

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Mobilní a webová aplikace pro střelecké soutěže



2021

Vedoucí práce:
Mgr. Petr Krajča, Ph.D.

Vít Řehák

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Vít Řehák
Název práce: Mobilní a webová aplikace pro střelecké soutěže
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2021
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: Mgr. Petr Krajča, Ph.D.
Počet stran: 46
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Vít Řehák
Title: Mobile and Web Application for Shooting Competitions
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2021
Study field: Applied Computer Science, full-time form
Supervisor: Mgr. Petr Krajča, Ph.D.
Page count: 46
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Práce zajišťuje administraci a vyhodnocování výsledků střeleckých soutěží pro náš klub sebeobránné střelby. Pravidla a způsob vyhodnocování je inspirován mezinárodní soutěží IPSC (International Practical Shooting Confederation) a závody typu LOS (Lidová Obranná Střelba). Práce sestává ze dvou částí. První část je webová aplikace, která obstarává administrativní úkony soutěží. Druhá část práce je aplikace pro Android. Tato aplikace dopomáhá rozhodčímu vyhodnocovat průběh soutěže a tréninků na místě konání akce.

Synopsis

Thesis provides administration and evaluation of shooting competitions and trainings for our Self-defense shooting club. Rules and methods of evaluation are inspired by IPSC (International Practical Shooting Confederation) and LOS (Self-defense shooting) competitions. Thesis consists of two major parts. First part is web application, which fulfills administrative tasks. Second part consists of application for Android mobile phone. This application helps referee to evaluate the progress of competition or training.

Klíčová slova: Střelecká soutěž; Android aplikace; Webová stránka

Keywords: Sport shooting competition; Android application; Web page

Chtěl bych moc poděkovat panu Mgr. Petru Krajčovi, Ph.D. za odbornou pomoc, cenné rady a za vedení mé práce.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	8
2	Motivace	10
3	Uživatelská dokumentace	11
3.1	Webová aplikace	11
3.1.1	Vstupní stránka	11
3.1.2	Vytvoření soutěže	12
3.1.3	Detail nadcházející soutěže	14
3.1.4	Úprava a zrušení soutěže vlastníkem	15
3.1.5	Detail vyhodnocené soutěže	15
3.1.6	Zobrazení údajů o účtu	16
3.1.7	Stránka pro tisk dokumentace	17
3.2	Android aplikace	18
3.2.1	Aktivita pro přihlášení	18
3.2.2	Aktivita seznamu soutěží uživatele	18
3.2.3	Aktivita seznamu stagí	19
3.2.4	Aktivita seznamu účastníků	19
3.2.5	Skórovací aktivita pro ohodnocení účastníků	20
4	Modely databází	21
4.1	Model databáze webové aplikace	21
4.2	Tabulky databáze webové aplikace	22
4.2.1	Tabulka users	22
4.2.2	Tabulka login_tokens	22
4.2.3	Tabulka android_login_tokens	22
4.2.4	Tabulka competitions	23
4.2.5	Tabulka stages	23
4.2.6	Tabulka stage_stats	23
4.2.7	Tabulka registrations	24
4.2.8	Tabulka disqualifications	24
4.2.9	Tabulka user_bans	24
4.3	Model databáze aplikace pro Android	25
4.4	Tabulky databáze mobilní aplikace	26
4.4.1	Tabulka users	26
4.4.2	Tabulka Participants	26
4.4.3	Tabulka competitions	26
4.4.4	Tabulka stages	27
4.4.5	Tabulka stage_stats	27
4.4.6	Tabulka registrations	28

5	Programátorská dokumentace	29
5.1	Architektura práce	29
5.2	Webová část	30
5.2.1	Spring MVC	30
5.2.2	HttpSession	31
5.2.3	Zpracování HTML šablony technologií Thymeleaf	32
5.2.4	Práce s databází	32
5.2.5	Komunikace mezi serverem a klientem	32
5.2.6	Přihlašovací systém a možnost zachování přihlášení	34
5.2.7	Synchronizace soutěží s Android klientem	34
5.2.8	Struktura projektu	36
5.3	Aplikace pro Android	37
5.3.1	Knihovna Volley	37
5.3.2	Přihlašovací systém	37
5.3.3	Android ROOM	37
5.3.4	Struktura projektu Android aplikace	40
	Závěr	42
	Conclusions	43
	A Přístup k webové aplikaci	44
	Literatura	46

Seznam obrázků

1	Příklad terčů s vyznačenými zónami.	9
2	Vstupní stránka pohledem přihlášeného uživatele	12
3	Ukázka vytvořených stagí v tabulce	14
4	Detail soutěže pohledem přihlášeného uživatele	15
5	Příklad výsledkové listiny soutěže	16
6	Výstřižek tisknutelné podpůrné tabulky pro rozhodčího soutěže .	17
7	Hlavní aktivita pohledem přihlášeného uživatele, který si založil několik soutěží a již na nich jsou zapsaní uživatelé	19
8	Skutečný terč oproti simulovanému terči v aplikaci	20
9	Diagram databáze Webové části práce	21
10	Diagram databáze pro Android aplikaci	25
11	Architektura práce	29
12	Struktura projektu webové aplikace	36
13	Struktura projektu v Android studiu	40

Seznam tabulek

Seznam vět

Seznam zdrojových kódů

1	Ukázka velmi jednoduchého controlleru	31
2	Ukázka odeslání požadavku pomocí Fetch API	33
3	Ukázka kódu vlastního TypeConverteru	38
4	Příklad DAO abstraktní třídy pro ukládání a získávání třídy lo- gin_token z databáze	38
5	Příklad použití třídy ExecutorService a Callable pro provedení dotazu na databázi	39

1 Úvod

V rámci našeho střeleckého klubu jsou pravidelně pořádány tréninky (vzácněji i soutěže) praktické střelby. Za praktickou střelbu se považují situace, kdy je po účastníkovi požadována pokročilá manipulace se střelnou zbraní. Je naprosto běžné, že se po soutěžícím požaduje přesun z místa na místo nebo například nucené přebití.

Rozhodčí připraví jednu a více tzv. „stagí“. Stage je scénář jednoho „kola“ soutěže nebo tréninku. Každá stage má pevný počet terčů. Tyto terče jsou rozmístěny dle plánu rozhodčího po ploše střeliště. Stage může obsahovat zpestření například v podobě střelby z krytu nebo přenos zátěže a další. Výše zmíněné možnosti jsou opět jen příklady. Vše záleží na fantazii a kreativitě samotného rozhodčího při vymýšlení a plánování stagí.

Účastníci jsou za své výkony ohodnoceni časem. Cílem je mít co nejnižší čas. Tedy vyřešit stage co nejrychleji s co nejlepšími zásahy a maximální snahou vyhnout se penalizacím při plnění stagí. Je nutné najít vhodný kompromis mezi přesností a rychlostí. Nerozvážené nebo uspěchané plnění stage může znamenat časové penalizace a špatný konečný výsledek.

Každá stage má přesně definovaný počet hodnocených zásahů. Je na soutěžícím, kolik nábojů na každý terč vystřelí. Nicméně každý další výstřel znamená čas navíc. Jakmile účastník ukončí střelbu, rozhodčí zkontroluje jeho zásahy v terčích a k základnímu času účastníka přičte penalizace za zásahy a případné udělené procedury.

Samotný terč je rozdělen na tři zóny (A – Alfa, C – Charlie, D – Delta). Alfa zóna – nejlepší možný zásah. Za zásah Alfa zóny není obvykle přičítán žádný čas navíc. Zásah do Charlie zóny je druhou nejlepší možností zásahu. Nicméně už je k základnímu času účastníka přičtena drobnější penalizace v sekundách. Obvykle se jedná o 2-3 sekundy. Zásah do Delta zóny obnáší penalizaci kolem 3-5 sekund. Nejhorší možností je úplné minutí terče, tedy že zásah vůbec není v terči. V tomto případě už je časová penalizace vysoká. Příklad velkých terčů s vyznačenými zónami můžeme vidět na obrázku 1

Pokud je v terči více zásahů, než má stage definováno hodnocených zásahů, hodnotí se vždy nejlepší zásahy. Pokud je stage nastavená na tři hodnocené zásahy, přičemž tři zásahy budou v Alfa zóně a další tři v Delta zóně, rozhodčí bude hodnotit tři zásahy v Alfa zóně.

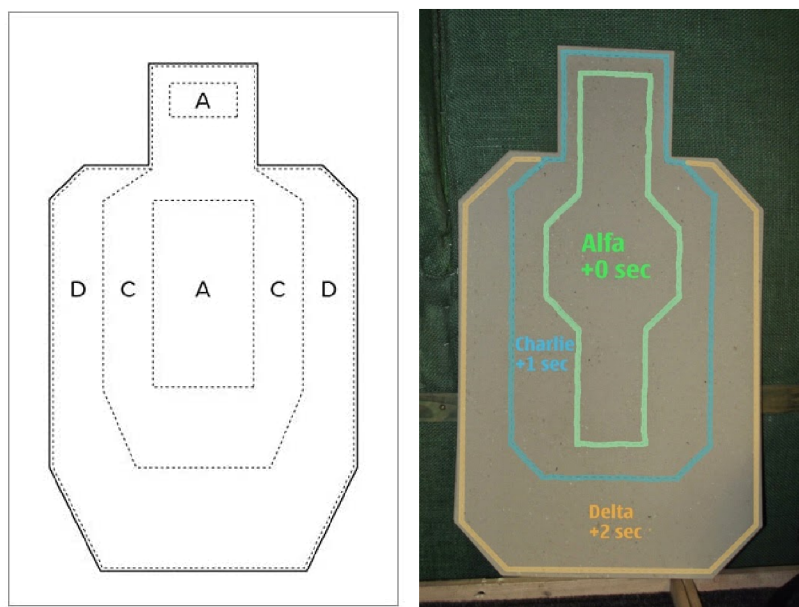
Velikost penalizací je zcela v kompetenci rozhodčího při definici stagí. Může se stát, že účastník udělá během plnění stage chybu. Jako příklad mohu uvést dvě situace. Například účastník splete nařízené pořadí terčů při přesunu nebo udělá přešlap při střelbě zpoza krytu. V tuto chvíli rozhodčí rozhodne o udělení tzv. „procedury“, tedy trestného času za pochybení účastníka při plnění stage. Procedura může být během plnění stage účastníkovi udělena i více. Například pokud účastník splete pořadí terčů, tak za každý výstřel na špatný terč mu bude udělena procedura.

Pokud rozhodčí usoudí, že si nepřeje, aby účastník dál pokračoval v soutěži,

může účastníka diskvalifikovat. Důvodem k diskvalifikaci bývá například absence na soutěži. Nicméně se také může jednat o závažné porušení bezpečnostních pravidel soutěže. Příkladem hrubého porušení bezpečnosti je situace, kdy účastník (byť i nechtěně) namíří zbraní do bezpečné zóny, tedy ohrozí ostatní účastníky.

Konečné ohodnocení účastníka je součet všech jeho časů ze všech stagi soutěže. Pokud účastník některou ze stagi nesplní, nemůže být hodnocen a je diskvalifikován ze soutěže.

Pokud je soutěž rozdělena na týmy, vyloučení jednoho člena týmu znamená diskvalifikaci celého týmu ze soutěže. Tým s diskvalifikovaným členem by byl zvýhodněn oproti ostatním týmům, protože by čas diskvalifikovaného člena nebyl započten.



Obrázek 1: Příklad terčů s vyznačenými zónami.

2 Motivace

Motivaci k vytvoření práce byl především zájem naučit se základní principy práce s REST API a frameworkem Spring. Snažil jsem se vytvořit rozsáhlejší projekt sestávající jak z webového, tak Android klienta. Jedním z mých cílů bylo pochopit koncept fungování přihlašovacího systému a jejich bezpečnosti.

Rozhodl jsem se tedy spojit bakalářskou práci s mým koníčkem – sportovní střelbou. Cílem je ulehčit a urychlit průběh střeleckých tréninků a soutěží.

Výsledky těchto akcí se oznamují na místě konání po jejím dokončení. Účastníci jsou hodnoceni časem, který se musí pro každého účastníka složitě počítat. K základnímu času je nutné připočítat penalizace za zásahy v tercích a za procedury. Nakonec je nutné výsledné časy ještě setřídit.

Počítání časů je pro běžného člověka velmi nepříjemné a nepohodlné. Taktéž lidský faktor představuje riziko chyby. Navíc se na výsledky musí dlouho čekat.

3 Uživatelská dokumentace

3.1 Webová aplikace

3.1.1 Vstupní stránka

Vzhled vstupní stránky pohledem přihlášeného uživatele můžeme vidět na obrázku 2. V navigačním menu se nachází logotyp stránek, který odkazuje na úvodní stránku. Na levé straně navigace se nachází tlačítko „*Přihlásit se*“, které zobrazí přihlašovací formulář.

V přihlašovacím formuláři uživatel zadá svůj email a heslo pro přihlášení. Pokud účet nemá, může si ho založit zvýrazněným odkazem v tomto formuláři.

Po vstupu na úvodní stránku má uživatel zobrazené dvě tabulky. První zobrazuje naplánované soutěže. Druhá zobrazuje vyhodnocené soutěže, kterých se uživatel zúčastnil. Pokud uživatel přihlášen není, druhá tabulka bude prázdná.

Pokud na stránku nahlíží přihlášený uživatel, místo tlačítka pro přihlášení bude zobrazeno jméno přihlášeného uživatele. Najede-li přihlášený uživatel kurzorem na své jméno, zobrazí se mu další možnosti.

- Detaily o účtu
- Odhlášení se
- Přehled uživatelů – Tato možnost se zobrazí pouze uživateli, který patří mezi administrátory.

Pokud je uživatel přihlášený, v obou tabulkách se mu zpřístupní tlačítko „*Detail*“.

Při kliknutí na toto tlačítko v první tabulce plánovaných soutěží se uživateli zobrazí stránka s detailem zvolené nadcházející soutěže. Zde má uživatel možnost se na soutěž zapsat (dokud není dosažena maximální kapacita). Pokud je uživatel na soutěži již zapsaný, může se odepsat.

Druhá tabulka obsahuje vyhodnocené a ukončené soutěže, kterých se přihlášený uživatel zúčastnil nebo je hodnotil. Při kliknutí na tlačítko „*Detail*“ u dané soutěže v této tabulce se uživateli zobrazí výsledková listina této soutěže

Na pravé straně pod jménem přihlášeného uživatele se nachází tlačítka „*Vytvořit soutěž*“. Zde si přihlášený uživatel může vytvořit svou vlastní soutěž. Tlačítko uživatele přesměruje na formulář pro vytvoření soutěže.

Plánované soutěže

Vytvořit soutěž

Název	Místo	Datum	Vedoucí	
Moje soutěž 2	Timpo, Olomouc	pátek 20. června 2025	První Prvák	DETAIL
Team Attack	Křelov, Na Fortu VII	neděle 20. února 2022	Vít Řehák	DETAIL
Moje soutěž	Praha	úterý 22. února 2022	Vít Řehák	DETAIL

Vaše historie

Název	Místo	Datum	Vedoucí	
KMFC Cup	Křelov	středa 2. června 2021	První Prvák	DETAIL
Tour de Headshot	Pekelné doly	sobota 3. července 2021	Milan Řehák	DETAIL
Skupinovka	Olomouc	pondělí 5. července 2021	Vít Řehák	DETAIL
Viditelnost	Mars	pátek 2. července 2021	Vít Řehák	DETAIL
Skupinová soutěž	Timpo Olomouc	neděle 23. března 2025	Vít Řehák	DETAIL

Obrázek 2: Vstupní stránka pohledem přihlášeného uživatele

3.1.2 Vytvoření soutěže

Na této stránce si přihlášený uživatel může nadefinovat a vytvořit svoji soutěž. Nepřihlášený uživatel bude přeměrován na stránku úvodní.

Formulář obsahuje následující povinné údaje:

- Název – Název soutěže, pod kterým se soutěž ostatním uživatelům objeví v tabulce nacházejících soutěží.
- Místo konání – Místo konání soutěže.
- Datum konání – Datum konání soutěže. Soutěž musí být vytvořena minimálně 48 hodin jejím konáním a samozřejmě datum nesmí být v minulosti.
- Rozdělení do týmů – Určuje, zda se jedná o soutěž na jednotlivce nebo na skupiny.
- Počet účastníků – Maximální počet účastníků pro soutěže jednotlivců.
- Počet týmů – V případě týmové soutěže určuje, kolik týmů bude soutěžit.
- Maximální počet účastníků v každém týmu – Počet členů v každém týmu. Počet je pro všechny týmy jednotný, všechny týmy mají stejný počet členů.

Následuje tabulka pro definice stagí. Každá soutěž musí mít minimálně jednu stage. Uživatel, který soutěž vytváří definuje počet sekund za jednotlivé zásahy zón a penalizace. Podle těchto údajů se potom budou vyhodnocovat stage v mobilní aplikaci vlastníka soutěže.

Vlastnosti stagí jsou následující:

- Název stage – Název stage.
- Počet terčů – Určuje počet terčů pro tuto stage.
- Hodnocené zásahy – Určuje, kolik zásahů se při vyhodnocování počítá.
- Penalizace za proceduru – Určuje, kolik sekund bude v dané stage uživatele stát jedna procedura.
- Alpha hit – Počet vteřin za zásah alfa zóny. V naprosté většině případů je počet sekund 0.
- Charlie hit – Počet sekund za zásah Charlie zóny. Nesmí být záporná hodnota.
- Delta hit – Počet sekund za zásah Delta zóny. Nesmí být záporná hodnota.
- Miss hit – Počet sekund za úplné minutí terče/chybějící zásah. Nesmí být záporná hodnota.

Stage jsou zobrazené v tabulce, která je z počátku prázdná. Novou stage uživatel vytvoří tak, že kurzorem klikne na červené tlačítko. Vzhled vstupní stránky pohledem přihlášeného uživatele můžeme vidět na obrázku „*Vytvořit novou stage*“. Po kliknutí bude nově vytvořená stage přidána do tabulky. Ukázku tří nově vytvořených stagí můžeme vidět na obrázku 3

Upravení stage funguje tak, že uživatel si myší navolí řádek v tabulce a následně zadává požadované informace do formuláře pod tabulkou. Úpravy stage jsou vidět ihned v tabulce.

Stage lze odstranit kliknutím na černý křížek na konci příslušného řádku.

Potvrzení vytvoření soutěže se vykoná stiskem tlačítka "Uložit soutěž". V případě validních vstupů je soutěž vytvořena a uživatel je přesměrován na úvodní stranu, kde je jeho soutěž zobrazena. Pokud je některý ze vstupů nevalidní, bude uživatel upozorněn hláškou. Pokud server neakceptuje vytvoření soutěže (např. odstranění kontrol validity JavaScriptu nebo podsunutí falešného requestu) bude ze strany serveru navrácena odpovídající výjimka.

Stage

Přidat novou Stage

Název stage	Pocet tercu	Hodnocene zasahy	Penalizace za proceduru	Alpha hit	Charlie hit	Delta hit	Miss
Stage 1	3	3	10	0	5	10	15 ✘
Stage 2	6	5	20	0	20	30	45 ✘
Stage 3	8	3	10	0	3	6	9 ✘

Nazev stage:

Pocet tercu:

Pocet hodnocenych zasahu:

Penalizace za proceduru:

Cas za zasah Alpha zony:

Cas za zasah Charlie zony:

Cas za zasah Delta zony:

Cas za miss:

ULOŽIT SOUTĚŽ

Obrázek 3: Ukázka vytvořených stagií v tabulce

3.1.3 Detail nadcházející soutěže

Po přeměrování na stránku detailu soutěže jsou zobrazeny konkrétní detaily o soutěži. Hned pod navigací jsou pro uživatele zobrazeny klíčové informace o soutěži. Je zobrazen název soutěže, místo konání, datum konání, zda je soutěž týmová nebo ne, popis soutěže a rozhodčí soutěže.

V případě, že uživatel není na soutěž přihlášen, zobrazí se tlačítko pro zapsání do soutěže „Zapsat se“ (eventuálně může být tlačítek více, pokud se jedná o skupinovou/týmovou soutěž). Pro zapsání soutěže musí uživatel zaškrtnout checkbox, kterým stvrzuje, že souhlasí s podmínkami, pravidly a bezpečnostními pokyny soutěže.

Pokud si vlastník zobrazí detail své soutěže, má k dispozici další funkcionality. Vlastník může soutěž zrušit, upravit nebo si vytisknout podpůrný dokumentační arch. Může také ze soutěže vyloučit zapsaného uživatele.

Příklad vzhledu stránky pohledem vlastníka soutěže můžeme vidět na obrázku 4

[Hlavní stránka](#) > [Detail Team Attack](#)

Detaily o soutěži

Název: Team Attack

Místo konání: Křelov, Na Fortu VII

Datum konání: neděle 20. února 2022

Počet týmů: 2

Rozhodčí: Vít Řehák

ZRUSIT SOUTEZ

UPRAVIT SOUTEZ

TISK DOKUMENTACE

Popis soutěže:

description

Skupina 1 - 2/2

Odepsat se

Index	Jméno	Příjmení	
1.	Milan	Řehák	Vyloučit
2.	Vít	Řehák	Vyloučit

Skupina 2 - 2/2

Index	Jméno	Příjmení	
1.	Willy	Wonka	Vyloučit
2.	Sedmák	Sedmý	Vyloučit

Obrázek 4: Detail soutěže pohledem přihlášeného uživatele

3.1.4 Úprava a zrušení soutěže vlastníkem

Vlastník má pravomoc pozměnit nebo zrušit svou vypsanou soutěž. Pro úpravu je vlastník soutěže přesměrován na stejný formulář jako při jejím vytvoření. Zrušení soutěže odepíše všechny zapsané uživatele a následně soutěž smaže z databáze. Soutěž je odstraněna z navrhovaných soutěží v tabulce na úvodní straně

3.1.5 Detail vyhodnocené soutěže

Pokud soutěž, na kterou se uživatel zapsal, již proběhla a je plně vyhodnocena, zobrazí se uživateli výsledky v tabulce na úvodní stránce.

Uživateli se zde zobrazí seřazená výsledná listina. V případě soutěže pro jednotlivce se zobrazí pouze výsledná listina jednotlivců. V případě skupinové soutěže se zobrazí jak výsledná listina jednotlivců, tak i výsledná listina skupin. Seřazení dle času jsou i účastníci v každé skupině.

Pod těmito údaji se na stránce zobrazí také tabulka s diskvalifikovanými uživateli. Pokud v soutěži žádné diskvalifikace nenastaly, tato tabulka se nezobrazí. Na obrázku 5 můžeme vidět vzhled výsledkové listiny soutěže o dvou týmech po dvou členech.

Pokud vlastník soutěže účastníka diskvalifikuje, má dále pravomoc tohoto účastníka zablokovat ze všech svých soutěží. Zablokovaný uživatel soutěže vlastníka neuvidí v seznamu a nebude mu umožněno se na tyto soutěže vlastníka zapsat.

KMFC Assistant					Vit Řehák ☰
Hlavní stránka > Výsledek Skupinová soutěž					
Nazev:	Skupinová soutěž				
Místo konání:	Tímpe Olomouc				
Datum konání:	neděle 23. března 2025				
Rozhodci:	Vit Řehák				
Skupiny					
1. Skupina - 00:02:10					
Jméno	Příjmení	Stage 1	Stage 2	Celkový čas	
Osmý	Osmák	00:00:08	00:00:41	00:00:49	
Sedmák	Sedmý	00:01	00:00:21	00:01:21	
2. Skupina - 00:05:27					
Jméno	Příjmení	Stage 1	Stage 2	Celkový čas	
První	Prvák	00:01:03	00:01:33	00:02:36	
Druhy	Ucastník	00:01:33	00:01:18	00:02:51	
3. Skupina - 00:07:30					
Jméno	Příjmení	Stage 1	Stage 2	Celkový čas	
Patečník	Patkovity	00:01:01	00:02:14	00:03:15	
Třetí	Prichozi	00:01:13	00:03:02	00:04:15	
Jednotlivci					
Umístění	Jmeno	Stage 1	Stage 2	Celkovy cas	
1.	Osmý Osmák	00:00:08	00:00:41	00:00:49	
2.	Sedmák Sedmý	00:01	00:00:21	00:01:21	
3.	První Prvák	00:01:03	00:01:33	00:02:36	
4.	Druhy Ucastník	00:01:33	00:01:18	00:02:51	
5.	Patečník Patkovity	00:01:01	00:02:14	00:03:15	
6.	Třetí Prichozi	00:01:13	00:03:02	00:04:15	

Obrázek 5: Příklad výsledkové listiny soutěže

3.1.6 Zobrazení údajů o účtu

Na této stránce má uživatel možnost si změnit jméno, email nebo heslo. Kliknutím na příslušné tlačítko mu bude zobrazen formulář s požadovanými údaji pro vykonání změny.

Dále je této stránce vypsan seznam uživatelů, které má právě přihlášený uživatel zablokované. Přihlášený uživatel má na této stránce možnost zrušení blokování uživatelů, které si předtím zablokoval.

3.1.7 Stránka pro tisk dokumentace

Vlastník soutěže si může vytisknout pomocný formulář ke své soutěži. Formulář obsahuje hodnotící listinu obsahující všechny zapsané uživatele se seznamem všech stagií. Kvůli omezeným prostorovým možnostem listu papíru pro eventuální tisk se do jedné tabulky zobrazí maximálně pět stagií. Pokud má soutěž více než 5 stagií, tak bude seznam účastníků vypsan znovu s dalšími stagemi. Tohle se děje pro každou pěticí stagií. Po navštívení stránky se sám od sebe otevře uživateli formulář pro tisk. Na obrázku 6 můžeme vidět ukázkou tabulky pro soutěž o dvou týmech po dvou členech v každém týmu.

20. 7. 2021	Team Attack																																																																											
Skupina 1	Čas celkem:																																																																											
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th rowspan="2">Jméno</th> <th colspan="6">Stage 1</th> <th colspan="6">Stage 2</th> <th colspan="6">Stage 3</th> </tr> <tr> <th>Čas</th> <th>A</th> <th>C</th> <th>D</th> <th>M</th> <th>Proc</th> <th>Čas</th> <th>A</th> <th>C</th> <th>D</th> <th>M</th> <th>Proc</th> <th>Čas</th> <th>A</th> <th>C</th> <th>D</th> <th>M</th> <th>Proc</th> </tr> </thead> <tbody> <tr> <td>Milan Řehák</td> <td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>Vít Řehák</td> <td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>	Jméno	Stage 1						Stage 2						Stage 3						Čas	A	C	D	M	Proc	Čas	A	C	D	M	Proc	Čas	A	C	D	M	Proc	Milan Řehák																			Vít Řehák																			
Jméno		Stage 1						Stage 2						Stage 3																																																														
	Čas	A	C	D	M	Proc	Čas	A	C	D	M	Proc	Čas	A	C	D	M	Proc																																																										
Milan Řehák																																																																												
Vít Řehák																																																																												
Skupina 2	Čas celkem:																																																																											
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th rowspan="2">Jméno</th> <th colspan="6">Stage 1</th> <th colspan="6">Stage 2</th> <th colspan="6">Stage 3</th> </tr> <tr> <th>Čas</th> <th>A</th> <th>C</th> <th>D</th> <th>M</th> <th>Proc</th> <th>Čas</th> <th>A</th> <th>C</th> <th>D</th> <th>M</th> <th>Proc</th> <th>Čas</th> <th>A</th> <th>C</th> <th>D</th> <th>M</th> <th>Proc</th> </tr> </thead> <tbody> <tr> <td>Willy Wonka</td> <td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>Sedmák Sedmý</td> <td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td> <td></td><td></td><td></td><td></td><td></td><td></td> </tr> </tbody> </table>	Jméno	Stage 1						Stage 2						Stage 3						Čas	A	C	D	M	Proc	Čas	A	C	D	M	Proc	Čas	A	C	D	M	Proc	Willy Wonka																			Sedmák Sedmý																			
Jméno		Stage 1						Stage 2						Stage 3																																																														
	Čas	A	C	D	M	Proc	Čas	A	C	D	M	Proc	Čas	A	C	D	M	Proc																																																										
Willy Wonka																																																																												
Sedmák Sedmý																																																																												

Obrázek 6: Výstřižek tisknutelné podpůrné tabulky pro rozhodčího soutěže

3.2 Android aplikace

Aplikace pro Android je pomůcka pro rozhodčí soutěží. Rozhodčímu umožní rychlé a jednoduché zadávání vstupů, které budou následně zpracovány dle definic soutěží, které si vytvořil na svém účtu ve webové stránce.

3.2.1 Aktivita pro přihlášení

Po prvotním otevření aplikace se uživateli objeví přihlašovací formulář. Zde použije stejné přihlašovací údaje, jaké používá na webové části práce. Pokud jsou přihlašovací údaje správné, je uživatel autorizován a přesunut do hlavního menu mobilní aplikace. Při neúspěšné autorizaci je zobrazena hláška o chybných přihlašovacích údajích.

3.2.2 Aktivita seznamu soutěží uživatele

Po úspěšném přihlášení je uživatel přesunut na tuto aktivitu. Aktivita zobrazuje seznam všech soutěží uživatele, které si vytvořil. V navigačním baru je tlačítko pro synchronizaci se serverem. Po stisknutí tlačítka jsou ze serveru staženy soutěže uživatele a jsou serveru odeslány dosavadní výsledky.

Po rozkliknutí ikony třech teček se rozvine kontextové menu, kde se uživatel může z aplikace odhlásit.

Každá položka soutěže v seznamu zobrazuje jméno soutěže, obsazení soutěže, datum konání a místo. Rozkliknutím konkrétní soutěže ze seznamu uživatel pokračuje do další aktivity, což je seznam stagí soutěže. Na obrázku můžeme vidět hlavní menu pohledem přihlášeného uživatele, který vlastní dvě soutěže. Na obrázku 7 vidíme hlavní menu pohledem přihlášeného uživatele, který si založil dvě soutěže a již na nich má zapsané účastníky.

V rámci testování je možné soutěž otevřít a začít hodnotit i před jejím datem konání. V ostré verzi aplikace toto nebude možné a soutěž půjde hodnotit nejdříve 24 hodin před jejím konáním.



Obrázek 7: Hlavní aktivita pohledem přihlášeného uživatele, který si založil několik soutěží a již na nich jsou zapsaní uživatelé

3.2.3 Aktivita seznamu stagi

Aktivita zobrazí seznam stagi pro danou vybranou soutěž. V seznamu je pro stage zobrazen její název a kolik uživatelů již bylo ohodnoceno.

3.2.4 Aktivita seznamu účastníků

Po rozkliknutí stage v předchozí aktivitě je zobrazena listina všech účastníků. Pro každého účastníka je zobrazen jeho index, jméno, příjmení, celkový čas za všechny stage a čas pouze za současnou stage.

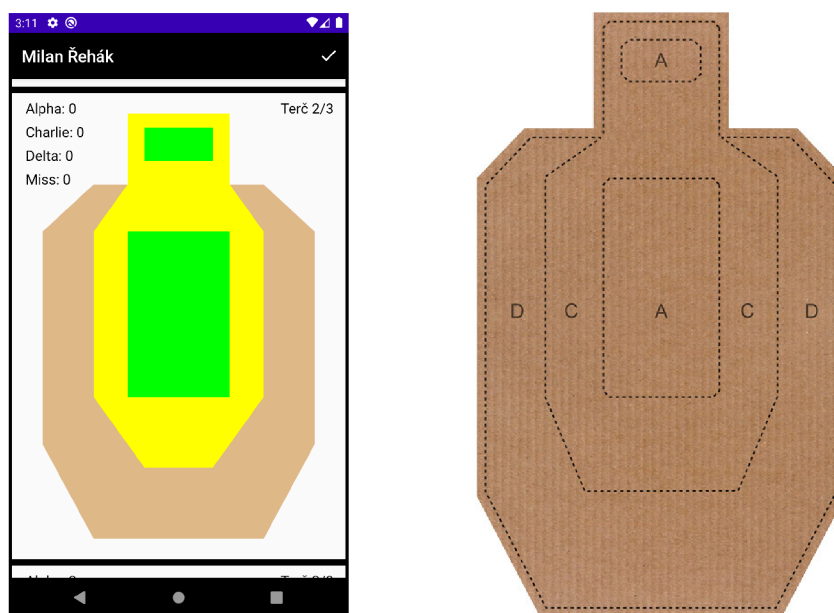
Při dlouhém kliknutí nad konkrétním uživatelem se otevře kontextové menu s možností diskvalifikace účastníka. Po stisknutí možnosti diskvalifikovat se otevře nová aktivita. V této aktivitě je uživatel požádán o zadání důvodu diskvalifikace. Po potvrzení diskvalifikace je uživatel posunut na konec seznamu a barevně označen jako diskvalifikovaný. Pokud se jedná o diskvalifikaci v rámci týmové soutěže, je diskvalifikován celý tým. Bylo by totiž výhodné si diskvalifikovat co nejvíce členů týmu, ať se jejich výsledkové časy neakumulují. Pokud se rozhodne diskvalifikaci zrušit, opět stačí dlouhým stiskem kliknout na diskvalifikovaného uživatele a vybrat položku „zrušit diskvalifikaci“. Diskvalifikovaný uživatel bude navrácen do soutěže. V případě týmové soutěže bude do soutěže navrácen celý tým.

V navigačním menu se nachází ikony "fajfky". Při stisku této ikony je potvrzeno ukončení současné aktivity. Uživatel je přesunut o jednu aktivitu níže. Je tedy navrácen zpět na seznam stagí. Druhou položkou v navigačním menu je zobrazení seřazených výsledků. Zobrazovat tyto výsledky má však smysl až po ohodnocení všech uživatelů na všech stagích.

3.2.5 Skórovací aktivita pro ohodnocení účastníků

Po rozkliknutí konkrétního účastníka soutěže je spuštěna „skórovací“ aktivita. Do prvního boxu se zadává základní čas, za který účastník stage dokončil. Tento čas se zadává v sekundách. Je možné k těmto sekundám doplnit i milisekundy. Do druhého boxu se zadává počet procedur, které byly účastníkovi uděleny.

Podstatnou částí této aktivity je vlastní komponenta představující standardní velký IPSC terč, který můžeme vidět na pravé straně v obrázku 8. Počet těchto terčových komponent je definovaný vlastností „počet terčů“ u každé stage. Uživatel nyní může klikáním do komponenty umisťovat zásahy. Pokud uživatel udělá při zadávání zásahů chybu, může dlouhým kliknutím na daný terč odstranit poslední zásahy. Každý dlouhý stisk odstraní poslední zadaný zásah. Po vložení zásahů všech terčů a zadání času a procedur, uživatel stiskne ikonu „fajfky“ v navigačním menu. V tuto chvíli je uživateli pro současnou stage vytvořena statistika a je vypočten jeho čas za tuto stage. Hodnota je potom připočtena k celkovému času účastníka. Pokud uživatel nezadá nějaký vstup korektně, bude na to upozorněn hláškou. Pokud uživatel do terče vloží méně zásahů, než je počet zásahů hodnocených, budou chybějící zásahy počítány jako miss (minuté).



Obrázek 8: Skutečný terč oproti simulovanému terči v aplikaci

4 Modely databází

Jako databázový systém byla pro webovou aplikaci zvolena MariaDB. MariaDB je volně dostupná MySQL relační databáze. Jako rozhraní mezi databází a aplikací je použita technologie JDBC (Java Database Connectivity).

4.1 Model databáze webové aplikace



Obrázek 9: Diagram databáze Webové části práce

4.2 Tabulky databáze webové aplikace

4.2.1 Tabulka users

Tabulka ukládá data o registrovaných uživateli.

- `id` – ID uživatele, primární klíč tabulky.
- `first_name` – Křestní jméno uživatele.
- `last_name` – Příjmení uživatele.
- `email` – Email, kterým uživateli slouží pro přihlášení. Každý email musí být unikátní, takže tato položka je stejně jako ID uživatele primární klíč.
- `password` – Heslo uživatele, které mu slouží k přihlašování. Uložené heslo je samozřejmě zahashované. Pro hashování hesel v mé práci byl použit algoritmus BCrypt
- `admin` – Položka, která určuje, zda má uživatel oprávnění k administrátorským akcím. Do těch akcí se řadí možnost zobrazení uživatelů a případně jejich smazání. Administrátor má dále pravomoc upravovat a odstraňovat soutěže uživatelů.

4.2.2 Tabulka login_tokens

Tabulka slouží k ukládání „tokenů“ pro „remember me“ funkcionalitu přihlašování webové aplikace. Pokud uživatel poskytne správný token, je automaticky přihlášen bez nutnosti zadávat přihlašovací údaje.

- `user_id` – ID uživatele, kterému tento token náleží.
- `token` – Náhodně generovaný kód, který slouží k autorizaci uživatele bez zadání přihlašovacích údajů.

4.2.3 Tabulka android_login_tokens

Tabulka slouží k ukládání verifikačních „tokenů“ pro mobilní aplikaci. Tyto tokeny slouží k autorizaci uživatele při požadavku na synchronizaci mobilní aplikace s webovým serverem.

- `user_id` – ID uživatele, kterému tento token náleží.
- `token` – Vygenerovaný textový řetězec.

4.2.4 Tabulka competitions

Tabulka obsahuje základní data o jednotlivých soutěžích

- id – Představuje ID soutěže. Položka je automaticky inkrementovaný primární klíč.
- name – Jméno soutěže.
- venue – Místo konání soutěže.
- owner_id – ID uživatele, který tuto soutěž založil. Tento uživatel je tedy hlavní rozhodčí.
- date – Datum konání soutěže.
- synced – Položka, která značí zda je soutěž dokončena.
- description – Popis soutěže, pravidel a bezpečnostních pokynů.

4.2.5 Tabulka stages

Do tabulky jsou ukládány data o jednotlivých stadiích soutěží.

- id – Představuje ID stage. Položka je automaticky inkrementovaný primární klíč.
- name – Jméno stage
- ranked_hits – Počet hodnocených zásahů pro všechny terče ve stadii.
- alpha_hit – Penalizace zásahu Alfa zóny v sekundách. Je jediná oblast zásahu, která může mít zápornou hodnotu.
- charlie_hit – Penalizace za zásah Charlie zóny v sekundách.
- delta_hit – Penalizace za zásah delta zóny v sekundách.
- miss_hit – Penalizace za chybějící zásah v sekundách.

4.2.6 Tabulka stage_stats

Tabulka ukládá data o průběhu plnění stage účastníky soutěží.

- competition_id – ID soutěže, se kterou je tato statistika spojena.
- user_id – ID uživatele, se kterým je tato statistika propojena.
- stage_id – ID stage, se kterou je tato statistika propojena.
- alpha_hits – Počet zásahů v alfa zóně.

- charlie_hits – Počet zásahů v Charlie zóně.
- delta_hits – Počet zásahů v delta zóně.
- miss_hits – Počet zásahů, které minuly terč.
- procedure_count – Počet udělených procedur.
- base_time – Čas v sekundách, za který byla stage dokončena. Neobsahuje započtené procedury a zásahy.
- total_time – Celkový čas, kde jsou započteny jak penalizace zásahů, tak penalizace za procedury dle definice dané stage.

4.2.7 Tabulka registrations

V této tabulce jsou uloženy jednotlivé registrace uživatelů na soutěže.

- id – ID registrace.
- user_id – ID uživatele, které se přihlásil na soutěž.
- competition_id – ID soutěže, na kterou se uživatel přihlásil.
- team – Číslo skupiny/týmu, do které se soutěžící přihlásil.

4.2.8 Tabulka disqualifications

Tabulku obsahuje data o diskvalifikací uživatelů ze soutěží.

- user_id – ID uživatele, který byl diskvalifikován.
- competition_id – ID soutěže, ze které byl uživatel diskvalifikován.
- description – Důvod, proč byl uživatel diskvalifikován ze soutěže.

4.2.9 Tabulka user_bans

Tabulka ukládá data o blokacích mezi uživateli na webu. Pokud si uživatel někoho zablokuje, tak zablokovaný uživatel neuvidí vypsané soutěže tohoto uživatele a nebude se na ně moci zapsat.

- user_id – ID uživatele, který byl diskvalifikován.
- competition_id – ID soutěže, ze které byl uživatel diskvalifikován.
- description – Důvod, proč byl uživatel diskvalifikován ze soutěže.

4.3 Model databáze aplikace pro Android

Pro perzistentní ukládání dat v mobilní aplikaci jsem zvolil technologii ROOM. ROOM je abstrakční vrstva nad SQLite databází.

Model databáze pro Android je zjednodušenou verzí modelu pro webovou aplikaci. Chtěl bych upozornit na změnu. Uživatelé zapsaní na soutěžích právě přihlášeného uživatele jsou nyní označováni jako Participants. Tabulka users ukládá data o právě přihlášeném uživateli v Android aplikaci.



Obrázek 10: Diagram databáze pro Android aplikaci

4.4 Tabulky databáze mobilní aplikace

4.4.1 Tabulka users

Tato tabulka ukládá data o právě přihlášeném uživateli v mobilní aplikaci. Tabulka nikdy nemá více než jeden řádek.

- id – ID právě přihlášeného uživatele.
- first_name – Křestní jméno uživatele.
- last_name – Příjmení uživatele.
- login_token – Token, kterým se uživatel autorizuje při požadavku na synchronizaci se serverem.

4.4.2 Tabulka Participants

Tato tabulka popisuje uživatele, kteří jsou přihlášení na soutěžích, které vytvořil právě přihlášený uživatel v Android aplikaci.

- id – ID uživatele, primární klíč tabulky
- index – Položka pro zachování pořadí, ve kterém se uživatelé na soutěž přihlásili.
- first_name – Křestní jméno uživatele.
- last_name – Příjmení uživatele.

4.4.3 Tabulka competitions

Tabulka ukládá data o soutěžích přihlášeného uživatele.

- id – Představuje ID soutěže. Položka je automaticky inkrementovaný primární klíč.
- name – Jméno soutěže.
- venue – Místo konání soutěže.
- owner_id – ID uživatele, který tuto soutěž založil. Tento uživatel je hlavní rozhodci.
- date – Datum konání soutěže.

4.4.4 Tabulka stages

Tabulka je totožná s tabulkou databáze webového serveru.

- id – Představuje ID stage. Položka je automaticky inkrementovaný primární klíč.
- name – Jméno stage.
- ranked_hits – Počet hodnocených zásahů pro všechny terče ve stagi.
- alpha_hit – Penalizace zásahu Alfa zóny v sekundách. Je jediná oblast zásahu, která může mít zápornou hodnotu.
- charlie_hit – Penalizace za zásah Charlie zóny v sekundách.
- delta_hit – Penalizace za zásah delta zóny v sekundách.
- miss_hit – Penalizace za chybějící zásah v sekundách.

4.4.5 Tabulka stage_stats

Tabulka ukládá data o jednotlivých statistikách uživatelů na soutěžích.

- competition_id – ID soutěže, se kterou je tato statistika spojena.
- user_id – ID uživatele, se kterým je tato statistika propojena.
- stage_id – ID stage, se kterou je tato statistika propojena.
- alpha_hits – Počet zásahů v alfa zóně.
- charlie_hits – Počet zásahů v Charlie zóně.
- delta_hits – Počet zásahů v delta zóně.
- miss_hits – Počet zásahů, které minuly terč.
- procedure_count – Počet udělených procedur.
- base_time – Čas v sekundách, za který byla stage dokončena. Neobsahuje započtené procedury a zásahy.
- base_time – Základní čas, tedy čas od startu do konce plnění stage.
- total_time – Celkový čas, kde jsou započteny jak penalizace zásahů, tak penalizace za procedury.

4.4.6 Tabulka registrations

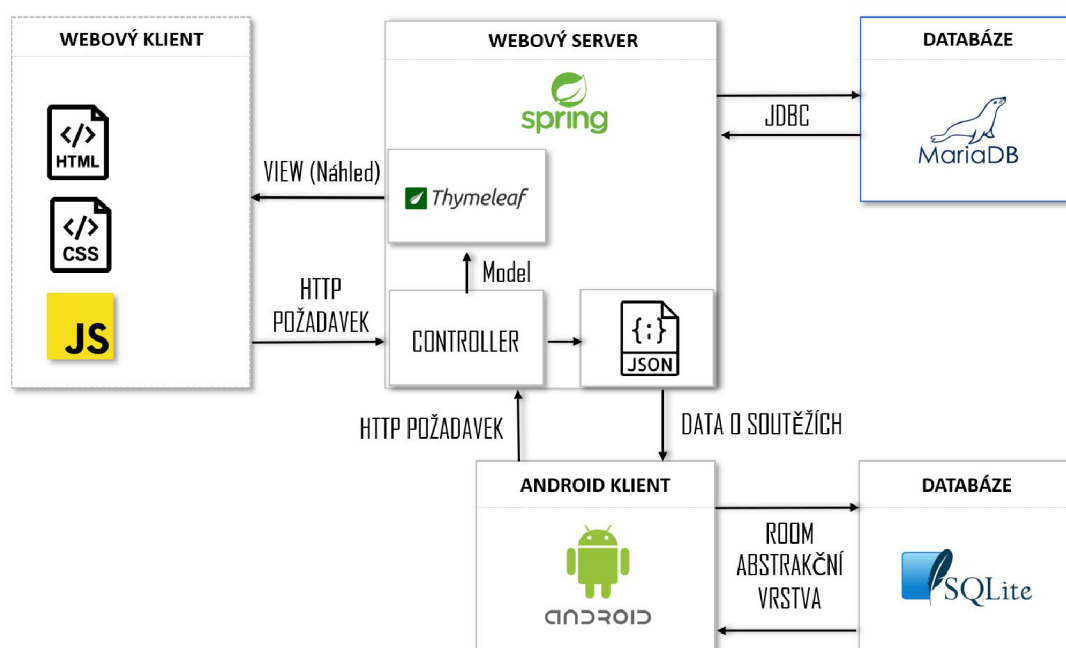
Tabulka je totožná s tabulkou registrations v modelu Webové aplikace.

- id – ID registrace
- participant_id – ID účastníka, které se přihlásil na soutěž.
- competition_id – ID soutěže, na kterou se uživatel přihlásil.
- team – Číslo skupiny/týmu, do které se soutěžící přihlásil.

5 Programátorská dokumentace

5.1 Architektura práce

Na obrázku 11 můžete vidět diagram představující architekturu práce. Základem je webový server vytvořený ve frameworku Spring. Klienti se serverem komunikují pomocí HTTP požadavků, na které server odpovídá. Server přijímá požadavky od webového klienta a Android klienta. Požadavek webového klienta je zpracováván podle MVC (Model–View–Controller) modelu. Při obdržení požadavku z Android klienta server odpovídá po autorizaci klienta odesláním požadovaných dat ve formátu JSON.



Obrázek 11: Architektura práce

5.2 Webová část

5.2.1 Spring MVC

Pro tvorbu webové aplikace je využit model MVC, který Spring nabízí. Prvním krokem je vytvoření tzv. controlleru. Controller je vytvořen tak, že třídě (nazvěme ji `ExampleController`) přidělíme anotaci `@Controller`. V takto anotované třídě můžeme nyní na jednotlivé metody anotacemi namapovat tzv. URL endpointy. Anotací určíme i typ metody HTTP požadavku. Mezi metody HTTP požadavků se řadí GET, POST, UPDATE, DELETE. Žádné dva endpointy nesmí mít stejnou metodu a URL.

- `@GetMapping`
- `@PostMapping`
- `@PutMapping`
- `@DeleteMapping`

Jako argument pro tyto anotace se zadává konkrétní url endpointu. V ukázce níže to je „/namepage“. Příklad jednoduchého controlleru můžeme vidět ve zdrojovém kódu [1](#). Po spuštění webového serveru (v základní verzi Springu je integrovaný Tomcat) bude webový server odposlouchávat požadavky od klientů.

Controller zpracovává data a může je zadat do tzv. modelu. Model uchovává zadaná data pro vytvoření pohledu (view). Data zanikají po vytvoření náhledu pro klienta.

View je náhled, který je zobrazen uživateli. V mé práci využívám technologii Thymeleaf pro vytváření a zpracování náhledu pomocí dynamických HTML šablon. Více k Thymeleafu v kapitole [5.2.3](#).

```

1 @Controller
2 public class ExampleController {
3
4     @GetMapping("/namepage")
5     public String getNamePage(Model model,
6                               HttpServletRequest request) {
7
8         // Přístup k session klienta. Parametr create určuje,
9         // zda má být session vytvořena, pokud neexistuje.
10        // Pokud existuje, bude to proměnné session uložena.
11        // Jinak proměnná session bude null.
12        HttpSession session = request.getSession(false);
13
14        // Vložení údaje do session
15        session.setAttribute("lastVisited", "namePageController");
16
17        // ... práce s daty
18        // ...
19        // ...
20
21        // Vkládání dat do modelu
22        model.addAttribute("name", "Jmeno");
23        model.addAttribute("surname", "Prjmeni");
24
25        // Vrací název šablony pro zpracování
26        return "page";
27
28        // přesmerování na jiný controller
29        // return "redirect:/namepage"
30    }
31
32 }

```

Zdrojový kód 1: Ukázka velmi jednoduchého controlleru

5.2.2 HttpSession

Session je velmi užitečná technologie pro dlouhodobější propojení mezi webovým serverem a klientem. Session umožňuje ukládání dat o klientovi. Data jsou dostupná dokud se klient neodpojí. Data tedy jsou dostupná navzájem mezi jednotlivými controllery. To je zásadní rozdíl oproti datům uloženým v modelu. Data v modelu nejsou zachována a zanikají po zpracování HTML šablony. Session zaniká, pokud se klient odpojí (vypne prohlížeč) nebo je klient delší dobu neaktivní.

Velkou výhodou je, že klient nemá přístup přímo k uchovávaným datům. Data jsou uložena na straně serveru a klient s nimi nemůže přímo manipulovat. Je tedy od těchto dat odstíněn. Klient má na své straně pouze cookie, která obsahuje identifikační token aktuální session. Server klienta autentizuje pomocí tohoto „session id“ tokenu. Klient s existující session tedy s požadavkem pošle

tento identifikační token, server ho přijme a ví, se kterými konkrétními daty pracovat.

V moji práci session používám velmi často, zejména v oblasti přihlašování.

5.2.3 Zpracování HTML šablony technologií Thymeleaf

Thymeleaf umožňuje zpracovávat šablony náhledu pro uživatele. Například na úvodní stránce „index“ jsou klientovi zobrazena data na základě situace a podmínek. Pokud uživatel není přihlášený, neuvidí detaily o soutěžích. Pokud přihlášený je, tyto detaily se mu zobrazí. Šablona má přístup k datům uloženým v aktuálním modelu a k datům uložených přímo v session. Thymeleaf je velmi užitečný například pro vytváření tabulek s dynamickým obsahem (proměnlivý počet sloupců/řádků apod.). Veškeré šablony se nachází v složce resources/templates.

5.2.4 Práce s databází

V mé práci je pro webovou část jako databázový systém využita MariaDB. Pro komunikaci mezi webovou aplikací je využito standardní JDBC rozhraní. Pro práci s databází jsou připraveny „DBManager“ třídy. Pro každý objekt reprezentující tabulku je tedy vytvořena odpovídající DBManager třída, která zajišťuje volání dotazů a práci přímo s databází.

Veškeré dotazy jsou předpřipravené pomocí třídy PreparedStatement. Dotazy jsou tedy pevně definované a nikdy nejsou skládány jako textový řetězec. Pokud by byl dotaz skládán jako textový řetězec, objevuje se riziko SQL Injection útoku.

5.2.5 Komunikace mezi serverem a klientem

Klient a server mezi sebou komunikují skrze tzv. requesty. Klient odešle požadavek na server. Server ho zpracuje a odešle odpověď. Odesílání a přijímání požadavků na straně webového klienta je zaopatřeno rozhraním Fetch. Fetch rozhraní nahrazuje starší třídu XMLHttpRequest. Do těla požadavku je vložena metoda požadavku a parametry, které chceme serveru odeslat. Následně je zavoláním funkce Fetch a požadavek odeslán na zadanou URL. Je nutné poznamenat, že „fetch“ je asynchronní funkce a musí se s ní podle toho zacházet. Ukázkou odeslání požadavku na server vidíme ve zdrojovém kódu [2](#)

```

1
2 // Příklad odeslání requestu na server
3 async function sendRequest(username, password) {
4
5     // Zadání adresy a konkrétního endpointu požadavku
6     // URL je globálně definovaná proměnná IP adresy
7     // kde server právě běží
8     let requestUrl = url + "/process";
9
10    // Inicializace těla requestu
11    let body = new URLSearchParams({
12        'username': username,
13        'password': password
14    });
15
16    // Odeslání požadavku
17    await fetch(requestUrl, {method:"POST", body:body})
18        .then(response => {
19        // Je odpověď OK (200)?
20        if (response.status !== 200) {
21            alert("Došlo k výjimce");
22            return;
23        }
24
25        // Přesměrování na jinou stránku
26        // po úspěšné a správné odpovědi serveru
27        if (response.redirected) {
28            window.location.href = response.url;
29        }
30    });
31 }

```

Zdrojový kód 2: Ukázka odeslání požadavku pomocí Fetch API

5.2.6 Přihlašovací systém a možnost zachování přihlášení

Po zadání přihlašovacích údajů uživatelem do příslušného formuláře bude vytvořen požadavek, který bude s těmito údaji odeslán na server. Pokud jsou údaje správné, bude klientovi navrácena session s jeho daty a bude přeměřován na úvodní stránku. Pokud jsou údaje špatné, server navrátí výjimku číslo 401 - UNAUTHORIZED. Toto přihlášení je aktivní do té doby, dokud se uživatel nevytáhne prohlížeč nebo se neodhlásí.

Při zaškrtnutí checkboxu v přihlašovacím formuláři „*Zachovat přihlášení*“ bude klientovi při úspěšném přihlášení odeslána cookie, která bude obsahovat verifikační token pro tohoto uživatele. Pokud tedy uživatel ukončí spojení a opět navštíví stránky webové aplikace, bude tento token použit k jeho autorizaci a bude opět přihlášen bez nutnosti opět zadávat email a heslo. Tato cookie je odstraněna, pokud se uživatel odhlásí.

5.2.7 Synchronizace soutěží s Android klientem

Při požadavku android klienta na synchronizace obdrží server od klienta tyto položky:

- token – Identifikační token uživatele přihlášeného na Android klientovi.
- ids – Všechny id soutěží, které má uživatel právě stažené.
- stageStats – Všechny statistiky všech soutěží, které patří přihlášenému uživateli.
- disqualifications – Všechny diskvalifikace ze všech soutěží, které jsou u klienta přítomny.

Server ověří token a spojí si s ním uživatele. Následně proběhne kontrola, zda uživatel má oprávnění pro dané soutěže. Je nutné tuto kontrolu udělat, aby se vynalézavý uživatel nepokoušel podstrkávat statistiky a diskvalifikace do cizích soutěží.

Všechny tyto údaje jsou doručeny v formátu JSON. Server pomocí Parser tříd získá data z JSONů. Pokud je výše zmíněná kontrola v pořádku, server vloží (existující aktualizuje) přijaté statistiky do databáze. Následně je aktualizován stav diskvalifikací. Jako poslední krok je kontrola vyhodnocení všech soutěží, které jsou přítomny u Android klienta. Soutěž je považována za vyhodnocenou, pokud každý nediskvalifikovaný účastník má záznam statistiky u každé stage. Například soutěž s patnácti účastníky a třemi stagemi je vyhodnocená v momentě, kdy od klienta přijde 45 statistik (45 instancí třídy stageStat). Pokud má soutěž diskvalifikované účastníky, je počet potřebných statistik snížen. Pokud má soutěž tři diskvalifikované účastníky a tři stage, je počet potřebných statistik snížen o 9. Již existující statistiky diskvalifikovaných účastníků nejsou počítány.

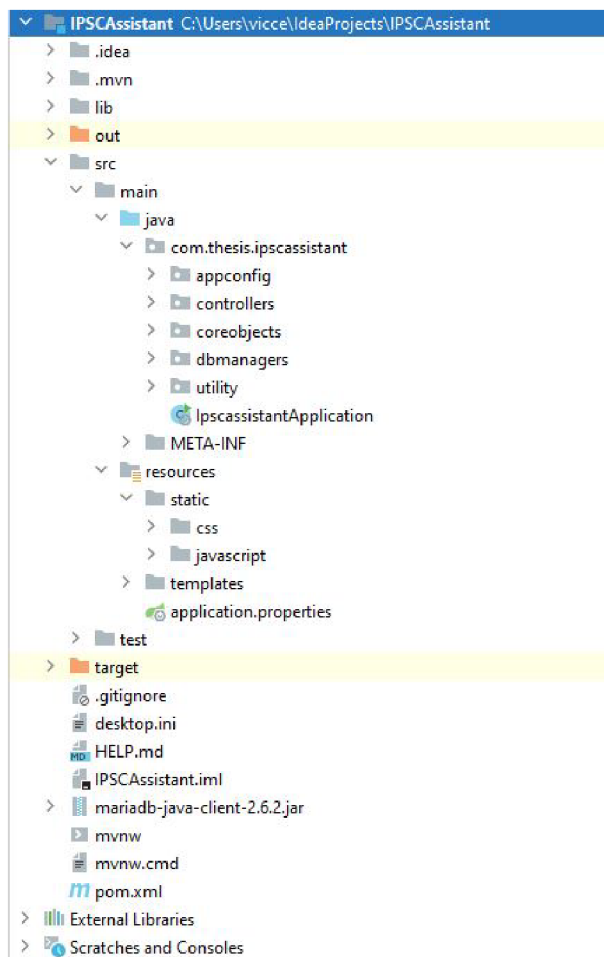
Ve chvíli, kdy má soutěž potřebný počet statistik, je označena jako vyhodnocená a zobrazí se na webové stránce všem zúčastněným uživatelům (a rozhodčímu) v tabulce historie soutěží na úvodní stránce webu. Uživatel, který byl dané soutěže součástí, si může zobrazit její výsledky.

Klientovi je odeslána odpověď v podobě instance třídy `AndroidSynchronizeReponse`. Klientovi jsou odeslány aktuální informace o soutěžích, které vlastní. Dále data o uživatelích přihlášených na soutěže aktuálně přihlášeného uživatele (tedy rozhodčího), registrace účastníků (tedy jeden účastník může být přihlášen na více soutěžích aktuálně přihlášeného uživatele), data o stagích, data o diskvalifikacích a existující statistiky těchto soutěží.

Data jsou následně vložena do databáze Android aplikace. Aplikace si následně tyto data zpracuje a složí pomocí SQL dotazů do odpovídajících tříd a objektů určených pro správnou funkci aplikace.

5.2.8 Struktura projektu

Na obrázku 12 můžeme vidět strukturu projektu webového serveru. Jako vývojové prostředí bylo využito IntelliJ IDEA. V následujících odstavcích popíšeme jednotlivé balíčky obsahující třídy zajišťující správnou funkci mobilní aplikace.



Obrázek 12: Struktura projektu webové aplikace

- appconfig – Třídy obsahující konfigurace frameworku Spring.
- controllers – Třídy představující jednotlivé controllery. Obsahuje i controllery pro zpracovávání dotazů z Android klienta.
- coreobjects – Třídy představující jednotlivé části soutěží.
- dbmanagers – Třídy určené pro práci s databází.
- utility – Třídy určené pro dekódování dat z JSONů.
- static – Pevně dané zdroje pro webovou stránku. Obsahuje CSS stylování a několik JavaScript souborů.

- templates – Obsahuje HTML šablony pro zpracování technologií Thymeleaf.
- pom.xml – Obsahuje centrální seznam knihoven a nastavení pro Maven.

5.3 Aplikace pro Android

5.3.1 Knihovna Volley

Jedná se o knihovnu zajišťující rozhraní pro přijímání a odesílání Http požadavků na server. Třídy jsou obsaženy v balíčku RequestServices a obstarávají odesílání a přijímání HTTP požadavků. Požadavky opět fungují asynchronně. V momentě, kdy je doručena odpověď ze serveru, je původní volající třídě přes callback delegováno, že požadavek je dokončený. Volající třída si nyní odpověď zpracuje.

5.3.2 Přihlašovací systém

Aplikace pro android je určena jako pomůcka pro rozhodčího soutěže. Střelnice, kde se tyto soutěže konají, jsou velmi často na odlehlých místech nebo v případě podzemních střelnic několik metrů pod zemí. Signál tedy nemusí být vždy dostupný. Aplikace je zamýšlena tak, aby se obešla bez internetového připojení. Internetové připojení je potřebné pouze na přihlášení uživatele a na synchronizaci stavu soutěží se serverem na požadavek uživatele.

Pro autorizaci požadavku je použit výše zmíněný android `_login_token`, který uživatel od serveru obdrží při úspěšném přihlášení. Každá synchronizace se verifikuje tímto tokenem.

Po spuštění aplikace je vždy kontrolováno, jestli jsou přítomna data o přihlášeném uživateli. Pokud ne, je zobrazena přihlašovací obrazovka. Pokud ano, je uživatel přesunut do hlavního přehledu svých soutěží (hlavního menu).

5.3.3 Android ROOM

Android ROOM je abstrakční vrstva nad SQLite databází. Veškeré trvalé ukládání dat je vykonáváno skrze tuto technologii. ROOM se velmi dobře používá, protože ho lze pomocí anotací relativně jednoduše ovládat. Třída `AppDatabase` obsahuje konfiguraci ROOMu. V této třídě jsou nadefinovány entity, se kterými se bude pracovat. Dále se definují `TypeConverter` třídy. ROOM totiž dokáže ukládat pouze čísla a textové řetězce. Takže například pro uložení data nebo času je nutné definovat tzv. „`TypeConverter`“. Příklad zdrojového kódu pro `TypeConverter` třídy `LocalTime` můžeme vidět ve zdrojovém kódu [3](#)

```

1 public class LocalTimeTypeConverter {
2     @TypeConverter
3     public static LocalTime toLocalTime(String localTimeString) {
4         if (localTimeString == null) {
5             return null;
6         } else {
7             return LocalTime.parse(localTimeString);
8         }
9     }
10
11     @TypeConverter
12     public static String toLocalTimeString(LocalTime localTime) {
13         if (localTime == null) return null;
14         else return localTime.toString();
15     }
16 }

```

Zdrojový kód 3: Ukázka kódu vlastního TypeConverteru

Konkrétní dotazy jsou definovány v tzv. „DAO“ interface třídách. DAO je rozhraní pro databázi. Umožňuje definovat operace bez toho, aby byla odhalena samotná databáze. Všechny používané DAO abstraktní třídy musí být deklarovány v AppDatabase třídě. Příklad zdrojového kódu DAO třídy můžeme vidět [zde 4](#).

```

1 @Dao
2 public interface LoginTokenDAO {
3
4     @Query("SELECT * FROM login_tokens")
5     public List<LoginToken> getAllLoginTokens();
6
7     @Insert(onConflict = OnConflictStrategy.REPLACE)
8     void insertLoginToken(LoginToken loginToken);
9
10    @Query("DELETE FROM login_tokens")
11    void removeAllTokens();
12
13 }

```

Zdrojový kód 4: Příklad DAO abstraktní třídy pro ukládání a získávání třídy login_token z databáze

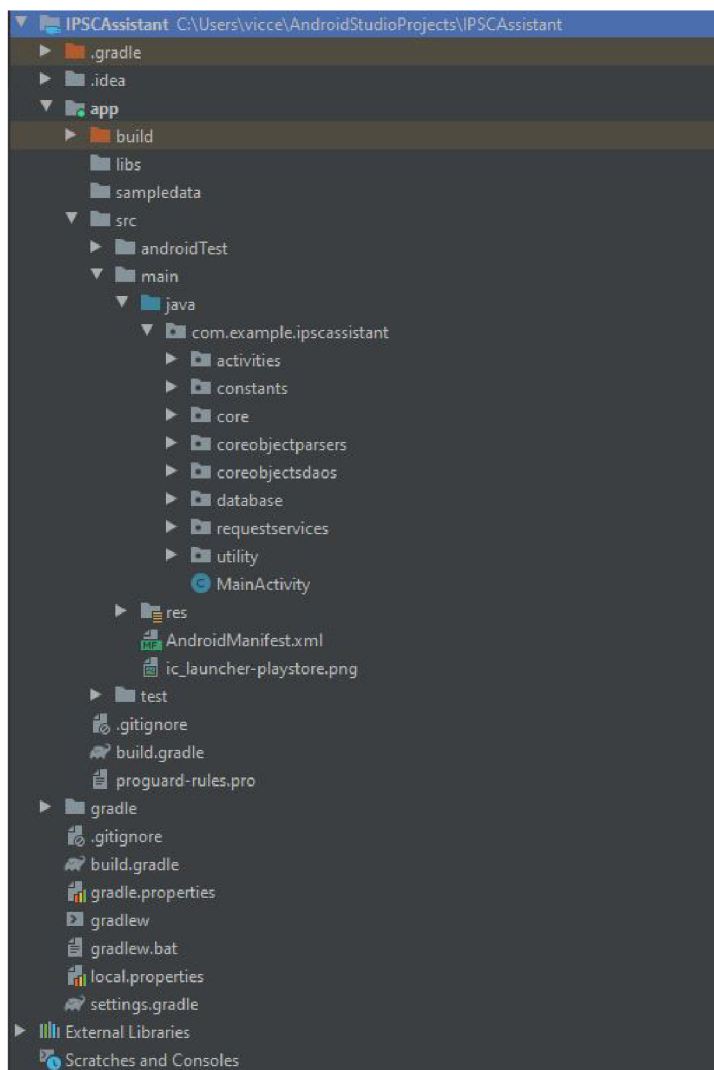
Volání těchto dotazů pro získání konkrétních dat mají na starosti třídy s koncovkou DBExecutor. Dotazy nelze volat z hlavního vlákna, protože hrozí zaseknutí UI dokud nebude dotaz dokončen. Pro zavolání dotazu ve vedlejším vlákně jsou využity třídy ExecutorService a Callable. Pomocí třídy ExecutorService je vytvořeno samostatné vlákno, které bude vykonávat funkci zadanou v Callable. Po dokončení výpočtu vlákna jsou data vydána hlavnímu vlákně. Příklad použití můžeme vidět v zdrojovém kódu 5

```
1 public static void insertDisqualification(final Context context,
2     final Disqualification disqualification) {
3     ExecutorService executorService = Executors.
4         newSingleThreadExecutor();
5     Callable<Boolean> callable = new Callable<Boolean>() {
6         @Override
7         public Boolean call() throws Exception {
8             AppDatabase database = AppDatabase.getDatabase(context);
9             database.disqualificationDAO().insertDisqualification(
10                disqualification);
11             return true;
12         }
13     };
14     try {
15         executorService.submit(callable).get();
16     } catch (ExecutionException e) {
17         e.printStackTrace();
18     } catch (InterruptedException e) {
19         e.printStackTrace();
20     }
```

Zdrojový kód 5: Příklad použití třídy ExecutorService a Callable pro provedení dotazu na databázi

5.3.4 Struktura projektu Android aplikace

Pro vytvoření aplikace pro Android bylo využito vývojové prostředí Android Studio. Na obrázku 13 můžeme vidět strukturu projektu. V následujících odstavcích popíši obsah a význam jednotlivých balíčků a složek.



Obrázek 13: Struktura projektu v Android studiu

- activities – Obsahuje zdrojové kódy aktivit pro ovládání Android aplikace.
- constants – Obsahuje statickou třídu s konstantami pro celý projekt.
- core – Obsahuje třídy reprezentující jednotlivé objekty klíčové pro průběh soutěží a správnou funkci aplikace.
- coreobjectparsers – Obsahuje třídy zodpovědné za kódování a dekodování formátu JSON.

- coreobjectdaos – Obsahuje DAO třídy pro „coreobjects“ třídy.
- database – Obsahuje konfigurační třídu pro ROOM databázi a třídy DBExecutor, které jsou zodpovědné za volání dotazů.
- utility – Pomocné třídy, obsahuje například adaptéry pro RecyclerView prvky nebo výše zmíněné třídy TypeConverter.
- res – Obsahuje zdroje a layouty aktivit a různých typů menu.

Závěr

Výsledkem této práce je mobilní a webová aplikace. Webová aplikace zaopatřuje administrativní část. Hlavními funkcionalitami jsou správa uživatelských účtů, zakládání soutěží, možnost se na soutěž zapsat, přehled nadcházejících soutěží a zobrazení výsledků soutěží. Vlastníkovi soutěže je na webu dále umožněno jeho soutěž upravit, zrušit a vytisknout si pomocnou dokumentaci připravenou pro danou jeho soutěž.

Mobilní aplikace slouží rozhodčímu při hodnocení soutěže v terénu. Umožní uživateli se přihlásit, synchronizovat své soutěže se serverem a uložit jejich stav na server. Po ohodnocení všech uživatelů na všech stadiích je soutěž považovaná za dokončenou a zobrazí se na webové stránce ve vyhodnocených soutěžích.

V budoucnu bych se rád věnoval vylepšení uživatelského rozhraní webové aplikace a implementaci dalších nastavitelných parametrů soutěží tak, aby byla aplikace co nejvíce univerzální.

Conclusions

The result of this thesis is mobile phone and web application. Web application fullfils the role of administrative tasks. Main features for web application are management of user accounts, login system, creation of competitions by users, the possibility to enlist and cancel enlistment, general overview of upcoming competitions and the possibility to display result of already evaluated and finished competition user was enlisted to. Furthermore, owners of competitions can manage their competitions, edit them, cancel them and print support chater for their competitions.

Android application helps the referee (the owner of the competition) to evaluate the competition at the spot of venue. User can login, sychronize his competitions with server and save the current state of competition. Competition is considered as evaluated and finished when all participants have been evaluated in all stages. Server then automatically evaluates if the competition is finished. If it is finished, participating users can see the results of the competition on webpage.

In the future, I would like to improve the user interface. Furthermore, I would like to implement additional possibilities in competition creation to make the application as universal as possible.

A Přístup k webové aplikaci

Webová aplikace bude dostupná na adrese [zde](#). Instalace webové aplikace je tedy nutná pouze v případě, pokud si přejete webovou aplikaci nainstalovat lokálně na Vaše zařízení. Doporučil bych využít již běžící prostředí, protože databáze je již naplněna ukázkovými daty.

Pokud si budete přát nainstalovat webový server na Vaše zařízení, postupujte prosím podle instrukcí obsažených v souboru *readme.txt* na DVD.

Přihlašovací údaje ukázkových uživatelů můžeme vidět níže. Samozřejmě si můžete založit i vlastní účet.

Uživatel „První Prvák“

- email: a@a.cz
- heslo: aaaaa

Uživatel „Druhý Druhák“

- email: b@b.cz
- heslo: bbbbb

bin/

Adresář obsahuje spustitelný *.jar* soubor webové aplikace. Soubor má v sobě integrovaný Tomcat server a všechny potřebné knihovny. Webový server je tedy ihned to spuštění v provozu.

doc/

Obsahuje text práce ve formátu *.pdf* vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce včetně všech příloh a všechny soubory potřebné k vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový kód textu, vložené obrázky apod.

src/

Adresář obsahuje všechny zdrojové kódy aplikace pro Android a webové aplikace.

readme.txt

Instrukce pro instalaci a spuštění webové aplikace a instalaci Android aplikace.

install/

Adresář obsahuje instalační soubor Android aplikace, instalátor databázového systému MariaDB a instalátor JRE 16.

Literatura

- [1] *Java*. Dostupný z: [⟨https://java.com/en/download/help/whatis_java.html⟩](https://java.com/en/download/help/whatis_java.html).
- [2] *Spring*. Dostupný z: [⟨https://spring.io/⟩](https://spring.io/).
- [3] *Maven*. Dostupný z: [⟨https://maven.apache.org/⟩](https://maven.apache.org/).
- [4] *Thymeleaf*. Dostupný z: [⟨https://www.thymeleaf.org/⟩](https://www.thymeleaf.org/).
- [5] *MariaDB*. Dostupný z: [⟨https://mariadb.com/⟩](https://mariadb.com/).
- [6] *Fetch API*. Dostupný z: [⟨https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API⟩](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API).
- [7] *Volley*. Dostupný z: [⟨https://developer.android.com/training/volley⟩](https://developer.android.com/training/volley).
- [8] *ROOM*. Dostupný z: [⟨https://developer.android.com/training/data-storage/room⟩](https://developer.android.com/training/data-storage/room).