



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
Katedra počítačů

Volejbalový statistický systém jako android aplikace

Android application providing volleyball statistics

Bakalářská práce

Studijní program: Otevřená informatika

Studijní obor: Softwarové systémy

Vedoucí práce: doc. Ing. David Šišlák, Ph.D.

Václav Štemberg
Praha 2018

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Štemberg** Jméno: **Václav** Osobní číslo: **421051**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Otevřená informatika**
Studijní obor: **Softwarové systémy**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Volejbalovy statisticky system jako android aplikace

Název bakalářské práce anglicky:

Android application providing volleyball statistics

Pokyny pro vypracování:

Navrhnete a vytvořte android viceuzivatelskou aplikaci realizující statisticky volejbalovy system, ve kterem budou data uložena na serveru. V systému budou jednoznačně definovány role uživateli a jejich oprávnění. Radně přihlášený uživatel získá přístup k uloženým datům a dle patřičné role má možnost tyto data upravovat. Při vytváření aplikace postupujte následujícími způsoby:

- 1) Vyhleďte a analyzujte dostupné systémy poskytující stejnou nebo obdobnou funkcionalitu.
- 2) Rozpracujte funkcionalitu vytvářeného systému a porovnejte s existujícími aplikacemi.
- 3) Seznamte se s technologiemi potřebnými pro vytvoření android aplikace komunikující se serverem prostřednictvím síťového spojení po internetu splňující bezdrátové standardy.
- 4) Navrhnete vhodné a intuitivní grafické rozhraní aplikace.
- 5) Implementujte všechny komponenty systému včetně serverové části s její databázovou strukturou.
- 6) Vyzkoušejte funkčnost aplikace na uživateli, tyto testy zdokumentujte a vyhodnotte.

Seznam doporučené literatury:

- [1] Burnette, Ed: Hello, Android: Introducing Google's Mobile Development Platform. Pragmatic Bookshelf; 4 edition, 2015.
- [2] Biehl, Matthias: RESTful API Design (API-University Series) (Volume 3). CreateSpace Independent Publishing Platform; 1 edition, 2016.
- [3] Hutten, Dennis: AWS: Amazon Web Services Tutorial The Ultimate Beginners Guide. CreateSpace Independent Publishing Platform, 2017.
- [4] Hutten, Dennis: AWS: Developers Guide to AWS Lambda The Ultimate Beginners Guide. Amazon Digital Services LLC, 2017.
- [5] Obe, Regina O. and Hsu, Leo S.: PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database. O'Reilly Media; 3 edition, 2017.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. David Šišlák, Ph.D., centrum umělé inteligence FEL

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

Datum zadání bakalářské práce: _____ Termín odevzdání bakalářské práce: _____

Platnost zadání bakalářské práce:
do konce letního semestru 2018/2019

doc. Ing. David Šišlák, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
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

Prohlášení

„Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.“

Praha, 20. května 2018

.....
podpis autora práce

Poděkování

Rád bych vyjádřil poděkování vedoucímu mé bakalářské práce panu doc. Ing. Davidu Šišlákovi, Ph.D. za profesionální přístup v rámci naší spolupráce. Dále bych chtěl poděkovat své rodině a blízkým za jejich psychickou i materiální podporu během studia.

Abstrakt

Práce je zaměřena na návrh a vývoj mobilní aplikace pro operační systém Android. Výstupem je aplikace realizující statistický volejbalový systém, ve kterém jsou data ukládána na serveru. Systém pomáhá trenérům i hráčům detekovat případné slabiny týmu či jednotlivců. Na základě zpracovaných dat lze pak provést patřičné kroky, které povedou k eliminaci těchto nedostatků. Práce popisuje již existující řešení, použité technologie pro návrh nové aplikace, její implementaci a závěrečné testování.

Klíčová slova: mobilní aplikace, Android, server, Amazon Web Services, volejbal, statistika

Abstract

This thesis focuses on the design and development of mobile application based on Android operating system. The thesis outcome is an application providing volleyball statistical system in which the data is stored on a server. This system helps coaches and players to detect team or individual weaknesses. Based on these data, appropriate steps can be taken to eliminate these shortcomings. The thesis describes existing solutions, used technologies, application design, implementation and final testing.

Keywords: mobile application, Android, server, Amazon Web Services, volleyball, statistics

Obsah

1	Úvod	17
1.1	Požadavky	17
2	Současná situace	19
2.1	Existující aplikace	19
2.1.1	DataProject	19
2.1.2	Individual Volleyball Stats.....	20
2.1.3	McStats-VBall Free	20
2.1.4	Quick Scout Volley Free.....	20
2.1.5	SoloStats 123 Volleyball.....	21
2.1.6	Volleyball Stat!	21
2.1.7	Volley Stats	21
2.2	Shrnutí	22
3	Použité technologie	23
3.1	JPA	23
3.2	REST	23
3.3	Formát JSON	23
3.4	Maven	23
3.5	Amazon Web Services	23
3.5.1	Cognito	24
3.5.2	API Gateway	24
3.5.3	Lambda.....	24
3.5.4	RDS	24
3.5.5	EC2.....	25
3.5.6	S3	25
3.5.7	VPC	25
3.5.8	Cloud Watch	25
3.5.9	IAM	25
3.6	Vývojové prostředí	25
3.6.1	IntelliJ IDEA	26
3.6.2	Android Studio	26
3.7	Git	26
4	Návrh aplikace	27
4.1	Požadavky	27
4.1.1	Funkční.....	27
4.1.2	Nefunkční.....	28

4.2	Případy užití.....	29
4.2.1	Účet	29
4.2.2	Tým.....	31
4.2.3	Člen týmu.....	32
4.2.4	Zápas.....	33
4.2.5	Statistiky.....	34
4.3	Dotazník ohledně GUI	35
4.4	Diagram aktivit	35
4.4.1	Obnova hesla.....	36
4.4.2	Vytvoření člena týmu.....	37
5	Implementace.....	39
5.1	Server.....	39
5.1.1	API.....	40
5.1.2	Správa uživatelů a autorizace	41
5.1.3	Lambda funkce.....	42
5.1.4	Databáze.....	43
5.2	Android aplikace	45
5.2.1	HTTP klient	45
5.2.2	Přihlášení, registrace.....	45
5.2.3	Základní menu	46
5.2.4	Týmy	46
5.2.5	Uživatelské role.....	47
5.2.6	Členové týmu	47
5.2.7	Propojení uživatele s členem týmu	47
5.2.8	Zápas.....	48
5.2.9	Statistiky.....	48
6	Testování.....	51
6.1	Cílová skupina.....	51
6.2	Metoda	51
6.2.1	Heuristická evaluace	51
6.2.2	Testování v simulované laboratoři.....	52
6.3	Testování	52
6.4	Nálezy.....	53
7	Závěr	55
8	Literatura	57
9	Přílohy	59

Terminologie

API	Application Programming Interface – aplikační rozhraní
AWS	Amazon Web Services – cloudová služba společnosti Amazon
Endpoint	Koncový bod, používaný zejména u serverové části API
GUI	Graphical User Interface – grafické rozhraní
Handler	Jde o obslužnou třídu použitou v serverové části aplikace, kde probíhá zpracovávání požadavků od uživatelů
Header	Hlavička, vyskytující se u HTTP požadavků a JWT tokenů
HTTP	Hyper Text Transfer Protocol – je internetový protokol pro přenos obsahu stránek
HTTPS	Hypertext Transfer Protocol Secure – zabezpečený internetový protokol HTTP
JPA	Java Persistence API – framework programovacího jazyka Java pro práci s databází
JSON	JavaScript Object Notation – formát pro reprezentaci dat
ORM	Object-relational mapping – Objektově relační mapování, používající se při návrhu databázové struktury
RDS	Relation Database Service – služba AWS pro realizaci relační databáze
REST	Representational State Transfer – architektura rozhraní API
URL	Definuje adresu daného zdroje (stránky, API, atd.) v internetu (z anglického Uniform Resource Locator)

Obrázky

Obrázek 1 - Případy užití - účet a základní funkce	30
Obrázek 2 - Případy užití - tým.....	31
Obrázek 3 - Případy užití - člen týmu	32
Obrázek 4 - Případy užití - zápas	33
Obrázek 5 - Případy užití - statistiky.....	34
Obrázek 6 - Diagram aktivit - obnova hesla	36
Obrázek 7 - Diagram aktivit - vytvoření člena týmu	37
Obrázek 8 - Schéma propojení služeb AWS s vazbami typu klient-server.....	40
Obrázek 9 - Mapování požadavků na API	41
Obrázek 10 - Ukázka JWT tokenu	42
Obrázek 11 - Doménový model	44
Obrázek 12 - Obrazovka pro přihlášení	45
Obrázek 13 - Základní menu.....	46
Obrázek 14 - Členové týmu	47
Obrázek 15 - Zadávání statistik ze zápasu	48
Obrázek 16 - Výsledné statistiky zápasu	49

Tabulky

Tabulka 1 - Nálezy testování.....	53
-----------------------------------	----

1 Úvod

Tato bakalářská práce se zabývá vývojem aplikace Android [1], sloužící k zaznamenávání statistik z volejbalových utkání. Aplikace usnadňuje uživateli práci při zadávání údajů a následně poskytuje užitečná data. Kromě samotných statistik lze pomocí aplikace volejbalové týmy spravovat.

V současné době existuje mnoho volejbalových týmů, které využívají systémů pro záznam statistiky. Na základě nasbíraných údajů přizpůsobí přípravu hry (tréninky), aby zvýšili své šance na výhru. Tento fenomén záznamu statistik je standardem především na světové úrovni volejbalu a nejvyšších národních ligách, které se hrají. Tyto systémy jsou velice sofistikované, robustní a lze do nich zaznamenat prakticky všechny údaje, jež nás napadnou. Existuje ale také velká skupina lidí, pro kterou je většina těchto funkcí zbytečná, nebo si systém jednoduše nemohou dovolit, ať už z důvodů finančních nebo přílišné náročnosti na její provoz. Konkrétně se jedná o nižší či mládežnické soutěže nebo různé amatérské ligy. A právě na tuto cílovou skupinu je zaměřena tato bakalářská práce. Osobně hraji volejbal již 13 let a stále častěji zjišťuji, že komunita těchto lidí je široká a uplatnění statistik stále žádanější.

Na rozdíl od již zmíněných náročnějších systémů, ke kterým je ve většině případů zapotřebí notebook, si naše cílová skupina vystačí pouze s telefonem nebo tabletem. Podmínkou používání je v našem případě i internetové připojení, což ale není v dnešní době tak velká překážka. Nutnost připojení k internetu je z toho důvodu, aby šlo zaznamenávat statistiky současně z více zařízení najednou. Z vlastní zkušenosti vím, že zaznamenat všechny činnosti hry jedním člověkem je prakticky nemožné.

Mezi základní parametry aplikace se řadí důkladné zabezpečení. To především z důvodů stále častějších úniků informací a hackerských útoků. Dalšími požadavky jsou ukládání dat na server, jednoduchý a přehledný design, snadné ovládání, rozšiřitelnost aplikace do budoucna a možnost zadávání dat z více zařízení najednou. Touto vlastností, spolu s ukládáním dat na server, bude naše aplikace vynikat oproti konkurenčním řešením.

Práce je rozdělena na několik částí. Nejprve se seznámíme s existujícími aplikacemi a porovnáme jejich výhody a nevýhody. Detailně uvedeme přehled použitých technologií při vývoji. Zvláště využití služby Amazon Web Services¹, kterou využíváme jako serverovou stranu. Dále se zaměříme na návrh aplikace a v neposlední řadě přiblížíme konkrétní implementaci. Ta je rozdělena na serverovou část a samotnou aplikaci Android. Závěr práce je věnován testování aplikace.

1.1 Požadavky

V této části si shrneme základní požadavky na aplikaci.

- Fungující Android aplikace na mobilních telefonech a tabletech
- Důkladné zabezpečení
- Vytvoření uživatelského účtu a ukládání dat na server
- Zadávání statistik všech hráčů (ze zápasu) z více zařízení najednou
- Zobrazení výsledných statistik za zápas

¹ Cloudová služba společnosti Amazon (<https://console.aws.amazon.com/>)

2 Současná situace

Pro zaznamenávání volejbalových statistik lze použít různé techniky. Od nejprimitivnějšího způsobu jako papír s tužkou nebo tabulka Excelu až po nejmodernější programy pro počítače vyvíjené přímo za tímto účelem. My se však zaměříme na mobilní aplikace a jejich využití v této problematice. Nyní si představíme některé existující aplikace.

2.1 Existující aplikace

Aplikace k zápisu statistik z volejbalového utkání na platformě Android již existují. Avšak stále se u většiny z nich setkáváme s určitými nedostatky, které odrazují lidi od jejich používání. V následujících kapitolách si podrobně představíme několik nejstahovanějších aplikací z distribuční služby Google Play, které jsou na trhu. Jednotlivé aplikace jsou uváděny dle jejich názvu, případně společnosti, která je vyvinula.

2.1.1 DataProject

DataProject² je společnost poskytující několik druhů aplikací pro zaznamenávání statistik. Ty jsou rozděleny dle typu koncových zařízení. Tedy např. poskytuje program pro tablet, který však nelze spustit na běžném telefonu. Kromě aplikací do telefonů a tabletů nabízí software pro stolní počítače. Nyní si představíme jednotlivé verze aplikací.

A. Essential Stats Volleyball

Je to aplikace, kterou lze nainstalovat na všechny Android zařízení verze Android 4.2 nebo vyšší. Omezením je minimální uhlopříčka zařízení o velikosti 5,0". Používání aplikace vyžaduje vytvoření účtu, což považují za jeho výhodu, protože se uživatelé všechna data automaticky ukládají na server. Po přihlášení může uživatel vytvořit svůj tým a následně vkládat data z utkání. Essential Stats Volleyball³ se vyznačuje přehledným designem a jednoduchým zahájením zápasu. Bohužel samotné zaznamenávání dat se odlišuje tím, že se ukládají pouze bodové akce hry a nikoli všechny ostatní důležité zákroky v rámci výměny, jak je to zvykem v profesionálních systémech. Výsledné statistiky pak nejsou kompletní. To považují za velký nedostatek. Nevýhodou je také schopnost zaznamenávat pouze jeden tým a nikoli obě hrající družstva.

B. Click&Scout

Tato aplikace je především pro tablety nebo zařízení větší než 7,0" s Androidem 4.2 nebo vyšší. Je velmi rozšířená oproti Essential Stats Volleyball, kde bylo jen omezené množství funkcionalit. Pro přihlášení využívá stejného účtu jako Essential Stats Volleyball. Je zde k dispozici kompletní příprava zápasu včetně rozmístění hráčů na hřišti a stanovení dalších podrobností. Ovšem zadávání dat z utkání probíhá klikáním do hřiště a podchycováním úplně všech činností na hřišti. Click&Scout⁴ je také funkčně omezena a pro plnou verzi je zapotřebí roční licence za €49.00 (cena platná k 28. 10. 2017).

² Data Project. Data Project sport software. dataproject.com. [Online] 1982. <http://www.dataproject.com/EN/en/Volleyball>.

³ Data Project. Essential Stats Volleyball. dataproject.com. [Online] 2017. <http://www.dataproject.com/Products/GLOBAL/en/Volleyball/>.

⁴ Click&Scout. dataproject.com. [Online] 2017. <http://www.dataproject.com/Products/GLOBAL/en/Volleyball/ClickAndScout>.

Souhrn aplikací DataProject

Máme tedy dvě Android aplikace od společnosti DataProject. Jedna, která je příliš zjednodušená s nedostatkem funkcí (Essential Stats Volleyball) a druhá která je pro naši cílovou skupinu zbytečně složitá a poměrně drahá (Click&Scout).

2.1.2 Individual Volleyball Stats

Jak již název napovídá, tento systém uchovává informace primárně o jednom hráči, což nevyhovuje našim požadavkům. Pro sledování celého týmu je zapotřebí zakoupit upgrade za 37,65 Kč (cena platná k 29. 10. 2017). Následující text se bude vztahovat pouze k placené verzi, protože ta jediná se podobá našim požadavkům. Individual Volleyball Stats⁵ tedy není zdarma a její design příliš nezaujme. Vkládání statistik je vcelku nepřehledné, nelze znamenávat údaje soupeře a celkový počet zaznamenaných zápasů je omezen na pouhých 5. Aplikace také nepodporuje vytvoření účtu, do kterého by se uživatel ukládala vložená data. Disponuje pouze exportem výsledků do e-mailu a samotné zobrazení statistik v zařízení není v tabulkách, což nepůsobí přehledně.

2.1.3 McStats-VBall Free

Tato aplikace se přibližuje našim představám, alespoň co do způsobu zadávání statistik. To probíhá výběrem hráče a následným zadáním činnosti, kterou provedl. Uživatelské rozhraní působí celkově přehledně a všechny základní funkce zadávání statistik jsou k dispozici. Jen chování aplikace v určitých momentech hry zobrazí okno pro vložení trajektorie letu míče. To může být pro někoho, kdo tuto funkci nepotřebuje velice zdržující proces. Až na tuto funkcionalitu, která je dostupná pouze u tabletů, jsou všechny ostatní funkce stejné jak na telefonu, tak tabletu. Opět zde ale chybí možnost vytvoření účtu pro ukládání dat. Bohužel ani tato aplikace není se všemi plnohodnotnými funkcemi zdarma. V základní verzi je např. omezen počet vytvořených týmů nebo počet odehraných setů v zápasu. Plná verze této aplikace je k dispozici za 199,99 Kč (cena platná k 29. 10. 2017). Výhodou je však přehledné zobrazení výsledných statistik zarovnaných do jednotlivých tabulek nebo automatické otáčení hráčů na virtuálním hřišti během utkání. To je dostupné i ve verzi zdarma. McStats-VBall⁶ také umožňuje import/export týmu a zápasu, nebo volbu zaznamenávání jednoho nebo obou hrajících týmů. Disponuje možností vrátit akci zpět pro případ, kdy uživatel provede nechtěnou akci v zadávání statistik. Ovšem její tlačítko je umístěno nešikovně ve skrytém menu, kde si jej málo lidí všimne a tím tento proces zabere i značné množství času. Jde tedy o přijatelnou aplikaci, ale stále nevyhovuje našim požadavkům.

2.1.4 Quick Scout Volley Free

Quick Scout Volley⁷ se vyznačuje především moderním a přívětivým designem. Oproti ostatním nabízí přehledné menu a vyznačuje se unikátním způsobem zadávání statistik. I tato aplikace je však v základní verzi značně omezena a častými upozorněními nutí uživatele k upgradu. To je velice nepříjemné a z dlouhodobého hlediska může dojít k odinstalaci programu ze strany nespokojeného uživatele. Tento problém jde vyřešit zakoupením plné verze programu, která ovšem stojí 439,99 Kč (cena platná k 29. 10. 2017). Je jednou z nejdražších aplikací mezi

⁵ Arrowshark. Individual Volleyball Stats. play.google.com. [Online] 2012. <https://play.google.com/store/apps/details?id=com.dvballa21.hittingchart>.

⁶ McStats.McStats-VBall Free. play.google.com. [Online] 2017. <https://play.google.com/store/apps/details?id=com.McStats.vbstatsfree>.

⁷ Dot Next. Quick Scout Volley. quickscoutvolley.com. [Online] 2016. <http://www.quickscoutvolley.com/>.

našimi porovnávanými. Její upgrade zajistí oproti základní verzi značné množství nových a plnohodnotných funkcionalit jako roční statistiky, možnosti exportu statistik do PDF a další funkce. V základní verzi aplikace je omezený počet vytvořených týmů a zápasů. Dále je omezeno zadávání statistik pouze jednoho týmu. Aplikace je dostupná pouze pro tablety nikoli telefony. A hlavně neumožňuje vytvoření účtu pro zálohování dat.

2.1.5 SoloStats 123 Volleyball

Jako jedna z mála poskytuje vytvoření účtu, na který lze provádět zálohy veškerých dat. Uživatel může zakládat týmy, ale pro jeho vytvoření platí určitá kritéria, která uživatele mohou limitovat. Např. nelze si předpřipravít tým s méně než 6 hráči. Samotnému zadávání statistik předchází kompletní nastavení zápasu. Jde např. o rozmístění hráčů na hřišti, určení, kdo je nahrávač nebo který tým začíná podáním. Následné zadávání dat ze hry je podobné jako u ostatních systémů s tím, že v SoloStats 123 Volleyball⁸ jsou tlačítka pro jednotlivé aktivity odděleny výraznými barvami. Pro někoho však toto rozpořádání může působit zmateně neboť i názvy akcí nejsou standardní. Výhodou je přehledné tlačítko, které vrátí akci zpět v případě uživatelské chyby. Všechny akce z utkání, které chceme zaznamenat, je však potřeba potvrdit speciálním tlačítkem. Z provedeného dotazníku vyplývá, že zmíněné tlačítko spíše zdržuje. Jako mínus považuji to, že druhý tým lze zaznamenávat pouze jako celek a nikoli jednotlivé hráče. Výsledné statistiky primárního týmu jsou rozděleny na jednotlivé úkony a hráče.

2.1.6 Volleyball Stat!

Další aplikace, která po zapnutí přesměruje uživatele přímo do oblasti vkládání dat z utkání. Chybí zde hlavní menu, které by uživateli pomohlo se zorientovat a rozlišit tak jednotlivé části programu. V horní oblasti displeje je sice určitá lišta s nabídkou, je však nepřehledná, a ne u všech tlačítek je na první pohled jasné, jaká funkce se za nimi skrývá. Volleyball Stat⁹ také nedisponuje možností přihlášení k účtu. Uživatel je tedy odkázán pouze na lokální úložiště v zařízení. Vkládání statistik je řešeno 6 duplicitními buňkami, ve kterých jsou jednotlivé úkony hráčů. Každá buňka tedy reprezentuje jednoho hráče na hřišti. Uživatelské rozhraní tak působí velmi nepřehledně a nevyužívá efektivně plochu obrazovky. Krom toho aplikace obsahuje reklamy a pro jejich odstranění je nutné zaplatit 99,99 Kč (cena platná k 30. 10. 2017).

2.1.7 Volley Stats

Jako poslední si představíme Volley Stats¹⁰. Jde o velice primitivní aplikaci s jednoduchým designem a minimálním množstvím funkcionalit. Uživatel může spravovat pouze jeden tým, hráči jsou identifikováni jen jmény a jejich profily nejde upravovat. Pro změnu je tedy nutné hráče smazat a vytvořit ho znovu. Při zápisu utkání lze zaznamenávat pouze vlastní statistiky, nikoliv i statistiky soupeře. Samotná procedura vkládání dat je velice omezená a velké množství akcí, které se běžně zaznamenávají, chybí. Statistika tedy poté nejsou dostatečně směrodatné. Zápas nelze rozdělit na jednotlivé sety, a tak máme k dispozici výsledky pouze za celý zápas. Co se týče konečných statistik, jsou velice zjednodušené a data nejsou vyznačena procentuálně, jak je v dnešní době zvykem. Obecně je aplikace velmi jednoduchá a nepodchycuje všechny důležité aspekty volejbalového utkání.

⁸ Rotatel23. SoloStats 123 Volleyball. solostatslive.com. [Online] 2017. <https://www.solostatslive.com/>

⁹ Tywyll, Geoffrey. Volleyball Stat! volleyballstat.towell4.us. [Online] 2017. <http://volleyballstat.towell4.us/>

¹⁰ Volley Stats. Cedric Devriendt. [Online] 2014. <https://play.google.com/store/apps/details?id=com.oncreate.volleystats>

2.2 Shrnutí

Po prozkoumání aplikací vidíme, že existuje mnoho aplikací, které se danou problematikou zabývají, avšak u každé z nich lze nalézt určité nedostatky. Ve většině případů se setkáváme s podobnými typy problémů. Jedním z nich je chybějící funkcionalita zálohování veškerých dat aplikace na server prostřednictvím vytvořeného účtu a tím zajistit bezproblémovou synchronizaci a přenositelnost dat mezi různými zařízeními. Dalším problémem je cena. Ne všechny programy jsou totiž zdarma, a když už jsou, tak funkčně zaostávají. Velkým problémem je také rozdílnost v zaznamenávání statistik u jednotlivých programů. Jejich výsledky pak nejsou porovnatelné s těmi profesionálními. Nedostatkem je i omezení uživatelského rozhraní na jeden světový jazyk a nemožnost volby zaznamenávat statistiky jednoho nebo obou hrajících týmů.

3 Použité technologie

V této kapitole se zaměříme na jednotlivé technologie, kterých se v práci využívá. Mezi technologie řadíme návrhové vzory, frameworky, využívané služby či programy.

3.1 JPA

JPA [2] je framework programovacího jazyka Java. Ten umožňuje objektově relační mapování (ORM), což usnadňuje práci s ukládáním objektů do databáze pomocí anotací. JPA si specifikuje určitá pravidla pro zápis třídy v Javě, kterými následně popisuje entitu v databázi. Tyto entity jsou následně spravovány objektem třídy EntityManager.

3.2 REST

Jedná se o jednu z nejrozšířenějších architektur rozhraní současnosti, která je navržena pro distribuované prostředí. Ta využívá ke komunikaci jednoduchého HTTP protokolu, s nímž byl souběžně vyvíjen. Autorem je Roy Fielding, který ho navrhl a popsal v roce 2000 v rámci své disertační práce. Dá se říci, že REST je sada omezení, kterou když aplikujeme na návrh systému, vytváří architektonický styl softwaru. Pokud tedy implementujeme všechny náležitosti REST popsané v práci R. Fieldinga, dostaneme systém, který má specifická pravidla pro data, komponenty, odkazy, komunikační protokoly a pro ty, kteří data zpracovávají [3]. Není určen pouze pro webové služby.

3.3 Formát JSON

Celým názvem JavaScript Object Notation [4]. Je to způsob zápisu dat, který se používá především pro posílání informací po internetu. Jeho základem je definovaná struktura uložených dat (datový formát). Ta může být členěna do polí nebo zapouzdřena v objektech. Oproti formátu XML je JSON považován za více „odlehčenou“ a jednodušší alternativu.

3.4 Maven

Další použitou technologií je Apache Maven [5]. Jde o nástroj pro správu a automatizaci buildů¹¹ aplikací. Cílem je tedy usnadnění práce při buildování aplikace. Podporuje různé programovací jazyky, ale primárně je určen pro jazyk Java. Hlavním cílem je vytvářet soubory jako jsou JAR, WAR a EAR. Dále lze provádět vedlejší operace, jako je např. vzdálené nasazení nebo kompletní vydání projektu díky zachování čitelnosti, znovupoužitelnosti a konzistentnosti během vývoje.

3.5 Amazon Web Services

Serverová část aplikace běží na Amazon Web Services (AWS) [6][7][8]. Jedná se o bezpečnou cloudovou službu¹² společnosti Amazon, nabízející řadu praktických služeb, které pokrývají oblasti, jako jsou databáze, API, správa uživatelů a mnoho dalších. Výhodou je poskytování služby, která umožňuje zobrazení statistik z využívání ostatních služeb. Tím má

¹¹ Proces vývoje softwaru.

¹² Poskytování služeb po internetu.

uživatel přehled např. o vytíženosti serveru. AWS bylo oficiálně založeno v roce 2006. Disponuje širokou sítí datových úložišť po celém světě. Tím je zajištěna rychlá dostupnost služeb velkému množství uživatelů a bezpečnost v případě havárie některého z datových center. AWS disponuje možností správy většiny jejích služeb pomocí příkazového řádku. Tato možnost se využije pro zautomatizování určitého druhu úkonu, který je zapotřebí vykonat.

Využívané služby AWS jsou běžně zpoplatněny, avšak pro tento projekt je využit omezený režim užívání služeb na jeden rok zdarma. Tato omezení nemají zásadní vliv na chod serveru ani jeho rychlost. Po skončení ročního období se tedy bude muset počítat s určitými náklady na provoz serverové části aplikace.

Hlavní důvod použití této služby je v ulehčení práce s implementací serverové strany. Nejužitečnější výhody jsou automatická škálovatelnost, snadná správa jednotlivých služeb, bezpečná správa uživatelských účtů, automatické zálohy databáze a další. Nyní si popíšeme hlavní služby AWS, které jsou v práci použity. Názvy jednotlivých služeb jsou zároveň nadpisy.

3.5.1 Cognito

Tato služba je určena pro administraci uživatelských účtů a jejich ověřování nezávisle na použité platformě koncového zařízení. Cognito ukládá a synchronizuje data koncových uživatelů, což usnadní práci při vývoji softwaru. Disponuje všemi základními prvky, které nás ve spojení s uživateli napadnou. Podporuje procesy jako registrace a následné ověření uživatele pomocí emailu nebo SMS zprávy, zapomenutí hesla, přihlašování, změny hesla a další. Umožňuje přihlášení pomocí jiných účtů, jako jsou Google, Amazon, Facebook nebo Twitter. Tím je zajištěno široké uplatnění, protože v dnešní době má většina uživatelů alespoň jeden z těchto účtů.

3.5.2 API Gateway

Jednou z využitých služeb AWS je API Gateway (dále jen API). Jde o část, která obsluhuje příchozí požadavky od uživatelů a patřičně na ně reaguje. API rozhoduje, které požadavky přijme a které odmítne. Funguje tedy jako vstupní a výstupní brána serveru. Na API také probíhá autorizace uživatelů, která rozhodne, zda jsou oprávněni dané akce provádět či nikoliv. Tato autorizace může být provedena buď vlastním systémem ověření nebo využitím služby Cognito popsané výše. Příchozí požadavky se třídí dle adresy URL a na základě té API rozhodne, co se bude dít dále s požadavkem, popř. kam se přepošle ke zpracování.

3.5.3 Lambda

Důležitou komponentou je Lambda, což je výpočetní služba, která funguje jako Serverless. Služba se dělí na jednotlivé Lambda funkce, které spouští kód v reakci na události, které zpracovávají. Ve většině případů zpracovávají požadavky, které přijdou na API Gateway. Pokud je v systému zakomponována i databáze, pak může sloužit jako prostředník mezi API a databází. Každá Lambda funkce má své specifické nastavení, které lze podřídit vykonávané akci. Lze např. využít předdefinovaných proměnných, pomocí kterých se lze připojit k databázi. Kód, který funkce vykonávají, může být naprogramován v následujících programovacích jazycích: Node.js, Java, C#, Go a Python.

3.5.4 RDS

Další službou je Relation Database Services (RDS), zaštiťující databáze. Poskytuje jednoduché nastavení a správu relačních databází jako jsou PostgreSQL, MySQL, MariaDB, Oracle, Amazon Aurora a Microsoft SQL Server. Stará se o aktualizaci běžícího softwaru, detekci

chyb a obnovu dat. Škálovatelnost databázových instancí je velká, od drobných databází až po gigantické. RDS disponuje automatickými zálohami a možností zálohy v libovolný okamžik kdy uživatel potřebuje. Lze tedy kdykoliv provést obnovu dat na základě některé zálohy. Možná je také migrace databázové instance na větší (v případě vysokého nárůstu spravovaných dat).

Pro správu databáze a jejího obsahu byl použit program pgAdmin 4. Ten umožňuje administraci a nástroje pro management PostgreSQL databází. Podporuje např. vytváření tabulek či jejich sloupců. Pomocí tohoto programu lze také přidávat, editovat nebo mazat data v konkrétních tabulkách.

3.5.5 EC2

Službou poskytující výpočetní prostředky je Elastic Compute Cloud (EC2). Ta uživatelům nabízí virtuální stroj, na kterém mohou spouštět své aplikace. EC2 poskytuje vývojářům nástroje pro vytváření odolných aplikací proti selhání a odděluje je od běžných scénářů selhání.

3.5.6 S3

Celým názvem Simple Storage Service (S3) je úložiště vytvořené pro ukládání a načítání libovolného množství dat z libovolného místa (webové stránky, mobilní aplikace nebo dnes populární internet věci). Amazon zaručuje 99,999999999% spolehlivost a také vysokou bezpečnost z hlediska regulačních požadavků. S3 má jednoduché webové rozhraní.

3.5.7 VPC

Virtuálním místem v rámci AWS je Virtual Private Cloud (VPC). Umožňuje spouštět zdroje AWS v této virtuální síti. VPC má oproti vlastním virtuálním sítím výhodu škálovatelnosti infrastruktury systému AWS. Poskytuje řadu bezpečnostních nastavení, co se týče příchozích a odchozích spojení ať už s okolním internetem nebo vnitřní sítí AWS.

3.5.8 Cloud Watch

Jedná se o službu pro monitorování jednotlivých zdrojů a aplikací, které na AWS běží. Díky této službě lze sledovat vytížení používaných zdrojů a případně je podle toho přizpůsobit. Cloud Watch disponuje vytvářením alarmů, které hlásí dané problémy či přílišné zatížení. Výsledné údaje jsou velmi přehledné a lze je zobrazit např. v grafech.

3.5.9 IAM

Jako poslední si představíme službu Identity and Access Management (IAM). Ta slouží ke spravování přístupů k jednotlivým službám a zdrojům AWS. Lze vytvářet uživatele nebo celé skupiny a nastavit jim patřičná oprávnění.

3.6 Vývojové prostředí

Pro implementaci serveru a mobilní aplikace byly použity následující programy. Důvodem pro jejich použití byly dobré zkušenosti z již předešlých prací v rámci vyučovaných předmětů.

3.6.1 IntelliJ IDEA

Celá část serveru, zpracovávající požadavky uživatelů, je vytvořena v programu IntelliJ IDEA¹³ (IDEA) od společnosti JetBrains. Jedná se o vývojové prostředí primárně určené pro programovací jazyk Java. K vývoji byla použita školní licence tohoto programu. Mezi výhody tohoto vývojového prostředí patří integrace služby pro verzování Git.

3.6.2 Android Studio

Samotná Android aplikace je vyvíjena v programu Android Studio¹⁴. Je to vývojové prostředí pro programování android aplikací, založené na IntelliJ IDEA. Na rozdíl od IDEA je však Android Studio volně dostupné. Jeho ovládání je velice intuitivní a podporuje snadné vytváření grafické podoby aplikace. Toto vývojové prostředí také podporuje integraci služby Git.

3.7 Git

Jedná se o distribuovaný systém správy verzí, který vytvořil Linus Torvalds v roce 2005 [9]. Veškeré zdrojové kódy této práce jsou pomocí tohoto systému verzovány. Konkrétně se využívá webové služby Bitbucket společnosti Atlassian. Ta s určitými omezeními oproti placené verzi poskytuje hosting zdarma. Pro účely této práce je ale postačující. Krom nástroje Git podporuje Bitbucket i nástroj Mercurial.

Výhodou této služby je velmi dobrý přehled nad zdrojovými kódy a jejich aktuálností. Vývoj softwaru je díky této technologii mnohem efektivnější a bezpečnější. Využívá se hlavně u velkých projektů, na kterých pracuje více osob.

¹³ <https://www.jetbrains.com/idea/>

¹⁴ <https://developer.android.com/studio/>

4 Návrh aplikace

Aplikace je navržena jako systém server-klient. Server zajišťuje služba AWS a klientem je aplikace android. Návrh aplikace je velice důležitým aspektem pro její budoucí rozšíření. Pokud by se již v počátku navrhla špatně, mohla by být případná vylepšení neuskutečnitelná. V této kapitole také zhodnotíme výsledky dotazníku vytvořeného na základě častých rozdílů aplikací zmíněných v kapitole 2.1.

4.1 Požadavky

Všechny požadavky na systém lze rozdělit na dvě kategorie, a to na funkční a nefunkční.

4.1.1 Funkční

Funkční požadavky specifikují požadované služby systému či úkony, akce a aktivity, které musí být vykonány. Měly by být podchyceny reakce systému na různé situace. V následující části si představíme funkční požadavky našeho systému.

A. Uživatelský účet

- **Registrace uživatele**
Systém bude umožňovat registraci nového uživatele.
- **Ověření uživatele**
Systém bude umožňovat ověření uživatelského emailu pomocí odkazu poslaného na zadaný email.
- **Přihlášení uživatele**
Systém bude umožňovat přihlášení již zaregistrovaného a ověřeného uživatele.
- **Zapomenutí hesla**
Systém bude umožňovat proces obnovy hesla v případě jeho ztráty.
- **Změna hesla**
Systém bude umožňovat přihlášenému uživateli změnu hesla.
- **Odeslání zpětné vazby**
Systém bude umožňovat odeslání zpětné vazby od uživatele.

B. Tým

- **Zobrazení týmů**
Systém bude umožňovat zobrazení týmů, ke kterým má nějaký vztah.
- **Založení týmu**
Systém bude umožňovat založení nového týmu.
- **Editování týmu**
Systém bude umožňovat úpravu týmu.
- **Smazání týmu**
Systém bude umožňovat smazání týmu.
- **Připojení k týmu**
Systém bude umožňovat uživateli připojit se do určitého týmu.

C. Členové týmu

- **Zobrazení členů týmu**
Systém bude umožňovat zobrazení členů týmu.
- **Přidání člena do týmu**
Systém bude umožňovat přidání člena do týmu.
- **Editování člena týmu**
Systém bude umožňovat změnu údajů člena týmu.
- **Smazání člena týmu**
Systém bude umožňovat smazání člena týmu.

D. Role

- **Typy rolí**
Systém bude podporovat 3 typy rolí: administrátor, trenér a hráč. Administrátor bude mít všechna oprávnění nad daným týmem. Roli trenéra bude umožněno vytvářet zápasy, upravovat je a následně zaznamenávat statistiky. Hráč bude mít práva pouze pro čtení.
- **Přidělení role**
Systém bude umožňovat přidělení role uživateli.
- **Odebrání role**
Systém bude umožňovat odebrání role uživateli.

E. Zápas

- **Zobrazení zápasů**
Systém bude umožňovat zobrazení zápasů daného týmu.
- **Vytvoření zápasu**
Systém bude umožňovat vytvoření zápasu.
- **Editace zápasu**
Systém bude umožňovat úpravu již vytvořeného zápasu.
- **Zadávání statistik**
Systém bude umožňovat zadávání skóre a statistik ze zápasu (všechny herní činnosti – podání, příjem, nahrávka, útok, blok, vybírání v poli a míče tzv. zadarmo).
- **Zobrazení statistik**
Systém bude umožňovat zobrazení statistik ze zápasu.

4.1.2 Nefunkční

Nefunkční požadavky slouží k tomu, aby vyvíjená aplikace byla kvalitní a stabilní. Jsou to omezení kladená na systém nebo celý proces vývoje. V našem případě do této části spadají následující požadavky.

A. Provoz serveru

Část aplikace bude provozována na vzdáleném serveru. K využití služeb aplikace bude nutné internetové připojení.

B. Spolehlivost / Persistence

Systém bude spolehlivý a bude zajištěno ukládání dat na serveru. Data na serveru budou uložena v databázi, která bude pravidelně zálohována.

C. Škálovatelnost

Server dokáže pojmout vysoký počet požadavků od uživatelů. Vysoký počet požadavků zároveň nesmí omezit rychlost odezvy serveru, případně omezit maximální počet připojených uživatelů.

D. Bezpečnost

Systém bude pracovat na bezpečném protokolu HTTPS. Systém bude podporovat autentifikaci při přihlášení a autorizaci při provádění akcí v rámci používání aplikace. Uživatelská hesla budou splňovat požadovanou složitost.

E. Správná práce s rolemi

Systém bude podporovat správnou práci s uživatelskými rolemi a bude dodržovat jejich oprávnění pro všechny uživatele. Pokud bude mít uživatel více rolí, bude se při ověřování prováděných akcí brát ta nejvyšší.

F. Rozšiřitelnost

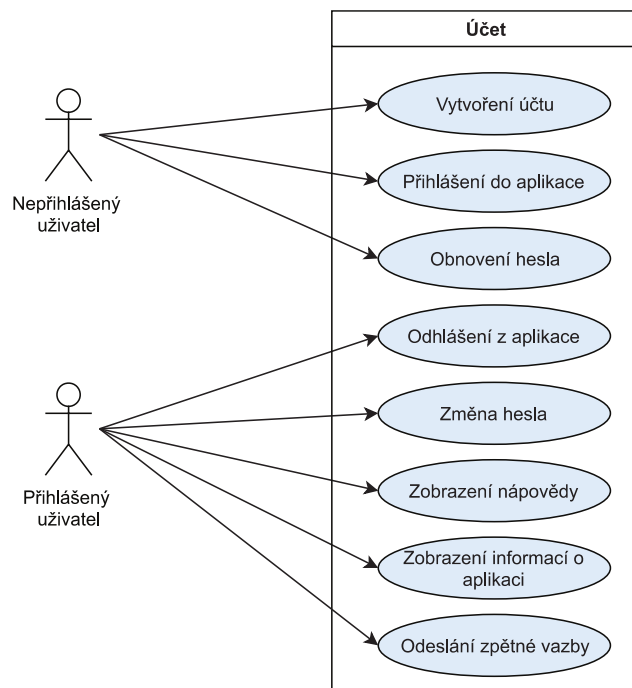
Systém bude navržen tak, aby byl do budoucna rozšiřitelný a zároveň byla možná případná vylepšení aplikace.

4.2 Případy užití

Tato část rozpracovává chování systému z hlediska uživatele. Na rozdíl od funkčních požadavků se zde zachází do větších detailů. Každý případ užití vychází alespoň z jednoho systémového požadavku. Případy užití si znázorníme pomocí diagramů. V diagramech vystupují aktéři, jimiž jsou v našem případě uživatelé, používající aplikaci. Konkrétně nepřihlášený uživatel a poté přihlášený uživatel, kterého ještě dělíme podle rolí na hráče, trenéra a administrátora.

4.2.1 Účet

V těchto případech užití se setkáváme i s aktérem, kterým je nepřihlášený uživatel. Obrázek 1 znázorňuje, které operace smí provádět. Dále se zde nachází základní funkce, které uživatelé mohou využít.



Obrázek 1 - Případy užití - účet a základní funkce

A. Vytvoření účtu

Uživatel zapne aplikaci a ta zobrazí stránku pro přihlášení. Zde uživatel klikne na tlačítko pro vytvoření účtu. Aplikace ho tímto přesměruje na stránku pro registraci, kde uživatel vyplní potřebné údaje a potvrdí tlačítkem. Systém zpracuje požadavek a v případě úspěšné registrace odešle potvrzovací e-mail. Uživatel klikne na odkaz v potvrzovacím e-mailu, který mu byl zaslán. Systém poté označí uživatelský účet jako ověřený.

B. Přihlášení do aplikace

Uživatel zapne aplikaci, ta následně vykreslí přihlašovací stránku. Na této stránce uživatel vyplní přihlašovací údaje a potvrdí tlačítkem. Systém ověří přihlašovací údaje a podle nich buď uživatele přihlásí, nebo nikoliv. Pokud uživatel již při předchozím přihlášení zaškrtl zapamatování přihlašovacích údajů, načtou se údaje automaticky při načtení této stránky. V tomto případě pak stačí rovnou kliknout na tlačítko pro přihlášení.

C. Obnovení hesla

Uživatel na přihlašovací stránce klikne na tlačítko pro případ zapomenutí hesla. Tím aplikace vykreslí stránku pro obnovu hesla s políčkem pro vyplnění e-mailové adresy. Uživatel ji vyplní a potvrdí tlačítkem. Systém ověří e-mailovou adresu a v případě úspěchu na ni odešle ověřovací kód a uživatele přesměruje na další stránku. Na té uživatel vyplní ověřovací kód, který mu byl zaslán na e-mail, zadá dvakrát nové heslo a potvrdí tlačítkem. Systém zkontroluje zadané údaje a v případě úspěchu nastaví uživateli nové heslo. V opačném případě je uživatel aplikací upozorněn na chybné údaje.

D. Odhlášení z aplikace

Přihlášený uživatel na domovské stránce klikne na tlačítko hlavní nabídky. Aplikace zobrazí postranní menu. Nyní uživatel klikne na horní část, která se týká jeho účtu. Tím aplikace změní spodní část menu. V tomto menu uživatel klikne na možnost odhlášení a systém následně uživatele odhlásí.

E. Změna hesla

Přihlášený uživatel si zobrazí v postranním menu svůj účet. V nabídce menu zvolí možnost změny hesla. Aplikace mu zobrazí novou stránku, kde je třeba vyplnit aktuální heslo a dvakrát nové heslo. Uživatel údaje vyplní a potvrdí tlačítkem. Systém zkontroluje data a v případě úspěchu přenastaví uživateli heslo. V opačném případě aplikace upozorní na špatně vyplněné údaje.

F. Zobrazení nápovědy / informací o aplikaci

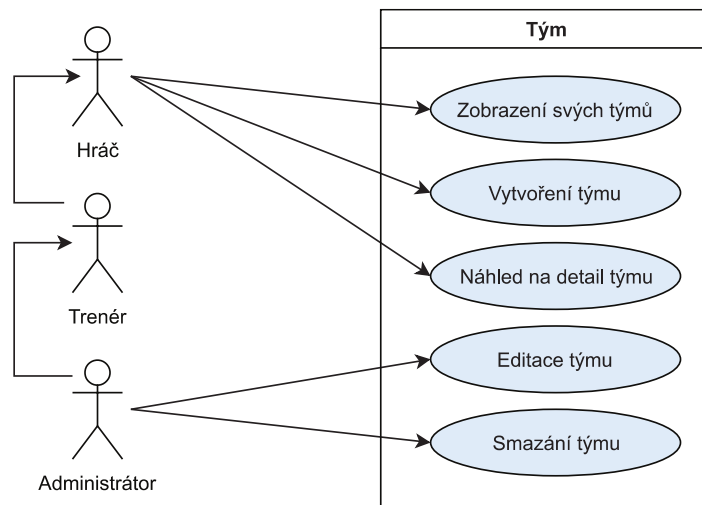
Přihlášený uživatel v postranním menu hlavní nabídky klikne na možnost nápovědy nebo informací o aplikaci. Aplikace následně zobrazí požadovanou stránku.

G. Odeslání zpětné vazby

Přihlášený uživatel zvolí v postranním menu hlavní nabídky možnost zpětné vazby. Tím aplikace vykreslí stránku pro zadání zpětné vazby. Uživatel následně napíše své připomínky a poznámky k aplikaci do textového pole a odešle kliknutím na tlačítko. Systém zpracuje požadavek.

4.2.2 Tým

Následujících případy užití (Obrázek 2) obsahují aktéry, kterými jsou vždy přihlášení uživatelé. Na tomto diagramu lze vidět jednotlivé případy užití vztahující se k položce tým.



Obrázek 2 - Případy užití - tým

A. Zobrazení svých týmů

Přihlášený uživatel v domovské stránce zobrazí postranní menu a klikne na týmy. Aplikace načte týmy daného uživatele.

B. Vytvoření týmů

Přihlášený uživatel si zobrazí své týmy a klikne na tlačítko plus. Aplikace zobrazí stránku pro vytvoření týmu. Uživatel vyplní potřebné údaje a potvrdí tlačítkem. Systém požadavek zpracuje a v případě úspěchu informuje o vytvoření týmu. Systém přesměruje uživatele zpět na jeho týmy.

C. *Náhled na detail týmu*

Přihlášený uživatel si zobrazí své týmy a klikne na jeden z nich. Aplikace zobrazí novou stránku s jeho podrobnostmi.

D. *Editace týmu*

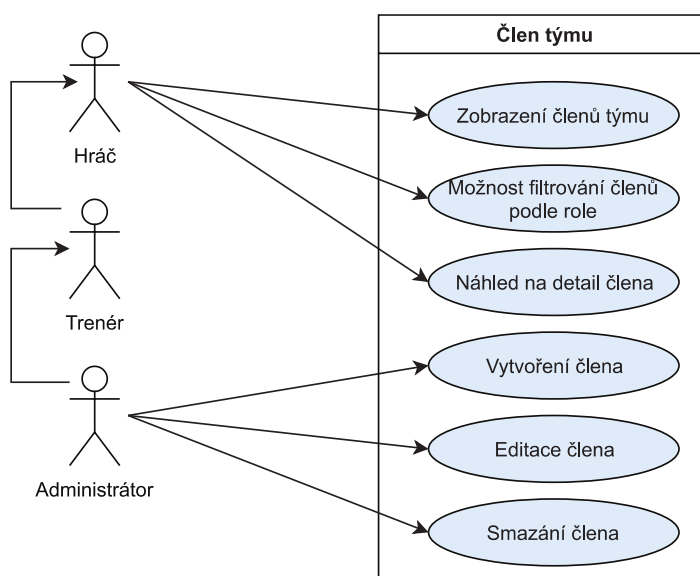
Přihlášený uživatel s rolí administrátora si zobrazí podrobnosti týmu, který chce upravovat. Pak změní potřebné údaje v předvyplněných formulářích a potvrdí změny kliknutím na tlačítko. Systém změny zpracuje, a pokud došlo k validním změnám, údaje uloží. V opačném případě aplikace zahlásí patřičnou chybu, nebo informaci, že údaje nebyly změněny.

E. *Smazání týmu*

Přihlášený uživatel s rolí administrátora si zobrazí detaily týmu k smazání. Klikne na tlačítko smazat. Aplikace zobrazí dialogové okno, v kterém uživatele vybízí k potvrzení akce. V případě potvrzení systém daný tým smaže. V opačném případě se nic nestane.

4.2.3 Člen týmu

Tyto případy užití jsou vztaženy k členům týmu a jejich správě. Konkrétní oprávnění jednotlivých uživatelů dle role popisuje Obrázek 3.



Obrázek 3 - Případy užití - člen týmu

A. *Zobrazení člena týmu*

Přihlášený uživatel si zobrazí podrobnosti daného týmu. Poté klikne na tlačítko pro zobrazení všech členů týmu. Systém následně zobrazí novou stránku, kde vypíše všechny členy aktuálního týmu.

B. *Možnosti filtrování členů podle role*

Přihlášený uživatel si zobrazí členy daného týmu a pomocí výsuvného menu zvolí roli, kterou chce zobrazit. Aplikace změnu vybrané role zachytí a podle ní systém zobrazí příslušné záznamy s danou rolí.

C. Náhled na detail člena

Přihlášený uživatel si zobrazí členy některého z týmů. Poté klikne na některého člena a aplikace zobrazí jeho podrobnosti.

D. Vytvoření člena

Přihlášený uživatel s rolí administrátora si zobrazí členy týmu, do kterého chce člena přidat. Klikne na tlačítko plus, sloužící pro přidání. Aplikace zobrazí novou stránku pro vytvoření člena. Uživatel vyplní formulář patřičnými daty a potvrdí tlačítkem. Systém požadavek zpracuje.

E. Editace člena

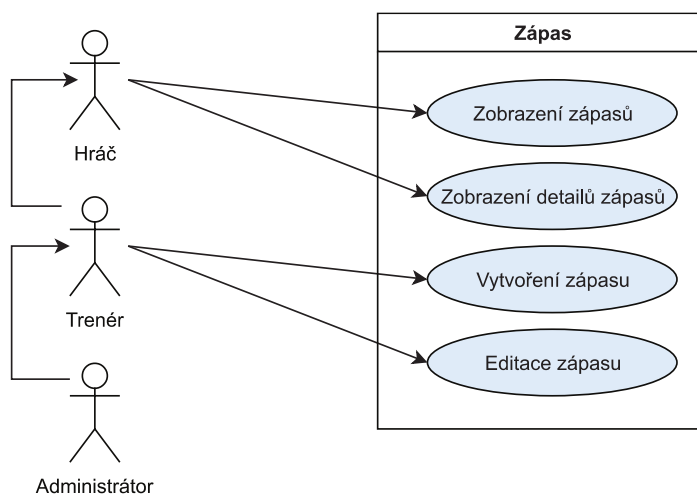
Přihlášený uživatel s rolí administrátora si zobrazí podrobnosti člena, kterého chce upravovat. Uživatel změní patřičné údaje v zobrazeném formuláři a potvrdí změny tlačítkem. Systém změny zpracuje.

F. Smazání člena

Přihlášený uživatel s rolí administrátora si zobrazí podrobnosti člena, kterého chce smazat. Klikne na tlačítko pro smazání. Systém vzápětí zobrazí dialogové okno pro potvrzení smazání. V případě potvrzení ze strany uživatele dojde ke smazání člena týmu. V opačném případě se nic nestane.

4.2.4 Zápas

K položce zápasu jsou uživatelům umožněny následující operace zobrazené níže (Obrázek 4).



Obrázek 4 - Případy užití - zápas

A. Zobrazení zápasů

Přihlášený uživatel si zobrazí některý tým a v podrobnostech klikne na tlačítko zápasy. Aplikace poté v nové stránce načte všechny zápasy, v kterých se tento tým vyskytuje.

B. Zobrazení detailů zápasu

Přihlášený uživatel si zobrazí zápasy některého týmu a klikne na jeden z nich. Aplikace zobrazí novou stránku s detaily daného zápasu.

C. Vytvoření zápasu

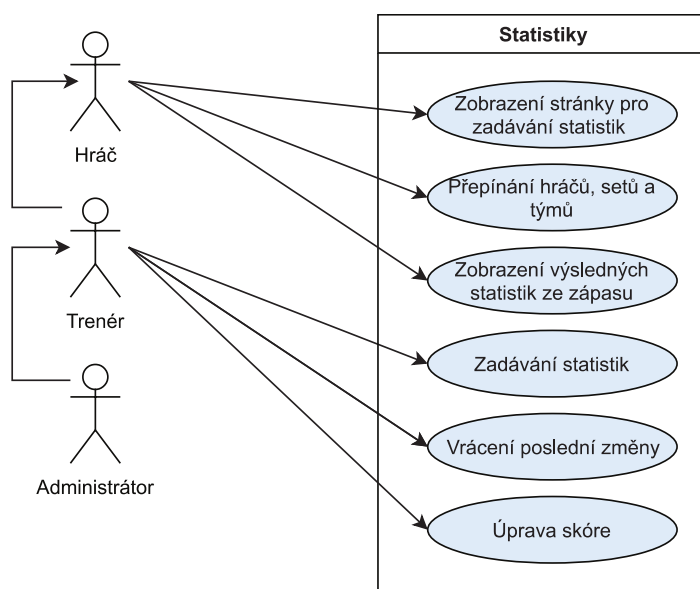
Přihlášený uživatel s rolí trenéra nebo administrátora si zobrazí zápasy některého z týmů, u kterého chce zápas vytvořit. Poté klikne na tlačítko plus. Aplikace zobrazí novou stránku pro přidání zápasu. Uživatel vyplní formulář a potvrdí tlačítkem. Systém následně zpracuje požadavek.

D. Editace zápasu

Přihlášený uživatel s rolí trenéra nebo administrátora si zobrazí podrobnosti některého zápasu, který chce upravovat. Uživatel změni potřebné údaje a potvrdí tlačítkem. Systém změny zpracuje.

4.2.5 Statistika

Co se týče zadávání statistik ze zápasu a následného náhledu na celkové statistiky jsou uživatelům k dispozici následující operace (popisuje Obrázek 5).



Obrázek 5 - Případy užití - statistiky

A. Zobrazení stránky pro zadávání statistik

Uživatel se přihlásí do aplikace a zobrazí některý zápas. V podrobnostech zápasu klikne na tlačítko pro načtení zápasu. Tím aplikace otevře novou stránku se zápasem, v které lze zadávat statistiky.

B. Přepínání hráčů, setů a týmů

Přihlášený uživatel si načte stránku pro zadávání statistik. Zde vybere jednotlivé hráče kliknutím na jejich číslo nebo iniciály. Aplikace zvoleného hráče obarví výraznou barvou tak, aby byl zřetelně vidět. Změnu setů uživatel provede pomocí tlačítek s šipkami. Aplikace vzápětí načte všechny údaje spojené s tímto setem. Kliknutím na název týmu, aplikace obarví název zvoleného týmu.

C. Zobrazení výsledných statistik ze zápasu

Přihlášený uživatel si zobrazí některý zápas a klikne na tlačítko pro načtení výsledků a statistik ze zápasu. Aplikace zobrazí přehled o zápasu a statistiky konkrétních hráčů za celé utkání.

D. Zadávání statistik

Přihlášený uživatel s rolí trenéra nebo administrátora si zobrazí stránku pro zadávání statistik ze zápasu. Označí některého z hráčů a poté klikne na tlačítka s konkrétní herní činností. Aplikace akci zpracuje a zobrazí případnou změnu skóre v případě, že šlo o bodovou akci.

E. Vrácení poslední změny

Přihlášený uživatel s rolí trenéra nebo administrátora si otevře stránku pro zadávání statistik, označí některého hráče, na kterém chce vrátit poslední změnu a klikne na tlačítko zpět. Systém vrátí poslední změnu daného hráče.

F. Úprava skóre

Přihlášený uživatel musí mít roli trenéra nebo administrátora. Zobrazí si stránku pro zadávání statistik a označí tým, u kterého chce provést změnu skóre. Následně klikne na tlačítko plus nebo mínus. Systém požadavek zpracuje a případně skóre patričným způsobem upraví.

4.3 Dotazník ohledně GUI

V rámci vyvíjení aplikace bylo provedeno zkoumání za účelem zjistit určité detaily v rámci procesu zadávání statistik z utkání. Na základě těchto nasbíraných dat bylo navrženo vhodné grafické rozhraní (GUI) aplikace. Zkoumání probíhalo formou dotazníku a testu se zúčastnilo 20 účastníků (převážně hrající volejbalisté). Otázky byly vytvořeny na základě již dříve zmíněných aplikací.

Z odpovědí jsem došel k závěru, že většina účastníků testu vyhodnotila potvrzovací tlačítko akce (např. přidání bodu některému z hráčů) jako zbytečné. Zvláště pokud aplikace disponuje možností tlačítka zpět, které předešlou akci vrátí. Dalším poznatkem průzkumu je, že ~80 % dotázaných by upřednostnilo zadávání statistik pomocí klikání na tlačítka oproti tzv. drag-and-drop (táhni a pusť) režimu. Posledním přínosem testu je určení způsobu seskupení tlačítek na obrazovce zařízení pro zadávání statistik. Účastníci vybírali ze dvou možností seskupení, a to podle typu herní činnosti nebo konečného výsledku dané činnosti. V tomto případě se ~95 % dotázaných se vyslovilo pro variantu seskupení tlačítek dle jejich herní činnosti.

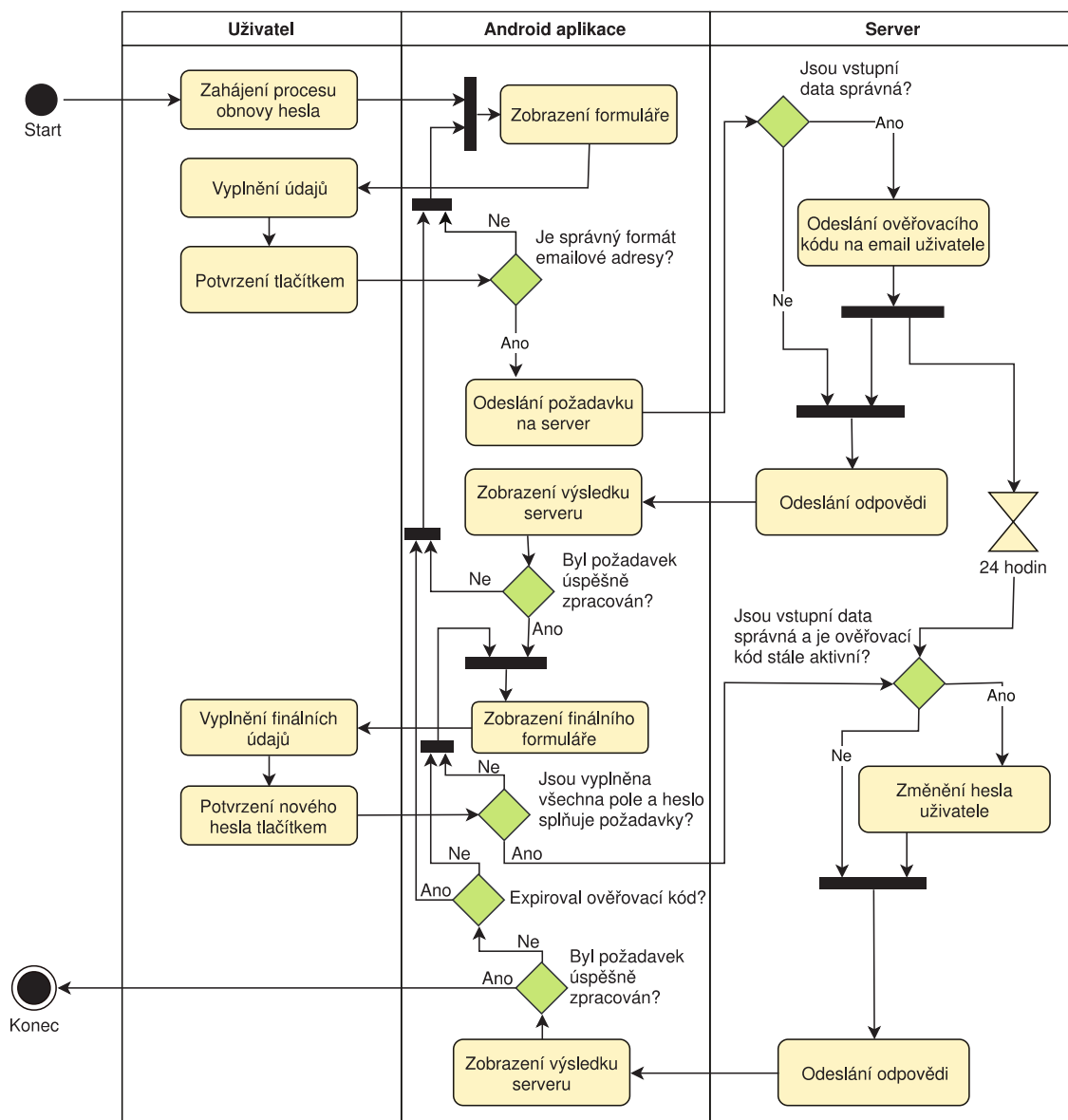
4.4 Diagram aktivit

Diagram aktivit znázorňuje průběh některého procesu v aplikaci a rozděluje jednotlivé úkony uživatele od těch systémových. V našem případě rozdělujeme proces do tří sekcí, kterými jsou uživatel, aplikace Android a server. Nyní si ukážeme případy interakce uživatele s aplikací, které se řadí k těm složitějším. Ostatní procesy vypadají podobně, ale jsou o něco jednodušší. Vybranými procesy jsou:

- Obnova hesla
- Vytvoření člena týmu
 - Vytváření ostatních typů entit v aplikaci probíhá obdobným způsobem jako vytvoření člena týmu.

4.4.1 Obnova hesla

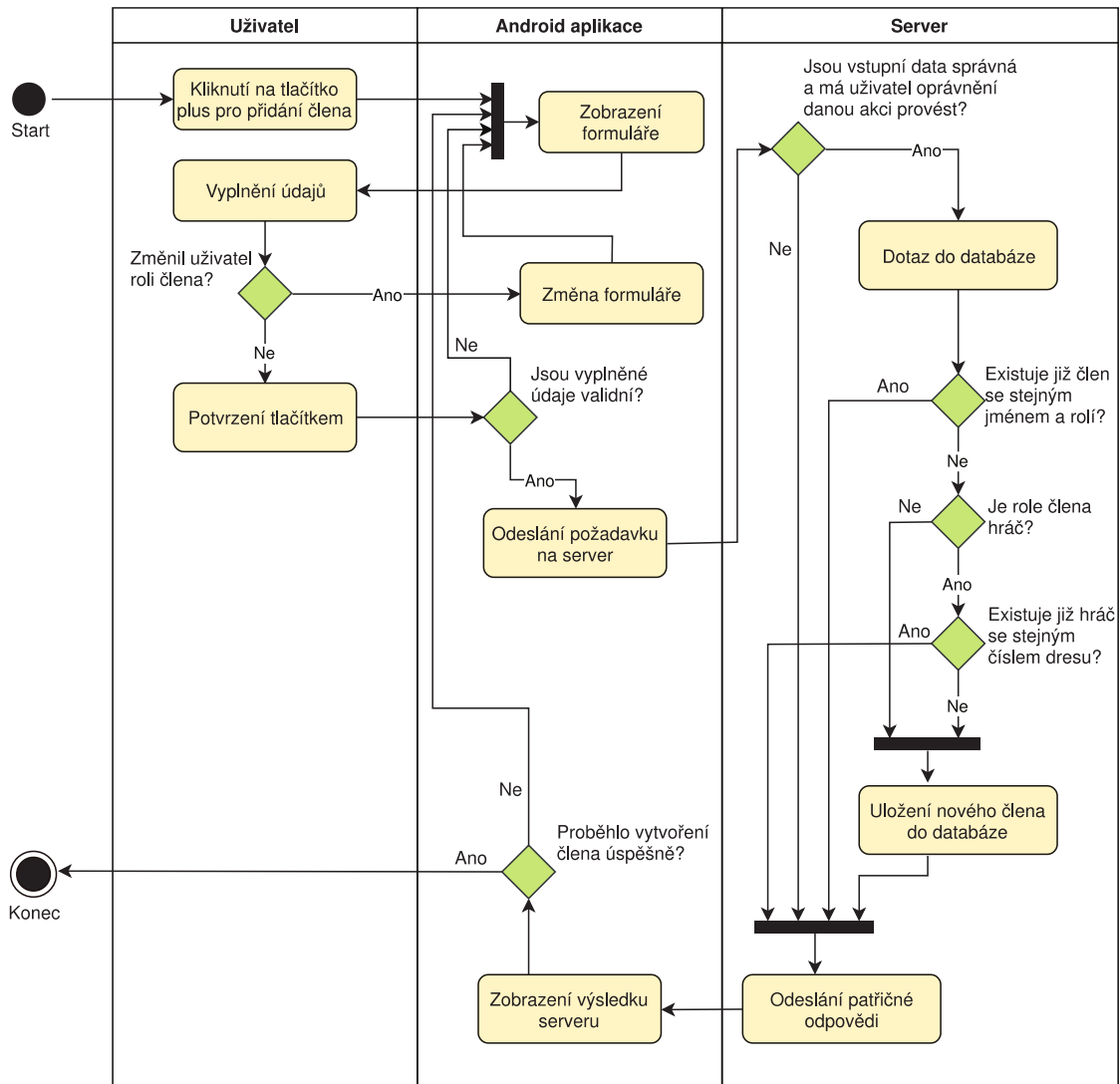
Obrázek 6 zobrazuje kompletní proces obnovy hesla. Začíná se vyplněním emailové adresy. Pokud ji uživatel nevyplní správně, může ji opravit. V případě úspěchu je požadavek odeslán na server, kde proběhne další kontrola. Pokud je vše v pořádku server odešle ověřovací kód a vrátí odpověď. Aplikace změnu zpracuje, a pokud je vše v pořádku vybědne uživatele k vyplnění poslední části údajů (jedním z nich je ověřovací kód). Aplikace opět zkontroluje vyplněné údaje, a pokud je vše v pořádku odešle požadavek na server. Zde proběhne další kontrola vstupních dat. Když jsou v pořádku, server provede změnu hesla a vrátí patřičnou odpověď. Pokud se tento scénář neuskuteční do 24 hodin od vygenerování ověřovacího kódu (lze vidět na diagramu - Obrázek 6), akce změny hesla neproběhne úspěšně. Aplikace poté danou odpověď zobrazí uživateli.



Obrázek 6 - Diagram aktivit - obnova hesla

4.4.2 Vytvoření člena týmu

Uživatel nejprve klikne na tlačítko pro přidání člena týmu. Aplikace vykreslí formulář, do kterého uživatel zadá údaje. Operaci potvrdí tlačítkem a aplikace zkontroluje, zda jsou vyplněná data správná. Pokud ano je požadavek odeslán na server, jinak je uživatel vyzván k úpravě formuláře. Na serveru proběhne hned několik kontrol, a pokud je vše v pořádku uloží se nový člen do databáze. Na základě odpovědi ze serveru aplikace zobrazí patřičné informace. V případě, že uložení proběhlo úspěšně, proces končí. Pokud ne je uživatel vyzván k překontrolování údajů. Tento proces popisuje Obrázek 7.



Obrázek 7 - Diagram aktivit - vytvoření člena týmu

5 Implementace

Aplikace je rozdělena na 2 základní části, a to serverovou část a aplikaci Android. Obě tyto části jsou vyvíjeny samostatně a jejich interakce je pomocí vytvořeného API. Veškerá data v aplikaci, se kterými uživatel přichází do styku, se ukládají na server. Zde jsou bezpečně uložena a připravena pro jejich znovupoužití. Důležitým aspektem celé aplikace je její zabezpečení z důvodu stále častějších výskytů hackerských útoků či úniku informací. Proto je na bezpečnost kladen vysoký důraz. Komunikace mezi serverem a aplikací je řešena pomocí zabezpečeného protokolu HTTPS. Hesla uživatelů musí splňovat kritérium alespoň 8 znaků a jednu číslici.

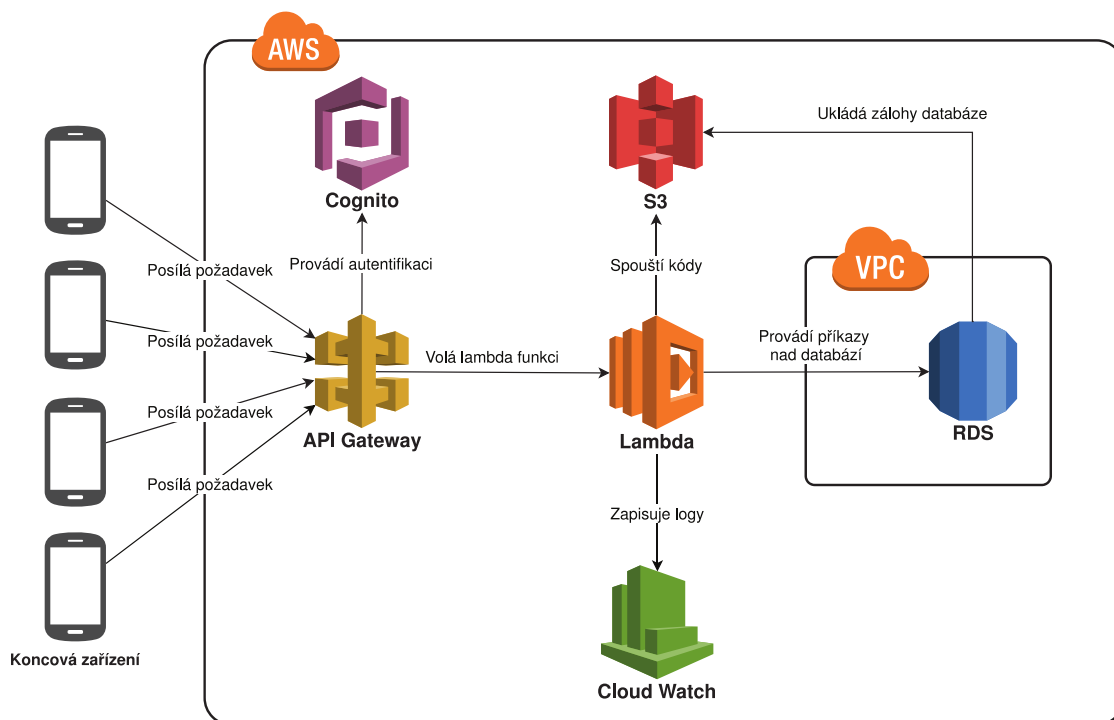
V rámci aplikace nesmíme opomenout dodržování ochrany osobních údajů. Jsou to určitá pravidla vydaná v nařízení Evropské unie. Tato pravidla se však na naši aplikaci nevztahují, protože neukládáme žádná citlivá data. Jediným údajem je emailová adresa uživatelů, ke které nemáme informaci, o jakou osobu se konkrétně jedná, a tudíž není splněna podmínka možnosti identifikace konkrétní fyzické osoby.

Důležitým kritériem celé aplikace je také objem posílaných dat v rámci komunikace mezi Android aplikací a serverem. Jde o to, aby byla data posílána co nejúsporněji. To je zajištěno pomocí komprimace části „body“ v HTTP požadavku. Nastavení této úpravy je provedeno pomocí údajů v hlavičce HTTP požadavku. Konkrétně jde o atribut „Accept-Encoding“ a hodnotu „gzip“, která určuje způsob komprimace. Dále je nutné povolit tuto proceduru na API Gateway. Po zakomponování procesu komprimace se objem přenášených dat snížil o více než polovinu původní velikosti. Pro představu, kolik dat přeneseme, si uvedeme na jednom volejbalovém utkání. Počítáme, že zápas má 5 setů, v každém setu se odehraje cca 50 výměn a v každé výměně může dojít k cca 10 změnám ve statistikách. To je dohromady 2500 požadavků za zápas. Tyto orientační údaje jsou za předpokladu velice vyrovnaného utkání (jde o přibližně horní hranici počtu požadavků). Na základě monitorování komunikace mezi aplikací a serverem došlo k následujícímu zjištění: velikost přenesených dat jednoho požadavku pro obnovení stránky zápasu je v průměru 0,22 kB při odesílání a 1,23 kB při přijímání odpovědi. S těmito údaji bude během celého zápasu přeneseno přibližně 3,625 MB.

5.1 Server

Použitý server je provozován službou AWS. Ta zajišťuje škálovatelnost celého systému. AWS podle aktuálního vytížení přiřadí vytížené službě další zdroje pro plynulý chod. Např. při nadměrném využití Lambda funkce AWS spustí její další instanci a tím se předejde zahlcení. Tento počet stejných instancí jedné Lambda funkce je omezen na 1000, což je pro naše účely plně dostačující. Architektura serveru je tzv. Serverless. To znamená, že některé služby jsou aktivní pouze tehdy, když jsou využívány. Tím se sníží finanční náklady na provoz serveru a zamezí se zbytečnému běhu služeb, když nejsou potřeba. Tento přístup má však i své úskalí. Spuštění služeb, které neběží, nemusí být nejrychlejší a tím riskujeme ztrátu uživatelů z důvodu pomalého běhu aplikace. Řešení této problematiky je dále popsáno v podkapitole 5.1.3.

Pro snazší pochopení celé struktury serveru nám poslouží Obrázek 8. Jedná se o schéma zapojení použitých komponent uvnitř serveru. Následně si popíšeme fungování hlavních komponent serveru.



Obrázek 8 - Schéma propojení služeb AWS s vazbami typu klient-server

5.1.1 API

Jako první služba v pořadí po odeslání požadavku na server je API Gateway. Sem přichází všechny požadavky od koncových uživatelů. Je tedy vstupní a výstupní branou serveru a provádí zakončení HTTPS. Struktura API rozděluje požadavky na základě použité adresy URL a typu metody. Adresa URL se skládá ze základní části identifikující službu API v AWS a z koncového bodu (Endpoint), podle kterého pozná, ke kterému objektu aplikace se požadavek vztahuje. Mezi základní použité metody HTTP dotazu jsou: GET, POST, PATCH a DELETE. GET slouží k dotazování na konkrétní data, na rozdíl od metody POST, pomocí které se ukládají nová data. PATCH je primárně určen pro změny dat a DELETE je určen pro mazání. Metoda GET a DELETE nepotřebují tělo HTTP dotazu a všechny potřebné údaje obsahují v adrese URL. Dle zmíněných kritérií API rozhodne, zda daný požadavek přijme či nikoliv. V každém případě vrátí uživateli patřičnou odpověď. Na API se také provádí vstupní autorizace uživatelů. To v našem případě probíhá pomocí služby Cognito, která rozhodne, zda má uživatel oprávnění dané akce provádět. Další autorizace na základě role uživatele se kontroluje v implementaci Lambda funkce před zpracováním konkrétní operace (podrobněji v kapitole 5.1.2). Výhodou použití API Gateway je možnost vytváření různých verzí. Např. při zásadnějších změnách v aplikaci může být nutná i změna struktury API. Abychom však zajistili funkčnost aplikace i těm uživatelům, kteří si aplikaci neaktualizují, poslouží nám k tomu tato funkčnost. Na API také probíhá přepis (mapování) parametrů URL a těla požadavku do upraveného požadavku, který se dál přepoše do Lambda funkce. Konkrétní proces mapování popisuje Obrázek 9.


```

{
  "endpoint": "assignment",
  "method": "$context.httpMethod",
  "body": $input.json('$'),
  "headers": {
    #foreach($header in $input.params().header.keySet())
      "$header": "$util.escapeJavaScript($input.params().header.get($header))"
    #if($foreach.hasNext),#end
  }#end
},
"query": {
  #foreach($queryParam in $input.params().queryString.keySet())
    "$queryParam":
"$util.escapeJavaScript($input.params().queryString.get($queryParam))"
  #if($foreach.hasNext),#end
}#end
}
}

```

Obrázek 9 - Mapování požadavků na API

5.1.2 Správa uživatelů a autorizace

Další základní komponentou serveru je služba Cognito zaštiťující uživatelské účty. Ta byla vytvořena přímo pro tyto účely. Cognito disponuje místem pro ukládání registrovaných uživatelů v aplikaci. Služba umožňuje širokou správu uživatelských účtů včetně jejich vytváření, blokování či mazání. Nastavují se zde podoby zpráv s různými informacemi, které jsou zaslány uživatelům prostřednictvím SMS nebo v našem případě e-mailů. Služba disponuje širokou škálou atributů, které lze uživatelům přiřadit. V rámci našeho systému shromažďujeme pouze e-mailovou adresu, která zároveň slouží k identifikaci uživatele a k přihlášení do aplikace.

Pokud se uživatel zaregistruje, obdrží e-mail s odkazem k ověření totožnosti. Platnost tohoto odkazu je 24 hodin. Pokud se na odkaz v tomto časovém intervalu neklikne, je třeba provést proces registrace znovu.

Ověřování uživatele probíhá na základě vygenerovaného JWT tokenu¹⁵ serverem. Ten je poslán uživateli v odpovědi po úspěšném přihlášení do aplikace. Následně jsou všechny požadavky na server doplněny o tento token. Konkrétně je přidán jako hodnota atributu „Authorization“ v hlavičce HTTP dotazu. Pomocí tohoto tokenu server pozná, o kterého uživatele se jedná. Jak už úplný název JWT říká, tokeny mají formát JSON. Skládají se ze 3 základních částí, které jsou odděleny tečkou. První částí je hlavička (Header), která obsahuje typ tokenu (v našem případě JWT) a typ hashovacího algoritmu. Druhá část obsahuje základní informace vztahující se k danému spojení či uživateli. Tato část se nazývá Payload. Poslední třetí částí je podpis (Signature), sloužící k ověření pravosti dat. Ten je vygenerovaný službou AWS. Jedním z důležitých atributů tokenu je jeho datum expirace. To je v rámci této aplikace nastaveno na 60 minut. Na Obrázek 10 je znázorněna struktura konkrétního JWT tokenu používaného v této aplikaci. Na levé straně je samotný token a na pravé straně je dekodovaný obsah tokenu.

¹⁵ Z anglického JSON Web Token. Jedná se o soubor znaků, který obsahuje údaje o přihlášení uživatele.

```

eyJraWQiOiIybkI2MmJQYnhpMGZnZiJ
wUG5INjByME9FeDhQZ2Z6bUVNN2o2MH
BjSjVFPsImFsZyI6IjTmU2In0.eyJzdWIiO
ilzNjc4ZmJiZC11NTmLTQxMzItODY1Ny1lZ
DRmNjFhNTAyMzYiLCJhdWQiOiI2NHYxa
DFpZDR1ZmRzYTlscDgyaW04cmNzNSIsIm
VtYWlsX3ZlcmlmWVkiJp0cnVILClmVudF9
pZCI6ImUwODc2ZTFkLTRjNGItMTFIOC05
YzgwLTI5ZWU1ODEzZkNyIsInRva2VuX
3VzZSI6Imk1YXV0aF90aW1lIjoxNTI1M
Dc0OTQxLCJpc3MiOiJodHRwczpcL1wvY29
nbml0by1pZHAudXMtZWZdC0xLmFtYXpv
bmF3cy5jb21cL3VzLWVhc3QtMV83OXM4
UEljaDgiLCJjb2duaXRvOnVzZXJ1YXV1Ijoi
MzY3OGZiYmQtZTUwZi00MTMyLTg2NTct
ZWQ0ZjYxYTUwMjIiwiZXhwIjoxNTI1M
Dc0NTQxLCJpYXQoIjE1MjUwNzQ5NDk5I
mVtYWlsIjoic3RlbWJlcmd2QGdtYWlsLmNv
bSJ9.Vhv9bsClvPdjSrTGWgtLk6z8SQzpevV
LFkZnWAS0hob1Rb5STCMpUuiIkHFEbsGa
yDfklGupbVkb5HLV72fwNvGWZJ6rPqUsPR
MIDJ3NBpD1Qx7YZ5TYrydEcogF_2q5qCB
YKVSNeAAclE1_aUaTG1zfmJehw9gB3NnT
bGYZb8x9xxDC7Jlhu8bz3hNApucDq1IOXZ
AYIxI2dsUaQukoonSCMfCL_9Ff3DyUQQN
OdY132j-GeO1n0MBftvabpxub936ce-g-
czkHVXqTvqWy07bx_B_GJf0gmSfUUrjRC
Wd9qcOZgEaPvDtvvs17PWM0AFUAKHeake
ktozu2g
{
  "kid": "2nB62bPbxi0fMfRpPnH60r0OEx8Pgfz
mEM7j60pcJ5E=",
  "alg": "RS256"
}
{
  "sub": "3678fbbd-e50f-4132-8657-
ed4f61a50236",
  "aud": "64v1h1id4ufdsa9lp82im8rcq5",
  "email_verified": true,
  "event_id": "e0876e1d-4c4b-11e8-9c80-
29ee581376d7",
  "token_use": "id",
  "auth_time": 1525074941,
  "iss": "https://cognito-idp.us-east-
1.amazonaws.com/us-east-1_79s8PIch8",
  "cognito:username": "3678fbbd-e50f-4132-
8657-ed4f61a50236",
  "exp": 1525078541,
  "iat": 1525074941,
  "email": "example@gmail.com"
}
RSASHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  [secret]
)

```

Obrázek 10 - Ukázka JWT tokenu

Na serveru v Lambda funkci probíhá kontrola prováděných operací uživatelem na základě jeho nejvyšší dostupné role týmu. Pokud uživatel nemá dostatečná oprávnění, server akci neprovede a vrátí uživateli patřičnou hlášku, že nemá dostatečná práva.

5.1.3 Lambda funkce

Po průchodu API a ověření totožnosti se dostáváme k dalšímu segmentu serveru Lambda. Zde se spouští naše kódy, které zpracovávají požadavky uživatelů. Tyto kódy jsou uloženy v další službě AWS a to v úložišti S3. Jelikož lambda funkce patří k těm částem serveru, které jsou tzv. Serverless, musíme zde řešit problém pomalého spuštění služby. Pokud tedy přijde na Lambda funkci požadavek, AWS ji spustí a následně požadavek zpracuje. V případě že Lambda funkce je již spuštěna, proběhne rovnou zpracování požadavku. Rychlost zpracování je pak tedy podstatně rychlejší.

V prvotní části vývoje aplikace odpovídala každému koncovému bodu API (Endpointu) jedna Lambda funkce. Tím byly služby jednoznačně rozděleny do jednotlivých funkcí. Každé funkci byl pak přidělen ve zdrojovém kódu tzv. Handler, pomocí kterého se požadavky zpracovávaly. Bohužel rychlost spuštění jednotlivých Lambda funkcí byla přibližně 12 sekund,

což mohlo mít negativní vliv na uživatele. Doba běhu funkcí přitom není neomezená. Přibližně po 25 minutách neaktivity se funkce ukončí. Tento časový údaj se může měnit podle aktuálního vytížení zdrojů AWS ostatními uživateli. Z těchto důvodů bylo potřeba zajistit, aby spuštěné funkce byly stále aktivní. To se dalo zajistit pravidelným dotazováním na danou funkci. Ovšem při velkém množství Lambda funkcí by se jednalo o příliš mnoho dotazů. Z důvodů omezení zdrojů AWS v rámci používání licence zdarma je počet spuštění Lambda funkcí omezený. Proto jsem zapouzdřil všechny funkce do jedné, a tu pravidelným dotazováním udržuji aktivní. Toto udržování zajišťuje služba Cloud Watch, která pomocí definovaného pravidla pravidelně posílá dotazy

na funkci. Sloučením funkcí do jedné jsme však přišli o přerozdělení dotazů do jednotlivých Handlerů. To se nyní provádí až uvnitř hlavního Handleru, který zpracovává všechny požadavky a dle údajů posílaných z API je třídí do podřadných Handlerů.

Lambda má umožněn přístup do databáze. Údaje k připojení jsou uloženy v konfiguraci Lambda funkce v AWS. Funkce si je při svém spuštění načte a uloží. Samotný kód tedy žádné přihlašovací údaje k databázi neobsahuje. V případě zcizení kódů nebudou data ohrožena. Samotná práce s daty je prováděna pomocí technologie JPA popsané dříve.

Po zpracování požadavku funkce vrátí odpověď v konkrétní zadané podobě, kterou očekává i Android aplikace. Odpověď je opět ve formátu JSON a obsahuje atributy „status“, „message“ a „body“. První atribut popisuje výsledek zpracování požadavku a ve druhém je případná zpráva k čemu během zpracování došlo. Poslední částí odpovědi je pole, v které mohou být uložena požadovaná data.

Každé spuštění funkce je zaznamenáváno službou Cloud Watch. Mezi základní zaznamenané údaje se řadí začátek a konec dotazování a také souhrnné informace o celém zpracování požadavku. Do Cloud Watch si lze logovat i vlastní informace ze spuštěného kódu pomocí třídy „Context“ poskytované přímo službou AWS pro podobné účely.

5.1.4 Databáze

Databáze je spravována službou RDS využívající objektově-relační databázový systém PostgreSQL. Jde o jeden z nejpoužívanějších databázových systémů. Instance použité databáze je umístěna ve virtuálním úložišti VPC. Databáze je na serveru pravidelně zálohována a tím je minimalizována případná ztráta dat uživatelů. Hesla nejsou uložena přímo v naší databázi, neboť o uživatelské účty se stará služba Cognito. Pro práci s databází se využívá program pgAdmin 4 v3.0¹⁶.

V rámci databáze se verzují jednotlivé záznamy v určitých tabulkách. To slouží k podrobnějšímu náhledu na data, která uživatel zadá nebo ke zpětnému ladění (např. kdo jakou změnu provedl atd.). Konkrétní princip verzování je následující. Každá tabulka, která je verzována má navíc 1 sloupec, který identifikuje konkrétní položku (objekt) a další 3 sloupce, které popisují bližší informace o dané změně.

Prvním sloupcem je identifikátor objektu (v databázi „id_název_objektu“), který se na rozdíl od základního identifikátoru jednotlivých řádků (v databázi „id“) kopíruje. Pomocí tohoto kopírujícího identifikátoru lze např. zobrazit všechny změny daného objektu (týmu, hráče atd.).

Další sloupec (v databázi „audit_record“) vypovídá, zda je konkrétní řádka záznamu aktuální či nikoliv. Aktuální záznam je označen číslem 0 a starý záznam pomocí 1. Aktuální záznam objektu je vždy jen jeden v rámci záznamů stejného objektu. To je zajištěno používáním transakcí při práci s databází. Nemůže se tedy stát, že by se v databázi vyskytly dva záznamy s „audit_record“ nastaveným na 0, nebo naopak žádný záznam nastavený na 0. Tato poslední

¹⁶ <https://www.pgadmin.org/>

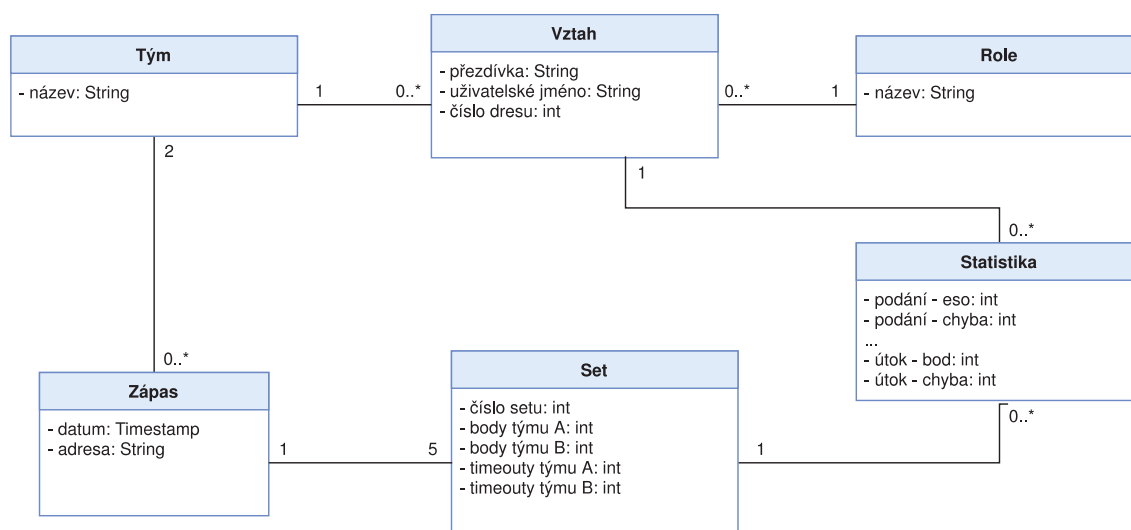
možnost nastává pouze tehdy, když je daný objekt uživatelem smazán. V ten okamžik jsou všechny záznamy objektu označeny jako staré.

Předposlední sloupec (v databázi „audit_date“) obsahuje časovou značku (datum a čas), která říká, kdy ke změně došlo. Podle toho lze v budoucnu dohledat potřebné informace. Nastavení datumu probíhá krátce před tím, než je řádek fyzicky uložen do databáze, aby byl co nejaktuálnější.

Posledním sloupcem je název uživatele, který změnu provedl (v databázi „audit_user“). Konkrétně se jedná o e-mail uživatele, pod kterým je zároveň zaregistrován v aplikaci.

Záznamy databáze jsou díky tomuto verzování neměnné objekty (angl. Immutable objects). To znamená, že jakákoliv změna v databázi je řešena novým záznamem. Záznamy se tedy nemažou nebo nemění. Jedinou výjimkou je atribut „audit_record“, který mění svoji hodnotu z 0 na 1.

V této části si popíšeme jednotlivé entity tvořící strukturu naší databáze. Na následujícím obrázku (Obrázek 11) se nachází doménový model aplikace.



Obrázek 11 - Doménový model

Entita Role louží jako seznam dostupných rolí v aplikaci. Celkově existují 3 základní role: administrátor, trenér a hráč. Vztah přiděluje uživatelům role v rámci nějakého týmu. Popisuje tedy vztah daného uživatele ke konkrétnímu týmu. Tato entita obsahuje atribut uživatelské jméno, což je zároveň e-mailová adresa případného uživatele. Tým je entita vytvořena přihlášeným uživatelem reprezentující volejbalový tým. Další entita je Zápás. Ta zahrnuje volejbalový zápas dvou týmů proti sobě. Počet týmů v zápase je vždy dva. Každý zápas má právě 5 setů. Set přísluší konkrétnímu zápasu. Zároveň set udržuje stav skóre a počet timeoutů obou týmů. Poslední entitou je statistika, která reprezentuje statistiku konkrétního hráče v konkrétním setu. Mezi statistiky patří činnosti spojené s podáním, příjmem, náhrou, útokem, blokem, vybíráním míče v poli a zpracováním míče.

5.2 Android aplikace

Samotná Android aplikace je vyvíjena pro verze 5.0 Lollipop (API 21) a vyšší. Tím by měla být zaručena dostupnost 84,7 % uživatelům používající zařízení na platformě Android (data jsou platná k 16. 5. 2018). Tato verze 5.0 přišla na trh v roce 2014. Bohužel v současnosti není oficiálně podporována a tím postrádá i bezpečnostní aktualizace. Avšak stále tuto verzi používá přes 20 % uživatelů, což je nezanedbatelný podíl, který nesmíme opomenout. Některým těmto zařízením je umožněn upgrade na verzi 6.0 Marshmallow, která je stále podporována. V následujících kapitolách jsou v některých případech uvedeny ukázky konkrétních GUI. Ty mají několik svých typických rysů, které se objevují skrze celou aplikaci (např. styly základních tlačítek mají stále stejný vzhled).

V případě že v aplikaci dojde k chybě, jsou tyto chyby automaticky odesílány na server, kde jsou ukládány. Tím se usnadní opravy chyb v aplikaci a práce uživatelů, kteří nemusí složitě popisovat, při jaké činnosti jim aplikace nefungovala správně.

5.2.1 HTTP klient

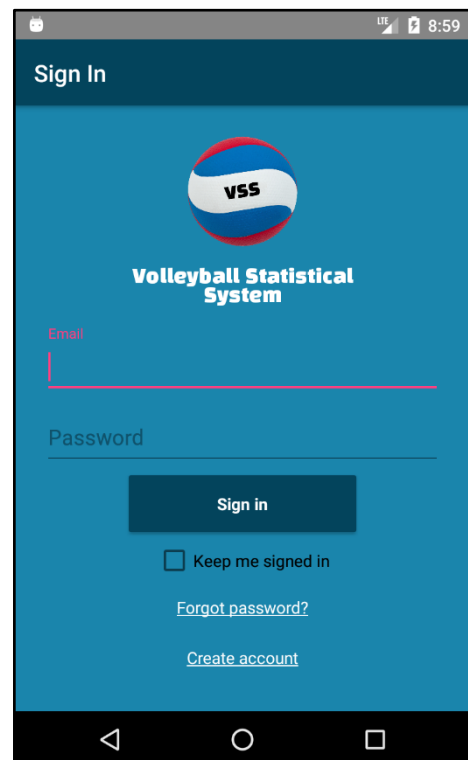
Aplikace využívá ke komunikaci se serverem klienta OkHttp¹⁷. Jde o technologii z roku 2013, která je v současnosti velice efektivní v porovnání s konkurencí. Konkrétně je použita verze 3.9.1. Tento klient podporuje protokoly HTTP/2¹⁸ a nově i SPDY, což je experimentální síťový protokol vyvíjený pod záštitou Googlu. OkHttp podporuje značné množství funkcí spojených se síťovými nástroji a je neustále vyvíjen a vylepšován.

5.2.2 Přihlášení, registrace

Po zapnutí aplikace je uživatel přesměrován na stránku pro přihlášení (Obrázek 12). Přihlášení probíhá pomocí e-mailové adresy a hesla. Pozitivním prvkem na přihlašování je možnost zapamatování hesla. To značně ulehčí práci a lidé se rychleji dostanou do aplikace. V případě, že uživatel nechce, aby byly údaje i nadále uloženy v zařízení, stačí při opětovném přihlášení odškrtnout zaškrťovací políčko, které slouží k zapamatování přihlašovacích údajů. Heslo je v zařízení uloženo v zašifrované podobě.

Pokud uživatel účet nemá, je k dispozici tlačítko pro vytvoření účtu. Na následující stránce uživatel vyplní svoji e-mailovou adresu a zadá dvakrát stejné heslo, které musí splňovat bezpečnostní kritéria. Po úspěšném odeslání požadavku registrace na server dojde k odeslání potvrzovacího e-mailu ze strany serveru na uživatelský e-mail. V potvrzovacím e-mailu se nachází odkaz, pomocí kterého uživatel svoji registraci potvrdí. Platnost tohoto odkazu je 24 hodin.

Důležitou funkcí je možnost obnovy hesla v případě že jej uživatel zapomene. Možnost zahájení



Obrázek 12 - Obrazovka pro přihlášení

¹⁷ <http://square.github.io/okhttp/>

¹⁸ Jde o druhou hlavní verzi protokolu HTTP

tohoto procesu se nachází na stránce přihlášení. Prvním krokem je napsání e-mailové adresy, ke které neznáme heslo. Po potvrzení a schválení serverem se na e-mailový účet odešle ověřovací kód a uživatel je v aplikaci přeměrován na další stránku. Na té je potřeba vyplnit ověřovací kód a zadat dvakrát nové heslo splňující opět bezpečnostní kritéria. Potvrzením se nastaví nové heslo a uživatel se již může znovu přihlásit. Tento proces obnovy hesla vyžaduje, aby byl v požadavku na server odeslán také tzv. Access token. Ten je vygenerován společně s hlavním JWT tokenem k ověřování totožnosti. Bez Access tokenu by obnova hesla nefungovala. Práci s tokeny zajišťuje sama aplikace, nikoliv uživatel.

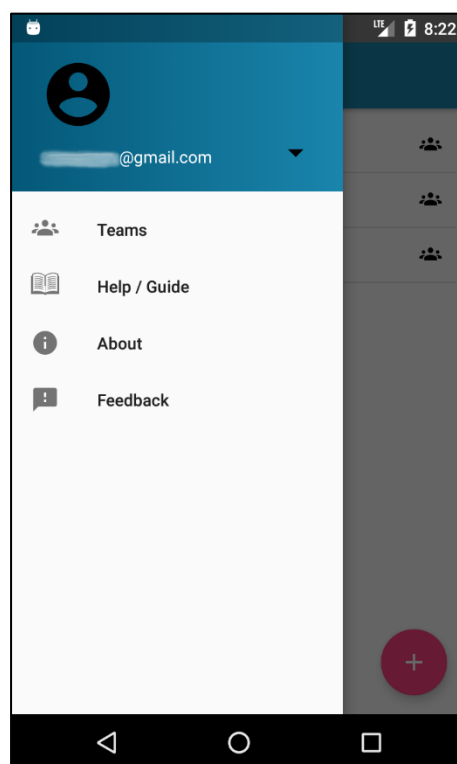
Pokud přihlášenému uživateli expiruje spojení, je odkázán na přihlašovací stránku, kde se musí znovu přihlásit a tím spojení obnoví.

5.2.3 Základní menu

Aplikace disponuje základním menu skrytým v levém postranním panelu zobrazeném na Obrázek 13. To má položky jako jsou týmy, nápověda, informace o aplikaci a zpětná vazba. V horní části menu je informace o přihlášeném uživateli. Kliknutím na tuto část uživatele se zobrazí jiné menu, v kterém jsou možnosti spojené výhradně s účtem. Jsou to možnosti odhlášení a změna hesla.

V nápovědě hlavního menu se uživatel dozví základní věci ohledně používání aplikace (např. jak propojit člena týmu s existujícím uživatelem). V této části se uživatel také dozví podrobnosti o tom, jak správně zadávat statistiky během utkání. V sekci informací o aplikaci se uživatel dozví aktuálně používanou verzi, autora a další informace spojené s vývojem.

Důležitým prvkem hlavního menu je možnost odeslání zpětné vazby. Zde je umožněno uživateli projevit svůj názor na aplikaci a vznést případné výtky či možná vylepšení. To celé je za účelem zdokonalení aplikace v budoucnu. Všechny tyto poznatky od uživatelů se ukládají na server do databáze, kde si je lze zobrazit. Maximální velikost jedné zprávy je 250 znaků.



Obrázek 13 - Základní menu

5.2.4 Týmy

V rámci aplikace si každý přihlášený uživatel může vytvářet své týmy. To se provede kliknutím na tlačítko plus v pravém dolním rohu displeje. Zde je potřeba vyplnit název týmu a jméno, pod kterým se bude uživatel v týmu vyskytovat. Ten, kdo tým vytvoří je automaticky jeho administrátorem. Název týmu musí být unikátní napříč celou aplikací. Pokud tedy někdo zvolí již existující jméno týmu, bude na to upozorněn příslušnou hláškou. V této hlášce se zobrazí e-mailový účet jednoho z administrátorů daného týmu pro jeho případné kontaktování. Pokud vše proběhne v pořádku, tým se vytvoří a uživatel ho uvidí na své domovské stránce v sekci s týmy. Tyto týmy jsou řazeny abecedně.

5.2.5 Uživatelské role

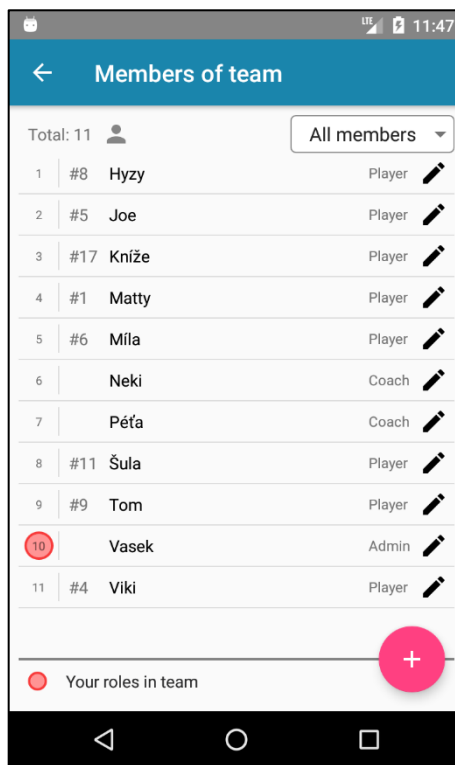
Každý uživatel má k danému týmu nějaký vztah. Tento vztah je vyjádřen konkrétní rolí. Na základě té se aplikace přizpůsobí. Konkrétní postup je takový, že jakmile uživatel zobrazí nějaký tým, spolu s daty o týmu se ze serveru načtou i jeho role v daném týmu. Podle těchto rolí se následně upraví grafické rozhraní. Například uživateli s rolí hráč se nezobrazí tlačítka pro úpravy či mazání. Zabránění neoprávněným operacím je tedy zamezeno dvojím způsobem, jak v Android aplikaci, tak na serveru.

5.2.6 Členové týmu

Zobrazením členů týmu se načtou členové se všemi rolami vztahující se k týmu (Obrázek 14). Tyto členy lze filtrovat podle jejich role a zobrazit si je tak přehledněji. Všichni členové jsou seřazeni abecedně dle jména. V seznamu členů jsou role vztahující se k aktuálně přihlášenému uživateli znázorněny červeným kolečkem, a to v místě číslování jednotlivých záznamů.

Vytvoření člena se provede tlačítkem plus v pravém dolním rohu displeje. Zde se nastaví jméno a role člena. Pokud je jako role zvolen hráč, zobrazí se kolonka pro zadání čísla dresu. Toto políčko je však nepovinné, protože na velkém množství volejbalových zápasů se smí hrát bez dresů s čísly. Potvrzením tlačítka se nový člen vytvoří. Jména členů jsou v rámci jednoho týmu a stejné role unikátní. Podobně to je i s čísly dresů, ty musí být v týmu také unikátní. Pokud některé z těchto omezení není dodrženo, je na to uživatel patřičně upozorněn.

Pro smazání je zapotřebí zobrazit si daného člena a poté kliknout na tlačítko pro smazání. Tuto operaci však může provést pouze administrátor týmu. V rámci procesu mazání je zavedeno určité omezení. A to v případě, že jde o mazání administrátora. Pokud by administrátor mazal sám sebe, systém mu to nedovolí, pokud by byl v daném týmu jediným administrátorem. To z toho důvodu, že tým by pak neměl kdo spravovat a zůstal by bez administrátora.



Obrázek 14 - Členové týmu

5.2.7 Propojení uživatele s členem týmu

Uživatel, se rozumí člověk, který je zaregistrovaný v aplikaci pod svým e-mailovým účtem. Člen týmu je instance vytvořená administrátorem. Pro propojení uživatele s členem týmu je zapotřebí následující postup. Nejprve je potřeba mít roli administrátora v daném týmu. Následně je potřeba zobrazit detaily člena, kterého chceme propojit. V podrobnostech člena je kolonka e-mail. Do té je třeba vyplnit e-mailovou adresu uživatele, kterého s tímto členem chceme propojit. Po úspěšné editaci se již přihlášenému uživateli zobrazí tým, v kterém se daný člen nachází.

Odpojení uživatele od týmu probíhá obdobným způsobem. V podrobnostech člena stačí smazat uživatelský e-mailový účet a tím proběhne odpojení od týmu s danou rolí. Člen však stále zůstává i nadále v týmu. Pouze není propojen s některým uživatelem.

5.2.8 Zápas

U každého týmu lze vytvářet a spravovat zápasy. Do této části aplikace má oprávnění zasahovat i trenér týmu. Spolu s administrátorem smí vytvářet a upravovat zápasy. Zápasy nelze smazat.

Vytvoření zápasu lze provést kliknutím na tlačítko plus v pravém dolním rohu. Následuje výběr datumu, času, soupeře a případně místa konání. Výběr datumu a času je proveden interaktivním zobrazením kalendáře a hodin. Pro zvolení soupeře je k dispozici možnost vyhledávání. To se aktivuje kliknutím na ikonu lupy. Následným psaním se začnou vyhledávat názvy týmů dle největší shody. Kliknutím na nějaký výsledek dojde k zvolení daného týmu jako soupeře. Pokud hledaný tým neexistuje, vyplní uživatel název soupeře ručně. Založením zápasu s existujícím týmem dojde k propojení k zápasu i ze strany druhého týmu.

5.2.9 Statistika

Do zadávání statistik se dostaneme přes tlačítko symbolizující načtení zápasu. V následujícím okně se zobrazí všechny potřebné prvky pro zadávání (Obrázek 15). Těmito prvky jsou názvy obou týmů, skóre za daný set, tlačítka pro změny setů a skóre, seznamy hráčů obou týmů, tlačítka pro zadávání statistik rozdělených dle činnosti a tlačítko zpět. Hráči jsou seřazeni podle jejich čísel dresů. V případě, že hráč číslo nemá, jsou mu vygenerovány iniciály, které se zobrazí v seznamu místo čísla. Uživatelé bez čísla jsou v seznamu uvedeni až po hráčích s čísly. Aktuální číslo probíhajícího setu je uvedeno na horní liště a k jejich přepínání slouží tlačítka na krajích obrazovky. Konkrétní rozvržení těchto komponent znázorňuje uvedený obrázek.

Pro zadání statistiky je potřeba označit nějakého hráče a poté kliknout na danou činnost. Podoba možností pro zadávání jednotlivých činností je inspirována dokumentem od společnosti Data Project [10], který se používá např. pro českou extraligu. To z toho důvodu, aby výsledné statistiky byly porovnatelné s těmi profesionálními. Pro vrácení změny slouží tlačítko zpět. Vrácení změny se vždy vztahuje k jednotlivým hráčům zvlášť. Úroveň vrácení změn je pouze jedna operace. Při opakovaném stisknutí tlačítka zpět nastane původní situace. Označený prvek je vždy jen jeden nebo žádný. Pokud tedy zadáváme statistiky hráče a klikneme na název týmu, původní políčko se odznačí.

Skóre se mění tlačítky plus a minus přímo pod ním. Pro změnu je však zapotřebí mít označený nějaký tým, jinak tlačítko nic nedělá. Označení týmu se provede kliknutím na název týmu. Pro skóre tlačítko zpět nefunguje z důvodu možnosti, že skóre lze i snížit.



Obrázek 15 - Zadávání statistik ze zápasu

Zobrazení výsledných statistik (Obrázek 16) zahrnuje výsledky jednotlivých setů a 4 základní činnosti (podání, příjem, útok a blok). Tyto údaje jsou uvedeny ve stejném rozvržení, jako na stránkách Českého volejbalového svazu. To proto, abychom lidem usnadnili práci při nahlédnutí na tyto statistiky. Hráči jsou v seznamu reprezentováni jednotlivými řádky. V rámci těchto údajů jsou zde statistiky nejen individuální, ale i za celý tým. Uvedený příklad výsledků je zobrazen na šířku.

Match summary														
Kladno volejbal cz				VS		Lokomotiva Beroun - muži								
Set:		1	2	3	4	5								
		31:7	25:21	0:0	0:0	0:0								
Statistics of: Kladno volejbal cz														
#	Name	Serve			Reception				Attack				Block Pts	
		Tot	Err	Ace	Tot	Err	Pos%	Per%	Tot	Err	Blo	Win		Pts%
17	Kniže	0	0	0	0	0	%	%	0	0	0	0	%	0
1	Matty	6	1	2	0	0	%	%	1	0	0	1	100%	1
6	Mila	3	0	2	7	1	85%	14%	1	0	0	1	100%	0
11	Šula	0	0	0	5	0	80%	20%	5	0	2	2	40%	0
9	Tom	1	0	1	0	0	%	%	2	1	1	0	0%	4
4	Viki	1	0	1	7	0	71%	42%	9	1	1	5	55%	0
All team		14	1	8	26	2	80%	26%	28	2	4	18	64%	5

Obrázek 16 - Výsledné statistiky zápasu

6 Testování

Jako většina produktů je i tento systém třeba otestovat. Testování je při vývoji velice důležitým prvkem, proto na něj musíme klást důraz. Cílem testování je odhalit případné nedostatky, při používání aplikace. Testuje se jak samotné GUI, tak i funkčnost a složitost systému. Výsledky jsou umístěny v poslední kapitole testování 6.4.

6.1 Cílová skupina

Aplikace je určena převážně pro osoby mající zkušenosti s ovládáním chytrých telefonů či tabletů s operačním systémem Android. Dále je potřeba, aby měly alespoň základní znalosti o běhu volejbalových utkání a vlastních pravidel volejbalu.

6.2 Metoda

V této kapitole si popíšeme použité metody, pomocí kterých byla aplikace otestována. Nejprve si popíšeme Heuristickou evaluaci a poté testování na uživateli.

6.2.1 Heuristická evaluace

Tato metoda spadá do skupiny testování bez uživatele. Konkrétně jde se o heuristickou analýzu Jakoba Nielsena z roku 1994 [11]. Tester provede daný úkol a pozoruje, zda v průchodu nebyla porušena některá pravidla (heuristiky), které by mělo uživatelské rozhraní aplikace splňovat. Tuto testovací metodu definují následující kritéria:

- **H1: Viditelnost stavu systému**
 - systém by měl dát uživateli vědět, co se právě odehrává
- **H2: Propojení systému a reálného světa**
 - komunikace systému s uživatelem by měla být uživateli srozumitelná (informace musí dávat uživateli smysl)
- **H3: Uživatelská kontrola a svoboda**
 - možnost vrátit se uživateli o krok zpět (do předchozího stavu)
- **H4: Konzistence a standard**
 - vzhled a ovládání by mělo být konzistentní
 - dodržení obecných zásad (uživatel by neměl být nucen přemýšlet)
- **H5: Prevence chyb**
 - schopnost systému zabránit chybovým stavům (nechtěným akcím)
- **H6: Rozpoznávání místo zapamatování**
 - prováděné operace by měl uživatel dělat intuitivně a neměl by být nucen si je pamatovat
 - podstatné informace by měly být na první pohled viditelné
- **H7: Flexibilní a efektivní použití**
 - schopnost systému poskytovat prvky zrychlující provedení akce

- **H8: Estetický a minimalistický design**
 - systém by neměl zobrazovat nepotřebné informace, které mohou být pro uživatele matoucí
- **H9: Pomoc uživatelům pochopit a vzpamatovat se z chyb**
 - chybové hlášky systému by měly být jasné, srozumitelné a měly by navrhnout řešení
- **H10: Náповěda a dokumentace**
 - všechny informace by měly být snadno vyhledatelné
 - náповěda (návod) by měla obsahovat postupy v jednotlivých krocích

6.2.2 Testování v simulované laboratoři

V této části testování se pracuje s účastníky, kteří spadají do cílové skupiny. Ti jsou testováni postupně za stejných podmínek, tak aby výsledky testu byly co nejméně ovlivněny. Účastník je upozorněn, že se testuje aplikace nikoliv on sám. Dále je informován, že všechny výsledky testování jsou anonymní a nebudou nijak zneužity. Během vypracovávání úkolů je dobré, aby účastník tzv. myslel nahlas. Jde o to, aby během vypracovávání jednotlivých úkolů říkal nahlas, co si myslí. Na základě těchto dat se pak lépe zaznamenávají myšlenkové pochody účastníků.

6.3 Testování

V této kapitole si popíšeme způsob aplikace metod testování popsanych v předchozí kapitole. Prvním způsobem testování je Heuristická evaluace a druhým testování na uživatelích.

Heuristickou evaluací budeme vyhodnocovat celou aplikaci. Během používání budeme zjišťovat, zda neporušíme některou z 10 heuristik, které jsou již definovány v kapitole 6.2.1.

K testování na uživatelích bylo přizváno 5 účastníků (2 ženy a 3 muži), kteří s aplikací nemají žádné zkušenosti. Testování probíhalo v uzavřené místnosti a bylo použito jedno stejné zařízení, na kterém test probíhal. Konkrétně Samsung Galaxy S5 Neo s verzí operačního systému 6.1 Marshmallow. Účastníkům byly zadány k vypracování stejné úkoly. Tyto úkoly byly rozděleny do tří kategorií. Úkoly typu 1 byly jednoduché, úkoly typu 2 byly složitější. Během těchto úkolů je vyžadována větší soustředěnost účastníka. Posledním typem úkolu 3 byly volitelné. Tyto obtížnosti však účastníci neznali. Nejprve se vypracovávaly jednodušší úkoly a postupně se přidávaly složitější. V závěru se testovaly volitelné úkoly. Veškeré počínání uživatelů během testování bylo zaznamenáváno a posléze vyhodnocováno. Na základě těchto dat byl vytvořen seznam jednotlivých nálezů v aplikaci. Úkoly k vypracování s danou kategorií byly následující:

- | | |
|---|---|
| 1) Zapnutí aplikace (typ 1) | 9) Vracení poslední změny (typ 2) |
| 2) Registrace nového uživatele (typ 1) | 10) Přidání bodů týmu soupeře v třetím setu (typ 2) |
| 3) Přihlášení do aplikace (typ 1) | 11) Odebrání jednoho hráče z týmu (typ 2) |
| 4) Vytvoření týmu (typ 1) | 12) Odeslání zpětné vazby (typ 2) |
| 5) Přidání hráčů do týmu (typ 1) | 13) Zjištění verze aktuální aplikace (typ 1) |
| 6) Vytvoření zápasu proti Kladnu, které již v aplikaci existuje (typ 2) | 14) Změnění hesla (typ 3) |
| 7) Zadání adresy k právě vytvořenému zápasu (typ 2) | 15) Odhlášení (typ 3) |
| 8) Zadání statistiky ze zápasu u jednoho z hráčů (typ 2) | 16) Obnovení (reset) hesla (typ 3) |
| | 17) Vypnutí aplikace (typ 3) |

6.4 Nálezy

Následující část patří shrnutí všech nálezů během celého testování. Každý nedostatek má určenou závažnost a typ porušené heuristiky. U nálezu je uveden popis problému a dále je popsáno možné řešení. Během počátečního testu se vyskytl problém s použitím zpětného tlačítka z důvodu špatné práce s jednotlivými stránkami aplikace. Následkem bylo cyklení jednotlivých stránek. Tento problém byl ihned vyřešen a další testování probíhala již bez tohoto nedostatku. Jednotlivé nálezy shrnuje Tabulka 1.

Závažnost problémů je rozdělena do tří kategorií.

- 1 (nízká) – malý problém, jde spíše o kosmetický nedostatek
- 2 (střední) – problém není kritický, ale může uživateli znepříjemnit ovládání
- 3 (vysoká) – závažný problém, který je třeba co nejdříve odstranit

Tabulka 1 - Nálezy testování

Závažnost	Porušená heuristika	Problém	Řešení
1	2	Uživatelé si nejsou jisti významem některých anglických slov (např. „Sign in“).	Přejmenování slov na známější termíny, nebo lépe přidání funkce volby jazyka.
2	7	U některých formulářů se stane, že zobrazená klávesnice překryje vyplňované kolonky.	Udělat obsah stránky rolovací, aby se přizpůsobila vysunutě klávesnici.
1	2	U detailu některých entit je tlačítko „Edit“, které je pro některé uživatele matoucí.	Lépe přejmenovat na např. „Save“ nebo „Save changes“.
1	2	Pro vyhledávání týmů nestačí psát do textového pole. Musí se kliknout na ikonu lupy.	Spustit vyhledávání při kliknutí na textové pole.

7 Závěr

Cíle bakalářské práce bylo dosaženo. Volejbalový systém, primárně určený pro zadávání statistik z volejbalového utkání, byl navržen a zrealizován. To včetně serveru a samotné Android aplikace. Během vývoje jsem přišel do styku s novými technologiemi a potýkal se různými problémy. Všechny tyto překážky jsem zvládl a vyřešil.

V první fázi práce byl proveden průzkum existujících aplikací a jejich porovnání. Následně jsem se seznámil s použitými technologiemi. Ve velké míře jsem se zabýval správným využitím služeb Amazon Web Services (AWS) a problematikou komunikace mezi serverem a aplikací. Toto spojení je řešeno pomocí protokolu HTTPS.

Grafické rozhraní aplikace bylo vyvinuto tak, aby bylo co nejvíce intuitivní. Základní vizuální prvky zachovávají svůj vzhled a styl. Tím je ovládání pro uživatele snazší. Také stav aplikace dodržuje určité zásady, díky kterým uživatel pozná, co právě aplikace dělá.

Serverová část je provozována pomocí služby AWS. Ta usnadňuje správu jednotlivých komponent serveru, jako jsou API, databáze a správa uživatelů. Serverová část zpracovávající požadavky uživatelů společně s Android aplikací jsou psány v jazyce Java. Konkrétně aplikace Android byla napsána ve vývojovém prostředí Android Studio.

V rámci poslední části práce bylo provedeno testování aplikace na uživateli. To obnášelo pozvání 5 účastníků, kteří plnili různé úkoly. Během těchto úkolů byla práce účastníků zaznamenávána a posléze vyhodnocena. Výsledky testování neodhalily žádný závažný problém. Nalezeno bylo pouze několik menších nedostatků. Mezi tyto problémy spadá chybějící možnost volby jazyka, horší zobrazení klávesnice spolu s textovými poli, nebo nejasná funkce tlačítka „Edit“.

Aplikace je vyvinuta tak, aby umožňovala řadu budoucích rozšíření. Mezi možná vylepšení patří následující funkcionality. Pro větší komfort uživatelů umožnit vícejazyčnost rozhraní aplikace. Navržen lepšího přizpůsobení grafického rozhraní pro tablety, včetně otáčení displeje v částech aplikace, kde to v současnosti není možné. Dále by mohla být zakomponována funkcionality tréninků. Pomocí té by bylo možné se domlouvat na tréninky a zároveň vést docházku. Dalším vylepšením by mohla být možnost přihlášení pomocí některé existující autority, jako např. Facebook nebo Google. Používání Google účtu k přihlášení by usnadnilo práci velkému množství lidí, protože pro používání zařízení s operačním systémem Android je třeba být přihlášený ke Google účtu. V neposlední řadě by bylo výhodné nahrát aplikaci na službu Google Play, pomocí které by se dosáhlo větší rozšířenosti.

8 Literatura

1. **Google, Open Handset Alliance.** *android.com*. [Online] 23. září 2008.
<https://developer.android.com/about/index.html>.
2. **Williams, Nick.** *Professional Java for Web Applications*. John Wiley & Sons, Incorporated, 2014. 9781118656518.
3. **Sandoval, Jose.** *Master Core REST Concepts and Create RESTful Web Services in Java*. Packt Publishing, 2009. 9781847196460.
4. **Sriparasa, Sai.** *JavaScript and JSON Essentials*. Packt Publishing, 2013. 9781783286041.
5. **Pillitu, Maurizio Turatti and Maurizio.** *Instant Apache Maven Starter*. Packt Publishing, 2013. 9781782167600.
6. **AWS Documentation.** *Amazon Web Services*. [Online] Amazon, Květen 2018.
<https://aws.amazon.com/documentation/>.
7. **Golden, Bernard.** *Amazon Web Services For Dummies*. Wiley, 2013. 9781118652268.
8. **Beach, Brian.** *Pro Powershell for Amazon Web Services*. Apress L. P., 2014. 9781430264521.
9. **Somasundaram, Ravishankar.** *Git*. Packt Publishing, 2013. 9781849517539.
10. **Project, Data.** *Software for the scouting and analysis of volleyball matches*. Via Dell'Elettricista 10, 40138 Bologna, Italy : Data Project, 2015.
11. **Nielsen, Jakob.** *Usability Engineering*. Morgan Kaufmann; 1 edition, 1993. 978-0125184069.

9 Přílohy

- Android
 - zdrojové kódy
 - instalační soubor APK
- Server
 - zdrojové kódy
 - vygenerovaný soubor JAR
- Dokumenty
 - bakalářská práce (PDF)
 - bakalářská práce (DOCX)
- Diagramy