



## ZADÁNÍ BAKALÁ SKÉ PRÁCE

<b>Název:</b>	Score table - Android
<b>Student:</b>	Michal Klusák
<b>Vedoucí:</b>	Ing. Ji í Chludil
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2017/18

### Pokyny pro vypracování

Cílem práce je vytvo ení frameworku sloužícího k tvorb aplikací pro zaznamenávání stavu kolektivních sportovních zápas .

1. Analyzujte pravidla pro hokej, fotbal a tenis (dvou/ ty hra). Pravidla se mohou lišit podle typu sout že. Dop esn te vlastnosti v tomto kontextu používaných entit: Game, Timer, Round, Contestant (Team/Person), Rules, Penalty, Advantage.
2. Navrhn te:
  - zapouzd ení entit a jejich kolekcí,
  - GUI pro modul fotbal,
  - RSS kanál pro integraci s jinými systémy (nap . web).
3. Implementujte:
  - pln funk ní aplikaci pro fotbal,
  - prototyp pro tenis a hokej.
4. Aplikaci podrobte vhodným test m.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.  
d kan

V Praze dne 9. února 2017



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



Bakalářská práce

## **Score table - Android**

*Michal Klusák*

Vedoucí práce: Ing. Jiří Chludil

9. května 2017



---

## Poděkování

Rád bych poděkoval vedoucímu mé práce, Ing. Jiřímu Chludilovi, za vedení a kontrolu mé bakalářské práce a Michalu Maněnovi, za rady při vytváření aplikace.



---

## Prohlášení

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

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 9. května 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Michal Klusák. Všechna práva vyhrazena.

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

### **Odkaz na tuto práci**

Klusák, Michal. *Score table - Android*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.



---

# Abstrakt

Tato bakalářská práce se zabývá vytvořením frameworku (šablony) a jeho následnému použití v mobilní aplikaci pro zaznamenávání aktuálního stavu kolektivních sportovních zápasů pro operační systém Android.

Nejprve jsem analyzoval několik sportů typu soupeř proti soupeři (např. fotbal, hokej, tenis atd.) a jejich společných vlastností, a poté jsem rozmyslel vlastnosti budoucích entit, z nichž bude výsledný celek složen. Vše následoval důkladný návrh těchto entit a také návrh grafického rozložení prvků na obrazovce zařízení pro fotbal. Na základě analýzy a návrhu byla provedena implementace již zmíněného frameworku, jeho následné využití při implementaci plně funkční aplikace pro fotbal a prototypů pro tenis a hokej. Nakonec bylo vše podrobena vhodným testům kvůli objevení případných chyb.

Výsledkem je tedy funkční mobilní aplikace sloužící k zaznamenávání stavu zápasu fotbalu a již zmíněné prototypy. Tato data lze následně posílat na server k případnému přenášení informací na jakékoliv zařízení. To umožní uživateli aplikace poskytnout aktuální informace o průběhu utkání komukoliv na světě na jakékoliv zařízení s připojením k internetu. Vytvořený framework však lze v budoucnu použít k implementaci aplikace i pro jiné sporty s rozdílnými pravidly.

**Klíčová slova** Mobilní aplikace, Android, Android Studio, Java, framework, sdílení stavu utkání, sportovní zápas, fotbal, hokej, tenis, JUnit, UI Automator, Firebase autentizace

# Abstract

This bachelor thesis shows the creation of a framework (template) and its usage in the mobile application to record current status of collective sports match for Android operating system.

At first I analyzed several player versus player types of sports (e.g. football, hockey, tennis, etc.) and their common features, and then I thought about properties of future entities from which the final product will be composed. After that followed a thorough design of these entities as well as the design of a graphical layout of the elements on screen of the device for football. Based on analysis and design, I implemented already mentioned framework, its subsequent use in implementation of a fully functional application for football and prototype application for tennis and hockey. Everything was subjected to appropriate tests to detect any errors.

The result is a functional mobile application for recording the status of football match and already mentioned prototypes. This data can then be sent to the server to transfer information to any device. This will allow the user of the application to provide up-to-date information on the course of the match to anyone in the world on any device with an internet connection. However, the framework can be used in the future, to implement the application for other different sports with different rules.

**Keywords** Mobile application, Android, Android Studio, Java, framework, sharing state of play, sport match, football, hockey, tennis, JUnit, UI Automator, Firebase authentication

---

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Analýza</b>	<b>5</b>
2.1 Porovnání s konkurencí . . . . .	5
2.2 Pravidla sportů . . . . .	8
2.3 Verze operačního systému . . . . .	15
2.4 Programovací jazyk . . . . .	17
2.5 Funkční a nefunkční požadavky . . . . .	17
2.6 Přihlašování uživatelů . . . . .	19
2.7 Případy užití a diagramy aktivit . . . . .	19
<b>3 Návrh</b>	<b>25</b>
3.1 Doménový model . . . . .	25
3.2 Diagram tříd . . . . .	27
3.3 Návrh uživatelského prostředí . . . . .	27
3.4 Odesílání změn . . . . .	33
<b>4 Realizace</b>	<b>37</b>
4.1 Diagram nasazení . . . . .	37
4.2 Měření času . . . . .	37
4.3 Práva aplikace . . . . .	38
4.4 Lokalizace . . . . .	39
4.5 Instalační manuál . . . . .	40
4.6 Uživatelský manuál . . . . .	41
4.7 Výsledný vzhled aplikace . . . . .	43
<b>5 Testování</b>	<b>45</b>
5.1 Testování programátorem . . . . .	45

5.2	Jednotkové testy . . . . .	45
5.3	Testování uživatelského prostředí . . . . .	46
5.4	Uživatelské testování . . . . .	50
<b>Závěr</b>		<b>55</b>
	Možnosti rozšíření aplikace . . . . .	55
<b>Literatura</b>		<b>57</b>
<b>A Seznam použitých zkratk</b>		<b>59</b>
<b>B Obsah příloženého CD</b>		<b>61</b>
<b>C Výsledný vzhled aplikace</b>		<b>63</b>
	C.1 Smartphone . . . . .	63
	C.2 Tablet . . . . .	65
<b>D Testovací scénáře pro uživatelské testování</b>		<b>71</b>
	D.1 1. testovací scénář . . . . .	71
	D.2 2. testovací scénář . . . . .	72
	D.3 3. testovací scénář . . . . .	72
<b>E Vstupní dotazník pro uživatelské testování</b>		<b>75</b>
<b>F Výstupní dotazník pro uživatelské testování</b>		<b>77</b>

---

## Seznam obrázků

2.1	Momentka z fotbalového utkání a výsledková digitální tabule . . .	9
2.2	Hokejové utkání a digitální kostka zavěšená nad ledem zobrazující skóre zápasu . . . . .	11
2.3	Loga porovnávaných hokejových turnajů . . . . .	12
2.4	Tenisové utkání s ukazatelem aktuálního skóre na pozadí . . . . .	13
2.5	Loga porovnávaných tenisových turnajů . . . . .	14
2.6	Diagram rozdělení podílu jednotlivých verzí systému Android . . .	16
2.7	Google Firebase . . . . .	20
2.8	Diagram případů užití frameworku . . . . .	21
2.9	Diagram případů užití aplikace pro fotbal . . . . .	23
3.1	Doménový model frameworku . . . . .	26
3.2	Diagram tříd frameworku . . . . .	28
3.3	Návrh hlavní obrazovky aplikace pro fotbal – základní pohled . . .	29
3.4	Návrh hlavní obrazovky aplikace pro fotbal – přihlášený uživatel .	29
3.5	Návrh hlavní obrazovky aplikace pro fotbal – přihlášení . . . . .	30
3.6	Návrh obrazovky nastavení aplikace pro fotbal – horní část . . . .	30
3.7	Návrh obrazovky nastavení aplikace pro fotbal – dolní část . . . .	31
3.8	Návrh obrazovky zápasu aplikace pro fotbal – základní hrací doba	31
3.9	Návrh obrazovky zápasu aplikace pro fotbal – prodloužení . . . . .	32
3.10	Návrh obrazovky zápasu aplikace pro fotbal – penaltový rozstřel .	32
3.11	Návrh obrazovky zápasu aplikace pro fotbal – konec zápasu . . . .	33
3.12	Relační model databáze . . . . .	34
4.1	Diagram nasazení . . . . .	38
4.2	Nastavení práv aplikace . . . . .	39
5.1	Logo Android Testing Support Library . . . . .	47
5.2	UI Automator Viewer . . . . .	48

C.1	Výsledný vzhled aplikace na telefonu – hlavní obrazovka a nastavení lokalizace . . . . .	63
C.2	Výsledný vzhled aplikace na telefonu – nastavení zápasu . . . . .	64
C.3	Výsledný vzhled aplikace na telefonu – zápas (základní hrací doba)	64
C.4	Výsledný vzhled aplikace na telefonu – zápas (penalty) . . . . .	65
C.5	Výsledný vzhled aplikace na telefonu – konec zápasu . . . . .	65
C.6	Výsledný vzhled aplikace na tabletu – hlavní obrazovka . . . . .	66
C.7	Výsledný vzhled aplikace na tabletu – nastavení lokalizace . . . . .	66
C.8	Výsledný vzhled aplikace na tabletu – nastavení zápasu . . . . .	67
C.9	Výsledný vzhled aplikace na tabletu – zápas (základní hrací doba)	68
C.10	Výsledný vzhled aplikace na tabletu – zápas (penalty) . . . . .	68
C.11	Výsledný vzhled aplikace na tabletu – konec zápasu . . . . .	69

---

## Seznam tabulek

2.1	Shrnutí hodnocení konkurenčních produktů . . . . .	8
2.2	Shrnutí délky zápasů různých věkových kategorií v českém fotbale	9
2.3	Tabulka rozdělení podílu jednotlivých verzí systému Android ze dne 6.3.2017 . . . . .	16





---

# Úvod

Doby, kdy jsme hledali výsledky sportovních zápasů na teletextu, čekali na televizní zprávy nebo si je druhý den přečetli v novinách, jsou nenávratně pryč. Průměrná doba, než se k nám jakákoliv změna v zápase dostane ihned po tom, co ve skutečnosti nastane, se dnes pohybuje ve vteřinách. Ať už sledujeme jakýkoliv sport nebo fandíme jakémukoliv týmu, ne vždy máme z mnoha důvodů možnost vidět svůj oblíbený tým naživo. Někomu stačí zjistit konečný výsledek, ale někdo se rád dozví, že padl gól ihned po tom, co ve skutečnosti padl, i když aktuálně nemá možnost utkání sledovat nebo na něm přímo být. V takovém případě mu stačí pouze chytrý telefon, připojení k internetu a ta správná aplikace, která ho zásobuje aktuálními informacemi z utkání. S takovými aplikacemi se v současnosti roztrhl pytel a některé z nich jsou velmi kvalitní. Ovšem většina má jednu společnou vlastnost – jsou spravovány a informacemi doplňovány svým tvůrcem, případně lidmi, které tvůrce aplikace zaměstnává. Uživatel většinou nemá možnost tyto údaje jakkoliv ovlivnit.

Chci vytvořit aplikaci, která bude umožňovat jakémukoliv uživateli sdílet jakýkoliv zápas. Jelikož je nemožné vytvořit aplikaci, která by byla natolik obecná, aby pokrývala více různých sportů, vytvořím po analýze sportů obecný framework, který posléze využiji k tvorbě aplikací pro několik různých sportů.



---

## Cíl práce

Cílem mé bakalářské práce je vytvořit framework<sup>1</sup>, který následně použiji pro naprogramování funkční mobilní aplikace pro fotbal a prototypů pro hokej a tenis. Celá tato tvorba se řídí klasickými postupy vývoje softwaru v softwarovém inženýrství. Výsledný framework lze v budoucnu použít k naprogramování aplikace pro libovolný sport typu *versus*, tedy kolektivní sport, kde se vždy utkají proti sobě dva týmy či dva jednotlivci. Následná aplikace tedy umožňuje uživateli zaznamenávat průběh utkání, které právě sleduje a komukoliv tento obsah přes internet zprostředkovávat.

V praxi to tedy znamená, že si libovolný uživatel do svého zařízení stáhne aplikaci pro daný sport, nastaví potřebné parametry zápasu (týmy, typ utkání, počáteční skóre atd.), vygeneruje webový odkaz sloužící pro sledování průběhu utkání, komukoliv, kdo bude chtít zápas sledovat pošle a dále už jen pomocí ovládacích prvků aplikace zápas zaznamenává.

Bakalářskou práci lze rozdělit na sběr požadavků (funkční a nefunkční požadavky), analýzu, návrh, realizaci a testování. Všechny tyto fáze se týkají jak frameworku, tak samotných aplikací.

---

<sup>1</sup>Framework je softwarová struktura (šablona), která se skládá z knihoven a funkcí a má jasná interní pravidla. Používá se pro zjednodušení vývoje softwaru. Funguje jako černá krabice, jejíž funkce využívá programátor, aniž by věděl, jak přesně uvnitř funguje. Většinou programátor využívá framework, který napsal někdo jiný, který obsahuje funkce, které on sám potřebuje a nechce se zdržovat jejich psaním, když je již někdo vytvořil dříve.



---

# Analýza

První fází každého cyklu vývoje softwaru je jeho analýza. V této fázi je důležitý kontakt se zákazníkem/zadavatelem. S ním si programátor, případně celý tým, sjednotí zadání a cíl softwaru, všechny jeho funkce a ostatní neméně důležité detaily, jako např. cílovou platformu. V praxi se velmi často opomíjí, to však může vést k nesprávnému pochopení zadání a tedy vytvoření něčeho, co se neprotíná s představou zadávajícího. V konečné fázi dochází kromě nespokojenosti zákazníka k finančním ztrátám a znehodnocení části již provedené práce.

## 2.1 Porovnání s konkurencí

V dnešní době exponenciálního rozmachu mobilních technologií a aplikací lze usoudit, že již může existovat software podobný tomu, který se chystám vytvořit. Je tedy důležité zjistit vše o přímé konkurenci, přijít s konkurenční výhodou a případně si od konkurence něco vypůjčit nebo se od ní v mnohém poučit.

### 2.1.1 Kritéria porovnání

Zvolím několik kritérií, podle kterých budu porovnávané produkty hodnotit. Každé kritérium bude ohodnoceno body, přičemž více bodů znamená lepší skóre.

**Možnost uživatele ovlivnit obsah** Alfa a omega mojí práce, která je na možnosti uživatele ovlivnit výsledný obsah založena. Jestliže toto bude uživateli umožněno, dostane produkt tři body. V opačném případě méně.

**Kvalita obsahu** Kvalita celého obsahu je bezpochyby nezanedbatelná, proto je nutné klást důraz i na ni. Hodnoceno 0-2 body.

**Textový přenos** Sdílení obsahu ve formě textu je opět pro porovnání velmi důležité, jelikož na rozdíl od přenosu videa šetří nároky na datový přenos. Kritérium bude hodnoceno třemi body za textový obsah. Jinak nula body.

**Mobilní aplikace** Pokud bude služba nabízet mobilní aplikaci, dostane jeden bod.

### 2.1.2 Livesport

Zřejmě nejznámější a nejstahovanější sportovní aplikace<sup>2</sup> v České republice, která má veliký ohlas i v zahraničí. Poskytuje živé informace o desítkách zápasech z desítek různých lig denně. Zprvu služba fungovala jen jako webová stránka, ale postupem času byly vytvořeny mobilní aplikace pro Android a iPhone/iPad. Nabízí mnoho sportů, včetně těch, na které je zaměřena má práce.

Jednotlivé zápasy si v aplikaci (i na webu) lze uložit do *Mých zápasů* a nastavit si, na co všechno chci být upozorňován. Je možné být upozorňován na změnu skóre, začátek zápasu, červenou kartu, konec zápasu atd. Dále aplikace umožňuje si vybrat oblíbenou soutěž, aby byla navrchu v přehledu soutěží či si zvolit svůj oblíbený tým, na který bude uživatel upozorňován pokaždé na vše, co si nastaví. Všechno nastavení lze uskutečnit pod přihlášeným účtem, aby bylo možno nastavení přenést na aplikaci v jiném zařízení či na jiné platformě. Kromě samotného průběhu zápasu poskytuje informace o sestavách a hráčích, aktuální kurzy, tabulky jednotlivých soutěží a další podrobnosti. Aplikace je na Androidu momentálně ve verzi 2.16.0. Je velmi často aktualizována a doplňována o nové funkce a je poměrně stabilní.

Data jsou do aplikace zadávána zaměstnanci firmy (případně kupována). Pouze Livesport tedy rozhoduje, které zápasy budou obsaženy a které ne a uživatel tuto volbu neovlivní. Výhodou tohoto řešení je výsledná kvalita dat. Služba tedy dostává dva body za kvalitu obsahu, ale nula bodů za možnost uživatele ho ovlivnit. K dispozici je pouze textový přenos, toto tedy hodnotím třemi body. V poslední řadě dostává bod za mobilní aplikaci.

### 2.1.3 Onlajny

Český web<sup>3</sup> s dlouholetou historií. Poskytuje výpis ve formě rozsáhlého textu (tři body), který popisuje dění na hřišti. Mnohdy tedy podává detailnější informace, avšak za cenu pomalejších aktualizací během zápasu.

Nemá nativní aplikaci pro chytré telefony (nula bodů), ale disponuje webovou stránkou, která je přizpůsobená i pro menší displeje a obsahuje jednodušší ovládání. To zajišťuje multiplatformnost (lze používat na jakémkoliv zařízení) za cenu stahování většího počtu dat při aktualizacích.

---

<sup>2</sup><http://www.livesport.cz/>

<sup>3</sup><http://www.onlajny.com/>

Uživatel má možnost částečně obsah ovlivnit. Je možné se elektronickou komunikací domluvit na spolupráci a pravidelně poskytovat průběh zápasů dohodnutého týmu. Na stránce se nachází výčet měst a týmů, které nejsou dosud pokryty. Není tím však zaručena kvalita textu ani jeho korektnost. Za možnost ovlivnění ze strany uživatele si web zaslouží dva body, avšak jen jeden bod za kvalitu obsahu.

Přehled zápasu je kromě textu doplněn i o sestavy, statistiky a další ukazatele. Lze si přepnout zobrazování textového výpisu pouze na důležité situace, branky, karty či jen na střídání. U prestižnějších zápasů bývá text doplněn o videa popisovaných situací.

### 2.1.4 Ustream, Youtube live, Livestream

Profesionální streamovací služby<sup>4 5 6</sup>, které obsahují velmi mnoho funkcí. Jako uživatel si musíte většinu funkcí zaplatit a ceny jsou obvykle velmi vysoké. Jestliže se Vám za službu platit nechce, funkcí zdarma je málo a mohou být omezené. Tyto služby jsou navíc zaměřeny spíše na přenášení audia a videa, což není cílem mé aplikace. Nejde tedy o přímou konkurenci, ale obsahují i prvky využitelné v mé práci. Například tzv. *Channels*, neboli kanály. Uživatel si vytvoří svůj vysílací kanál, na který může streamovat svůj obsah a tento kanál poté nastavit jako veřejný s tím, že k němu může mít přístup kdokoliv (může se zobrazit na hlavní stránce v listu kanálů) a nebo ho lze nastavit jako soukromý a poslat odkaz jen tomu, komu bude chtít. Zároveň může na kanálu shromažďovat záznamy z minulých zápasů a nemusí pro každý zápas vytvářet nový link.

Služby dostávají dva body za možnost ovlivnit obsah (bod dolů za placené funkce), dva body za kvalitu obsahu, bod za mobilní aplikace, ale nula bodů za textový přenos.

### 2.1.5 Sopcast

Služba<sup>7</sup>, která je zcela zdarma a nabízí tzv. *P2P živé vysílání*, neboli že zveřejňovaný obsah neputuje přes žádný server, ale pouze od klienta přímo ke klientovi. Výhodou je, že s rostoucím množstvím uživatelů celková dostupná přenosová kapacita roste. [1]

Bohužel i tato služba funguje spíše pro přenos audia a videa a textový přehled vůbec neumožňuje.

Je tedy ohodnocena třemi body za neomezené možnosti ze strany uživatele, dvěma body za kvalitu obsahu, ale nula body za nemožnost sdílet textově. Služba nabízí aplikaci pro Android, dostává tedy ještě bod.

---

<sup>4</sup><http://www.ustream.tv/>

<sup>5</sup><https://www.youtube.com/>

<sup>6</sup><https://livestream.com/>

<sup>7</sup><http://www.sopcast.com/>

## 2. ANALÝZA

---

Kritérium	Livesport	Onlajny	Ustream, ..	Sopcast
Rozsah ovlivnění	0	2	2	3
Kvalita obsahu	2	1	2	2
Textový přenos	3	3	0	0
Mobilní aplikace	1	0	1	1
<b>Celkem</b>	<b>6</b>	<b>6</b>	<b>5</b>	<b>6</b>

Tabulka 2.1: Shrnutí hodnocení konkurenčních produktů

### 2.1.6 Závěr porovnání

Konkurenční služby v hodnocení dopadly velmi podobně, avšak žádnou přímou konkurenci k mé aplikaci jsem mezi nimi nenalezl. Nelze jednoznačně prohlásit, že taková neexistuje, ale když jsem se k ní nebyl schopen pomocí svých prostředků dostat, zřejmě není tolik v povědomí internetu a jeho uživatelů. To mi dává jistou výhodu či naději, že by má aplikace mohla vyslyšet jistý typ poptávky po tomto typu produktu. Celé hodnocení je shrnuto v tabulce 2.1.

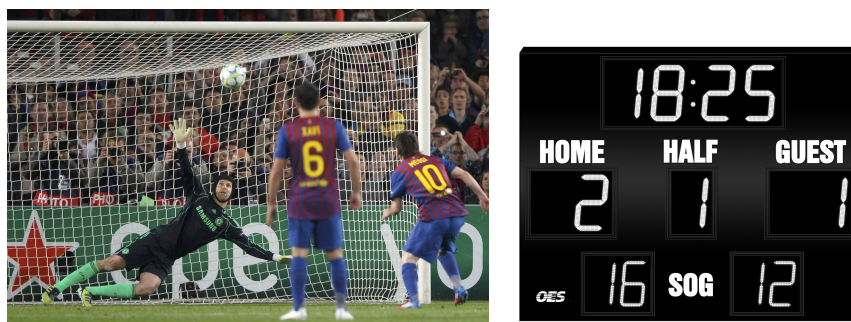
## 2.2 Pravidla sportů

Jedním z cílů této práce je vytvořit framework, který bude schopen pokrýt většinu sportů, nezávisle na pravidlech. Proto postupně analyzuji všechny tři základní sporty, ke kterým později s jeho použitím vytvořím jednotlivé aplikace. Tyto tři následující sporty se v pravidlech a struktuře navzájem liší, proto jsou dobrými kandidáty na otestování funkčnosti frameworku. U většiny sportů lze nalézt více variant, které existují z důvodu např. cílové věkové skupiny (děti nehrají fotbalový zápas 90 minut jako dospělí), kontinentu, země či destinace, kde je sport provozován nebo pohlaví (ženy hrají v tenise zápas na méně setů než muži). Je důležité počítat se všemi variantami. Pro přesnost zde ještě připomínám, že práce pokrývá pouze sporty typu *versus*. Tedy sporty, ve kterých se vždy utkávají dva protivníci mezi sebou, ať už je protivníkem myšlený tým, klub či jednotlivec. Zároveň chci dodat, že se budu rozepisovat pouze o takových sportovních pravidlech, které moje práce pokrývá. Tyto sporty disponují velkým množstvím pravidel a rozebírání těch nepodstatných by bylo zbytečně zdlouhavé.

### 2.2.1 Fotbal

Začnu analýzou pravidel fotbalu, jelikož jde o hlavní testovací sport pro mou bakalářskou práci. Zároveň si dovoluji říci, že z hlediska nároků, které fotbal klade na framework, jde o sport nejméně náročný. Navíc s ním mám nejvíce osobních zkušeností.





Obrázek 2.1: Momentka z fotbalového utkání a výsledková digitální tabule

Kategorie	Počet period	Doba periody [min]
Muži	2	45
Starší dorost	2	45
Mladší dorost	2	40
Starší žáci	2	35
Mladší žáci	2	35
Starší přípravka	3	17
Mladší přípravka	3	15

Tabulka 2.2: Shrnutí délky zápasů různých věkových kategorií v českém fotbale

Obecně trvá základní hrací doba fotbalového zápasu 90 minut, neboli dva poločasy o délce 45 minut. Tato doba se může měnit na základě věkové kategorie (rozeberu později) a délce nastavení. Nastavení je doba, která přichází vždy na konci každého z poločasů. O její délce rozhoduje rozhodčí zápasu a obvykle trvá mezi 0 až 5 minutami. Může však být i delší. Tato doba reflektuje události během zápasu, které mohli každou z polovin nějakým způsobem pozdržet – ošetřování hráče, střídání, případně výpadek osvětlení na stadionu. Důvod existence nastavení je prostý, ve fotbale je zaveden hrubý čas. To znamená, že se při jakémkoliv přerušení čas nezastavuje. Dojde-li v průběhu k výrazné prodlevě, rozhodčí si jí zapamatuje a na konci periody tyto prodlevy sečte a nastaví.

Jak jsem již zmiňoval výše, každý ze sportů se provozuje ve více věkových kategoriích. Klasicky se fotbal dělí na přípravku, žáky, dorost a muže. Toto rozdělení je použito i u dalších dvou zkoumaných sportů, nebudu ho tedy již dále vyjmenovávat. Pro představu zde bude stačit, když porovnám jednotlivé věkové kategorie pouze v Česku a u zahraničních soutěží mezi sebou budu porovnávat jiné vlastnosti a jen v dospělé kategorii.

Z obsáhlé tabulky na webu [2] jsem sestavil jednodušší tabulku 2.2, obsahující pouze pro mě potřebné informace. Lze z ní vyčíst, že čím nižší věková kategorie, tím vyšší počet period a naopak nižší délka jedné periody.

## 2. ANALÝZA

---

Zatím jsem zmiňoval pouze základní hrací dobu, ale v některých soutěžích, většinou vyřazovacích duelech, se při rovnosti skóre obou soupeřů pokračuje jedním ze dvou scénářů:

1. Penaltový rozstřel – Toto je jediná fáze, která není nijak časově omezena. V této chvíli se čas už vůbec neměří. Penaltový rozstřel, neboli penalty, probíhá tak, že se vždy střídají oba týmy v pokutových kopech<sup>8</sup> na brankáře svého soupeře. Takto oba týmy odehrají každý 5 pokutových kopů a porovnají, kdo jich proměnil více. Ten tým vyhrává. V případě rovnosti pak dále kope každý tým vždy po jedné penaltě čekajíc na chybu soupeře až do rozhodnutí.
2. Prodloužení (a případný penaltový rozstřel) – Prodloužení má, stejně jako základní hrací doba, také dva poločasy, ale oba trvají pouze 15 minut. Celé prodloužení tak zabere minimálně 30 minut plus případné nastavení, které se zde vyskytuje také. V případě, že se ani v prodloužení zápas nerozhodne (na konci je skóre obou týmů stejné), nastává penaltový rozstřel, který je zmíněný výše.

Některé evropské soutěže, např. Evropská liga či Liga mistrů, mají ve vyřazovacích částech poněkud složitější systém. Každé kolo se hraje na dva zápasy, kde tým nastoupí jednou na svém hřišti a jednou na hřišti soupeře. Skóre z obou zápasů se sčítá a do dalšího kola postupuje tým s vyšším skóre. V případě, že je skóre shodné, platí pravidlo, že góly vstřelené na soupeřově hřišti mají větší cenu. Uvedu raději příklad. Řekněme, že tým A hrál s týmem B na domácím hřišti 1:1 a na soupeřově hřišti 2:2. Tým A tedy postupuje rovnou po základní hrací době druhého zápasu, jelikož i přes rovnost skóre dal dva góly na soupeřově hřišti a soupeř dal na hřišti týmu A pouze jeden. Za předpokladu, že bude skóre shodné a oba týmy vsítli stejný počet gólů na hřišti soupeře, dochází v druhém hraném zápase k prodloužení, případně k penaltovým kopům.

### 2.2.2 Hokej

Hokejová pravidla jsou o trochu složitější než ta fotbalová. Obecně se hokejový zápas dělí na třetiny, přičemž každá třetina trvá přesně 20 minut. Na rozdíl od fotbalu se zde používá čas čistý, tedy při každém přerušení dochází k jeho pozastavení. Jelikož čas zastavují lidé, je zde prostor pro chyby. Když se jim nepodaří zastavit časomíru včas, musejí ji např. o vteřinu vracet zpět. Je tedy nutné, aby toto umožňoval i framework a aplikace.

Čas ve většině hokejových utkáních ubíhá sestupně, neboli perioda začíná např. na 20 minutách a poté odbíhá až do nuly. Nebývalo tomu tak vždy. Dříve

---

<sup>8</sup>Pokutový kop, neboli penalta, probíhá tak, že vybraný hráč z týmu dostane pokus vstřelit gól brankáři týmu soupeře, přičemž si musí postavit míč na vymezenou značku vzdálenou 11 metrů od brány, ve které gólman stojí.



Obrázek 2.2: Hokejové utkání a digitální kostka zavěšená nad ledem zobrazující skóre zápasu

tento systém měly jen zámořské soutěže a v evropských a mezinárodních čas ubíhal obráceně, ale postupem času se těm zámořským přiblížily. Je tedy nutné s touto možností i nadále počítat.

Dále je tu otázka, jak zápas pokračuje, zda-li nebyl rozhodnut v základní hrací době. Ve všech velikých hokejových soutěžích je v pravidlech, že zápas nelze skončit remízou, ať právě probíhá jakákoliv fáze sezony. Zápas tedy zpravidla pokračuje prodloužením, které se může u soutěží lišit jak svou délkou, tak maximálním počtem hráčů, kteří mohou být na ledě.

Vybral jsem si k analyzování rozdíly mezi jednotlivými světovými soutěžemi. Tentokrát nemám v plánu rozebírat mládežnické kategorie v České republice, jelikož jsem nenašel žádné vhodné materiály.

Následuje výčet mnou vybraných soutěží.

- Tipsport ELH – Nejvyšší česká liga, nazývaná také Extraliga.
- KHL (Kontinentální hokejová liga) – V současnosti druhá nejprestižnější po NHL. Původně to byla soutěž pouze ruská, avšak později se rozšířila do dalších evropských států.
- NHL (National Hockey League) – Nejvyšší a nejprestižnější hokejová soutěž na americkém kontinentu a pravděpodobně i na světě.
- Mistrovství světa – Mezinárodní turnaj konaný každý rok, který má jednu nevýhodu – v době jeho konání se ještě stále dohrává NHL, která se nepřerušuje. Na tomto turnaji mohou být tedy jen hráči z jiných soutěží nebo hráči, jejichž týmy již vypadly.
- Světový pohár – mezinárodní turnaj, který pořádá zámořská NHL. V roce 2016 se konal poprvé od roku 2004.
- Olympijský turnaj – Jelikož se Olympijské hry konají pouze jednou za 4 roky a hráči jsou v tomto případě pouštěni z NHL, je tento turnaj velmi prestižní a sledovaný po celém světě.



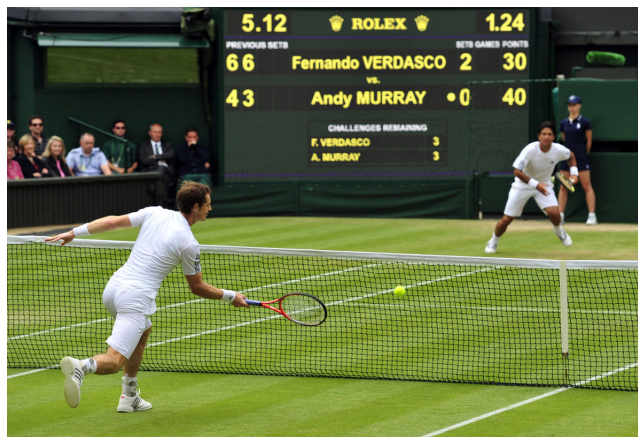
Obrázek 2.3: Loga porovnávaných hokejových turnajů

- Euro Hockey Tour – Série mezinárodních hokejových turnajů složená z turnajů KAJOTbet Hockey Games (Česko), Oddset Hockey Games (Švédsko), Karjala Cup (Finsko) a Channel One Cup (Rusko). Tyto turnaje se odehrávají v průběhu sezóny a nejsou tak prestižní jako již zmíněné turnaje výše. Většinou jsou pojaté spíše jako příprava na ně.

Stejně jako např. u fotbalu, prodloužení nastává, jestliže po vypršení základní hrací doby je shodné skóre obou celků. Prodloužení v základní (skupinové) fázi je u všech posuzovaných soutěží stejné a hraje se maximálně 5 minut do rozhodnutí. Tedy jakmile padne gól, tak celek, který jej vstřelil, vyhrává a utkání je ukončeno. Toto prodloužení se hraje se čtyřmi hráči na každé straně (tři v poli a gólman). V případě, že utkání není rozhodnuto ani v prodloužení, jsou na řadě samostatné nájezdy, které střídavě odjezdí tři různí hráči z každého týmu a jestliže i poté je počet úspěšně zakončených nájezdů stejný, jede vždy nájezd po jednom každý ze soupeřů až do rozhodnutí.

Nyní porovnám rozdílná pravidla soutěží ve fázi playoff, tedy ve vylučovací fázi na konci sezóny. Vyjmenované soutěže zde lze rozdělit do tří skupin. První skupinou je NHL, Světový pohár a nově i KHL [3]. Světový pohár pořádá sama asociace NHL a KHL se od americké soutěže minulý rok rozhodla přiučít. Zde se ve všech vyřazovacích zápasech hraje nekonečné prodloužení až do rozhodnutí, tedy že po základní hrací době následuje přestávka a poté se hraje další 20-ti minutová perioda. Jestliže ani v té se nerozhodne tak je opět přestávka a opět další perioda a tak dále, dokud jeden z týmů nerozhodne. Samostatné nájezdy se tedy nejezdí.

Druhou skupinou je olympijský turnaj, mistrovství světa a Euro Hockey Tour. Tyto soutěže jsou všechny mezinárodní a pod hlavičkou asociace IIHF (International Ice Hockey Federation), která má svá jednotná pravidla. Tato pravidla určují, že ve všech zápasech až na finálový trvá prodloužení 10 minut a hraje se se čtyřmi hráči v poli na každé straně. Poté následují případně



Obrázek 2.4: Tenisové utkání s ukazatelem aktuálního skóre na pozadí

nájezdy, které zahajuje střídavě 5 hráčů z každého týmu. Jak jsem již zmínil, finálový zápas liší a to pouze tím, že prodloužení netrvá 10, nýbrž 20 minut.

Do poslední skupiny patří pouze česká extraliga, neboli Tipsport ELH. Zde se prodloužení v playoff hraje dlouhé 20 minut a poté následují případné nájezdy, které zahajuje pět hráčů z každé strany. Je tu ale jeden rozdíl. O postup ve vyřazovací části se totiž hraje na 4 vítězné zápasy v každém kole. Ovšem jakmile za stavu série 3:3 dochází k sedmému zápasu, prodloužení se hraje nekonečné až do rozhodnutí. Tedy po základní hrací době následují další 20-ti minutové periody, dokud zápas není rozhodnut. U finálové série se hraje nekonečné prodloužení už od 5. zápasu.

### 2.2.3 Tenis

Posledním analyzovaným sportem je tenis. Tenisový zápas na rozdíl od dvou dříve jmenovaných sportů není závislý na čase vůbec. Čas se zde uvádí, ovšem pouze jako orientační hodnota.

Pokud jde o bodování, je tenis naopak složitější. Jestliže hráč získá bod (míč), je stav 15:0. Pokud získá druhý, je stav 30:0, poté 40:0 a dalším bodem získává game (hru). Jestliže hráč získá bod za stavu 40:40, má výhodu, kterou musí ještě potvrdit, aby získal game. Pokud jí nepotvrdí a bod získá protihráč, hra se vrací opět do stavu 40:40. Těchto gamů potřebuje alespoň 6 a minimálně o dva více než soupeř, aby získal set. Jestliže nastane stav 6:6, následuje tzv. zkrácená hra (tiebreak). Jde o samostatnou hru, která má za úkol rychleji rozhodnout o vítězi aktuálního setu. Oba hráči začínají na skóre 0:0 a musejí dosáhnout alespoň na 7 bodů a o dva více než soupeř. Hráč, kterému se to povede, vyhrává aktuální set poměrem 7:6. Obvykle se utkání hrají na dva nebo tři vítězné sety.

## 2. ANALÝZA

---



Obrázek 2.5: Loga porovnávaných tenisových turnajů

Předchozí popsané sporty se dělily na soutěže, které obvykle trvají celý rok. Tenis se dělí na kratší turnaje, konané po celém světě. Vymenuji a popíši zde ty nejprestižnější z nich.

- French Open – Tenisový turnaj hraný na přelomu května a června ve Francii. Nejsledovanější událost ve Francii. Je jediným turnajem ze čtyř nejvýznamnějších turnajů velké čtyřky (French Open, Australian Open, US Open, Wimbledon) hraný na otevřených antukových dvorcích.
- Australian Open – Turnaj, který se každoročně pořádá v druhé polovině ledna v Austrálii v Melbourne. Hraje se na středně tvrdém akrylátovém povrchu.
- US Open – Turnaj hraný na přelomu srpna a září v Queens v New Yorku. Hraje se na tvrdém povrchu nazývaném DecoTurf. [4]
- Wimbledon – Nejslavnější a nejstarší tenisový turnaj na světě. Hraje se na přelomu června a července v jihozápadní londýnské části Wimbledon. Jediný z velké čtyřky turnajů, který se hraje na trávě.
- Davis Cup – Největší mužský tenisový turnaj reprezentačních mužstev.
- Fed Cup – Největší ženský tenisový turnaj reprezentačních mužstev.

Na těchto turnajích platí velmi podobná pravidla bodování. Ženy hrají na dva vítězné sety a muži na tři. U všech turnajů je v pravidlech ukotvena výše popsaná zkrácená hra, ale jen do rozhodujících sad, poté se hraje jen nekonečný počet gamů. Výjimkou je US Open, kde se zkrácená hra uplatňuje i v rozhodujících sadách.

Ještě bych rád zmínil juniorský tenis. Například v České republice založil Český tenisový svaz dva projekty s názvy Minitenis a Babytenis pro hráče ve věku 6-7 let, resp. 8-9 let. [5]

Zápas se v Minitenisu hraje měkkými míči a vyhrává hráč, který dosáhne 10 bodů. Babytenis se hraje s odlehčenými míči na jeden set do 6-ti gamů, přičemž za stavu 6:6 nastává zkrácená hra.

### 2.2.4 Shrnutí analýzy sportů

Tyto tři právě analyzované sporty ukázaly, že je nutné, aby framework umožňoval nastavení délky času jedné periody a jestli bude čas ubíhat sestupně, vzestupně či jestli bude měření času vůbec vyžadováno. Počet period musí být flexibilní. Každá z částí zápasu může být jinak dlouhá a mít jiná pravidla pro její ukončení. Některé periody mohou být ukončeny uživatelem stiskem tlačítka a jiné po prostém vypršení jejich času. Dále musí systém podle skóre určit, zda zápas ukončit nebo pokračovat dalším kolem (např. penaltové kopy). Poslední důležitou vlastností frameworku je skóre. Tenis velmi dobře reflektuje vyšší nároky na složitější počítání skóre a i s tím je nutno si poradit.

## 2.3 Verze operačního systému

V zadání práce je jasně uvedeno, že aplikace je cílena na operační systém Android od firmy Google. Tento operační systém je ve vývoji již od roku 2003, tehdy ještě pod samostatnou společností Android Inc., kterou teprve až o dva roky později Google koupil. Za svou existenci si systém prošel mnoha změnami a vyšel v mnoha úpravách, od verze 1.5 pojmenovávaných podle sladkostí. Ovšem důležitějším ukazatelem je číslo verze a číslo API levelu. [6]

„Úroveň API je celočíselná hodnota, která jednoznačně identifikuje API revizi frameworku, který je nabízený verzí systému Android.“ [7] Neboli každá verze systému Android má své unikátní číslo API. Toto číslo je pro programátora velmi zajímavé, jelikož se musí před samotným vývojem rozhodnout, na kterou úroveň API bude jeho aplikace cílená. Mnohdy to není tak lehké rozhodnutí, jak by se mohlo zdát. Konfigurace Android projektu umožňuje zvolit tři parametry týkající se verze API:

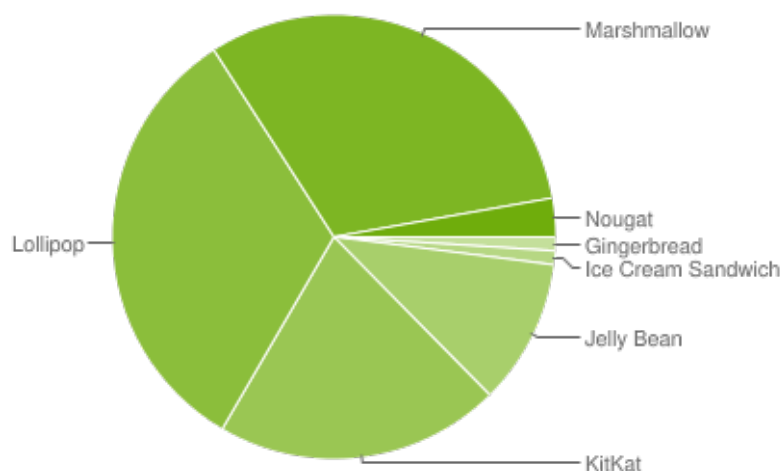
1. `minSdkVersion` – Udává minimální úroveň API, na které je možné aplikaci nainstalovat. Na nižších verzích systému aplikace nainstalovat nepůjde a ani se nezobrazí v nabídce na Google Play
2. `maxSdkVersion` – Tímto parametrem, který se používá jen ve výjimečných případech, lze omezit cílovou verzi systému i shora. Avšak je to neobvyklé, jelikož se Google snaží zajistit, aby aplikace s nižší verzí API fungovaly i na všech systémech s vyšší verzí API
3. `targetSdkVersion` – Tento parametr udává, na kterou verzi systému je aplikace odladěná a cílená. Systém s vyšší verzí API dokáže emulovat běh systému s nižší verzí API tak, že to nepozná ani samotná aplikace [8]

## 2. ANALÝZA

---

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	1.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.0%
4.1.x	Jelly Bean	16	3.7%
4.2.x		17	5.4%
4.3		18	1.5%
4.4	KitKat	19	20.8%
5.0	Lollipop	21	9.4%
5.1		22	23.1%
6.0	Marshmallow	23	31.3%
7.0	Nougat	24	2.4%
7.1		25	0.4%

Tabulka 2.3: Tabulka rozdělení podílu jednotlivých verzí systému Android ze dne 6.3.2017



Obrázek 2.6: Diagram rozdělení podílu jednotlivých verzí systému Android

Všechny tyto parametry jsou v zásadě nepovinné, ale vyplatí se je nastavit, jelikož mají své výchozí hodnoty, které nám nemusí vyhovovat.

Zásadní je pro mě zvolit si správné `minSdkVersion`, tedy minimální verzi API, jelikož když zvolím moc nízkou, aplikace sice pojede na mnohem větším počtu zařízení, avšak samotný vývoj bude náročnější, jelikož nebudu moci použít některé novější funkce z vyšších API. Na druhou stranu když zvolím úroveň moc vysokou, mohu přijít o mnoho potenciálních podporovaných zařízení. V tabulce 2.3 můžeme vidět rozdělení jednotlivých verzí na zařízeních s Androidem.

Podle této tabulky, mého názoru a domluvě s vedoucím jsem se rozhodl zvolit jako minimální API verzi 19, tedy verzi systému 4.4 s označením KitKat.



Tato verze je již sice starší, ale zároveň má stále velmi vysoký podíl a mnoho novějších funkcí.

## 2.4 Programovací jazyk

Jelikož je cílovou platformou operační systém Android a jelikož jsem se rozhodl pro nativní aplikaci bez jakýchkoliv komplikací, vyplývá z toho, že programovacím jazykem bude Java a to konkrétně ve verzi 7, kterou Android plně podporuje. [9]

Výhodou rozdělení projektu na framework a aplikaci je to, že bude framework zcela oddělen od závislosti na cílové platformě. Framework plánuji napsat čistě v Javě (včetně testů) a poté ho jen využít v mobilní aplikaci.

## 2.5 Funkční a nefunkční požadavky

Již jsem zmínil, že se analýza velmi často opomíjí. Sběr funkčních a nefunkčních požadavků je jednou ze základních a nejdůležitějších částí analýzy. Slouží ke sjednocení požadavků zákazníka (zadavatele) a představ vývojářů. Lze se vyhnout velmi častým případům, kdy po nezanedbatelném čase stráveném prací na projektu vnikne něco, co se rozchází s požadavky zákazníka. Zároveň chrání vývojáře v případech, kdy zákazník tvrdí, že po nich chtěl něco jiného, avšak vývojáři mají v ruce zadavatelem dříve potvrzené zadání.

### 2.5.1 Funkční požadavky

#### 2.5.1.1 Aplikace

**F1.1 Vytvoření nového zápasu** Uživatel klikne na tlačítko pro vytvoření nového zápasu a bude požádán o zadání informací o obou týmech, o zápase a dalších nastaveních.

**F1.2 Nastavení vlastností zápasu** Uživatel se již po vytvoření zápasu, ale ještě před jeho samotným započítím rozhodne, že chce změnit nastavení či vlastnosti zápasu.

**F1.3 Zaznamenávání stavu zápasu** Uživatel pomocí ovládacích prvků zaznamenává a mění průběžný stav hry v reálném čase.

**F1.4 Streamování obsahu** Vše, co uživatel zadává, se následně posílá na server, kde je pomocí např. webové stránky zobrazováno jiným uživatelům (konzumentům).

**F1.5 Identifikace** Systém bude umožňovat uživateli se přihlásit pod svým účtem (např. pomocí Google účtu).

**F1.6 RSS kanál** Aplikace bude poskytovat RSS kanál pro integraci s jinými systémy.

### 2.5.1.2 Framework

**F2.1 Obecnost** Architektura frameworku musí být navržena tak, aby byla schopna pokrýt všechna pravidla všech sportů typu *versus*.

## 2.5.2 Nefunkční požadavky

### 2.5.2.1 Aplikace

**N1.1 Platforma** Aplikace poběží na mobilních zařízeních se systémem Android minimálně ve verzi 4.4 (KitKat) – level API 19.

**N1.2 Paměť** Aplikace by neměla být náročná na paměť, aby ji bylo možno používat i na méně výkonných zařízeních.

**N1.3 Dostupnost přes síť** Jestliže bude uživatel chtít vysílat zápas přes internet, je potřeba, aby měla aplikace přístup na síť, aby mohla odesílat informace na server.

**N1.4 Rychlost** Při posílání stavu zápasu přes internet by nemělo docházet k velikým prodlevám.

**N1.5 Bezpečnost proti přístupu cizího uživatele** Koupený virtuální obsah uživatele aplikace musí být přístupný až po jednoznačné identifikaci daného uživatele, aby se k cizímu obsahu nemohl dostat někdo jiný (lze řešit například spárováním s Google účtem).

**N1.6 Displej** Aplikace bude podporována pouze na dotykových telefonech s větším displejem a tabletech, od velikosti displeje s uhlopříčkou alespoň 5.1 palců. Zároveň obrazovka zápasu poběží pouze v režimu *landscape*, tedy na šířku. To vše proto, aby byla pro uživatele zajištěna co nejpříznivější ovladatelnost.

**N1.7 Jazyk** Aplikace poběží ve dvou jazycích – v češtině a angličtině.

### 2.5.2.2 Server

**N2.1 Bezpečnost proti podvržení dotazu (Cross-site scripting)** Server musí být odolný vůči podvržení dat. Když aplikace posílá data o aktuálním stavu zápasu, je nutné, aby nebylo možné data podvrhnout — tedy, aby útočník nemohl poslat falešný požadavek, který by server jako falešný nerozpoznal. Je důležité tedy zavést jednoznačnou identifikaci požadavků z aplikace.

**N2.2 SQL injection** Server musí být odolný vůči jedné z nejznámějších a nezákladnějších hackerských technik zvané SQL injection. Jedná se o napadení databázové vrstvy podsunutím svého nebezpečného kódu díky neošetřeným vstupům.

### 2.5.2.3 Framework

**N3.1 Rozšiřitelnost** Framework musí být napsán tak, aby ho bylo možné snadno v budoucnu rozšiřovat a upravovat.

### 2.5.2.4 Ostatní

**N4.1 Dokumentace** K aplikaci musí být vytvořena dokumentace.

## 2.6 Přihlašování uživatelů

V předchozí kapitole jsem zmiňoval funkční požadavek na identifikaci uživatelů pomocí přihlašování ke svému účtu. Vytváření a následná správa uživatelů není součástí mé práce a ani není pokryta mými schopnostmi a zkušenostmi. Existuje možnost (poslední dobou velmi využívaná) jak vyřešit identifikaci nabídnutím uživateli propojení aplikace s jeho účtem u jiného poskytovatele. Správa účtu zůstane na třetí straně a uživatel si nebude muset pamatovat další přihlašovací údaje. Třetí strana toto samozřejmě musí ve svém systému umožňovat. Zbývá jen vybrat tu správnou službu/y.

Existuje platforma jménem **Google Firebase**. Tato služba nabízí velmi mnoho funkcí. Poskytuje například databáze a jejich správu, Cloudové řešení, testování, hosting, statistiky a analýzu aplikace, report pádů aplikace, indexování atd. Některé služby jsou placené a některé zcela zdarma, jako právě zmíněná autentizace uživatelů pomocí služeb třetích stran. Nabízí přímou podporu účtů od Facebooku, Googlu, Twitteru a GitHubu. Já využiji podporu Googlu s tím, že kdykoliv v budoucnu půjde aplikace rozšířit i o další z nich.

## 2.7 Případy užití a diagramy aktivit

Po rešerši konkurence a jednotlivých sportů, výběru platformy a požadavcích na framework a aplikaci, neboli velmi teoretických částech, se dostávám k části o trochu více praktické, tedy vytváření diagramů. Těch existuje mnoho druhů od jednodušších po složitější. Ne všechny, které se učí ve škole, se nakonec používají v praxi. To může platit i o některých z těch, které jsem vybral, avšak vybral jsem ty nejznámější.



Obrázek 2.7: Google Firebase

### 2.7.0.1 Případy užití

Případy užití (use Cases) by měly být prvním a nejjednodušším diagramem, který vývojář vytvoří. Je to diagram, který znázorňuje základní funkce aplikace/frameworku z pohledu různých kategorií osob (uživatel, administrátor, programátor atd.) v různých fázích – např. kde se zrovna osoba v aplikaci nachází. Lze z něj tedy vyčíst různé scénáře práce s aplikací/frameworkem. Slouží pro jednoduchou představu vývojářů, avšak díky své jednoduchosti se v některých případech ukazuje i zákazníkovi opět kvůli sjednocení zadání. Měl by tedy být vytvořen tak, aby ho byl schopen pochopit člověk bez jakýchkoliv vývojářských znalostí či předchozích zkušeností.

Většina funkcí frameworku i aplikace jsou velmi jednoduché a prosté, proto stačí jejich textový popis.

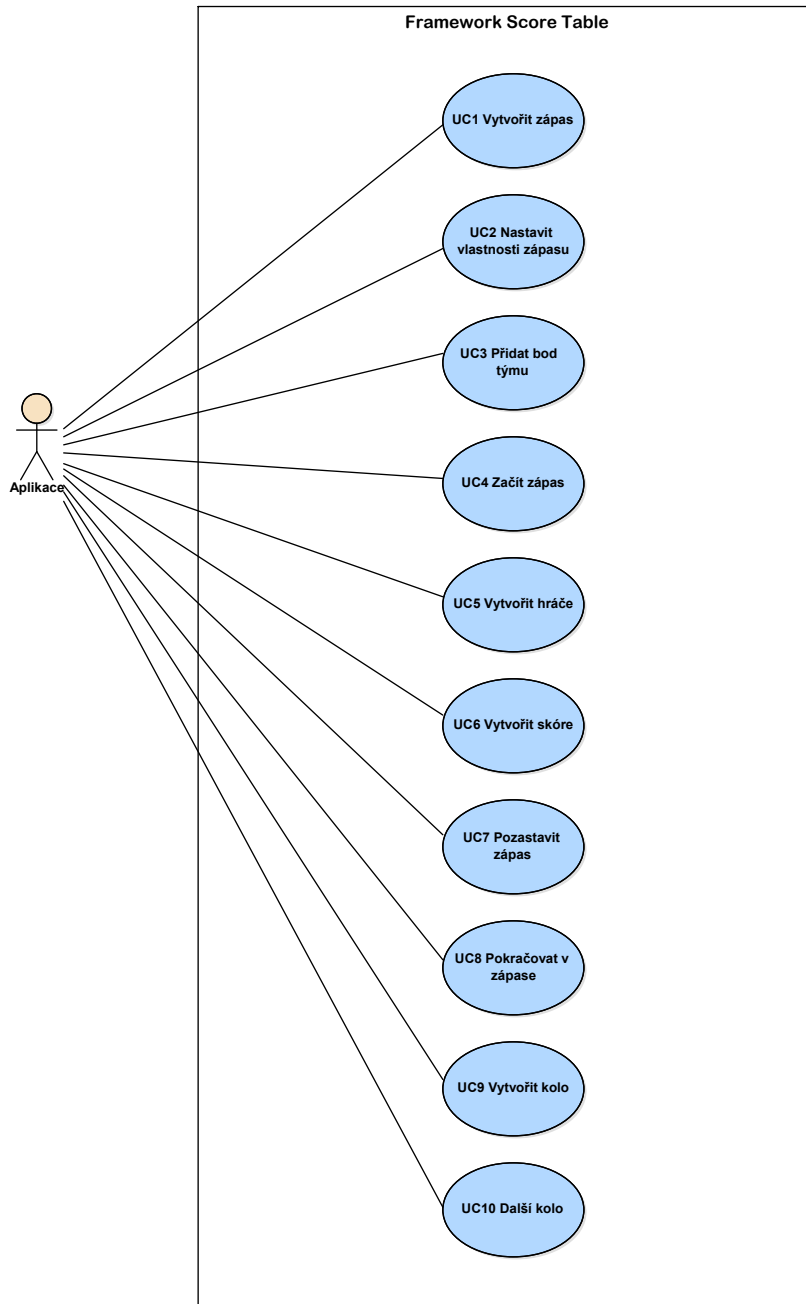
### 2.7.1 Framework

V mém případě potřebuji vytvořit diagram případů užití jak pro framework, tak pro všechny tři sportovní aplikace, které hodlám vytvářet. Diagram pro framework je na obrázku 2.8.

V případě frameworku je účastníkem pouze aplikace, která je jediná, kdo k němu má přístup.

**UC1 Vytvořit zápas** Umožňuje vytvořit nový objekt se zápasem, který dále poskytuje ostatní funkce k ovládní zápasu.

**UC2 Nastavit vlastnosti zápasu** Umožňuje nastavit některé specifikace zápasu.



Obrázek 2.8: Diagram případů užití frameworku

## 2. ANALÝZA

---

**UC3 Přidat bod týmu** Umožňuje libovolnému týmu přidat bod do skóre. Tento bod se přidá podle toho, jakým způsobem uživatel vytvořil systém skóre daného sportu.

**UC4 Začít zápas** Umožňuje odstartovat vytvořený zápas (může spustit časomíru, je-li k dispozici).

**UC5 Vytvořit hráče** Umožňuje vytvořit oba účastníky zápasu.

**UC6 Vytvořit skóre** Umožňuje vytvořit skóre tak, aby se řídilo pravidly požadovaného sportu.

**UC7 Pozastavit zápas** Umožňuje pozastavit běžící zápas (může pozastavit časomíru, je-li k dispozici).

**UC8 Pokračovat v zápase** Umožňuje znovu pokračovat v pozastaveném zápase (může opět spustit časomíru, jeli k dispozici).

**UC9 Vytvořit kolo** Umožňuje vytvořit fázi (část), ze kterých se zápas skládá a mezi kterými bude postupem přecházet.

**UC10 Další kolo** Umožňuje přejít ve hře do dalšího kola. V případě, že žádné kolo již není k dispozici, hru ukončí.

### 2.7.2 Aplikace

Na obrázku 2.9 je diagram případů užití aplikace pro fotbal. Diagramy pro ostatní sporty lze nalézt v příloze.

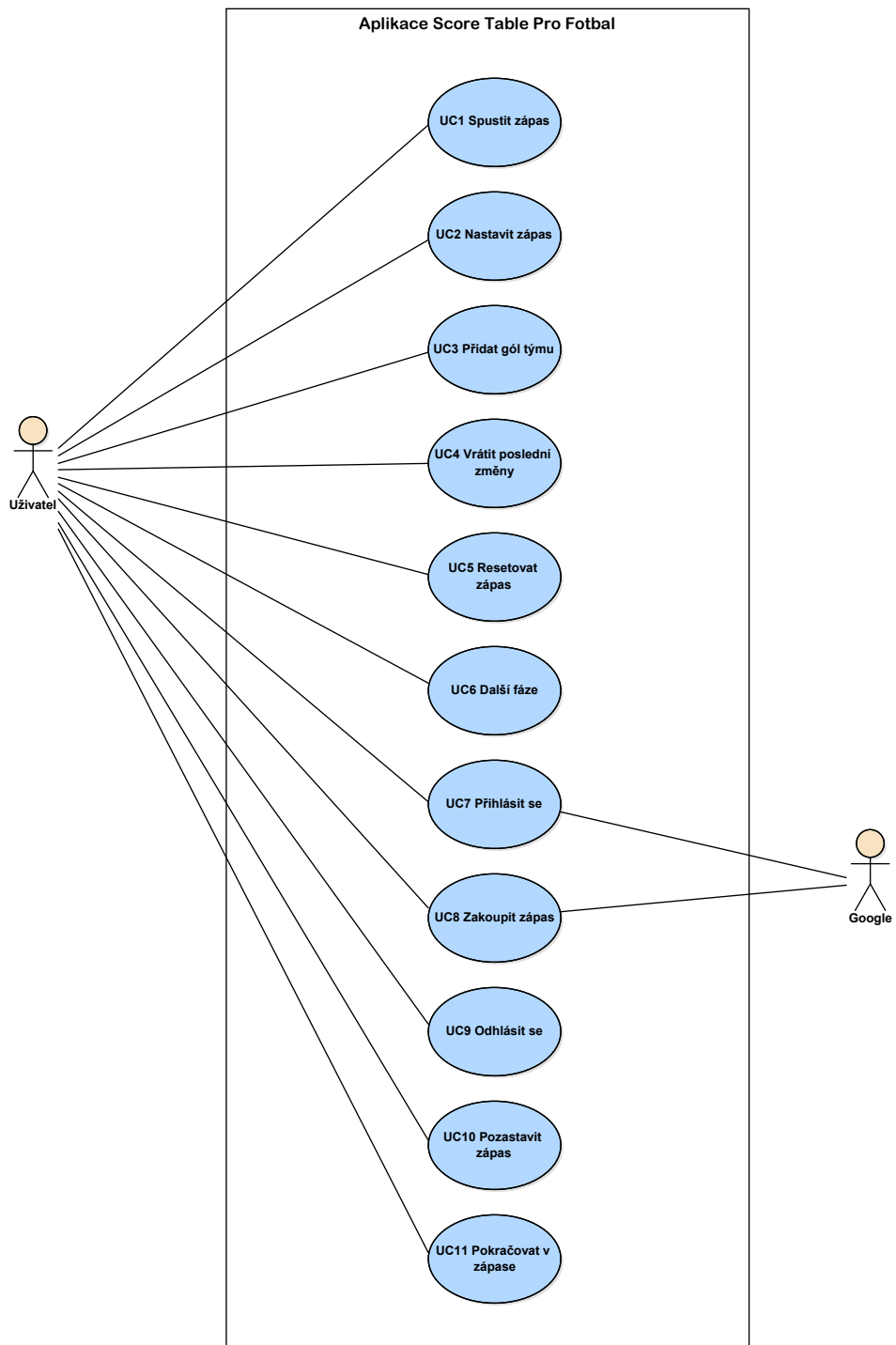
**UC1 Spustit zápas** Odstartuje celý zápas a spustí časomíru.

**UC2 Nastavit zápas** Zobrazí uživateli obrazovku s nastavením vlastností zápasu.

**UC3 Přidat bod týmu** Vybranému týmu přidá bod.

**UC4 Vrátit poslední změny** Zápas a tedy i obrazovka budou navraceny do stavu před poslední změnou vyvolanou uživatelem či systémem.

**UC5 Resetovat zápas** Všechn dosavadní průběh zápasu bude zahozen a zápas uveden do původní podoby před svým spuštěním.



Obrázek 2.9: Diagram případů užití aplikace pro fotbal

## 2. ANALÝZA

---

**UC6 Další fáze** Jestliže uživatel klikne na toto tlačítko, zápas bude posunut do své další fáze (např. z 1. poločasu do 2.poločasu). Pokud žádná fáze již nezbývá, zápas bude ukončen a vyhodnocen.

**UC7 Přihlásit se** Uživatel se může přihlásit nebo si nejprve založit svůj účet od Googlu.

**UC8 Zakoupit zápas** Uživatel si může zakoupit příslušný počet zápasů pro svůj účet.

**UC9 Odhlásit se** Systém uživatele odhlásí a znemožní mu přístup k placeným funkcím.

**UC10 Pozastavit zápas** Pozastaví zápas a časomíru.

**UC11 Pokračovat v zápase** Opět zápas a časomíru spustí.



---

# Návrh

Druhou důležitou fází vývoje softwaru je jeho návrh. Zde vycházíme z předchozí specifikace požadavků a analýzy a na nich stavíme architekturu systému a tvoříme uživatelské rozhraní.

## 3.1 Doménový model

Diagram pod názvem doménový model lze označit za počáteční fázi návrhu softwaru. Jelikož hodlám programovat objektově, všechny objekty (části) je možné jednoduše znázornit graficky.

Tyto objekty se zobrazují jako třídy, které jsou oproti skutečným třídám velmi zjednodušené a neobsahují některé metody a nedůležité atributy. Třídy mají mezi sebou vztahy, kterých může být několik druhů s různou multiplicitou.

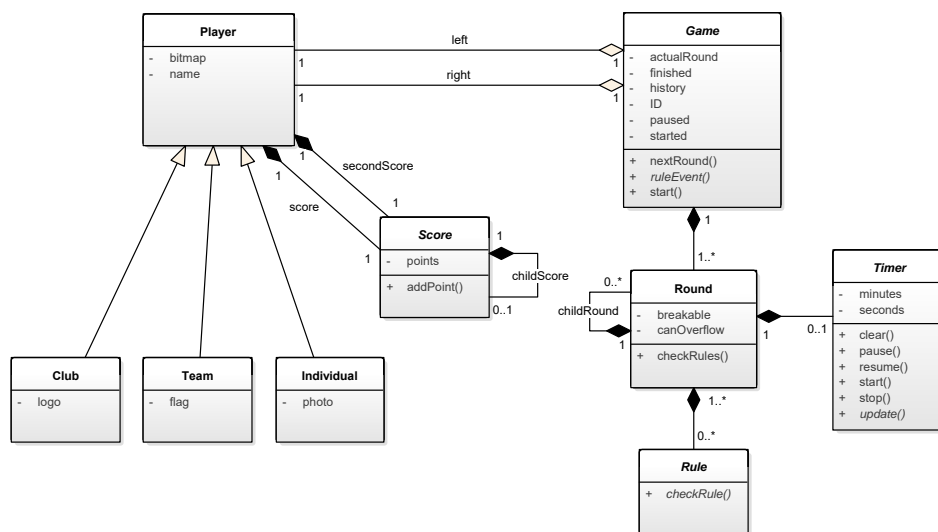
Graficky se třídy vykreslují jako obdélníky, obsahující atributy, a vztahy jako přímký identifikované čísly – jejich multiplicitou.

Hlavní myšlenkou doménového modelu je zobrazit systém jako celek složený z částí a vztahy panující mezi těmito částmi tak, aby to pochopil jakýkoliv člověk, který nemusí být nutně vývojářem. Důležité je, že tento diagram je platformově nezávislý, neboli nezáleží, na jakém operačním systému software poběží a v jakém jazyce bude naprogramován.

Framework, znázornění doménovým diagramem, je na obrázku 3.1. Následně popíšu jeho hlavní entity.

**Score** Tato entita zaznamenává skóre každého hráče. Jelikož se skóre jednotlivých sportů může výrazně lišit, má několik důležitých vlastností. Funkce `addPoint()` je abstraktní, tudíž ji programátor při použití frameworku musí dopsat podle toho, jakým způsobem se v jeho sportu přidává hráči bod. Díky analýze tenisu jsem si uvědomil, že skóre může mít několik vrstev. Proto každý objekt `Score` může mít ještě `childScore` stejného typu. Lze tedy rekurzivně vytvořit hierarchii nekonečného počtu vrstev skóre.

### 3. NÁVRH



Obrázek 3.1: Doménový model frameworku

**Player** Objekt samotného hráče. Hráčem může být jak jednotlivec, tak tým či klub. Každý hráč má právě dva objekty typu `Score`. Jedno hlavní a druhé vedlejší, použitelné např. pro penalty ve fotbale. Dále hráč disponuje svým jménem a případně i fotografií, vlajkou nebo logem.

**Round** Představuje jedno kolo sportu (např. třetinu v hokeji). Disponuje několika proměnnými, které určují, jak se bude chovat a pravidly, které kontroluje.

**Rule** Entita `Rule` reprezentuje jedno pravidlo, které když je splněno, je zápas ukončen a zavolána metoda hry `ruleEvent()`, kde se programátor postará o vše, co je potřeba vykonat po ukončení hry.

**Timer** `Timer` se stará o časový průběh hry. Tento objekt vlastní abstraktní metodu `update()`, která se zavolá každou vteřinu od spuštění časovače. Tuto metodu programátor využije, pakliže potřebuje například každou vteřinu aktualizovat ukazatel času.

**Game** Hlavní objekt hry. Představuje právě jeden hraný zápas. Skládá se ze dvou hráčů a ze samotných výše zmíněných kol, které postupně vykonává a přepíná mezi nimi. Těchto kol může mít neomezeně. Také volá kontrolu všech pravidel ve správnou chvíli.

## 3.2 Diagram tříd

Diagram tříd (class diagram) v zásadě vychází z doménového modelu, ale je mnohem podrobnější. Tento diagram zobrazuje již objekty a jejich atributy a metody tak, jak budou později naprogramovány. Atributy a metody již mají své typy a jsou vyobrazeny jak veřejné, tak privátní. Tento diagram je již platformově závislý, neboli je vytvářený pro konkrétní programovací jazyk.

Diagram slouží spíše programátorům, pro zákazníka může být složitý a zbytečný, jelikož zákazníka většinou nezajímá vnitřní struktura softwaru, ale jestli je funkční a splňuje jeho požadavky.

Model je určený hlavně pro větší projekty, pro velké informační systémy, ve kterých se musejí programátoři složitěji orientovat. Ovšem pokud ho programátor použije i pro menší projekt, případně jen sám pro sebe, nebude toho litovat. Diagram ho donutí více přemýšlet o celém systému, který vytváří a může odhalit některé chyby mnohem dříve.

Návrh tříd frameworku pro programovací jazyk Java je na obrázku 3.2.

## 3.3 Návrh uživatelského prostředí

Před samotným programováním je dobré si rozvrhnout, kde se budou nacházet jednotlivé prvky na obrazovkách. K tomuto účelu jsem si nejprve zvolil nejjednodušší a nejpoužívanější metodu – tužku a papír. Ovšem později bylo potřeba mé návrhy převést do čitelnější a uhlazenější formy. K tomuto účelu jsem si vybral program Balsamiq Mockups<sup>9</sup> ve verzi 3.

Aplikaci bych rád navrhl velmi jednoduchou a přehlednou. Bude sestávat ze tří obrazovek.

### 3.3.1 Hlavní obrazovka

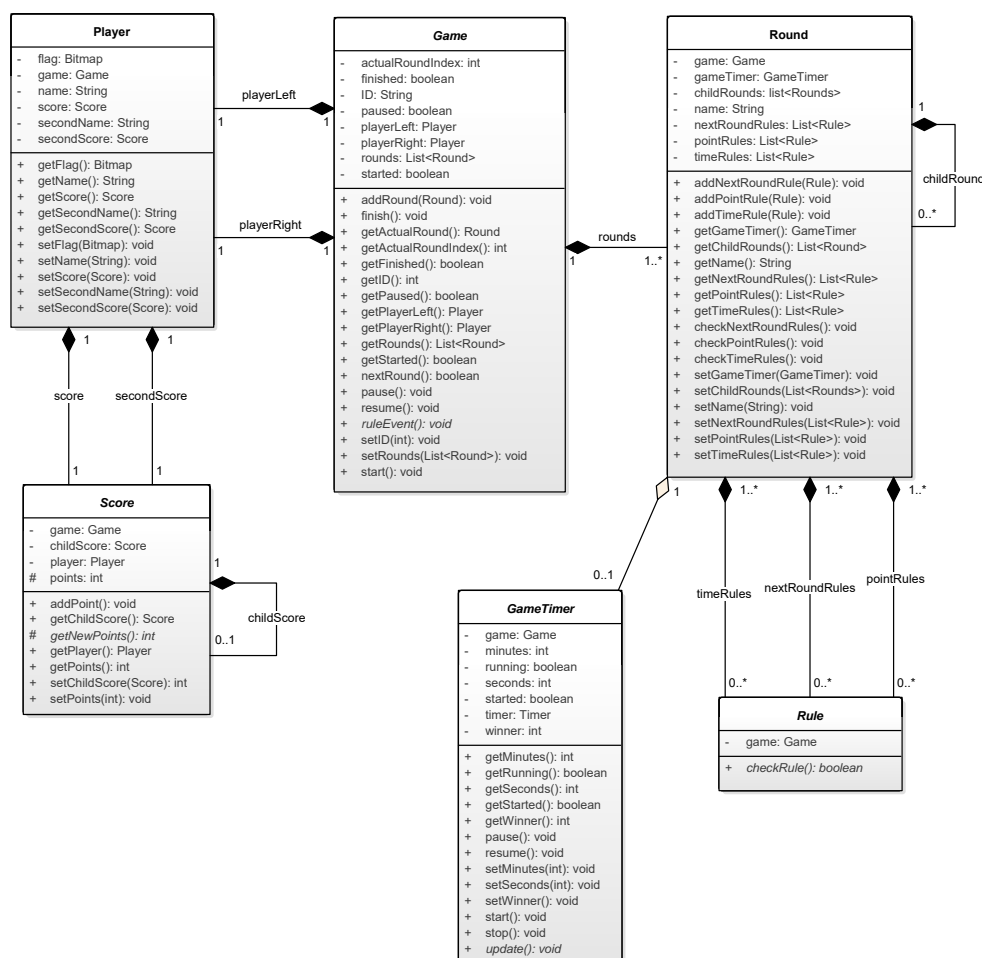
Začnu hlavní obrazovkou, která se ukáže uživateli po spuštění aplikace. Měla by být pro uživatele jednoznačná. Tlačítka musejí být správně pojmenována, aby bylo jasné, k čemu slouží i bez nahlédnutí do dokumentace. Zde se moc grafických prvků nacházet nebude. Důležitá je navigace, neboli hlavní menu aplikace, které by mělo logicky být umístěno uprostřed. Jelikož je aplikace velmi prostá, budou se zde nacházet pouze tlačítka k vytvoření nového zápasu, přihlášení a nastavení. Na obrázcích 3.3, 3.4 a 3.5 lze vidět výsledný návrh.

### 3.3.2 Nastavení

Jestliže uživatel klikne na tlačítko pro vytvoření nového zápasu, bude před samotným zápasem nejprve přesunut na jeho základní nastavení. Zde může nastavit vlastnosti konkrétního zápasu, jako například informace o obou soupeřích, nastavit fázi daného turnaje atd. Jestliže má vše potřebné nastavené,

<sup>9</sup><https://balsamiq.com/products/mockups/>

### 3. NÁVRH



Obrázek 3.2: Diagram tříd frameworku

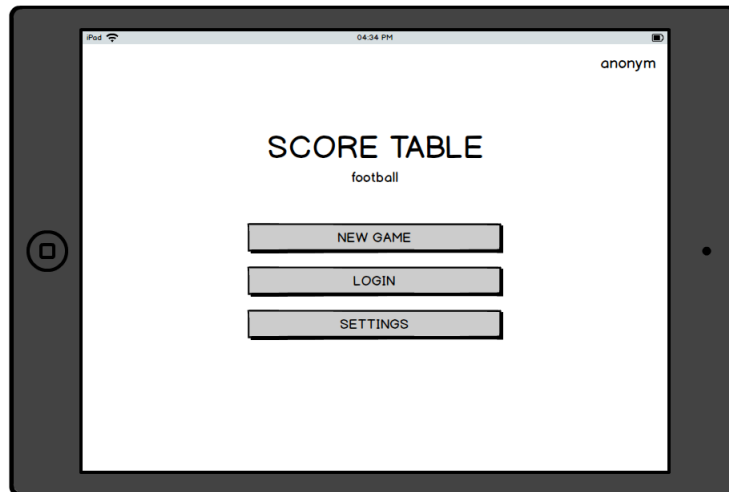
tlačítkem k vytvoření zápasu se přesune na další obrazovku s jeho samotným průběhem. Obrázky 3.6 a 3.7 jsou výsledkem mého návrhu.

#### 3.3.3 Zápas

Uživatel se již dostal k samotnému průběhu zápasu, zřejmě na nejdůležitější obrazovku aplikace. Ta bude rozdělena na dvě poloviny, z nichž bude každá patřit jednomu ze soupeřů. Budou se zda nacházet základní informace o soupeřích (jméno a grafická identifikace jako např. logo) a ovládací prvky průběhu zápasu. Na obrázcích 3.8, 3.9, 3.10 a 3.11 je znázorněn jeden z možných průběhů zápasu.

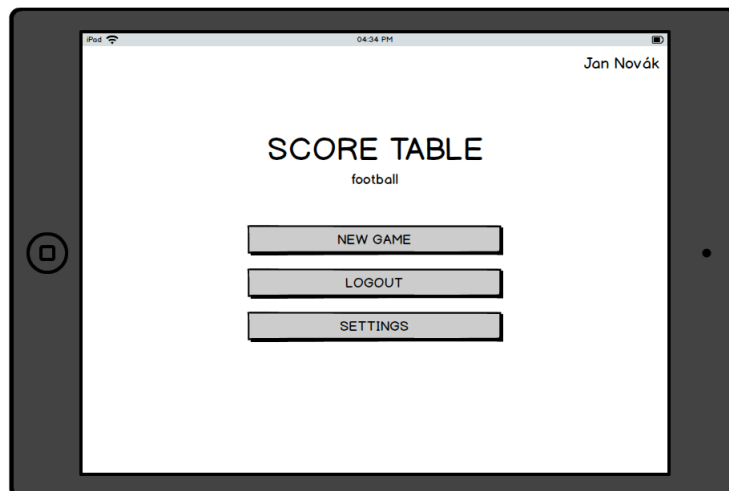
### 3.3. Návrh uživatelského prostředí

main screen



Obrázek 3.3: Návrh hlavní obrazovky aplikace pro fotbal – základní pohled

main screen logout

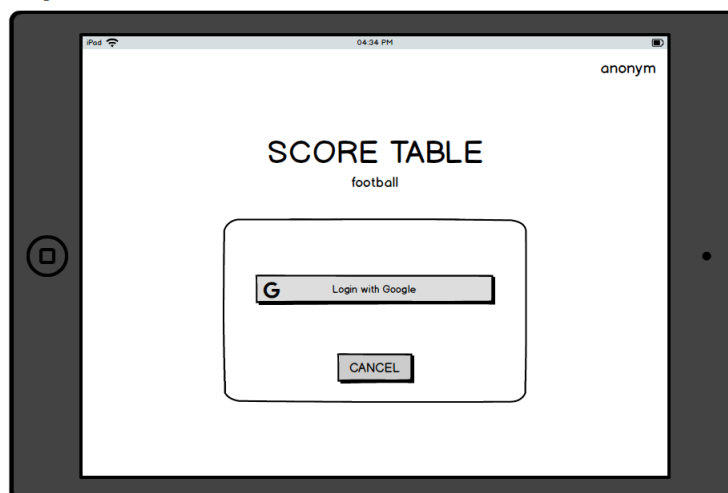


Obrázek 3.4: Návrh hlavní obrazovky aplikace pro fotbal – přihlášený uživatel

### 3. NÁVRH

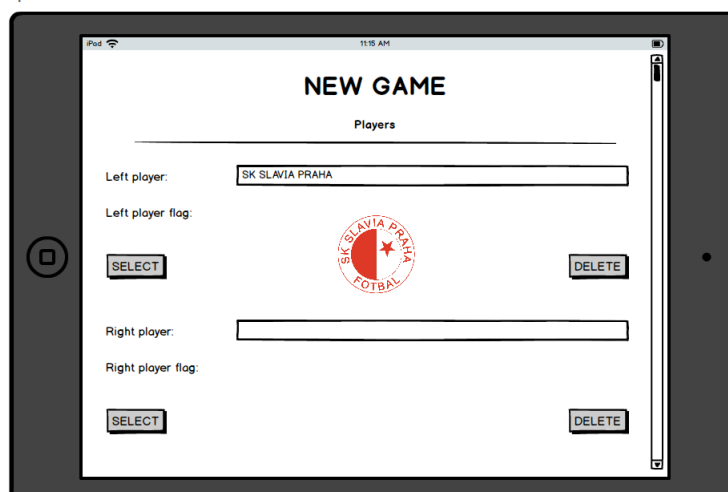
---

main screen login



Obrázek 3.5: Návrh hlavní obrazovky aplikace pro fotbal – přihlášení

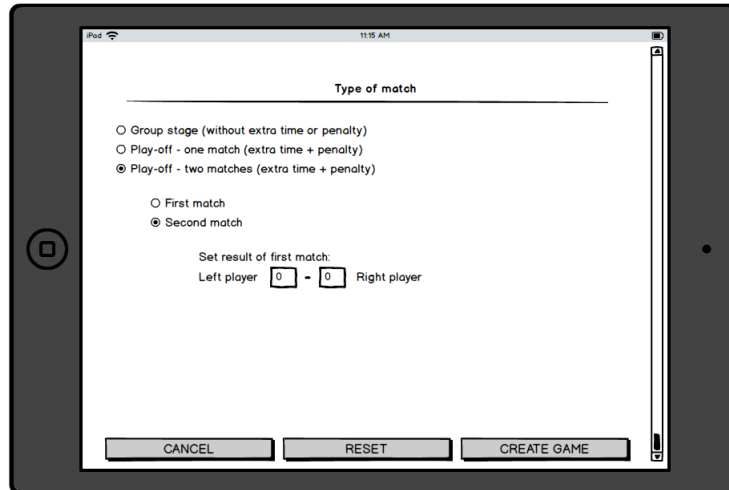
settings top



Obrázek 3.6: Návrh obrazovky nastavení aplikace pro fotbal – horní část

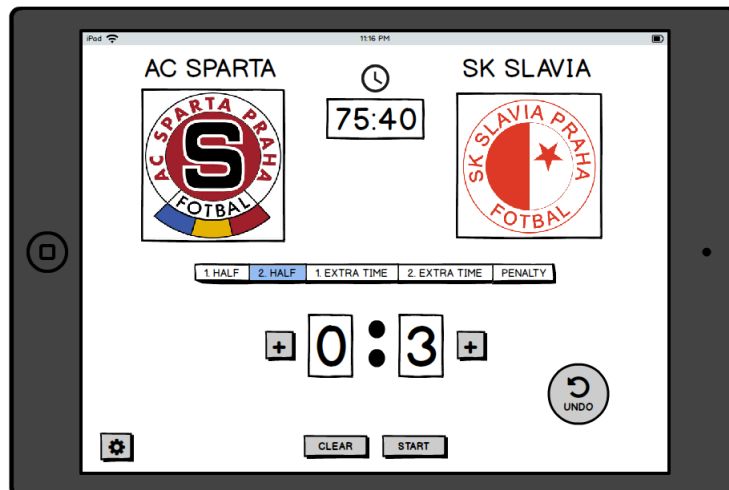
### 3.3. Návrh uživatelského prostředí

settings bottom



Obrázek 3.7: Návrh obrazovky nastavení aplikace pro fotbal – dolní část

full time



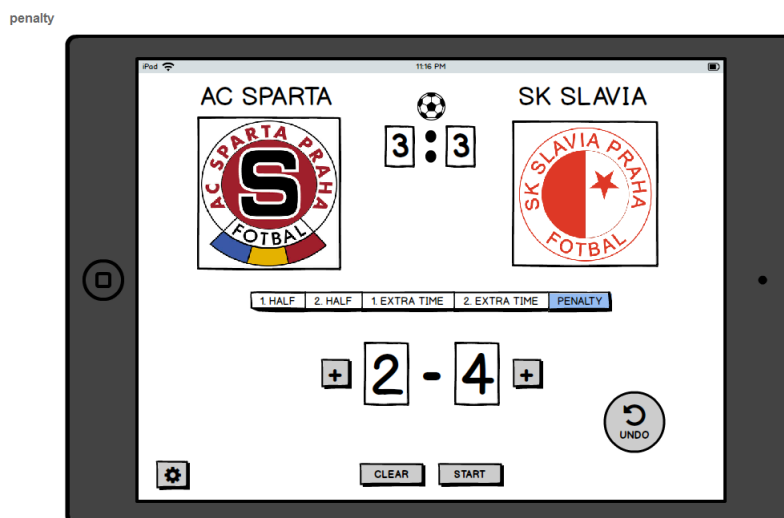
Obrázek 3.8: Návrh obrazovky zápasu aplikace pro fotbal – základní hrací doba

### 3. NÁVRH

---



Obrázek 3.9: Návrh obrazovky zápasu aplikace pro fotbal – prodloužení



Obrázek 3.10: Návrh obrazovky zápasu aplikace pro fotbal – penaltový rozstřel





Obrázek 3.11: Návrh obrazovky zápasu aplikace pro fotbal – konec zápasu

### 3.4 Odesílání změn

Hlavní funkcí aplikace bude informace o průběhu zápasu někomu sdílet. Všechno, co uživatel zaznamená, aplikace pošle na server. Server tyto informace přijme, zpracuje a pozmění stav na příslušné webové stránce, která průběh zápasu zobrazuje.

Takovou komunikaci je nutné správně navrhnout, aby byla co možná nejméně datově objemná, kvůli úspoře přenosu dat uživatele i kvůli minimálnímu zpoždění zobrazení změn na webu.

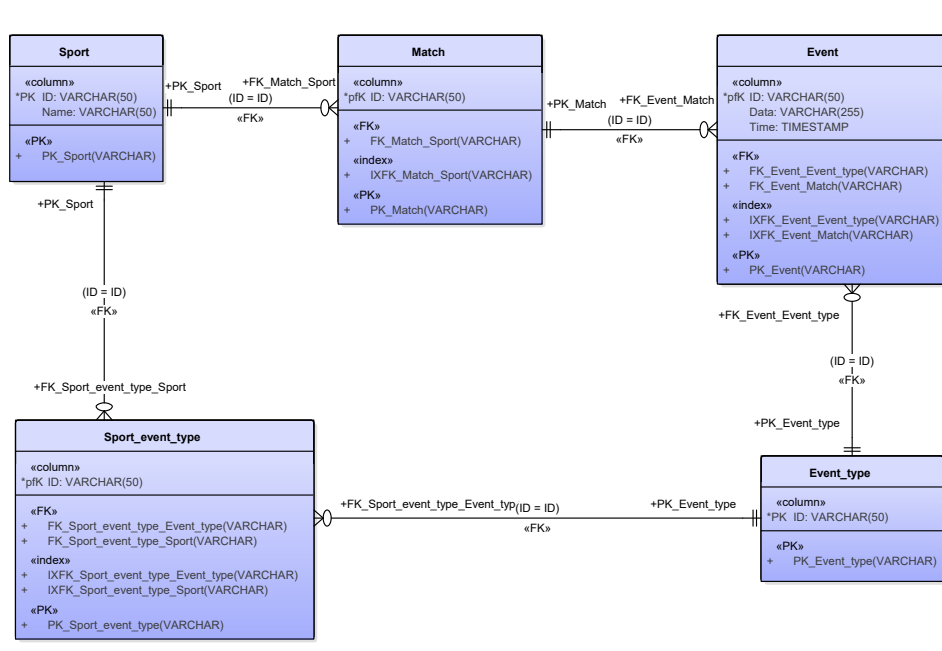
#### 3.4.1 Server

Komunikace aplikace se serverem bude probíhat následovně. Aplikace pošle požadavek na vytvoření nového zápasu s identifikací sportu. Server zápasu vygeneruje unikátní ID, zápas si zaregistruje ve své databázi a vytvoří pomocí některé z dostupných knihoven RSS kanál (3.4.3). Poté ID pošle aplikaci zpět. Dále už jen aplikace, s pomocí identifikačního ID posílá změny v zápase na server, který si je ukládá do své databáze.

#### 3.4.2 Databáze

Důležitou částí celého řešení je databáze, do které se budou ukládat všechna data o jednotlivých zápasech a server z ní poté bude informace načítat při zobrazení stavu zápasu uživatelům. Návrh relačního modelu této databáze je na obrázku 3.12.

### 3. NÁVRH



Obrázek 3.12: Relační model databáze

#### 3.4.3 RSS kanál

RSS (Rich Site Summary) je XML formát, který je používán jako informační zdroj, který dokáže uživatele automaticky upozorňovat na nové zprávy [11]. K jeho čtení je nutné mít RSS čtečku. Ta ovšem bývá obsažena ve většině internetových prohlížečích, e-mailových klientech nebo je možné si jednoduše nějakou pořídit. Aktuálně se RSS specifikace používá ve formátu 2.0, který oproti předchozím verzím obsahuje více informace jako autora či datum příspěvku.

Jeden takový RSS kanál hodlám navrhnout i já pro svou aplikaci.

```
<?xml version="1.0" encoding="ISO-8859-2"?>
```

```
<rss version="2.0">
  <channel>
    <title>SCORE TABLE: SK Slavia Praha - AC Sparta Praha
    </title>
    <link>http://scoretable.cz/match/288/rss</link>
    <description>Aktuální informace ze sportovních zápasů.
    </description>
    <language>cs</language>
    <pubDate>Tue, 4 Apr 2017 20:00:00 GMT</pubDate>

    <item>
      <title>Gól - domácí tým</title>
```

```
<link>http://scoretable.cz/match/288/</link>
<description>Domácí tým vstřelil gól (25.minuta).
Aktuální stav zápasu:
SK SLAVIA PRAHA 2:1 AC SPARTA PRAHA</description>
<author>Username123</author>
</item>
<item>

<title>Konec prvního poločasu</title>
<link>http://scoretable.cz/match/288/rss</link>
```

...



---

## Realizace

Předposlední fází mé bakalářské práce je samotná implementace. Zde budu vycházet z návrhů v předchozí kapitole, hlavně z diagramu tříd. Ten víceméně stačí přepsat a doplnit o vnitřní logiku. Samotná implementace by tedy neměla být díky důkladné analýze a návrhu nijak zvlášť složitá.

### 4.1 Diagram nasazení

Diagram nasazení popisuje fyzickou architekturu našeho řešení, neboli zobrazuje fyzická zařízení a jejich propojení [12].

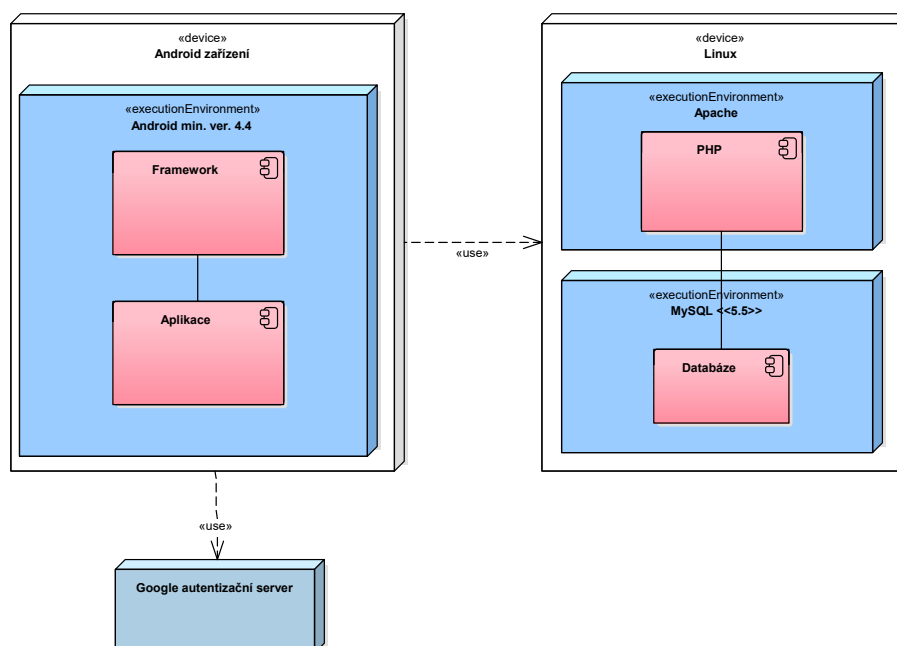
Tento diagram je znázorněn na obrázku 4.1.

Je na něm vidět jak celý systém, běžící na OS Android a složený z frameworku a aplikace, které spolu navzájem komunikují, využívá autentizační server Googlu pro přihlašování uživatelů a posílá data na server. Tento server běží na OS Linux a je tvořen webovým serverem, který komunikuje s MySQL databází.

### 4.2 Měření času

V mnoha sportech je čas důležitým prvkem. U některých je jen ukazatelem odehrané doby (např. tenis) a u některých je nezbytnou veličinou, která rozhoduje o průběhu utkání (např. fotbal).

Třída pro měření času bude součástí samotného frameworku, měla by tedy pokrývat všechny druhy sportů, případně být aplikací přizpůsobitelná (nastavitelná). Je samozřejmostí, že musí běžet na samostatném vláknu, aby nezpomalovala samotnou aplikaci. K tomu by bylo nejlepší využít třídu `AsyncTask`, případně `Service`. Bohužel tyto třídy jsou pouze součástí Android SDK a ne samotné Javy. Tedy pro framework nepoužitelné. Použijte tedy základní třídu Javy pro vytváření vláken `Thread`, ze které obě zmíněné vycházejí. Díky ní poběží čas na svém vláknu a ne na hlavním.



Obrázek 4.1: Diagram nasazení

Vytvořil jsem tedy třídu `GameTimer`, která obsahuje klasické metody pro ovládání jako např. `start()`, `stop()`, `pause()` atd. Tato třída každou vteřinu aktualizuje čas a zavolá metodu `update()`, kterou si programátor může přizpůsobit podle svého. V aplikaci tuto funkci používám k aktualizaci ukazatele času na hlavní obrazovce zápasu.

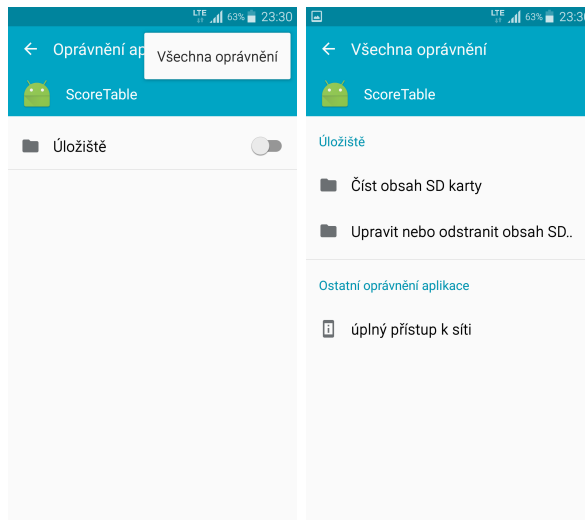
### 4.3 Práva aplikace

Kvůli bezpečnosti zařízení, každá aplikace potřebuje k některým úkonům práva od uživatele. Ve starších verzích systému tyto práva uživatel odsouhlasil už při instalaci aplikace. Od Androidu verze 6.0 je možné práva schválit až při běhu aplikace v době, kdy skutečně jsou potřeba, případně je později v nastavení zpětně vypnout/zapnout [13]. Samozřejmě čím více práv aplikace požaduje, tím menší má od uživatele důvěru. Programátor by se tedy měl snažit jich využívat co nejméně a jen ty nezbytně nutné.

Na obrázku 4.2 je ukázka pozdějšího nastavení práv aplikace.

Práva aplikace se přidávají do `AndroidManifest.xml`. Následuje seznam použitých práv v mé práci.

- `android.permission.INTERNET` – Aplikace má práva k neomezenému používání internetu



Obrázek 4.2: Nastavení práv aplikace

- `android.permission.READ_EXTERNAL_STORAGE` – Právo ke čtení z externího úložiště (ve většině případech jde o SD kartu)
- `android.permission.WRITE_EXTERNAL_STORAGE` – Právo k zapisování na externí úložiště

## 4.4 Lokalizace

Dnes je téměř samozřejmostí, aby byla aplikace multijazyčná. Jak bylo řečeno v analýze, aplikaci hodlám vytvořit v českém a anglickém jazyce. K tomu existuje v Androidu velmi praktický nástroj. Jakýkoliv text lze ukládat do souboru `strings.xml` pod libovolným jménem a na text v aplikaci pomocí jména odkázat. K lokalizaci poté programátor vytvoří druhý soubor `strings.xml`, který je stejný jako první, ale text je v jiném odpovídajícím jazyce. Tento soubor uloží do složky `values-x`, kde `x` nahradí za dvoupísmennou zkratku jazyka podle standardu ISO 639-1 [14]. Aplikace podle nastavení uživatele použije ten správný soubor a programátor se o to v samotném kódu nemusí vůbec starat. Jestliže není u složky označena žádná lokalizace, je považována za výchozí a program se do ní dívá v případě, že k dispozici není žádný jiný překlad. Následuje příklad dvou lokalizací, nejprve v češtině a poté v angličtině.

```
..\values\strings.xml
```

```
<resources>
  <string name="button_new_game">NOVÁ HRA</string>
  <string name="button_login">PŘIHLÁSIT SE</string>
  <string name="button_logout">ODHLÁSIT SE</string>
```

```
<string name="button_settings">NASTAVENÍ</string>
<string name="button_about">O APLIKACI</string>
</resources>

..\values-en\strings.xml

<resources>
  <string name="button_new_game">NEW GAME</string>
  <string name="button_login">LOGIN</string>
  <string name="button_logout">LOGOUT</string>
  <string name="button_settings">SETTINGS</string>
  <string name="button_about">ABOUT</string>
</resources>
```

Vše musí být zabaleno do XML tagů `<resources>` a dále stačí používat jen tagy `<string>` a jejich atribut `name` pro jednotlivé textové řetězce.

Při vytváření vzhledu pomocí XML se k textu dostaneme tak, že místo samotného textu napíšeme referenci na řetězec podle toho, jak jsme si ho pojmenovali. Například k textu s nastavením se dostaneme následujícím způsobem:

```
android:text="@string/button_settings"
```

A ve zdrojovém kódu aplikace ho získáme takto:

```
getString( R.string.button_settings );
```

Tento způsob lokalizace má více výhod. Například, když se v aplikaci stále opakuje stejný text, lze ho ve `strings.xml` vytvořit pouze jednou a vícekrát se na něj odkazovat. Jestliže se později rozhodneme text změnit, změníme ho pouze na jednom místě a změna se promítne všude, kde je na něj odkazováno.


## 4.5 Instalační manuál

Pro méně zkušenější uživatele je lepší doplnit instalační příručku.

### 4.5.1 Nároky na aplikaci

Aplikaci *ScoreTable* lze spustit pouze na zařízení opatřené displejem o velikosti uhlopříčky alespoň 5.1 palců a operačním systémem **Android 4.4 nebo novějším**.

### 4.5.2 Stažení a instalace aplikace

1. Ve svém zařízení přejděte do seznamu aplikací a najděte zde aplikaci **Obchod Play** . Některá zařízení nemusí seznam aplikací obsahovat. V takovém případě aplikaci najdete přímo na hlavní obrazovce. V případě, že ji stále nevidíte, zkuste se pomocí prstu posouvat doleva nebo doprava a prozkoumat tak celou hlavní obrazovku.



2. Aplikaci **Obchod Play** spusťte.
3. V horní části obrazovky se nachází vyhledávání, kde zadejte **ScoreTable** a přesuňte se na stránku s aplikací.
4. Na stránce aplikace klikněte na **Instalovat**, potvrďte požadovaná oprávnění aplikace a vyčkejte, dokud se aplikace sama nestáhne a nenainstaluje.
5. Aplikaci můžete otevřít tlačítkem **Otevřít** po dokončení instalace rovnou z **Obchod Play**, případně ji najdete na hlavní obrazovce zařízení nebo v seznamu všech aplikací.

## 4.6 Uživatelský manuál

Aplikace bude obsahovat manuál, který poradí uživatelům, kteří si s ovládním nebudou vědět rady. Tento manuál bude k dispozici jak v češtině (pro českou verzi aplikace) tak v angličtině (pro anglickou verzi aplikace). Popisuje základní funkce aplikace pro zaznamenávání fotbalového zápasu.

Níže je zobrazen manuál v češtině.

### 4.6.1 Přihlášení

Jestliže se chcete přihlásit, na hlavní obrazovce klikněte na tlačítko **Přihlásit**. Zobrazí se Vám nabídka s výčtem účtů, ke kterým se lze přihlásit.

**Google** Pro přihlášení na Google účet klikněte na bílé tlačítko s ikonkou Googlu **Přihlásit se**. Zobrazí se Vám nabídka již použitých účtů na tomto zařízení a pod ním možnost si jiný účet přidat. Řiďte se tedy dále pokyny v aplikaci.

### 4.6.2 Odhlášení

Pro odhlášení klikněte na **Odhlásit** na hlavní obrazovce. O správném odhlášení budete informováni aplikací. Případně to zjistíte tak, že v pravém horním rohu bude místo Vašeho účtu jméno *Anonym* a tlačítko pro odhlášení se změní na tlačítko pro přihlášení.

### 4.6.3 Zakoupení zápasů

Možnost zakoupení zápasů pod svým účtem najdete pod tlačítkem **Nastavení**. Abyste si mohli zápasy zakoupit, musíte být přihlášení k účtu, pro který hodláte zápasy kupovat. Aplikací budete vyzváni k zadání počtu zápasů ke koupení a poté ke zvolení a potvrzení platby. Vše proběhne přímo v aplikaci.

### 4.6.4 Změna jazyku aplikace

Aplikace je dostupná ve dvou jazycích – češtině a angličtině. Pro přepínání mezi těmito jazyky klikněte na hlavní obrazovce na **Nastavení (Settings)**. V nastavení aplikace v horní části obrazovky vyberte požadovaný jazyk kliknutím na vlaječku a vše potvrďte tlačítkem **Uložit & Ukončit (Save & Quit)**.

Kromě dvou jazyků je tu ještě třetí volba *Výchozí (Default)*. Jestliže ji zvolíte, aplikace bude ve stejném jazyce, jako je nastaven Váš operační systém. Jestliže je Váš operační systém v jiném jazyce než češtině nebo angličtině, aplikace bude anglicky.

### 4.6.5 Vytvoření a nastavení zápasu

Pro vytvoření nového zápasu klikněte na hlavní obrazovce na tlačítko **Nový zápas**. Zobrazí se Vám obrazovka s nastavením všech potřebných parametrů zápasu. Nejprve nastavte jména a loga obou soupeřů, poté počáteční parametry a nakonec typ daného zápasu. Tato část je velmi důležitá, jelikož určuje, zda-li zápas půjde do prodloužení (případně penaltových kopů) a za jakých okolností.

Až s nastavením budete hotovi, vše potvrďte tlačítkem **Vytvořit zápas**.

### 4.6.6 Ovládání fotbalového zápasu

Ovládání samotného fotbalového utkání je velmi jednoduché a intuitivní. Kvůli uživatelské přívětivosti je obrazovka pouze v režimu *landscape*, neboli na šířku. Každá z polovin obrazovky patří jednomu soupeři. V horní části obrazovky se nacházejí spíše informativní prvky o obou soupeřích a čas. Uprostřed je graficky zobrazeno, ve které fázi se zápas aktuálně nachází a dolní část slouží k ovládání utkání.

Tu si teď popíšeme detailněji. Ve středu najdete skóre, obklopené z obou stran tlačítky **+**. Pomocí těch soupeřům přidáte vstřelené góly. Může se stát, že góly přidáte chybně. K nápravě slouží tlačítko s obrázkem zahnuté šipky, umístěné v dolní pravé části obrazovky, které skóre vrátí do stavu před poslední změnou.

V levém dolním rohu se nachází tlačítko s obrázkem ozubeného kola. To slouží ke zrušení zápasu a návratu do nastavení.

Uprostřed dolní části obrazovky je nejprve tlačítko **Pauza**, které slouží k pozastavení právě probíhajícího zápasu. V případě již pozastaveného zápasu je zde místo něj tlačítko **Pokračovat**, které, jak název napovídá, je k opětovnému spuštění zápasu, který byl pozastaven.

Následuje tlačítko, které vždy obsahuje název příští fáze zápasu. Tímto tlačítkem se uživatel dostane do další fáze.

Jestliže se fáze zápasu dostane až k penaltám, použijte tlačítka **+** a **X** k úspěšnému pokutovému kopu, resp. neúspěšně proměněné penaltě toho hrá-

če/týmu, na jehož straně se nacházejí. Zpočátku jsou aktivní na obou stranách. Kterou stranu zvolíte jako první, ta jako první penalty bude zahrávat a poté se vám aktivní tlačítka budou střídat až do rozhodnutí zápasu.

### **4.7 Výsledný vzhled aplikace**

Obrázky z výsledného vzhledu aplikace se nachází v příloze v sekci C.



# Testování

Poslední fází, kterou se budu zabývat, je testování aplikace. Často se stává, že se v praxi podceňuje, ale jde o velmi důležitou součást životního cyklu vývoje softwaru. Každý střední až větší softwarový projekt se neobejde bez pořádného plánu (strategie) testování. U větších projektů existuje samostatný tým, který se soustředí pouze na samotné testování. V tomto případě jsem tím týmem já sám. Existují dokonce i techniky, jako např. *Test-driven development*, kde jsou programátory/testery napsány testy ještě před vlastním napsáním funkcionality. Tuto metodu budu u některých částí používat i já.

Níže rozeberu druhy testů, které budou mnou na aplikaci prováděny.

## 5.1 Testování programátorem

Jde vlastně o zkontrolování kódu programátorem ihned po jeho napsání. Mnohdy je lepší, aby kontrolu provedl jiný programátor, než ten, který ho napsal. V praxi je tento postup podceňovaný a některé chyby jsou zbytečně objeveny až v dalších fázích testování, což může být finančně více nákladné. [15]

Jelikož aplikaci programuji sám, tyto testy budu také provádět sám.

## 5.2 Jednotkové testy

Atomické testy každé části systému, v objektově orientovaném programování jde o testování tříd a jejich metod. Každá tato metoda se testuje oproti předpokládanému výsledku za přesně jasných vytvořených podmínek. Nejčastěji se testují krajní hodnoty, které by metodě mohly dělat problémy, zda jsou správně ošetřené.

Těmito testy hodlám kompletně testovat celý rozsah frameworku i aplikace. Programátor si může napsat testovací třídy sám, ale většinou je to zbytečné, jelikož existuje mnoho frameworků, které jsou pro jednotkové testy již připravené. Tím, který budu využívat já, je v Javě nejznámější **JUnit**, který

byl v minulosti portován i na mnoho jiných programovacích jazyků (C++, C#, Python atd.). Momentálně je ve verzi 5, který vyšla v době psaní této bakalářské práce. Mně ovšem bude vyhovovat stabilní verze 4.12.

V Android Studiu je framework JUnit již zabudovaný, tudíž ani není potřeba ho stahovat a do projektu přidávat. Při vytvoření nového projektu IDE automaticky vytvoří základní testovací ukázkovou třídu ve vlastní složce, aby byly testy odděleny od zdrojových kódů aplikace. Přesto si lze přítomnost frameworku ověřit v Gradlu (*build.gradle*), kde se v `dependencies` nachází tento řádek:

```
testCompile 'junit:junit:4.12'
```

Jeho samotné použití je velmi jednoduché. Nabízí několik typů funkcí. Jejich názvy jsou na programátorovi, ale pakliže chceme, aby je systém spustil ve správnou dobu, je nutné jejich typ zadat před samotnou definicí danými klíčovými slovy. Ve své práci si vystačím se třemi typy funkcí – `@Before`, `@Test`, `@After`. Podle jejich názvů lze jednoduše odhadnout, že funkce označená klíčovým slovem `@Before` se zavolá před testováním (např. pro vytvoření testovacích objektů), druhý typ udává samotné testovací funkce a třetí se zavolá po testování (např. uvolnění paměti).

### 5.3 Testování uživatelského prostředí

Na rozdíl od předchozích testů, kde jsme testovali funkčnost aplikace nezávisle na uživatelském prostředí, zde se zaměříme pouze na něj.

V této chvíli nám nezáleží na tom, jak samotný kód vypadá, jde zde pro nás pouze černá krabička, která po předchozích testech funguje správně a vrací správné výsledky. Tyto výsledky se ovšem ještě musejí promítnout ven, aby je uviděl samotný uživatel a aplikaci mohl plně ovládat. Je tedy důležité otestovat samotné uživatelské prostředí. Je zde několik možností, jak toho dosáhnout. První je, že vezmeme tablet, na něm spustíme aplikaci a zkusíme dělat s aplikací vše možné, co nás v tu chvíli zrovna napadne. Možná nějaké chyby objevíme, ale určitě jich mnoho přehlédneme. Lepší tedy bude sepsat si nějaký scénář, v jakém pořadí a co testovat, abychom nějakou komponentu nebo nějaký její stav neopomněli. Tento způsob je mnohem preciznější než předchozí, ale je velmi časově náročný. Mezitím, co zdlouhavě testujeme, bychom mohli programovat další části aplikace. Navíc je zde možnost, že uděláme sami chybu při testování. Lepší by tedy bylo tuto práci automatizovat.

#### 5.3.1 Výběr frameworku pro UI testování

Opět existuje několik frameworků, kterých k tomu lze využít. V Androidu jsou nejčastější dva – **Espresso** a **UI Automator**. Oba jsou součástí Android knihovny **Testing Support Library**.



Obrázek 5.1: Logo Android Testing Support Library

Rozdíl mezi nimi není nikterak zásadní. Oba jsou pouze pro Android a oba testují uživatelské prostředí aplikace. Rozdíl mezi nimi je ten, že Espresso má více přímý přístup k vnitřním layoutům, které chceme testovat, tudíž je více provázaný s aplikací. Díky tomu ovšem nemůže ovládat cokoli jiného v telefonu, než komponenty aplikace. Lze ho použít v Androidu ve verzi 2.2 (API level 8) a vyšší [16]. Naproti tomu UI Automator přistupuje k aplikaci více zvenčí. Není závislý na aplikaci, lze mu nařídít cokoli, co by mohl v telefonu udělat sám uživatel (např. zapnout Wi-fi). UI Automator je k dispozici pro Android od verze 4.3 (API level 18).

### 5.3.2 UI Automator

Po uvážení jsem si zvolil UI Automator. Jak jsem již naznačil, tento framework funguje tak, že si uživatel napíše v Javě jakýsi scénář způsobem, že pomocí funkcí určí, co se má postupně vykonávat (jako kdyby vše ovládal uživatel), dále kód zkompile do souboru s příponou *.jar* a tuto aplikaci pošle na zařízení spustí. Pak už jen může sledovat automatické ovládání zařízení programem.

### 5.3.3 Přidání do projektu

Před samotným testováním je nutné přidat příslušnou knihovnu do projektu. To se provede přidáním následujícího kódu do *Gradlu* do části *dependencies*:

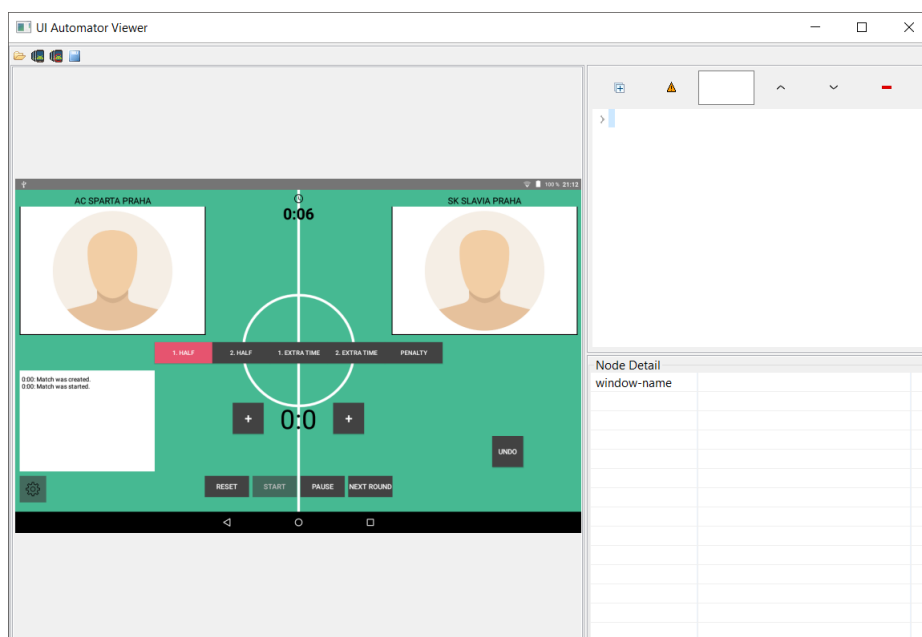
```
androidTestCompile 'com.android.support.test.' +  
    'uiautomator:uiautomator-v18:2.1.2'
```

V kapitole o jednotkových testech jsem zmínil, že Android Studio při vytváření nového projektu automaticky předpřipraví ukázkovou testovací třídu pro jednotkové testy. Vytvoří také jednu pro testování uživatelského prostředí. Tuto třídu sice nepoužiji, ale lze se jí při vytvoření své vlastní inspirovat.

### 5.3.4 UI Automator Viewer

Jelikož se v UI Automatoru pracuje s ovládacími prvky a ne vždy tester zná zdrojový kód aplikace, je nutné zjistit, jak tyto prvky identifikovat, abychom s nimi mohli pracovat. K tomu slouží program **UI Automator Viewer**, který je také součástí Testing Support Library.

## 5. TESTOVÁNÍ



Obrázek 5.2: UI Automator Viewer

Do tohoto programu se promítá reálný obsah Vaší obrazovky ať už z emulátoru nebo přímo z Vašeho zařízení propojeného s počítačem. Aktuální stav obrazovky si potom můžete programem „vyfotit“, tedy si udělat takový obrázek aktuální obrazovky. UI Automator View na tomto „snímku“ identifikuje všechny objekty a poskytne Vám k nahlédnutí všechny jejich dostupné vlastnosti, jako např. jejich ID, pomocí kterých se poté můžete v testovacím kódu s objekty spojit a ovládat je, jako kdyby je ovládal sám uživatel.

Na obrázku 5.2 je náhled aplikace pro fotbal v tomto programu.

### 5.3.5 Vytvoření testu

Pro napsání UI testu je potřeba vytvořit vlastní třídu JUnit (stejně jako u jednotkového testování) a do ní umístit testovací metodu, ve které bude samotné testování. V této metodě lze využít několik typů objektů, které jsou rozhraním mezi naším testovacím kódem a komponentami zařízení. Pomocí těchto objektů můžu zařízení ovládat. Popíšu zde několik nejdůležitějších.

- **UiDevice:** Reprezentuje zařízení uživatele. Obsahuje funkce k samotnému ovládní operačního systému. Zároveň poskytuje informace o jeho aktuálním stavu, jako např. rotaci, šířku, výšku atd.
- **UiObject:** Reprezentuje libovolný UI element v zařízení (tlačítko, label atd.).



- **UiSelector**: Využívá se k propojení `UIObject` a reálného objektu na obrazovce.
- **UiCollection**: Kolekce více UI elementů.
- **UiScrollable**: Komponenta, která umožňuje posunování (scrolování). Používáme ji, pokud se v libovolném elementu, pokud to umožňuje, potřebujeme pohybovat nahoru nebo dolů.

Tyto objekty poskytují různé funkce, jak s nimi pracovat. Ukážu několik důležitých, které v testování využívám. Všechny lze případně najít v dokumentaci [17].

Začnu funkcemi pro ovládání zařízení. Samotné zařízení si načteme následujícím způsobem:

```
UiDevice mDevice = UiDevice
    .getInstance(InstrumentationRegistry
    .getInstrumentation());
```

Jelikož není UI Automator závislý na konkrétní aplikaci, nikdy si nemůžeme být jistí, kde se právě v telefonu nacházíme v době zapnutí testu. Je tedy nutné se nejprve před každým samotným testem přesunout na hlavní obrazovku zařízení zmáčknutím *HOME* tlačítka, načtením seznamu aplikací, vyhledáním aplikace a jejím zapnutím. Pokud toto proběhne před každým testováním, máme zaručeno, že testování proběhne ve správně zapnuté aplikaci.

Ke zmáčknutí *HOME* tlačítka slouží funkce `pressHome()`. Pokud samotné zařízení nedisponuje tímto tlačítkem, operační systém Android toto tlačítko automaticky nabídne softwarově. Jestliže zmíněnou funkci uživatel zavolá, systém použije jednu z dostupných variant.

Následně využijeme funkci `wait()` objektu zařízení k tomu, abychom počkali nějakou dobu, pokud by načtení hlavní obrazovky nějakou chvíli trvalo (u pomalejších zařízeních).

Samotnou aplikaci poté spustím tímto kódem:

```
String PACKAGE = "cz.cvut.fit.scoretable.android";
Context context = InstrumentationRegistry.getContext();
Intent intent = context.getPackageManager()
    .getLaunchIntentForPackage(PACKAGE);
intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK);
context.startActivity(intent);
```

Řetězec `PACKAGE` označuje celý název aplikace, podle Java balíčkové hierarchie. Poté si načteme kontext aplikace a vytvoříme objekt typu `Intent`, který v Androidu slouží k vykonávání operací. Nakonec tento objekt odstartujeme.

Tím bychom měli zařízení připravené ve stavu k samotnému testování aplikace.

Dále si potřebujeme načíst tlačítka a následně na ně kliknout. Tlačítko si načteme pomocí některé z jeho identifikačních vlastností, kterou zjistíme pomocí programu UI Automator Viewer.

```
UiObject myButton = new UiObject(  
    new UiSelector().text("CANCEL") );
```

Tímto načteme tlačítko *CANCEL* pomocí textu, který obsahuje. Lze ho načíst i např. pomocí jeho ID, pozice, pořadí v seznamu atd. Kliknutí na něj provedeme funkcí `click()`, případně `clickAndWaitForNewWindow()`.

Jestliže chceme otestovat, zda objekt obsahuje text, který očekáváme, nejprve si z objektu metodou `getText()` samotný text vytáhneme a následně ho porovnáme s očekávanou hodnotou pomocí JUnit funkce `assertEquals()`.

S těmito úzkými základy si většinou tester vystačí. V případě, že by chtěl používat i funkce jiné, vše najde v již zmíněné dokumentaci [17].

### 5.3.6 Testovací scénáře

Vytvořím několik testovacích scénářů, podle kterých se budou mnou napsané testy řídit. Je důležité, aby pokrývaly nejdůležitější komponenty v aplikaci. Jak jsem již zmínil, UI Automator dokáže nahradit jakoukoliv interakci uživatele, nemusí komunikovat pouze s mojí aplikací. Je tedy nutné, aby sám aplikaci před samotným testováním zapnul, případně zapnul připojení k internetu, bude-li k testu potřeba.

Níže je testovací scénář přihlášení uživatele, ostatní scénáře jsou v příloze v sekci D. Rád bych dodal, že scénáře jsou psány pro aplikaci se zaměřením na fotbal, tedy primární aplikaci mé práce.

1. Zkontroluj, zda je v aplikaci v pravém horním rohu napsán text *Anonym*. Jestliže ano, pokračuj, jestliže ne, klikni na tlačítko pro odhlášení a poté pokračuj.
2. Klikni na tlačítko pro přihlášení uživatele.
3. Vyber přihlášení pod účtem Google.
4. Zvol z nabídky účet s mou emailovou adresou.
5. Zkontroluj, zda se jméno přihlášeného uživatele v pravém horním rohu shoduje se jménem mým.
6. Klikni na tlačítko pro odhlášení uživatele.
7. Opět otestuj správnost odhlášení jako v prvním kroku.

## 5.4 Uživatelské testování

Všechny dosud provedené testy by měly funkčnost aplikace prověřit dostatečně. Nicméně nezaručují, že bude přívětivá a přehledná pro uživatele, kteří

s ní nemají zkušenosti. Bylo by dobré, aby aplikaci otestoval ještě někdo zvenčí, kdo se nepodílel na jejím vzniku.

Z toho důvodu se ještě provádějí uživatelské testy (usability testing). Testovací osobě (osobám) předám jednoduché instrukce, ve kterých bude řečeno, co má uživatel provést (např. přihlaste se k účtu od Googlu). Tyto instrukce jsou velmi prosté, aby detailní provedení úkonu musel uživatel vykonat sám bez pomoci (u složitějších úkonů s dokumentací).

Jelikož každý člověk má s aplikacemi, případně sportem, různé zkušenosti (nebo vůbec žádné), je dobré vědět, jak jsou na tom jednotliví uživatelé, kteří budou mou aplikaci testovat. Vytvořil jsem tedy několik škatulek, do kterých je mohu zařadit. Nejlépe by měla být aplikace testována uživateli z každé škatulky, ale k tomu je nutné sehnat potřebné lidi a to může představovat problém. Každý, kdo bude aplikaci testovat, musí vyplnit nejdříve vstupní dotazník, kde se zařadí do správné škatulky, poté aplikaci projít podle vytvořeného scénáře a nakonec zodpovědět mnou připravené otázky, které mi mohou pomoci v dalším vývoji.

#### 5.4.1 Testující

Testování proběhlo úspěšně v laboratoři ČVUT. K dispozici jsem měl pět uživatelů, v různých věkových kategoriích, všichni s bohatými zkušenostmi s chytrými telefony. Bohužel, neměl jsem k dispozici ani jednoho respondenta, který by měl větší než pouze základní znalosti fotbalu. To беру za jedinou kaňku na samotném testování.

Dále je vypsán detailnější popis jednotlivých uživatelů, kteří se na testování aplikace podíleli. Dodávám, že jde pouze o muže.

- 35-50 let, vlastník zařízení s Androidem, výborné ovládání chytrých telefonů, znalost základů fotbalových pravidel
- 25-35 let, vlastník zařízení s Androidem, výborné ovládání chytrých telefonů, znalost základů fotbalových pravidel
- 35-50 let, vlastník zařízení s Androidem, výborné ovládání chytrých telefonů, neznalost fotbalových pravidel
- 18-25 let, vlastník zařízení s Androidem, výborné ovládání chytrých telefonů, znalost základů fotbalových pravidel
- 25-35 let, vlastník zařízení s Androidem, výborné ovládání chytrých telefonů, neznalost fotbalových pravidel

#### 5.4.2 Vstupní dotazník

Dotazník, který dostali respondenti k zodpovězení před samotným otestováním aplikace se nachází v sekci E.

### 5.4.3 Testovací scénář

Následuje seznam instrukcí, který jsem vytvořil pro uživatelské testování aplikace *ScoreTable*.

1. Přihlaste se k Google účtu pod těmito údaji:
  - Přihlašovací jméno: *scoretabletest@gmail.com*
  - Heslo: *scoretable*
2. Vytvořte nový zápas, ve kterém proti sobě nastoupí jako domácí (levý) tým *SK SLAVIA PRAHA* a jako tým hostí (pravý) *AC SPARTA PRAHA*, který je hrán dvoukolově s tím, že první zápas skončil remízou 1:1. Ostatní nastavení ponechte být tak, jak je ve výchozím stavu.
3. Spusťte zápas.
4. Zaznamenejte, že ve 25. minutě skóroval domácí tým.
5. Ve 47. minutě přejděte do druhého poločasu.
6. Ve druhém poločasu, přesněji v 76. minutě, přidejte gól týmu hostí.
7. Zjistili jste, že gól, který padl, nakonec neplatí. Vraťte tedy skóre po posledním gólu zpět.
8. V 90. minutě přidejte opět gól týmu hostí. Stav tedy bude 1:1.
9. Druhý poločas ukončete v 91. minutě a pokračujte do prodloužení.
10. V průběhu celého prodloužení nepřidávejte žádnému z týmů gól a postupně se v 105. a 121. přepněte až do pokutových kopů.
11. Penalty zahájí a zároveň vyhraje v poměru 5:3 domácí tým. *SK SLAVIA PRAHA* tedy vyhrává celý zápas.
12. Vraťte se do hlavního menu a odhlaste.

### 5.4.4 Výstupní dotazník

Výstupní dotazník, který byl účastníkům předložen k vyplnění po otestování aplikace se nachází v sekci F.

### 5.4.5 Výsledek testování

Uživatelské testování mé aplikace bylo velmi přínosné. Ukázalo se, že aplikace je v některých částech velmi nepřehledná či neintuitivní. Některé prvky nebyly dost viditelné a uživatelé si jich nevšimli hned napoprvé. Zároveň se potvrdilo, že někdo, kdo nemá s fotbalem větší zkušenosti, se v aplikaci, hlavně v ovládání samotného zápasu, bude velmi málo orientovat. Proto jsem se rozhodl na základě těchto poznatků aplikaci částečně poupravit a tyto negativa napravit. Bohužel si myslím, že fotbalový laik zkrátka nemá šanci se dobře zorientovat napoprvé v aplikaci, která slouží k zaznamenávání fotbalových zápasů. K tomuto účelu bude přiložena dokumentace, která by ho měla správně nasměrovat. Zároveň se domnívám, že jelikož je aplikace vytvořena hlavně pro fotbalové fanoušky, tak někdo, kdo fotbalu moc nerozumí, jí používat nebude. To ale nic nemění na její prvotní nepřehlednosti, kterou jsem, doufám, alespoň snížil.



---

# Závěr

Hlavním cílem této práce byl správně navržený a funkční framework, který lze použít pro vývoj aplikací na sdílení stavu zápasů libovolných sportů typu *versus*. Vedlejším úkolem bylo zároveň pomocí frameworku vyvinout aplikaci pro sdílení průběhu fotbalového utkání a prototypy pro hokej a tenis a navrhnout RSS pro integraci s jinými systémy. Toto všechno jsem bez výjimky splnil. Myslím, že jsem se během této práce zlepšil v návrhu software a v jeho vývoji na platformu Android. Bude-li to možné, rád bych se na vývoji této aplikace podílel dále a zdokonaloval ji. Také bych rád aplikaci umístil do internetového obchodu **Google Play**, aby mohla být konečně zpřístupněna.

## Možnosti rozšíření aplikace

Existuje mnoho možností, jak bych mohl v budoucnu aplikaci rozšířit. V současnosti je možné se v ní přihlašovat pomocí Google účtu, avšak žádnou funkci to zatím neplní. Tímto jsem aplikaci pouze připravil na možnost využití Google In-app Billing, tedy placení přímo v aplikaci. Uživatelé by mohli platit za speciální funkce v zápasech a tímto by aplikace mohla vydělávat.

Dále by bylo možné vytvořit plnohodnotný web s přehledem právě probíhajících zápasů. Každý uživatel, který zápas vytvoří, by mohl mít možnost nastavit, zda bude zápas viditelný pro všechny, či nikoliv.

Hlavním rozšířením by mělo být umožnit sdílet více informací o zápase (karetní tresty, sestavy, střídání atd.)

A v poslední řadě by aplikace měla držet krok s vývojem Androidu, který je v současnosti velmi svižný a jeho nepřeborné možnosti narůstají.





---

## Literatura

- [1] *PEER-TO-PEER SÍTĚ: JAK FUNGUJÍ A KDE JE PROBLÉM [online]*. [cit. 21.03.2017]. Dostupné z: <http://www.bezpecne-online.cz/surfuj-bezpecne/sosani-a-sdileni-dat/peer-to-peer-site-jak-funguji-a-kde-je-problem.html>
- [2] *Souhrnná pravidla pro fotbalové soutěže [online]*. [cit. 13.03.2017]. Dostupné z: [http://borek.ivao.cz/fotbal/pravidla\\_souhrn.html](http://borek.ivao.cz/fotbal/pravidla_souhrn.html)
- [3] *KHL nezaostává, vedení ligy potvrdilo změnu formátu prodloužení [online]*. [cit. 15.03.2017]. Dostupné z: <http://www.hokej.guru/khl-nezaostava-vedeni-ligy-potvrdilo-zmenu-formatu-prodlouzeni/>
- [4] *Hard kurty [online]*. [cit. 20.04.2017]. Dostupné z: <http://www.tenis-schmid.cz/nabidka-povrchu/hard-court.htm>
- [5] *Minitenis a Babytenis [online]*. [cit. 21.03.2017]. Dostupné z: <http://www.minitenis.cz/>
- [6] *Historie Androidu v kostce aneb Od verze 1.0 až po Android M [online]*. [cit. 13.03.2017]. Dostupné z: <https://www.svetandroida.cz/historie-androidu-201506>
- [7] *What is API Level? [online]*. [cit. 13.03.2017]. Dostupné z: <https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>
- [8] *Android - minSDKVersion, targetSDKVersion, Build Target [online]*. [cit. 13.03.2017]. Dostupné z: <https://www.skoumal.net/cs/android-minsdkversion-targetsdkversion-maxsdkversion-project-build-target/>
- [9] *Use Java 8 Language Features [online]*. [cit. 13.03.2017]. Dostupné z: <https://developer.android.com/guide/platform/j8-jack.html>

- [10] *UML 2 Activity Diagrams: An Agile Introduction [online]*. [cit. 27.03.2017]. Dostupné z: <http://www.agilemodeling.com/artifacts/activityDiagram.htm>
- [11] *Vše podstatné o RSS [online]*. [cit. 29.03.2017]. Dostupné z: <https://www.root.cz/clanky/vse-podstatne-o-rss/>
- [12] Mlejnek, J.: *Návrh – rozhraní a komponenty [online prezentace]*. České vysoké učení technické v Praze, Praha, 2017, [cit. 10.04.2017]. Dostupné z: [https://edux.fit.cvut.cz/courses/BI-SI1/\\_media/lectures/07/07.prednaska.pdf](https://edux.fit.cvut.cz/courses/BI-SI1/_media/lectures/07/07.prednaska.pdf)
- [13] *Android 6.0 Marshmallow, thoroughly reviewed [online]*. [cit. 15.04.2017]. Dostupné z: <https://arstechnica.com/gadgets/2015/10/android-6-0-marshmallow-thoroughly-reviewed/5/#h1>
- [14] *Providing Resources [online]*. [cit. 24.04.2017]. Dostupné z: <https://developer.android.com/guide/topics/resources/providing-resources.html#AlternativeResources>
- [15] *Fáze a úrovně provádění testů [online]*. [cit. 08.04.2017]. Dostupné z: <http://testovanisoftwaru.cz/category/druhy-typy-a-kategorie-testu/>
- [16] *Testing Support Library [online]*. [cit. 08.04.2017]. Dostupné z: <https://developer.android.com/topic/libraries/testing-support-library/index.html>
- [17] *android.support.test.uiautomator [online]*. [cit. 05.05.2017]. Dostupné z: <https://developer.android.com/reference/android/support/test/uiautomator/package-summary.html>

## Seznam použitých zkratk

**XML** Extensible markup language

**IDE** Integrated Development Environment

**UI** User interface

**OS** Operating system



---

## Obsah přiloženého CD

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



# Výsledný vzhled aplikace

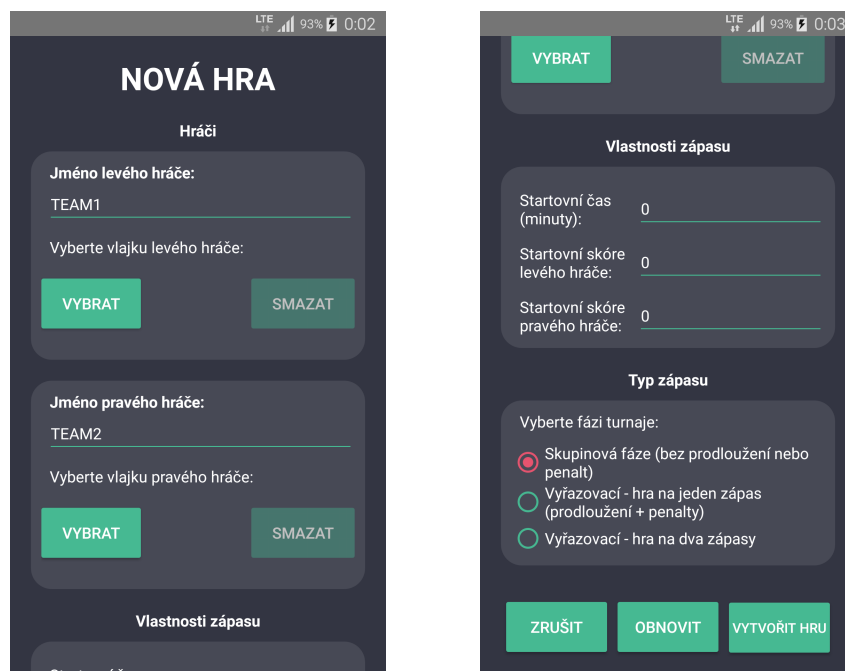
## C.1 Smartphone

Na obrázcích níže lze vidět výsledný vzhled aplikace na telefonu Samsung Galaxy S5 s operačním systémem Android 6.0.1 a displejem o velikosti uhlopříčky 5,1 palců a rozlišením 1920x1080 pixelů.



Obrázek C.1: Výsledný vzhled aplikace na telefonu – hlavní obrazovka a nastavení lokalizace

## C. VÝSLEDNÝ VZHLED APLIKACE

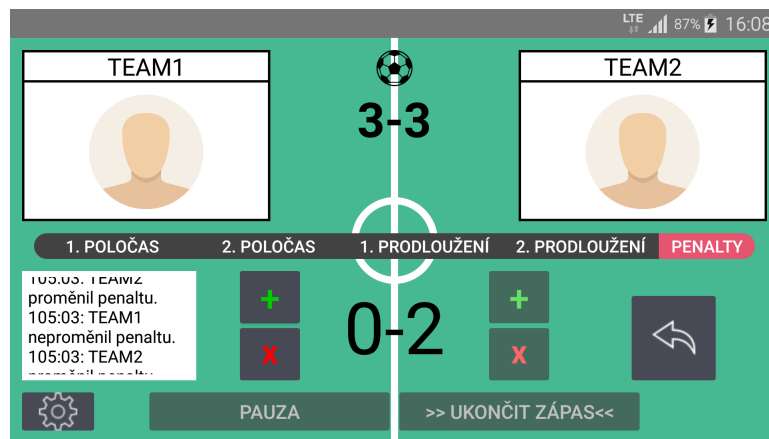


Obrázek C.2: Výsledný vzhled aplikace na telefonu – nastavení zápasu

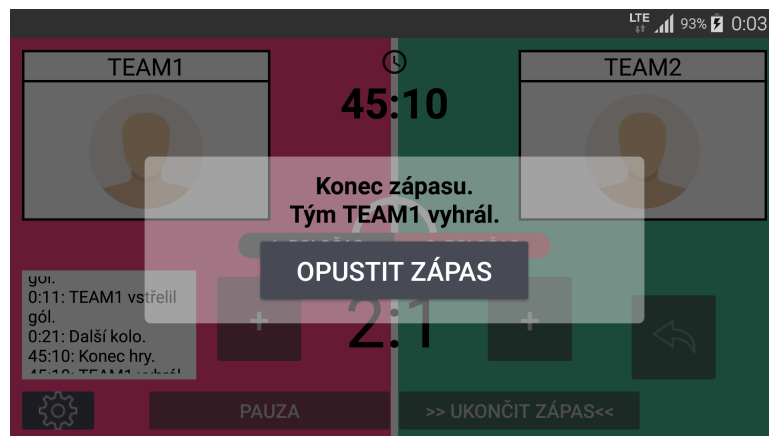


Obrázek C.3: Výsledný vzhled aplikace na telefonu – zápas (základní hrací doba)





Obrázek C.4: Výsledný vzhled aplikace na telefonu – zápas (penalty)



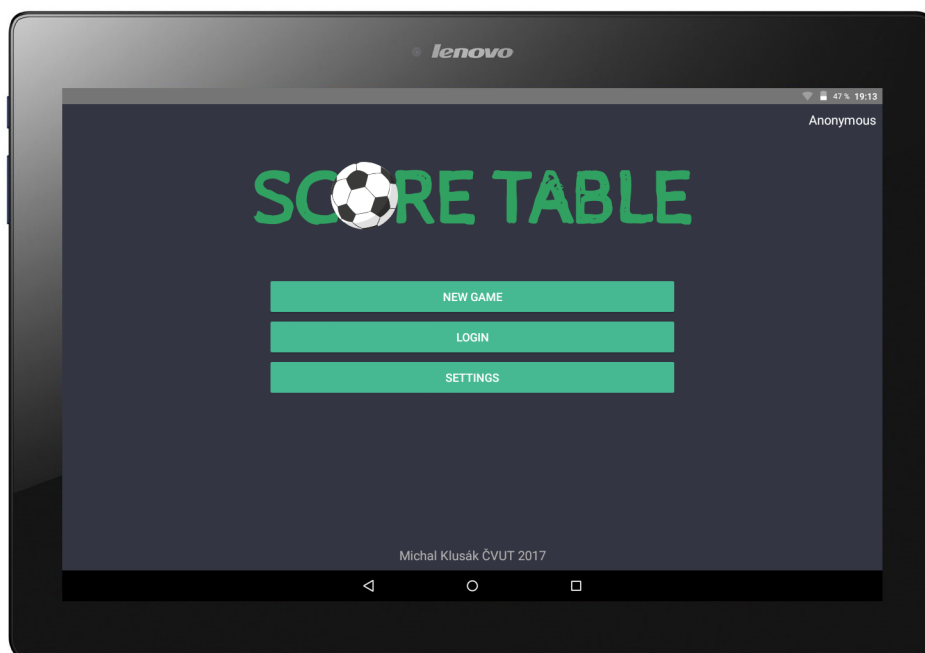
Obrázek C.5: Výsledný vzhled aplikace na telefonu – konec zápasu

## C.2 Tablet

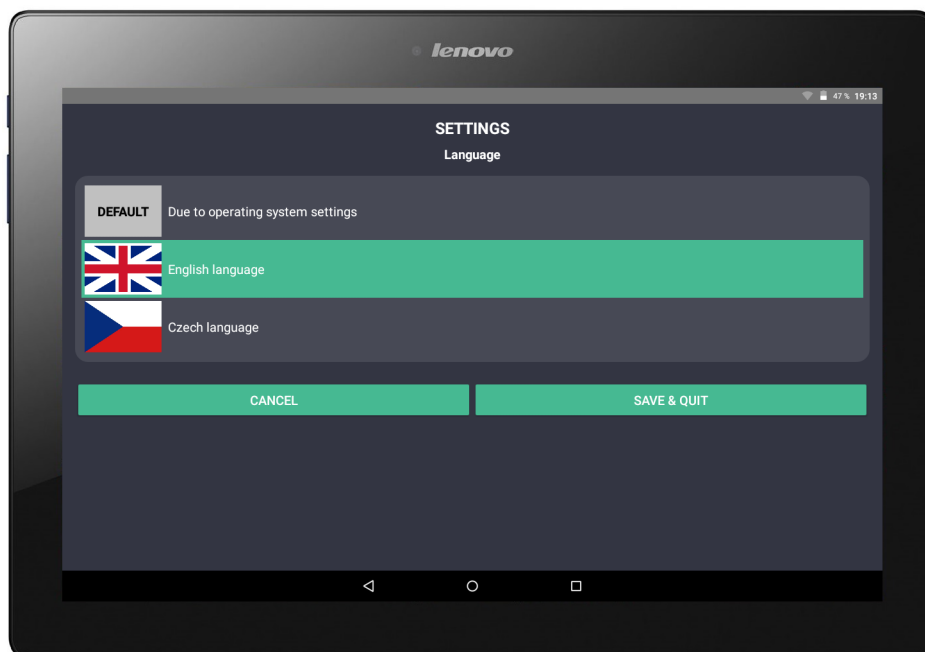
Na obrázcích níže se nachází výsledný vzhled aplikace na tabletu Lenovo TAB2 A10-30 s operačním systémem Android 5.1.1 a displejem o velikosti uhlopříčky 10,1 palců a rozlišením 1280x800 pixelů.

## C. VÝSLEDNÝ VZHLED APLIKACE

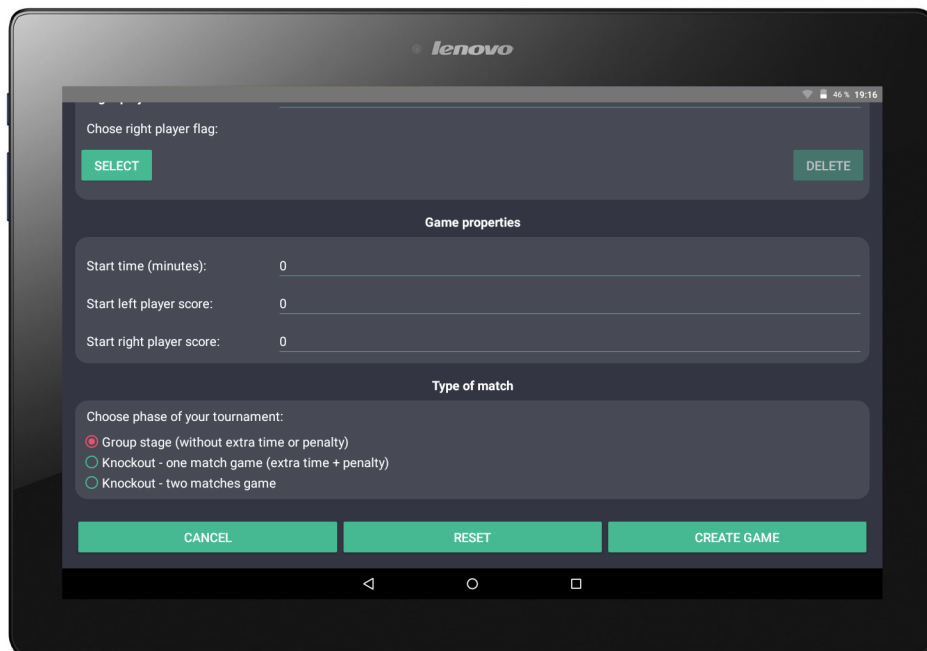
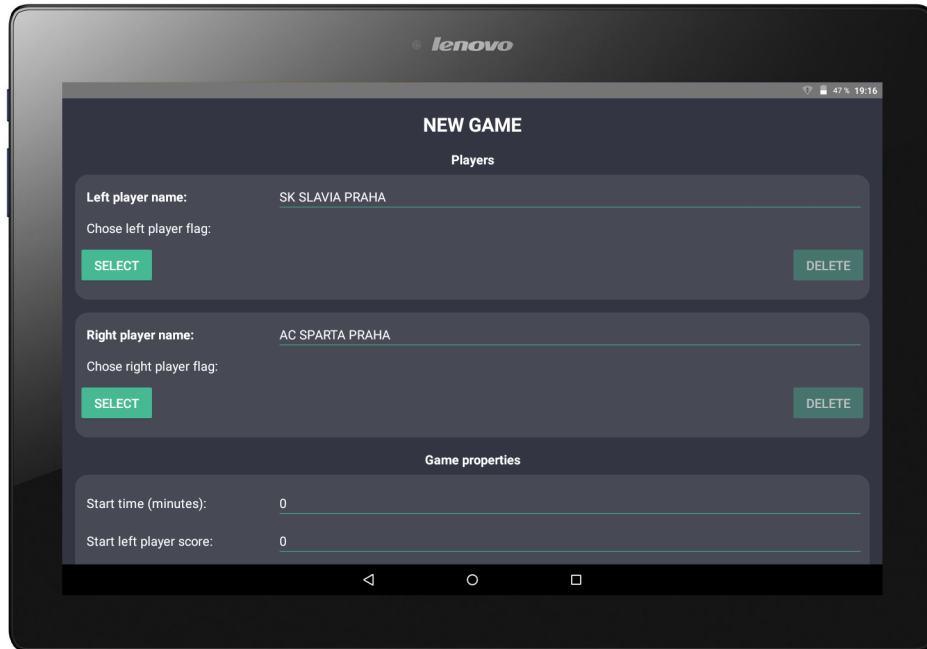
---



Obrázek C.6: Výsledný vzhled aplikace na tabletu – hlavní obrazovka



Obrázek C.7: Výsledný vzhled aplikace na tabletu – nastavení lokalizace



Obrázek C.8: Výsledný vzhled aplikace na tabletu – nastavení zápasu

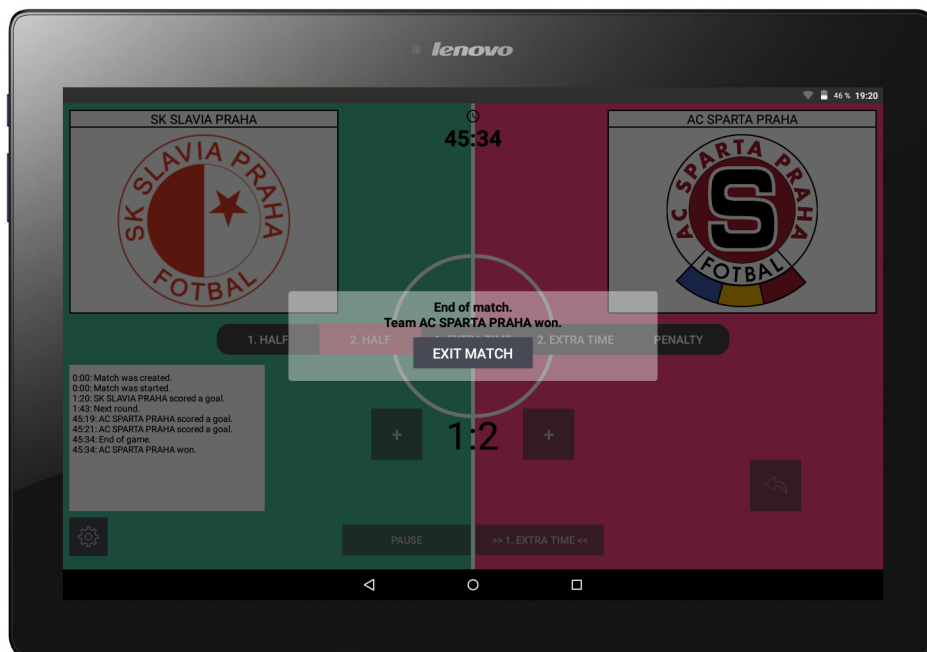
### C. VÝSLEDNÝ VZHLED APLIKACE



Obrázek C.9: Výsledný vzhled aplikace na tabletu – zápas (základní hrací doba)



Obrázek C.10: Výsledný vzhled aplikace na tabletu – zápas (penalty)



Obrázek C.11: Výsledný vzhled aplikace na tabletu – konec zápasu



---

## Testovací scénáře pro uživatelské testování

### D.1 1. testovací scénář

1. Klikni na tlačítko pro zahájení nového zápasu.
2. Do první kolonky se jménem pravého hráče napiš text *SK Slavia Praha*.
3. Do druhé kolonky pro levého hráče napiš *AC Sparta Praha*.
4. Do pole pro zadání startovního času zadej číslo 5.
5. Do pole pro zadání startovního skóre levého hráče zadej číslo 3.
6. Do pole pro zadání startovního skóre pravého hráče zadej číslo 1.
7. Ve výběru fáze turnaje zvol 3. možnost.
8. Dále, ve druhém výběru, zda jde o první nebo druhý zápas, zvol druhou možnost.
9. Toto odhalí třetí nabídku, kde zadej do prvního textového pole číslo 2 a do druhého číslo 1.
10. Proveď kliknutí na tlačítko pro obnovení nastavení.
11. Zkontroluj, zda je textové pole pro zadání jména levého hráče prázdné.
12. Zkontroluj, zda je textové pole pro zadání jména pravého hráče prázdné.
13. Zkontroluj, zda je v poli pro zadání startovního času zápasu zadáno číslo 0.
14. Zkontroluj, zda pole pro zadání startovního skóre levého hráče obsahuje číslo 0.

15. Zkontroluj, zda pole pro zadání startovního skóre pravého hráče obsahuje číslo 0.
16. Zkontroluj, zda je ve výběru fáze turnaje vybrána první možnost.

## D.2 2. testovací scénář

1. Zkontroluj, pokud třetí tlačítko obsahuje text *NASTAVENÍ*.
2. Pokud ano, klikni na něj, na další obrazovce zvol druhou možnost (Anglický jazyk) a klikni na tlačítko **ULOŽIT & ZAVŘÍT** a ověř, zda původní tlačítko teď obsahuje text *SETTINGS*.
3. Pokud ne, klikni na něj, na další obrazovce zvol třetí možnost (Český jazyk) a klikni na tlačítko **SAVE & QUIT** a ověř, zda původní tlačítko teď obsahuje text *NASTAVENÍ*.

## D.3 3. testovací scénář

1. Klikni na tlačítko pro zahájení nového zápasu.
2. Do první kolonky se jménem pravého hráče napiš text *SK Slavia Praha*.
3. Do druhé kolonky pro levého hráče napiš *AC Sparta Praha*.
4. Jako startovní čas zadej číslo 10.
5. Jako startovní skóre levého hráče zadej číslo 1.
6. Jako startovní skóre pravého hráče zadej číslo 1.
7. V nastavení typu zápasu zaškrtni třetí možnost a po otevření další nabídky zaškrtni druhou možnost.
8. Jako skóre z prvního zápasu zadej do prvního pole číslo 2 a druhého pole číslo 1.
9. Vše potvrď tlačítkem pro vytvoření nového zápasu.
10. Klikni na tlačítko **START**.
11. Zkontroluj, zda je v poli s minutami zápasu číslo 10.
12. Zkontroluj, zda je v poli skóre levého hráče číslo 1.
13. Zkontroluj, zda je v poli skóre pravého hráče číslo 1.
14. Počkej dvě minuty.



15. Zkontroluj, zda v ukazateli času je dvanáct minut.
16. Klikni na tlačítko pro přidání gólu pravému hráči.
17. Klikni na tlačítko pro přidání gólu levému hráči.
18. Zkontroluj, zda je skóre 2–2.
19. Klikni na tlačítko pro vrácení poslední akce.
20. Potvrď toto vrácení stiskem tlačítka **Ano**.
21. Zkontroluj, zda je skóre 1–2.
22. Klikni na tlačítko pro přechod do dalšího kola.
23. Klikni na tlačítko pro přechod do dalšího kola.
24. Zkontroluj, zda se zápas nachází ve fázi 1. prodloužení.
25. Klikni na tlačítko pro přechod do dalšího kola.
26. Klikni na tlačítko pro přechod do dalšího kola.
27. Zkontroluj, zda se zápas nachází ve fázi penalt.
28. Přidej levému hráči gól.
29. Klikni na tlačítko pro neproměnění penalty u pravého hráče.
30. Přidej levému hráči gól.
31. Klikni na tlačítko pro neproměnění penalty u pravého hráče.
32. Přidej levému hráči gól.
33. Zkontroluj, zda je penaltové skóre 3–0.
34. Klikni na tlačítko pro neproměnění penalty u pravého hráče.
35. Zkontroluj, zda byl zápas ukončen.
36. Klikni na tlačítko **OPUSTIT ZÁPAS**.
37. Počkej 2 vteřiny.
38. Zkontroluj, zda se aplikace nachází na hlavní obrazovce.



## Vstupní dotazník pro uživatelské testování

1. Jaká je Vaše věková skupina?
  - a) Méně než 18 let
  - b) 18-25 let
  - c) 25-35 let
  - d) 35-50 let
  - e) 50-65 let
  - f) Více než 65 let
2. Jaký operační systém ve svém telefonu používáte?
  - a) Android
  - b) Apple IOS
  - c) Windows Phone
  - d) Jiný
3. Vlastníte a alespoň občas používáte tablet?
  - a) Ano
  - b) Ne
4. Jak dobře ovládáte chytrá zařízení (telefon, tablet, atd.)?
  - a) Výborně (stahuji aplikace, používám kalendář, nastavuji zařízení atd.)
  - b) Průměrně (volám, píši zprávy atd.)
  - c) Neovládám

## E. VSTUPNÍ DOTAZNÍK PRO UŽIVATELSKÉ TESTOVÁNÍ

---

5. Jak často sledujete nějaký sport?

- a) Často
- b) Příležitostně
- c) Nesleduji

6. Jak dobře znáte pravidla fotbalu?

- a) Jsem expert
- b) Jsem fanoušek
- c) Zním pouze základy
- d) Vím, že se hraje s kulatým míčem
- e) Neznám vůbec

## Výstupní dotazník pro uživatelské testování

1. Ohodnoťte prosím jako ve škole, jak dobře se Vám aplikace používala (pouze ovládání a přehlednost, nezávisle na zkušenostech se sportem):
  - a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
2. Zasekl jste se s něčím v průběhu používání (např. nevýrazné tlačítko)?
3. Nastala v průběhu jejího používání neočekávaná chyba (závada)?
4. Narazil jste při testování na něco, co by Vás od aplikace odradilo?

## F. VÝSTUPNÍ DOTAZNÍK PRO UŽIVATELSKÉ TESTOVÁNÍ

---

5. Změnil byste na aplikaci něco?

6. Máte nějaké další komentáře k aplikaci?

Děkuji za otestování aplikace SCORE TABLE a zpětnou vazbu vyplněním dotazníků.