

# **Framework pro vývoj aplikací pro děti s postižením pro platformu Android**

## **Framework for Application Development for Children with Disabilities for the Android Platform**



## Zadání diplomové práce

Student: **Bc. David Martiník**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: **Framework pro vývoj aplikací pro děti s postižením pro platformu Android**  
**Framework for Application Development for Children with Disabilities for the Android Platform**

### Zásady pro vypracování:

Výstupy této diplomové práce budou součástí systému EduKin. Cílem práce je vytvořit programový rámec neboli framework pro platformu Android, který usnadní vývoj aplikací pro děti vyžadující zvýšenou péči rodičů a učitelů. Tyto aplikace by jim měly umožnit rozvíjet jejich jemnou motoriku, paměť či vyjadřování. Tento programový rámec tedy musí obsahovat API ulehčující rozpoznávání a práci s gesty a periferiemi, sdílení a synchronizaci s cloudovým řešením EduKin a vhodné komponenty uživatelského rozhraní.

### Jednotlivé body zadání:

1. Popis operačního systému Android se zaměřením na práci s gesty.
2. Navrhnout architekturu frameworku pro usnadnění vývoje aplikací pro hendikepované.
3. Implementace frameworku a testovací aplikace.
4. Propojení vytvořených aplikací pro operační systém Android s Windows Azure.
5. Otestovat a nasadit aplikace do prostředí Google Play.

### Seznam doporučené odborné literatury:

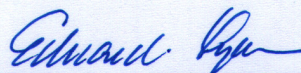
[1] Web pro vývojáře pro operační systém Android: <http://developer.android.com/index.html>

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

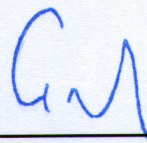
Vedoucí diplomové práce: **Ing. Jan Martinovič, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka  
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.  
děkan fakulty



Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2013

*Karlík*  
.....

Rád bych na tomto místě poděkoval všem lidem, kteří nám pomohli s čímkoli, co se týkalo projektu EduKin, protože bez nich by tato práce nevznikla. Především děkuji vedoucímu práce, panu Ing. Janu Martinovičovi Ph.D., který téma práce vymyslel.



## **Abstrakt**

Cílem této práce bylo vytvořit programový rámec neboli framework pro mobilní platformu Android, který bude součástí systému EduKin. Ten si bere za cíl vytvořit jednoduchou platformu pro výuku a rozvoj dětí se speciálními potřebami. Jejich rodičům a učitelům by měl sloužit jako nástroj pro zadávání úkolů a sledování pokroků. Samotný framework si klade za cíl ulehčit vývoj aplikací pro systém EduKin na platformu Android. Jedná se tedy hlavně o komunikaci s cloudovou službou provozovanou na Windows Azure a API usnadňující práci s gesty a periferiemi daných zařízení. Dále je v práci popsáno vytvoření aplikací nad tímto frameworkem, které byly konzultovány přímo s učiteli a testovány s jejich žáky.

**Klíčová slova:** EduKin, Android, cloud, Windows Azure, programový rámec, WCF, JSON

## **Abstract**

Goal of this work was to create framework for Android platform, which will be part of system EduKin. This project is meant to create simple, but powerful platform for education and exercise of disabled children. It should serve as a tool for assigning tasks and monitoring progress. Framework itself should ease development of applications for EduKin system for Android platform. Communication with cloud service hosted on Windows Azure and API for easier work with gestures and peripherals are main goals. There is also described the creation of applications using this framework, which were consulted with teachers and tested by thiers students.

**Keywords:** EduKin, Android, cloud, Windows Azure, framework, WCF, JSON

## Seznam použitých zkratk a symbolů

WCF	– Windows Communication Foundation
JSON	– JavaScript Object Notation
SDK	– Software Development Kit
JNI	– Java Native Interface
ADT	– Android Developer Tools
WSDL	– Web Services Description Language
SOAP	– Simple Object Access Protocol
GUI	– Graphical User Interface

## Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>EduKin</b>	<b>6</b>
2.1	Vznik . . . . .	6
2.2	Prvotní představa . . . . .	7
<b>3</b>	<b>Cílová platforma</b>	<b>9</b>
3.1	Výběr cílové platformy . . . . .	9
3.2	Popis vybrané platformy . . . . .	10
<b>4</b>	<b>Analýza</b>	<b>18</b>
4.1	Specifikace požadavků . . . . .	18
4.2	Funkční specifikace . . . . .	20
4.3	Procesy . . . . .	22
4.4	Případy užití . . . . .	23
4.5	Lokální databáze . . . . .	24
4.6	Součásti frameworku . . . . .	24
<b>5</b>	<b>Návrh frameworku</b>	<b>30</b>
5.1	Komunikace se službami systému EduKin . . . . .	30
5.2	Vykreslování herní grafiky . . . . .	31
5.3	Preferovaný formát dat . . . . .	33
5.4	Rozvržení knihoven . . . . .	33
<b>6</b>	<b>Návrh grafického uživatelského rozhraní</b>	<b>36</b>
6.1	Směrnice návrhu uživatelského rozhraní pro systém EduKin . . . . .	36
6.2	Ukázky návrhů jednotlivých obrazovek . . . . .	37
<b>7</b>	<b>Implementace a testování</b>	<b>39</b>
7.1	Ukázky kódů . . . . .	39
7.2	Problémy při implementaci . . . . .	39
7.3	Aplikace Kreslení . . . . .	41
7.4	Aplikace Asociace . . . . .	41
7.5	Testování . . . . .	42
7.6	Konzultace . . . . .	44
<b>8</b>	<b>Závěr</b>	<b>45</b>
<b>9</b>	<b>Reference</b>	<b>46</b>
	<b>Přílohy</b>	<b>46</b>
<b>A</b>	<b>Obsah přiloženého DVD</b>	<b>47</b>

## Seznam tabulek

- |   |                                                                                                                       |    |
|---|-----------------------------------------------------------------------------------------------------------------------|----|
| 1 | Srovnání vykreslených snímků za vteřinu (fps) s určitým počtem pohybujících se obrázků (sprite), zdroj: [2] . . . . . | 32 |
|---|-----------------------------------------------------------------------------------------------------------------------|----|



## Seznam obrázků

1	Schéma systému EduKin . . . . .	8
2	Oficiální logo platformy, zdroj: <a href="http://developer.android.com">http://developer.android.com</a> . . . . .	11
3	Koláčový graf zobrazující distribuci verzí na základě dat přístupů během 14 dnů od 18.2.2013 k Google službám, zdroj: <a href="http://developer.android.com">http://developer.android.com</a>	12
4	Jednotlivé vrstvy systému Android, zdroj: <a href="http://developer.android.com">http://developer.android.com</a>	14
5	Životní cyklus activity, zdroj: <a href="http://developer.android.com">http://developer.android.com</a> . . . . .	16
6	Diagram průběhu registrace . . . . .	25
7	Diagram průběhu přihlášení . . . . .	26
8	Diagram výběru uživatele ze skupiny . . . . .	27
9	Hlavní funkce aplikací . . . . .	28
10	E-R diagram databáze . . . . .	29
11	Návrh výběru účtu na mobilním telefonu . . . . .	37
12	Návrh výběru skupiny a žáka . . . . .	38
13	Návrh výběru skupiny a žáka pro menší displeje . . . . .	38
14	Aplikace kreslení . . . . .	42
15	Aplikace asociace . . . . .	43

## Seznam výpisů zdrojového kódu

- 1 Ukázka výběru všech aktivních uživatelů i s jejich osobními údaji pomocí vytvořené databázové vrstvy . . . . . 39
- 2 Ukázka asynchronního načtení obrázku do ImageView komponenty . . . 39

## 1 Úvod

Cílem této práce bylo vytvoření programového rámce, který zjednoduší vývoj aplikací a her pro systém EduKin pro mobilní platformu Android. Tento framework usnadňuje práci s gesty, periferiemi, odstiňuje vývojáře od komunikace se službami systému EduKin, práce s lokální databází a poskytuje další vývojářské nástroje pro zjednodušení vývoje.

První část je věnovaná popisu systému EduKin, jak vznikal, komu má pomáhat, k čemu je určen a jak funguje jako celek.

Druhá část je věnována analýze programového rámce, výběru cílové platformy, jejímu popisu, specifikaci požadavků, procesům, případům užití, lokální databázi a součástí frameworku.

Třetí část popisuje návrh programového rámce, výběr jednotlivých externích knihoven, zvolení preferovaného datového formátu a rozvržení knihoven.

Část čtvrtá se věnuje popisu návrhu grafického uživatelského rozhraní, určuje směrnice pro budoucí vývojáře aplikací nad tímto programovým rámcem a obsahuje návrhy obrazovek předpřipravených v rámci frameworku.

Pátá část obsahuje ukázky kódů, popis problémů, které se při implementaci objevily, popis dvou testovacích aplikací, jejich testování a konzultace s odborníky.

Poslední kapitola je věnována zhodnocení dosažených výsledků, výčet plánů do budoucna a popis nově osvojených znalostí.



## 2 EduKin

Projekt EduKin je zaměřen na vytvoření platformy a jednoduchých nástrojů pro vývojáře aplikací pro osoby se speciálními potřebami. Dále si bere za cíl ulehčit práci učitelům a rodičům pomocí sledování pokroků a jednoduchého sdílení dat mezi členy svých skupin. Díky atraktivnímu zpracování mohou být děti mnohem více vtaženy do hry, tudíž udrží pozornost při cvičení déle.

### 2.1 Vznik

Prvotní impuls pro výběr téma této práce byl návrh vedoucího na sestavení týmu pro celosvětovou soutěž Imagine Cup pořádanou společností Microsoft. Ta měla v roce 2012 jako jedno z osmi témat "Zpřístupnění základního vzdělání pro všechny děti", které se nám líbilo nejvíce. Nově vznikající tým proto začal vymýšlet projekt, se kterým bychom se zapojili do soutěže s tím, že bychom využili moderní technologie jako Windows Azure, Kinect for Windows a tablety s operačním systémem Android.

Postupně se vytvořily dva návrhy, přičemž první byl vytvoření sociální sítě pro zaměstnance a zaměstnavatele, kde by si mohli jednotliví uživatelé vyměňovat zkušenosti a informace. Tento návrh byl po konzultaci s vedoucím diplomové práce odmítnut pro přílišnou podobnost s již úspěšnou sociální sítí profesionálů LinkedIn.

Druhý návrh stavěl na tom, že se vytvoří systém pro děti s postižením, který bude umožňovat rodičům a učitelům sledovat pokroky dítěte a zadávat mu úkoly. Přitom by se využívaly moderní technologie, jako je Windows Azure, Kinect for Windows a tablety s mobilními operačními systémy. Na Windows Azure by měla být nejen uživatelská data, ale i námi vytvořený a spravovaný obchod s aplikacemi, do kterého by mohly přispívat vývojářské firmy prakticky z celého světa. Tento druhý návrh byl nakonec vybrán a byla odeslána přihláška na Imagine Cup v kategorii Software Design.

Abychom získali základní informace o problematice, bylo nutné vyhledat odborníky, kteří by byli ochotní spolupracovat. Jako první byla navštívena pedagogicko-psychologická poradna. Zde byl získán obecný náhled na danou problematiku, co děti potřebují, jaké jsou stávající pomůcky a dostupný software. Právě ten je v současné době problém, protože nové programy jsou drahé a starší už nejsou většinou kompatibilní se současnými operačními systémy.

Další informace byly získány v Brně od členky občanského sdružení Nová rodina. Toto sdružení prosazuje využití zařízení Xbox a Kinect pro rozvíjení zdravotně znevýhodněných dětí. Zde bylo získáno mnoho rad o tom, jak by hry pro tyto děti měly vypadat a co je nevhodné na těch, co již existují.

První kolo soutěže Imagine Cup se v roce 2012 konalo v Brně v prostorách Masarykovy univerzity 26. dubna 2012. Na tuto část soutěže již musely být nachystány menší ukázky a hlavně podrobný popis toho, jakou má projekt životaschopnost, analýza rizik a přibližný časový plán vývoje. Pro tuto příležitost bylo připraveno několik prototypových aplikací. Díky praktickým zkušenostem s vývojem pro platformu Android, byla vytvořena aplikace pro trénování jemné motoriky pomocí obkreslování předkreslených tvarů. Tyto tvary bylo možné sdílet s dalšími uživateli, kteří měli EduKin účet v našem

systemu na Windows Azure, který vzniká v rámci diplomové práce Infrastruktura založená na Windows Azure pro projekt Edukin[6]. Aby byl vyvíjený systém co možná nejkomplexnější, sdílená data mohou být zobrazována ve stejných aplikacích pro různé platformy, takže když bylo vytvořeno malování pro Kinect for Windows, mohli uživatelé místo jemné motoriky na tabletu, cvičit hrubou motoriku před svým osobním počítačem se stejnou sadou tvarů. Dále ještě pro Kinect for Windows vznikla hra šipky, která funkci sdílení pochopitelně neměla, protože se zde nevytváří žádný obsah. Po odprezentování a úspěšném předvedení našich aplikací nám porota dala cenu za nejlepší nápad.

Vytvořit a spravovat vlastní obchod s aplikacemi by bylo příliš zdlouhavé a nákladné, takže bylo rozhodnuto vyvíjený framework po jeho dokončení prodávat dalším tvůrcům aplikací pro postižené. K tomu navíc vytvořit sadu ukázkových aplikací, které by byly zadarmo nebo za nějaký menší obnos, aby se projekt dostal do všeobecného povědomí.

S vylepšeným business plánem byla odeslána přihláška na soutěž Nápad roku pořádanou společností Vodafone. Této soutěže se může zúčastnit každý, kdo splnil poměrně jednoduché podmínky, takže i proto byla účast velká a přihlásilo se 156 projektů, z kterých do druhého kola postoupilo třicet. Projekt EduKin se nakonec umístil ve velké konkurenci na šestém místě.

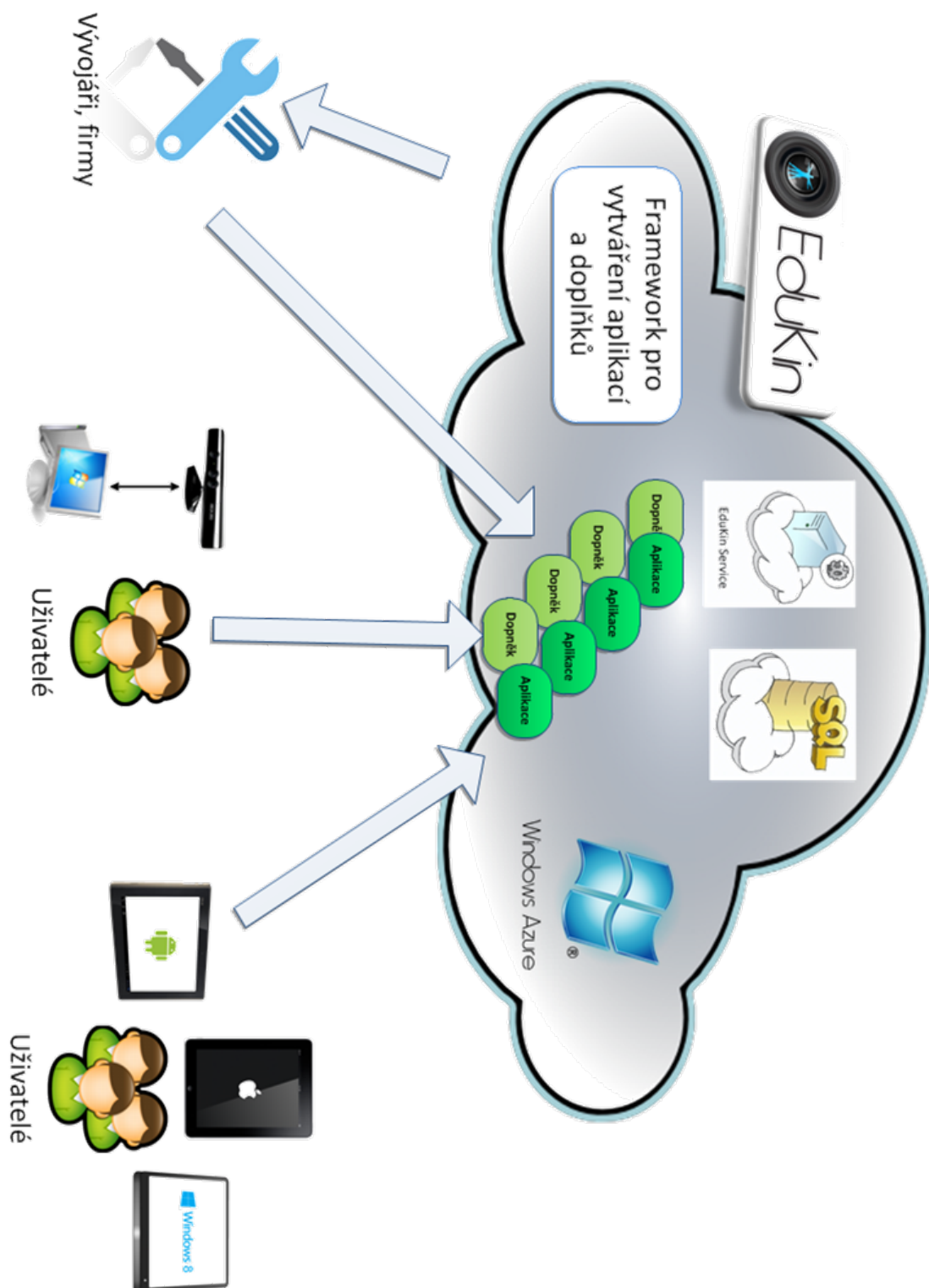
## 2.2 Prvotní představa

Fungování systému jako celku je zobrazeno na obrázku 1. Na něm jsou vidět tři části systému:

**EduKin Service** - provozovaný na Windows Azure, který by uchovával uživatelská data a staral by se o synchronizaci mezi ostatními částmi. Jeho webové rozhraní by také poskytovalo přístup pro vývojáře třetích stran, kteří by pomocí něj mohli přidávat nové aplikace a hry.

**Uživatelé tabletů** - jejichž aplikace a hry komunikují se službami systému EduKin.

**Uživatelé Kinect for Windows** - používající svoje osobní počítače ke hraní her pomocí Kinect a k přístupu ke svému účtu pomocí webového portálu.



Obrázek 1: Schéma systému EduKin



## 3 Cílová platforma

Tato kapitola popisuje cílovou platformu. Nejdříve srovnání existujících platform a po té určení, pro kterou z nich se bude programový rámec vyvíjet. Druhá polovina kapitoly je věnovaná popisu vybrané platformy.

### 3.1 Výběr cílové platformy

Jako první je třeba si definovat, na jaký typ tabletových operačních systémů se zaměřit. Dříve existovaly převážně zařízení označovaná jako tablet-PC, což byl v podstatě notebook ovládatelný i pomocí stylusu, typicky s klasickým operačním systémem pro osobní počítače[12]. S příchodem roku 2010 se začaly objevovat tablety s mobilními operačními systémy uzpůsobenými na ovládání pomocí dotyků prstů a to bez klávesnice či myši. Ty byly levnější a také uživatelsky přívětivější a proto se velmi rychle rozšířily. Je tedy zřejmé, že je vhodné zvolit modernější a rozšířenější typ tabletů, díky kterému lze oslovit skutečně velké masivy lidí. Tyto moderní tablety s mobilními operačními systémy mají navíc typicky integrovanou vlastní distribuční službu pro aplikace, což je další výhoda oproti klasickým operačním systémům.

Dalším krokem byla volba platformy, pro kterou bude programový rámec vyvíjen. V současnosti existují v podstatě tři velmi rozšířené operační systémy: iOS, Android a Windows 8. Poslední jmenovaný se však dostal do prodeje až bezmála rok po zahájení této práce, tudíž bylo nutné zvolit mezi operačními systémy iOS a Android.

Nejdříve tedy vznikl seznam relevantních výhod a nevýhod daných mobilních platform. Následný výběr probíhal na základě informací získaných na konzultacích.

- iOS
  - Výhody
    - \* Kompilovaný nativní kód aplikací
    - \* Velmi složitá instalace nelegálně získané kopie aplikace
    - \* Hojně rozšířený ve Spojených státech amerických
    - \* Malý počet zařízení na testování, minimální počet různých verzí systému
  - Nevýhody
    - \* Pro vývoj je nutné vlastnit zařízení s Mac OS X
    - \* Dražší vývojářský účet, který je nutno každý rok obnovovat
    - \* Horší cenová dostupnost zařízení (iPad) v ČR
    - \* Nemožnost jiné distribuce aplikací než skrze Apple App Store

- Android
  - Výhody
    - \* Velké cenové rozpětí tabletů
    - \* Možnost výběru způsobu distribuce aplikací
    - \* Možnost použití existujících Java knihoven
    - \* Osobní vývojářská zkušenost
  - Nevýhody
    - \* Nelegálně získané aplikace lze jednoduše nainstalovat
    - \* Kód vytvořený v jazyce Java běží na Dalvik Virtual Machine, což vede ke snížení rychlosti
    - \* Velké množství zařízení s různou kvalitou hardware a úpravami operačního systému

Po zvážení všech pro a proti byl nakonec na základě dříve sesbíraných informací vybrán po další vývoj operační systém Android, avšak s iOS je nadále do budoucna počítáno, protože pokud by měl projekt uspět mezinárodně, nelze takovouto platformu přehlížet. Důvody proč byl zvolen Android:

1. zařízení jsou cenově dostupnější;
2. možnost distribuce testovacích verzí aplikací bez nutnosti publikace;
3. rychlý začátek vývoje díky osobním zkušenostem s vývojem pro vybranou platformu.

## 3.2 Popis vybrané platformy

Android je open-source softwarová platforma založená na operačním systému Linux vyvíjená společností Google Inc. Je určena především pro mobilní zařízení, jako jsou chytré telefony, tablety či PDA s dotykovou obrazovkou, jak uvádí Ed Burnette [7] a Reto Meier [10]. Avšak již existují produkty, jako jsou chytré televize, herní konzole nebo úsporné notebooky, ve kterých své uplatnění, byť v upravené podobě, tento operační systém také našel. Android je dnes nejrozšířenější mobilní operační systém vůbec a podle statistik společnosti Google se denně aktivuje přes jeden milión zařízení s tímto operačním systémem. Na obrázku 2 je vyobrazeno jeho logo.

### 3.2.1 Historie

Podle [9] byl Android od roku 2003 původně vyvíjen stejnojmennou společností, kterou založili Andy Rubin, Rich Miner, Nick Sears a Chris White. Do roku 2005 byla tato společnost pouze jednou z mnoha, což se ale změnilo ve chvíli, kdy ji v tom roce odkoupila společnost Google. Ta po akvizici pokračovala v rychlém vývoji pod vedením Andy Rubina a v získávání mobilních patentů, aby mohla v roce 2007 oficiálně ohlásit Android,



Obrázek 2: Oficiální logo platformy, zdroj: <http://developer.android.com>

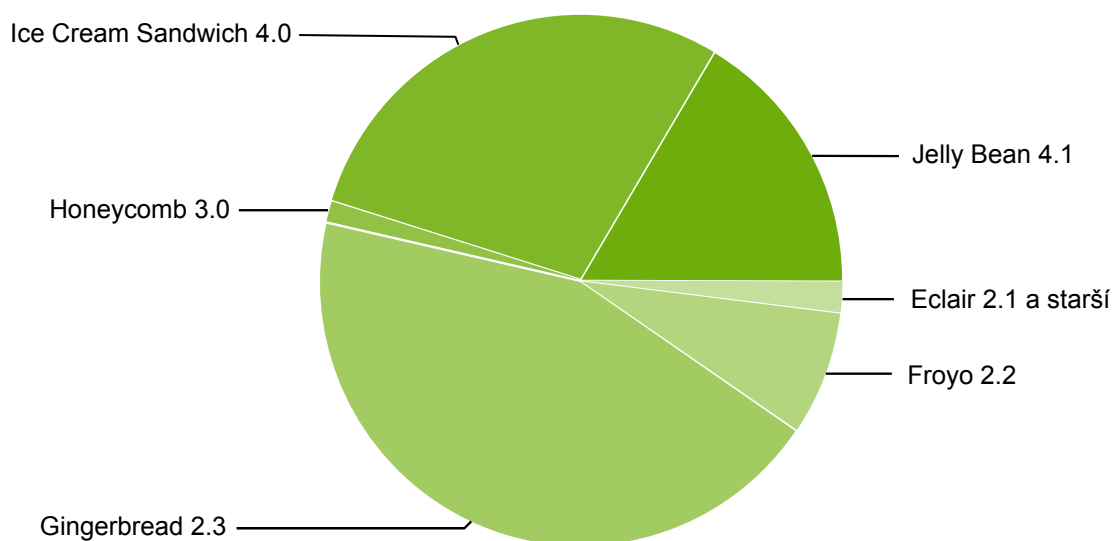
jako jejich novou mobilní platformu, a vznik Open Handset Alliance. Toto konsorcium sdružuje 84 technologických a mobilních společností, které chtějí přinést více otevřenosti do mobilního ekosystému. Dále také prosazovat nové otevřené standardy, poskytovat vývojářům nástroje pro vytváření inovativních řešení a kooperovat další vývoj systému Android [11].

Pro podporu rychlého rozvoje platformy společnost Google uspořádala v roce 2008 Android Developer Challenge, která byla určená pro vývojáře nových aplikací. Ti zde soutěžili v různých kategoriích o peněžitou odměnu v maximální výši 275 000 dolarů. Porota sestavená společností Google měla za úkol vybrat deset nejlepších, kteří získali právě onu odměnu a dalších deset získalo 100 000 dolarů. Celkově do této soutěže přispívali vývojáři z 70 zemí a vytvořili na 1800 aplikací [1].

### 3.2.2 Historie verzí

Jak bylo již naznačeno v nevýhodách platformy Android, existuje poměrně velké množství verzí systému, které se stále používají z důvodu pomalé aktualizace starších modelů zařízení jejich výrobci, jak je vidět na obrázku 3. To vede k vysoké fragmentaci a klade vyšší nároky na vývoj aplikací. S ohledem na to, že pro EduKin jsou primární cílová zařízení tablety, byla zvolena jako minimální podporovaná verze operačního systému Android 3.0. Hlavní důvod je, že tato verze je první, která byla opravdu připravena pro zařízení s obrazovkou větší než 5 palců a zároveň poskytuje aplikační rozhraní vhodné pro vývoj aplikací pro tablety. Dalším důvodem je přítomnost hardwarové akcelerace vykreslování uživatelského rozhraní a zajištění alespoň minimální hardwarové konfigurace tabletu. Tedy obsah následujícího popisu verzí už nebude zahrnovat verze, na které nejsou aplikace cílené, navíc je obdobný popis obsažen v autorově bakalářské práci.





Obrázek 3: Koláčový graf zobrazující distribuci verzí na základě dat přístupů během 14 dnů od 18.2.2013 k Google službám, zdroj: <http://developer.android.com>

- Android 3.0 (Honeycomb)
  - Verze byla vydána 22. února 2011
  - Aplikace umožňují povolení hardwarové akcelerace pomocí grafického čipu
  - Podpora pro vícejádrové procesory
  - Nové prvky uživatelského rozhraní pro tablety
  - Lepší podpora dotykových gest
  - Nový animační framework
- Android 4.0 (Ice Cream Sandwich)
  - Verze byla vydána 19. října 2011
  - Sloučení tabletové a mobilní verze, což přineslo novinky z verze 3 na telefony
  - Vylepšené rozpoznávání řeči a API pro převod textu na řeč
  - Vylepšená podpora pro uživatele se zrakovým postižením
- Android 4.1/4.2 (Jelly Bean)
  - Verze byla vydána 9. Července 2012 na akci Google I/O
  - Zvýšení responsivity systému díky projektu Butter
  - Podpora pro více uživatelský přístup
  - Další vylepšení podpory pro uživatele se zrakovým postižením
  - Vylepšení nástrojů pro vývojáře

### 3.2.3 Architektura

Android je postaven nad operačním systémem Linux, části jeho kódu jsou napsány v jazyce C, C++ a Java [3]. Protože je Android mobilní platforma, bere si z Linuxu jen některé části, například grafické uživatelské rozhraní X Windows Server by bylo příliš náročné a špatně ovladatelné. Android SDK obsahuje aplikační rozhraní a nástroje, čímž poskytuje vše potřebné pro vývoj aplikací v jazyce Java. Android je vícevrstvý operační systém, díky čemuž zvyšuje svou bezpečnost. Na obrázku 4 jsou tyto vrstvy znázorněny, následuje jejich detailnější popis [4]:

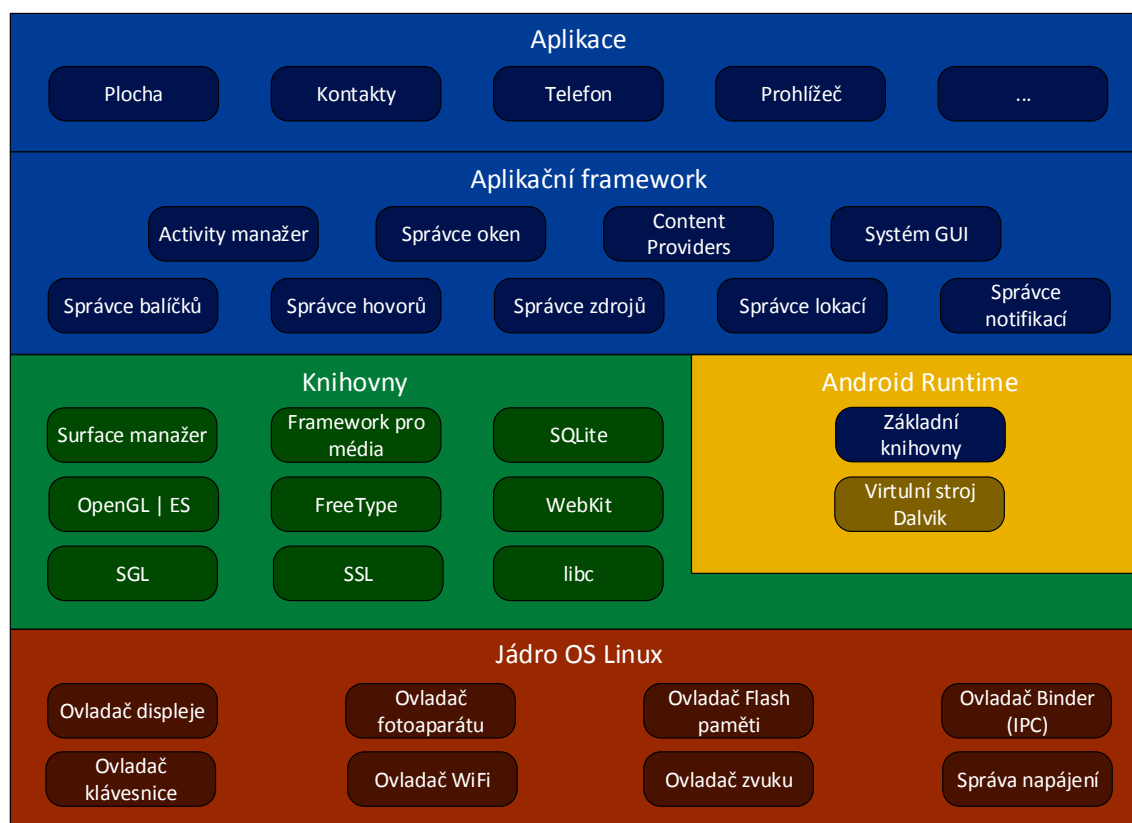
**Linuxové jádro** - se ve verzi 2.6 stará o bezpečnost, správu paměti a procesů, sítí a ovladače, také se chová jako abstraktní vrstva mezi hardwarem a zbytkem softwaru ve vyšších vrstvách. Každá aplikace na této platformě běží pod vlastním procesem a s vlastními oprávněními přidělenými systémem. Toto jádro má navíc tu výhodu, že lze poměrně snadno sestavit na různých architekturách.

**Android Runtime** - zahrnuje soubor základních knihoven, které poskytují většinu funkcionality jako knihovny Java Standard Edition, avšak využívají licenčně nezátížené implementace Apache Harmony. Z tohoto důvodu vznikl virtuální stroj Dalvik, který je optimalizován tak, aby efektivně umožnil běh několika jeho instancí najednou. To proto, že každá aplikace nejen že běží ve vlastním procesu, ale má i vlastní instanci tohoto virtuálního stroje.

**Knihovny** - jsou soubor C a C++ kódů, které jsou používány různými komponentami operačního systému Android. Používání těchto knihoven je umožněno vývojářům pomocí aplikačního rozhraní skrze Android aplikační framework. Jsou to například knihovny pro přehrávání zvuku a videa OpenCORE, engine webového prohlížeče LibWebCore, 2D grafický engine SGL, knihovna pro vizualizaci 3D grafiky OpenGL ES, nástroj pro vykreslování vektorových fontů FreeType a odlehčená relační databáze SQLite.

**Aplikační framework** - umožňuje vývojářům plně přistupovat ke stejnému aplikačnímu rozhraní tak, jako předinstalované aplikace, tedy až na výjimky v podobě Google Play nebo Dialer. Architektura aplikací je navržena tak, aby lze jednoduše opakovaně používat jejich komponenty, aplikace tudíž mohou zpřístupnit své schopnosti jiným aplikacím. Například lze z jedné aplikace vyvolat aplikaci Galerie a v ní vybrat fotku, která bude předána zpět. Toto ovšem s sebou nese bezpečnostní rizika, která Android řeší systémem oprávnění. Jejich seznam je uvedený v Manifest souboru aplikace, který se schvaluje při instalaci aplikace. Pokud se aplikace pokusí udělat akci, kterou nemá uvedenou v tomto seznamu, je vyvolána výjimka.

**Aplikace** - jsou nejvyšší vrstva, ke které už má přístup uživatel zařízení. Jde o aplikace jak předinstalované, tak o aplikace vývojářů třetích stran. Ty lze instalovat přímo z Google Play obchodu nebo z alternativního obchodu s aplikacemi či přímo z SD karty zařízení.



Obrázek 4: Jednotlivé vrstvy systému Android, zdroj: <http://developer.android.com>

### 3.2.4 Aplikační komponenty

Aplikační komponenty jsou základní stavební kameny každé aplikace. Každá může sloužit jako vstupní bod systému do aplikace. Aby toto bylo umožněno, musí být daná komponenta zaregistrována v aplikačním Manifest souboru. Komunikace pak mezi nimi probíhá pomocí zpráv (Intent). Kupříkladu po klepnutí na tlačítko nápověda, aplikace vytvoří tuto zprávu obsahující informaci o tom, že se má spustit webový prohlížeč a otevřít stránku s nápovědou [5]. Aplikační komponenty systému Android jsou tyto:

**Activity** - reprezentuje jednu obrazovku s grafickým uživatelským rozhraním. *Activity* v rámci celé aplikace i systému jsou na sobě nezávislé, díky tomu mohou být vyvolány *Intent* zprávou z jiné *activity* nebo *service*. Aplikace se typicky skládá z více komponent typu *activity*, například aplikace Gmail obsahuje *activity* pro zobrazení seznamu příchozích e-mailů a další pro zobrazení detailu vybraného e-mailu. *Activity* se seznamem po klepnutí na vybranou položku předá zprávu do *activity* pro detail obsahující identifikátor záznamu, který má zobrazit. Vzhled a rozložení prvků uživatelského rozhraní lze definovat pomocí XML ve zdrojích aplikace nebo je vytvářet za běhu. *Activity* mají vlastní životní cyklus znázorněný na obrázku 5.



**Service** - realizuje multitasking na operačním systému Android. Zatímco *activity* se uspí, když uživatel opustí aplikaci pomocí tlačítka Home nebo když se spustí jiná *activity*, *service* běží stále dál nezávisle na komponentě typu *activity*, která ji vyvolala. Jelikož ve své podstatě jde o službu, *service* nemá žádné uživatelské rozhraní, ale umožňuje připojení jiné *activity*, která ji bude ovládat. Existují dva typy těchto služeb, *remote* a *local*. *Remote* služby běží dokud se samy neukončí nebo dokud je explicitně neukončí jiná aplikační komponenta. *Local* služby běží jenom po dobu, kdy je k ní připojená komponenta, která jí spustila. Typické použití je pro dlouho trvající činnosti, jako je přednačítání dat, přehrávání hudby či záznam trasy pomocí GPS modulu.

**Content provider** - umožňuje aplikaci sdílet data s ostatními aplikacemi, přitom nezáleží, kde jsou tato data uložena. Tyto aplikace se na tato data mohou dotazovat a dokonce je i měnit. *Content provider* musí implementovat standardní funkce aplikačního rozhraní, aby aplikace mohla vykonat potřebné transakce. Každá takto vytvořená komponenta by měla definovat vlastní oprávnění nutné k tomu, aby k ní mohla přistupovat jiná aplikace. Uživatel tak při instalaci vidí, k jakým datům stávající aplikace má nově instalovaná aplikace přístup. Příklad použití je přístup k údajům z kalendáře, kdy přes *content provider* lze získat nadcházející události.

**Broadcast receiver** - je aplikační komponenta, která umožňuje reagovat na systémové zprávy, jako je spuštění telefonu, připojení k internetu či pořízení fotografie. Typicky se používá pro spuštění synchronizace po přihlášení do sítě.

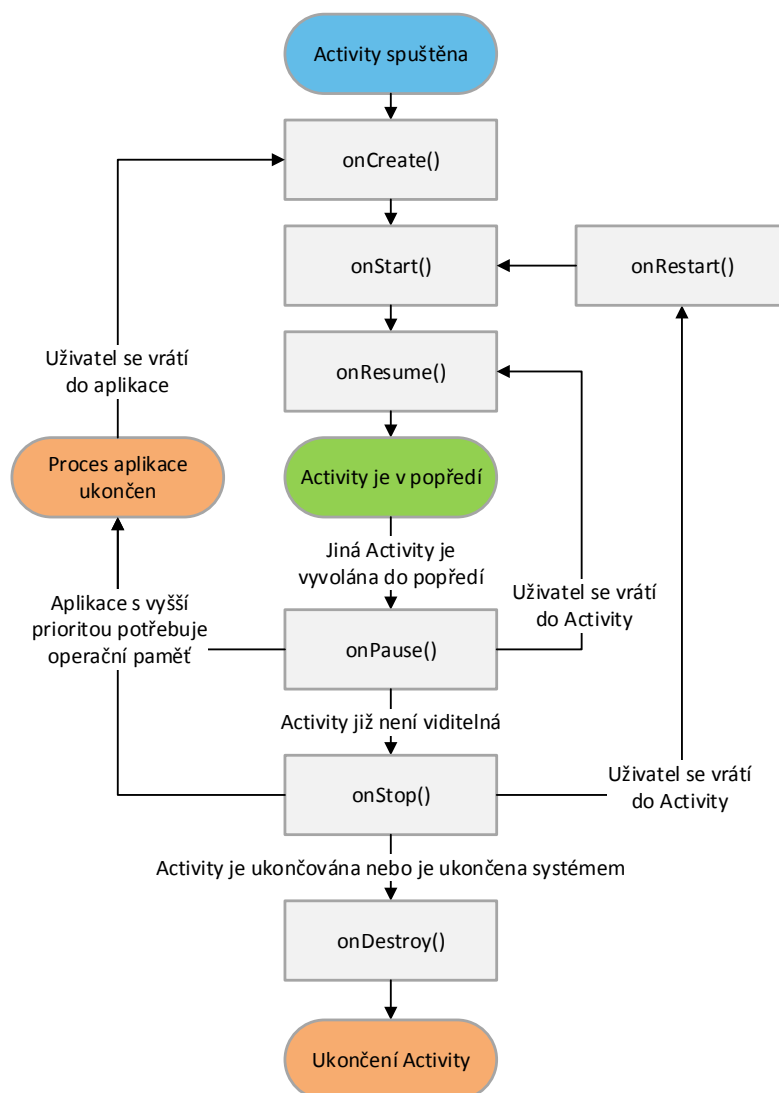
### 3.2.5 Vývoj aplikací

Vývoj samotné aplikace pro platformu Android probíhá v jazyce Java s možností tvorby grafického uživatelského rozhraní v XML a volání C/C++ kódu přes rozhraní JNI. Pomocí XML lze ale také definovat například vizuální styly, tvary, barvy, lokalizované texty a další. Tyto zdroje jsou pak při sestavení zkomprimovány a jinak upravovány pro maximalizaci výkonu. Pro sestavení aplikace je nutné mít nainstalováno oficiální Android SDK, které lze stáhnout z <http://developer.android.com>. To je dostupné pro operační systém Linux, Mac OS X a Windows a obsahuje různé nástroje od obfuskátoru<sup>1</sup> kódu ProGuard, po tvorbu 9Patch<sup>2</sup> obrázků. Na výše zmíněné webové adrese se také nachází dokumentace, návody a ukázky kódů, ale také směrnice návrhu uživatelského rozhraní.

Společnost Google zvolila jako jí oficiálně podporované IDE Eclipse v minimální verzi 3.6.2 (Helios), ke které dodává doplněk ADT, který realizuje o integraci Android SDK.

<sup>1</sup>Obfuskace je proces pro znesnadnění čitelnosti kódu po dekompilaci jinou osobou. Jsou odstraněny komentáře, bílé znaky a především jsou přejmenovány proměnné, názvy metod a tříd s využitím polymorfismu.

<sup>2</sup>Obrázky, které mají definované části, jež se mohou opakovat. To například umožňuje vytvoření tlačítek s oblými rohy, které nejsou deformované podle velikosti onoho tlačítka.



Obrázek 5: Životní cyklus activity, zdroj: <http://developer.android.com>

Díky této architektuře lze vyvíjet aplikace i v jiných vývojářských prostředích jako je IntelliJ nebo NetBeans, ale vývojář by pak měl počítat s tím, že se objeví problémy.

Android SDK po instalaci neobsahuje žádný emulátor nebo SDK některé z verzí systému. Ty je třeba stáhnout pomocí nástroje Android SDK Manager. Je to z důvodu narůstajícího počtu verzí systému, což by způsobilo nárůst velikosti SDK v řádu gigabajtů.

### 3.2.6 Publikace aplikace

Ve chvíli, kdy je aplikace hotová a řádně otestovaná, nejlépe pomocí automatizovaných testů, je vše připraveno k jejímu umístění na Google Play. To je výchozí obchod s apli-

---

kacemi pro platformu Android, který spravuje společnost Google. Ještě před nahráním aplikace na server, je podle vlastních zkušeností dobré provést vyčištění projektu, což způsobí znovuvytvoření seznamu zdrojů. Ten se právě může během vývoje poškodit a posléze způsobovat nesmyslné chyby a pády aplikace. Dále je nutné aplikaci sestavit v release módu a podepsat pomocí svého vygenerovaného klíče. Na závěr použijeme program `zipalign.exe` z Android SDK, který provede optimalizaci obrázků a dalších zdrojů tak, aby při běhu zabíraly co nejméně operační paměti.

Po tom, co jsou tyto požadavky splněny, můžeme instalační balíček aplikace s příponou `apk` nahrát na <https://play.google.com/apps/publish> pod vaším vývojářským účtem. Kromě samotného balíčku je nutné nahrát také ikonu aplikace v rozměru 512 na 512 pixelů a další dva snímky obrazovky. Velmi vhodné z hlediska propagace je také přidat další volitelnou grafiku. Při publikaci lze také zvolit země, ve kterých lze aplikaci nainstalovat. Po publikaci servery provedou analýzu vaší aplikace a pokusí se odhalit jakýkoli nežádoucí kód. Tento proces trvá asi hodinu. Pokud aplikace není označena za nebezpečnou, je zpřístupněna všem uživatelům, které jste nadefinovali.

Tato webová konzole také zpřístupňuje zobrazení nahlášených pádů aplikace, počet aktivních a celkových instalací, různé statistiky, hodnocení, komentáře a příjmy z prodeje aplikací.

## 4 Analýza

Tato kapitola se bude zabývat analýzou požadavků na základě informací získaných během konzultací s odborníky a lidmi z praxe. Konkrétně zde bude specifikace požadavků, funkční specifikace a ukázky diagramů případů užití. Na závěr bude určeno z jakých částí se bude framework skládat.

### 4.1 Specifikace požadavků

Přesná definice požadavků je velmi důležitá pro správný návrh jakéhokoli systému, proto je mu věnována následující podkapitola obsahující specifikaci funkčnosti programového rámce, ale i nutných programových a hardwarových zdrojů.

#### 4.1.1 Specifikace funkčnosti programového rámce

Prvotní vize koncepce systému EduKin byla zaměřena na spolupráci tabletů a osobních počítačů s připojeným Kinect for Windows s Windows Azure. Tato spolupráce je také jeden z hlavních požadavků.

Aby toto bylo možné, musí na straně serveru být dostupné standardizované aplikační rozhraní, které budou klientské aplikace využívat. Z tohoto důvodu musí framework obsahovat knihovnu zaštitující komunikaci s WFC službami systému EduKin a nástroje pro práci s binárními daty z úložiště Azure Storage. Toto zahrnuje autentizaci, získání informací o účtu (profilová fotka, jméno a příjmení), získání rolí přihlášeného uživatele a skupin. Dále pak získání a odesílání nastavení dané aplikace, dosažených výsledků, složek a metadat souborů. Pomocí těchto metadat, což je popis vlastností daného souboru uloženého v Azure Storage, je umožněno tato data získat či odeslat na server.

S komunikací se serverem souvisí dostupnost připojení k internetu. To může mít různou kvalitu v rámci institucí a u větších komplexů nemusí být ve třídách dostupné vůbec. Proto je dalším požadavkem režim bez přístupu na internet. Ten by měl být vyřešen ukládáním existujících dat do paměti zařízení, nejlépe do relační databáze, která bude z části reflektovat strukturu databáze na serveru. Na platformách Android i iOS je součástí systému databázový systém SQLite. Knihovna by měla ulehčit práci s databází s pomocí vlastního aplikačního rozhraní.

Pro vytváření her je nutné také do programového rámce zahrnout knihovnu, která bude obsahovat robustní aplikační rozhraní pro vykreslování herní plochy. Opět na obou platformách je součástí systému knihovna OpenGL, nad kterou by se mělo stavět. Problém je, že vytvoření takové knihovny by samo o sobě vydalo na diplomovou práci, proto bylo rozhodnuto použít některou z již existujících. Ta by měla být schopna vykreslovat 2D grafiku i 3D modely a alespoň základní prvky grafického uživatelského rozhraní.

Pro vybranou knihovnu pro tvorbu her je nutné také vytvořit knihovnu pro detekci gest, která bude připravená na uživatele se špatnými motorickými schopnostmi. Android i iOS opět takové knihovny nabízejí již v základu, ale ty nejsou přizpůsobeny pro tyto uživatele.



Součástí snad každého programového rámce je kolekce nástrojů, které usnadní integraci a práci s daným frameworkem. Nástroje by měly obsahovat třídy usnadňující asynchronní načítání obrázků, graficky atraktivní prvky grafického uživatelského rozhraní a různé užitečné metody.

Nad tímto vším by měly být postaveny 3 předpřipravené obrazovky, které budou konsistentní ve všech aplikacích zařazených do systému EduKin. Jsou to tyto:

**Obrazovka přidání účtu** - umožní odstínit vývojáře od registrace nového účtu nebo prvního přihlášení uživatele. Bude tedy obsahovat grafické uživatelské rozhraní s formuláři pro první přihlášení a registraci účtu. Po platné verifikaci údajů je nový účet automaticky přidán mezi účty uložené v zařízení.

**Obrazovka výběru účtu** - zobrazí seznam účtů uložených v zařízení, které jsou tedy dostupné i bez připojení k internetu, a tlačítko pro přidání nového, které vyvolá obrazovku pro přidání účtu. Po tom, co se uživatel úspěšně přihlásí, je jeho účet nastaven jako aktivní. K tomu následně přistupuje vývojář aplikace, který neřeší samotné přihlášení, pouze po spuštění své aplikace vyvolá tuto obrazovku.

**Obrazovka výběru účtu ze skupiny** - zobrazí seznam skupin, které patří přihlášenému uživateli. Po výběru skupiny je zobrazen seznam klientů, kteří byli do této skupiny přiděleni. Po vybrání účtu klienta je nastaven aktivní účet skupiny a aplikace se přepne do rozhraní pro něj, zatímco aktivní účet je zachován.

#### 4.1.2 Systémové specifikace

Aplikace vytvořené pomocí tohoto programového rámce budou minimálně vyžadovat zařízení s operačním systémem Android ve verzi 3.0 Honeycomb. Toto se může lišit v závislosti na aplikaci stejně jako programová vybavenost zařízení. Například převod českého textu na řeč se může realizovat jinou aplikací.

#### 4.1.3 Technické specifikace

Programový rámec bude využívat převážně nativních prvků uživatelského rozhraní systému Android, proto pro jeho provoz musí postačovat každé zařízení, na kterém plynule běží Android 3.0 a vyšší. Toto opět ale nemusí platit pro všechny aplikace postavené nad tímto programovým rámcem, neboť ty mohou vyžadovat specifické senzory či grafický procesor, který podporuje OpenGL 2.0.

#### 4.1.4 Uživatelé

Finální aplikace by měly rozlišovat i role jednotlivých uživatelů, aby zobrazily vhodné rozhraní pro danou roli podle jejich práv. Popis rolí a jejich práv je obsažen v podkapitole 4.2, následuje seznam těchto rolí:

- klient,
- rodič,
- učitel,
- ředitel,
- vývojář,
- administrátor.

#### 4.1.5 Použitelnost

Při tvorbě jednotlivých obrazovek je nutné dbát rad odborníků, se kterými budou probíhat konzultace a také směrnic společnosti Google pro návrh a tvorbu uživatelského rozhraní. To z toho důvodu, aby vytvořené GUI bylo co možná nejvíce uživatelsky přívětivé a zároveň vizuálně zapadalo do koncepce designu platformy.

#### 4.1.6 Spolehlivost

Díky synchronizaci s Windows Azure jsou uživatelská data zálohována společností Microsoft, takže v případě poškození přístroje či jeho odcizení, stačí aby se uživatel pouze přihlásil na jiném zařízení. Výsledné aplikace by měly využívat podpory režimu bez přístupu na internet a v případě, že je zařízení v tomto režimu, ukládat si vytvořená data pro pozdější odeslání na server.

Pro případ slabé konektivity musí aplikace obsahovat volbu pro manuální spuštění synchronizace, aby se mohla stáhnout data, která při předchozí synchronizaci nebyla přijata korektně.

#### 4.1.7 Výkonnost

Jak již bylo naznačeno výše, samotný programový rámec bude využívat nativní grafické prvky uživatelského rozhraní a tedy by měl být poměrně nenáročný. Avšak je více než vhodné, aby vývojář koncové aplikace v konfiguračním Manifest souboru povolil hardwarovou akceleraci.

### 4.2 Funkční specifikace

V této podkapitole budou popsány role aktérů systému. Ti by měli mít různé možnosti přímo v aplikacích i na webovém portálu systému EduKin.

### 4.2.1 Klient

Klient je nejzákladnější role systému, určená hlavně dětem, případně klientům, kteří vyžadují zvýšenou péči. Uživatel s touto rolí má na portále EduKin možnost hodnotit aplikace, spravovat a sdílet svá aplikační data a posílat zprávy ostatním uživatelům.

V rámci aplikace by měl mít co nejméně možností, ať se nemusí zabírat jinými věcmi než cvičením. Například v aplikaci kreslení by měl mít co nejjednodušší přístup ke kreslení, aby se nemusel zdržovat nastavováním a podobnými činnostmi.

### 4.2.2 Rodič

Role rodič má všechna práva jako klient, ale navíc je mu umožněno vytvářet nové uživatele s rolí klienta. Ti pak jsou svázáni s jeho účtem a on pak může kontrolovat jejich postup, sdílet s nimi data a podobně. Toto je realizováno systémem skupin, do kterých rodič účty svých potomků může přiřazovat. Ti pak mohou využívat aplikace, které rodič vlastní. Rodič může vytvářet skupiny typu Rodina.

V aplikaci by měl mít rodič zpřístupněny všechny možnosti, které aplikace nabízí. Uvedeme opět příklad na aplikaci kreslení:

- vizuálně méně efektivní seznam tvarů (ale co nejpřehlednější),
- vytváření tvarů,
- zobrazení seznamu skupin a klientů v nich,
- měnit nastavení u vybraného klienta.

### 4.2.3 Učitel

Tato role je prakticky stejná jako role rodič, ale s malými obměnami. Učitel vytváří skupiny typu Třída. Nastavení aplikací jednotlivých žáků, které pochází od učitele, nemůže uživatel v roli rodič měnit.

### 4.2.4 Ředitel

Role ředitel přejímá všechny oprávnění role učitel a přidává vlastní. Tato role nejen že může přistupovat do všech skupin v rámci ústavu či instituce, kterou vede, ale je informován pomocí push notifikací o aktivitě uživatelů v roli učitel.

### 4.2.5 Vývojář

Uživateli s touto rolí je umožněno přidávat globální aplikace a jejich varianty pro různé platformy. Zde také získá jejich identifikátory pro další komunikaci se službami systému EduKin. Role vývojář také umožní definování položek nastavení pro globální aplikace, čímž je zajištěna jejich synchronizace pro různé platformy.

#### 4.2.6 Administrátor

Uživatel v roli administrátor má všechny možnosti jako výše popsané role a navíc může měnit údaje a data ostatních uživatelů a aplikací. Tato role má tedy nejvyšší oprávnění ze všech.

### 4.3 Procesy

Tato podkapitola je věnována popisu nejdůležitějších procesů. Jelikož cílem této práce je vytvoření frameworku, jsou zde rozepsány procesy, které jsou v programovém rámci předpřipraveny pro vývojáře. Například průběh nahrávání nových souborů do systému EduKin se může lišit případ od případu, avšak níže popsané procesy by měly zůstat stejné v rámci všech aplikací pro tento systém.

#### 4.3.1 Registrace nového uživatele

Uživatel má na výběr dvě možnosti, jak může získat nový účet, a to buď přes portál EduKin nebo pomocí aplikace. V obou případech musí vyplnit registrační formulář, kde jsou povinné parametry jméno, jedinečná e-mailová adresa, která není spojena s žádným účtem v databázi, heslo a pohlaví. Pokud již účet se zadanou e-mailovou adresou existuje, je uživatel vyzván k zadání jiné adresy. Toto ověření se děje z toho důvodu, že e-mailová adresa slouží jako přihlašovací jméno. Když je formulář vyhodnocen jako správně vyplněný, je do schránky na zadané adrese odeslán aktivační e-mail s odkazem pro aktivaci účtu, viz obrázek 6.

#### 4.3.2 Přihlášení uživatele

Při přihlašování uživatele mohou nastat dva stavy, buď se již na zařízení již někdy přihlásil nebo ještě ne, přestože vlastní účet, viz obrázek 7.

V prvním případě prostě vybere svůj profil z těch uložených na zařízení a vyplní textové pole pro zadání hesla v zobrazeném dialogu. Pokud není zařízení připojeno k internetu, je heslo ověřeno oproti kontrolnímu součtu v lokální databázi, jinak je uživatel ověřen pomocí přihlašovací služby systému EduKin. V případě, že heslo nesouhlasí, je uživatel požádán o zadání korektního hesla.

V druhém případě musí uživatel vyplnit přihlašovací formulář, jež se skládá z přihlašovacího jména, což je jeho e-mailová adresa, a hesla. Tyto údaje musí být ověřeny pomocí přihlašovací služby, tudíž zařízení musí být pro tuto operaci připojeno k internetu, jinak je uživatel vyzván k připojení svého zařízení k internetu. V případě že jsou vloženy údaje správné, uživatel je uložen v zařízení pro usnadnění dalšího používání a k případnému používání bez přístupu na internet.

#### 4.3.3 Práce se skupinami

Skupiny jsou určeny převážně rodičům a učitelům pro jednoduchou organizaci a přepínání mezi účty klientů. Jejich tvorba a přiřazování probíhá pomocí portálu EduKin.

Když po přihlášení uživatel v roli učitel nebo rodič vyvolá nabídku skupiny, zobrazí se mu seznam skupin, které vytvořil. V nich je obsažen seznam uživatelů, které si do nich zařadili. Po provedení výběru uživatele je stále přihlášen původní uživatel, ale díky jeho pravomocem jsou zobrazena data vybraného uživatele ze skupiny. Průběh je znázorněn na obrázku 8.

## 4.4 Případy užití

Diagram na obrázku 9 poskytuje rychlý přehled o tom, co by výsledné aplikace mohly s použitím tohoto programového rámce umět. Dále jsou zde nejdůležitější postupy přesně zachyceny v textové podobě.

### 4.4.1 Registrace

<b>Stručný popis</b>	Cílem je vytvoření uživatelského účtu pro přístup k webovému rozhraní EduKin a synchronizaci se WCF službou na Microsoft Azure.
<b>Prekondice</b>	Vlastní aktivní e-mailová adresa.
<b>Postkondice</b>	Účet je vytvořen, po jeho aktivaci je umožněno se k němu přihlašovat. Jako přihlašovací údaje slouží právě e-mailová adresa a zadané heslo.
<b>Základní průběh</b>	Po správném vyplnění registračního formuláře na webu systému EduKin a ověření jedinečnosti e-mailové adresy v databázi, je na ni odeslána zpráva s aktivačním odkazem.
<b>Alternativní průběh</b>	Uživatel vyplní formulář přímo v aplikaci, pokud je zadaný e-mail nepoužívaný, je mu do schránky zaslána zpráva s aktivačním odkazem.

### 4.4.2 Přihlášení

<b>Stručný popis</b>	Slouží k autentizaci uživatele a k získání autentizačního tokenu, který pak následně aplikace posílá při každém požadavku.
<b>Prekondice</b>	Aktivovaný uživatelský účet.
<b>Postkondice</b>	Uživatel je autentifikován.
<b>Základní průběh</b>	Uživatel se přihlašuje z daného zařízení poprvé, proto musí vyplnit přihlašovací jméno a heslo. Pokud zadá nesprávné údaje, je vyzván ke kontrole zadaných údajů.
<b>Alternativní průběh</b>	Uživatelův účet je již uložen na zařízení a proto již zadává pouze heslo. Pokud je nesprávné, je vyzván k jeho kontrole.



### 4.4.3 Skupiny

<b>Stručný popis</b>	Slouží k rychlému přepínání uživatelů v rámci skupiny, kterou vytvořil přihlášený učitel či rodič.
<b>Prekondice</b>	Přihlášený učitel nebo rodič má vytvořenu alespoň jednu skupinu s uživateli.
<b>Postkondice</b>	Vybraný uživatel ze skupiny je nyní aktivní.
<b>Základní průběh</b>	Rodič nebo učitel vybere skupinu a pak klienta ze skupiny, se kterým bude provádět požadovanou činnost.
<b>Alternativní průběh</b>	Rodič nebo učitel klikne na tlačítko zpět, čímž zruší výběr skupiny a klienta, po té se vrátí zpět do té části aplikace, kde kliknul na volbu Skupiny.

## 4.5 Lokální databáze

Návrh struktury lokální databáze je na diagramu 10. Tento návrh z velké části reflektuje strukturu databáze na databázovém serveru systému EduKin. Výjimky tvoří atributy s prefixem `_Db`, které slouží k umožnění lokálního ukládání dat v režimu bez přístupu na internet.

## 4.6 Součásti frameworku

Je zřejmé, že programový rámec bude třeba rozdělit do logických celků, které se budou starat o různé aspekty konečné aplikace místo vývojáře. Pro splnění cíle práce budou potřeba tyto:

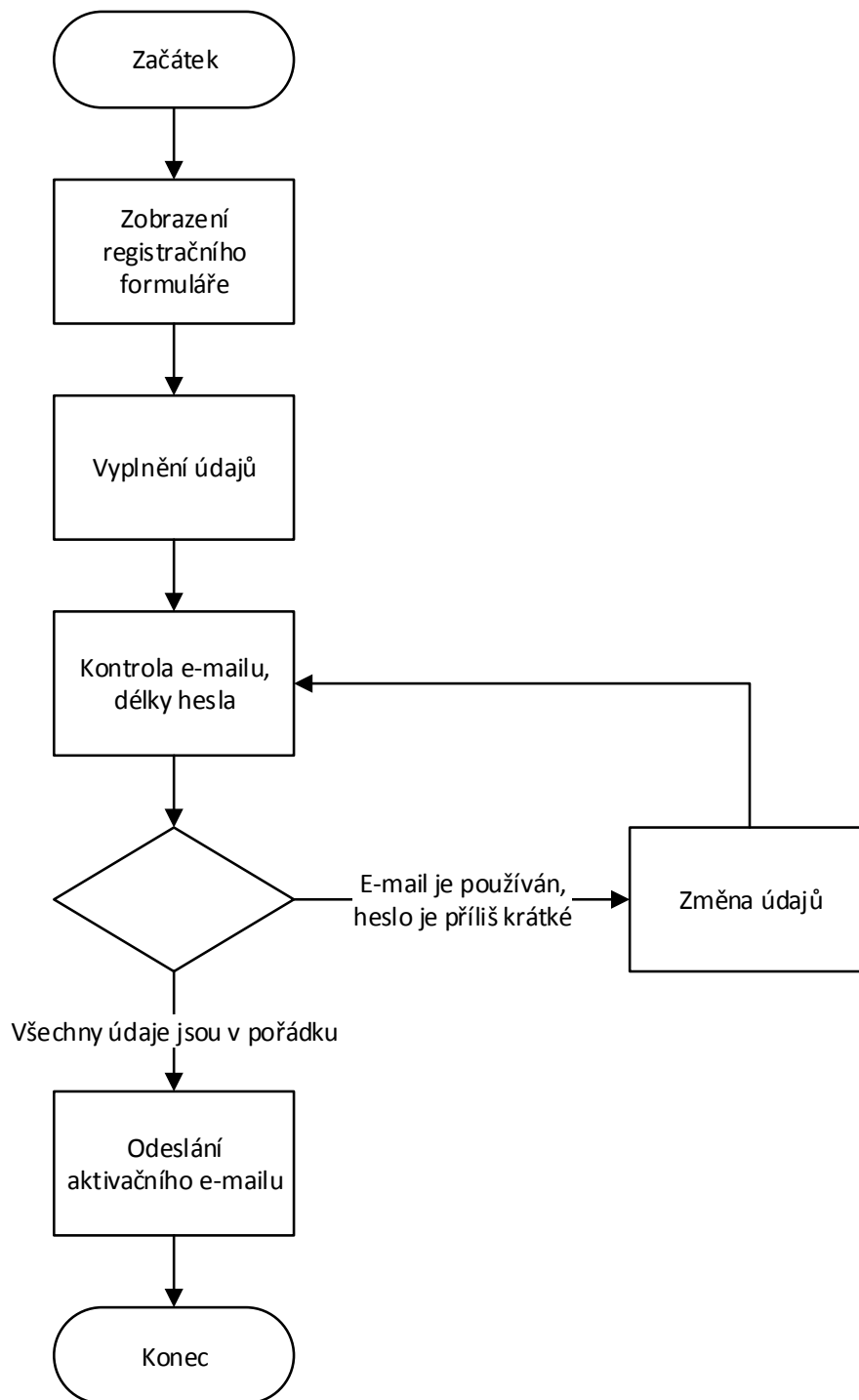
**Komunikace se servery systému EduKin** - poskytne jednoduché aplikační rozhraní pro synchronní i asynchronní volání metod služeb systému EduKin.

**Předvytvořené aplikační komponenty** - jsou převážně formuláře, které jsou pro všechny aplikace společné, například pro registraci, přihlášení nebo skupiny.

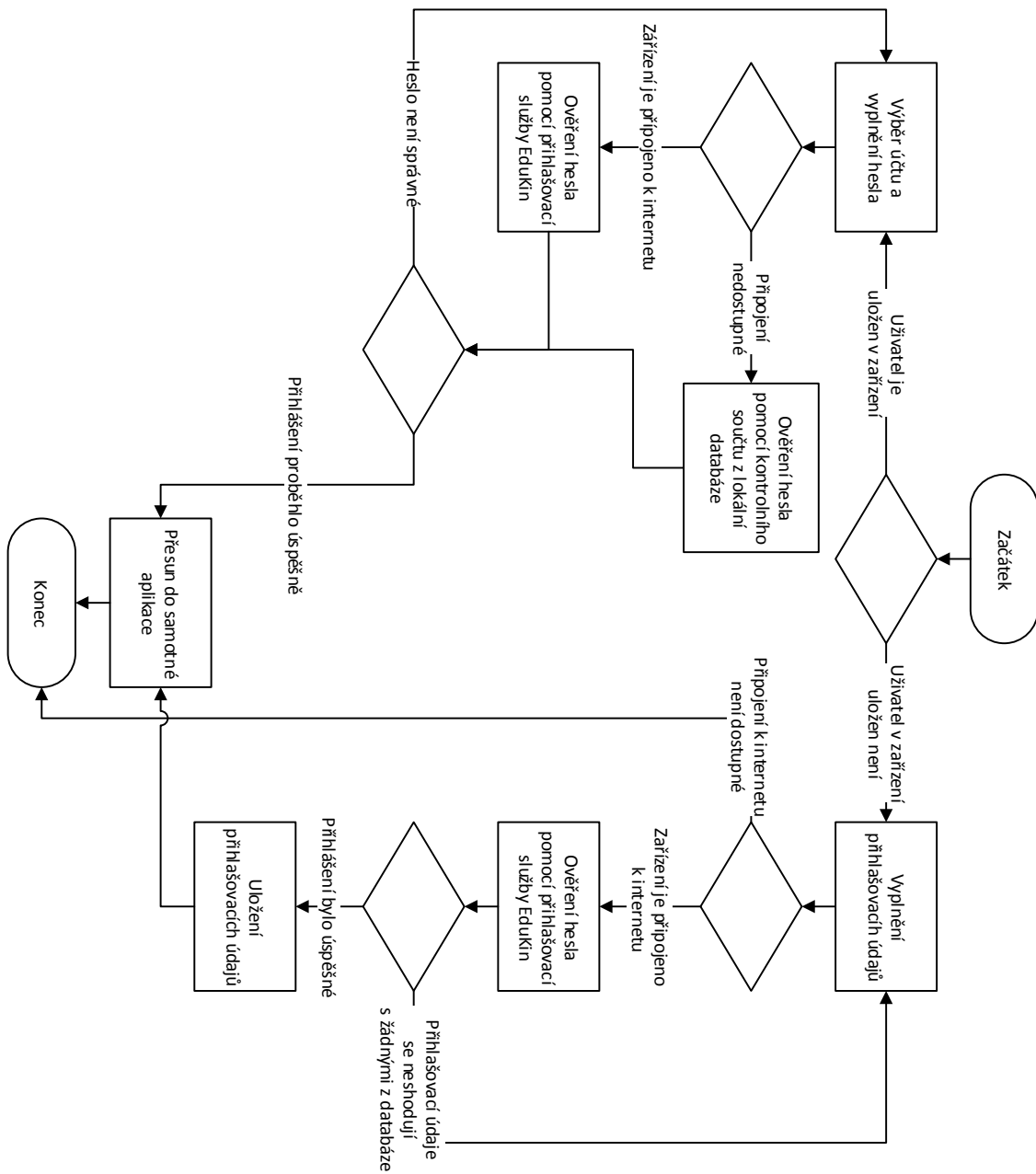
**Rozpoznávání gest** - by měla být část, která zjednoduší vývojářům zachycování gest prstů, jako je přiblížení či tažení nebo rotace.

**Obsluha periférií** - zjednoduší zpracování údajů hlavně z polohového senzoru, s příchodem nových čidel by se tato část rozšiřovala.

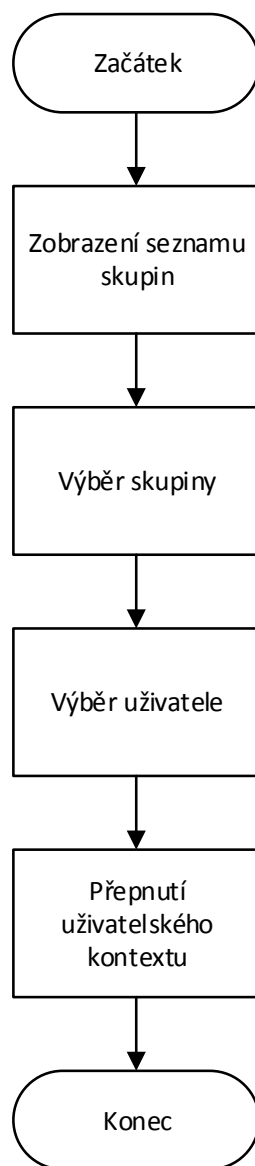
**Pomůcky** - jsou část s různorodými nástroji. Obsahují databázovou vrstvu pro lokální databázi, užitečné metody, například zjištění, zda je zařízení připojeno k internetu či nikoliv, a prvky uživatelského rozhraní, které budou vytvořeny.



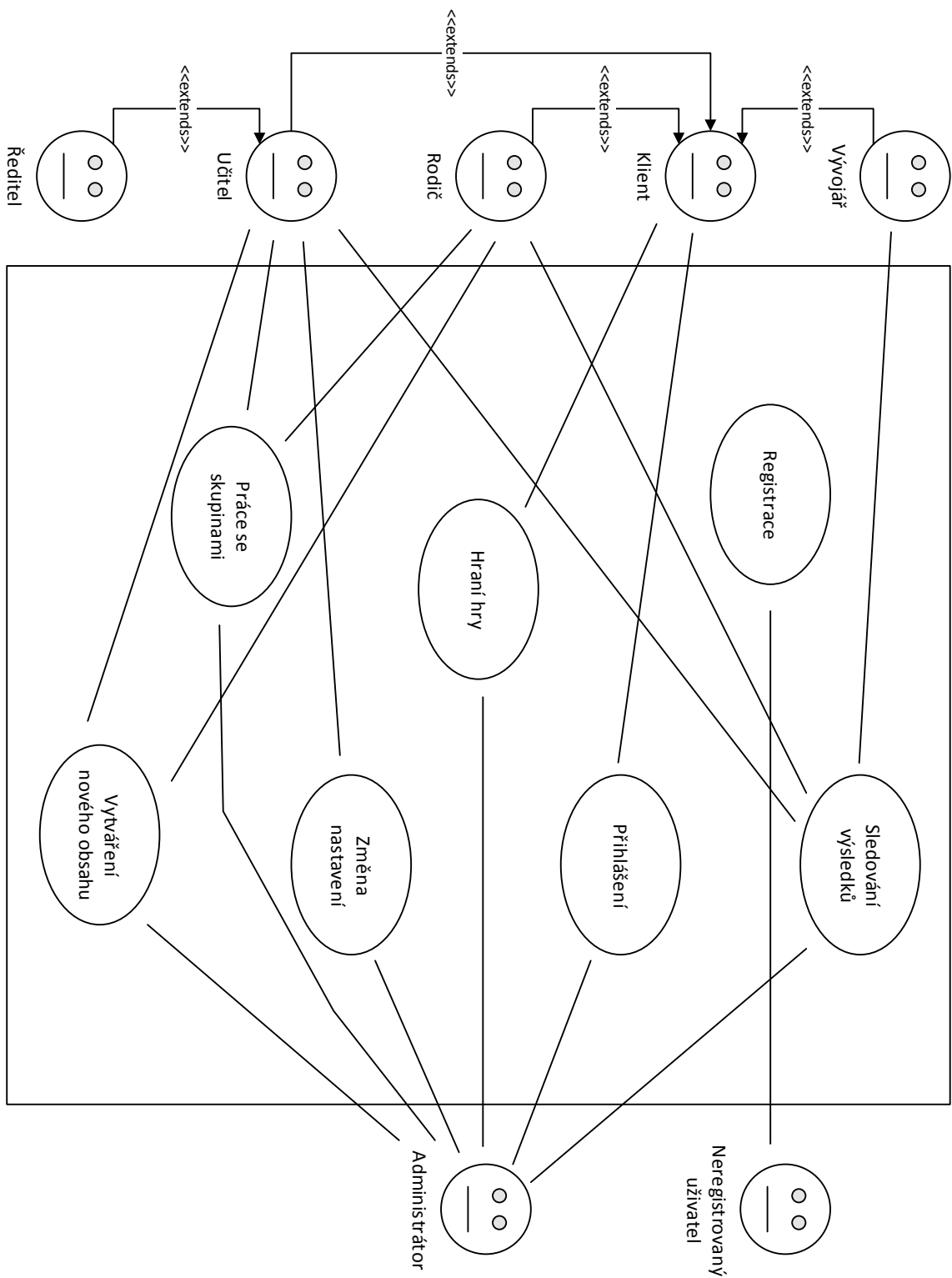
Obrázek 6: Diagram průběhu registrace



Obrázek 7: Diagram průběhu přihlášení

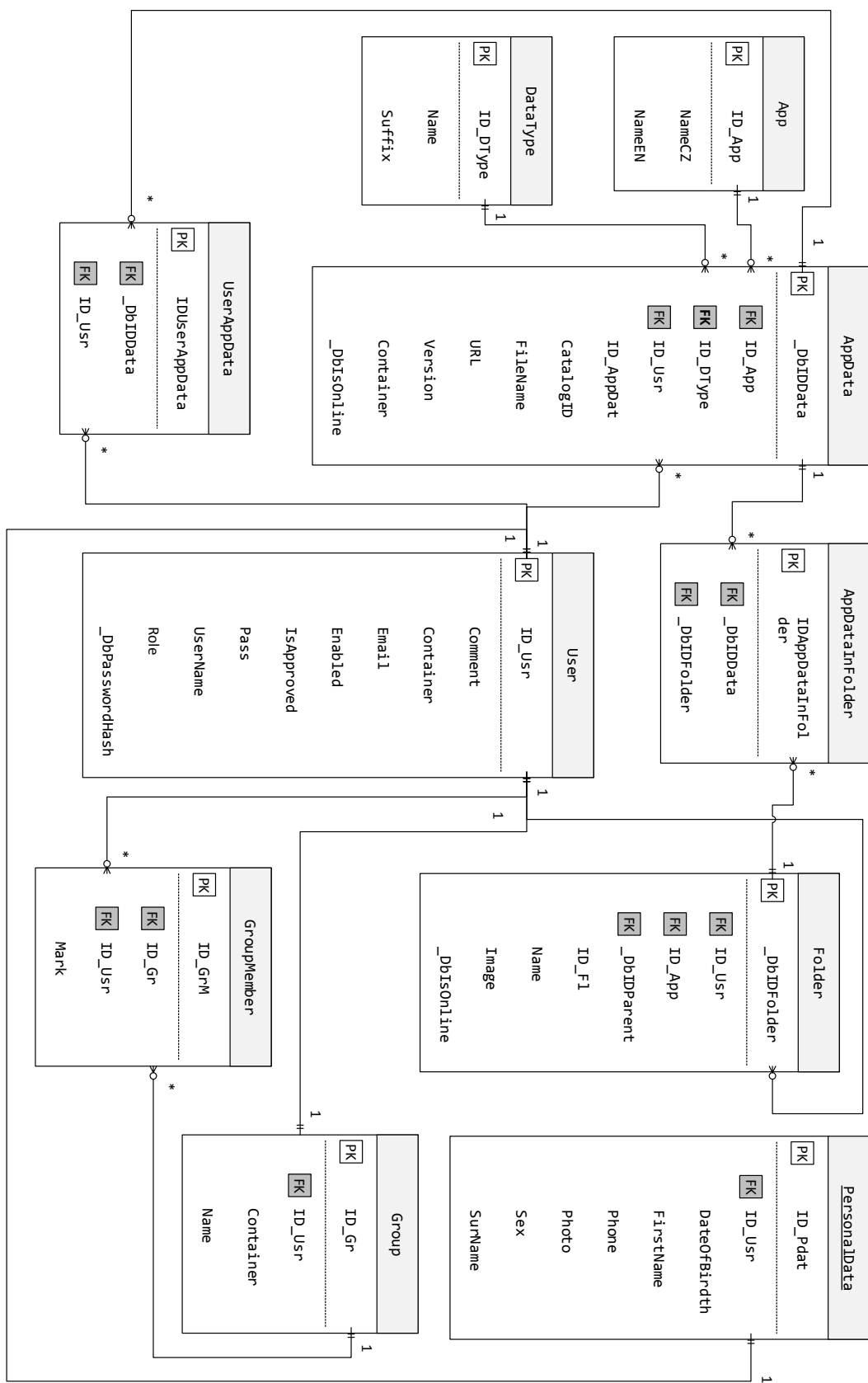


Obrázek 8: Diagram výběru uživatele ze skupiny



Obrázek 9: Hlavní funkce aplikací





Obrázek 10: E-R diagram databáze

## 5 Návrh frameworku

Následující text obsahuje popis řešení komunikace se službami systému EduKin, výběr knihovny pro vykreslování herní grafiky a preferovaný formát dat. Dále detailně popisuje návrh programového rámce, jeho části a jejich propojení.

### 5.1 Komunikace se službami systému EduKin

Serverovou část systému EduKin bylo rozhodnuto provozovat na platformě Windows Azure. Pro tuto platformu jsou primárně určeny služby typu WCF, pro které bylo nutné zjistit, jakými metodami lze s těmito službami komunikovat z Android zařízení.

WCF služby jsou aplikace vytvořené pro běhové prostředí Microsoft .NET Framework, které jsou orientovány na poskytování určité služby skrze aplikační rozhraní. To je definováno pomocí souboru WSDL, který obsahuje hlavičky metod a popis objektů vyměňovaných při komunikaci s klientem či jinou WCF službou. Komunikace v základním nastavení probíhá pomocí SOAP, což je specifikace protokolu pro výměnu strukturovaných dat pro webové služby. SOAP zprávy jsou ve formátu XML a jejich struktura je standardizována. Od verze .NET Frameworku 3.5 je však navíc možné k těmto službám přistupovat jako k REST službám, díky zavedení podpory serializace dat do formátů XML a JSON. Rozdíl je ve klientském volání metod. Ty musí být identifikovány na základě URL.

Jelikož Android SDK neobsahuje podporu pro generování kódu pomocí WSDL souborů, jako je tomu například u .NET Frameworku, bylo třeba vytvořit nebo vybrat existující knihovnu pro serializaci dat a komunikaci s danou službou. Existují tři možnosti jak dosáhnout kýženého výsledku:

1. vytvoření vlastní knihovny;
2. vygenerování kódu pomocí aplikace třetí strany;
3. použití existující knihovny.

V prvním případě by bylo třeba nastudovat dokumentaci standartu SOAP, což by bylo časově náročné a následná implementace serializace, deserializace a komunikace by prakticky znemožnila úvodní rychlé prototypování aplikací pro soutěž Imagine Cup.

Druhý případ by velice ulehčil při tvorbě programového rámce a byl by odolný vůči změnám v implementaci služeb, díky možnosti okamžitého znovu-vygenerování kódů. Bylo nalezeno několik webových a počítačových aplikací umožňující převod z WSLD na kód, bohužel velmi nekvalitních. Většina takto vygenerovaných kódů nešla bez úprav ani sestavit, jiné potřebovaly další knihovny zabírající několik megabytů. Ty, které bylo možné s několika opravami sestavit a spustit, však nefungovaly. Pouze jedna aplikace, pojmenovaná Apache Axis2, dokázala vygenerovat kód, který korektně komunikoval se službami systému EduKin. Bohužel pouze s knihovnamy, které nejsou dostupné v systému Android a navíc vygenerovaný kód měl více než 80 000 řádků.

Třetí možnost vyžadovala nalezení již existující knihovny s tím, že ji bude možné použít i na omezeném souboru knihoven, které platforma Android má oproti standardní

edici Javy. Po nalezení možností, zvážení kvality aplikačního rozhraní a podpory vývo-  
jářů, byla vybrána knihovna Ksoap2-android. Ta sice neumí automatizovaně serializovat  
a deserializovat datové objekty, ale poskytuje poměrně jednoduché rozhraní pro imple-  
mentaci této funkcionality. Její dokumentace sice není příliš kvalitní, ale na portálu, kde  
je umístěn její kód, jsou ukázky potřebných operací.

Po zvážení všech pro a proti a vyzkoušení základních operací, bylo zvoleno využití  
knihovny Ksoap2-android. Tato volba se bohužel později ukázala jako špatná, důvody  
jsou detailně rozepsány v kapitole 7.

## 5.2 Vykreslování herní grafiky

První prototypová hra určená pro soutěž Imagine Cup byla vytvořena pouze pomocí  
nativních Android prvků grafického uživatelského rozhraní. Byly u ní provedeny opti-  
malizace vykreslování, jako například překreslování té části kreslicího plátna, které se  
opravdu změnilo. Ani toto však nepřinášelo naprosto uspokojivé výkonnostní výsledky,  
navíc zde není podpora pro jednoduché vykreslování animací, částicových efektů, 3D  
grafiky a podobně.

Tudíž bylo třeba prozkoumat dostupné knihovny pro tvorbu her. Nalezeno jich bylo  
poměrně velké množství, následují však ty, které jsou opravdu používané:

- AndEngine
  - knihovna s otevřeným zdrojovým kódem
  - většina kódů je napsaná v jazyce Java
  - nízká aktivita vývojáře
  - velké množství jednoduchých návodů
  - podpora vykreslování vektorových obrázků ve formátu SVG
- Libgdx
  - knihovna s otevřeným zdrojovým kódem pod licencí Apache License 2.0
  - zdrojové kódy jsou napsány z třetiny napsány v jazyce C, z třetiny v jazyce  
Java a z třetiny v jazyce C++
  - velmi aktivní vývoj
  - vysoký výkon díky použití nativního kódu
  - aplikační rozhraní pro 3D grafiku
  - možnost kód sestavit pro platformy Android, Java SE, WebGL a iOS
  - komplikovaná tvorba uživatelského rozhraní

- JMonkeyEngine
  - knihovna s otevřeným zdrojovým kódem pod licencí New BSD
  - knihovna vystavuje aplikační rozhraní pro práci v jazyce Java
  - na platformě Android jsou bohužel výsledné aplikace pomalé
  - integrované vývojářské prostředí pro tvorbu scén
- Unity 3D
  - komerční nástroj pro tvorbu her, cena základní licence je \$400
  - možnost skriptovat v jazyce C# (Mono Framework), JavaScript a Boo
  - těžkopádná tvorba uživatelského rozhraní
  - kód lze sestavit pro platformy Android, iOS, Linux, Windows, Mac OS X a pro doplněk do prohlížeče, je však nutné mít patřičné licence
  - propracované vývojářské prostředí pro vytváření scén
  - integrovaný obchod se zdroji jako skripty, 3D modely či audio záznamy

Po nalezení použitelných knihoven bylo třeba určit kritéria, podle kterých budou hodnoceny. Mezi hlavní vlastnosti programového rámce patří rychlost, cena a aktivita vývojářů. Jako sekundární atributy byly určeny schopnost vykreslovat 3D grafiku a možnost sestavení aplikace pro více platform.

Po aplikaci kritérií poměrně přesvědčivě vyšla knihovna Libgdx, protože prakticky splňuje všechny požadavky na knihovnu kladené.

Hry vytvořené pomocí JMonkeyEngine jsou příliš pomalé, navíc byla podpora pro Android přidána nedávno, takže vyladění chyb bude pro vývojáře práce na delší dobu.

Unity 3D je velmi kvalitně zpracovaný nástroj, avšak tomu odpovídá i cena, navíc vývoj pro Mono Framework by znemožnil využívání nativních Android prvků uživatelského rozhraní.

AndEngine obsahuje velké množství předpřipravených komponent, problém ovšem je, že jeho vývojář je již velmi málo aktivní, navíc tento programový rámec nedosahuje takové výkonnosti jako Libgdx viz tabulka 1.

Počet obrázků	50	150	300	500
Android GUI	29 fps	25 fps	11 fps	4 fps
AndEngine	29 fps	27 fps	16 fps	10 fps
Libgdx	30 fps	29 fps	29 fps	24 fps

Tabulka 1: Srovnání vykreslených snímků za vteřinu (fps) s určitým počtem pohybujících se obrázků (sprite), zdroj: [2]

### 5.3 Preferovaný formát dat

Pro vývojáře dalších aplikací, ale i pro vývojáře celého systému EduKin, bylo nutné vybrat vhodný formát pro ukládání dat, jako jsou například tvary v aplikaci malování, které jsou popsány pomocí vektorů. Ten by měl umožnit jednoduchou deserializaci dat na různých platformách a být co možná nejúspornější.

V podstatě šlo o rozhodnutí, zda použít formát XML nebo JSON. Po uvážení výše definovaných požadavků byl zvolen formát JSON, protože nejen že je úspornější, ale navíc existují knihovny pro převod JSON ↔ datové objekty pro každou z pro nás zajímavých platform. Pro platformu Android byla vybrána knihovna od společnosti Google s názvem Gson. Ta se navíc využívá pro serializaci a deserializaci dat při komunikaci se serverem, viz kapitola 7.

Výše zmíněný formát je doporučovaný nicméně není pro vývojáře aplikací závazný, avšak velmi ulehčí práci administrátorům systému i uživatelům, kterým se tak sníží objem přenášených dat.

### 5.4 Rozvržení knihoven

V kapitole 4 je mimo jiné seznam součástí programového rámce i s jejich stručným popisem. Bylo však nutné některé knihovny více rozmělnit.

#### 5.4.1 EdukinModel

EdukinModel je knihovna obsahující veškeré třídy definující datové objekty, které je vhodné vystavovat pro vývojáře aplikací. Tyto datové objekty využívá především databázová vrstva obsažená v EdukinUtils a vrstva pro komunikaci EdukinClient. Všechny knihovny a aplikace, které budou využívat uvedených knihoven EdukinUtils a EdukinClient, budou muset obsahovat odkaz na EdukinModel. Tato knihovna je použitelná i ze standardní edice Java.

#### 5.4.2 EdukinClient

EdukinClient je knihovna obsahující metody pro synchronní komunikaci se službami systému EduKin. Synchronní je z důvodu možnosti použití i v rámci aplikací pro standardní edici jazyka Java. Operační systém Android má totiž vlastní nástroje pro efektivní práci s vlákny a ty nejsou jinak dostupné. Proto v budoucnu, pokud bude třeba naprogramovat klienta pro Java SE, musí být vytvořena knihovna využívající tuto, která bude asynchronně volat její metody.

#### 5.4.3 AndroidEdukinClient

AndroidEdukinClient staví nad knihovnou EdukinClient a umožňuje pomocí svého aplikačního rozhraní volat své metody, jež asynchronně vykonají komunikaci se službami právě pomocí EdukinClient. Tato knihovna využívá platformě závislou třídu systému Android s názvem AsyncTask, proto ji nelze použít na vývoj pro standardní edici Java.

#### 5.4.4 EdukinUtils

EdukinUtils obsahuje nástroje, které mají sloužit pro ulehčení vývoje koncových aplikací. Mezi tyto nástroje patří:

- databázová vrstva,
- nástroj pro asynchronní načítání a cachování obrázků,
- vlastní vytvořené prvky uživatelského rozhraní.

Databázová vrstva by měla být vytvořena tak, aby odstínila vývojáře od převodu výsledku databázového dotazu na instanci třídy z EdukinModel. Vývojář by měl specifikovat dotaz, který chce vykonat nad databází, pouze pomocí volání metod, které postupně interně sestaví dotaz podle jejich vstupů a nakonec vrátí již samotné instance tříd modelu. Aplikační rozhraní této vrstvy by mělo být podobné tomu z Entity Framework, známého z programového rámce .NET.

Protože předškolní děti nebo děti se speciálními potřebami neumí většinou číst, je nutné jim prezentovat data ve formě obrázků. Mobilní zařízení však mají velmi omezenou kapacitu operační paměti, proto je nutné ji efektivně využívat, zvláště pokud pracujeme s bitmapami. Třída ImageLoader slouží pro asynchronní načítání bitmap a jejich cachování. Samotné synchronní načítání či generování bitmapy provádí třídy definované vývojářem, které implementují rozhraní ILoader. Třída ImageLoader je proto odstíněna od závislosti na zdroji dat a tak je zajištěna univerzálnost řešení. Podobné knihovny již existují, ale nebyla nalezena žádná, která by umožňovala načítání dat jak z internetu, tak z vnitřní paměti zařízení.

Mezi vlastní vytvořené prvky uživatelského rozhraní patří ImageFlow, což je třída rozšiřující nativní Android třídu Gallery. Ta je pouze jednoduchá a neobsahuje žádné efekty, jako je 3D natočení či odrazy jednotlivých obrázků. ImageFlow právě toto umožňuje, díky toho je pro uživatele v roli klient výběr zobrazovaných dat jednodušší.

#### 5.4.5 EdukinLogin

Knihovna EdukinLogin obsahuje předpřipravené *activity* pro přidání a přihlášení uživatele, výpis seznamu uživatelů v zařízení a seznam skupin vytvořených daným uživatelem. Tyto *activity* využívají výše popsaných knihoven k přihlášení uživatele, získání informací o něm a jeho skupinách. Dále tato knihovna obsahuje třídy EdukinAccountManager a EdukinGroupsManager, které implementují návrhový vzor singleton. Jejich funkcí je vytvoření business logické vrstvy pro zmíněné *activity* a zároveň pro jednoduchý přístup aplikací k objektové reprezentaci přihlášeného uživatele.

#### 5.4.6 EdukinLibgdxGestures

EdukinLibgdxGestures staví nad Libgdx knihovnou a implementuje její rozhraní InputProcessor, které slouží pro práci se vstupy, jako jsou doteky na displej, pohyb myši či stisk tlačítka klávesnice. Jejím úkolem je zachytávat gesta jako posunutí jedním nebo



dvěma prsty, rotace nebo přiblížení. U těchto pohybů zachytává jejich začátek, průběh a ukončení.

#### **5.4.7 EdukinLibgdxPeripherals**

EdukinLibgdxPeripherals využívá aplikačního rozhraní knihovny Libdgx a poskytuje vlastní pro zachycování změn náklonu zařízení pomocí senzoru akcelerometr.

## 6 Návrh grafického uživatelského rozhraní

Tato kapitola je věnována návrhu grafického uživatelského rozhraní programového rámce a směrnic pro aplikace na něm postavených.

### 6.1 Směrnice návrhu uživatelského rozhraní pro systém EduKin

Pro zachování konzistence uživatelského rozhraní vůči ostatním aplikacím dostupným pro platformu Android, je nutné se z větší části řídit podle směrnic společnosti Google. Ty lze nalézt na webové adrese <http://developer.android.com/design>.

Ty návrhářům do detailu popisují, jak má aplikace pro systém Android vypadat, od velikosti různých ikon pro různě jemné displeje, přes doporučené barvy, fonty, mezery mezi jednotlivými elementy, až po vizuální a hmatovou odezvu.

Velikost displeje tabletu a mobilního telefonu se podstatně liší, proto je důležité, aby vývojáři připravili definice uživatelského rozhraní adekvátně k efektivnímu využití zobrazovací plochy zařízení. Je vhodné připravit tyto definice v souborech XML, které platforma Android podporuje. U některých obrazovek je totiž nutné rozlišovat i mezi sedmi palcovými a deseti palcovými tablety. Systém Android automaticky detekuje jaký soubor s definicí uživatelského rozhraní se má použít, podle toho v jakých složkách jsou tyto definice umístěny.

Vizuální styl aplikací pro systém EduKin by měl reflektovat jejich zaměření a cílovou skupinu.

- Pro děti
  - Vizuální atraktivita, která ovšem neruší, obrázková pozadí obrazovek
  - Jednoduché ovládání, méně možností, vést přímo ke cvičení
  - Při použití tlačítek je nutné přidat ikony, ať je na první pohled jasné, jaká akce bude provedena
- Pro rodiče a učitele
  - Tato část by se měla co nejvíce blížit designovým směrnicím společnosti Google, aby aplikace byly přehledné, designově čisté a barevně střídme
  - Seznamy obrázků by měly být realizovány pomocí nativního Android prvku uživatelského rozhraní GridView
  - Přihlášený uživatel v roli rodič nebo učitel by měl mít možnost z hlavní nabídky aplikace rychle vybrat volbu skupiny, přidání dat a manuálního spuštění synchronizace
  - Pro role rodič a učitel musí být také přístupná možnost nastavení pro jednotlivé klienty v jejich skupinách

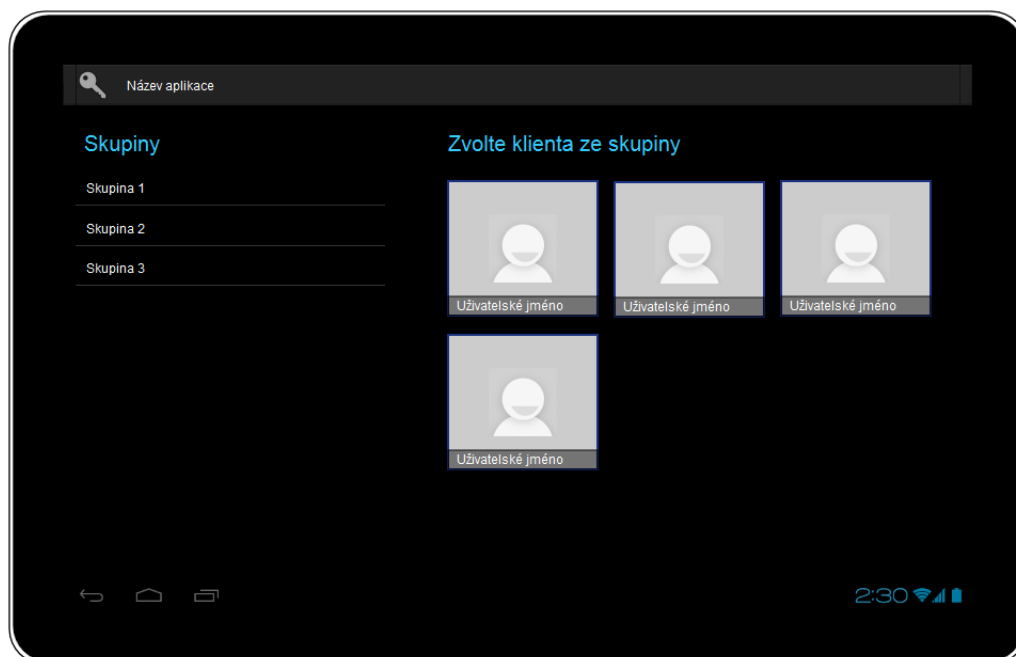
## 6.2 Ukázky návrhů jednotlivých obrazovek

Následuje ukázka vytvořených návrhů uživatelského rozhraní vybraných obrazovek. Tyto návrhy byly vytvořeny pomocí open-source programu Pencil. Na obrázku 11 je zobrazen návrh na rozložení prvků uživatelského rozhraní obrazovky pro výběr účtu.



Obrázek 11: Návrh výběru účtu na mobilním telefonu

Na obrázcích 12 a 13 je předveden návrh uživatelského rozhraní, který efektivně využívá většího displeje tabletu.



Obrázek 12: Návrh výběru skupiny a žáka



Obrázek 13: Návrh výběru skupiny a žáka pro menší displeje

## 7 Implementace a testování

Tato kapitola je věnována popisu průběhu implementace a hlavně problémů a změn, které bylo nutné provést oproti návrhu programového rámce. Dále zde je popis aplikací pomocí něj vytvořených a průběh jejich testování.

### 7.1 Ukázky kódů

Následují dvě ukázky kódů z knihovny EdukinUtils. Výpis kódu 1 je ukázka použití aplikačního rozhraní datové vrstvy, která byla vytvořena tak, aby syntaxí připomínala Entity Framework.

---

```
final LocalDbContext dbContext = new LocalDbContext(mContext);
List<User> activeAccounts = dbContext.Users.include("PersonalData").where(DbHelper.
    USER_IS_APPROVED + " = 1").select();
dbContext.close();
```

---

Výpis 1: Ukázka výběru všech aktivních uživatelů i s jejich osobními údaji pomocí vytvořené databázové vrstvy

Ukázka kódu 2 je použití nástroje ImageLoader pro načtení obrázku, například v seznamu účtů na zařízení. Vytvoření bitmapy provádí synchronně vývojářem vytvořená instance třídy implementující rozhraní ILoader. Vytvořená bitmapa je automaticky přidána do cache pro případné další použití a po té automaticky nastavena předané instanci ImageView.

---

```
final ILoader<User> loader = new ProfilePictureLoader(context);
final ImageLoader<User> mImageLoader = new ImageLoader<User>(loader, true, true);
final User acc = activeAccounts.get(position);
final ImageView avatar = new ImageView(context);

mImageLoader.loadImage(avatar, position, acc); //starts asynchronous image loading
```

---

Výpis 2: Ukázka asynchronního načtení obrázku do ImageView komponenty

### 7.2 Problémy při implementaci

Následující text je věnován problémům s komunikací se službami, tvorbou uživatelského rozhraní pomocí knihovny Libgdx a odesílání dat na server ze zařízení s operačním systémem Android 3.

#### 7.2.1 Problém při komunikaci se službami

Jako největší problém při implementaci programového rámce se ukázala komunikace s WCF službami systému EduKin pomocí protokolu SOAP. Knihovna, která byla vybrána za tímto účelem, sice zvládala volání jednoduchých metod služeb a příjem dat z nich, ale odesílání kompozitních datových typů<sup>3</sup> vedlo na serveru vždy k chybě nebo byla

---

<sup>3</sup>Datový typ, který se skládá z souboru primitivních datových typů nebo dalších kompozitních datových typů.

metodě předána prázdná instance třídy. Situaci navíc ztěžovalo to, že služby používají pro komunikaci datové objekty, které vygeneroval Entity Framework z tabulek databáze. Ty obsahují pro klienta naprosto zbytečné informace a tak zvyšují objem přenesených dat při komunikaci.

Při volbě knihovny pro tento typ komunikace byl brán zřetel, zda tato knihovna má implementováno a opravdu její poněkud slabá dokumentace jasně obsahuje příklady, jak posílat takovýto typ dat [8]. Bohužel až při implementaci bylo zjištěno, že jejich oficiální návody jsou nefunkční z důvodu špatné práce se jmennými prostory, které protokol SOAP využívá. Po čtyřech dnech porovnávání správného tvaru zpráv a toho co generovala Ksoap2-android knihovna a pokusů o získání validní zprávy, bylo rozhodnuto zvolit jiné řešení i za cenu úpravy služeb na Windows Azure.

Řešení spočívalo v použití JSON formátu pro komunikaci. To znamenalo minimální úpravy služeb (přidání koncového bodu, přidání atributu k hlavičkám metod), ale přepsání celé knihovny EdukinClient, protože knihovna Ksoap2-android zajišťovala komunikaci samotnou. Místo ní se začala používat knihovna Gson pro serializaci a deserializaci JSON dat. Toto mělo pozitivní efekt, protože se tak knihovna EdukinClient zmenšila, díky absenci nutnosti vlastního převodu dat a navíc se objem přenášených dat zmenšil.

Bohužel ani toto nebylo finální řešení, protože se po několika testech ukázalo, že služby na volání některých metod vůbec neodpoví. Při jejich testování na lokálním emulátoru Windows Azure metody proběhly, ale stále nic nevracely. Po zapnutí protokolování chyb do souboru bylo zjištěno, že výše zmíněné vygenerované datové typy z Entity Frameworku nelze serializovat do formátu JSON.

Konečným řešením tedy bylo vytvoření jednoduchých datových objektů, které WCF služby umí serializovat do JSON formátu. Pro zachování stávajícího aplikačního rozhraní byly existující služby nezměněny, pouze přibyly nové, které pracují už s novými serializovatelnými datovými typy.

## 7.2.2 Problémy s tvorbou uživatelského rozhraní pomocí Libgdx

Jedním z druhotných požadavků na knihovnu pro tvorbu her byla podpora multiplatformnosti, což Libgdx splňuje. Proto bylo rozhodnuto vytvořit aplikaci kreslení pouze pomocí něj. Problém nastal, když bylo nutné vytvořit složitější formuláře či vlastní prvek uživatelského rozhraní. Systém pro tuto tvorbu je připraven na jednoduché menu pro hry, ale pro vytvoření rozhraní pro rodiče nebo učitele je naprosto nedostačující.

Z tohoto důvodu tedy bylo rozhodnuto všechny knihovny, u kterých to lze, vytvořit tak, ať je lze použít na systému Android i v aplikacích vytvořených pomocí standardní edice Java. Tak lze alespoň některé části znovu-použít v případě, že by vznikala aplikace pro osobní počítače s grafickým prostředím k tomu určeném, jako je například Swing.

## 7.2.3 Problém s odesíláním dat na Android 3

Na tabletech s verzí 3 operačního systému Android odesílání nově vytvořených souborů vedlo k výjimce za běhu aplikace. To je způsobováno tím, že se i přes to, že je provedeno volání příkazu pro nastavení množství odesílaných dat `setFixedLengthStreamingMode`

(délka) na vytvořeném spojení, parametr hlavičky požadavku Content-Length nezmění. To pak vede k tomu, že služba ukončí spojení, protože předpokládá, že přijala už všechna data. Jakmile server ukončí spojení, kód klientské aplikace vyvolá výjimku, že server ukončil předčasně spojení.

Vzhledem k tomu, že identický kód funguje korektně na zařízeních s verzí systému Android 4 a výše, je jasné, že se jedná chybu přímo dané verze systému. Bohužel zatím nebyla nalezena cesta, jak popsané chování obejít.

### 7.3 Aplikace Kreslení

Aplikace Kreslení vznikala původně jako prototyp pro soutěž Imagine Cup po konzultaci v pedagogicko-psychologické poradně. Samotná hra spočívá v tom, že se dítě snaží co nej-  
přesněji obkreslit vybraný tvar. V momentě, kdy procentuálně dosáhne přednastaveného minima, je tvar považován za hotový.

Rodiče a učitelé mají k dispozici možnost tvary pro děti vytvářet, nastavovat obtížnost a procentuální práh zmíněný výše. Také jim tato aplikace umožňuje přístup ke skupinám.

Tvary jsou ukládány v JSON formátu jako soubor bodů. Tento soubor také obsahuje původní velikost plátna, na kterém byl vytvořen, aby jej na zařízeních s jiným rozlišením obrazovky bylo možné korektně zobrazit.

Tato Aplikace původně vznikla za pomoci čistě Android prvků uživatelského rozhraní, ale z důvodu nedostačujícího výkonu byla přepsána za pomoci knihovny Libgdx. Kód aplikace využívá aplikační rozhraní dostupné pouze pro grafické procesory podporující OpenGL 2. Postavička na obrázku 14 slouží dětem jako indikace toho, jak se jim daří.

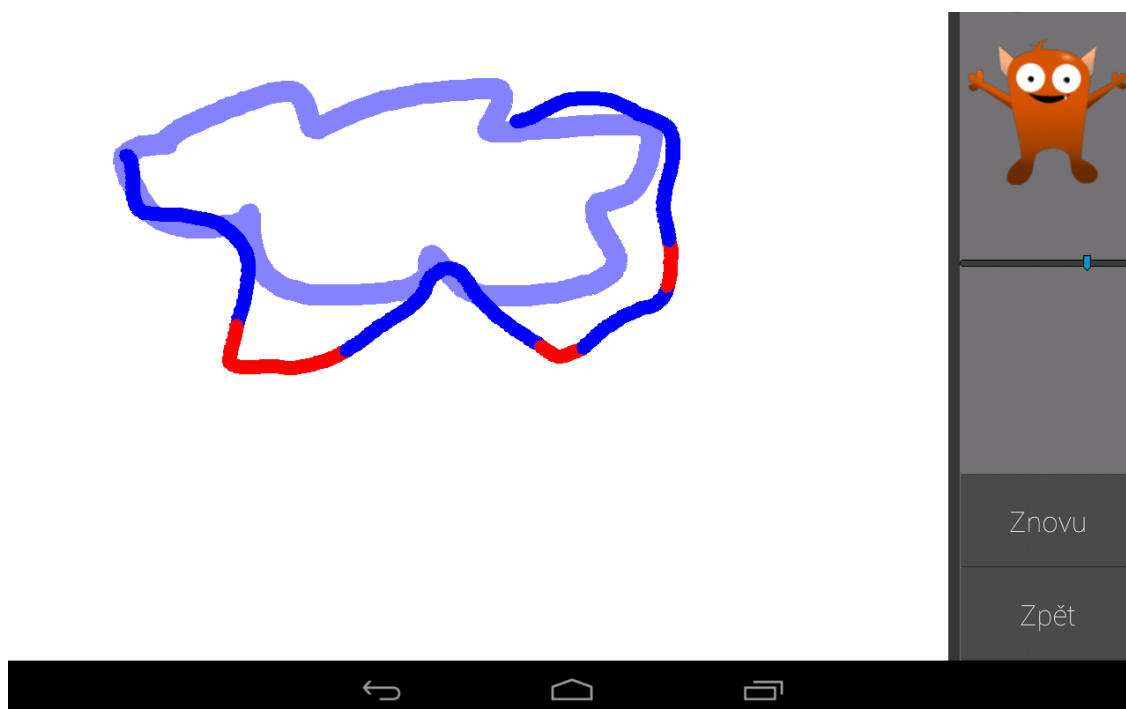
### 7.4 Aplikace Asociace

Aplikace Asociace vznikala za spolupráce se soukromou mateřskou školou pro děti s autismem, jako nástroj pro komunikaci a učení se dorozumívat. To probíhá pomocí obrázkového seznamu asociací, který je stromově členěný do logických celků. Tato struktura je realizovaná pomocí složek, které mají také své obrázky. Po kliknutí na vybraný obrázek je přečten text s ním asociovaný pomocí externí knihovny pro převod textu na řeč, viz obrázek 15.

Aplikace pro všechny role zobrazuje v zásadě stejné rozhraní, ale rodiče a učitelé mají v akční liště možnosti přidávat složky, asociace, zobrazovat skupiny a ručně spouštět synchronizaci.

Samotné soubory asociací jsou prosté obrázky nejlépe ve formátu JPG. K uchování jejich asociovaného textu dostačují souborová metadata, která se používají při komunikaci se službami systému EduKin.





Obrázek 14: Aplikace kreslení

## 7.5 Testování

Testování je velmi důležitá součást vývoje každého softwarového díla a zvláště u programového rámce mohou potencionální vývojáři aplikací chyby odradit od jeho používání. Tento programový rámec a aplikace zmíněné výše byly testovány na různých reálných zařízeních a v emulátoru s různým nastavením. Snaha byla otestovat vše na co největším počtu zařízení, protože si systém Android výrobci zařízení upravují a stalo se například, že tablet od společnosti Samsung neobsahoval barevné schéma prvků uživatelského rozhraní, které na jiných tabletech je. Seznam testovacích zařízení:

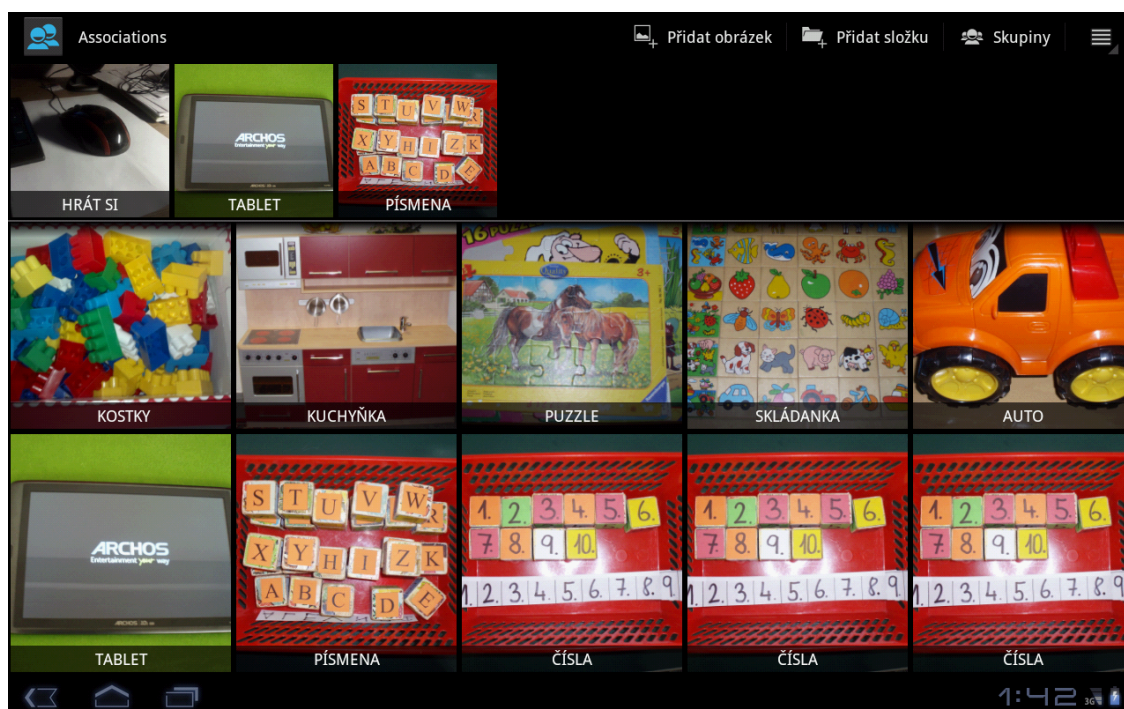
**Samsung Nexus S** je telefon se 4" displejem, jedno-jádrovým procesorem a Android 4.1.

**Samsung Galaxy Nexus** je chytrý telefon s 4,65" displejem, dvou-jádrovým procesorem a s verzí operačního systému Android 4.2.

**Asus Nexus 7** je tablet s rozlišením 1280 na 800 pixelů, má 7" displej, procesor nVidia Tegra 3 a Android 4.2.

**Ainol Novo Elf II** je levnější 7" tablet s dvou-jádrovým procesorem a verzí systému 4.0.4.

**Asus Transformer TF101** je tablet s 10" displejem, procesorem nVida Tegra 2 a systémem Android 4.0, který byl k dispozici ze začátku vývoje.



Obrázek 15: Aplikace asociace

**Archos 101 G9** je levnější 10" tablet s operačním systémem Android ve verzi 4.0.4.

**Samsung Galaxy Tab 10.1** byl zapůjčen společně s tabletem Archos ze soukromé školky. Jde o 10" tablet s verzí Android 3.2.

Součástí testování bylo také sledování objemu přenesených dat a doba trvání synchronizace. Pro aplikaci Asociace však toto nelze odhadovat, protože uživatel na portálu EduKin může nahrát momentálně v podstatě libovolně veliké obrázky, které sledované parametry přímo ovlivňují.

Průměrná velikost jednoho tvaru pro aplikaci kreslení je 5,38 kB, ta se odvíjí od délky čáry, protože s ní lineárně narůstá i počet bodů tvaru. V oblasti s dobrým příjmem signálu je aplikace schopna stáhnout 30 tvarů průměrně do 4,86 vteřin. Jelikož aplikace musí být funkční i bez přístupu na internet jsou stažené soubory uloženy do externí paměti zařízení a znovu se již nestahují.

## 7.6 Konzultace

Od ledna tohoto roku probíhaly poměrně pravidelné návštěvy s konzultacemi aplikací Kreslení a především Asociace. Konzultováno bylo s dvěma učitelkami ze soukromé mateřské školy pro děti s autismem. Ty navrhly několik několik úprav pro aplikaci asociace:

- zobrazování aktuálně vybraného dítě ze skupiny,
- větší písmena u popisů asociací,
- rychle volby ano a ne dostupné odkudkoli nezávisle na složce,
- nějakou formu permanentního přihlášení,
- při zanořování do složek zobrazovat cestu i s obrázky,
- vylepšení čtení jednotlivých asociací tím, že by se nakonec přečetla věta i se skloňováním.

## 8 Závěr

Výsledný programový rámec obsahuje funkcionalitu popsanou ve specifikaci požadavků a byly pomocí něj vytvořeny testovací aplikace. Ty jsou v současnosti stále vylepšovány za pomoci informací a připomínek získaných během konzultací s odborníky.

EduKin jako celek tvoří jedinečný ekosystém, který ostatní aplikace určené pro děti postrádají. Hlavní přínos frameworku je ulehčení práce vývojáře při vývoji pro systém EduKin, díky kterého je naprosto odstíněn od komunikace se službami, tvorby dotazů na lokální databázi, přihlašování či načítání a cachování obrázků. Další přínosná věc je, že vývojář nemusí nutně použít všechny knihovny programového rámce, když například nechce komunikovat se serverem, může použít jen knihovnu na detekci gest.

Aplikace Kreslení i Asociace jsou v současné době ve zkušebním provozu v soukromé mateřské škole pro děti s autismem. Ty denně pracují především s aplikací Asociace, čímž je zajištěn zisk cenné zpětné vazby.

Do budoucna je v plánu zakomponovat do aplikací výše zmíněné připomínky učitelek a graficky je více zatraktivnit. Co se týká programového rámce, je potřeba zajistit opravu nahrávání souborů pro Android 3. Dalším možným vylepšením je přidání mobilního agenta pro automatickou synchronizaci a řešení offline stavu. Po ukončení testovacího provozu by aplikace měly být zpřístupněny dalším institucím i jednotlivcům.

Díky různým schůzkám a konzultacím s odborníky a lidmi z praxe bylo získáno cenné know-how o tvorbě aplikací a her pro osoby se speciálními potřebami. Navíc práce obohatila znalosti o komunikaci s WCF službami, jejich nastavení a možností protokolování chyb. Došlo také k prohloubení znalosti systému Android a k získání základních znalostí k tvorbě aplikací a her pomocí knihovny Libgdx.

## 9 Reference

- [1] ADC I Top 50 Gallery. In: *Google Developers* [online]. [cit. 2013-03-16].  
Dostupné z: [https://developers.google.com/android/adc/adc\\_gallery/](https://developers.google.com/android/adc/adc_gallery/)
- [2] Android sdk vs AndEngine vs libGDX. [online]. [cit. 2013-04-03].  
Dostupné z: <http://hackedprojects.blogspot.cz/2012/03/android-sdk-vs-andengine-vs-libgdx.html>
- [3] Android, the world's most popular mobile platform. In: *Android Developers* [online]. [cit. 2013-03-11].  
Dostupné z: <http://developer.android.com/about/index.html>
- [4] App Framework. In: *Android Developers* [online]. [cit. 2013-03-11].  
Dostupné z: <http://developer.android.com/about/versions/index.html>
- [5] Application Fundamentals. In: *Android Developers* [online]. [cit. 2013-03-11].  
Dostupné z: <http://developer.android.com/guide/components/fundamentals.html>
- [6] BOJKO, Vojtěch. *Infrastruktura založená na Windows Azure pro projekt Edukin*. Ostrava, 2013. Diplomová práce. Vysoká škola báňská - Technická univerzita Ostrava. Vedoucí práce Mgr. Pavla Dráždilová.
- [7] BURNETTE, Ed. *Hello, Android: Introducing Google's Mobile Development Platform*. Pragmatic Bookshelf. ISBN 1-934356-56-5.
- [8] CodingTipsAndTricks. [online]. [cit. 2013-04-09].  
Dostupné z: <http://code.google.com/p/ksoap2-android/wiki/CodingTipsAndTricks>
- [9] MARVAN Filip. *Mobilní operační systém Android*. [online]. [cit. 2013-02-19].  
Dostupné z: <http://diit.cz/clanek/mobilni-operacni-system-android>
- [10] MEIER, Reto. *Professional Android Application Development*. Wrox Programmer to Programmer. ISBN 978-0470344712.
- [11] Overview. In: *Open Handset Alliance* [online]. [cit. 2013-03-16].  
Dostupné z: [http://www.openhandsetalliance.com/oha\\_overview.html](http://www.openhandsetalliance.com/oha_overview.html)
- [12] Tablet PC. In: *POOH.cz* [online]. [cit. 2013-04-23].  
Dostupné z: [http://www.pooch.cz/clanky/tabletPC/tablet\\_pc.asp](http://www.pooch.cz/clanky/tabletPC/tablet_pc.asp)

## **A Obsah přiloženého DVD**

Na přiloženém DVD se nachází následující položky:

- Text diplomové práce ve formátu PDF.