poděkování poputuje k mé rodině, která mě podporovala po celou dobu studia, neméně pak při tvorbě závěrečné práce. Nakonec, avšak se stejnou důležitostí, děkuji všem, kteří se na této práci podíleli svými názory a byli ochotni věnovat mi svůj drahocenný čas.

## Prohlášení

Prohlašuji, že jsem tuto práci vypracoval samostatně a že jsem uvedl všechny použité zdroje a literaturu.

V Rakovníku, 19. května 2021

Štěpán Bendl

## Abstrakt

Existuje mnoho aplikací, které podporují sportovní výzvy, avšak je málo takových, které by myslely i na nesporthovce. Cílem této práce je vypracovat analýzu na mobilní aplikaci pro vytváření a sdílení výzev mezi přáteli. Dále aplikaci navrhnout, naimplementovat a otestovat. Nová aplikace by měla vyplnit mezeru na trhu. Bude umožňovat zakládání výzev na jakémkoliv téma. Do výzev pak bude možné přizvat přátele a společně si o něco zasoutěžit.

**Klíčová slova:** mobilní aplikace, android, výzva, firebase, kotlin, NoSQL, analýza, implementace, testování

**Vedoucí:** Ing. David Kukačka

## Abstract

There are a lot of applications that supports sport challenges, although there are very few, which are suitable for non-athletes. The aim of this bachelor thesis is to analyze, design, develop and test mobile application for sharing challenges. This application should fill the gap in the market. It will allow to create challenges on any topic. It will also be possible to invite friends to challenges and compete with each other.

**Keywords:** mobile application, android, challenge, firebase, kotlin, NoSQL, analysis, implementation, testing

**Title translation:** Mobile application for sharing challenges

# Obsah

<b>1 Úvod</b>	<b>1</b>	8.2.1 Realtime Database .....	38
<b>2 Sociologické faktory</b>	<b>3</b>	8.2.2 Authentication .....	39
2.1 Sociologické faktory výzev .....	3	8.2.3 Storage .....	40
<b>3 Analýza trhu</b>	<b>5</b>	8.3 Diagram nasazení .....	40
3.1 Strava .....	5	<b>9 Frontendová architektura</b>	<b>43</b>
3.2 Garmin Connect .....	6	9.1 Model View Controller .....	43
3.3 CHLNGE .....	7	9.2 Model View Presenter .....	44
3.4 Habit Challenge .....	8	9.3 Model View ViewModel .....	45
3.5 Shrnutí .....	9	9.4 Volba architektury .....	46
<b>4 Prostor pro novou aplikaci</b>	<b>11</b>	<b>10 Struktura databáze</b>	<b>47</b>
4.1 Vydefinování nové aplikace .....	11	10.1 Mitigace cyklických závislostí..	47
4.1.1 Shrnutí .....	13	10.2 Zanořování dat .....	47
4.2 Business canvas .....	13	<b>11 Testování</b>	<b>49</b>
<b>5 Uživatelské testování</b>	<b>15</b>	11.1 Dělení testů .....	49
5.1 Popis typických uživatelů .....	15	11.1.1 Automatizované vs manuální	49
5.1.1 Sportovec .....	16	11.1.2 Jednotkové vs integrační ...	50
5.1.2 Umělkyně .....	17	11.1.3 Lokální vs instrumentační ..	52
5.1.3 Trenér .....	18	11.2 Testovací scénáře .....	52
5.2 Testování .....	18	11.2.1 TC1 - Registrace .....	52
5.3 User stories .....	20	11.2.2 TC2 - Přihlášení .....	52
5.3.1 Zobrazení aktivních výzev...	20	11.2.3 TC3 - Vytvoření výzvy ....	53
5.3.2 Připojení k výzvě .....	20	11.2.4 TC4 - Připojení se k výzvě .	53
5.3.3 Vytvoření výzvy .....	20	11.2.5 TC5 - Přidání pokroku	
5.3.4 Zobrazení odznaků .....	21	do výzvy .....	53
5.3.5 Přidání záznamu o splnění ..	21	11.2.6 TC6 - Zobrazení pokroku	
<b>6 Návrh aplikace</b>	<b>23</b>	ve výzvě .....	54
6.1 Diagram tříd .....	23	11.2.7 TC7 - Zobrazení aktivních	
6.2 Use case diagramy .....	24	výzev .....	54
6.2.1 Běžný uživatel .....	25	11.2.8 TC8 - Zobrazení splněných	
6.2.2 Týmový administrátor .....	29	výzev .....	54
6.3 Shrnutí .....	31	11.2.9 TC9 - Vyhledání uživatele ..	54
<b>7 Typy mobilních aplikací</b>	<b>33</b>	11.2.10 TC10 - Přidání přítele ....	54
7.1 Nativní aplikace .....	33	11.2.11 TC11 - Zobrazení aktivity	
7.2 Webové aplikace .....	34	přátel .....	55
7.3 Hybridní aplikace .....	34	11.3 Vyhodnocení testů .....	55
7.4 Podíl na trhu .....	35	<b>12 Prostor pro vylepšení</b>	<b>57</b>
7.5 Volba typu aplikace .....	35	12.1 Zhodnocení aplikace .....	57
<b>8 Technologie použité pro</b>	<b>37</b>	12.2 Možná budoucí vylepšení .....	59
<b>implementaci</b>	<b>37</b>		
8.1 Klientská část .....	37		
8.1.1 Android .....	37		
8.1.2 Kotlin .....	37		
8.1.3 Gradle .....	38		
8.2 Serverová část .....	38		

<b>13 Závěr</b>	<b>61</b>
<b>Literatura</b>	<b>63</b>
<b>A Business canvas</b>	<b>67</b>
<b>B Prototyp aplikace</b>	<b>69</b>
<b>C Výsledky uživatelského testování</b>	<b>71</b>
<b>D Screenshoty z aplikace</b>	<b>73</b>
<b>E Odkaz adresáře projektu na školní GitLab</b>	<b>79</b>



## Obrázky

3.1 Typická funkcionalita aplikace Strava, zdroj:[4]. . . . .	6
3.2 Typická funkcionalita aplikace Garmin Connect, zdroj:[5] . . . . .	7
3.3 Typická funkcionalita aplikace CHLNGE, zdroj:[6] . . . . .	8
3.4 Typická funkcionalita aplikace Habit Challenge, zdroj:[7] . . . . .	8
3.5 Prostor na trhu . . . . .	9
4.1 Prostor pro novou aplikaci . . . . .	12
5.1 Avatar sportovce, zdroj:[9] . . . . .	16
5.2 Avatar umělkyně, zdroj:[10] . . . . .	17
5.3 Avatar trenéra, zdroj:[11] . . . . .	18
6.1 Diagram tříd . . . . .	24
6.2 Use case diagram - běžný uživatel	25
6.3 Use case diagram - týmový administrátor . . . . .	29
7.1 Zastoupení operačních systémů na trhu, zdroj:[13] . . . . .	35
8.1 Firebase Realtime Database, zdroj:[25] . . . . .	39
8.2 Diagram nasazení . . . . .	41
9.1 MVC architektura, zdroj:[29] . . . . .	44
9.2 MVP architektura, zdroj:[30] . . . . .	45
9.3 MVVM architektura, zdroj:[32] . . . . .	46
11.1 Testovací pyramida, zdroj:[36] . . . . .	51
A.1 Business canvas . . . . .	67
B.1 Prototyp aplikace používaný k uživatelskému testování . . . . .	69
D.1 Přihlášení . . . . .	73
D.2 Záložka dashboard . . . . .	74
D.3 Záložka explore . . . . .	75
D.4 Záložka aktivity přátel . . . . .	76
D.5 Detail výzvy . . . . .	77
E.1 QR kód odkaz . . . . .	79

## Tabulky

3.1 Porovnání stávajících aplikací . . . . .	9
4.1 Porovnání aplikací . . . . .	11
4.2 Funkcionalita nové aplikace . . . . .	12
12.1 Implementované funkcionality aplikace . . . . .	58



# Kapitola 1

## Úvod

Myslím, že letíme na Měsíc z toho důvodu, že v povaze lidí je čelit výzvám.

---

*Neil Armstrong - astronaut USA,  
první člověk na Měsíci*

Věta, jež zazněla z úst prvního člověka, který měl možnost spatřit planetu Zemi z povrchu Měsíce se stala inspirací pro tuto bakalářskou práci. Je opravdu v povaze lidí čelit výzvám? Pokud ano, proč tomu tak je? Existuje nástroj, který by umožnil výzvy vytvářet? A jak by takový nástroj měl vypadat? Všechny tyto otázky jsem si položil před začátkem tvorby své bakalářské práce s cílem nalézt v průběhu psaní odpověď. Zároveň je cílem a nedílnou součástí této práce navrhnout a naimplementovat nástroj, který by tvorbu výzev umožňoval. Nástroj by se při tom měl opírat právě o odpovědi na výše zmíněné otázky.

Tato práce se v úvodních kapitolách zabývá analýzou výzev. Proč jsou lidé soutěživí a jak zvýšit jejich soutěživost, jsou otázky, které jsou na základě sociologické studie v těchto kapitolách zodpovězené. Dále je provedena analýza několika již existujících nástrojů, které systém výzev podporují. A protože se práce zabývá tvorbou mobilní aplikace, jsou tyto nástroje vybrány z řad mobilních aplikací. V této kapitole nechybí ani popis možnosti zpeněžení aplikace. Jsou zde popsány možné přístupy, jak zvýšit počet uživatelů a jak přítomnost těchto uživatelů převést ve finanční zisk.

Ve čtvrté kapitole se pak práce zabývá sběrem uživatelských požadavků, jakožto nedílnou součástí vývoje softwaru. Další kapitoly jsou věnovány návrhu aplikace a volbě technologií, které budou pro implementaci použity.

V závěrečných kapitolách jsou mimo jiné popsány typické problémy při vývoji softwaru a zároveň je nastíněno zvolené řešení. Nechybí ani postup použitý při testování aplikace a výčet možných budoucích vylepšení.



## Kapitola 2

### Sociologické faktory

Lidé rádi soutěží. A samozřejmě lidé také rádi vyhrávají. Zde však nastává ten problém, že aby mohl existovat vítěz, musí existovat také poražený. V průběhu dějin lidstva se však zdá, že si lidé soutěžení v té či oné podobě užívají. Od starořeckých olympijských her (sahajících až do roku 776 př. n. l.), až po moderní sporty jako fotbal nebo hokej. Když se podíváme pozorně, všimneme si, že konkurence prostupuje celou moderní společností. Ekonomové označují konkurenci jako nezbytnou sílu pro udržení efektivních a produktivních trhů. Konkurence hraje také velkou roli v politice (volby), mezinárodních vztazích (státy soutěží o moc a zdroje) a samozřejmě je součástí téměř všech sportů. Pro většinu lidí je na povaze soutěže něco nevysvětlitelně přesvědčivého. Možná je to proto, že, jak tvrdí někteří vědci, „konkurenceschopnost“ je biologický rys, který se vyvinul společně se základní potřebou lidského přežití[1].

Průzkum sociologických faktorů je prováděn z důvodu zvýšení šance na získávání nových uživatelů a zároveň jejich následné udržení. Čím více bude aplikace přizpůsobená touhám uživatelů, tím více pro ně bude zábavnější. To zároveň znamená větší šanci, že stávající uživatelé doporučí aplikaci svým přátelům a známým, tedy přivou nové uživatele. A nejlepší požitek z aplikace má právě takový uživatel, který, jak je mimo jiné uvedeno v dalších odstavcích, soutěží se svými přáteli.

#### 2.1 Sociologické faktory výzev

Podle sociologické studie z Notre Dame Law School zvyšují lidskou soutěživost tři faktory, a to:

1. Míra relevance výzvy pro soutěžícího. Tzn. čím je výzva soutěžícímu bližší, tím bude soutěživější. Např. pro sportovce bude výzva v běhu relevantnější než pro programátora.
2. Výkonnost ve srovnání s ostatními soutěžícími. Tedy čím si budou soutěžící výkonnostně blíže, tím pro ně bude výzva zábavnější a budou podávat lepší výkony. Např. profesionální fotograf bude fotit lepší fotografie, pokud se bude snažit překonat rovněž profesionální fotografy, než pokud se bude snažit překonat amatérské fotografy.

3. Soutěživost se rovněž zvyšuje, pokud soutěžíme s někým sobě blízkým. Např. pokud jsou do výzvy zapojeni sourozenci, budou se snažit víc, než pokud by proti sobě soutěžili dva neznámí lidé.

Z těchto faktů si tedy můžeme odnést následující věci:

- Je vhodné podporovat nejen sportovní výzvy, ale poskytnout uživatelům možnost soutěžit v čemkoliv, co je napadne.
- Veřejné výzvy rozdělit do více skupin podle výkonnosti.
- Podpora soukromých (přístupné pouze po pozvání zakladatelem) i veřejných (přístupné pro všechny uživatele) výzev. Uživatelé tak mají možnost vybrat si mezi soutěžení s přáteli nebo otestování svých schopností s větší skupinou.

Dále tato studie ukazuje vývoj soutěživosti v závislosti na počtu účastníků. Studie ukazuje, že s klesajícím počtem účastníků soutěživost roste. Proto je vedle veřejných výzev, které dnes podporuje většina aplikací, dobré podporovat i soukromé výzvy. Uživatelé pak budou více zapojení do výzvy, a tím pádem i soutěživější[2].

## Kapitola 3

### Analýza trhu

Nejvýznamnějšími existujícími řešeními, které systém výzev podporují, jsou dnes aplikace Strava a aplikace Garmin Connect. S roustoucí popularitou těchto aplikací se zvedla také vlna podobných, méně používaných aplikací. Právě již existujícím konkurenčním aplikacím je věnována tato kapitola. Z existujících aplikací si totiž můžeme odnést jak jejich kladné, tak záporné stránky. To pomůže vyvarování se chyb, kterých se aplikace dopustily, či stále dopouštějí. Zároveň se po analýze silných stránek dozvíme, co by nemělo v nově tvořené aplikaci chybět. Tím vzniká základ pro definování nové aplikace. V neposlední řadě analýzou trhu zjišťujeme, zda je pro novou aplikaci na trhu prostor.

Při sestavování analýzy jsem čerpal především z Google Play<sup>1</sup>, neboť se jedná o největší obchod s aplikacemi. AppStore<sup>2</sup> pak obsahuje většinou aplikace se stejným obsahem jako na Google Play.

#### 3.1 Strava

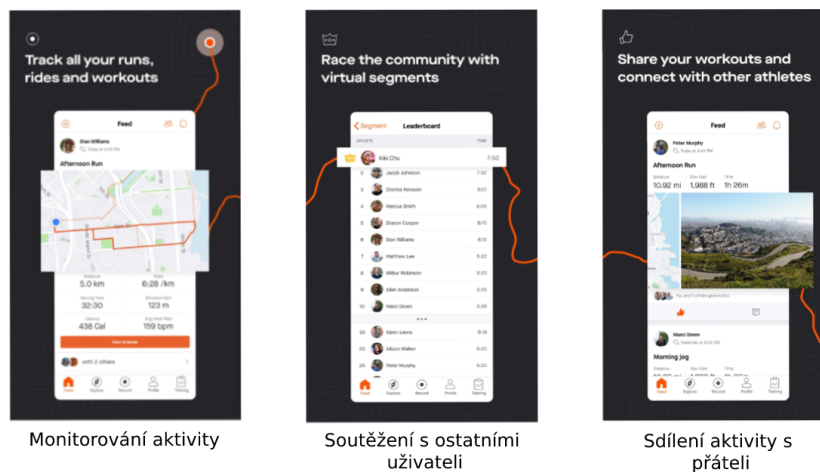
Strava je jednou z nejpoužívanějších aplikací pro sledování a sdílení sportovních aktivit. Nabízí podporu pro běh, jízdu na kole, plavání, chůzi a ostatní cvičení. Dále Strava podporuje sdílení proběhlých aktivit mezi ostatními uživateli. Uživatelé se mohou rovněž přidat do klubů a lépe tak sledovat aktivity ostatních členů klubu. Kdybychom se podívali na systém výzev v této aplikaci, nalezneme mnoho výzev vytvořených vývojovým týmem Stravy. Výzvy se liší v typu sportu a náročnosti. Většina výzev v aplikaci trvá jeden měsíc, tedy začíná první den v měsíci a končí spolu s koncem měsíce. Jakmile uživatel dokončí výzvu, může to sdílet s přáteli ve svých příspěvcích, tzv. „feedu“. Za dokončení výzvy rovněž získá trofej, která je pak vidět na jeho profilu. Zajímavou funkcionalitou jsou rovněž segmenty, kde uživatel může poměřit svůj čas na předem určeném úseku trati.

Aplikace nepodporuje zakládání vlastních výzev, pouze účast na existujících. Strava je rovněž zaměřena pouze na sportovní aktivity. Výzvy jiných typů než sportovních v ní tedy nenajdeme.

<sup>1</sup>Obchod s aplikacemi pro platformu Android.

<sup>2</sup>Obchod s aplikacemi pro platformu iOS.

Mezi největší výhody aplikace patří velká popularita mezi uživateli, možnost sdílení aktivit na jiné sociální sítě a možnost propojení s ostatními aplikacemi (např. Garmin Connect, Samsung Health, Fitbit a jiné)[4].



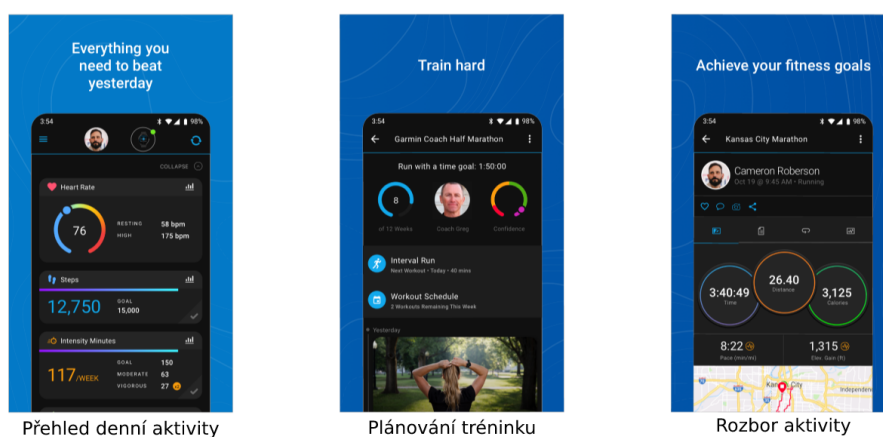
**Obrázek 3.1:** Typická funkcionlita aplikace Strava, zdroj:[4].

## 3.2 Garmin Connect

Aplikace Garmin Connect je podobná aplikaci Strava. Narozdíl od Stravy ovšem cílí na majitele chytrých zařízení značky Garmin, kterým slibuje synchronizaci se smartphonem. Podobně jako ve Stravě je zde možnost zaznamenávání sportovních aktivit, avšak pouze pokud vlastníte chytré zařízení Garmin. Tudíž nelze aktivitu zaznamenávat pouze mobilním telefonem. V případě potřeby je možné aktivitu zadat manuálně.

Aplikace disponuje podobným systémem výzev jako Strava. Můžeme se zde setkat s výzvami, které se liší podle sportu a náročnosti (např. počet km). Výzvy jsou rovněž vždy ohraňeny datem trvání, nejčastěji se jedná o jeden týden nebo jeden měsíc. Za splnění výzvy uživatel dostane odznak, který je pak viditelný ostatními uživateli na jeho profilu. Aplikace podporuje vytváření vlastních výzev a následné vyzvání přátel. Systém vytváření výzev je velmi jednoduchý, uživatel zvolí přátele, které se chystá vyzvat, načej vybere sport a trvání výzvy. Výběr sportů je při vlastních výzvách omezený na chůzi, běh, jízdu na kole a plavání. U uživatelem vytvořených výzev chybí např. nastavení cílového počet kroků, či možnost zvolit odměnu pro vítěze výzvy. Jako nevýhodu aplikace vidím nutnost přihlášení přes účet Garmin. Každý nový uživatel se tedy musí zaregistrovat a nejde se přihlašovat přes účty třetích stran jako Google nebo Facebook. Aplikace však následné propojení s těmito stranami podporuje[5].





Přehled denní aktivity

Plánování tréninku

Rozbor aktivity

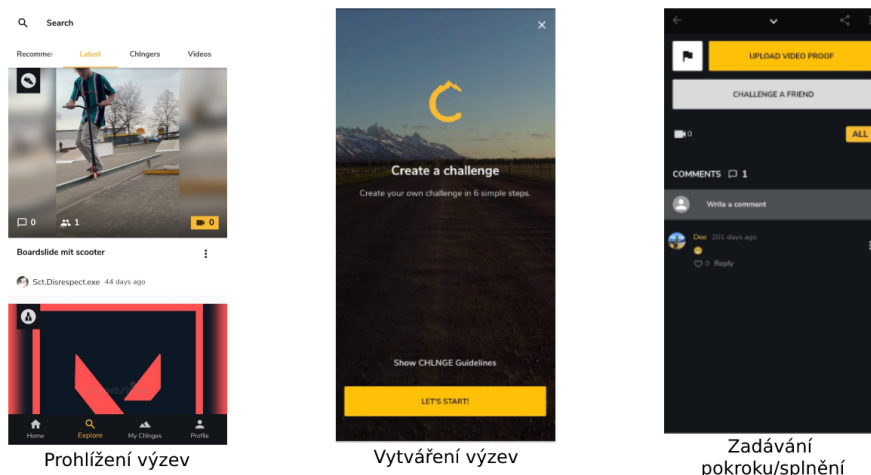
Obrázek 3.2: Typická funkcionalita aplikace Garmin Connect, zdroj:[5]

### 3.3 CHLNGE

Aplikace CHLNGE je typově nejpodobnější aplikaci tvořené v rámci této práce. Umožňuje vytváření a zapojování se do výzev v různých oborech (sport, dovednosti, zábava a jiné). Uživatel může vytvořit buď výzvu veřejnou nebo soukromou, přičemž po vytvoření obou typů je nutná kontrola od vývojářů aplikace. U veřejné výzvy je možné se buď zúčastnit sám, nebo vyzvat přítele. Po vytvoření a schválení soukromé výzvy je možné pozvat více přátel.

Dokončení výzvy uživatel potvrzuje nahráním důkazu o splnění výzvy. Typicky se jedná o krátké video. Ostatní uživatelé pak hodnotí, zda výzvu splnil. V aplikaci není implementována možnost vsázet se s přáteli o určitou věc, možnost nastavení pravidelnosti výzvy (např. „Každý den uběhneme 5 kilometrů“), možnost soutěžit o nejlepší splnění výzvy (např. „Kdo uběhne do konce měsíce nejrychleji 5 kilometrů“) ani možnost nastavit trvání výzvy. Naopak dobře zpracované je hodnocení ostatních uživatelů při plnění výzev[6].

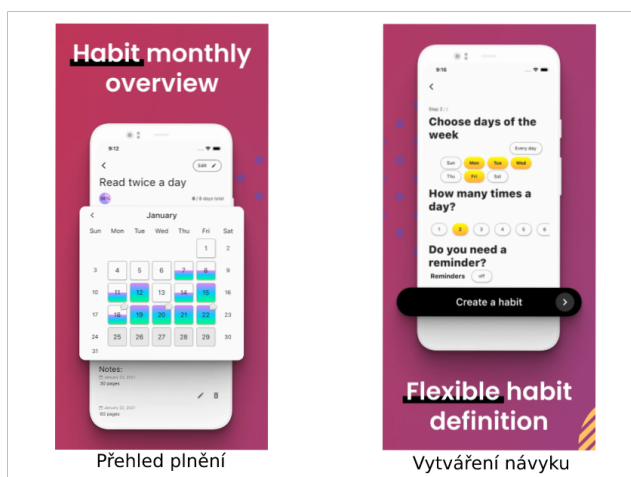
### 3. Analýza trhu



Obrázek 3.3: Typická funkcionalita aplikace CHLNGE, zdroj:[6]

## 3.4 Habit Challenge

Aplikace slouží pro uživatele, kteří si chtějí osvojit návyk. Návyk lze vytvořit jednoduše. Uživatel zvolí jméno návyku a dny, kdy chce tento návyk vykonávat, popřípadě kolikrát denně ho chce vykonávat. Pak už jen zaznamenává, zda za určitý den návyk splnil či nesplnil. Aplikace podporuje zaznamenávání více návyků, jakožto i notifikaci o nesplnění návyku pro daný den[7].



Obrázek 3.4: Typická funkcionalita aplikace Habit Challenge, zdroj:[7]

## 3.5 Shrnutí

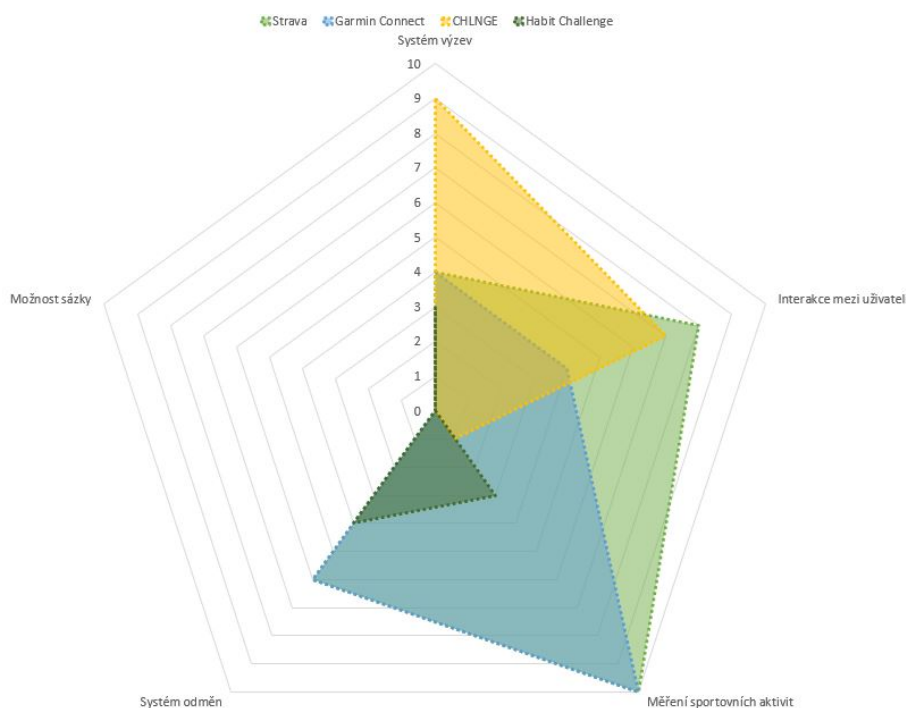
Aplikací na vytváření výzev není mnoho. Trh je nejvíce zaplněn aplikacemi pro sledování sportovních aktivit, které se mimo jiné snaží zabavit uživatele systémem výzev. Aplikace jsou u uživatelů oblíbené, avšak u těchto aplikací chybí možnost vytvořit si vlastní výzvu i v jiném oboru než sport, či lepší možnosti přizpůsobení výzev. Aplikace CHLNGE se zabývá možností zakládání výzev i v jiných oborech než jen sport, je však jedinou takovou aplikací, kterou jsem našel. A i tato aplikace podle mého názoru ponechává mezeru na trhu. Chybí zde větší modifikovatelnost výzev a systém odměn za splnění výzvy.

Funkcionalitu již existujících aplikací jsem se pokusil zachytit do následující tabulky. Aplikace jsem hodnotil na škále 0 - 10 v pěti aspektech. Hodnocení je subjektivní na základě výsledků této kapitoly.

	Strava	Garmin Connect	CHLNGE	Habit Challenge
Systém výzev	4	4	9	3
Interakce mezi uživateli	8	4	7	0
Měření sportovních aktivit	10	10	1	3
Systém odměn	6	6	0	4
Možnost sázky	0	0	0	0

**Tabulka 3.1:** Porovnání stávajících aplikací

Hodnoty z tabulky 3.1 lze následně zpracovat do grafu, který ukazuje mezeru na trhu s mobilními aplikacemi.



**Obrázek 3.5:** Prostor na trhu



## Kapitola 4

### Prostor pro novou aplikaci

V předešlé kapitole jsou popsány konkurenční aplikace s jejich klady i zápory. Tato kapitola se zaměřuje na její výstupy a na jejich základě definuje prostor pro novou aplikaci.

#### 4.1 Vydefinování nové aplikace

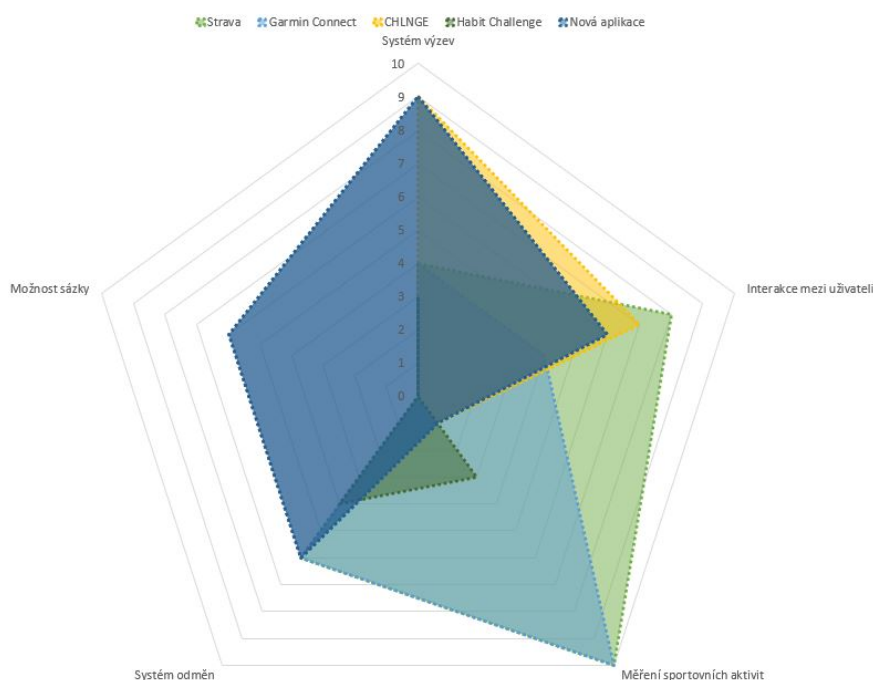
Při definování nové aplikace jsem se zaměřil na předešlou kapitolu, kde jsou porovnávány již existující aplikace. V předešlé kapitole se nachází tabulka 3.1, kde jsou subjektivně ohodnoceny již existující aplikace. Tato tabulka je zde rozšířena o plánované hodnoty nové aplikace.

	Strava	Garmin Connect	CHLNGE	Habit Challenge	Nová aplikace
Systém výzev	4	4	9	3	9
Interakce mezi uživateli	8	4	7	0	6
Měření sportovních aktivit	10	10	1	3	1
Systém odměn	6	6	0	4	6
Možnost sázky	0	0	0	0	6

**Tabulka 4.1:** Porovnání aplikací

Z tabulky 4.1 pak vychází graf 4.1, na kterém je znázorněno, jak by měla nová aplikace vyplnit prostor na trhu. Z grafu vychází, že aplikacím, uvedeným v předchozí kapitole, chybí především možnost sázek a systém odměn. Většina aplikací podporuje interakci mezi uživateli. Z toho si lze vzít ponaučení a rovněž implementovat tuto funkcionalitu v nové aplikaci.

#### 4. Prostor pro novou aplikaci



**Obrázek 4.1:** Prostor pro novou aplikaci

Vše výše uvedené můžeme pak shrnout do tabulky, která znázorňuje jakou funkcionalitu by měla nová aplikace podporovat. V tabulce zároveň nalezneme existující aplikaci, kterou je funkcionalita inspirována, pokud tedy funkcionalita v některé z aplikací existuje. Zde je také potřeba uvést, že v nové aplikaci nebudou všechny funkcionality z časových důvodů opravdu implementovány. Je však vhodné zmínit vše v návrhu, aby v případě potřeby a zájmu byla možnost rozšíření aplikace.

Funkcionalita	Zdrojová aplikace	Součástí této aplikace
Přihlášení přes Google nebo Facebook	Strava	Ano
Feed s novinkami	Strava	Ano
Badge/Trofeje	Garmin Connect/Strava	Ano
Sdílení výzev pouze mezi přáteli	Garmin Connect	Ano
Založení účtu v aplikaci	Garmin Connect	Ano
Zakládání vlastních výzev	CHLNGE	Ano
Přidávání a hodnocení krátkých videí	CHLNGE	Ano
Sdílení výzev se všemi uživateli aplikace	CHLNGE	Ano
Komentování v průběhu výzvy	CHLNGE	Ano
Možnost zvolit parametry výzvy	-	Ano
Možnost zvolit předmět sázky	-	Ano
Možnost zvolit trvání výzvy	-	Ano
Notifikace o nesplněném úkolu	-	Ano
Měření sportovních aktivit	Strava/Garmin Connect	Ne

**Tabulka 4.2:** Funkcionalita nové aplikace

### ■ 4.1.1 Shrnutí

Na předchozích řádcích je rozebráno, která funkcionalita je vhodná a která méně. Zde bych rád vše shrnul dohromady.

Nová aplikace by se měla soustředit v první řadě na výzvy. Měla by umožňovat vytváření výzev, které nebudou pouze ze sportovního odvětví, ale budou pokryta i ostatní odvětví. Při tvorbě výzvy bude uživateli umožněno zvolit velké množství atributů, což zajistí přizpůsobitelnost výzvy.

Dále by v aplikaci neměla chybět podpora pro interakci uživatelů. Tedy sdílení multimédií z doby vykonávání výzvy, reagování na pokrok ostatních uživatelů nebo komentování pokroku. Pokud toto vezmeme v potaz, mělo by se jednat o aplikaci, která bude nejen motivovat své uživatele k plnění výzev, ale rovněž plnit funkci sociální sítě.

Aplikace by rovněž měla mít implementovaný systém odměn. Tím jsou myšleny odznaky za splnění výzvy, podobně jako je tomu v aplikaci Strava nebo Garmin Connect.

Poslední funkcionalitou, kterou zde zmíním, je možnost zvolit předmět sázky. V tom vidím jeden z hlavních rozdílů oproti ostatním aplikacím, které tuto funkcionalitu nepodporují.

Co bych naopak od aplikace neočekával, je měření a zaznamenávání sportovních aktivit. Aplikací s takovou funkcionalitou existuje mnoho a jen stěží je možné vytvořit nástroj, který existující aplikace překoná.

## ■ 4.2 Business canvas

Business canvas je nástroj, který pomáhá pochopit obchodní model. Tvorba a použití vede k poznatkům o zákaznících, jaké hodnotové nabídky se nabízejí prostřednictvím jakých kanálů a jak je možné produkt zpeněžit. Jelikož při větším množství uživatelů nebude provoz aplikace zdarma, je vhodné se zamyslet i nad možnostmi monetizování uživatelů aplikace. Stejně tak pokud by aplikaci chtěla zpracovat firma, bude od aplikace očekávat výdělek.

V rámci této práce jsem business canvas rovněž zpracoval (viz. příloha A). Jsou v něm znázorněni klíčoví partneři, finance, infrastruktura a zákazníci.

V business canvasu nalezneme několik aktivit, které by měly pomoci k získání uživatelů po startu aplikace. Pro aplikace, které mají příjem založený na reklamách, je totiž počet uživatelů klíčový. Platí zde rovnice "více uživatelům bude zobrazeno více reklam", a tudíž s počtem uživatelů roste příjem.

Prvním prvkem jsou akce označené v business canvasu jako "Invitation events". Předpokladem zde je, že uživatelé aplikace dostanou za pozvání určitého počtu nových uživatelů na určitou dobu verzi bez reklam, která je jinak zpoplatněná. Stávající uživatelé tak nejen získají nové vyzyvatele, ale zároveň získají i zpoplatněnou funkcionalitu zdarma. Jedná se o klasický referral program<sup>1</sup>, který můžeme vidět např. u poskytovatelů internetu, internetových obchodů či jiných služeb.

<sup>1</sup>System, ve kterém za pozvání nového uživatele získáte odměnu.





## Kapitola 5

### Uživatelské testování

Uživatelské testování nad prototypem přináší výhody brzké zpětné vazby potencionálních uživatelů. Mimo to, že se dozvíme, zda je o aplikaci zájem, můžeme i pozorovat interakci testerů s prototypem. Na to pak můžeme reagovat pozměněním návrhu či při implementaci.

Tato kapitola popisuje celý průchod uživatelským testováním. Od přípravy, přes samotné testování na potencionálních uživateliích po vyhodnocení výsledků testování.

#### 5.1 Popis typických uživatelů

Typičtí uživatelé neboli uživatelské osobnosti jsou charakteristiky imaginárních uživatelů, kteří odrážejí uživatelskou základnu aplikace. Osobnosti jsou psány, jako by byl uživatel skutečný, ale zároveň poukazují na společné vlastnosti napříč určitou skupinou lidí. V nejlepším případě by měly vlastnosti typických uživatelů odpovídat vlastnostem reálných uživatelů.

Definování typických uživatelů poskytuje realistický odraz toho, jaké skupiny uživatelů můžeme očekávat. Uživatelská osobnost pomáhá pochopit, jak se uživatel chová, co zažívá a jaké jsou jeho cíle. Poskytuje vývojáři empatii s koncovým uživatelem. To umožňuje vyhnout se vývoji nepotřebných funkcí a zaměřit se na to, co by osobě uživatele bylo užitečné a praktické[8].

### ■ 5.1.1 Sportovec



**Obrázek 5.1:** Avatar sportovce, zdroj:[9]

**Jméno:** Jan Novák

**Věk:** 24 let

**Pohlaví:** Muž

**Povolání:** Finanční poradce

**Záliby:**

- Sportování - lezení na stěně, fotbal, lyžování
- Cestování
- Nové technologie
- Trávení času s přáteli

**Technická zručnost:**

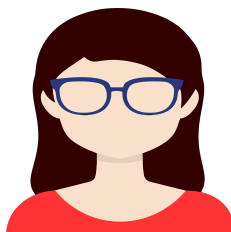
- Počítač - Průměrná
- Internet - Nadprůměrná
- Mobilní telefon (Android) - Průměrná

**Příklad použití** V pondělí a ve středu ráno chodím běhat. V mém fotbalovém týmu chodíme všichni alespoň jednou týdně. Každý měsíc soutěžíme, kdo naběhává nejvíce. Po běhu si dám snídani a odcházím do práce. Po práci jdu buď domů, nebo na fotbalový trénink.

V pátek večer chodím s přáteli lézt na stěnu. Vždy, když se objeví nějaká nová těžká cesta, tak se vsadíme. Ten, kdo ji vyleze poslední, platí večeři. O víkendu mám většinou fotbalový zápas. V týmu si vedeme různé statistiky. Třeba na počet vstřelených gólů nebo počet karet.

V zimě jezdím s přáteli na týden na lyže. Vsázíme se, kdo za týden najezdí nejvíce kilometrů.

## ■ 5.1.2 Umělkyně



**Obrázek 5.2:** Avatar umělkyně, zdroj:[10]

**Jméno:** Alžběta Králová

**Věk:** 28 let

**Pohlaví:** Žena

**Povolání:** Úřednice

**Záliby:**

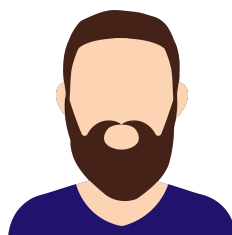
- Kreslení
- Návštěvy uměleckých výstav
- Fotografování

**Technická zručnost:**

- Počítač - Podprůměrná
- Internet - Průměrná
- Mobilní telefon (Android) - Nadprůměrná

**Příklad použití:** Baví mě kreslit a s kamarádkami každý týden pořádáme soutěž o nejhezčí obraz. Vybereme nějaké zajímavé téma, nakreslíme obrázek, ten pak vyfotíme a vybereme vítěze. S přítelem také rádi fotíme. Baví mě se s ním předhánět, kdo zachytí daný moment lépe. Mám pocit, že nás to oba ve fotografování posouvá dál. Často se také účastníme různých fotografických soutěží. Jednou za čas také mezi svými přáteli fotografy uspořádáme fotografickou soutěž my. Je to zábava.

### 5.1.3 Trenér



Obrázek 5.3: Avatar trenéra, zdroj:[11]

**Jméno:** Josef Pospíšil

**Věk:** 38 let

**Pohlaví:** Muž

**Povolání:** Kondiční trenér

**Záliby:**

- Sportování - posilování, florbal
- Čtení knih
- Tvorba videí

**Technická zručnost:**

- Počítač - Nadprůměrná
- Internet - Průměrná
- Mobilní telefon (Android) - Nadprůměrná

**Příklad použití** Pro své klienty rád vymýšlím různé výzvy. Každý měsíc vyberu jeden cvik, a kdo se ve cviku nejvíce zlepší, ten vyhraje lekci zdarma. Třeba kliky. Na začátku udělám se všemi klienty tolik kliků, kolik zvládnou. Stejně tak na konci. Kdo se zlepší o největší množství, vyhrává lekci. Pro klienty je to skvělá motivace.

Také trénuji florbal, konkrétně mladší žáky. V oddílu se jednou za čas domluvíme a uděláme soutěž mezi věkovými kategoriemi. Soutěž trvá většinou 14 dní a je typu „Která kategorie naběhá nejvíce kilometrů, vyhrává večeri.“. Děti to nejen baví, ale i tím přirozeně rozvíjí fyzickou kondici a soutěživost.

## 5.2 Testování

V týdnu od 23.11.2020 do 29.11.2020 proběhlo uživatelské testování. Předem byl vytvořený low-fidelity prototyp aplikace, na kterém měli respondenti

plnit úkoly. Prototyp lze nálezt v příloze B. Pro tvorbu úkolů byly použité user stories. Odezvy respondentů a odpovědi na otázky týkající se jak úkolů, tak jejich názoru na aplikaci, byly zaznamenány do tabulky. Prototyp jsem testoval postupně na čtyřech respondentech. Každý uživatelský test trval přibližně 20 minut. Nejprve byli respondenti seznámeni s konceptem aplikace, poté plnili úkoly a odpovídali na mé dotazy a nakonec jsem se zeptal na jejich pohled na tento typ aplikace. Po celou dobu měli respondenti zapnuté sdílení obrazovky.

### Úkoly, které měli respondenti splnit:

1. Zjistěte, v jakých výzvách jste zapojený(á) a jaký je váš aktuální pokrok.
2. Vytvořte výzvu a pozvěte do ní své přátele.
3. Zjistěte, kdo jsou vaši přátelé. Pozvěte jednoho přítele do výzvy, které se také účastníte.
4. Připojte se k již existující výzvě. Zjistěte, co je potřeba udělat pro splnění a zaznamenejte svoji aktivitu.
5. Zjistěte, jaké výzvy už jste splnil(a).

### Z odpovědí respondentů (viz. příloha C) vychází následující vylepšení:

- Zvýraznit výzvy, ve kterých je uživatel zapojený.
- Přidat postup uživatele v jednotlivých výzvách na hlavní obrazovku.
- K výzvám přidat žebříček s pořadím a pokrokem účastníků.
- Záznamy o aktivitách přátel odebrat z hlavní obrazovky.
- Možnost zrušit již zasloupanou pozvánku.
- Přidat potvrzení o pozvání přítele do výzvy.
- Možnost pozvat uživatele do výzvy přímo z obrazovky výzvy.
- Přidat historii výzev.
- Přidat kategorie výzev.
- Nezobrazovat všechny aktivity přátel, ale pouze důležité milníky v rámci výzev.
- Pozvat přátele, kteří ještě aplikaci nepoužívají.
- Přidat možnost podněcovat přátele k soutěžení v reálném čase.
- Přidat možnost založení a následného připojení se do týmu.
- Přidat možnost soutěžení v rámci týmu.

- Přidat možnost soutěžení mezi týmy.

Z odpovědí respondentů bylo patrné, že v práci je potřeba ještě zapracovat na designu a uživatelské přívětivosti aplikace. Hlavní poznámky jsou již uvedeny výše. Dále z odpovědí respondentů vyplývá, že by danou aplikaci používali a ve většině případů by ji doporučili i známým. Přínosné rovněž bylo, že každý z respondentů navrhl, co by se v aplikaci dalo zlepšit, či co do aplikace přidat.

## ■ 5.3 User stories

Úkoly k uživatelskému testování byly vytvořeny na základě následujících user stories.

### ■ 5.3.1 Zobrazení aktivních výzev

**Název:** Zobrazení aktivních výzev

**User story:** Jako uživatel chci mít možnost zobrazit své aktivní výzvy, protože chci mít přehled v jakých výzvách jsem zapojený.

**Akceptační kritéria:** Uživatel si může zobrazit výzvy, do kterých je zapojený.

### ■ 5.3.2 Připojení k výzvě

**Název:** Připojení k výzvě

**User story:** Jako uživatel chci mít možnost připojit se k výzvě, protože chci soutěžit se svými přáteli.

**Akceptační kritéria:** Uživatel se může připojit do výzvy a přidávat záznamy o pokroku.

### ■ 5.3.3 Vytvoření výzvy

**Název:** Vytvoření výzvy

**User story:** Jako uživatel chci mít možnost založit výzvu, protože chci uspořádat výzvu pro své přátele.

**Akceptační kritéria:** Uživatel má možnost založit výzvu.

#### ■ 5.3.4 Zobrazení odznaků

**Název:** Zobrazení odznaků

**User story:** Jako uživatel chci mít možnost zobrazit odznaky, protože chci vidět, jaké výzvy jiný uživatel splnil.

**Akceptační kritéria:** Uživatel má možnost zobrazit odznaky za splněné výzvy na profilu přítele.

#### ■ 5.3.5 Přidání záznamu o splnění

**Název:** Přidání záznamu o splnění (pokroku)

**User story:** Jako uživatel chci mít možnost přidat záznam o splnění, protože chci zaznamenávat svůj postup ve výzvě.

**Akceptační kritéria:** Uživatel má možnost přidat záznam o splnění (pokroku).





# Kapitola 6

## Návrh aplikace

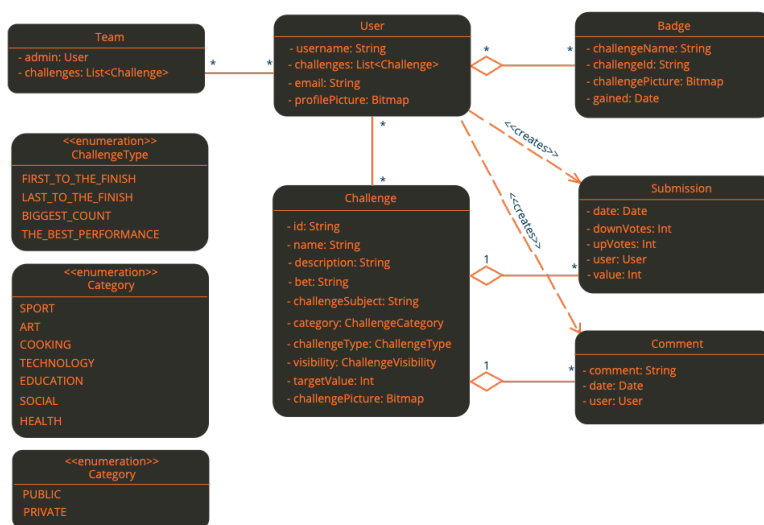
Touto kapitolou se dostáváme od analýzy, která byla popsána v předchozích kapitolách, k návrhu. Kapitola obsahuje typické diagramy používané v softwarovém inženýrství, podle kterých pak bude aplikace implementována.

### 6.1 Diagram tříd

Diagram tříd byl navržený s ohledem na analytickou část práce. Z té plyne, že většina funkcionality bude obsahovat výzvu jako hlavního aktéra. Jelikož by výzva měla být plně přizpůsobitelná potřebám uživatele, obsahuje i třída výzvy atributy, které jí činí přizpůsobitelnou. Nejzajímavějším z atributů je „ChallengeType“, tedy typ výzvy. Zde můžeme vidět několik typů:

- `FIRST_TO_THE_FINISH` - Slouží pro výzvy typu „Kdo první splní činnost, vyhrává“. Např. Kdo první uběhne 25 kilometrů, vyhrává vstupenku do aquaparku.
- `BIGGEST_COUNT` - Slouží pro výzvy typu „Kdo udělá nejvíce opakování, vyhrává“. Např. Kdo udělá nejvíce kliků bez přestávky, vyhrává permanentku do posilovny.
- `THE_BEST_PERFORMANCE` - Slouží pro výzvy typu „Kdo udělá danou činnost nejlépe, vyhrává“. Např. Kdo vyfotí nejhezčí fotografii, vyhrává poukaz na zájezd.
- `LAST_TO_THE_FINISH` - Slouží pro výzvy typu „Kdo bude poslední, prohrává“. Např. Kdo skončí v závodě poslední, platí všem ostatním soutěžícím večeři.

Mimo těchto předdefinovaných atributů má výzva také složitější atributy jako například „ChallengeSubject“, který reprezentuje činnost potřebnou pro splnění výzvy. Dalším takovým atributem je „Bet“. Ten reprezentuje předmět sázky, tedy o co se uživatelé vsadili. Uživatel do výzvy přidává „Submission“, tedy záznam o pokroku ve výzvě. Podle toho je pak počítán jeho pokrok ve výzvě. Je zde také znázorněna možnost přidání se do týmu.



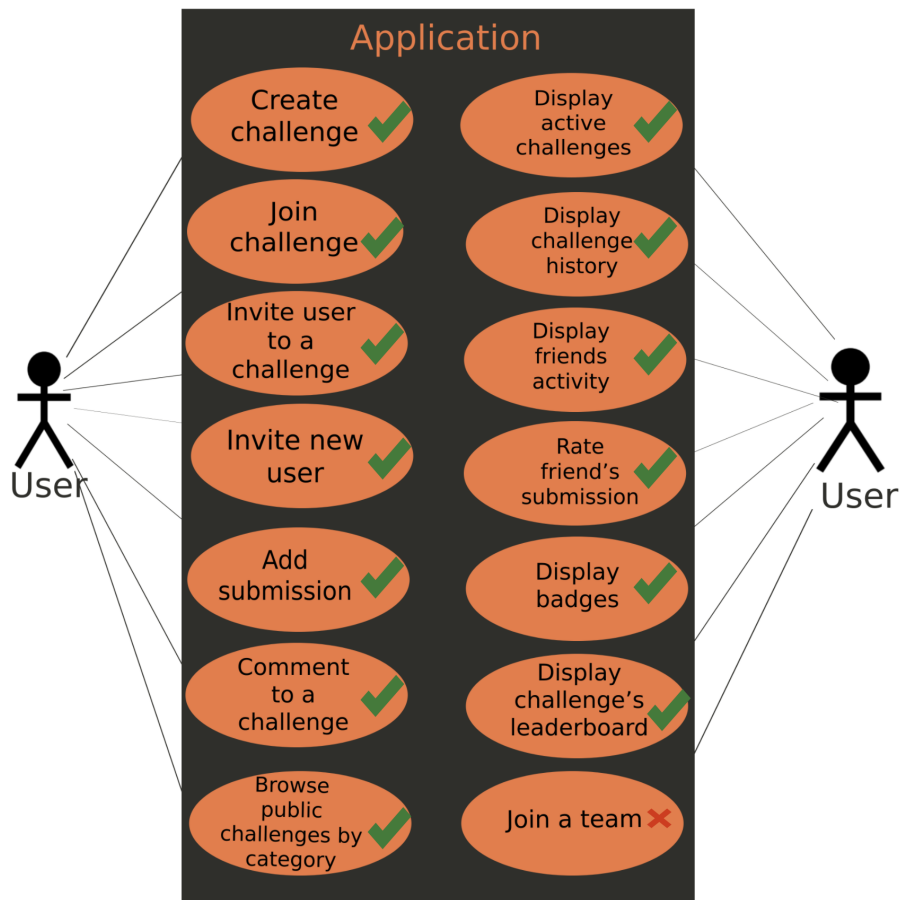
Obrázek 6.1: Diagram tříd

## 6.2 Use case diagramy

Diagram případů užití je rozdělen na dva typy. První je diagram případů užití pro běžného uživatele, následuje diagram případů užití pro týmového administrátora. U diagramu běžného uživatele jsou označeny zeleným znakem ty případy užití, které jsou v plánu pro implementaci v rámci této práce. Červeným znakem jsou označeny ty, jejichž implementace v plánu v rámci této práce není. U diagramu týmového administrátora tyto znaky nejsou k nalezení. Důvodem je, že implementování funkcionality pro týmy není v rámci této práce v plánu.

Scénáře pro případy užití jsou vytvořeny pouze pro vybrané případy užití. Vytváření scénářů pro všechny případy užití je nad rámec této práce.

### 6.2.1 Běžný uživatel



Obrázek 6.2: Use case diagram - běžný uživatel

## ■ Scénář 1

**Název:** Připojit se k výzvě

**Předpoklady:**

- Uživatel je přihlášený do aplikace
- Uživatel se neúčastní zvolené výzvy

**Základní scénář:**

1. Uživatel se chce připojit k výzvě
2. Uživatel najde výzvu v seznamu výzev
3. Uživatel přejde na detail výzvy
4. Systém zobrazí detail výzvy
5. Uživatel najde tlačítko "Připojit se" a klikne na něj
6. Systém zobrazí potvrzovací dialog
7. Uživatel potvrdí připojení
8. Systém zobrazí hlášku potvrzující připojení do výzvy

**Alternativní scénáře:**

- Pozvánka
  - Začátek: 2.
  - Sjednocení: 3.
    1. Uživatel dostane pozvánku do výzvy
- Odmítnutí připojení
  - Začátek: 7.
  - Sjednocení: 5.
    1. Uživatel nepotvrdí připojení do výzvy
    2. Systém zobrazí detail výzvy

## ■ Scénář 2

**Název:** Pozvat uživatele do výzvy

### **Předpoklady:**

- Uživatel je přihlášený do aplikace
- Vyzyvatel se účastní zvolené výzvy
- Pozvaný uživatel se neúčastní zvolené výzvy

### **Základní scénář:**

1. Uživatel chce pozvat jiného uživatele do výzvy
2. Uživatel najde výzvu v aplikaci a přejde na detail výzvy
3. Systém zobrazí detail výzvy
4. Uživatel najde tlačítko "Pozvat přítele" a klikne na něj
5. Systém zobrazí obrazovku s přáteli
6. Uživatel zvolí přítele, kterého chce pozvat
7. Systém zobrazí potvrzovací dialog
8. Uživatel potvrdí pozvání
9. Systém zobrazí hlášku s potvrzením pozvání

### **Alternativní scénáře:**

- Žádní dostupní přátelé
  - Začátek: 5.
  - Sjednocení: 6.
- 1. Uživatel nemá žádného přítele, který by se již výzvy neúčastnil
- 2. Systém zobrazí obrazovku pro přidávání přátel
- 3. Uživatel si najde jiného uživatele a přidá si ho jako přítele
- 4. Systém zobrazí potvrzovací hlášku o přidání přítele

### ■ Scénář 3

**Název:** Pozvat nového uživatele do aplikace

#### **Předpoklady:**

- Uživatel je přihlášený do aplikace

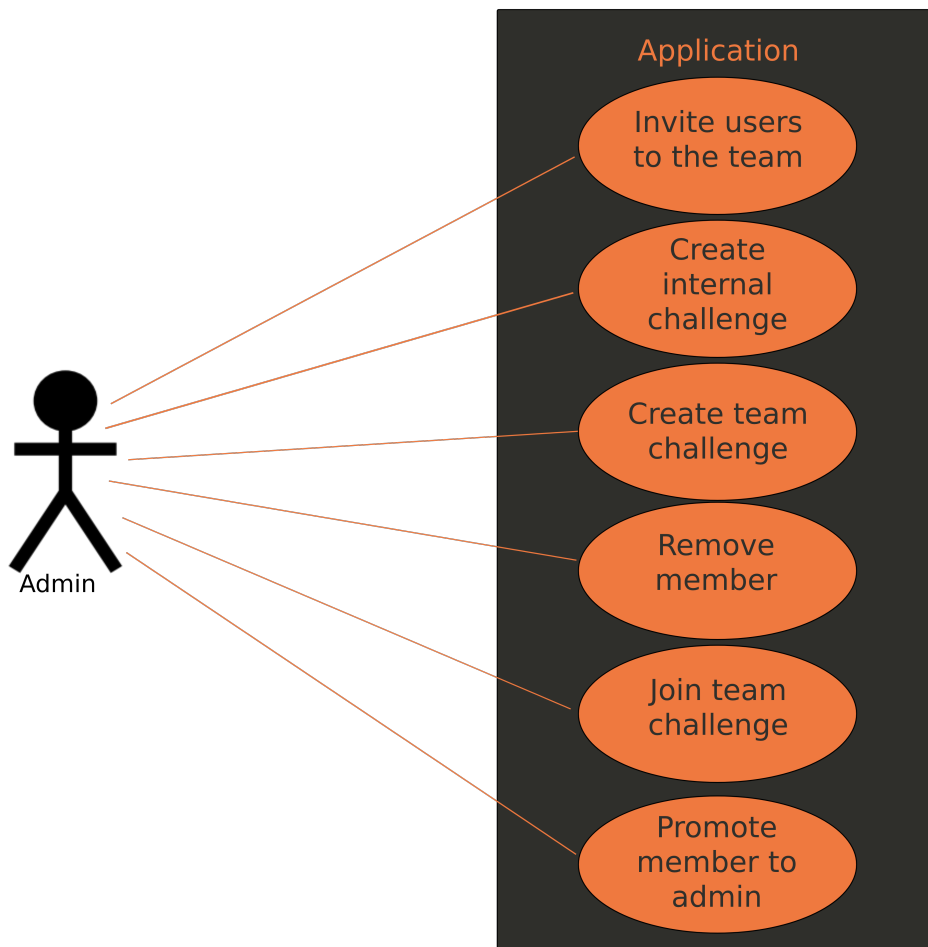
#### **Základní scénář:**

1. Uživatel chce pozvat nového uživatele k používání aplikace
2. Uživatel přejde na obrazovku, kde se nachází tlačítko "Pozvat" a klikne na něj
3. Systém zobrazí obrazovku s detaily pozvánky
4. Uživatel vyplní údaje do pozvánky
5. Systém zkontroluje, zda jsou všechny údaje v pořádku
6. Systém zobrazí indikaci, že je možné pozvat specifikovaného uživatele
7. Uživatel klikne na tlačítko "Odeslat pozvánku"
8. Systém odešle pozvánku podle specifikace
9. Systém zobrazí hlášku s potvrzením pozvání

#### **Alternativní scénáře:**

- Uživatel již existuje
  - Začátek: 6.
  - Sjednocení: 6.
    1. Uživatel, který má být do aplikace pozváný je již v aplikaci registrovaný
    2. Systém zobrazí indikaci, že specifikovaný uživatel již existuje
- Chybné parametry
  - Začátek: 6.
  - Sjednocení: 6.
    1. Některý ze zadaných parametrů není validní
    2. Systém zobrazí indikaci u parametru, který není validní
    3. Uživatel opraví označený parametr

### 6.2.2 Týmový administrátor



Obrázek 6.3: Use case diagram - týmový administrátor

## ■ Scénář 1

**Název:** Vytvořit mezitýmovou výzvu

### **Předpoklady:**

- Uživatel je přihlášený do aplikace
- Uživatel je administrátor týmu

### **Základní scénář:**

1. Uživatel chce vytvořit týmovou výzvu
2. Uživatel zvolí přechod na obrazovku vytváření výzev
3. Systém zobrazí obrazovku pro vytváření výzev
4. Uživatel vyplní atributy výzvy
5. Uživatel zvolí týmovou výzvu na obrazovce pro vytváření výzev
6. Systém zkontroluje validitu atributů výzvy
7. Systém indikuje správnost atributů
8. Uživatel klikne na tlačítko "Vytvořit"
9. Systém zobrazí hlášku o úspěšném vytvoření

### **Alternativní scénáře:**

- Nevalidní atributy výzvy
  - Začátek: 7.
  - Sjednocení: 7.
    1. Některý ze zadaných atributů není validní
    2. Systém označí nevalidní atribut
    3. Uživatel nahradí nevalidní atribut za validní
- Pozvaný tým nepřijme pozvání
  - Začátek: 11.
  - Sjednocení: 11.
    1. Pozvaný tým nepřijme pozvání do výzvy
    2. Systém upozorní zakladatele výzvy na nepřijetí pozvání



## ■ 6.3 Shrnutí

Z návrhu aplikace plyne, jaké by měly být vazby mezi jednotlivými entitami, a jaké funkcionality by měla výsledná aplikace podporovat. Diagram tříd ukazuje vazbu téměř všech tříd na třídu „Challenge“, což odpovídá závěrům z analytické části práce, tedy vytvořit aplikaci, jejíž hlavní funkcionality se pohybuje okolo výzev. Doménový model pak doplňují diagramy případů užití. Ty ukazují jaké funkcionality bude mít uživatel k dispozici při práci s aplikací.

Jazyk UML nabízí kromě dvou výše zmíněných typů diagramů i mnoho dalších. Pro potřeby této aplikace jsou však dle mého názoru tyto dva typy diagramů dostatečně vypovídající a dobře znázorňují podstatu tvořené aplikace.



# Kapitola 7

## Typy mobilních aplikací

V dnešní době můžeme narazit na vícero typů mobilních aplikací. Jedno ze základních dělení je na nativní, webové a hybridní aplikace. V této kapitole jsou popsány uvedené typy aplikací podrobněji a nechybí ani výčet výhod a omezení. Dalším typickým dělením, se kterým dnes už přišel do styku téměř každý, je dělení na aplikace pro platformu Android a pro platformu iOS. Toto rozdělení má mnoho společných aspektů s rozdělením „nativní-webová-hybridní“ a i toto rozdělení je v zahrnuto v této kapitole.

### 7.1 Nativní aplikace

Pojmem nativní aplikace je obvykle označována taková aplikace, která byla napsána pomocí nativního vývojového jazyka a nástrojů specifických pro danou platformu. Např.: Nativní aplikace pro iOS byla napsána pomocí jazyků Swift či Objective-C. Naopak nativní aplikace pro Android byla napsána pomocí jazyků Kotlin nebo Java.

Protože jsou tyto aplikace vyvíjeny pomocí platformově specifických řešení, má vývojář úplný a snadnější přístup ke všem možnostem zařízení. Těmito možnostmi je myšlen např. přístup k sensorům zařízení, zapisování a čtení ze souborového systému nebo ovládání gesty. Nativní aplikace mají rovněž tendenci být výkonější, jelikož jejich kód je blíže k hardwarovému vybavení mobilního zařízení.

Nevýhoda nativních aplikací je již zmíněná platformová specifická. To mimo jiné znamená, že žádná aplikace napsaná pro iOS pomocí jednoho z programovacích jazyků pro iOS nepůjde spustit v systému Android. Stejně tomu je i naopak. Z toho vyplývá jednoduchý fakt, pokud chceme vyvíjet nativní aplikaci, která bude fungovat na obou těchto platformách, musíme aplikaci vyvinout specialně pro každou platformu. Tento způsob je pracnější a nákladnější.

Tento druh aplikací je rovněž dostupný téměř výhradně v obchodě s aplikacemi. Pro Android je to GooglePlay, pro iOS je to AppStore. To sebou přináší nevýhody v podobě nutnosti dodržování stanovených pravidel a regulací[12].

## 7.2 Webové aplikace

Pro některé vývojáře je tato nejistota ohledně regulací a schvalování aplikací nežádoucí. Řešením je pak webová aplikace. Ta dává vývojářům svobodu nabízet svojí aplikaci mimo obchody s aplikacemi a zároveň nabízet své aplikace rovněž uživatelům desktopových zařízení. Jedná se o aplikaci napsanou pomocí HTML, CSS a JavaScriptu, pro který je možné využít širokou škálu knihoven. Jednou z dalších výhod je, že aplikaci lze vytvořit pro všechny platformy jedním kódem. Není tedy potřeba aplikaci programovat zvlášť pro každou platformu, jako tomu je u nativních aplikací.

Při používání tohoto přístupu je však mobilní aplikace omezena na možnosti uživatelova mobilního prohlížeče. To například znamená, že vývojář ztrácí přístup k sensorům zařízení nebo k souborovému systému. Jednou z hlavních nevýhod tohoto přístupu je také to, že aplikace potřebuje internetové připojení po celou dobu provozu. Výzvou při vývoji webové mobilní aplikace je rovněž to, že mnoho běžných ovládacích prvků uživatelského rozhraní nativně neexistuje a je nutné je znovu vytvořit. To pak může vést k tomu, že aplikace nebude fungovat zcela podle očekávání uživatele[12].

## 7.3 Hybridní aplikace

Toto řešení je kombinací mezi oběma řešeními výše, odtud plyne název hybridní aplikace. Jádro aplikace je psáno pomocí webových technologií (HTML, CSS, JavaScript), které jsou poté zapouzdřeny v nativní aplikaci. Díky použití pluginů mohou mít tyto aplikace téměř plný přístup k funkcím mobilního zařízení. I přesto však můžeme narazit při vývoji mobilní aplikace na mnoho problémů s přístupem k těmto funkcím. Obecně platí, že každá platforma může vyžadovat jiný přístup k určitým funkcím, načež je potřeba řešení tohoto problému vývojářem. To vyžaduje čas a zdroje navíc.

Srdcem hybridní mobilní aplikace je stále jen aplikace napsaná pomocí HTML, CSS a JavaScriptu. Místo toho, aby se aplikace zobrazovala v prohlížeči uživatele, je spuštěna z nativní aplikace a z vlastního integrovaného prohlížeče, který je pro uživatele v podstatě neviditelný. Například aplikace pro iOS by k zobrazení naší aplikace používala WKWebView, zatímco v Androidu by k provedení stejné funkce používala prvek WebView.

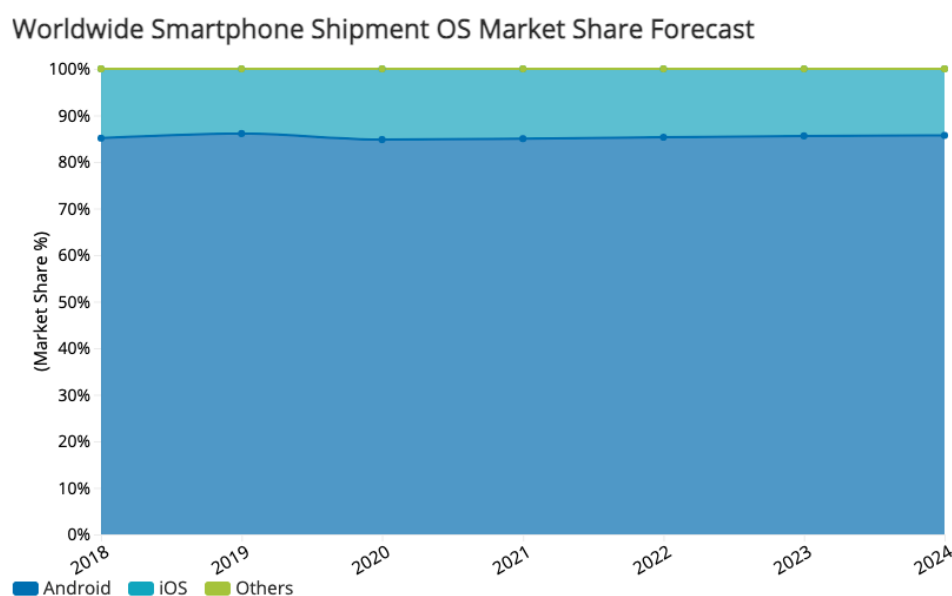
Takovýto kód je poté vložen do obálky nativní aplikace pomocí nástrojů, jako například Apache Cordova. Řešení tohoto typu vytvářejí nativní aplikace, které slouží jako schránka pro prvky webview (WKWebView pro iOS, WebView pro Android). Do webview komponenty se poté načte určená webová aplikace.

Tato možnost sebou bezpochyby nese i své nevýhody. Tou největší je zde nutnost znovuvytváření prvků uživatelského rozhraní. S tím se pojí i fakt, že aplikace pro Android a pro iOS mají většinou rozdílné prvky uživatelského rozhraní a stejně tak se liší jejich skladba na obrazovce. V praxi to tedy znamená, že uživatelské rozhraní je nutno naprogramovat pro každou

platformu zvlášť[12].

## 7.4 Podíl na trhu

Důležitým aspektem při rozhodování, jaký typ aplikace zvolit je i podíl jednotlivých platforem na trhu s chytrými telefony. Z grafu na obrázku 7.1 vyplývá, že na trhu převládají mobilní zařízení s operačním systémem Android. Konkrétně se jedná o zhruba 85% trhu oproti 15% u operačního systému od firmy Apple. Na grafu je rovněž zachycena predikce vývoje trhu. Zde nevidíme žádný předpokládaný zvrát.



Obrázek 7.1: Zastoupení operačních systémů na trhu, zdroj:[13]

## 7.5 Volba typu aplikace

Vezmeme-li v potaz všechny výše uvedené výhody a nevýhody, vychází nám dvě možná řešení. Prvním je tvorba nativní aplikace a druhým tvorba hybridní aplikace.

Při tvorbě nativní aplikace je vývoj aplikace pro obě platformy, tedy jedné aplikace pro Android a jedné pro iOS, nad rámec této práce. Vzhledem k zastoupení operačních systémů na trhu se zde pak nabízí nativní aplikace pro operační systém Android.

V případě vývoje hybridní aplikace by odpadal problém volby operačního systému. Hybridní aplikace by pak zároveň byla kompatibilní s téměř 100% zařízení trhu. Nevýhodou hybridní aplikace je však tvorba dvou uživatelských rozhraní, což ve výsledku opět vede k tvorbě dvou aplikací s tou výhodou, že by se za uživatelským rozhraním skrývala totožná rozhodovací logika.

S přihlédnutím k mému zájmu o systém Android a zároveň k rozsahu práce byla zvolena nativní aplikace pro operační systém Android. Vývoj aplikace pro platformu iOS je ponechán jako další možnost k rozšíření této práce.

# Kapitola 8

## Technologie použité pro implementaci

Vyvíjenou aplikaci můžeme rozdělit na dvě části, klientskou část a serverovou část. V této kapitole jsou představeny jednotlivé technologie použité k vývoji každé z částí.

### 8.1 Klientská část

Pro klientskou část byly zvoleny technologie Android SDK v kombinaci s programovacím jazykem Kotlin a nástrojem pro sestavování programů Gradle. Kombinace těchto tří technologií je momentálně vývojáři nejoblíbenější a zároveň nejpoužívanější při vývoji aplikací pro platformu Android[14]. Právě z tohoto důvodu je tato kombinace zvolená i při implementaci této práce. Technologie jsou v této kapitole krátce popsány.

#### 8.1.1 Android

Android je operační systém vyvíjený společností Google pod open-source licenci<sup>1</sup>. Pro tvorbu Android aplikací je nutné použít kolekci nástrojů a knihoven určených právě pro vývoj takovýchto aplikací. Tato sada se nazývá Android SDK[15] a obsahuje mimo jiné například ukázkové zdrojové kódy aplikací nebo emulátory mobilních zařízení s operačním systémem Android. Android SDK je vydáváno společností Google společně s novou verzí operačního systému Android a zajišťuje kompatibilitu s nejnovějšími funkcemi systému.

#### 8.1.2 Kotlin

Kotlin je staticky typovaný programovací jazyk běžící nad JVM<sup>2</sup>. Jeho hlavním vývojářem je tým programátorů ze společnosti JetBrains. I když není Kotlin syntakticky kompatibilní s Javou<sup>3</sup>, je navržen pro interoperabilitu

<sup>1</sup>Open-source licence ve zkratce umožňuje svobodné používání, úpravy a sdílení softwaru[16].

<sup>2</sup>Java Virtual Machine je program, jehož účelem je spouštět jiné programy. JVM má dvě primární funkce: umožnit běh programů Java na jakémkoli zařízení nebo operačním systému a spravovat a optimalizovat programovou paměť[17].

<sup>3</sup>Java je programovací jazyk a výpočetní platforma, kterou společnost Sun Microsystems poprvé vydala v roce 1995[18].

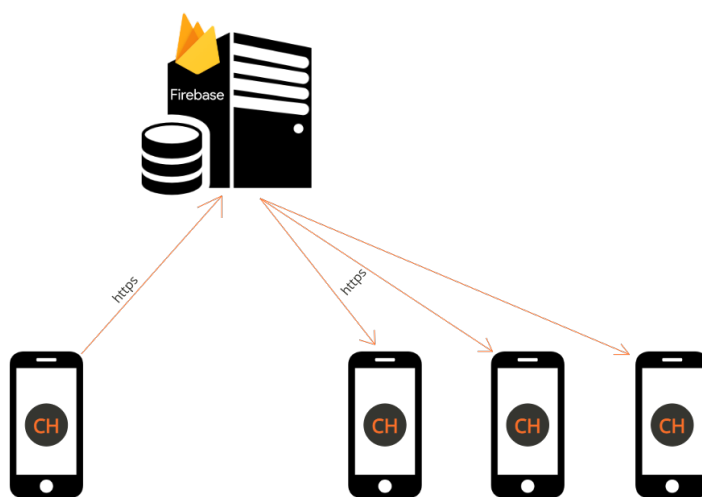




vývoje. Zde přichází na pomoc služba Realtime database. Jedná se o NoSQL<sup>5</sup> databázi, která obsahuje mnoho užitečných funkcí.

Tato služba umožňuje synchronizovat data mezi uživateli v reálném čase. To usnadňuje uživatelům aplikace přístup k datům z jakéhokoliv zařízení a zároveň umožňuje snadnější spolupráci. Při každém nahrání dat na server služba Realtime database uloží data do cloudu<sup>6</sup> a zároveň upozorní ostatní klientské aplikace na změnu dat.

Další výhodou Realtime Database je podpora pro zařízení bez připojení k internetu. Kdykoliv se uživatel odpojí od internetu, Realtime Database využívá lokální úložiště pro ukládání a načítání dat. Po připojení k internetu jsou pak data synchronizována se serverem[25].



**Obrázek 8.1:** Firebase Realtime Database, zdroj:[25]

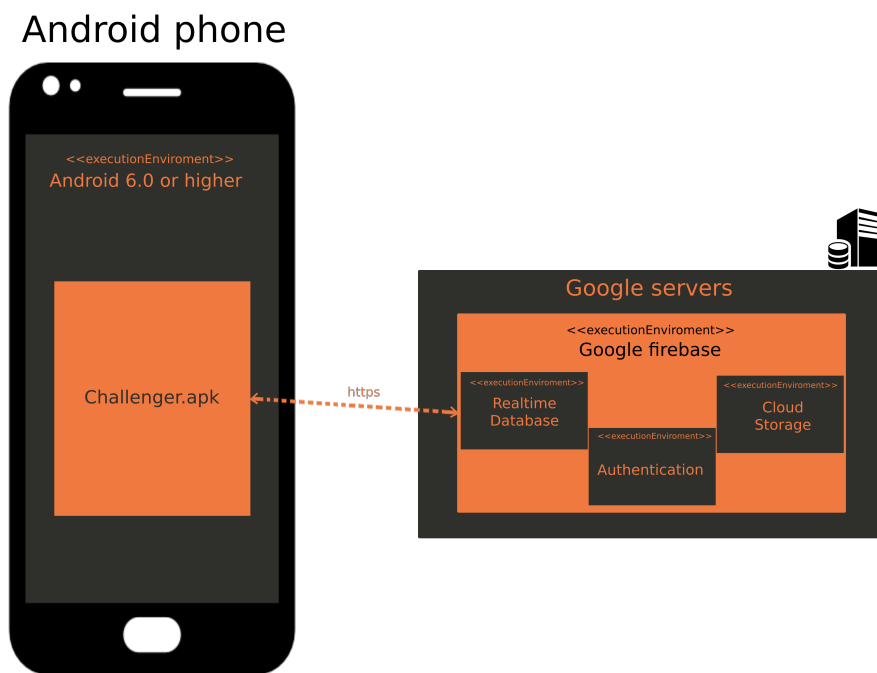
### 8.2.2 Authentication

Aplikace obvykle potřebují autentizovat uživatele, aby přizpůsobily svůj vzhled, obsah a zároveň udržovali klientská data v bezpečí. Služba Firebase Authentication nejen že tuto funkcionalitu umožňuje, ale zároveň je i přívětivá

<sup>5</sup>Databáze NoSQL neukládají data stejně jako relační databáze do relačních tabulek. Existují různé typy NoSQL databází na základě jejich datového modelu. Mezi nejčastěji používané typy patří dokumentové databáze, pár klíč – hodnota, široký sloupec a graf. NoSQL databáze poskytují flexibilní schémata a jsou snadno škálovatelné s velkým množstvím dat a velkým zatížením uživatelů[23].

<sup>6</sup>Cloud storage je model cloud computingu, který ukládá data na internetu prostřednictvím poskytovatele cloud computingu. Poskytovatel spravuje a provozuje datové úložiště jako službu. Tím vývojář eliminuje nákup a správu vaší vlastní infrastruktury pro ukládání dat. To způsobuje větší míru volnosti, globální měřítko a odolnost vůči poškození dat. Další výhodou je možnost přístupu k datům „kdykoli a kdekoli“[24].





Obrázek 8.2: Diagram nasazení



## Kapitola 9

### Frontendová architektura

Přístupy k organizaci komponent Android aplikací se za posledních několik let značně posunuly. Komunita se vzdálila od poněkud monolitického modelu Model View Controller (MVC) ve prospěch modulárnějších, lépe testovatelných architektur[28].

Model View Presenter (MVP) a Model View ViewModel (MVVM) jsou dvě z momentálně nejpoužívanějších architektur pro tvorbu Android aplikací. V komunitních fórech proběhlo mnoho diskuzí o tom, která z těchto architektur je lepší, kde jsou jejich výhody a naopak v čem spočívají slabiny. V této kapitole jsou postupně představeny všechny tři zmíněné architektury. V poslední sekci je pak zvolena architektura, která bude použita pro tvorbu aplikace tvořené v rámci této práce.

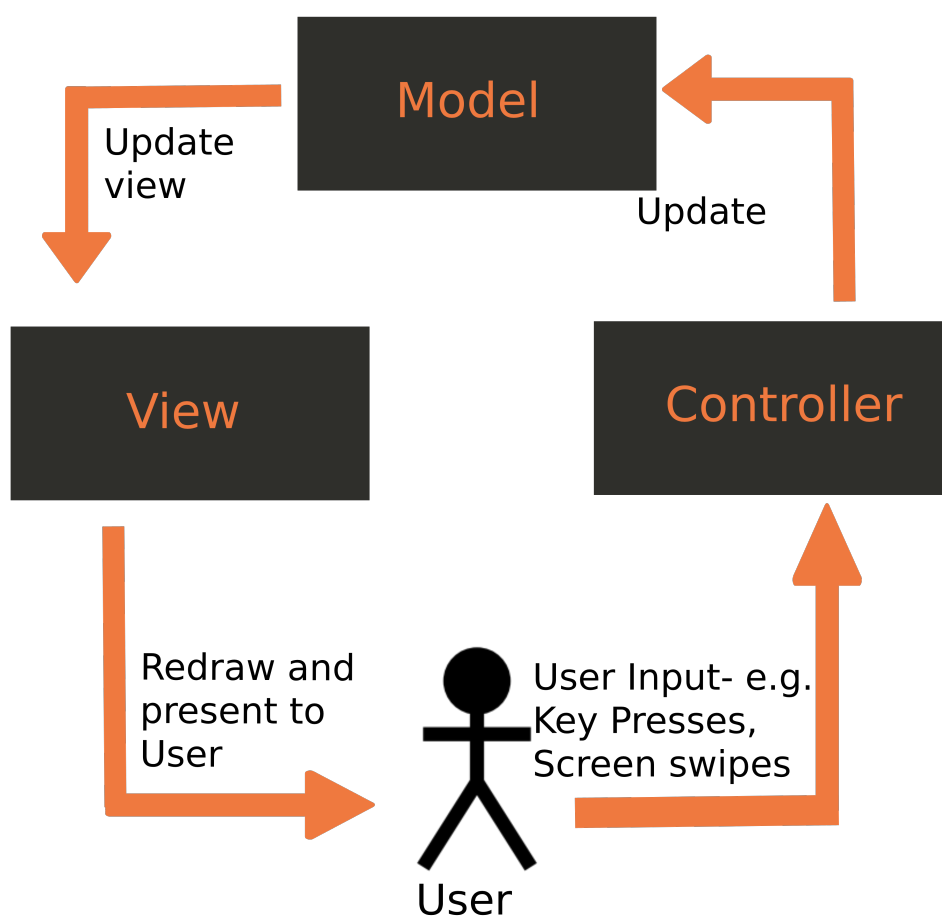
#### 9.1 Model View Controller

Přístup Model View Controller odděluje aplikaci do tří komponent. Každá komponenta je pak odpovědná za určitou část programu.

**Model** řeší přístup k datům a zároveň business logiku aplikace. Je to v podstatě mozek celé aplikace. Není vázán na View ani Controller a díky tomu je pak v mnoha případech přepoužitelný.

**View** je reprezentací modelu. Jeho zodpovědností je vykreslování uživatelského rozhraní a komunikace s controllerem v případě interakce uživatele. V architektuře MVC view neobsahuje logiku k reagování na uživatelskou interakci, tedy například kliknutí na tlačítko či zadání hodnoty. To umožňuje volnější spojení s modelem a controllerem a větší flexibilitu této komponenty.

**Controller** je komponenta, která spojuje aplikaci dohromady. Pokud uživatel klikne na tlačítko, controller rozhodne jak upravit model. Dále pak na základě změn v modelu může controller aktualizovat stav view. V případě Android aplikace je controller téměř vždy představován aktivitou nebo fragmentem[28].



Obrázek 9.1: MVC architektura, zdroj:[29]

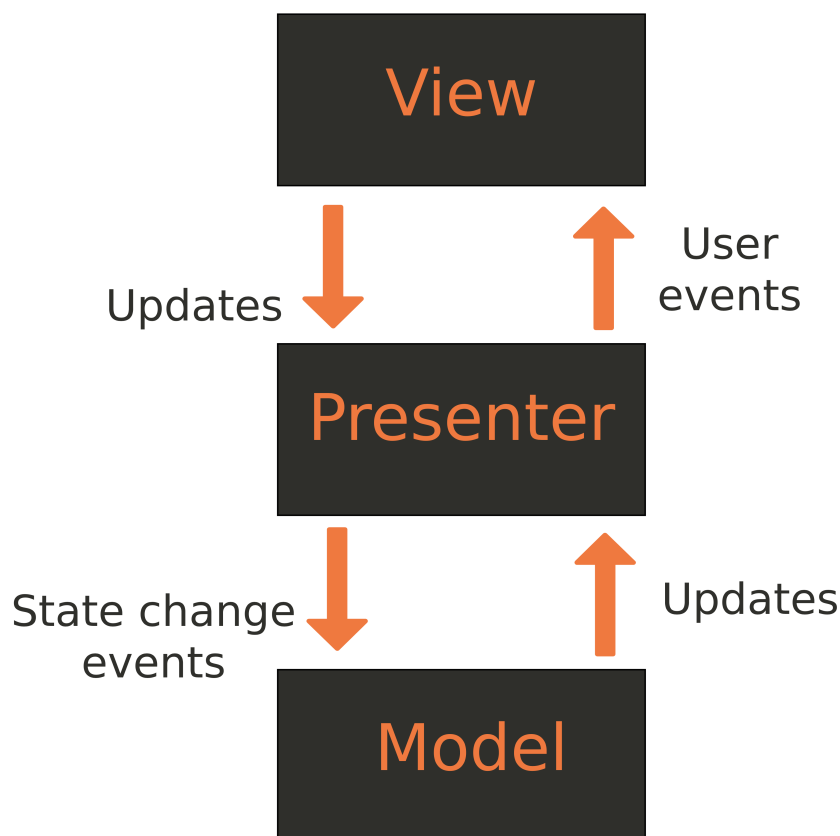
## 9.2 Model View Presenter

Model View Presenter dělí controller tak, aby přiřazená view/activity vazba mohla vzniknout i bez vazby na ostatní zodpovědnosti controlleru.

**Model** má stejnou funkci jako u MVC.

**View** má podobnou funkci jako u MVC. Jedinou změnou zde je, že aktivity/fragment jsou nyní považovány za součást view, což je celkem přirozeným krokem.

**Presenter** je v podstatě controller z MVC kromě toho, že není vůbec vázán na View, pouze implementuje jeho rozhraní. To řeší jak problémy testovatelnosti, tak problémy s modularitou, které nastávají u MVC[28].



Obrázek 9.2: MVP architektura, zdroj:[30]

## 9.3 Model View ViewModel

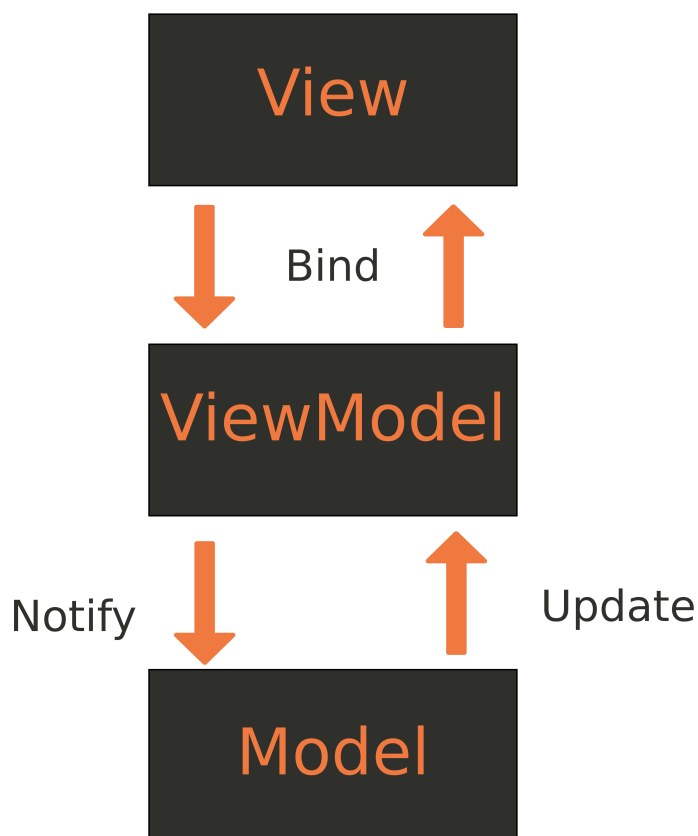
Model View ViewModel v kombinaci s Data Binding knihovnou <sup>1</sup> přináší výhody snazšího testování a modularity. Zároveň snižuje množství kódu, který je potřeba ke spojení view a modelu.

**Model** má stejnou funkci jako u MVC a MVP.

**View** se flexibilně váže na pozorovatelné proměnné poskytované ViewModelem.

**ViewModel** je zodpovědný za zabalení modelu a přípravu pozorovatelných dat potřebných pro zobrazení. Dále také umožňuje view předávat události dále do modelu, který tyto události poté zpracovává. ViewModel by však neměl být na view vázaný[28].

<sup>1</sup>Data Binding knihovna umožňuje svázat komponenty uživatelského rozhraní se zdroji dat v aplikaci [31].



Obrázek 9.3: MVVM architektura, zdroj:[32]

## 9.4 Volba architektury

Jak architektura MVP, tak i architektura MVVM, je vhodná pro vývoj na platformě Android. MVVM se však zdá být modernější a lépe testovatelný. Architektura MVVM je rovněž v oficiální dokumentaci doporučenou architekturou pro tvorbu Android aplikací[33]. Proto jsem se rozhodl jít při vývoji aplikace cestou architektury Model View ViewModel.



## Kapitola 10

### Struktura databáze

V kapitole 7 je uvedeno, že při implementaci bude použita služba Realtime Database. Tato služba poskytuje NoSQL databázi.

NoSQL databáze jsou navrženy tak, aby se odlišily od typického modelu řádků a sloupců, který používají relační databáze. Je běžnou chybou si myslet, že NoSQL databáze nemají žádný datový model. Naopak u NoSQL databází je popis struktury důležitý, aby nedocházelo k cyklickým závislostem mezi datovými objekty.

#### 10.1 Mitigace cyklických závislostí

Při používání NoSQL databází je potřeba mít neustále na paměti, že v aplikaci pracujeme s objekty, které na sebe mají určité závislosti. Například každý uživatel se může účastnit několika výzev. Objekt uživatele v sobě tedy obsahuje i seznam výzev. Zároveň ale nesmíme zapomenout na to, že každý objekt výzvy v sobě rovněž obsahuje seznam účastníků, tedy seznam uživatelů. V relačním světě by toto nebyl problém. Avšak v NoSQL světě hrozí v případě špatné implementace nekonečný cyklus závislostí „uživatel-výzva“. Proto je potřeba neuchovávat v databázi celý objekt výzvy u objektu uživatele, ale pouze nějakou referenci, podle které je pak možné výzvu v databázi dohledat. Tato reference může být například id výzvy.

Dobrým přístupem může být i to, že si v databázi mimo id uchováme i informaci, se kterou chceme dále pracovat. Například pokud bychom chtěli zobrazit uživateli seznam jeho výzev, uchováme si společně s id výzvy i její jméno. Pak není nutné si ke každé výzvě dohledávat data, protože je možné si jméno výzvy přecíst rovnou z databáze. Tento přístup sice vede k duplicitě dat, avšak u neměnných atributů objektu nám může ušetřit spoustu času při načítání dat z databáze.

#### 10.2 Zanořování dat

Jelikož Firebase Realtime Database umožňuje zanoření dat až do hloubky 32 úrovní, můžete být v pokušení zanořovat více datových struktur do sebe. Když však načtete data v umístění ve vaší databázi, načtete také všechny jeho

podřízené uzly[34]. V praxi je proto nejlepší udržovat datovou strukturu co nejvíce plochou, aby nedocházelo ke zbytečnému stahování nepotřebných dat. To by totiž vytěžovalo nejen serverovou strukturu, což by způsobovalo vyšší poplatky, ale také klientské datové tarify. Zároveň plochá datová struktura rovněž šetří výpočetní výkon zařízení, jelikož zařízení nemusí zpracovávat větší množství dat. To má za následek mimo jiné i menší spotřebu baterie.

# Kapitola 11

## Testování

Testování softwaru je nedílnou součástí vývojového cyklu. Zajišťuje kvalitní software bez velkého množství chyb a nepřesností. Testy můžeme rozdělit mnoha způsoby. Automatizované či manuální, jednotkové či integrační nebo například lokální či instrumentační. V této kapitole je stručně popsáno dělení do jednotlivých kategorií a zároveň je zde k nalezení několik navržených testovacích scénářů pro testování finální aplikace.

### 11.1 Dělení testů

Jak je již zmíněno v úvodu této kapitoly, testy lze dělit do různých kategorií. V této podkapitole jsou některé z těchto kategorií popsány. Stejně tak jako u většiny dělení, existuje mnohem více možností, jak testy rozdělit. Popis všech možných dělení je nad rámec této práce.

#### 11.1.1 Automatizované vs manuální

Hlavním rozdílem mezi těmito typy testů je vykonavatel. Zatímco u automatizovaných testů je vykonavatelem testu počítač, u manuálních testů musí testy provádět člověk. Stejně pravidlo platí i u vyhodnocení výsledku testu. Manuální testy vyhodnocuje člověk, automatizované počítač. Dalšími rozdíly pak jsou například trvání testu, náklady na tvorbu testu nebo náklady na provedení. Pokud bychom porovnali tyto atributy:

- Trvání testu je typicky kratší u automatizovaných testů.
- Náklady na tvorbu testu jsou z pravidla vyšší u automatizovaných testů.
- Náklady na provedení testu jsou vyšší u manuálních testů.

Obecně platí, že není dobré pokrývat software pouze automatizovanými nebo pouze manuálními testy. Nejlepšího výsledku dosáhneme, pokud tyto dvě varianty vhodně zkombinujeme.

### 11.1.2 Jednotkové vs integrační

Jednotkové testování neboli tzv. „Unit testování“ je metoda při které se testují jednotky zdrojového kódu. Obvykle se jedná o samostatné funkce, které dostanou testovací parametr a očekává se od nich správný parametr na výstupu.

Integrační testování kontroluje integraci mezi moduly softwaru. Integrační test tedy pak nespouští pouze samostatnou komponentu, jako tomu je u jednotkového testu, ale obvykle spouští více komponent najednou. Jako příklad můžeme uvést test integrace s databází. Test potřebuje pro svůj průběh nejen komponentu zajišťující komunikaci s databází, ale i samotnou databázi. Z těchto informací lze snadno vyvodit, že pokud mluvíme o jednotkovém a integračním testováním, mluvíme rovněž o kategorii automatizovaných testů. Pro tvorbu jednotkových a integračních testů se na platformě Android nejčastěji používá knihovna JUnit v kombinaci s frameworkem Roboelectric.

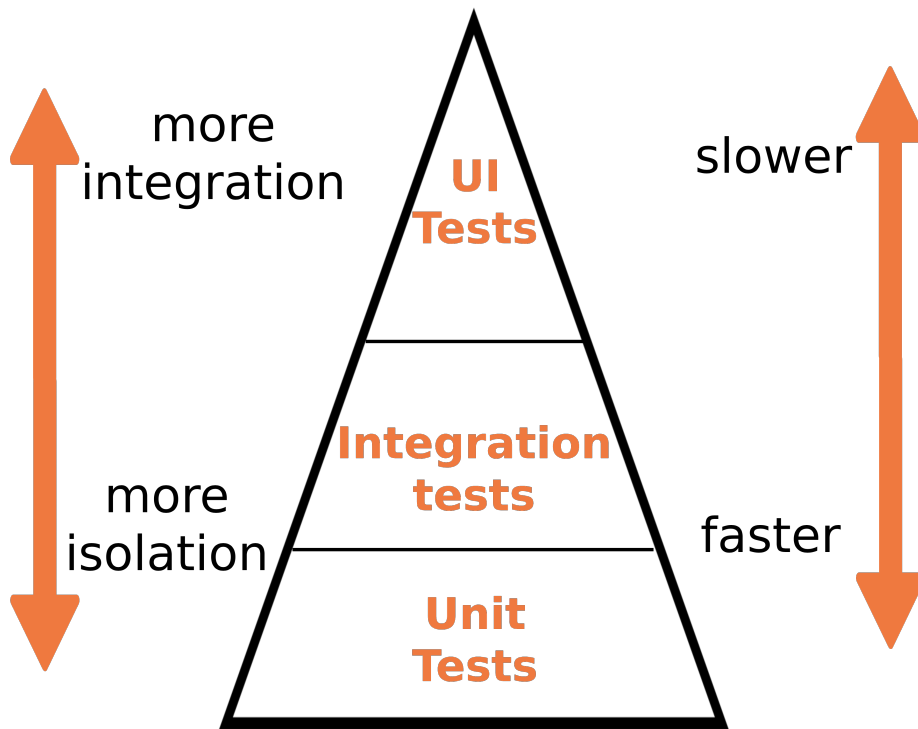
Třetím typem testů, který je vhodné zde zmínit jsou testy uživatelského rozhraní, neboli „UI testy“. UI testy ověřují, zda uživatelské rozhraní aplikace funguje tak, jak je očekáváno. Můžeme zde testovat například to, že uživatelský vstup způsobí správnou změnu do uživatelského rozhraní, viditelnost prvků uživatelského rozhraní a podobně. Testy uživatelského rozhraní se na platformě Android tvoří nejčastěji pomocí frameworků<sup>1</sup> UI Automator a Espresso.

Dohromady tyto tři typy testů tvoří tzv. testovací pyramidu. S konceptem testovací pyramidy poprvé přišel Mike Cohn v jeho knize „Succeeding with Agile“. Testovací pyramida je vizuální metafora, která radí vývojářům aby přemýšleli o různých vrstvách testování. Rovněž říká, kolik testů je třeba provést v každé vrstvě pyramidy[36].

Ve spodní vrstvě testovací pyramidy se nacházejí jednotkové testy. Z toho vyplývá, že se jedná o nejpočetnější část testů. Dalším prvkem pyramidy jsou integrační testy a na vrcholu se nachází testy uživatelského rozhraní. Pro pyramidu platí, že čím hlouběji jsme, tím jsou testy rychlejší více izolované od ostatních komponent. Naopak čím výše jsme, tím jsou testy pomalejší a vyžadují integraci více komponent. Tedy jednotkové testy by měly být nejvíce izolované a nejrychlejší, zatímco testy uživatelského rozhraní jsou typicky nejpomalejší a testují integraci většiny komponent aplikace[37].

---

<sup>1</sup>Framework je platforma pro vývoj softwarových aplikací. Poskytuje základ, na kterém mohou vývojáři softwaru vytvářet programy pro konkrétní platformu. Framework může například zahrnovat předdefinované třídy a funkce, které lze použít ke zpracování vstupu, správě hardwarových zařízení a interakci se systémovým softwarem[35].



Obrázek 11.1: Testovací pyramida, zdroj:[36]

### 11.1.3 Lokální vs instrumentační

Dalším a zároveň posledním představeným dělením jsou lokální a instrumentační testy.

Lokální testy jsou spouštěny pouze na místním počítači a jsou kompilovány tak, aby mohli proběhnout na lokální JavaVirtualMachine (JVM). To přináší minimální čas provádění testů, jelikož testy nepřistupují přímo k funkcím Androidu. Pokud testy potřebují použít některou z funkcí Androidu, používá se např. framework Roboelectric k předdefinování výstupů systému a jejich následné použití[38].

Instrumentační testy běží na fyzickém zařízení s operačním systémem Android, nebo na emulátoru s tímto operačním systémem. Tyto testy mají přístup k instrumentačním informacím zařízení, což může být například tzv. „context“<sup>2</sup> aplikace. Tento přístup se používá pro testy, které mají složitější závislosti na funkce systému Android a bylo by obtížné tyto závislosti předem nastavit[38].

## 11.2 Testovací scénáře

V této podkapitole jsou uvedeny testovací scénáře průchodem aplikací, které pokrývají základní funkcionality aplikace.

### 11.2.1 TC1 - Registrace

**Průchod scénářem:**

1. Uživatel klikne na tlačítko „Register“.
2. Uživatel vyplní emailovou adresu, dvakrát zadá heslo a klikne na tlačítko „Continue“.
3. Uživatel zadá jméno, věk, vybere profilový obrázek a klikne na tlačítko „Finish“.

Testovací data: Nový uživatel, existující uživatel, neplatný email, rozdílná hesla, prázdné hodnoty.

### 11.2.2 TC2 - Přihlášení

**Průchod scénářem:**

1. Uživatel zadá email, heslo a klikne na tlačítko „Login“.

Testovací data: Neexistující uživatel, existující uživatel, prázdné hodnoty.

**Alternativní průchod:**

---

<sup>2</sup>Context je rozhraní s globálními informacemi o prostředí aplikace. Jedná se o abstraktní třídu, jejíž implementaci zajišťuje systém Android. Umožňuje přístup k prostředkům a třídám specifických pro aplikaci. Zároveň také zajišťuje volání pro operace jako je spuštění aktivity, přijímání externích informací nebo například zobrazování notifikací[39].

1. Uživatel klikne na tlačítko „Sign in with Google“ a postupuje podle pokynů na obrazovce.

Testovací data: Neexistující uživatel, existující uživatel.

### ■ 11.2.3 TC3 - Vytvoření výzvy

**Průchod scénářem:**

1. Uživatel přejde na záložku „Explore“.
2. Uživatel klikne na FAB tlačítko s nápisem „+“.
3. Uživatel zadá potřebné hodnoty, vybere obrázek výzvy a klikne na tlačítko „Create“.

Testovací data: Všechny údaje vyplněné, prázdné hodnoty, existující výzva.

### ■ 11.2.4 TC4 - Připojení se k výzvě

**Průchod scénářem:**

1. Uživatel přejde na záložku „Explore“.
2. Uživatel si vybere výzvu v seznamu a klikne na ní.
3. Uživatel klikne na tlačítko „Join challenge“.

Testovací data: Výzva, které se uživatel ještě neúčastní, výzva, které se uživatel již účastní.

### ■ 11.2.5 TC5 - Přidání pokroku do výzvy

**Průchod scénářem:**

1. Uživatel přejde na záložku „Dashboard“.
2. Uživatel vybere výzvu ze seznamu aktivních výzev.
3. Uživatel klikne na FAB tlačítko s nápisem „+“.
4. Uživatel zadá hodnotu pokroku.

Testovací data: Aktivní výzva, nadcházející výzva, číselná hodnota, textová hodnota.

**Alternativní průchod:**

1. Uživatel přejde na záložku „Explore“.
2. Uživatel vybere výzvu ze seznamu výzvy, které se účastní.
3. Uživatel klikne na FAB tlačítko s nápisem „+“.
4. Uživatel zadá hodnotu pokroku.

Testovací data: Aktivní výzva, nadcházející výzva, výzva, které se uživatel neúčastní, číselná hodnota, textová hodnota.

### ■ 11.2.6 TC6 - Zobrazení pokroku ve výzvě

#### Průchod scénářem:

1. Uživatel přejde na záložku „Dashboard“.
2. Uživatel vybere výzvu ze seznamu aktivních výzev.
3. Uživatel si zobrazí svůj pokrok ve výzvě.

Testovací data: Aktivní výzva, nadcházející výzva, výzva, které se uživatel neúčastní.

### ■ 11.2.7 TC7 - Zobrazení aktivních výzev

#### Průchod scénářem:

1. Uživatel přejde na záložku „Dashboard“.

Testovací data: Žádná aktivní výzva, více aktivních výzev.

### ■ 11.2.8 TC8 - Zobrazení splněných výzev

#### Průchod scénářem:

1. Uživatel přejde na záložku „Profile“.
2. Uživatel si zobrazí odznaky za dokončené výzvy.

Testovací data: Žádné dokončené výzvy, více dokončených výzev.

### ■ 11.2.9 TC9 - Vyhledání uživatele

#### Průchod scénářem:

1. Uživatel přejde do záložky „Friends“.
2. Uživatel napíše přítelův email do prvku pro vyhledávání.

Testovací data: Existující uživatel, neexistující uživatel.

### ■ 11.2.10 TC10 - Přidání přítele

#### Průchod scénářem:

1. Uživatel přejde do záložky „Friends“.
2. Uživatel napíše přítelův email do prvku pro vyhledávání.
3. Uživatel klikne na tlačítko „Add as friend“.

Testovací data: Existující uživatel, neexistující uživatel, uživatel již přidáný do přátel, uživatel nepřidáný do přátel.



### ■ 11.2.11 TC11 - Zobrazení aktivity přátel

#### Průchod scénářem:

1. Uživatel přejde do záložky „Friends“.

Testovací data: Uživatel bez přátel, uživatel s přáteli.

## ■ 11.3 Vyhodnocení testů

Testovací scénáře jsem procházel průběžně poslední dva týdny vývoje aplikace. Narazil jsem díky tomu na mnoho chyb, které se mi podařilo opravit. Nejvíce chyb bylo v ošetření vstupů, například ošetření prázdných hodnot u vstupních textových polí ze scénářů 1, 2 a 3. Dále jsem občas narazil na chyby typu pomalého načítání dat, obzvláště multimediálního obsahu. Ty byly způsobené používáním cloudového úložiště a nepoužitím technologie LiveData. Odhalené chyby jsem odstranil právě použitím technologie LiveData.



# Kapitola 12

## Prostor pro vylepšení

Tato kapitola se zabývá hodnocením vzniklé aplikace a možnými budoucími vylepšeními.

### 12.1 Zhodnocení aplikace

Vzniklá aplikace podporuje vytváření a následnou práci s výzvami, což byl hlavní záměr aplikace. V analýze jsou uvedené různé typy výzev. Bohužel různé typy výzev vyžadují komplexnější logiku vyhodnocování, a proto jsou z původních čtyř typů výzev implementovány pouze typy tři. Typ „BEST\_PERFORMANCE“ zůstal nenaimplementovaný. I přesto hodnotím aplikaci jako přínosnou, vzhledem k tomu, že v momentálně dostupných aplikacích není volení typu výzvy možné. Černou tečkou u výzev je možnost zakládání pouze veřejných výzev a nemožnost vytváření výzev soukromých. To zůstalo rovněž bez implementace. Kladně hodnotím možnost zvolení předmětu sázky, což byla jedna z věcí, které ostatní aplikace neumožňují.

Další funkcionalitou popsanou v analýze je přidávání fotografií a videí při zadávání pokroku ve výzvě, a jejich následné zobrazení, popřípadě zhodnocení přáteli. Zde se povedlo implementovat velkou část z této funkcionality. Aplikace podporuje přidávání multimédií při zadávání pokroku ve výzvě. Rovněž je možné kladně či záporně zareagovat na aktivitu přátel. Zde však chybí výčet uživatelů, kteří reagovali. Je tedy možné zobrazit počet kladných a záporných hodnocení, avšak není možné zjistit, kdo toto hodnocení přidal. Nedostatkem zde je rovněž přidávání komentářů k aktivitě přátel, což aplikace neumožňuje.

Interakci mezi uživateli rovněž hodnotím kladně. Uživatelé se mohou navzájem přidávat do přátel, sledovat svůj pokrok ve výzvách a zároveň si jednotlivé aktivity hodnotit. Systém pozvánek do výzev je ovšem stále nedopracovaný. Aplikace momentálně umožňuje uživateli upozornit přítele na výzvu, ale přítel se poté musí přidat k výzvě běžným způsobem. To znemožňuje zaslat notifikaci uživateli, který přítele pozval.

Z mého pohledu největším nedostatkem je funkcionalita okolo týmů. V analýze je popsána možnost zakládání a přidávání uživatelů týmů, stejně tak jako interní týmové výzvy a mezitýmové výzvy. Ani jedna z těchto funkcionalit není v aplikaci z časových důvodů implementována.

Ani architektura aplikace nezůstala bez chyby. Zde se podepsala nezkušenost s návrhem architektury pro mobilní aplikace. Jako hlavní problém architektury místy nadměrné množství zpětných volání v kódu (tzv. callbacků), což místy způsobuje nečitelnost kódu. Naopak si myslím, že se dobře podařilo oddělit jednotlivé vrstvy aplikace, tedy model, view a viewmodel. Celkově architekturu hodnotím jako průměrnou.

Implementaci aplikace hodnotím kladně, aplikace je funkční a splňuje účel, za kterým byla vytvořena, tedy umožnit uživatelům tvorbu přizpůsobitelných výzev. Jako každá aplikace, i tato má své nedostatky. I přes to si myslím, že by mohla být dobrým pomocníkem pro motivaci přátel a známých. Nedostatky aplikace jsou způsobeny převážně neznalostí technologie a časovou dotací.

Implementované a neimplementované funkcionality jsou pro přehlednost shrnuty v následující tabulce.

Funkcionalita	V plánu pro implementaci	Naimplementováno
Přihlášení přes Google nebo Facebook	Ano	Ano
Založení účtu v aplikaci	Ano	Ano
Feed s novinkami	Ano	Ano
Získávání odznaků za splnění výzvy	Ano	Ano
Zakládání vlastních výzev	Ano	Ano
Sdílení výzev se všemi uživateli aplikace	Ano	Ano
Přidávání a hodnocení uživatelského obsahu	Ano	Ano
Přidávání pokroku ve výzvě	Ano	Ano
Možnost zvolit parametry výzvy	Ano	Ano
Možnost zvolit předmět sázky	Ano	Ano
Možnost zvolit trvání výzvy	Ano	Ano
Vyhodnocení výzvy	Ano	Ano
Zobrazení aktivních výzev	Ano	Ano
Zobrazení historie výzev	Ano	Ano
Zobrazení žebříčku výzvy	Ano	Ano
Zobrazení nadcházejících výzev	Ne	Ano
Přidávání uživatelů do přátel	Ne	Ano
Komentování v průběhu výzvy	Ano	Ne
Prohlížení výzev dle kategorií	Ano	Ne
Notifikace o nesplněním úkolu	Ano	Ne
Sdílení výzev pouze mezi přáteli	Ano	Ne
Připojení se do týmu	Ne	Ne
Funkcionality v rámci týmu	Ne	Ne

**Tabulka 12.1:** Implementované funkcionality aplikace

## ■ 12.2 Možná budoucí vylepšení

Vylepšení aplikace se nabízí opravdu mnoho. Rád bych zde vypíchnul doimplementování funkcionality okolo týmů a lepší možnost přizpůsobení výzev. To by podle mého názoru přilákalo velké množství uživatelů. Zároveň by bylo přínosné aplikaci vydat do obchodu Google Play, aby byla přístupná všem uživatelům operačního systému Android. Tím by se aplikace mohla distribuovat mezi známými a stala by se zábavnější pro všechny, jelikož aplikace nebude přinášet správný požitek, pokud jí bude používat jednotlivec bez skupiny přátel. Dalším velkým přínosem pro aplikaci by pak byla verze pro operační systém iOS. Osobně plánuji aplikaci více odladit a do obchodu Google Play vydat.



# Kapitola 13

## Závěr

Cílem práce bylo provést analýzu, navrhnout, naimplementovat a nakonec i otestovat mobilní aplikaci pro sdílení výzev. Jinými slovy, projít si vývojovým cyklem aplikace od začátku až do konce. V práci je celý cyklus popsán.

V úvodních kapitolách práce analyzuje závěry sociologické studie ohledně faktorů ovlivňujících soutěživost lidí, což slouží jako odrazový můstek pro definování nové aplikace. Dále je začátek práce věnován analýze již existujících aplikací, které mají systém výzev implementovaný. Analýza těchto aplikací se pak rovněž posloužila jako zdroj informací pro návrh nové aplikace, která se soustředí na systém výzev více do hloubky. Práce dále obsahuje i business canvas, kde nalezneme několik aktivit, které mohou pomoci se zvýšením počtu uživatelů a případného zpeněžení aplikace.

Po stanovení konceptu aplikace byl vytvořen low-fidelity prototyp pro úvodní uživatelské testování. Zároveň byly vydefinovány skupiny typických uživatelů aplikace, aby bylo možné vybírat respondenty patřící do cílové skupiny uživatelů aplikace. Výstupy uživatelského testování jsou následně použity k upřesnění dalšího směřování aplikace. Přínosným zjištěním byl i fakt, že o aplikaci tohoto typu byl mezi respondenty zájem.

Dalším obsahem práce je technický popis řešení. V této části nechybí typické diagramy využívající se v softwarovém inženýrství, popis použitých technologií, výběr architektury a další. Na základě této části je tedy posléze psán zdrojový kód tvořené aplikace.

V kapitole „Testování“ se práce věnuje možnostem testování aplikace a návrhu základních testovacích scénářů. Je zde popsán i průběh testování vzniklé aplikace a vyhodnocení testovacích scénářů.

V rámci práce byla rovněž implementována funkční aplikace pro platformu Android. Implementace má bezpochyby své nedostatky a umožňuje mnoho budoucích vylepšení. Největší výzvou při tvorbě byl i přes dobrou dokumentaci služeb Firebase problém synchronizace dat mezi vzdáleným serverem a klientskou aplikací. Zde se projevila nezkušenost s tvorbou aplikací na platformě Android. Přesto však aplikaci považuji za přínosnou a splňující zadání. Užitečnou zkušeností pro mě je průchod celým vývojovým cyklem, stejně tak jako seznámení se s novými technologiemi. Největší výzvou při tvorbě byl i přes dobrou dokumentaci služeb Firebase problém synchronizace dat mezi vzdáleným serverem a klientskou aplikací. Zde se projevila nezkušenost

s tvorbou aplikací na platformě Android, avšak i tento problém byl nakonec zdárně vyřešený. Z mého pohledu tedy projekt hodnotím jako úspěšný a věřím, že tvorbou nabyté zkušenosti mi zůstanou už navždy.





## Literatura

- [1] Van der Linden, S., Ph.D. The Psychology of Competition, [cit. 2020.10.27]. Dostupné z: <https://www.psychologytoday.com/us/blog/socially-relevant/201506/the-psychology-competition>
- [2] Stephen M. Garcia, Avishalom Tor & Tyrone M. Schiff. The Psychology of Competition: A Social Comparison Perspective, [cit. 2020.11.02]. Dostupné z: [https://scholarship.law.nd.edu/law\\_faculty\\_scholarship/941](https://scholarship.law.nd.edu/law_faculty_scholarship/941)
- [3] How Long Should Your Next Video Be? - Wistia Blog, [cit. 2020.11.07]. Dostupné z: <https://wistia.com/learn/marketing/optimal-video-length>
- [4] Strava - Obchod Google Play, [cit. 2020.11.15]. Dostupné z: <https://play.google.com/store/apps/details?id=com.strava>
- [5] Garmin Connect - Obchod Google Play, [cit. 2020.11.15]. Dostupné z: <https://play.google.com/store/apps/details?id=com.garmin.android.apps.connectmobile>
- [6] CHLNGE - Obchod Google Play, [cit. 2020.11.15]. Dostupné z: <https://play.google.com/store/apps/details?id=de.chlnge.app>
- [7] Habit Challenge - Obchod Google Play, [cit. 2020.11.15]. Dostupné z: <https://play.google.com/store/apps/details?id=org.luksza.HabitChallenge>
- [8] What is a user persona? User persona definition, benefits and examples, [cit. 2021.02.13]. Dostupné z: <https://airfocus.com/glossary/what-is-a-user-persona/>
- [9] Male profile avatar with brown hair vector image on vectorstock, [cit. 2021.05.05]. Dostupné z: <https://www.vectorstock.com/royalty-free-vector/male-profile-avatar-with-brown-hair-vector-12055105>
- [10] Female profile avatar image on pngkit, [cit. 2021.05.05]. Dostupné z: [https://www.pngkit.com/png/full/115-1150121\\_this-free-icons-png-design-of-female-avatar.png](https://www.pngkit.com/png/full/115-1150121_this-free-icons-png-design-of-female-avatar.png)

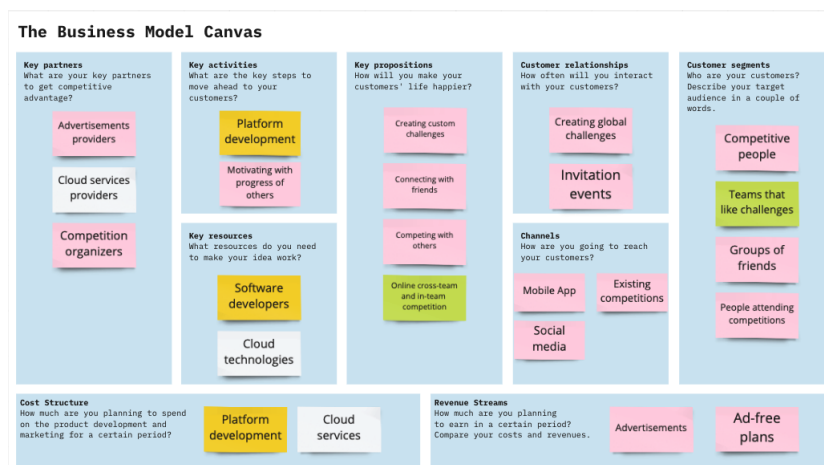
- [11] Male Avatar PNG icons, [cit. 2021.05.05]. Dostupné z: <https://www.iconspng.com/image/93970/male-avatar>
- [12] Griffith, C. Ionic article: What is hybrid app development?, [cit. 2021.02.15]. Dostupné z: <https://ionic.io/resources/articles/what-is-hybrid-app-development>
- [13] Detailed market size and share trends empower companies selling mobile phones to get ahead of market changes and compete more effectively, [cit. 2021.05.05]. Dostupné z: <https://www.idc.com/promo/smartphone-market-share/os>
- [14] Kotlin + Gradle: a technology combination sure to foster developer happiness and productivity, [cit. 2021.05.18]. Dostupné z: <https://gradle.org/kotlin/>
- [15] What is Android SDK? - definition from Techopedia, [cit. 2021.01.12]. Dostupné z: <https://www.techopedia.com/definition/4220/android-sdk>
- [16] Licenses and standards, [cit. 2021.04.23]. Dostupné z: <https://opensource.org/licenses>
- [17] Tyson, W., & Tyson, M. What is the Jvm? Introducing the Java virtual machine, [cit. 2021.04.29]. Dostupné z: <https://www.infoworld.com/article/3272244/what-is-the-jvm-introducing-the-java-virtual-machine.html>
- [18] What is Java, [cit. 2021.04.29]. Dostupné z: [https://java.com/en/download/help/whatis\\_java.html](https://java.com/en/download/help/whatis_java.html)
- [19] Stdlib - Kotlin programming language, [cit. 2021.01.23]. Dostupné z: from <https://kotlinlang.org/api/latest/jvm/stdlib/>
- [20] Null safety: KOTLIN, [cit. 2021.04.24]. Dostupné z: <https://kotlinlang.org/docs/null-safety.html#nullable-types-and-non-null-types>
- [21] Gradle, [cit. 2021.01.03]. Dostupné z: [https://docs.gradle.org/current/userguide/what\\_is\\_gradle.html](https://docs.gradle.org/current/userguide/what_is_gradle.html)
- [22] Serverless architectures, [cit. 2021.04.24]. Dostupné z: <https://martinfowler.com/articles/serverless.html>
- [23] What is nosql? Nosql databases explained, [cit. 2021.04.24]. Dostupné z: <https://www.mongodb.com/nosql-explained>
- [24] Netanel, N. (2018). Copyright: What everyone needs to know, [cit. 2021.04.24]. Dostupné z: <https://aws.amazon.com/what-is-cloud-storage/>

- [25] Firebase realtime database, [cit. 2021.05.05]. Dostupné z: <https://firebase.google.com/docs/database>
- [26] Firebase authentication, [cit. 2021.04.24]. Dostupné z: <https://firebase.google.com/docs/auth>
- [27] Cloud storage for Firebase, [cit. 2021.04.24]. Dostupné z: <https://firebase.google.com/docs/storage>
- [28] Maxwell, E. MVC vs. MVP VS. MVVM on Android, [cit. 2021.04.24]. Dostupné z: <https://academy.realm.io/posts/eric-maxwell-mvc-mvp-and-mvvm-on-android/>
- [29] A typical Model-View-Controller software pattern, [cit. 2021.04.24]. Dostupné z: [https://www.researchgate.net/figure/A-typical-Model-View-Controller-MVC-software-pattern\\_fig4\\_316916116](https://www.researchgate.net/figure/A-typical-Model-View-Controller-MVC-software-pattern_fig4_316916116)
- [30] Model View Presenter, [cit. 2021.04.24]. Dostupné z: <http://aalmiray.github.io/griffon-patterns/images/mvp.png>
- [31] Data binding Library : Android developers, [cit. 2021.04.24]. Dostupné z: <https://developer.android.com/topic/libraries/data-binding>
- [32] Model View View Model, [cit. 2021.04.24]. Dostupné z: [https://cdn-images-1.medium.com/max/1600/1\\*8MiNUZRqM1XDtjtifxTSqA.png](https://cdn-images-1.medium.com/max/1600/1*8MiNUZRqM1XDtjtifxTSqA.png)
- [33] Guide to app architecture: Android Developers guide, [cit. 2021.05.18]. Dostupné z: <https://developer.android.com/jetpack/guide#recommended-app-arch>
- [34] Structure your database | firebase realtime database, [cit. 2021.04.24]. Dostupné z: <https://firebase.google.com/docs/database/android/structure-data>
- [35] Framework, [cit. 2021.04.29]. Dostupné z: <https://techterms.com/definition/framework>
- [36] Vocke, H. The practical test pyramid, [cit. 2021.04.25]. Dostupné z: <https://martinfowler.com/articles/practical-test-pyramid.html>
- [37] Cohn, M. Succeeding with agile: Software development using Scrum. Upper Saddle River, NJ etc.: Addison-Wesley.
- [38] Build effective unit tests : Android developers, [cit. 2021.04.25]. Dostupné z: <https://developer.android.com/training/testing/unit-testing>
- [39] Context : Android developers, [cit. 2021.04.25]. Dostupné z: <https://developer.android.com/reference/android/content/Context>



# Příloha A

## Business canvas

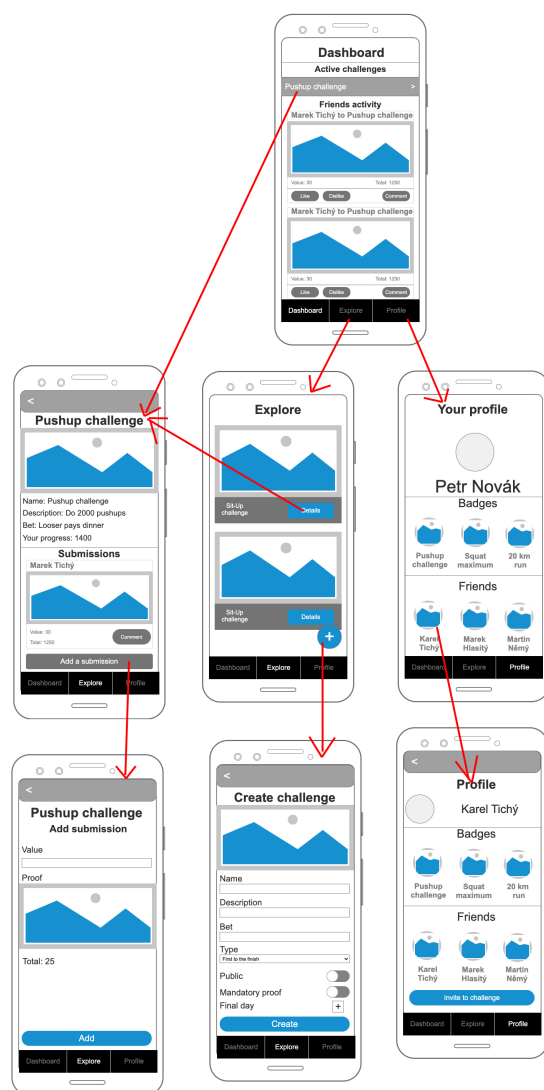


Obrázek A.1: Business canvas



# Příloha B

## Prototyp aplikace



Obrázek B.1: Prototyp aplikace používaný k uživatelskému testování







## Příloha C

### Výsledky uživatelského testování

Tabulka je dostupná na adrese:

- Dlouhý link: [https://docs.google.com/spreadsheets/d/1cqj97nW1pKgA4Tk91f\\_y4TSxGW3VCEgplBqktA1WfqU/edit?usp=sharing](https://docs.google.com/spreadsheets/d/1cqj97nW1pKgA4Tk91f_y4TSxGW3VCEgplBqktA1WfqU/edit?usp=sharing)
- Krátký link: <http://bit.ly/3rPFCUu>



## Příloha D

### Screenshots z aplikace

#### Login or register

E-Mail Address

---

Password

---

LOGIN

Don't have an account?

REGISTER



Sign in with Google

Obrázek D.1: Přihlášení

## Dashboard

### Active challenges



Summiting

33.4 %



Book reading

0.0 %


### Upcoming challenges



Cross-country skiing

  
Dashboard

  
Explore

  
Friends activity

  
My profile

Obrázek D.2: Záložka dashboard

## Explore



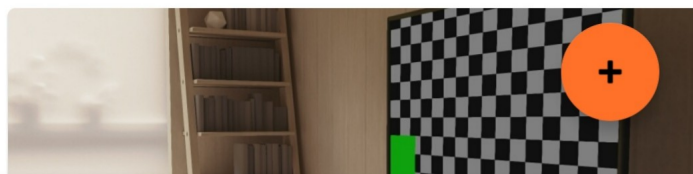
### Summiting

Summit 3 hills above 500m




### Cross-country skiing


Ride a total of 50 km on cross country skis



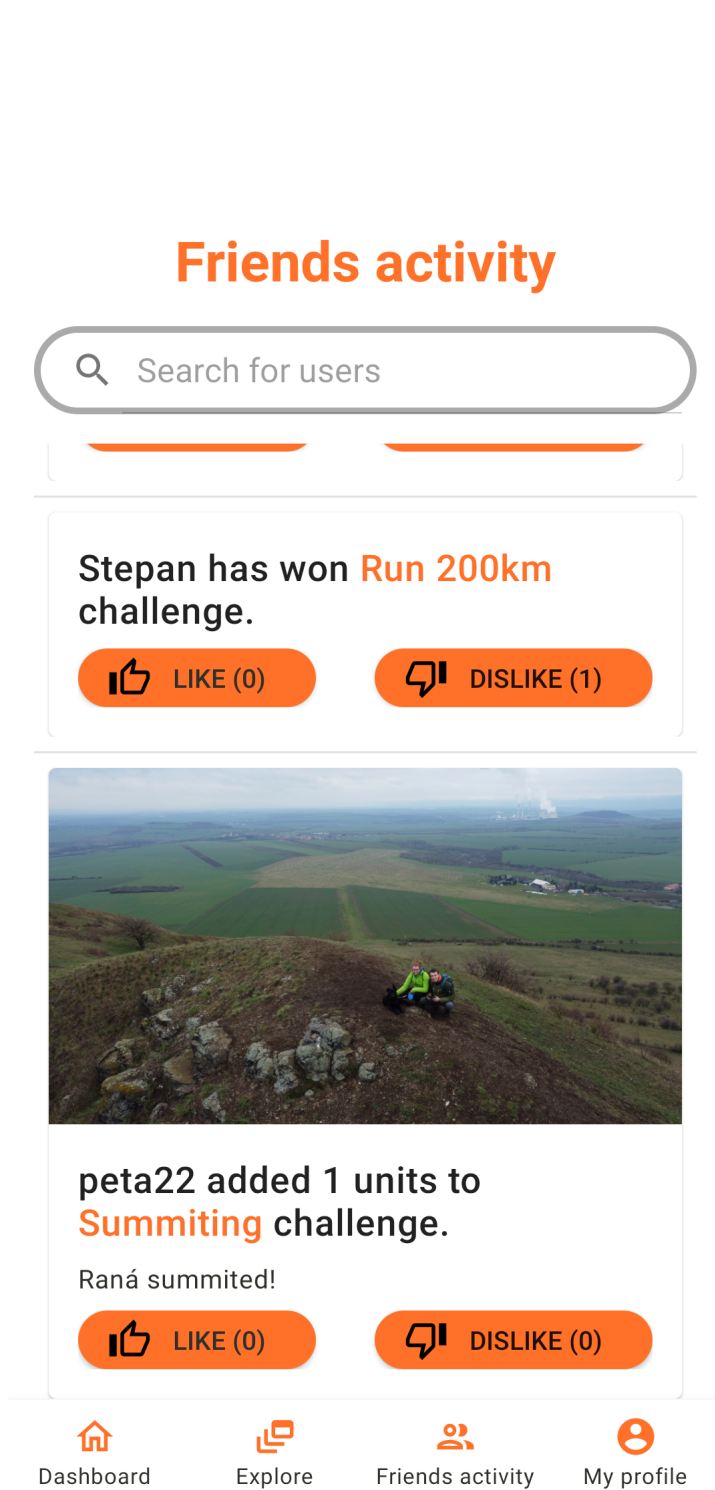
  
Dashboard

  
Explore

  
Friends activity

  
My profile

Obrázek D.3: Záložka explore





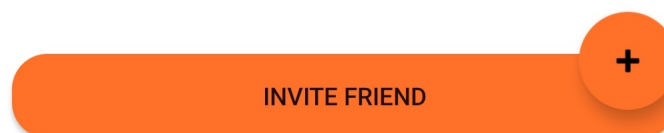
Obrázek D.4: Záložka aktivity přátel



**Name:** Summiting  
**Description:** Summit 3 hills above 500m  
**Bet:** Winner gets a free beer from every other participant  
**From:** 01/5/2021 03:51:59  
**To:** 30/6/2021 03:52:04  
**Target value:** 3  
**Challenge type:** FIRST\_TO\_THE\_FINISH

## Leaderboard

1		bendlste	2
2		peta22	1



Obrázek D.5: Detail výzvy

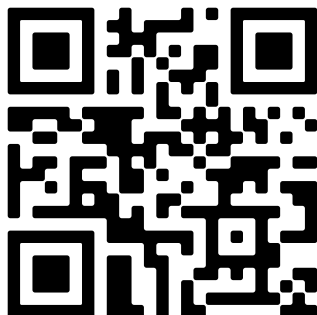




## Příloha E

### Odkaz adresáře projektu na školní GitLab

- Krátký odkaz: <https://bit.ly/2QqSW3P>
- Celý odkaz: <https://gitlab.fel.cvut.cz/bendlste/challenger>



Obrázek E.1: QR kód odkaz