

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Sběr statistických údajů pro basketbalový klub
Michal Pieca

Bakalářská práce
2014

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Michal Pieca**
Osobní číslo: **I10173**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Sběr statistických údajů pro basketbalový klub**
Zadávající katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem bakalářské práce je navrhnout a implementovat aplikaci pro mobilní zařízení s operačním systémem Android se zaměřením na sběr vybraných statistických ukazatelů basketbalového týmu v reálném čase. Součástí bakalářské práce bude také zhodnocení možností pro online produkci webových stránek z nasbíraných údajů.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

[1.] Meier R. **Professional Android 4 Application Development**. 2012. ISBN-13: 978-1118102275.

[2.] Goldberg K. H. **XML: Visual QuickStart Guide (2nd Edition)**. 2008 ISBN-13: 978-0321559678.

[3.] UJBÁNYAI, Miroslav. **Programujeme pro Android**. Praha: Grada, 2012. ISBN 978-80-247-3995-3.

[4.] MLÝNKOVÁ. **ČESKÁ SPOLEČNOST PRO SYSTÉMOVOU INTEGRACI. XML technologie: principy a aplikace v praxi**. Praha: Grada, 2008. ISBN 978-80-247-2725-7.

Vedoucí bakalářské práce:

Ing. Michael Bažant, Ph.D.
Katedra softwarových technologií

Datum zadání bakalářské práce: **20. prosince 2013**

Termín odevzdání bakalářské práce: **9. května 2014**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 31. března 2014

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 6. 5. 2014



Michal Pieca

Poděkování

Chtěl bych poděkovat svému vedoucímu práce Ing. Michaelu Bažantovi, Ph.D. za cenné rady, podněty a čas, který mi byl z jeho strany věnován.

Anotace

Tato práce se zabývá návrhem a implementací aplikace pro sběr statistických údajů při basketbalovém utkání. V první kapitole jsou popsány nástroje a znalosti potřebné k programování aplikací. V další části jsou definovány a specifikovány požadavky a zbytek práce se zabývá realizací aplikace, databázovou strukturou a popisem tříd.

Klíčová slova

aplikace, operační systém Android, databáze, basketbal

Title

Statistical data collection for basketball club

Annotation

This work deals with the design and implementation of application do collecting of statistical data in basketball match. The first chapter deals with the description of the tools and knowledge needed for programming. In the next section are defined and specified requirements and the rest of the work deals with the implementation of application, database structure and class descriptions.

Keywords

application, operating system Android, database, basketball

Obsah

Úvod	11
1 Programování aplikací pro operační systém Android	13
1.1 Software	13
1.1.1 Java Development Kit (JDK)	13
1.1.2 Eclipse	13
1.1.3 Android Software Development Kit (SDK)	14
1.1.4 Android Development Tools (ADT)	14
1.2 Programovací a značkovací jazyky	14
1.2.1 Java	15
1.2.2 Extensible Markup Language (XML)	15
1.2.3 Structured Query Language (SQL)	16
1.3 Testovací zařízení	16
1.3.1 Android emulátor	16
1.3.2 Mobilní telefon, tablet	17
2 Požadavky na vyvíjenou aplikaci	18
2.1 Funkční požadavky	18
2.2 Nefunkční požadavky	20
3 Specifikace požadavků	21
3.1 Specifikace funkčních požadavků	21
3.1.1 Hlavní menu aplikace	21
3.1.2 Zakládání nového zápasu	21
3.1.3 Technický zápis	22
3.1.4 Prohlížení vytvořených zápasů	24
3.1.5 Správa družstev	26
3.1.6 Správa hráčů	27
3.1.7 Nastavení informací o klubu	28
3.1.8 Nastavení hry	28
3.2 Specifikace nefunkčních požadavků	29
3.2.1 Rychlost	29
3.2.2 Jednoduchost ovládání	29
3.2.3 Spolehlivost	29

3.2.4	Portabilita	29
3.2.5	Správné zobrazení.....	30
3.2.6	Ukládání do databáze.....	30
3.2.7	Přehlednost aplikace	30
3.2.8	Určení pro koho je aplikace.....	30
4	Databázový model	31
4.1	Přehled tabulek	32
4.1.1	Družstva.....	32
4.1.2	Družstvo majitel	33
4.2	Zápasy.....	33
4.2.1	Hráči	34
4.2.2	Hráči přítomní	35
4.2.3	Záznamy	35
4.2.4	Záznamy protihráče	35
4.2.5	Úkony	36
5	Realizace aplikace.....	37
5.1	Seznam balíčků a tříd	39
5.2	Popis tříd.....	40
5.2.1	Balíček com.app.technickyZapis.druzstvo	40
5.2.2	Balíček com.app.technickyZapis.hrac	41
5.2.3	Balíček com.app.technickyZapis.knihovna	43
5.2.4	Balíček com.app.technickyZapis.nastaveni.....	45
5.2.5	Balíček com.app.technickyZapis.technickyZapis	45
5.2.6	Balíček com.app.technickyZapis.zapas	48
6	Prezentace dat na internetu	50
	Závěr	51
	Použitá literatura	52

Seznam zkratek

ADT	Android Development Tools
ARM	Advanced RISC Machine
AVD	Android Virtual Devices
DCL	Data Control Language
DDL	Data Definition Language
DML	Data Manipulation Language
HTML	Hyper Text Markup Language
JDK	Java Development Kit
JRE	Java Runtime Enviroment
JVM	Java Virtual Machine
QEMU	Quick EMUlator
RISC	Reduce Instruction Set Computing
SDK	Software Development Kit
SQL	Structured Query Language
TCC	Transaction Control Commands
UML	Unified Modeling Language
XML	eXtensible Markup Language

Seznam obrázků

Obrázek 1 - Databázový model	31
Obrázek 2 – Vazby tabulek v databázovém modelu	32
Obrázek 3 - Ukázka vazeb třídy TechnickyZapis	38

Seznam tabulek

Tabulka 1 - Specifikace testovacích zařízení	17
--	----

Úvod

Hlavním cílem této bakalářské práce je navrhnout a naprogramovat aplikaci pro zařízení s operačním systémem Android, která by sloužila ke sběru statistických údajů během basketbalového utkání, s možností vyhodnocení těchto údajů jak pro hráče samotného, tak i pro celé družstvo.

Android je operační systém vyvinutý společností Google, který byl zpočátku určen pro mobilní zařízení. V dnešní době byl tento operační systém zaveden do dalších zařízení. Příkladem mohou být tablety, běžné počítače, televize a TV boxy. Android je v dnešní době velice rozšířený a oblíbený, vyvíjen konsorciem Open Handset Alliance, jehož cílem je neustálý vývoj mobilních technologií a možnost nabídnout bohatší, levnější a lepší mobilní systém. [1]

Hlavním důvodem pro vytvoření této aplikace je usnadnění při sběru a vyhodnocování dat. Dříve byly jednotlivé údaje zaznamenávány na papír a poté buď vypočítány na kalkulačce, nebo přepsány do počítače a vyhodnoceny. Tyto způsoby byly velice pracné a pomalé. V době zadání této bakalářské práce neexistovaly aplikace tohoto typu. Dnes už takové aplikace jsou, ale žádná z těchto aplikací neobsahuje dostatečné členění střeleckých úkonů. Tyto aplikace člení střelbu pouze na střelbu ze střední vzdálenosti, trojkovou střelbu a trestnou střelbu. Tato aplikace zmíněné činnosti rozšiřuje o nájezd a podkošovou střelbu. To povede k získání přesnějších údajů.

V první kapitole jsou shrnuty softwarové požadavky, programovací jazyky a možnosti testování aplikace. V jednotlivých částech jsou pak vyjmenovány konkrétní programy a jazyky, které jsou stručně popsány.

Druhá kapitola obsahuje seznam funkčních a nefunkčních požadavků. Funkční požadavky jsou důležité pro stanovení důležitých funkcí. Požadavky jsou rozděleny a označeny na části, které spolu souvisejí. Nefunkční požadavky stanovují celkové požadavky na aplikaci.

Třetí kapitola obsahuje specifikaci funkčních i nefunkčních požadavků. Požadavky vypsané v druhé kapitole jsou zde kompletně popsány a doplněny o další informace.

Čtvrtá kapitola obsahuje databázový model. Je zde popsáno k čemu jsou jednotlivé tabulky použity, dále pak výčet atributů a vazby mezi jednotlivými tabulkami.

Pátá kapitola slouží k popisu tříd. Nejprve jsou balíčky a třídy vyjmenovány a dále pak jednotlivě popsány, včetně některých metod, které třída obsahuje. U popisu některých tříd jsou uvedeny ukázky zdrojového kódu.

Šestá kapitola se zabývá zhodnocením možnosti prezentovat sebraná data na internetu. Jedná se pouze o teoretickou část a naprogramovaná aplikace tuto funkci neobsahuje. Je zde zhodnoceno, jaké způsoby prezentace dat na internetu by bylo možné implementovat a také jakých prostředků by bylo zapotřebí.

V závěru je zhodnocení aplikace, její využitelnost v praxi a výčet dalších funkcí, o které by aplikace mohla být do budoucna rozšířena.

1 Programování aplikací pro operační systém Android

Aplikace pro platformu Android je možné vyvíjet a testovat na počítačích s operačními systémy Windows, GNU/Linux a Mac OS. K vývoji aplikací je zapotřebí mít nainstalováno Java Development Kit, Android Software Development Kit, vývojové prostředí Eclipse a Android Developnet Tools plugin pro Eclipse. Dnes už existuje možnost pro vývojáře s operačním systémem Windows si stáhnout kompletní balíček těchto nástrojů, který je vyvinut společností Google a obsahuje všechny výše uvedené nástroje. Dále je zapotřebí znát programovací jazyk Java, značkovací jazyk XML, dotazovací jazyk SQL a samozřejmě také operační systém Android a postup programování těchto aplikací. [1]

1.1 Software

K vývoji aplikací pro operační systém Android je zapotřebí mít nainstalovaný následující software. Veškerý software je volně dostupný a nemá žádná omezení.

1.1.1 Java Development Kit (JDK)

JDK je softwarový balíček pro vývoj aplikací a apletů pro platformu Java. Balíček se skládá JRE, který obsahuje Java knihovny a virtuální stroj, díky kterému je možné spouštění Java aplikací na daném operačním systému. JDK dále obsahuje kompilátor, který zdrojový kód přeloží na byte-kód, ten je pak vykonán pomocí virtuálního stroje JVM. V JDK nalezneme i sadu základních knihoven a nástroje pro vývoj, testování a monitorování Java aplikací. [2]

1.1.2 Eclipse

Vývojové prostředí Eclipse je určeno pro programování v jazyce Java. Podporuje základní funkce pro programování a ladění aplikací. V základní verzi Eclipse jsou integrovány pouze prostředky pro standartní vývoj Javy, například kompilátor a debugger, ale neobsahuje vizuální nástroj pro grafický návrh aplikací.

Eclipse podporuje rozšíření na další programovací jazyky, například C, C++, PHP pomocí pluginů. Pluginy také umožňují rozšířit toto prostředí o zápis XML nebo HTML a návrh UML.

Právě díky těmto pluginům je Eclipse velmi silným nástrojem a je možné rozšířit toto vývojové prostředí i o možnost programovat aplikace pro operační systém Android. [3]

1.1.3 Android Software Development Kit (SDK)

Android SDK je balíček nástrojů pro vývoj aplikací pro operační systém Android. Tento balíček obsahuje nástroje pro vývoj, ladění, testování aplikací, vizuální nástroj pro návrh rozložení aplikací, správu virtuálních zařízení AVD, Android emulátor umožňující běh programu na běžných operačních systémech a další programy.

Android SDK je dostupný pro všechny hlavní platformy operačních systémů Mac OS, Windows i GNU/Linux. [4]

1.1.4 Android Development Tools (ADT)

Android Development Tools je plugin pro vývojové prostředí Eclipse, který je vytvořen tak aby propojil Android SDK s Eclipse a vytvořil tak mocný nástroj pro vyvíjení Android aplikací.

ADT rozšiřuje možnosti Eclipse o jednoduché vytváření projektů Android pomocí průvodce, vývoj uživatelských rozhraní za pomoci grafického nástroje, přidávání balíčků Android API, které umožňují vytvářet aplikace přímo pro zařízení s konkrétní verzí systému a využívat jejich funkce. Nakonec i exportovat aplikace do speciálního formátu apk, který je určen přímo k distribuci. [4]

1.2 Programovací a značkovací jazyky

Pro vývoj aplikací pro operační systém Android je důležité znát následující programovací, značkovací a dotazovací jazyky. Hlavním programovacím jazykem pro psaní aplikací je Java, ten slouží pro psaní veškerých funkcí a algoritmů. Jazyk XML je v Androidu využíván pro psaní souborů, které slouží k zobrazení aplikace na displeji. Zobrazovací soubory mají speciální strukturu a využívají speciálních značek, vyvinutých přímo pro Android. Poslední potřebný jazyk je SQL, ten slouží pro vytvoření a úpravu databáze a pro práci s daty uloženými v databázi.

1.2.1 Java

Java je objektově orientovaný programovací jazyk vyvinutý firmou Sun Microsystems v současné době vyvíjený firmou Oracle. Programovací jazyk Java je řazen mezi nejpoužívanější programovací jazyky, není závislý na architektuře zařízení a je volně přenositelný. Díky tomu je používán pro psaní programů pro různá zařízení, jako jsou například čipové karty, mobilní telefony, různá vestavěná zařízení, osobní počítače a velké distribuované systémy, kde spolupracují počítače po celém světě.

Jedna z výhod je, že syntaxe jazyka Java je podobná jako u jazyků C a C++, je však jednodušší a odstraňuje konstrukce, které dělaly programátorům potíže.

Další výhody jsou, že Java je jazyk bezpečný, jak z hlediska kontroly dat, tak i z hlediska ochrany operačního systému před nepřátelským kódem. Přestože patří mezi interpretované jazyky, je výkonný, protože překládá do strojového kódu jen opravdu to, co je potřeba. Je distribuovaný, což znamená, že je navržen pro podporu práce aplikací po síti. Je dynamický, tudíž je zde možnost rozšiřovat knihovnu za běhu programu o nové třídy a funkce z externích zdrojů nebo samotným programem. Podporuje vývoj a zpracování více vláknových aplikací. Používá silnou typovou kontrolu a tím pádem musí mít všechny použité proměnné definovaný svůj datový typ. Java využívá automatickou správu paměti, která je realizována pomocí Garbage collectoru, ten má za úkol automaticky vyhledat a uvolnit již nepoužívanou paměť pro další použití.

Mezi nevýhody se řadí pomalejší start programů, protože je kód nejprve přeložen do byte-kódu a teprve pak je převeden v JVM do strojového kódu a spuštěn. Další nevýhodou je větší paměťová náročnost při běhu programu, způsobená prostředím JVM, které musí být celé v paměti. [5]

1.2.2 Extensible Markup Language (XML)

XML patří do skupiny jazyků značkovacích, to znamená, že neslouží k vykonávání algoritmů. Jazyk XML stejně tak jako jazyk HTML používá tagy (značky), ale je navržen pouze k přenosu dat mezi aplikacemi, nikoli k zobrazování.

V tomto jazyku nejsou žádné předdefinované značky, a proto je musí vytvořit uživatel. Všechny značky jsou párové, to znamená, že každá značka musí mít i uzavírací značku, která udává informaci, kde data pro danou značku končí.

XML dokument je tvořen datovou strukturou strom. Dokument musí mít svůj kořenový element a každý element stromu může mít libovolné množství potomků a atributů.

Jazyk XML je často využíván, protože zjednodušuje sdílení a přenos dat mezi jednotlivými aplikacemi, bez nutnosti úpravy. Díky XML je možné vytvářet další jazyky. [6]

1.2.3 Structured Query Language (SQL)

SQL je jazyk dotazovací, neprocedurální, který obsahuje nástroje pro tvorbu databází a manipulaci s daty. Jazyk SQL patří mezi jazyky deklarativní, definuje tedy, co se má udělat, ale ne způsob, jakým se to má udělat. SQL jazyk se nepoužívá pro psaní samostatných programů, ale vkládá se do jiného procedurálního programovacího jazyka.

Samostatný jazyk SQL lze využít pouze při připojení pomocí příkazové řádky nebo speciálního programu na server, kde je možno přímo zadávat příkazy.

Jazyk SQL rozdělujeme podle systematické kategorizace na čtyři části. První je označována jako DDL, pomocí které můžeme nadefinovat vlastní databázové struktury, např. tabulky, indexy, pohledy, procedury a další. Těmito příkazy lze také měnit strukturu těchto objektů nebo je rušit. Druhá se nazývá DML, která slouží pro manipulaci s daty, tedy vkládání, dotazy nad databází, úprava nebo mazání. DCL je třetí částí, která slouží k řízení provozu na databázi a dále pak ke správě uživatelů a nastavování práv uživatelům databáze. Poslední čtvrtou část příkazů nazýváme TCC a ty slouží k řízení transakcí. [7]

1.3 Testovací zařízení

Aplikace pro operační systém Android je možné testovat dvěma způsoby. První způsob je na virtuálním stroji v počítači, který dokáže emulovat operační systém Android. Druhý způsob je pak přímo na zařízení s tímto operačním systémem.

1.3.1 Android emulátor

Emulátor lze použít ke spuštění, ladění a testování aplikace bez nutnosti fyzického zařízení s možností zvolit si velikost zobrazovacího zařízení. Android emulátor plnohodnotně napodobuje většinu funkcí přístroje, ale nepodporuje připojení USB,

fotoaparát a záznam videa, sluchátka, simulaci baterie, Bluetooth , Wi-Fi , NFC a OpenGL ES 2.0.

Emulátor Android provádí svou práci prostřednictvím open source technologie nazývané QEMU. Jde o stejnou technologii, která umožňuje emulaci z jednoho operačního systému na jiný, bez ohledu na procesor. QEMU umožňuje emulace na úrovni procesoru.

Díky emulátoru Android je procesor v počítači využíván na základě architektury ARM. ARM je 32bitový mikroprocesor založený na RISC, jehož jednoduchosti a rychlosti je dosaženo snížením počtu instrukcí v instrukční sadě.

ARM je široce používán v mobilních zařízeních a dalších vestavěných zařízeních, kde nižší výkon a spotřeba je důležitá. Většina mobilních zařízení na trhu používá procesory založené na této architektuře. [1]

1.3.2 Mobilní telefon, tablet

Testovat aplikace lze přímo na zařízení. Pomocí USB kabelu lze propojit zařízení s počítačem a poté v zařízení nastavit „režim ladění“ v části pro vývojáře.

K testování aplikace bude použit telefon HTC Desire S a Tracer OVO 2.0. V následující tabulce jsou uvedeny specifikace zařízení.

	HTC Desire S	Tracer OVO 2.0
Operační systém	Android 4.0.4	Android 4.0.4
Velikost displeje	3.7“	7“
Rozlišení displeje	800x480	1024x768
Frekvence procesoru	1 GHz	1.6 GHz
Vnitřní paměť	1.1 GB	8 GB
Paměť RAM	768 MB	1024 MB

Tabulka 1 - Specifikace testovacích zařízení

2 Požadavky na vyvíjenou aplikaci

Důležitou částí vývoje aplikace je definice funkčních a nefunkčních požadavků. Požadavky udávají požadované služby aplikace a omezení kladená na aplikaci. Definice požadavků slouží jako hlavní dokument pro další vývoj aplikace. Jednotlivé definice jsou zkoumány a detailně popsány v následující kapitole.

2.1 Funkční požadavky

Definice funkčních požadavků jsou rozděleny do skupin podle souvislostí s částí aplikace.

Menu - M

- M1 – Aplikace bude obsahovat hlavní menu.

Nový zápas - NZ

- NZ1 – Aplikace bude umožňovat vytvořit nový zápas prostřednictvím formuláře
- NZ2 – Formulář pro nový zápas bude obsahovat položky Soupeř, Místo, Datum a Čas
- NZ3 – Formulář bude obsahovat tlačítko pro přidání nového družstva do systému
- NZ4 – Po přidání nového družstva bude toto družstvo nastaveno do položky Soupeř jako aktuální
- NZ5 – Formulář bude obsahovat tlačítka „Zrušit“ a „Další“
- NZ6 – Po stisknutí tlačítka „Další“ bude zobrazeno okno pro výběr hráčů
- NZ7 – Aplikace bude ukládat nové zápasy do databáze spolu s vybranými hráči

Technický zápis – TZ

- TZ1 – Aplikace bude obsahovat aktivitu pro sběr dat při basketbalovém zápase.
- TZ2 – Aktivita bude obsahovat tlačítka pro jednotlivé úkony hráče na hřišti
- TZ3 – Aktivita při volbě hráče a úkonu na hřišti uloží data do databáze
- TZ4 – Aktivita bude zobrazovat čas
- TZ5 – Aktivita po skončení čtvrtiny automaticky nastaví začátek další čtvrtiny
- TZ6 – Hráče bude možné střídat
- TZ7 – Aktivita bude obsahovat menu
- TZ8 – Hráče bude možné přidat do hry
- TZ9 – Hráče bude možné odebrat ze hry
- TZ10 – Hráče bude možné přidat do zápasu
- TZ11 – Aktivita bude zobrazovat aktuální bodový stav
- TZ12 – Aktivita bude umožňovat zobrazit aktuální statistiku
- TZ13 – Aktivita bude umožňovat zaznamenávat bodový stav soupeře
- TZ14 – Aktivita bude umožňovat vzít zpět několik posledních úkonů hráčů

- TZ15 – Aktivita bude umožňovat z aktivity pro sběr dat odejít, s možností opět později pokračovat

Zápasy – Z

- Z1 – Aplikace bude obsahovat aktivitu pro správu vytvořených zápasů
- Z2 – Zápasy budou zobrazeny v seznamu
- Z3 – V seznamu zápasů bude možnost vyhledávat
- Z4 – Po zvolení zápasu bude spuštěna aktivita s informacemi o tomto zápase
- Z5 – Aktivita bude nabízet možnost prohlížet statistiku v případě, že je zápas ukončený
- Z6 – Aktivita bude nabízet možnost pokračovat v zápase v případě, že se jedná o neukončený zápas
- Z7 – Aktivita bude umožňovat zobrazit statistiku hráče, při volbě hráče ze seznamu
- Z8 – Zápas bude možné smazat

Hráči – H

- H1 – Aplikace bude obsahovat aktivitu pro správu hráčů
- H2 – Hráči budou zobrazeni v seznamu
- H3 – Aktivita obsahuje menu
- H4 – V seznamu hráčů bude možné vyhledávat
- H5 – Aktivita se seznamem hráčů bude umožňovat vytvořit nového hráče
- H6 – Po zvolení hráče ze seznamu bude spuštěna aktivita s informacemi o tomto hráči
- H7 – Aktivita bude nabízet možnost hráče upravit
- H8 – Aktivita bude nabízet možnost hráče smazat
- H9 – Aktivita bude umožňovat zobrazit statistiku hráče z určitého zápasu

Družstva – D

- D1 – Aplikace bude obsahovat aktivitu pro správu družstev
- D2 – Družstva budou zobrazena v seznamu
- D3 – Aktivita bude obsahovat menu
- D4 – V aktivitě se seznamem družstev bude možné vyhledávat
- D5 – Aktivita se seznamem družstev bude umožňovat vytvořit nové družstvo
- D6 – Po zvolení družstva ze seznamu, bude spuštěna aktivita s informacemi o družstvu
- D7 – Aktivita bude umožňovat družstvo upravit
- D8 – Aktivita bude umožňovat družstvo smazat

Nastavení – N

- N1 – Aplikace bude obsahovat nastavení s položkami Informace o týmu a Nastavení hry

- N2 – Informace o týmu budou zobrazeny v nové aktivitě
- N3 – Informace o týmu bude možné upravit
- N4 – Nastavení hry bude obsahovat položky Délka čtvrtiny a Maximální počet hráčů v zápase

2.2 Nefunkční požadavky

- Aplikace bude reagovat v reálném čase
- Aplikace bude jednoduchá na ovládání
- Aplikace bude spolehlivá
- Aplikace bude portabilní
- Aplikace bude správně zobrazována na zařízeních s různou velikostí zobrazovacího zařízení
- Aplikace bude data ukládat do databáze
- Aplikace bude přehledná
- Aplikace bude sloužit ke sběru dat pouze pro jedno družstvo

3 Specifikace požadavků

V této kapitole jsou detailně popsány výše uvedené funkční a nefunkční požadavky. Ke každému požadavku jsou doplněny informace, které naznačují, jak by měla aplikace fungovat a jakou strukturu by měla mít.

3.1 Specifikace funkčních požadavků

Specifikace funkčních požadavků slouží k detailnímu popisu definovaných požadavků.

3.1.1 Hlavní menu aplikace

Požadavek M1 – Při spuštění aplikace je uživateli zobrazeno hlavní menu, které slouží jako hlavní rozcestník aplikace. Menu osahuje tlačítka, kde má uživatel na výběr, zda chce založit nový zápas, procházet již vytvořené zápasy, spravovat hráče nebo družstva, položka nastavení a možnost ukončit aplikaci.

3.1.2 Zakládání nového zápasu

Požadavek NZ1 – Při zakládání nového zápasu, je uživateli zobrazen formulář, kde je možné vyplnit údaje o zápase.

Požadavek NZ2 – Na prvním místě je zobrazena rozbalovací nabídka pro volbu soupeře, kde jsou uvedeny názvy družstev. Dále je zde volba místa zápasu, v podobě další rozbalovací nabídky, které slouží k určení, zda se jedná o domácí nebo o venkovní utkání. Na třetím a čtvrtém místě je zobrazeno datum, které je automaticky nastaveno na aktuální datum a čas v podobě editovacího textového pole. Při volbě prvku Datum nebo Čas, je vždy zobrazeno nové dialogové okno. V případě volby prvku Čas je uživateli zobrazeno okno s možností nastavit hodinu a minutu začátku zápasu. Čas v dialogovém okně je automaticky nastaven na aktuální hodinu. Při volbě data je zobrazeno dialogové okno s aktuálním datem. Po nastavení je datum a čas nastaven do příslušného textového editovacího pole.

Požadavek NZ3 – Vedle nabídky pro výběr soupeře je tlačítko se symbolem plus (+), které slouží k spuštění formuláře pro přidání nového družstva do systému.

Požadavek NZ4 – Při stisknutí tlačítka pro přidání nového družstva, je uživateli zobrazen formulář, kde je nutno vyplnit údaje o družstvu. Všechny údaje jsou povinné a

uživatel je musí vyplnit. Po uložení nového soupeře do databáze je uživateli znovu zobrazen formulář pro vytváření nového zápasu. Nově přidaný soupeř je nastaven jako aktuální v nabídce pro výběr soupeře.

Požadavek NT5 – Na spodní straně formuláře jsou tlačítka „Zrušit“ a „Další“. Tlačítko „Zrušit“ slouží pro návrat do hlavního menu. Tlačítko „Další“ pro spuštění aktivity se seznamem hráčů.

Požadavek NZ6 – V případě, že jsou všechny údaje vyplněny, je po stisknutí tlačítka „Další“ uživateli zobrazeno okno pro výběr hráčů, kteří se zúčastní zápasu. V seznamu jsou zobrazeni všichni hráči, kteří jsou uloženi v databázi a jsou aktivní. Volba hráče proběhne pomocí zaškrtačacího políčka, přičemž maximální počet hráčů je deset až dvanáct v závislosti na nastavení hry. Při dosažení nastavené hodnoty není možno zvolit dalšího hráče, ale je možné již zvoleného hráče odebrat a zvolit jiného.

Požadavek NZ7 – Po dokončení nastavení a stisknutí tlačítka „Hotovo“ jsou údaje uloženy do databáze. Informace o zápase jsou uloženy do tabulky zápasů a vybraní hráči do tabulky přítomných hráčů. Dále se okno uzavře a spustí se okno s technickým zápisem.

3.1.3 Technický zápis

Požadavek TZ1 – Hlavní částí aplikace je aktivita pro sběr dat při utkání. Tato aktivita je rozdělena na více částí. První dvě dále popsané části, jsou pro sběr dat nejdůležitější. V první části je seznam právě hrajících hráčů, v druhé části samotné činnosti, které mohou být při hře vykonány.

Požadavek TZ2 – Mezi činnosti, které hráč může na hřišti vykonat, patří podkošová střelba, nájezd, střela ze střední vzdálenosti, střela zpoza trojkové čáry, trestný hod, doskok obranný, doskok útočný, získaný míč, ztracený míč, faul na hráče, faul hráče, asistence a blok. Jednotlivé činnosti jsou v aplikaci reprezentovány tlačítky.

Požadavek TZ3 – Sběr dat probíhá zvolením jednoho právě hrajícího hráče ze seznamu hráčů a poté výběrem činnosti, kterou vykonal. Pomocí těchto dvou hlavních částí je zaznamenávána většina údajů z utkání. Seznam hráčů je na obrazovce reprezentován polem přepínačů, přičemž vždy je aktivní pouze jeden.

Požadavek TZ4 – Basketbalový zápas je rozdělen ve většině soutěží na čtyři desetiminutové a v NBA na čtyři dvanáctiminutové úseky s možností neomezeného počtu pětiminutových prodloužení, při nerozhodném výsledku zápasu. V další části je tedy zobrazen čas zápasu, který je možné zastavit nebo spustit. Čas se automaticky zastavuje při přidávání hráčů do zápasu, přidávání hráčů do hry, odebrání hráčů ze hry a při střídání. Čas je spouštěn a zastavován stisknutím tlačítka s aktuálním časem.

Požadavek NZ5 – Čas je automaticky nastaven po konci čtvrtiny na začátek další čtvrtiny, pokud se nejednalo o čtvrtinu poslední. Po odehrání poslední čtvrtiny je zobrazen dotaz, zda chce uživatel zápas ukončit. Potvrdí-li uživatel ukončení zápasu, nebude už dále možné přidávat hráčům úkony a do databáze se uloží ukončení zápasu. Nepotvrdí-li uživatel ukončení zápasu, bude možné hráčům ještě přidat úkony a v seznamu zápasů se zobrazí jako neukončený.

Požadavek TZ6 – Aktivita obsahuje možnost střídat hráče na hřišti za jiného přítomného hráče, který na hřišti aktuálně není. Po stisknutí tlačítka „Střídat“ je zobrazeno dialogové okno rozdělené na dvě části. V první části jsou hráči ve hře a v druhé části jsou hráči ke střídání. Aktivita hlídá, aby byl zvolen stejný počet hráčů střídajících a hráčů ke střídání.

Požadavek TZ7 – Při stisknutí tlačítka menu je uživateli zobrazeno menu s položkami: „Statistika“, „Přidat hráče do zápasu“, „Přidat hráče do hry“, „Odebrat hráče ze hry“ a „Konec“.

Požadavek TZ8 – Při volbě položky „Přidat hráče do hry“ z menu je uživateli zobrazeno dialogové okno se seznamem hráčů, kteří byli přidáni do zápasu. Ke každému hráči je přiřazeno zaškrtačací pole, které slouží k volbě hráče. Při volbě hráčů je kontrolováno, aby počet zvolených hráčů nepřesáhl maximální povolený počet hráčů na ve hře. Po přidání hráčů je seznam hráčů na hřišti aktualizován.

Požadavek TZ9 – Při volbě položky „Odebrat hráče ze hry“ z menu je uživateli zobrazeno dialogové okno se seznamem hráčů, kteří jsou aktuálně ve hře. Ke každému hráči je přiřazeno zaškrtačací pole, které slouží k volbě hráče. Po odebrání hráčů je seznam hráčů na hřišti aktualizován.

Požadavek TZ10 – Aktivita umožňuje přidat hráče do zápasu i během zápasu. Při volbě položky „Přidat hráče do zápasu“ z menu, je uživateli zobrazeno dialogové okno se seznamem hráčů. V seznamu jsou zobrazeni pouze ti hráči, kteří ještě nebyli do zápasu přidáni. Ke každému hráči je přiřazeno zaškrtačací pole, které slouží k volbě hráče. Při volbě hráčů je kontrolováno, aby nebyl přesážen maximální povolený počet hráčů v zápasu. Po zvolení hráčů jsou přidání hráči uloženi do databáze k danému zápasu.

Požadavek TZ11 – V horní části aplikace je zobrazen aktuální bodový stav obou družstev. Bodový stav je aktualizován při přiřazení činnosti hráči nebo přidání bodů soupeři.

Požadavek TZ12 – Kdykoli během zápasu je možné zobrazit aktuální vyhodnocení výsledků družstva a hráčů. Statistika je zobrazena v tabulkách. Tuto aktivitu lze zobrazit stisknutím tlačítka menu a volby položky „Statistika“. Na prvním místě je zobrazena tabulka statistik pro celé družstvo a pod ní statistiky pro jednotlivé hráče spolu s počtem bodů a koeficientem užitečnosti. Tabulky pro hráče jsou označeny jménem hráče.

Požadavek TZ13 – Aktivita umožňuje přidávat body soupeři. Aplikace slouží pro sběr statistických údajů pouze pro jedno družstvo a tak pro přidání bodů soupeři jsou zde pouze tři tlačítka podle bodového ohodnocení za proměnný koš.

Požadavek TZ14 – Záznamy hráčů na hřišti jsou ukládány do databáze a zároveň do pole. Toto pole slouží pro možnost vzít posledních několik zadaných činností zpět při chybě uživatele. Například při překlepnutí nebo přiřazení činnosti nesprávnému hráči.

Požadavek TZ15 – Při opuštění aktivity jsou uloženy potřebné údaje do databáze, pro možnost pozdějšího pokračování v zápasu. Při spuštění neukončeného zápasu, jsou data do aktivity načtena a nastavena.

3.1.4 Prohlížení vytvořených zápasů

Požadavek Z1 – V aplikaci je možné prohlížet vytvořené zápasy. Do této sekce se lze dostat z hlavního menu zvolením položky „Zápasy“.

Požadavek Z2 – Zápasy jsou zobrazeny v seznamu, který je rozdělen na zápasy ukončené a přerušené. Zápasy jsou v seznamu seřazeny od nejnovějšího po nejstarší.

Každý zápas je označen datem a názvem soupeřova družstva. Přerušené zápasy jsou označeny jako neukončené.

Požadavek Z3 – V seznamu zápasů je možné vyhledávat. Po stisknutí tlačítka se symbolem lupy je zobrazeno vyhledávací pole. Během zadávání vyhledávacího dotazu je seznam filtrován a zobrazuje výsledky hledání, které odpovídají zadanému řetězci. Všechny zápasy je možné opět zobrazit buď smazáním vyhledávacího dotazu, nebo stisknutím symbolu křížek (X) vedle vyhledávacího pole. Vyhledávací pole je možné skrýt opětovným stisknutím tlačítka se symbolem lupy.

Požadavek Z4 – Po zvolení libovolného zápasu ze seznamu, jsou zobrazeny informace o tomto utkání. Mezi tyto informace patří název soupeře, datum zápasu, zda se jedná o utkání domácí či utkání na hřišti soupeře, skóre a seznam hráčů, kteří se utkání účastnili.

Požadavek Z5 – Jestliže je zápas ukončen, je zde zobrazeno tlačítko „Statistika“. Při stisknutí tohoto tlačítka je spuštěna aktivita, kde jsou zobrazeny na prvním místě statistické výsledky pro celé družstvo a dále pak statistika pro jednotlivé hráče. Ve statistice se nachází rozepsané jednotlivé střelecké úkony s vypočítanou úspěšností střelby v procentech, a pod nimi další úkony hráče. Ve statistice pro hráče je zobrazen čas, který hráč odehrál, počet bodů a koeficient užitečnosti, který je vypočítán ze sebraných údajů. Prohlížet statistiku lze také stisknutím tlačítka menu a volbou „Statistika“. Toho se dá využít především při neukončeném utkání.

Požadavek Z6 – Není-li zápas ukončen, místo tlačítka „Statistika“ je zobrazeno tlačítko „Pokračovat v zápase“. Při stisknutí tohoto tlačítka je spuštěna aktivita pro sběr dat, kde jsou načteny údaje z tohoto utkání, které byly uloženy při přerušení aktivity pro sběr dat.

Požadavek Z7 – Při volbě hráče ze seznamu je spuštěna aktivita, kde je zobrazena statistika zvoleného hráče v zápase. V aktivitě je uvedeno jméno hráče, počet odehraných minut, počet proměněných bodů, koeficient užitečnosti, střelecké statistiky a další úkony.

Požadavek Z8 – Vybraný zápas je možné smazat. V takovém případě jsou smazány všechny údaje o zápase. To znamená, že nebudou zobrazeny ani mezi odehranými zápasy ani ve statistikách hráčů.

3.1.5 Správa družstev

Požadavek D1 – Aplikace obsahuje aktivitu pro prohlížení družstev uložených v systému. Aktivitu se seznamem družstev je možné spustit při zvolení položky „Družstva“ v hlavním menu aplikace.

Požadavek D2 – V aktivitě se seznamem družstev jsou zobrazena všechna aktivní družstva, která jsou uložena v databázi. Seznam je řazen abecedně dle názvu družstva.

Požadavek D3 – V horní části aplikace se nachází menu, které obsahuje možnosti vyhledávání a přidání nového družstva.

Požadavek D4 – Aktivita umožňuje vyhledat družstvo v seznamu. Vyhledávání je značeno obrázkem lupy a při jejím stisknutí se objeví textové editovací pole pro zadání názvu družstva. Družstva se filtrují během psaní vyhledávacího dotazu. Všechna družstva je možné opět zobrazit buď smazáním vyhledávacího dotazu, nebo stisknutím symbolu křížek (X) vedle vyhledávacího pole. Při opětovném stisknutí symbolu pro vyhledávání je vyhledávací pole skryto.

Požadavek D5 – Při stisknutí tlačítka pro přidání nového družstva je spuštěna aktivita s formulářem, který obsahuje textové editovací pole pro název družstva. Po stisknutí tlačítka „Hotovo“ je zkontrolováno, zda je název vyplněn. V případě, že název vyplněn není, je uživatel informován o nutnosti vyplnit údaj. Jestliže jsou údaje vyplněny, je družstvo uloženo do databáze a aktivita pro vytvoření nového družstva je ukončena.

Požadavek D6 – Při výběru družstva ze seznamu je spuštěna aktivita s informacemi o tomto družstvu. Družstvo je možné upravit nebo smazat.

Požadavek D7 – Při stisknutí tlačítka menu v aktivitě s informacemi o družstvu a zvolení položky „Upravit“ je zobrazeno nové okno s textovým editovacím polem, které je předvyplněné názvem zvoleného družstva. Při uložení je zkontrolováno, zda je název zadán a jestliže je vše v pořádku, informace se uloží do databáze, ukončí se okno pro úpravu a jsou znovu zobrazeny informace o družstvu s upravenými daty. V opačném případě je uživatel varován o nutnosti vyplnit informace.

Požadavek D8 – Při stisknutí volby „Smazat“ z nabídky menu je zobrazen dotaz na smazání družstva. Jestliže je dotaz potvrzen, družstvo je nastaveno na neaktivní, okno s informacemi o družstvu je uzavřeno a je zobrazen aktualizovaný seznam družstev.

Družstva nejsou z databáze odstraněny úplně z důvodu možnosti zobrazení odehraných zápasů a statistik.

3.1.6 Správa hráčů

Požadavek H1 – Aplikace obsahuje aktivitu pro správu hráčů. Tato aktivita lze spustit z hlavního menu aplikace stisknutím tlačítka „Hráči“.

Požadavek H2 – Hráči jsou v aktivitě zobrazeni v seznamu, který je řazen abecedně podle příjmení. Hráči jsou získáni z databáze. V seznamu jsou zde zobrazeni pouze aktivní hráči.

Požadavek H3 – V horní části této aktivity je zobrazeno menu s možnostmi vyhledávání, které je značeno lupou a s možností přidat nového hráče, která je značena symbolem plus (+).

Požadavek H4 – Při volbě vyhledávání je zobrazeno editovací textové pole, do kterého lze vepsat jméno, příjmení či číslo hráče. Během psaní je seznam filtrován a jsou zobrazeni pouze hráči s odpovídajícím vyhledávacím dotazem. Všechny hráče je možné opět zobrazit smazáním vyhledávacího dotazu nebo stisknutím symbolu křížek (X) vedle vyhledávacího pole. Při opětovné volbě symbolu pro vyhledávání je vyhledávací pole skryto.

Požadavek H5 – Při zvolení volby pro přidání nového hráče je spuštěna aktivita s formulářem, který obsahuje textová editovací pole pro jméno, příjmení a číslo dresu, které jsou povinné a dále pak nepovinný údaj datum narození. Při výběru položky datum narození je zobrazeno dialogové okno pro výběr data. Při vyplnění všech povinných údajů a stisknutí tlačítka „Hotovo“ je hráč uložen do databáze, okno pro vytvoření nového hráče je uzavřeno a opět je zobrazen seznam hráčů včetně nově přidaného hráče. Nejsou-li údaje v pořádku, uživatel je varován a požádán o nápravu.

Požadavek H6 – Je-li zvolen hráč ze seznamu, spustí se aktivita s údaji o tomto hráči. Je zde uvedeno jméno, příjmení, číslo dresu a datum narození.

Požadavek H7 – Údaje o hráči je možné upravit stisknutím tlačítka menu a volbou „Upravit“. V tomto případě je zobrazen formulář, který je stejný jako při vytváření nového hráče, již s předvyplněnými editovacími poli a informacemi o hráči. Po úpravě jsou

informace uloženy do databáze, formulář je uzavřen a je znovu zobrazeno okno s informacemi o hráči s aktuálními údaji.

Požadavek H8 – V menu aktivity se nachází také položka „Smazat“ a při její volbě je zobrazeno dialogové okno s dotazem na smazání. Je-li smazání potvrzeno, hráč je nastaven jako neaktivní a okno s informacemi je uzavřeno. Hráči nejsou z databáze mazáni úplně, ale jsou nastaveni na neaktivní. Kdyby byl hráč odstraněn z databáze, v zobrazení zápasů, kterých se zúčastnil, by už nebyl zobrazen a zápis by již nebyl správný.

Požadavek H9 – Aktivita s informacemi o hráči obsahuje také tlačítko „Statistika“, při jeho stisknutí je zobrazen seznam zápasů, kterých se hráč účastnil. Při zvolení zápasu je zobrazena tabulka se statistickými údaji. Kde je uveden odehraný čas, koeficient užitečnosti, tabulka úspěšnosti střelby a tabulka s dalšími úkony.

3.1.7 Nastavení informací o klubu

Požadavek N1 – V hlavním menu aplikace je položka „Nastavení“, která slouží pro spuštění aktivity s položkami Informace o týmu a Nastavení hry.

Požadavek N2 – Aplikace umožňuje nastavit základní údaje o klubu, který aplikaci využívá. Toto nastavení se skrývá pod položkou „Informace o týmu“. Okno s informacemi o družstvu obsahuje: Název družstva, Adresa haly, Jméno trenéra a Kontakt. Název družstva je zobrazen v aktivitě pro sběr dat. Informace o družstvu jsou uloženy v databázi.

Požadavek N3 – Informace o družstvu je možné upravit. Po stisknutí tlačítka menu se zobrazí položka „Upravit“ a při jejím zvolení je spuštěna nová aktivita s formulářem. Formulář obsahuje textová editovací pole, která jsou předvyplněná aktuálními informacemi o družstvu. Po úpravě jsou informace uloženy a je opět zobrazeno okno s informacemi o družstvu.

3.1.8 Nastavení hry

Požadavek N4 – Nastavení hry slouží pro nastavení délky čtvrtiny a nastavení maximálního počtu hráčů v zápase. Do tohoto nastavení se lze dostat z hlavního menu zvolením položky „Nastavení“ a dále pak položky „Nastavení hry“. Nastavení probíhá zvolením hodnot z rozbalovacích nabídek, kde délka čtvrtiny má možnosti deset nebo dvanáct minut a počet hráčů má možnosti deset, jedenáct a dvanáct. Možnost nastavení

délky čtvrtiny je zde z důvodu možné budoucí úpravy pravidel a možnost použití aplikace ve více soutěžích. Počet hráčů účastnících se utkání se v různých soutěžích liší, a proto je zahrnuto v nastavení hry. Nastavení maximálního počtu hráčů se projeví při přidávání hráčů do zápasu.

3.2 Specifikace nefunkčních požadavků

Nefunkční požadavky slouží k stanovení požadavků na aplikaci jako na celek. Stanovují například požadavky na rychlost a spolehlivost aplikace.

3.2.1 Rychlost

Aplikace slouží k zadávání dat při basketbalovém utkání a tudíž je žádoucí, aby veškeré zadané údaje byly v co nejkratším čase zpracovány a uloženy do databáze. Jakákoliv prodleva v systému by mohla způsobit získání nepřesných údajů.

3.2.2 Jednoduchost ovládání

Provedení jakékoli akce v aplikaci musí být rychlé a jednoduché. Uživatel nesmí být ničím zdržován, aby stihl zadat všechny potřebné údaje, které byly provedeny na hřišti, v co nejkratším čase. Je tedy potřeba, aby aplikace byla rozdělena do přehledných úseků a aby veškeré potřebné prvky byly dobře viditelné na obrazovce.

3.2.3 Spolehlivost

Aplikace musí být spolehlivá. Nesmí nastat ztráta dat při náhlém ukončení aplikace nebo při neočekávaném chování uživatele. Všechny uživatelské vstupy musí být ošetřeny a kontrolovány.

3.2.4 Portabilita

Aplikace musí správně fungovat na všech zařízeních s operačním systémem Android od verze 2.2 nazývané Froyo až po verzi 4.4 s názvem Kitkat. Minimální verze 2.2 je zvolena z důvodu podpory aplikace Google Play sloužící k vyhledání a stažení aplikací v databázi, která na zařízeních s nižší verzí nebyla podporována. Dalším důvodem je, že se tento operační systém rychle vyvíjí a v dřívějších verzích by nemusely být některé funkce použité v aplikaci podporovány.

3.2.5 Správné zobrazení

Aplikace musí být vyvíjena tak, aby byla grafická prezentace zobrazována správně na zařízeních s různými velikostmi a rozlišením displejů. Je tedy potřeba navrhnout rozložení tak, aby se aplikace na všech zařízeních zobrazovala stejně, nebo alespoň podobně, podle toho, na jakém zařízení je nainstalována, tzn. aby nevznikaly případy, že se zobrazí pouze část aplikace nebo ve špatném rozložení.

3.2.6 Ukládání do databáze

Pro aplikaci je potřeba navrhnout a vytvořit databázi, která bude schopna pojmout všechna data z aplikace. Bude využita interní databáze zařízení, ve které budou vytvořeny tabulky pro družstva, zápasy, hráče, přítomní hráče na zápase, záznamy ze zápasu a jednotlivé úkony hráče. Databáze bude sloužit jako hlavní úložiště všech získaných dat a budou zde uloženy ihned po provedení akce.

3.2.7 Přehlednost aplikace

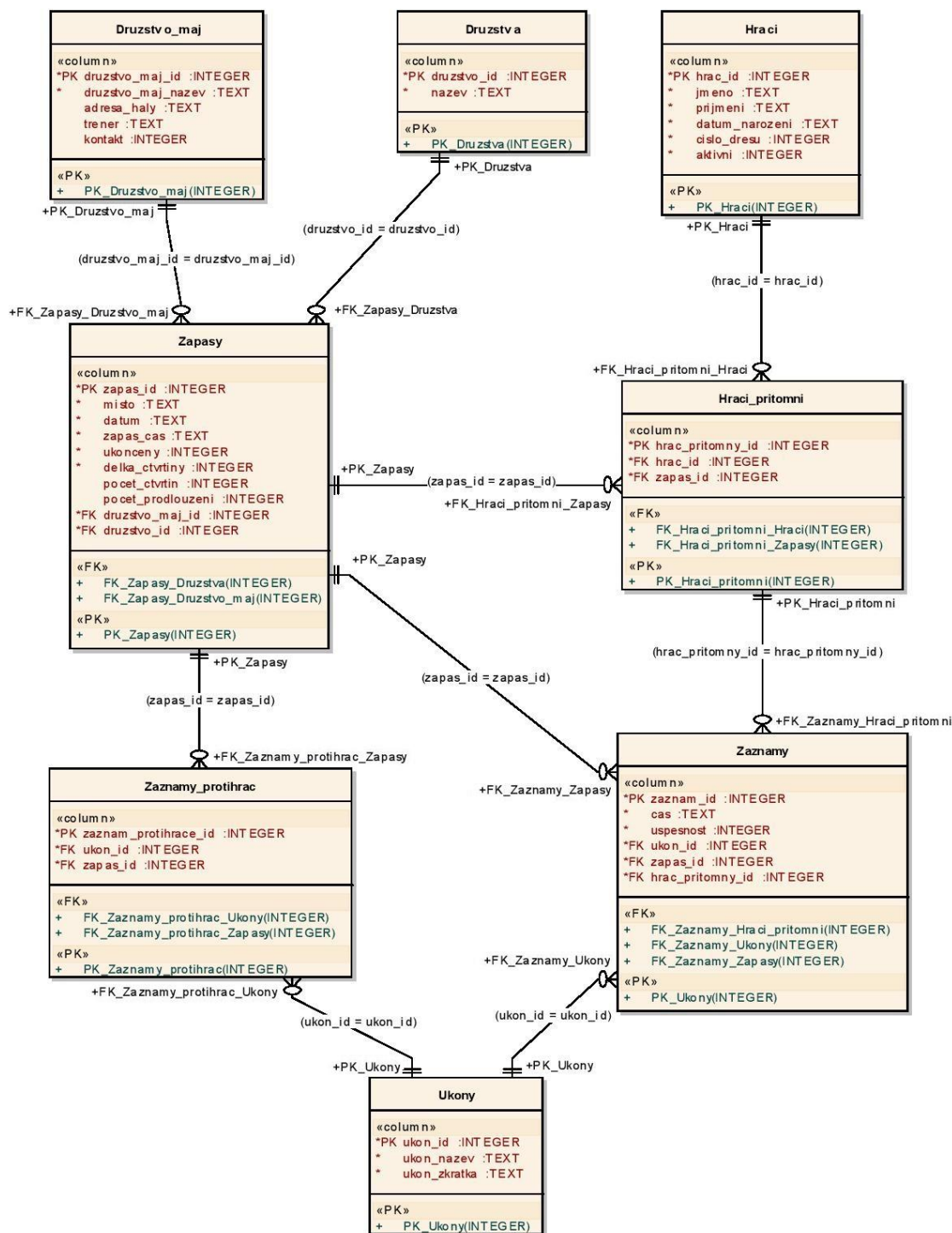
Aplikace musí být intuitivní a přehledná. Uživatel musí vždy vědět, kde se nachází a mít možnost vrátit se odkud přišel. Rozvržení je tedy potřeba navrhnout tak, aby bylo všechno dobře viditelné a aby měl uživatel dobrý přehled o tom, kde se co nachází. Toto je velmi důležité při sběru dat během zápasu, kdy je třeba se rychle orientovat v aplikaci.

3.2.8 Určení pro koho je aplikace

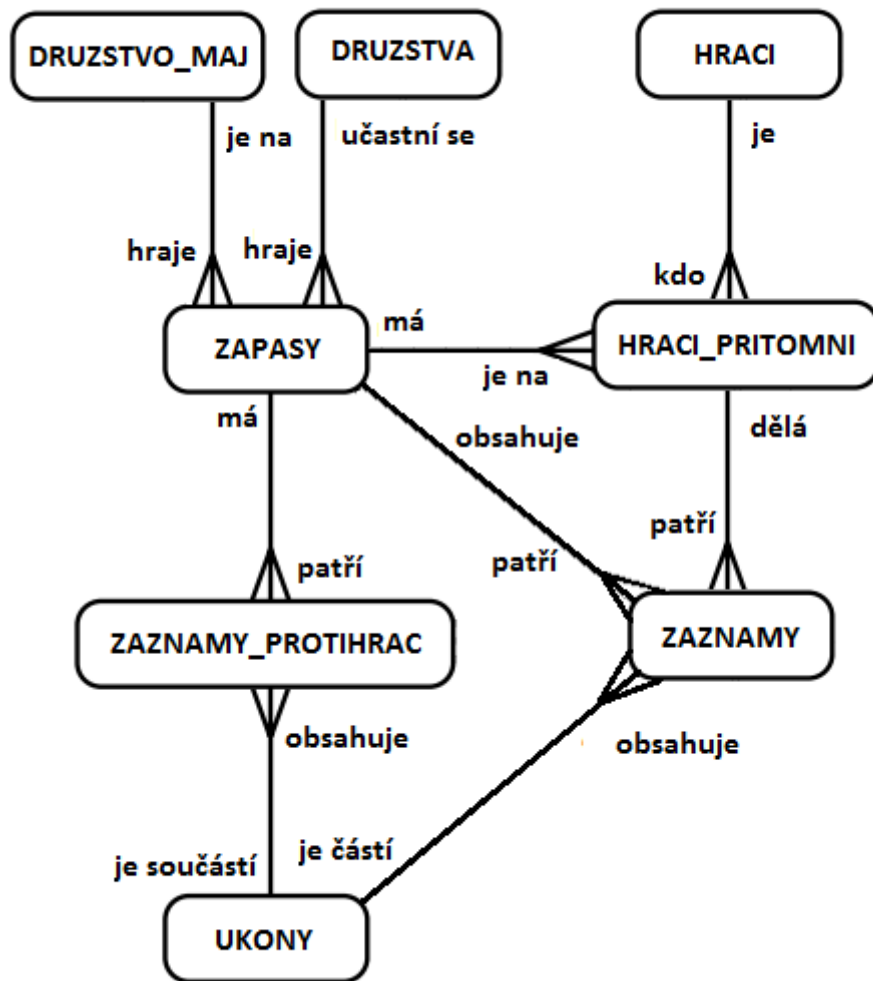
Aplikace bude sloužit ke sběru statistických údajů při basketbalovém utkání. Aplikace je určena ke sběru dat pouze pro jeden tým. Trenér či asistent zadává údaje o činnostech hráčů na hřišti do aplikace. Ke každému hráči je daná činnost přiřazena individuálně. Činnost hráčů soupeřova družstva pro konkrétní hráče není zaznamenávána, ale do aplikace je zaznamenáno pouze skóre soupeřova družstva.

4 Databázový model

Následující diagramy zobrazují strukturu databáze a vztahy mezi tabulkami.



Obrázek 1 - Databázový model



Obrázek 2 – Vazby tabulek v databázovém modelu

4.1 Přehled tabulek

Přehled tabulek uvádí atributy jednotlivých tabulek a popis k čemu tabulka slouží. V popise je vysvětlen i vztah jednotlivých tabulek.

4.1.1 Družstva

Název tabulky: druzstva

Atributy

- druzstvo_id INTEGER PRIMAY KEY AUTOINCREMENT
- druzstvo_nazev TEXT NOT NULL
- aktivni INTEGER NOT NULL

Tabulka družstva slouží k uchování informací o družstvech soupeřů. Tato tabulka je propojena s tabulkou `zapasy` přes atribut `druzstvo_id`, který je primárním klíčem tabulky s násobností 1:0..N.

Atribut `aktivni` slouží při výpisu družstev k určení, zda se má daný tým vypsat. Jestliže hodnota nastavena na 1, družstvo je vypisováno, jestliže hodnota nastavena na 0, družstvo ve výpise není zobrazeno.

4.1.2 Družstvo majitel

Název tabulky: `druzstvo`

Atributy:

- `druzstvo_maj_id` INTEGER PRIMAY KEY AUTOCREMENT
- `nazev` TEXT NOT NULL
- `adresa_haly` TEXT NOT NULL
- `trener` TEXT NOT NULL
- `kontakt` INTEGER NOT NULL

Tato tabulka slouží k uložení informací o družstvu uživatele aplikace. Do této tabulky aplikace neumožňuje ukládat data. Informace o družstvu lze pouze upravit.

Tabulka je propojena přes atribut `druzstvo_maj_id`, který je primárním klíčem tabulky, s tabulkou `zapasy` s násobností 1:0..N.

4.2 Zápasy

Název tabulky: `zapasy`

Atributy

- `zapas_id` PRIMARY KEY INTEGER AUTOINCREMENT
- `druzstvo_maj_id` INTEGER NOT NULL
- `druzstvo_id` INTEGER NOT NULL
- `misto` TEXT NOT NULL
- `datum` TEXT NOT NULL
- `zapas_cas` TEXT NOT NULL
- `delka_ctvrtiny` INTEGER NOT NULL
- `pocet_ctvrtin` INTEGER NOT NULL
- `pocet_prodlouzeni` INTEGER NOT NULL

- ukonceny INTEGER NOT NULL

Tabulka uchovává informace o vytvořených zápasech. V každém záznamu je uloženo identifikační číslo uživatele družstva, identifikační číslo družstva soupeře, zda se jedná o domácí či venkovní utkání, datum zápasu, čas začátku zápasu, délka čtvrtiny v sekundách, kolik čtvrtin bylo odehráno, kolik prodloužení bylo odehráno a zda byl zápas ukončen.

Jestliže je atribut `ukonceny` nastaven na hodnotu 0, zápas ukončený není a lze v něm pokračovat. Jestliže je nastaven na hodnotu 1, zápas je ukončený a nelze k němu zapisovat další informace.

Tabulka je propojena s tabulkami `druzstvo` a `druzstva` násobností 0.. N:1 primárními klíči těchto tabulek a dále s tabulkami `hraci_pritomny`, `zaznamy` a `ukony` násobnostmi 1:0..N atributem `zapas_id`, který je primárním klíčem této tabulky.

4.2.1 Hráči

Název tabulky: `hraci`

Atributy

- `hrac_id` PRIMARY KEY INTEGER AUTOINCREMENT
- `jmeno` TEXT NOT NULL
- `prijmeni` TEXT NOT NULL
- `datum_narozeni` TEXT
- `cislo_dresu` INTEGER NOT NULL
- `aktivni` INTEGER NOT NULL

Tabulka hráčů udržuje informace o jednotlivých hráčích. Atribut `aktivni` je využíván při výpisu hráčů. Jestliže hodnota tohoto atributu je nastavena na 0, hráč se ve výpisu nezobrazuje a tudíž ani nemůže být přidán do zápasu, je-li hodnota nastavena na 1 hráč je aktivní a lze s ním provádět veškeré operace.

Tabulka je propojena s tabulkou `hraci_pritomni` primárním klíčem `hrac_id` s násobností 1:0..N.

4.2.2 Hráči přítomní

Název tabulky: hraci

Atributy

- hrac_pritomny_id PRIMARY KEY INTEGER AUTOINCREMENT
- hrac_id INTEGER NOT NULL
- zapas_id INTEGER NOT NULL

Tabulka pritonmi_hraci udržuje informace o tom, kteří hráči se účastnili konkrétního zápasu. Tabulka je propojena s tabulkami hraci a zapasy primárními klíči těchto tabulek násobností 0..N:1. Dále je pak propojena s tabulkou zaznamy primárním klíčem hrac_pritomny_id násobností 1:0..N.

4.2.3 Záznamy

Název tabulky: zaznamy

Atributy

- zaznam_id PRIMARY KEY INTEGER AUTOINCREMENT
- hrac_pritomny_id INTEGER NOT NULL
- zapas_id INTEGER NOT NULL
- ukon_id INTEGER NOT NULL
- cas TEXT NOT NULL
- uspesnost INTEGER NOT NULL

Tabulka zaznamy slouží k udržení dat o jednotlivých úkonech hráče na hřišti při daném zápase. Do tabulky se ukládají všechny záznamy o činnostech na hřišti a střelbě spolu s úspěšností pokusu. Z těchto záznamů je později počítána statistika. Dále jsou zde uloženy i záznamy o střídání a lze z nich vypočítat kolik času hráč strávil na hřišti.

Tabulka zaznamy je propojena s tabulkami zapasy, hraci_pritomni a ukony primárními klíči těchto tabulek násobnostmi 0..N:1.

4.2.4 Záznamy protihráče

Název tabulky: zaznamy_protihrac

Atributy

- zaznam_protihrac_id PRIMARY KEY INTEGER AUTOINCREMENT
- ukon_id INTEGER NOT NULL

- `zapas_id` INTEGER NOT NULL

Tato tabulka slouží k ukládání soupeřových bodů při daném zápase. Jsou zde uloženy pouze úspěšné pokusy a body jsou přiřazeny celému družstvu, nikoli jednotlivému hráči.

Tabulka `zaznamy_protihrac` je propojena s tabulkou `zapasy` primárním klíčem `zapasy_id` s násobností 0..N:1.

4.2.5 Úkony

Název tabulky: `ukony`

Atributy

- `ukon_id` PRIMARY KEY INTEGER AUTOINCREMENT
- `ukon_nazev` TEXT NOT NULL
- `ukon_zkratka` TEXT NOT NULL

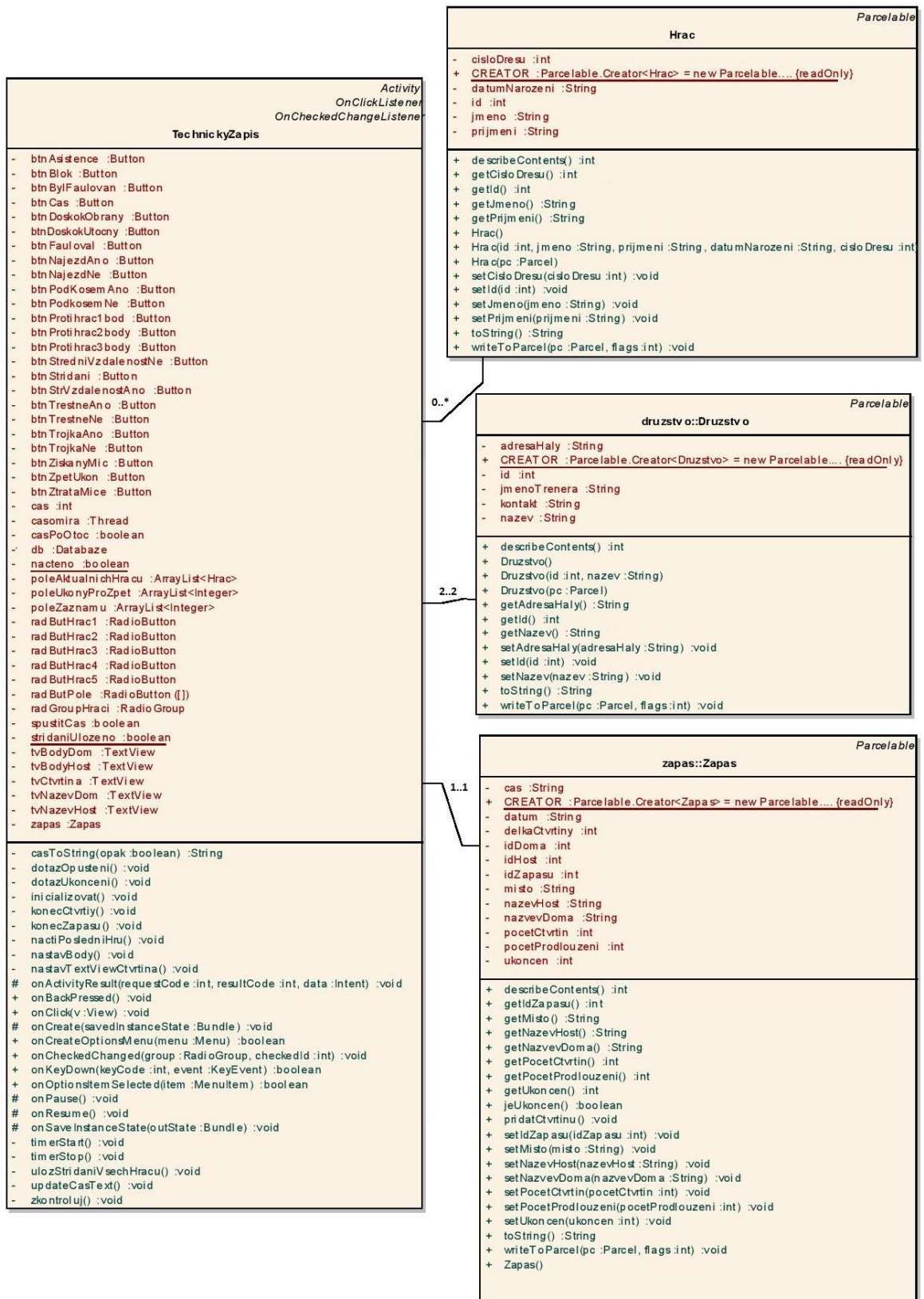
Tabulka `ukony` udržuje seznam všech úkonů, které může hráč na hřišti vykonat. Do této tabulky není možný uživatelský zápis. Tabulka je naplněna údaji při vytváření databáze.

Tabulka `ukony` je propojena s tabulkami `zaznamy_protihrac` a `hraci_pritomni` primárním klíčem `ukon_id` s násobností 1:0..N.

5 Realizace aplikace

Veškeré třídy, které slouží k získání a zobrazení informací a dat jsou potomkem třídy `Activity`, která slouží jako hlavní komunikační nástroj s uživatelem. Tyto třídy nemají konstruktor a jsou vytvářeny voláním metody `onCreate(Bundle)`, která je zděděna z rodičovské třídy. O zobrazení dat se starají soubory typu `xml`, které obsahují speciální značky pro nastavení a zobrazení prvků na obrazovce. Soubory `xml` pro zobrazení lze v aplikaci nalézt ve složce `res/layout`. V každé třídě v metodě `onCreate(Bundle)` je volána metoda `setContentView(...)`, v které je nastavena cesta k zobrazovacímu souboru `xml`. Instance třídy, která je potomkem třídy `Activity` dále bude nazývána aktivita.

Následující diagram slouží k zobrazení vztahů třídy `TechnickyZapis`, která je hlavní částí aplikace a slouží pro sběr dat, s třídami `Hrac`, `Zapas` a `Druzstvo`, které obsahují potřebné informace. Ve třídě `TechnickyZapis` je pole hráčů, které obsahuje instance třídy `Hrac` s informacemi o hráčích. V tomto poli jsou uloženi hráči, kteří jsou aktuálně ve hře. Instance třídy `Zapas` je ve třídě `TechnickyZapis` využita pro uložení aktuálních informací o zápase. Obsahuje například identifikační čísla družstev, počet odehraných čtvrtin, délku čtvrtiny a další. Instance třídy `Druzstvo` ve třídě `TechnickyZapis` slouží pro získání názvů družstev účastnících se zápasu.



Obrázek 3 - Ukázka vazeb třídy TechnickyZapis

5.1 Seznam balíčků a tříd

Třídy v aplikaci jsou rozděleny do balíčku podle jejich souvislosti, a to z lepší orientace v aplikaci. V následujících seznamech jsou zobrazeny nejprve názvy balíčků a pod nimi vždy třídy, které obsahuje.

- com.app.technickyZapis.druzstvo
 - Druzstva
 - Druzstvo
 - DruzstvoInfo
 - DruzstvoMajInfo
 - DruzstvoMajUpravit
 - DruzstvoNove
 - DruzstvoUpravit

- com.app.technickyZapis.hrac
 - Hrac
 - Hraci
 - HracInfo
 - HracNovy
 - HracStatistikaZapas
 - HracStatistikaZapasy
 - HracUpravit

- com.app.technickyZapis.knihovna
 - Databaze
 - Konstanta
 - Statistika

- com.app.technickyZapis.knihovna
 - Nastaveni
 - NastaveniHra

- com.app.technickyZapis.technickyZapis
 - OdebratHraceZeHry
 - PridatHraceDoHry
 - PridatHraceDoZapasu
 - Stridani
 - TechnickyZapis
 - Zaznam

- com.app.technickyZapis.zapas

- Zapas
- ZapasInfo
- ZapasNovy
- ZapasStatistika
- Zapasy

5.2 Popis tříd

5.2.1 Balíček `com.app.technickyZapis.druzstvo`

5.2.1.1 Třída *Druzstva*

Tato třída slouží k získání seznamu družstev z databáze. Jednotlivá družstva jsou pak převedena na řetězec, uložena do pole a předána jako parametr `ListView`, které slouží pro zobrazení rolovacího seznamu položek získaných z pole. V této třídě je implementována funkce vyhledávání, která umožňuje nalézt družstvo v seznamu. Funkce filtruje seznam během zadávání vyhledávacího řetězce a zobrazuje družstva, která mu odpovídají. Z třídy je možné spustit novou aktivitu s formulářem pro přidání nového družstva nebo aktivitu s detailem, kde jsou vypsané informace o zvoleném družstvu.

5.2.1.2 Třída *Druzstvo*

Třída `Druzstvo` uchovává informace o družstvu. Třída obsahuje atributy identifikační číslo družstva, název družstva, adresa haly, jméno trenéra a kontakt. Dále pak konstruktor bezparametrický i parametrický. V třídě jsou metody pro nastavení a získání hodnot atributů. Třída implementuje třídu `Parcelable`, která objekt převádí na pole atributů a tím umožňuje předávat objekt mezi aktivitami.

5.2.1.3 Třída *DruzstvoInfo*

Třída `DruzstvoInfo` slouží k zobrazení informací o daném družstvu. Při spouštění aktivity získá objekt typu `Druzstvo`, z kterého získá informace o družstvu a vkládá je do textového pole typu `TextView`. Dále třída obsahuje metody ke spuštění nové aktivity s úpravou informací, které předá instanci třídy `Druzstvo` a metodu, která slouží k odstranění družstva. V této metodě je volána metoda `upravDruzstvo(...)` z třídy `Databaze`.

5.2.1.4 Třída *DruzstvoMajInfo*

Třída získává informace o družstvu majitele aplikace z databáze pomocí metody `getDruzstvoaj(...)` z třídy `Databaze` a nastavuje je do textových polí typu `TextView`. Třída obsahuje metodu, která spustí novou aktivitu s formulářem pro editaci údajů a předá jí instanci třídy `Druzstvo` s informacemi o družstvu uživatele aplikace.

5.2.1.5 Třída *DruzstvoMajUpravit*

Třída `DruzstvoMajUpravit` slouží k úpravě údajů o družstvu uživatele aplikace. Jsou zde čtyři editovací textová pole typu `EditText` pro název družstva, adresu haly, jméno trenéra a kontakt. Do těchto polí jsou při spuštění aktivity nastaveny aktuální údaje získané z databáze. Při odesílání dat je volána metoda `upravDruzstvoMaj(...)` z třídy `Databaze` pro aktualizaci údajů družstva.

5.2.1.6 Třída *DruzstvoNove*

`DruzstvoNove` slouží k vytvoření nového družstva. Je zde editovací textové pole typu `EditText` pro název družstva. Při odesílání dat je kontrolováno, zda je pole vyplněno a jestliže je vše v pořádku je volána metoda `vlozDoDruzstva(...)` z třídy `Databaze` pro uložení dat do databáze a aktivita ukončena. V opačném případě je zobrazeno dialogové okno s upozorněním nutnosti vyplnit údaj.

5.2.1.7 Třída *DruzstvoUpravit*

`DruzstvoUpravit` slouží k úpravě informací o družstvu soupeře. Při spuštění aktivity je získána instance třídy `Druzstvo` s informacemi o družstvu a z objektu je získán název družstva a je nastaven do editovacího textového pole typu `EditText`. Při odesílání informací je volána metoda `upravDruzstvo(...)` z třídy `Databaze` pro úpravu údajů družstva v tabulce `druzstva`.

5.2.2 Balíček `com.app.technickyZapis.hrac`

5.2.2.1 Třída *Hrac*

Třída `Hrac` zaobaluje informace o hráči. Třída obsahuje atributy identifikační číslo, jméno, příjmení, číslo dresu a datum narození. Jsou zde metody pro nastavení a získání hodnot jednotlivých atributů. Instanci třídy je možné převést na řetězec pomocí metody

`toString()`. Třída implementuje třídu `Parcelable`, která převádí objekt na pole atributů a umožňuje tak předávání objektu mezi aktivitami.

5.2.2.2 Třída *Hraci*

Třída `Hraci` slouží k zobrazení seznamu hráčů. Třída získává pole hráčů z databáze pomocí metody `getHraciPole()` z třídy `Databaze`. Pole hráčů je převedeno na pole řetězců, které je předáno jako parametr při vytváření objektu `ListView`. `ListView` slouží k zobrazení pole řetězců v podobě seznamu. Třída obsahuje metodu pro vyhledávání v seznamu hráčů, metodu pro spuštění aktivity s informacemi o hráči po zvolení hráče ze seznamu, které předá objekt typu `Hrac` s informacemi o hráči a metodu pro spuštění aktivity pro přidání nového hráče do databáze.

5.2.2.3 Třída *HracInfo*

Třída `HracInfo` slouží k získání a zobrazení informací o hráči. Při spuštění aplikace je získána instance třídy `Hrac` s informacemi o hráči a informace jsou nastaveny do textových polí typu `TextView`. Třída obsahuje metodu pro spuštění aktivity pro úpravu hráče a metodu pro skrytí hráče v seznamu hráčů pomocí metody `upravHrace(...)` z třídy `Databaze`.

5.2.2.4 Třída *HracNovy*

Tato třída slouží k vytvoření nového hráče. Jsou zde čtyři textová editovací pole pro jméno, příjmení, datum narození a číslo dresu typu `EditText`. Při odeslání dat je volána metoda `vlozDoHracu(...)` z třídy `Databaze`, která hráče uloží do tabulky `hraci`.

Pro pole datum narození jsou zde metody, které po kliknutí na pole zobrazí dialogové okno, kde je možné vybrat datum, místo softwarové klávesnice. Datum je po výběru nastaveno do editovacího textového pole.

5.2.2.5 Třída *HracStatistikaZapas*

Třída `HracStatistikaZapas` slouží k zobrazení statistických údajů hráče pro daný zápas. Při spuštění aktivity je získána instance třídy `Zapas` a `Hrac`. Pomocí metody `getZaznamyHraceZeZapasuPole(...)` z třídy `Databaze` je získáno pole záznamů hráče z daného zápasu. Toto pole je pak předáno v konstruktoru jako atribut při vytváření objektu `Statistika`, kde jsou vypočítány všechny statistické údaje. Jednotlivé údaje

jsou pak získány pomocí metod třídy `Statistika`, které jsou uloženy nastaveny do textových polí a zobrazeny.

5.2.2.6 Třída `HracStatistikaZapasy`

Třída `HracStatistikaZapasy` slouží k zobrazení odehraných zápasů hráče. Seznam zápasů je získán z databáze pomocí metody `getZapasyHracePole(...)` z třídy `Databaze`, která vrátí pole zápasů, které hráč odehrál. Pole zápasů je převedeno na pole řetězců a zobrazeno pomocí objektu `ListView`, kterému je předáno jako atribut při vytváření objektu. Třída obsahuje metodu pro vyhledávání v poli, která filtruje seznam zápasů během zadávání vyhledávacího řetězce. Vyhledávat lze podle názvu soupeře nebo podle data zápasu.

5.2.2.7 Třída `HracUpravit`

Třída slouží k úpravě údajů hráče. Při spuštění je získána instance třídy `Hrac` s nastavenými informacemi o hráči. Tyto údaje jsou nastaveny do jednotlivých editovacích textových polí typu `EditText`. Při odesílání upravených informací je volána metoda `upravHrace(...)` z třídy `Databaze` pro aktualizaci dat v tabulce hráčů v databázi.

5.2.3 Balíček `com.app.technickyZapis.knihovna`

5.2.3.1 Třída `Databaze`

Třída `Databaze` je hlavní třídou pro práci s databází. Při instalaci aplikace jsou vytvořeny tabulky a naplněny daty. V třídě jsou implementovány metody pro vkládání dat do databáze, úpravu dat v databázi, získávání dat z databáze a mazání dat z databáze.

Následující kód vytváří v databázi tabulku `hraci`.

```
db.execSQL("CREATE TABLE " + DATABASE_HRACI + " (" +
KLIC_HRAC_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
KLIC_JMENO_HRACE + " TEXT NOT NULL, " +
KLIC_PRIJMENI_HRACE + " TEXT NOT NULL, " +
KLIC_DATUM_NAROZENI + " TEXT NOT NULL, " +
KLIC_CISLO_DRESU + " INTEGER NOT NULL, " +
KLIC_AKTIVNI + " INTEGER NOT NULL);");
```

Jednotlivé klíče jsou konstanty třídy, které jsou využívány ve všech metodách této třídy.

```

private static final String DATABASE_HRACI = "hraci";
private static final String KLIC_HRAC_ID = "hrac_id";
private static final String KLIC_JMENO_HRACE = "jmeno";
private static final String KLIC_PRIJMENI_HRACE = "prijmeni";
private static final String KLIC_DATUM_NAROZENI = "datum_narozeni";
private static final String KLIC_CISLO_DRESU = "cislo_dresu";
private static final String KLIC_AKTIVNI = "aktivni";

```

Další metoda ukazuje způsob vkládání záznamu do tabulky hraci. Je zde využita třída ContentValues, která slouží k uložení sady hodnot. Instance této třídy je předána jako parametr databázové metody insert(...), ze systémové třídy SQLiteOpenHelper.

```

public long vlozDoHracu(String jmeno, String prijmeni,
    String datumNarozeni, int cisloDresu) {
    ContentValues values = new ContentValues();
    values.put(KLIC_JMENO_HRACE, jmeno);
    values.put(KLIC_PRIJMENI_HRACE, prijmeni);
    values.put(KLIC_DATUM_NAROZENI, datumNarozeni);
    values.put(KLIC_CISLO_DRESU, cisloDresu);
    values.put(KLIC_AKTIVNI, 1);
    return ourDatabase.insert(DATABASE_HRACI, null, values);
}

```

5.2.3.2 Třída Konstanta

Třída Konstanta obsahuje konstanty využívané v aplikaci. Konstanty jsou využívány v několika třídách aplikace, a proto je zde pro ně samostatná třída. Třída obsahuje modifikátor final, který určuje, že členové třídy jsou neměnitelní. V třídě jsou pouze statické konstanty, které je možné volat pomocí tečkové notace bez vytvoření třídy Konstanta.

Příklad volání konstanty AKTIVNI

```
Konstanta.AKTIVNI
```

Ukázka třídy Konstanta

```

public final class Konstanta {
    public static final int AKTIVNI = 1;
    ...
}

```

5.2.3.3 Třída Statistika

Třída Statistika obsahuje parametrický konstruktor. Parametrem je pole záznamů hráče nebo družstva, které lze získat z databáze. Při vytváření objektu, je v cyklu projito

polem a je získáno identifikační číslo úkonu a úspěšnost úkonu. Dále je využit přepínač switch, který údaje dělí podle identifikačního čísla a ukládá je do příslušných atributů. Střelecké záznamy jsou dále děleny na úspěšné a neúspěšné. Třída dále obsahuje metody pro získání jednotlivých údajů a metody pro výpočet statistických hodnot.

Následující metoda ukazuje příklad výpočtu procenta úspěšnosti střelby trestného hodu. Pole `trestne[Konstanta.PROMENENE]` obsahuje počet proměněných trestných hodů a `trestne[Konstanta.NEPROMENENE]` počet trestných hodů neproměněných. Metoda `Math.round(...)` hodnotu po výpočtu zaokrouhluje. Výsledná hodnota je následně vrácena jako návratová hodnota metody.

```
public double vratTrestneHody() {
    double tr = Math.round(trestne[Konstanta.PROMENENE] /
        (trestne[Konstanta.NEPROMENENE] + trestne[Konstanta.PROMENENE]) *
        100);
    return tr;
}
```

5.2.4 Balíček `com.app.technickyZapis.nastaveni`

5.2.4.1 Třída `Nastaveni`

Třída `Nastaveni` slouží k zobrazení a výběru možností nastavení. Možnosti jsou zobrazeny v seznamu typu `ListView`. V třídě jsou metody pro spuštění aktivit `DruzstvoMajInfo` a `NastaveniHry`.

5.2.4.2 Třída `NastaveniHra`

Třída `NastaveniHra` obsahuje dvě rolovací nabídky. První nabídka je pro nastavení délky čtvrtiny a druhá nabídka slouží pro nastavení maximálního počtu hráčů v jednom zápase. Po nastavení údajů a stisknutí tlačítka „Uložit“ jsou údaje uloženy do binárních souborů. Tyto hodnoty jsou využity v dalších aktivitách.

5.2.5 Balíček `com.app.technickyZapis.technickyZapis`

5.2.5.1 Třída `OdebratHraceZeHry`

Třída `OdebratHraceZeHry` slouží k odebrání hráče, který je momentálně ve hře. Při vytváření instance třídy, je získáno pole hráčů, kteří jsou aktuálně na hřišti. Dále je volána metoda `pridejCheckboxy()`, která pro každého hráče vytvoří vlastní zaškrťovací políčko typu `CheckBox`, ke kterému je hráč přiřazen. Po výběru hráčů, které

si uživatel přeje odebrat ze hry, je volána metoda, která aktualizuje pole hráčů, odebraným hráčům přidá záznam o střídání do databáze pomocí metody `vlozDoZaznamy(...)` z třídy `Databaze` a aktualizované pole vrátí aktivitě, která tuto třídu volala.

5.2.5.2 Třída `PridatHraceDoHry`

Třída `PridatHraceDoHry` slouží k přidání hráčů do hry. Při vytváření instance třídy je získáno pole hráčů, kteří jsou aktuálně na hřišti a pole hráčů, kteří jsou přítomni na zápasu. Ke každému hráči z pole přítomných hráčů je vytvořeno zaškrťovací pole typu `CheckBox`, ke kterému je hráč přiřazen. Během výběru je kontrolováno, aby počet vybraných hráčů nepřesáhl maximální počet povolených hráčů na hřišti. Po výběru je pole aktuálních hráčů doplněno o nové hráče a předáno aktivitě, která tuto aktivitu spustila.

5.2.5.3 Třída `PridatHraceDoZapasu`

Třída `PridatHraceDoZapasu`, slouží k přidání hráčů z databázové tabulky `hraci` do tabulky `hraci_pritomni`. Při spuštění aktivity je z databázové tabulky `hraci` získáno pole hráčů. Ke každému hráči z pole je vytvořeno zaškrťovací políčko typu `CheckBox`. Po zvolení hráčů je volána metoda `vlozDoPritomnychHracu(...)` z třídy `Databaze`, která vloží záznamy o hráčích do tabulky `hraci_pritomni`.

5.2.5.4 Třída `Stridani`

Třída `Stridani` slouží k vystřídání jednoho či více hráčů, kteří jsou aktuálně na hřišti za hráče, kteří jsou uloženi v databázové tabulce `hraci_pritomni` pro daný zápas. V metodě `onCreate()` je získáno pole aktuálních hráčů získané z předešlé aktivity a pole přítomných hráčů na zápasu z databázové tabulky `hraci_pritomni` pomocí metody `getHraciPritomniPol(...)`. Z pole přítomných hráčů jsou odstraněny hráči, kteří jsou již na hřišti a pro hráče v obou polích jsou vytvořeny zaškrťovací okna typu `CheckBox`, ke kterým jsou jednotliví hráči přiřazeni. Obě skupiny zaškrťovacích polí jsou od sebe viditelně odděleny. Při výběru hráčů je kontrolováno pomocí metody `getCheckBoxOnClick(...)` aby byl vybrán stejný počet hráčů, kteří půjdou ze hry s počtem hráčů, kteří půjdou na hřiště. Po provedení výběru je vytvořeno nové pole z hráčů, kteří zůstávají ve hře a hráčů, kteří jsou do hry přidáni a toto pole vrátí aktivitě, která tuto třídu spustila. Střídající hráči jsou zaznamenáni do databázové tabulky `zaznamy`, pro výpočet celkového času stráveného ve hře.

5.2.5.5 Třída *TechnickyZapis*

Třída `TechnickyZapis` je hlavní částí aplikace a slouží ke sběru dat. V metodě `onCreate()` je získán objekt typu `Zapas` s informacemi o zápase, které jsou dále využívány a upravovány.

Třída obsahuje pět přepínacích polí typu `RadioButton`, ke kterým jsou přiřazeni hráči, kteří jsou aktuálně ve hře. Tyto hráče je možné střídat. Při stisknutí tlačítka „Střídat“ je spuštěna aktivita `Stridani`, které je předáno pole aktuálních hráčů a po ukončení této aktivity je vráceno nové pole. Každý hráč z pole je následně přiřazen k jednomu přepínacímu poli pomocí metody `zkontroluj()`, která nejprve zkontroluje počet hráčů v poli a podle toho nastaví počet povolených polí, přičemž ostatní pole zakáže. Povoleným polím pak přiřadí hráče.

Třída obsahuje množství tlačítek pro přidávání úkonu hráči a jejich akce jsou odchytávány metodou `onClick(...)`, kde podle zvoleného tlačítka je volána metoda `vlozDoZaznamy(...)` z třídy `Databaze`, které je předáno identifikační číslo hráče, identifikační číslo úkonu, úspěšnost úkonu, identifikační číslo zápasu a čas, ve kterém byl úkon vykonán a tyto údaje jsou následně uloženy do databázové tabulky `zaznamy`. S uložením záznamu je zároveň záznam uložen do pole, které umožňuje vzít úkon zpět při stisknutí tlačítka „Zpět“.

V metodě `onClick(...)` jsou dále volány metody `timerStart()` a `timerStop()`, které spouštějí a zastavují odpočítávání času při stisknutí tlačítka, které zobrazuje aktuální čas.

Při stisknutí tlačítka menu jsou zobrazeny možnosti: Statistika, Přidat hráče do zápasu, Přidat hráče do hry, Odebrat hráče ze hry a Konec. Tyto volby jsou obsluhovány metodou `onOptionsItemSelected(...)` a v závislosti na volbě je spuštěna aktivita.

Třída dále obsahuje metodu `nactiPosledniHru()`. Tato metoda slouží k načtení hry, která nebyla ukončena a umožňuje pokračovat v zápase přesně od místa, kde byl zápas přerušen. Je zde získán a nastaven čas přerušení, čtvrtina, hráči, kteří byli v době přerušení na hřišti a bodový stav obou družstev.

Metoda `konecCtvrtiny()` kontroluje počet odehraných čtvrtin a v případě, že jich je méně než čtyři, zvýší počet odehraných čtvrtin, uloží tento stav do databáze a nastaví čas na začátek další čtvrtiny. Jestliže jsou odehrány všechny čtvrtiny je volána metoda `dotazUkonceni(...)`, která zobrazí dialogové okno s dotazem na ukončení zápasu. Při souhlasu je do databáze uložena informace o ukončení zápasu.

5.2.5.6 Třída Zaznam

Třída `Zaznam` obaluje informace o jednotlivém záznamu. Obsahuje atributy `idZaznamu`, `idZapasu`, `idHrace`, `idUkonu`, `uspesnost` typu `Integer` a `cas` typu `String`. Tato třída slouží pro uložení informací o jednotlivém záznamu, při výpočtu statistických údajů.

5.2.6 Balíček `com.app.technickyZapis.zapas`

5.2.6.1 Třída Zapas

Třída `Zapas` obaluje informace o zápase. Třída implementuje třídu `Parcelable`, což umožňuje objekt převést na pole atributů a zpět na objekt. Toto slouží k předávání informací o zápase mezi aktivitami.

5.2.6.2 Třída ZapasInfo

Třída `ZapasInfo` slouží k zobrazení informací o daném zápase. Jsou zde textová pole pro zobrazení soupeře, data zápasu a bodového výsledku typu `TextView`. Při spuštění třídy je získán objekt typu `Zapas` s informacemi o daném zápase. Z objektu jsou zjištěny potřebné informace a nastaveny do textových polí. Dále je zde seznam hráčů, kteří se účastnili zápasu. Seznam hráčů je získán z databáze na základě identifikačního čísla zápasu metodou `getHraciPritomniPole(...)` z třídy `Databaze`. Po zvolení hráče je spuštěna aktivita se statistikou hráče v tomto zápase. Třída obsahuje metodu pro odstranění zápasu, která volá metodu `smazZapas(...)` z třídy `Databaze`, která smaže údaje v jednotlivých tabulkách, kde se identifikační číslo zápasu vyskytuje. Třída dále umožňuje spustit aktivitu `TechnickyZapis` v případě, že je zápas neukončený a aktivitu `ZapasStatistika` k zobrazení statistických údajů pro celé družstvo spolu se statistikami pro jednotlivé hráče.

5.2.6.3 Třída ZapasNovy

Tato třída slouží k získání údajů o zápase. Metodou `nactiCas(...)` je získána délka čtvrtiny a nastavena do instance třídy `Zapas`. V třídě jsou dále dvě rozbalovací nabídky typu `Spinner` pro vybrání soupeře a místa konání utkání. Soupeři jsou načteni metodou `vlozDoSpinnerDruzstva()`, ve které je volána metoda pro získání družstev z databáze. Dále jsou zde dvě textová editovací pole typu `EditText`, které slouží k zadání data a času. Na textová editovací pole jsou nastaveny metody, které místo zobrazení softwarové klávesnice zobrazí dialogová okna s výběrem času a místa a po zvolení tyto hodnoty nastaví do příslušných polí. Získané informace jsou nastaveny do objektu typu `Zapas` a při spouštění následující aktivity je objekt této aktivity předán.

5.2.6.4 Třída ZapasStatistika

Třída `ZapasStatistika` slouží k zobrazení vypočítaných statistických údajů, jak pro celé družstvo, tak i pro jednotlivé hráče, na jedné obrazovce. V metodě `onCreate(...)` je nejprve získán objekt typu `Zapas` ze spouštěcí aktivity. Dále jsou získány záznamy ze zápasu pomocí metody `getZaznamyZeZapasuPole(...)` z třídy `Databaze`, která získá veškeré záznamy ze zápasu podle identifikačního čísla zápasu. Získané záznamy jsou předány jako parametr v konstruktoru třídy `Statistika`, která záznamy roztrídí. Instance třídy `Statistika` je pak předána metodě `vratTabulku(...)` jako parametr spolu s instancí třídy `Zapas` a poté je dynamicky vytvořena tabulka s celkovou statistikou družstva pro daný zápas. Dále je z databáze získán seznam přítomných hráčů. Pro každého hráče je v cyklu vždy získáno pole úkonů. Pole úkonů je opět předáno třídě `Statistika` a poté je instance této třídy předána metodě `vratTabulku(...)` spolu s instancí třídy `Hrac`.

5.2.6.5 Třída Zapasy

Třída `Zapasy` slouží k zobrazení seznamu vytvořených zápasů. Pole zápasů typu `Zapas` je získáno z databáze metodou `getZapasyPole()`. Toto pole je převedeno na pole řetězců pomocí metody `vratPoleZapasu()`. Toto pole je předáno jako parametr `ListView`, které slouží k zobrazení pole řetězců v podobě rolovacího seznamu. Třída obsahuje metodu pro vyhledávání v seznamu zápasů. Třída dále obsahuje metodu, která po zvolení zápasu spustí novou aktivitu s informacemi o tomto zápase.

6 Prezentace dat na internetu

Možností jak prezentovat data na internetu je více. Jedna z nich je exportování výsledných dat například do formátu pdf a následné vložení na webovou stránku. K tomu by bylo nutné v aplikaci naprogramovat metodu, která by výsledná data dokázala uložit do tohoto formátu, vložení souboru na webové stránky by bylo možné nahráním souboru na server a přidáním odkazu na tento soubor do webových stránek.

Další možností by mohlo být vytvoření databáze, která by byla stejná jako databáze v aplikaci, do které by byla data nahrávána buď už během zápasu, kdyby měl uživatel možnost připojení k internetu, nebo popřípadě až po skončení zápasu a možností internetového připojení. K této možnosti by bylo zapotřebí vytvořit již zmíněnou databázi a naprogramovat webovou aplikaci, která by tato data zpracovávala a zobrazovala. Dále pak do mobilní aplikace přidat funkci, která by s webovou aplikací komunikovala a data na ni odesílala. Odesílání dat by mohlo nastat při vykonání jakéhokoli zadání úkonu do aplikace nebo odesílat skupinu dat po určitém časovém intervalu, v případě, že by uživatel měl dostupné internetové připojení, nebo najednou, až by připojení k internetu bylo dostupné. Toto řešení by umožňovalo s daty provádět stejné operace jako s daty v mobilní aplikaci a vytvořit nové operace s daty. Další výhodou u tohoto řešení by byla možnost výsledná data upravit v případě zjištění nesrovnalostí s technickým zápisem z utkání. V případě možnosti data odesílat během utkání by výhodou byla možnost zobrazit aktuální statistiky hráčů i družstva.

Závěr

Hlavním cílem této bakalářské práce bylo vytvoření aplikace pro sběr statistických údajů při basketbalovém utkání. Aplikace byla navržena tak, aby ovládání bylo intuitivní a uživatel nemusel dlouho hledat požadovanou možnost. Sebraná data jsou uložena v databázi, která byla navržena tak, aby práce s daty byla co nejjednodušší.

Testování aplikace proběhlo zatím jen při simulovaném zápase, konkrétně ze záznamu ze zápasu. Při tomto testování nebyly odhaleny žádné nedostatky, které by byly nepřípustné pro použití v praxi a znemožňovaly uživateli zadávat data v reálném čase. Běh aplikace nebyl omezován ani při zadávání velkého počtu dat a výpočet statistik se jeví rychlý i při velkém počtu záznamů.

Aplikace by do budoucna mohla být rozšířena o již zmíněnou webovou aplikaci s možností prezentace dat na internetu, která by sloužila pro zpřístupnění dat lidem, kteří mají zájem tato data zkoumat. Dalším rozšířením by mohlo být přidání aktivity, která by umožňovala kompletní záznam úkonů ke konkrétním hráčům v obou družstvech. Jako další rozšíření se jeví možnost přidání dalších výpočtů statistických dat a možnost jejich zobrazení.

Použitá literatura

- [1] KOMATINENI, Satya a Dave MACLEAN. *Pro Android 4*. New York: Springer Science+Business Media New York, 2012. ISBN 978-1-4302-3930-7.
- [2] GRANT, Allen. *Beginning Android 4*. New York: Springer Science+Business Media New York, 2012. ISBN 978-1-4302-3984-0.
- [3] TIŠNOVSKÝ, Pavel. Eclipse: integrované vývojové prostředí pro Javu i další programovací jazyky. In: *Fedora* [online]. 2012 [cit. 2014-05-02]. Dostupné z: <http://fedora.cz/eclipse-integrované-vývojové-prostředí-pro-javu-i-další-programovací-jazyky/>
- [4] HAVRYLUK, Michal. Naučíme vás programovat aplikace pro android.: Začínáme. In: *Mobil.idnes.cz* [online]. 2012 [cit. 2014-05-02]. Dostupné z: http://mobil.idnes.cz/naucime-vas-programovat-aplikace-pro-android-zaciname-prave-dnes-phe-/aplikace.aspx?c=A120410_125436_aplikace_ham
- [5] SEMECKÝ, Jiří. Naučte se Javu: Úvod. *Interval.cz* [online]. 2002 [cit. 2014-05-02]. Dostupné z: <http://interval.cz/clanky/naucte-se-javu-uvod/>
- [6] GOLDBERG, Kevin Howard. *XML: Visual QuickStart Guide*. 2. vyd. Berkeley: Peachpit Press, 2008. ISBN 978-0321559678.
- [7] LACKO, Luboslav. *SQL kapesní přehled*. Brno: Computer press a.s., 2005. ISBN 80-251-0788-4.
- [8] MEIER, Reto. *Professional Android 4 Application Development*. John Wiley & Sons, Inc.: Indianapolis, 2012. ISBN 978-111810227-5.
- [9] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. Vyd. 1. Překlad Bogdan Kiszka. Brno: Computer Press, 2007, 567 s. ISBN 978-80-251-1503-9.