



VÝVOJ ANDROID MOBILNÍ APLIKACE NAD PLATFORMOU UNICORN UNIVERSE

Bakalářská práce

Studijní program: B6209 – Systémové inženýrství a informatika

Studijní obor: 6209R021 – Manažerská informatika

Autor práce: **Vít Apeltauer**

Vedoucí práce: Ing. Vladimíra Zádová, Ph.D.





THE DEVELOPMENT OF ANDROID MOBILE APPLICATION FOR UNICORN UNIVERSE PLATFORM

Bachelor thesis

Study programme: B6209 – System Engineering and Informatics

Study branch: 6209R021 – Managerial Informatics

Author: **Vít Apeltauer**

Supervisor: Ing. Vladimíra Zádová, Ph.D.



TECHNICKÁ UNIVERZITA V LIBERCI
Ekonomická fakulta
Akademický rok: 2014/2015

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Vít Apeltauer**
Osobní číslo: **E12000458**
Studijní program: **B6209 Systémové inženýrství a informatika**
Studijní obor: **Manažerská informatika**
Název tématu: **Vývoj Android mobilní aplikace nad platformou Unicorn Universe**
Zadávající katedra: **Katedra informatiky**

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

1. Charakteristika platformy Unicorn Universe
2. Přístupy k vývoji mobilních aplikací pro Android a jejich vyhodnocení
3. Návrh a realizace mobilní aplikace +4U Album
4. Zhodnocení přínosu vytvořené aplikace

Rozsah grafických prací:

Rozsah pracovní zprávy: **30 normostran**

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

Seznam odborné literatury:

CASTLEDINE, E., M. EFTOS and M. WHEELER. Build mobile websites and apps for smart devices. 1st ed. Collingwood, VIC, Australia: SitePoint, 2011. ISBN 09-870-9084-4.

SCHILD, H. Java 7: výukový kurz. 1. vyd. Brno: Computer Press, 2012. ISBN 978-80-251-3748-2.

UJBÁNYAI, M. Programujeme pro Android. 1. vyd. Praha: Grada, 2012. ISBN 978-80-247-3995-3.

MEIER, R. Professional Android 4 application development. 1st ed. Indianapolis: Wiley, 2012. ISBN 978-111-8262-153.

Elektronická databáze článků ProQuest (knihovna.tul.cz).

Vedoucí bakalářské práce: **Ing. Vladimíra Zádová, Ph.D.**

Katedra informatiky

Konzultant bakalářské práce: **Roman Fiala**

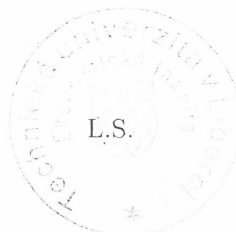
Unicorn a. s.

Datum zadání bakalářské práce: **31. října 2014**

Termín odevzdání bakalářské práce: **7. května 2015**



doc. Ing. Miroslav Žižka, Ph.D.
děkan



doc. Ing. Jan Skrbek, Dr.
vedoucí katedry

V Liberci dne 31. října 2014

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že tištěná verze práce se shoduje s elektronickou verzí, vloženou do IS STAG.

Datum:

Podpis:

Poděkování

Úvodem bych chtěl poděkovat všem, kteří přispěli ke vzniku této bakalářské práce. Především vedoucí mé práce Ing. Vladimíře Zádové, Ph.D. za její odborné vedení a pomoc, kterou mi během psaní poskytovala.

Anotace

Cílem této bakalářské práce je návrh a vývoj Android mobilní aplikace Plus4U Album sloužící pro platformu Unicorn Universe. Účelem samotné aplikace je umožnit svým uživatelům nahrávat obrázky z chytrého zařízení do dokumentů obsažených v informačních systémech postavených na této platformě.

V teoretické části práce představuje digitální stavebnici Unicorn Universe. Rozebírá její části a popisuje její užití. Dále tato práce charakterizuje platformu Android, uvádí rozdělení přístupů k vývoji Android aplikací a provádí jejich vyhodnocení.

Praktická část bakalářské práce se zabývá samotným vývojem Android aplikace pro stavebnici Unicorn Universe. Uvádí zde návrh aplikace, popisuje její vývoj, a zhodnocuje přínos vytvořené aplikace.

Klíčová slova

Unicorn, Unicorn Universe, Android, Plus4U

Annotation

The aim of this thesis is design and development of Android mobile application called Plus4U Album which will serve Unicorn Universe platform. The purpose of the application itself is to allow its users to upload images from smart devices to documents contained in information systems built on the Unicorn Universe platform.

The theoretical part of this thesis introduces Unicorn Universe platform. It analyzes parts of the Unicorn Universe and describes its use. The work then characterizes the Android platform, divides approaches to the development of Android applications and evaluates them.

The practical part of the thesis deals with the development of Plus4U Album Android mobile application for the Unicorn Universe platform. At first, this part describes the application design and its creation, then it evaluates the contribution of the Plus4U Album application.

Key Words

Unicorn, Unicorn Universe, Android, Plus4U

Obsah

Seznam ilustrací a seznam tabulek	10
Seznam zkratk a značek	11
Úvod	12
1. Zhodnocení současného stavu	14
2. Platforma Unicorn Universe.....	16
2.1 Plus4U.....	16
2.2 Unicorn Approach.....	17
2.3 Platforma UU – základní prvky	18
2.3.1 Unicorn Universe Operating System.....	18
2.3.2 Unicorn Universe Process	22
2.3.3 Unicorn Universe Business Modeling Language	25
2.3.4 Unicorn Universe Applications	27
3. Platforma Android, trendy ve vývoji aplikací a jejich vyhodnocení	31
3.1 Požadavky na vývoj aplikací pro Android.....	31
3.2 Komponenty Android aplikace	32
3.3 Přístupy k vývoji aplikací pro Android.....	33
3.3.1 Vývojová prostředí	34
3.3.2 Způsoby programování.....	37
4. Vývoj Android mobilní aplikace Plus4U Album nad platformou UU	42
4.1 Zadání.....	43
4.2 První etapa vývoje aplikace.....	43
4.3 Druhá etapa vývoje aplikace	48
4.4 Vyhodnocení a přínos aplikace Plus4U Album	52
Závěr	55
Seznam použité literatury	57
Seznam příloh	60

Seznam ilustrací a seznam tabulek

Obrázek 1: Základní díly stavebnice UU	21
Obrázek 2: Schéma artefaktu v UU	22
Obrázek 3: Klíčové a podpůrné procesy podniku dle metodiky uuP	24
Obrázek 4: Metamodel jazyka uuBML	26
Obrázek 5: Pět procesů vývoje uuApps.....	27
Obrázek 6: Princip datové komunikace mezi aplikací, systémem a UU.....	28
Obrázek 7: Použité vývojářské nástroje v systémovém a aplikačním vývoji.....	29
Obrázek 8: Příklad uživatelského prostředí aplikace Studijní informační systém	30
Obrázek 9: Odlišnost v zobrazování dvou fragmentů v závislosti na velikosti displeje	39
Obrázek 10: Zobrazení nové galerie na telefonu a tabletu	49
Obrázek 11: Zobrazení vybraných obrázků v AlbumActivity	50
Obrázek 12: Nahrávání obrázků do služby.....	51
Obrázek 13: Dialogová okna informující o výsledku nahrávání.....	52
Tabulka 1: Alternativní vývojová prostředí (IDE) Android aplikací	36
Tabulka 2: Zjednodušená struktura aplikace Plus4U Album v první etapě vývoje	43
Tabulka 3: Zjednodušená struktura aplikace Plus4U Album v druhé etapě vývoje.....	48

Seznam zkratek a značek

ADT	Android Development Tools
BEM	Business Environment Management (podnikové prostředí systému)
BT	Business Territory (informační systém, součástí UU)
EBC	Executive Briefing Centre (vývojové centrum společnosti Unicorn)
HLC	High Level Concept (dokumentace)
IDE	Integrated Development Environment
JDK	Java Development Kit
JDT	Java Development Tools
MT	My Territory (informační systém, součástí UU)
OHA	Open Handset aliance
REST	Representational State Transfer
TGA	Top Gun Academy
TUL	Technická univerzita v Liberci
uuAPI	Unicorn Universe Application Programming Interface (programové rozhraní)
uuApps	Unicorn Universe Applications
uuBML	Unicorn Universe Business Modeling Language
uuOS	Unicorn Universe Operating System
uuP	Unicorn Universe Process
UU	Unicorn Universe
XML	Extensible Markup Language (rozšiřitelný značkovací jazyk)

Úvod

Mobilní zařízení a aplikace pro ně postavené jsou v současné době velmi rozšířené a jejich využití v informačním systému přináší mnoho výhod, z kterých může každá společnost těžit. Vznik mobilních aplikací s přístupem k internetu umožnil on-line propojení mobilního zařízení s informačním systémem. Vzhledem k mobilitě zařízení využívajících technologii Android je schopen uživatel komunikovat se službou prostřednictvím internetového připojení rychle odkudkoliv a kdykoliv, přičemž nemusí využívat těžkých, velkých a špatně přenositelných zařízení. Využívání mobilních aplikací přináší v informačním systému podniku pružnější a rychlejší reakci pracovníků i zákazníků na řešené otázky a problémy. Jejich používání přináší lehčí spolupráci a komunikaci řešitelů. Mobilní aplikace umožňují zaměřit či specializovat svou funkci pouze na určité operace informačního systému. V důsledku toho umožňují uživateli vybrat a provést požadavek rychleji, jednodušeji a uživatelsky přívětivěji.

Pro bakalářskou práci bylo zvoleno téma Vývoj Android mobilní aplikace nad platformou UU, protože problematika vývoje Android aplikací autora této práce zaujala. Pro vývoj aplikace byla vybrána platforma Android, jelikož aplikace na ní vytvořené se těší stále vyšší popularitě [1]. Mobilní zařízení, které tuto platformu využívají jsou také jedny z nejrozšířenějších. Bakalářská práce vznikala v rámci roční řízené praxe ve společnosti Unicorn, proto je aplikace vyvinutá nad platformou UU, kterou společnost využívá ke stavbě konečných informačních systémů, s nimiž pak probíhá spolupráce prostřednictvím internetové služby Plus4U. Aplikace této bakalářské práce bude spolupracovat se službou Plus4U.

Hlavním cílem bakalářské práce je, jak už z tématu práce vyplývá, vytvořit konkrétní Android mobilní aplikaci nad digitální stavebnicí UU prostřednictvím vlastního řešení, tzn. bez poskytnutí firemních podkladů o vývoji mobilních aplikací. Dílčím cílem práce je představit platformu UU společnosti Unicorn, poskytnout náhled na platformu Android a popsat, rozebrat a vyhodnotit přístupy k vytváření mobilních aplikací na platformě.

Obsahem této práce je zhodnocení současného stavu řešené problematiky, které řeší první kapitola. Poté přichází na řadu teoretická část bakalářské práce, která řeší stanovené dílčí

cíle. Je zde vyobrazena technologie UU a Android, s kterou aplikace práce spolupracuje a na níž je postavená. Dále již následuje praktická část práce, která obsahuje postup realizace konkrétní mobilní aplikace, vyhodnocení a přínos výsledku.

1. Zhodnocení současného stavu

Tato kapitola se zabývá prameny, zdroji, odbornými studiemi, publikacemi a aplikacemi, které se souvisí s tématem této práce. Prostudování některých z uvedených zdrojů předcházelo napsání práce a bylo z nich při psaní čerpáno. Většina uvedené literatury je zaměřena na platformu Android a programování mobilních aplikací, protože předmětem této práce je právě vývoj Android mobilní aplikace. Kromě knižních publikací byly významným zdrojem pro tuto práci webové stránky a odborné články zabývající se problematikou platformy Android a zejména internetová služba Plus4U, v jejímž informačním systému mají zaměstnanci společnosti Unicorn přístup k mnoha elektronickým zdrojům a interním informacím o společnosti a o digitální stavebnici UU. Velmi dobrým zdrojem prací, knih, časopisů a článků o platformě Android je také databáze ProQuest, ke které zprostředkovává přístup portál knihovny TUL.

V době zadání bakalářské práce bylo vytvořeno již více než deset Android mobilních aplikací, které propojení a komunikaci s internetovými řešeními postavenými na digitální stavebnici informačních systémů UU také řešili. Autorovi této bakalářské práce podklady k předešlým aplikacím poskytnuty nebyly, protože vedení firmy požadovalo vlastní přístup řešení.

Mimo skupinu společností Unicorn existuje samozřejmě také velké množství obdobných aplikací, které propojení mobilního zařízení s informačním systémem využívají. Podobnou aplikací se například zabýval L. Novák, který sepsal bakalářskou práci na téma *Aplikace na objednávání jídel pro platformu Android*. Současná bakalářská práce se ale zabývá komunikací s rozsáhlým informačním systémem postaveným na stavebnici UU, zaměřuje se na práci se systémovými dokumenty – artefakty které běží na operačním systému Unicorn Universe Operating System a její funkce se zabývá nahráváním obrázků z telefonu do služby Plus4U. Umožňuje tedy i sdílení multimediálních dat uživatelova chytrého telefonu.

Pro platformu Android byla sepsána užitečná kniha od M. Ujbányai. Ten ve své knize s názvem *Programujeme pro Android* uvádí informace o anatomii Android aplikací a o jejich programování. Dále je také význačná literární publikace *Android 4 Průvodce*

programováním mobilních aplikací od A. Granta. V této knižní publikaci lze nalézt velmi mnoho informací o programování aplikací pro operační systém Android, o jejich složení a o práci s jejich komponentami. Dalšími významnými knihami jsou *Build mobile websites and apps for smart devices* od autorů E. Castledine, M. Eftose a M. Wheelera, dále *Professional Android 4 application development* od R. Meiera a ohledně principu programování samotného jazyka Java je kvalitním zdrojem publikace *Java 7: výukový kurz* od H. Schildta.

Kromě literárních publikací ale existují i jiné zdroje. Na internetu se nachází řada článků, výukových tutoriálů a dokonce i videí o programování pro platformu Android, které svou kvalitou dokáží téměř i nahradit zdroje z literárních publikací. Pro samotné programování aplikace této práce jsou kvalitním zdrojem informací k platformě oficiální dokumentace uvedené na portálu *developer.android.com*, a problémy a otázky řešené na webu *stackoverflow.com*.

Ke zpracování druhé kapitoly Platforma Unicorn Universe byly relevantní zdroje, které poskytla společnost Unicorn. Primárně bylo čerpáno z dokumentů společnosti Unicorn Systems, Unicorn Universe, Unicorn Top Gun Academy (TGA) a školy Unicorn College. Veškeré interní dokumenty byly k dispozici online přes službu Plus4U.

Kromě výše uvedených zdrojů, které jsou přístupny spíše pro zaměstnance společností Unicorn, existují odborné příručky, jež obsahují informace o teoretické metodice podnikání nazvané Unicorn Universe Process (dále jen uuP) a jsou velmi dobrými zdroji pro pochopení platformy UU. Jmenují se Unicorn ES Powered Company (což je jiný název pro metodiku uuP) a dělí se na tři díly STRATEGIE, MANAGEMENT a LIDÉ [2]. Tvůrcem příruček je V. Kovář a zaměstnanci skupiny společností Unicorn.

2. Platforma Unicorn Universe

Skupina společností Unicorn se zaměřuje na vývoj informačních systémů a zavádění informační podpory do podniků. Jedním z velmi důležitých aspektů, na které se skupina Unicorn zaměřuje (prostřednictvím společnosti Unicorn Universe), je správa a rozvoj její digitální stavebnice (platformy) UU. [3]

Digitální stavebnice UU vznikla jako systémové řešení k problému s řízením organizace. Staví na praktických zkušenostech, umožňuje rychlý, levný vývoj komplexních informačních systémů, přičemž cílem skupiny společností Unicorn je rozšířit informační systémy na ní postavené do celého světa. Vytvořené informační systémy a řešení na platformě jsou prodávány zákazníkům, nebo slouží interně pro záležitosti společností. Stavebnice v sobě obsahuje několik předem předpřipravených řešení, z kterých se dá v každém podniku, vzhledem k jeho potřebám a požadavkům, postavit kvalitní informační systém. Ten se v případě potřeby dále může rozšiřovat o další „díly“ stavebnice, jež si může zákazník přikoupit či vytvořit prostřednictvím společností Unicorn. [4] Řešení na platformě UU jsou uživatelům (podnikům a jejich zaměstnancům) k dispozici kdykoliv a odkudkoliv při zajištění zabezpečeného přístupu k informacím pomocí internetové služby Plus4U. [3]

2.1 Plus4U

Značka Plus4U označuje internetovou službu (dříve nazývanou také Unicorn Universe), která poskytuje řešení vybudovaná na platformě UU a i další služby, které na UU postavené nejsou, lidem a společnostem. Vybudovanými řešeními je zde myšlen prostor, kde mohou lidé a společnosti virtuálně řídit svoji organizaci, snadno sdílet informace se svým okolím a nabízet své produkty a služby ostatním. Uživatel si pomocí služby Plus4U může ve svém prostoru organizovat nejen své podnikání, ale i své osobní aktivity či jakékoliv jiné aktivity svého života. Obecněji je Plus4U prezentační vrstva informačních systémů postavených na UU a dalších služeb (elektronický obchod, apod.), která umožňuje řadě podniků pečovat o své zákazníky. [5]

Díky této službě mohou uživatelé uchovávat své informace bezpečně s ochranou proti nežádoucímu úniku. Velkou přidanou hodnotou uživatelům je, že mají umožněný přístup ke svým datům odkudkoliv, po dobu 24 hodin denně, 7 dní v týdnu pomocí svého webového prohlížeče [3]. Službu Plus4U provozuje firma Plus4U Net.

2.2 Unicorn Approach

Základní myšlenkou skupiny společností Unicorn je přístup k podniku a podnikání zvaný Unicorn Approach. Jedná se o soupis pravidel, zásad, metod a principů, podle kterých se vytváří teoretická řešení informačních systémů podniků, dle kterých jsou samotné informační systémy postaveny pomocí stavebnice UU. Unicorn Approach se zabývá řízením podniků a organizací v souvislosti s informačním a znalostním managementem, kde znalostním managementem je myšlen systematický přístup k tvorbě, uchování, sdílení a aktivnímu využívání znalostí s určitým cílem [6].

Cílem přístupu Unicorn Approach je úspěšný podnik. Vymezuje dva základní body, kdy lze považovat podnik za úspěšný:

1. Podnik přidává svým zákazníkům dlouhodobě přidanou hodnotu prostřednictvím svých produktů a služeb.
2. Podnik prosperuje (přidává hodnotu sám sobě).

Přístup Unicorn Approach ukazuje sám o sobě teoretické a principiální řešení k problematice v oboru řízení podniku. Vysvětlen je takto:

„Unicorn Approach (Přístup Unicorn) je principiální přístup k podniku a podnikání, ve kterém z Vámi zvoleného portfolia produktů a služeb a s ohledem na Vámi preferovanou dělbu práce, preferované technologie a odhadovaná rizika, odvodíte organizační strukturu, popíšete a předepíšete pracovní postupy jednotlivých podnikových procesů pro všechny zúčastněné role (z řad spolupracovníků, zákazníků, dodavatelů atp.), zdůrazníte klíčové produkty a meziprodukty, pro které rovněž předepíšete strukturu informací a uvedete tento mechanismus v život. Váš podnik se pak stane úspěšným podnikem, a navíc dobrým systémem, který více méně předvídatelně plní účel pro který byl zřízen“. [6 s. 10]

Přístup Unicorn Approach definuje hlavní myšlenky jako je rozdělení pracovního běhu v podniku na lidi, role, činnosti, produkty. Udává potřebu virtualizace podniku – vše podstatné, co se týče podniku, je v systému uloženo a označeno jako *artefakt* (to je jakýsi dokument / nositel informace) nebo je jeho součástí. Dále stanovuje hlavní *aspekty* výroby produktu v podniku. Ty udávají, že produkt má být vyroben v určité kvalitě, kvantitě, ve stanoveném termínu a rozpočtu s použitím příslušných prostředků a know-how. Dále také přístup popisuje různé pohledy jak má vypadat digitální pracovní prostor podniku. Problematiku Unicorn Approach, její návaznost a řešení mohou řešit různé systémy, jedním z nich (nikoliv jediný možný) je právě Unicorn Universe, který z tohoto přístupu vychází, je dle něj postavený. Pro Unicorn Approach je tedy UU systémový řešením či také *nástrojem* pro podrobné řízení.

Přístup k podniku a podnikání se pod názvem Unicorn Approach předává dalším lidem a společnostem jako unikátní know-how ověřené v praxi. Uvnitř skupiny Unicorn se však dopodrobna dále rozděluje na jednotlivé procesy, oblasti a reprezentuje se pod jiným názvem – jako metodika Unicorn Universe Process. [7]

2.3 Platforma UU – základní prvky

Jedním z hlavních *aspektů*, kterými se tato práce zabývá je stavebnice UU. Obsahem stavebnice UU není jen platforma samotná, nýbrž existují celkem čtyři základní prvky, z kterých se UU skládá. Jsou to Unicorn Universe Operation System (zkratkou uuOS), Unicorn Universe Process (zkratkou uuP), Unicorn Universe Applications (zkratkou uuApps) a Unicorn Universe Business Modeling Language (zkratkou uuBML). [5]

2.3.1 Unicorn Universe Operating System

Jedná se o objektový operační systém virtuálního superpočítače. Je to právě ta část, jež je univerzální platformou pro snadný vývoj a provoz komplexních informačních systémů. Systém uuOS poskytuje uživatelům možnost řízení všech svých procesů, usnadňuje správu a sdílení různých informací, zprostředkovává komunikaci, rozvíjí spolupráci mezi lidmi nebo celými organizacemi. Obsahuje více než 700 visual use casů (VUC), které

reprezentují naprogramované vizuální operace jež lze s artefakty (dokumenty) provádět. [4] V UU se zároveň VUC berou jako jednotky, kterými se měří obsáhlost jednotlivých aplikací, informačních řešení. Nemusejí představovat přesně stejnou velikost, ale týkají se jen jednoho procesu, funkčnosti. Nevizuálními naprogramovanými operacemi jsou use casey (zkratkou UC).

Přínosy uuOS [8]

✓ Informace

Operační systém UU umožňuje ukládat prakticky jakékoliv informace (texty, soubory, strukturovaná data), ty je pak možné sdílet a užívat pro řízení podniku. Dál je jen na uživatelích jestli dávají do systému potřebná data, jestliže ano, systém splňuje podmínku obsahovosti každé, pro podnik důležité informace.

Dalšími výhodami ukládání informací je jasné rozlišení, jestli je informace majetkem podniku, či člověka. Zařaditelnost informace jednoznačně do určité organizační jednotky v rámci podniku a složky. Jednoznačné spojení informačních dat se souvisejícími činnostmi (úkol, který je potřeba udělat, je spojen s daty na kterých je potřeba provést). Zjistitelnost zodpovědné role za udržování informace (systém díky tomu nezastarává, zodpovědná osoba informace aktualizuje). Možnost vytvořit mezi informacemi jakékoliv vazby a ty po provázání dále spravovat (v systému tak vzniká virtuální obraz skutečných vztahů v podniku). Rozdělitelnost dat dle podobností, šablon (každá informace vzniká dle nějaké šablony – metaartefaktu, díky tomu stejné procesy a data vypadají stejně, zbývá více času na nestandardní problémy).

✓ Zabezpečení, práva, zodpovědnost

V dnešní době by měl umět každý systém zabezpečovat informace, dávat uživatelům různá přístupová práva, úkoly, kompetence. Operační systém uuOS tuto podmínku splňuje velmi dobře díky tomu, že oproti jiným systémům navíc odděluje zaměstnance jako lidi od jejich know-how pomocí pracovních rolí, stanovuje jim různá práva a zodpovědnost. Každý člověk v systému tedy zastává určitou roli, která mu je přiřazena. Role je oddělena od něj samotného tak, aby ji bylo možné libovolně předávat jiným lidem a aby uživatelé, kteří potřebují hovořit s určitým kompetentním člověkem, komunikovali pomocí jeho role. Role

bude vždy obsazená. Když člověk v ní obsazený práci opustí, vezme si dovolenou, či z jiného důvodu nemůže vykonávat úkony role, jiný zaměstnanec do jeho role nastoupí. Jde o jakousi „hodnotu“ člověka v systému, která určuje povinnosti, práva, zodpovědnost a pracovní prostor zaměstnance. Zároveň jsou každé roli přiřazena práva na informace pro něj potřebná, k nepotřebným informacím nemá přístup. Díky tomu lze dynamicky měnit organizační strukturu podniku.

✓ *Další...*

Velkou výhodou systému je také možnost automatizace běžných uživatelských úkonů prostřednictvím tzv. skriptování (programování nových UC, VUC) a prostřednictvím vytvořeného aplikačního rozhraní Unicorn Universe Application Programming Interface (uuAPI), jež obsahuje proměnné, metody a třídy postavené na uuOS, které lze v kódu volat. Díky této možnosti automatizace lze průběh každého procesu přizpůsobit přesně podle potřeb zákazníka.

V důsledku přidělování nastavených rolí, patřících k organizační struktuře, je i přidělování práv na informace systému v rukou manažerů, nikoliv v rukou IT administrátorů, kteří se v jiných společnostech o práva starají. Podnik by neměl být závislý na jejich správě.

Informační systém postavený na uuOS je vytvořen jako cloudové řešení. A proto se společnosti nemusí zabývat instalací, provozem a údržbou informačního systému, ale mohou ho přímo využívat odkudkoliv a kdykoliv pomocí internetové služby, která se staví k systému vždy jako zprostředkovatel uživatelského přístupu.

Stavební díly UU [4]

Jako každá stavebnice se i uuOS skládá z několika částí či dílů, které dohromady tvoří platformu. Na obrázku lze vidět některé z nich.



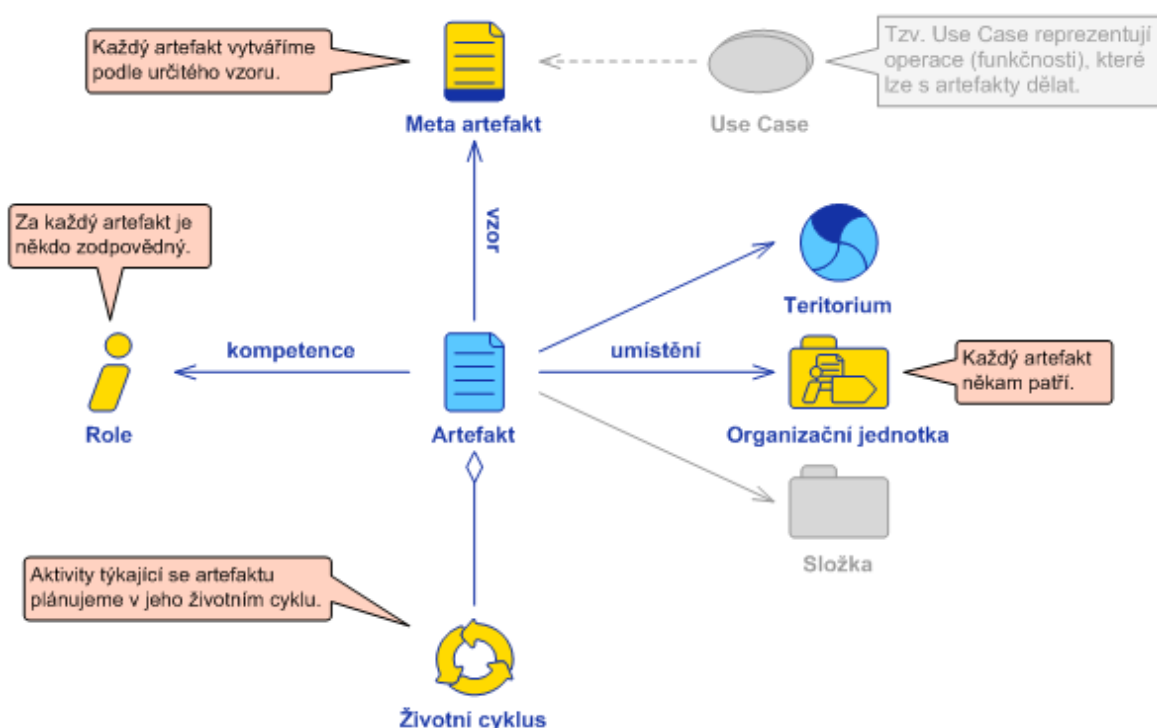
Obrázek 1: Základní díly stavebnice UU
Zdroj: [4], Vybrané komponenty uuOS

Business Territory (BT) a My Territory (MT) představují celé informační systémy uvnitř UU. BT slouží společnostem a organizacím pro řízení jejich aktivit, práce, podnikových procesů. MT slouží k osobnímu užití lidí. Ti v něm mají své vlastní úložné místo, do kterého mohou bezpečně vkládat či v něm vytvářet své dokumenty a pokud chtějí lze je sdílet s ostatními. Zatímco BT organizuje celé společnosti, řídí komunikaci mezi nimi a ukládá jejich informace, v MT si každý člověk může organizovat svůj vlastní osobní život a je k dispozici zdarma.

Další komponentou stavebnice je organizační jednotka. Ta slouží v rámci informačních systémů k jednoznačnému rozdělení podnikových dat do stromové organizační struktury. Patří mezi tzv. organizační artefakty společně se složkou a rolí. V UU je organizační jednotka určena k popisu skutečné organizační jednotky firmy (produkční divize, oddělení, projektu apod.) K řízení slouží organizační jednotka jako určitý ukladatel informací, jsou v něm uspořádány všechny důležité informace. Pro lepší orientaci je možné organizační jednotku dále členit pomocí složek. Základní rozdíl mezi organizační jednotkou a složkou je především v používání přístupových práv. Organizační jednotka rozděluje strukturu dle přístupů, každá „vyšší“ jednotka má přístupová práva k těm „nižším“, které spadají pod ní. Naopak „nižší“ jednotky k těm „vyšším“ přístup nemají. Složka slouží pouze pro rozdělení dat dle podobnosti, spadá pod jednotky.

Za vytvoření a správu struktury je zodpovědná určitá role, jež je kompetentní za celou organizační jednotku (např. Ředitel projektu, divize, streamu, centrální jednotky apod.). Role které se pohybují v rámci organizační jednotky pod ní, spadají také pod tuto „velitelskou“ roli organizační jednotky.

Velmi důležitou částí UU je Artefakt. To je základní nositel informace v Unicorn Universe. Může mít různou podobu podle svého účelu, vždy ale platí, že jeho základní součástí je určitý obsah (uložení informací nějakého typu), životní cyklus (historie práce s artefaktem, ukládání informací o artefaktu) a přístupová práva (ty určují kdo ho může číst, upravovat, mazat). Každý artefakt patří do určitého teritoria, organizační jednotky, složky a je za něj vždy někdo odpovědný (spravuje ho nějaká role). Tento základní nositel informace se také podrobněji skládá z určitých use casů, které reprezentují určité funkčnosti, jež lze nad artefaktem provádět. Struktura artefaktu je názorně zobrazena na obrázku níže.



Obrázek 2: Schéma artefaktu v UU
Zdroj: [9], Základní charakteristiky artefaktu

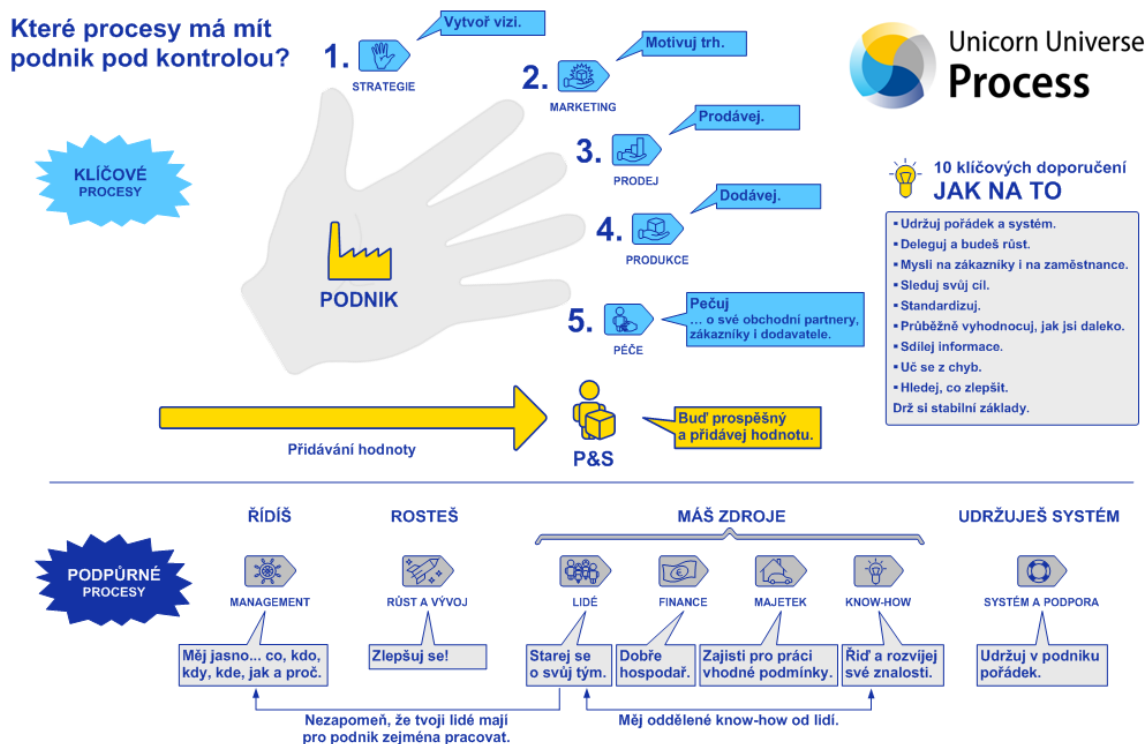
2.3.2 Unicorn Universe Process

Metodika uuP slouží jako nástroj pro řízení organizace. Je to soupis teoretických myšlenek a principů. Přináší výhodu pro jakékoliv podnikání a nabízí návod jak řídit veškeré business procesy. Je navržena jako kolaborační systém (systém podporující spolupráci). Na Unicorn Universe Process Portálu služby Plus4U administrátorka L. Přezvinská uvádí:

„*Unicorn Universe Process je metodika pro řízení všech podnikových procesů s přímou podporou korporátního informačního systému vybudovaného nad platformou Unicorn Universe.*“ [7 s. 1] Podstatou této metodiky je seznam určitých myšlenek, jež společnost Unicorn vytvořila a používá. Jak už bylo řečeno výše, tato metodika se dále mezi lidmi předává jako unikátní know-how ověřené v praxi pod názvem Unicorn Approach.

Výhod má metodika hned několik. Je například použitelná pro celý podnik (vyhledá a nabídne řešení pro jakýkoliv proces v podniku). Podporují ji konkrétní nástroje v UU. Umožňuje rychlou implementaci. Snadno se přizpůsobí každému podnikání. Její procesy se dají standardizovat, tj. zapamatovat a rychleji opakovat, srovnávat mezi sebou, zlepšovat. Nabízí možnost přehledně organizovat podnikové procesy a informace. Sám V. Kovář uvádí, že když věnujeme alespoň pět procent svého času na udržování pořádku, naše celková výkonnost se zvedne až o třicet procent [6].

Metodika uuP rozděluje každý podnik na dvanáct základních procesů, pro něž definuje svoje myšlenky, postupy a sadu principiálních předpisů připravených pro prostředí Plus4U na platformě UU. Tyto procesy jsou na sobě navzájem nezávislé a jejich zavedení v podnikových procesech zlepšuje funkci společnosti a podmínky pro podnikání. Níže na obrázku je vidět jejich rozdělení.



Obrázek 3: Klíčové a podpůrné procesy podniku dle metodiky uuP
Zdroj: [2], Základní schéma pro metodiku Unicorn Universe Process

Klíčové procesy:

- 1) Strategie – hlavní proces vytváření podnikové vize a v návaznosti na ní celkové strategie společnosti
- 2) Marketing – proces motivování zákazníků a trhu prostřednictvím marketingu, marketingového mixu
- 3) Obchod – proces konečného prodeje produktů a služeb zákazníkovi, důležité je využití každé příležitosti k uzavření obchodu
- 4) Produkce – proces výroby, realizace a dodání výrobků a služeb zákazníkovi, dle osvědčených způsobů v praxi
- 5) Péče – proces pečování o vztahy mezi podnikem a jeho zákazníky, dodavateli, distributory a partnery, ukazuje důležitost udržování kontaktů a budování jména

společnosti, podle tohoto procesu v uuP nestačí pouze systémy CRM (Customer Relationship Management), jež se starají jen o zákazníka

Podpůrné procesy:

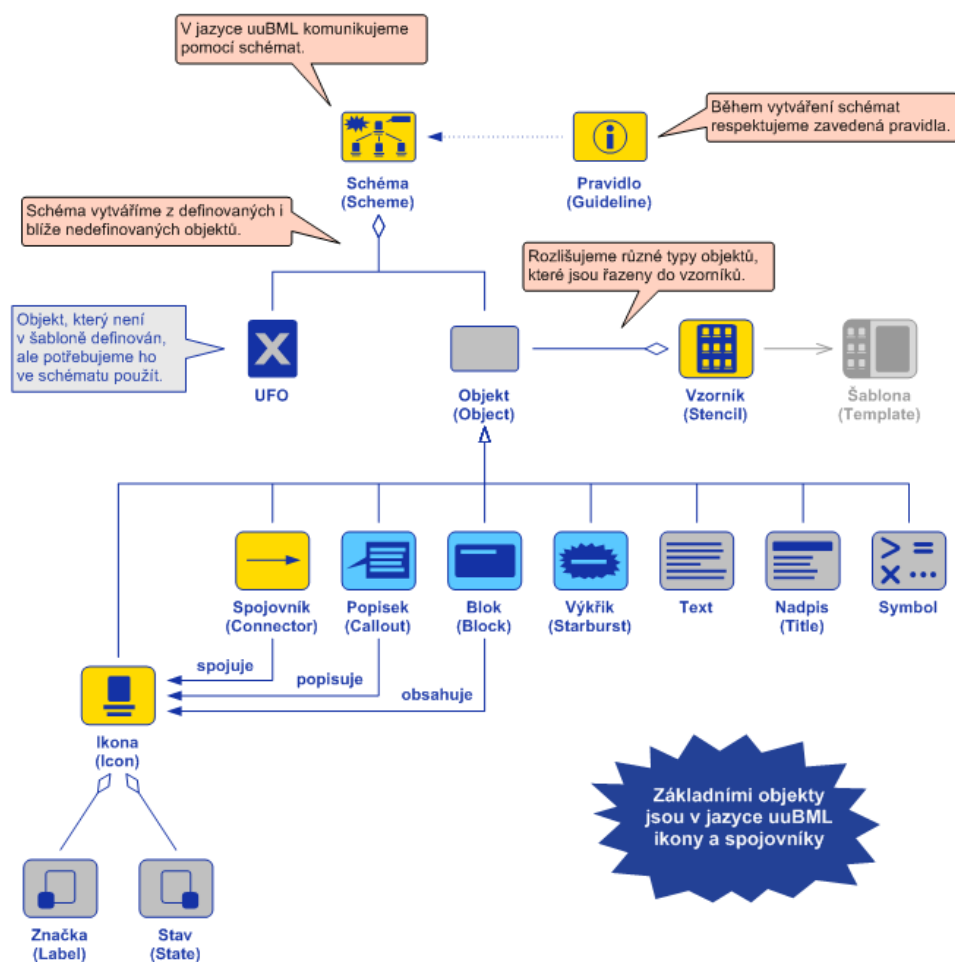
- 6) Management – proces organizace a dělení práce manažerské vrstvy v podniku, zahrnuje jasné stanovení úkolů, pravomocí, výše postavení a zodpovědnosti vůči podnikovým zdrojům
- 7) Růst a vývoj – proces řízení a plánování růstu a vývoje podniku, ukazuje, že každý by měl mít stanovený nějaký cíl a dle něj se vyvíjet
- 8) Lidé – proces péče o zaměstnanecké týmy, zahrnuje organizaci jejich práce, stará se o řízení jejich projektů
- 9) Finance – proces péče o finanční bohatství společnosti, stanovuje uvolňování financí na nákupy, investice, projekty, apod., zabezpečuje získávání financí
- 10) Majetek – proces, který udává kroky k vytváření, pečování, poskytování a prodeje majetku společnosti
- 11) Know-how – proces vytváření know-how a péče o něj, řídí zabezpečení know-how a jeho bezpečné a dopodrobna naplánované uvolňování mezi zaměstnance
- 12) Systém a podpora – proces organizace pořádku ve firmě, péče o podpůrný systém, zahrnuje vytváření a správu podpory pro systém a vzniklé problémy

Klíčové procesy zde vytváří přidanou hodnotu jak pro koncového zákazníka tak i pro firmu, zatímco podpůrné procesy zlepšují podmínky pro fungování podniku. Účelem podpůrných procesů je podpořit procesy klíčové.

2.3.3 Unicorn Universe Business Modeling Language

Je modelovací jazyk, který usnadňuje pochopení problému, jeho rozdělení do částí, zdokumentování a následné řešení. Zprostředkovává kvalitní a srozumitelnou komunikaci

mezi řešiteli problému. Díky své jasnosti a jednoduchosti je snadno pochopitelný i pro lidi, kteří ho vidí poprvé. Vytváření a práce s uuBML nepožaduje žádné speciální znalosti a schopnosti, jazyk má jednoduchý, srozumitelný grafický styl. Jazyk „zjednodušuje“ pohled na situaci svými přehlednými schématy. Základními objekty v jazyku jsou ikony a spojovníky, které modelují objekty, vztahy mezi nimi a jejich chování [9]. Společnost Unicorn ho používá při práci na platformě UU, při práci s metodikou uuP, v technické dokumentaci informačních systémů a jeho částí, a při komunikaci se zákazníky, dodavateli, partnery. Modelování se provádí v prostředí programu Microsoft Office Visio. Na obrázku níže je zobrazen metamodel jazyka uuBML. Lze si zde prohlédnout některé grafické prvky, které se při modelování používají.

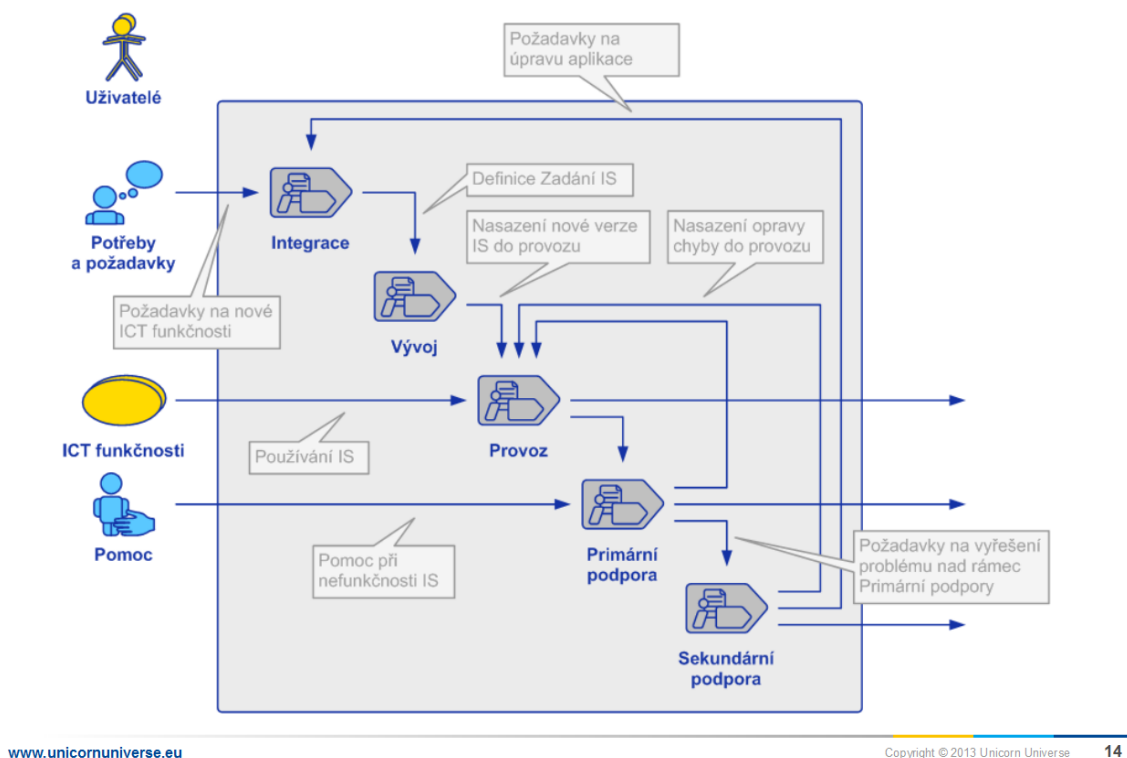


Obrázek 4: Metamodel jazyka uuBML
Zdroj: [9], Metamodel jazyka uuBML

2.3.4 Unicorn Universe Applications

Aplikace uuApps jsou větší součástí platformy UU. Je to soubor mnoha různých informačních řešení již vytvořených na platformě UU nebo na jiných technologiích (např. aplikace této bakalářské práce, která bude spadat do uuApps je postavená na platformě Android). Sada uuApps slouží pro rychlou stavbu informačních systémů skládáním těchto předpřipravených modulů na sebe. [10]

UU - 5 procesů

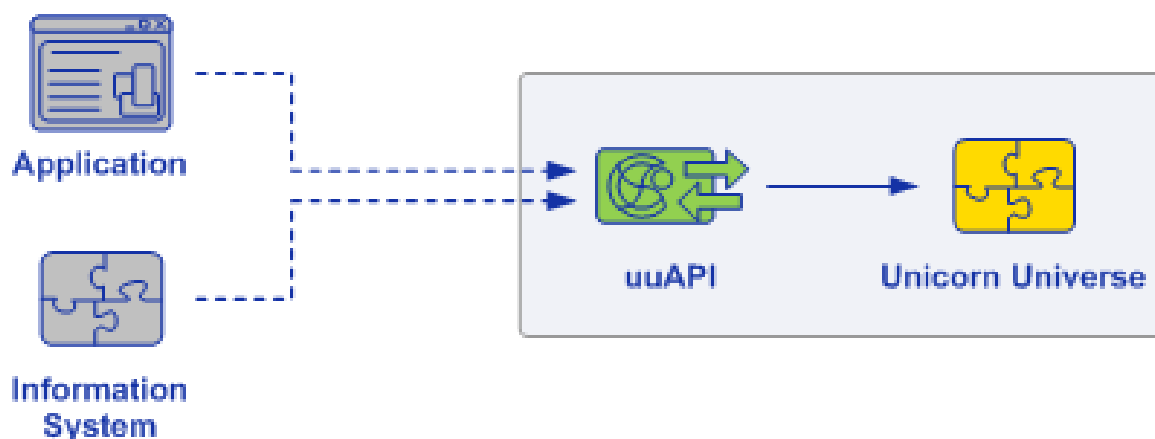


Obrázek 5: Pět procesů vývoje uuApps
Zdroj: [11], Procesy zajišťující funkční IS

Vývoj informačních systémů Unicorn a zavádění aplikací uuApps do informačních systémů se provádí dle pěti stanovených kroků, kterými jsou integrace, vývoj, provoz, primární podpora a sekundární podpora (viz obrázek výše). V prvním kroku se vytváří požadavky na aplikaci, vyjednává se kdo, co, jak, proč bude dělat, apod. Výstupem prvního kroku je zadání k aplikaci. Druhý krok přebírá zadání a začíná samotný vývoj aplikace programátory, vedoucími projektů, manažery. V této části je také aplikace

otestována. Výstupem druhého kroku je nasazení nové verze IS se zabudovanou aplikací do provozu. V třetím kroku nastává provoz otestované aplikace v systému. Čtvrtý krok nastává při drobných chybách v systému, které odstraní primární podpora. Jedná se o lehké problémy jako že tiskárně došel papír, apod. Není-li primární podpora schopna problém vyřešit nastává další krok, sekundární podpora. V něm proběhne řešení složitějšího problému v systému. Kroky primární podpora a sekundární podpora se po ukončení vrací zpět do procesu provozu.

Sada uuApps se v kódu vytváří pomocí příkazů z Unicorn Universe Application Programming Interface (ve zkratce uuAPI), které slouží k datové komunikaci mezi programem a platformou UU. Je to určité rozhraní (interface) určené k programování aplikací na REST-tové architektuře (orientované na volání dat, nikoliv procedur). Obsahuje programátory předem nadefinované procedury, funkce, metody, třídy, atp. pro vytváření skriptů a aplikací pro stavebnici UU. Určuje způsob jakým budou tyto třídy, funkce, atd. volány ze zdrojového kódu programu. Princip volání funkcí UU pomocí přednaprogramovaných příkazů uuAPI je zobrazen na následujícím obrázku.

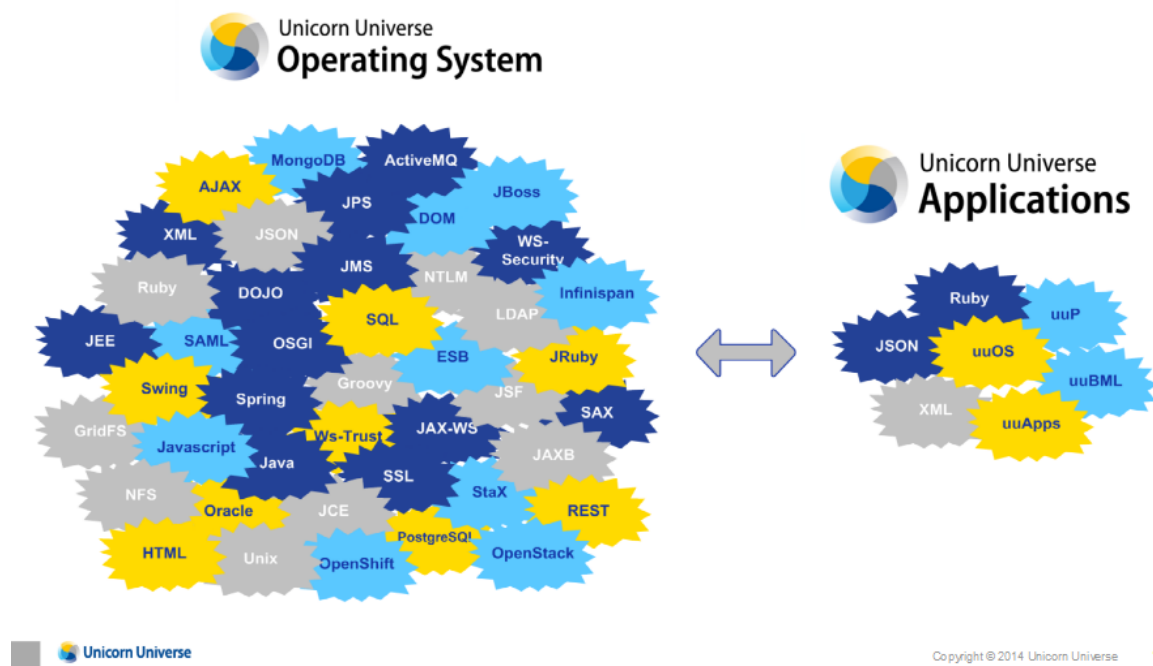


Obrázek 6: Princip datové komunikace mezi aplikací, systémem a UU
Zdroj: interní dokumenty Unicorn Top Gun Academy

Díky uuApps jsou cílová řešení vyvinutá rychleji a snadněji. Vývoji pomocí uuApps se říká aplikační vývoj, naopak vývoji od „základů“ přes platformu uuOS je označeno jako systémový vývoj. Ne vždy je aplikační vývoj možným řešením, ale když ano, je použit

jako výhodný způsob pro zákazníky i řešitele. Na následujícím obrázku můžete názorně vidět rozdíl mezi potřebnými vývojářskými nástroji pro aplikační a systémový vývoj.

Systémový vývoj vs. Aplikační vývoj



Obrázek 7: Použité vývojářské nástroje v systémovém a aplikačním vývoji
Zdroj: [10], Systémový vývoj vs. Aplikační vývoj

Řešení v sadě uuApps je opravdu hodně. Rozdělují se na tři oblasti: Základní, Rozšířené, Řešení na míru. Mezi Základní patří například aplikace uuP Strategy, uuP Management, uuP Marketing, uuP Sales, uuP Finance. Tyto aplikace slouží k správě vstupů do strategie, ukládání strategických dokumentů, správě klientů, živnostníků, zákazníků, komunikace atd. Pomáhají k řízení marketingového mixu, prodeje produktů a služeb a k spoustě dalších jiných funkcí a procedur. Do Rozšířených uuApps patří spíše aplikace, které si zákazník přikupuje ke svému již zavedenému řešení. Patří sem např. Postman, Webman, Finman, Contact Management a mnohé další, které přinášejí další výhody pro zákaznickou společnost. Jejich užití je různé, Postman například slouží k řízení pošty, Finman k správě financí. Řešení na míru se týkají zakázek, které byly vytvářeny na základě speciálního

prání klienta. Nemusí být znovu použitelné v jiných informačních systémech, pro jiné společnosti, ale ta možnost tu je. Do této sady se řadí aplikace jako Řízení cestovní kanceláře, E-learningový systém, Správa rybářského svazu, Řízení personální agentury, Studijní informační systém a spoustu dalších. Jak vypadá uživatelské prostředí Studijního informačního systému v prostředí Plus4U lze vidět na následujícím obrázku. [10]

UNICORN TOPGUN ACADEMY +4U

Metodiky Vývoje Software, Metodiky řízení projektu, RUP

Portál | Plán misí | Hodnocení | Pravidla | Náměty a podněty

Production Division Manager ** Program

Metodiky Vývoje Software, Metodiky řízení projektu, RUP

Základní informace Uprav

Začátek: 16.10.2013 12:00
 Konec: 16.10.2013 18:00
 Místo: EBC C7, 1.5 + VK
 Lektori:
 UED Lector (Šlosar Zbyněk)
 UED Lector (Kodým Roman)
 UED Lector (Buchlák Peter)

Pošli zpětnou vazbu

Agenda

Metodiky vývoje softwaru

- Úvod do metodik v IT a jejich rozdělení:
 - vývojové metodiky,
 - metodiky řízení projektů,
 - metodiky quality assurance,
 - metodiky poskytování IT služeb a organizace práce (ITIL, Cobit)
- Principy vývojových metodik (vodopád, iterace, inkrement, spirála, prototyp, ...)
- Rozdělení na klasické (tradiční metodiky) a agilní metodik, manifest agilního vývoje
- Agilní metodiky
 - Scrum
 - Extreme Programming

Metodiky řízení projektu

- Prince 2
- PMI
- IPMA

RUP

Harmonogram

Čas	Téma
12:00 - 14:00	Metodiky řízení projektu - Zbyněk Šlosar
14:00 - 14:30	Pauza
14:30 - 16:00	Metodiky vývoje softwaru - Peter Buchlák
16:00 - 18:00	RUP - Roman Kodým

Zdroje / studijní materiály

- RUP - Prezentace

We use Plus4U powered by Unicorn Universe.

Obrázek 8: Příklad uživatelského prostředí aplikace Studijní informační systém
 Zdroj: <https://unicornuniverse.eu/cz/uuapps/images/img1.PNG>, [vid. 2015-02-14]

3. Platforma Android, trendy ve vývoji aplikací a jejich vyhodnocení

Roku 2005 společnost Google Inc. získala, akvizicí společnosti Android Inc., základ platformy Android. Koncem roku 2007 byla společností Google založena aliance Open Handset Alliance (OHA), která se skládá z několika desítek podniků z oblasti mobilních operátorů, výrobců software, výrobců polovodičových součástek pro hardware a výrobců mobilních telefonů. Těmito podniky jsou například firmy Intel, HTC, LG, Google, Motorola, T-Mobile, Samsung Electronics atd. Cílem této aliance bylo vytvoření otevřené platformy pro mobilní zařízení a rozvoj mobilních technologií s nižšími náklady na vývoj a distribuci. Otevřená platforma v tomto případě znamená, že by měla být k dispozici každému, kdokoli na ní může vytvářet aplikace, stavět software, číst její zdrojové kódy. Po založení OHA byl představen první produkt, který byl založený na platformě Android a postavený na jádře Linux 2.6. Po tomto kroku stal se Android stěžejním produktem OHA, dostal se na trh a začal boj s konkurencí. [12]

V současné době je již platforma Android vyvinutá na vysoké úrovni. Operační systém Android, který je její součástí, je instalován na mnoha mobilních zařízeních po celém světě. Najdeme ho na chytrých mobilech, tabletech, mobilních či miniaturizovaných počítačích, na moderních televizích a přehrávačích. Existuje názor, že Android brzy bude i v automobilech, letadlech a robotech [13]. Všechna tato zařízení s operačním systémem Android mohou využívat širokou škálu jednoduše ovladatelných a kvalitních aplikací vytvořených na platformě profesionálními i amatérskými vývojáři. Celkově se počítačová platforma, kromě operačního systému, skládá dále z softwarového prostředí (middleware), uživatelského rozhraní a z aplikací. Operační systém Android na mobilních zařízeních je jeden z nejoblíbenějších. [1]

3.1 Požadavky na vývoj aplikací pro Android

Vývoj Android aplikací vyžaduje znalost koncepce objektově orientovaného programování (OOP), programovacího jazyka Java nebo jiného podporujícího jazyka (např. C#), a jazyka XML. Před samotným vývojem je třeba si vybrat vývojové prostředí, tzv. IDE (Integrated

Development Environment) s podporou vývoje pro Android, které slouží programátorovi jako nástroj k programování aplikací. Vybrat si lze například mezi prostředími Eclipse (Googlem podporované), Android Studio, které Google odkoupil nedávno a plánuje ho v budoucnu podporovat [14] a mnoha dalšími.

Dále je třeba seznámit se s principy programování aplikací pro platformu Android a tedy s komponentami Android aplikací. V následující podkapitole jsou pro názornost uvedeny některé z nich.

3.2 Komponenty Android aplikace

Každá Android aplikace se skládá z určitých částí, které je potřeba zvlášť programově obsloužit a provázat mezi sebou komunikací tak, aby aplikace fungovala v pořádku. Jednou z nejdůležitějších částí je tzv. *activity*. Ta představuje prezentační a aplikační část aplikace, je nositelem jedné obrazovky, jež se zobrazuje uživateli. *Activity* obsahuje grafické uživatelské rozhraní, které se většinou vytváří jako XML dokument, dále zpracovává vstupy od uživatele, spravuje procesy, alokuje paměť, přepíná mezi jinými *activity*, volá *Activity Managera*, který řídí její životní cyklus, volá knihovny a provádí všechny důležité kódy, které potřebuje. Každá aplikace s uživatelským rozhraním má vždy hlavní *activity* a popřípadě nějaké vedlejší. Část programu *activity* dědí vždy od třídy *Activity*, která je součástí platformy (je jejím potomkem). Další komponentou a také podobnou součástí *activity* je *service*. Ta na rozdíl od *activity* nemá žádné uživatelské rozhraní, běží pouze jako proces na pozadí a představuje tedy jen aplikační část aplikace. Používá se tam kde nemá interakce s uživatelem zrovna smysl. Většinou se vytváří k nějakým dlouho trvajícím úkonům, jako je třeba přístup k vzdáleným zdrojům, kde je předpokládána dlouhá doba odezvy (připojení k serveru). Ne vždy je potřeba *service* v aplikaci použít. [12]

Pro modelování grafického rozhraní pro *activity* se používají vizuální komponenty nazývané *views*, nebo také *widgety*. Každý *view* je potomkem třídy *View* a je jakýmsi grafickým stavebním prvkem aplikace. *View* existuje několik druhů, patří mezi ně např. *TextView* pro zobrazení textu, *EditTextView* pro umožnění jeho editace, *ButtonView* pro zobrazení tlačítka, *ScrollView* pro umožnění procházení většího obsahu zobrazených dat na

obrazovce a mnoho dalších. Podtřídou *View* je třída *ViewGroup*, která umožňuje „obalit“ několik *view* a zobrazit je uživateli, přičemž nejdůležitějšími z *ViewGroup* jsou skupiny tříd *Layout*. Ty se ve své skupině tříd liší tím, jak rozmisťují své *view*. V projektu se ukládají v souborech typu *.xml* a jsou nejčastějším zdrojem grafického uživatelského rozhraní pro *activity*. Soubory obsahující třídy *Layout* spadají v aplikaci do skupiny souborů zvané *Resources* a jsou datovou částí aplikace. [12]

Resources (přeloženo jako suroviny) jsou zdroje dat pro *activity* i *layout*. Slouží k ukládání a následnému volání různých zdrojů pro aplikaci, jako jsou řetězce znaků, pole, soubory typu *.xml* (*layouty*), obrázky, apod. Patří do datové části aplikace. [15]

Dalšími komponentami, jež se v hojné míře používají v Android aplikacích jsou *Content Provider*, *Intent* a *Broadcast Receiver*. *Content Provider* je určité aplikační rozhraní, které slouží pro sdílení dat mezi různými aplikacemi a různými *activity* částmi. Svými funkcemi dokáže procházet různá data v mobilním zařízení. *Intent* v aplikaci slouží jako aplikační komponenta či také jako rozhraní, používá se k spouštění dalších *activity* částí. Zprvu určí která *activity* se z té stávající má zapnout, pak jí umožní předat nějaká data při jejím zapínání a zapne ji. Umožňuje jinou *activity* také zapnout se zasláním požadavku na navrácení dat. Tato data se při správném výsledku navrátí do prvotní *activity* znovu přes *intent*. *Broadcast Receiver* slouží k „naslouchání“ oznámení, které se šíří v telefonu, a potom na ně nějakým způsobem reaguje. Tato aplikační komponenta může reagovat třeba na změnu připojení k internetu, na dokončení stažení dat, oznámení o nízkém stavu baterie, doručení souboru a tak dále. [12]

Existuje mnoho dalších komponent, z kterých se aplikace může skládat. Komponenty výše uvedené jsou jen ty základní.

3.3 Přístupy k vývoji aplikací pro Android

Tato kapitola rozebírá přístupy k vývoji Android aplikací. Jsou zde rozepsány přístupy a nástroje k vývoji Android aplikací dle dvou hledisek. V první podkapitole je rozebrán přístup k vývoji dle různých vývojových prostředí, které si může programátor vybrat.

V druhé podkapitole je na přístupy brán zcela jiný náhled, jsou zde rozebrány měnící se odlišné způsoby k samotnému programování aplikací.

3.3.1 Vývojová prostředí

Aby mohl programátor vůbec začít s vývojem aplikací pro platformu Android, musí mít nainstalované nějaké podporující vývojové prostředí (IDE). Pro Android jich existuje mnoho, z kterých si vývojář může vybrat. Níže jsou podrobněji rozebrány některé nejznámější z nich.

Jak již bylo zmíněno, jedním ze dvou Googlem podporovaných prostředí je dobře zavedené IDE Eclipse, na kterém musí být nainstalovaný *plugin* ADT (Android Developer Tools), jenž umožňuje vývoj pro platformu Android propojením vývojového prostředí s nástroji Android SDK. V tomto prostředí se programuje v programovacím jazyku Java. Pro úplnou instalaci tohoto prostředí je nutné tedy mít nainstalované Eclipse, dále JDK (Java SE Development Kit), další nástroj pro vývojáře obsahující nástroje a knihovny pro vývoj aplikací pro platformu Java, Android SDK (Software Development Kit) nástroj, který umožňuje vývoj pro různé operační systémy, a také již zmíněný ADT *plugin* [12]. Toto prostředí je zadarmo, je celkem příjemné a přehledné. V současné době však společnost Google doporučuje používat nové prostředí jménem Android Studio, které nabízí nejnovější IDE aktualizace a několik dalších výhod. [16]

Druhým podporovaným prostředím je relativně nové prostředí Android Studio, jež bylo uvedeno k dispozici vývojářům v polovině roku 2013 společností Google. Toto prostředí je a v budoucnu bude Googlem oficiálně podporované a rozvíjené. Android Studio je společným dílem společnosti Google a JetBrains a je zcela zdarma. Založeno je na propracované platformě IntelliJ IDEA [14], díky čemuž nabízí programátorům pomocné funkce jako navigace v kódu, našeptávání metod či celých bloků kódu, *refactoring* (vylepšování vnitřní struktury kódu), analýzu kódu, stavbu a generaci více verzí apk – spustitelných souborů, a obsahuje také různé kódové předpřipravené šablony, bohatý *layoutový* (pro aplikaci vizuální) XML editor s podporou grafických úprav pomocí myši, *buildovací* nástroj Gradle, jenž v sobě obsahuje prvky *buildovacích* nástrojů z jiných IDE (např. Ant, Maven), atp. [17] Prostředí Android Studio vzniklo, aby umožnilo lepší a

pohodlnější vývoj aplikací pro platformu Android. Stejně jako v IDE Eclipse se i v tomto prostředí programuje v jazyce Java, ale oproti Eclipse nabízí Android Studio snazší instalaci, rychlejší a přehlednější IDE, lepší funkce usnadňujících práci programátorům. Z těchto funkcí je například významný snadnější překlad *stringových* (textových) souborů aplikace pomocí tabulek do jiných jazyků (čeština). Prostředí také obsahuje mnohem propracovanější *designový* mód aplikace, který umožňuje procházet náhledy mobilních zařízení v různých rozlišeních najednou a je velmi přesný – po vložení prvku se graficky stane to co programátor očekává. Výhodou Android Studia je také možnost využití Google Cloud Platformy v aplikaci, což dovoluje aplikaci ukládat svá data přímo na internetu. [14]

Kromě dvou výše zmíněných prostředí si ale může vývojář vybrat i taková, která nejsou oficiálně společností Google podporována. Některé jsou k dispozici zdarma, některé nikoliv. Jedním z takových je například prostředí Neatbeans s *pluginem* pro platformu Android. Prostředí Neatbeans je také zadarmo, je příjemné pro vývoj aplikací, ale přesto většina vývojářů na internetu doporučuje používat Eclipse (či Android Studio), protože je podle nich efektivnější [18] a také z důvodu podpory, kdy *pluginy* ADT jsou oficiálně vyvíjeny zaměstnanci Googlu, kdežto *pluginy* pro Neatbeans ne [19]. Je tedy obecně spolehlivější použít Eclipse či Android Studio. Avšak nejdůležitějším měřítkem pro rozhodování, které prostředí využít je určitě cit samotného vývojáře, který by si měl prostředí vyzkoušet a rozhodnout se které mu vyhovuje.

Existují ale také Googlem nepodporovaná IDE prostředí, které svému účelu slouží lépe než Android Studio či Eclipse. To platí například pro prostředí, jež umožňují vývoj v jiných programovacích jazycích, jež někteří vývojáři více uvítají. Nebo také pro prostředí, které disponují možností tzv. *cross-platformového* vývoje (vývoje podporujícího více platform najednou). Ta jsou pro spoustu vývojářů velmi výhodná, určitě jich spoustu uvítá možnost vyvíjení aplikace, která bude použitelná jak na zařízeních Android, tak i na zařízeních s jiným operačním systémem jako iOS, Windows Phone, Blackberry a dalších. Jedním *cross-platformním* prostředím, které se používá i ve společnosti Unicorn System, je například Xamarin. Toto prostředí zadarmo není, ale umožňuje vývoj pro různé platformy, a jako programovací jazyk používá C#. [20] Další alternativní prostředí umožňující vývoj Android aplikací jsou zobrazeny v následující tabulce. V prvním sloupci tabulky je uvedené jméno prostředí, v druhém sloupci je uvedeno jaký programovací jazyk prostředí

podporuje, ve třetím podporuje-li *cross-platformní* vývoj a ve čtvrtém je odkaz na stránku pro více informací o konkrétní platformě.

Tabulka 1: Alternativní vývojová prostředí (IDE) Android aplikací

Jméno platformy	Jazyk	Podpora C-P	Internetová stránka
AIDE (Android IDE)	HTML5/C/C++	Ano	http://www.android-ide.com/
Application Craft	HTML5	Ano	http://www.applicationcraft.com/
Basic4Android	BASIC	Ne	http://www.basic4ppc.com/
Cordova	HTML5	Ano	https://cordova.apache.org/
Corona	Lua	Ano	http://coronalabs.com/
Intel XDK	HTML5	Ano	https://software.intel.com/en-us/html5/tools
IntelliJIDEA	Java	Ne	https://www.jetbrains.com/idea/features/android.html
Kivy	Python	Ano	http://kivy.org/#home
MIT App Inventor	Blocks	Ano	http://appinventor.mit.edu/explore/
Monkey X	BASIC	Ano	http://www.monkeycoder.co.nz/
MonoGame	C#	Ano	http://www.monogame.net/
MoSync	HTML5/C/C++	Ano	http://www.mosync.com/
NS BASIC	BASIC	Ano	https://www.nsbasic.com/
PhoneGap	HTML5	Ano	http://phonegap.com/
RFO Basic	BASIC	Ne	http://laughton.com/basic/
RhoMobile Suite	Ruby	Ano	http://www.motorolasolutions.com/US-EN/Business+Product+and+Services/Software+and+Applications/RhoMobile+Suite
Titanium	JavaScript	Ano	http://www.appcelerator.com/titanium/titanium-sdk/
Xamarin	C#	Ano	http://xamarin.com/

Zdroj: [20], List of Alternative Android App Development IDEs

Vyhodnocení přístupů

Možností v jakém IDE prostředí aplikaci vyvíjet je opravdu mnoho. Výběr platformy se bude určitě v každém případě lišit dle typu samotného uživatele. Záleží samozřejmě na tom, jestli je ochoten si připlatit, či si vybere nějaké prostředí přístupné zdarma. Programátoři, kteří neprogramují v Javě jistě rádi uvítají IDE jako je například AIDE, Kivy, které podporují jiné programovací jazyky. Avšak vzhledem k tomu, že na rozvíjení každého prostředí někdo pracuje, vznikají stále nové *pluginy*, přičemž některé z nich přinášají podporu jiných programovacích jazyků i do prostředí postavených primárně na Javě jako je například Neatbeans [21]. Samotné společnosti vyvíjející mobilní aplikace jistě rádi uvítají spíše nějaké IDE, které umožňuje vývoj pro více platform, jako je

zmíněný Xamarin. Každému programátorovi může také vyhovovat jiné prostředí dle jeho vlastního stylu, jak na něho působí.

Dle autora této bakalářské práce by si každý začínající vývojář měl vyzkoušet nejlépe více prostředí, jestliže si chce dobře vybrat a rozhodnout se mezi nimi. Jako první ovšem doporučuje vyzkoušet samotné Android Studio, vzhledem k jeho podpoře a kvalitě. Toto prostředí je jednoduché, rychlé a nabízí spoustu šikovných funkcí, díky čemuž u něj spousta nových vývojářů hned na začátku jistě zůstane.

3.3.2 Způsoby programování

Další přístupy k vývoji mobilních aplikací pro Android se dají rozdělit v návaznosti na změny samotné platformy. Od začátku vydání první verze Android jsou pravidelně prováděny jeho aktualizace, které opravují chyby, přidávají nové funkce, mění knihovny, aktualizují a přizpůsobují se novým, vylepšujícím a měnícím se mobilním zařízením.

Aktualizacemi Android vznikalo v průběhu let hned několik verzí platformy, přičemž každá změnila více či méně přístup k vývoji aplikací pro Android. Čím větší je rozdíl ve verzích platformy, na které byla aplikace vytvořena v době jejího vydání, tím větší rozdíl je vidět i v jejím zdrojovém kódu, ve způsobu vytvoření a v grafickém prostředí.

Jednotlivé verze platformy byly pojmenovány anglicky podle sladkostí kromě prvotní verze 1.0 a 1.1. Verze 1.5 se nazývá Cupcake, v. 1.6 Donut, v. 2.0 a 2.1 Eclair, v. 2.2 Froyo, v. 2.3 a 2.4 Gingerbread, v. 3.0 až 3.2 Honeycomb, v. 4.0 až 4.0.4 Ice Cream Sandwich [12], v. 4.1 až 4.3 Jelly Bean, v. 4.4 KitKat a jméno nejnovější verze 5.0 a 5.1 je Lollipop. Ke všem těmto verzím vznikly i příslušné API (zkratka pro application programming interface), což jsou sbírky tříd, metod, funkcí, procedur, protokolů, atd. pro programování aplikací nad touto platformou. Sada API začíná úrovní API 1 pro verzi Android 1.0 a končí nejaktuálnější úrovní API 22, k níž je přiřazena nová verze Android 5.1. Číselné hodnoty sady API znázorňují ke kolikáté revizi frameworku API proběhlo. [22]

Aby se v aplikacích se staršími verzemi Android dalo používat některých funkcí z novějších verzí (např. *fragmenty* viz dále), vznikly knihovny tzv. Support Libraries (podporující knihovny). Tyto knihovny zajišťují kompatibilitu různých funkcí novějších verzí platformy Android se staršími verzemi tak, že přidávají potřebné API příkazy i do projektu, který je založen na starší Android verzi, která normálně k těmto příkazům přístup nemá [1]. V kódu se metody těchto podporujících knihoven volají stejně, nebo jen s malými změnami, oproti nově vytvořeným knihovnám, takže programátor s jejich pomocí skoro ani nepozná rozdíl mezi použitými knihovnami. Díky tomuto se sice přístup k vývoji aplikace s novými verzemi mění, ale jako nahrazující minulý ho mohou používat i ti, kteří vyvíjejí pro starší platformy. Pro podporu práce vývojářů a jejich orientace ve změnách existuje také stránka <http://developer.android.com.>, která obsahuje dokumentaci k platformě Android. Jsou na ní uvedeny i příklady, jaké použít funkce a kódy, nejde-li pro nějakou určitou potřebu aplikace podporující knihovny použít.

Fragmentové a nefragmentové přístupy k vývoji mobilních aplikací

V návaznosti na změny platformy Android, vzniku stále novějších mobilních technologií a API, vyzpozoval a rozdělil autor této práce chronologické změny přístupů k vývoji Android aplikací na dva přístupy a těmi jsou *fragmentový* a *nefragmentový*.

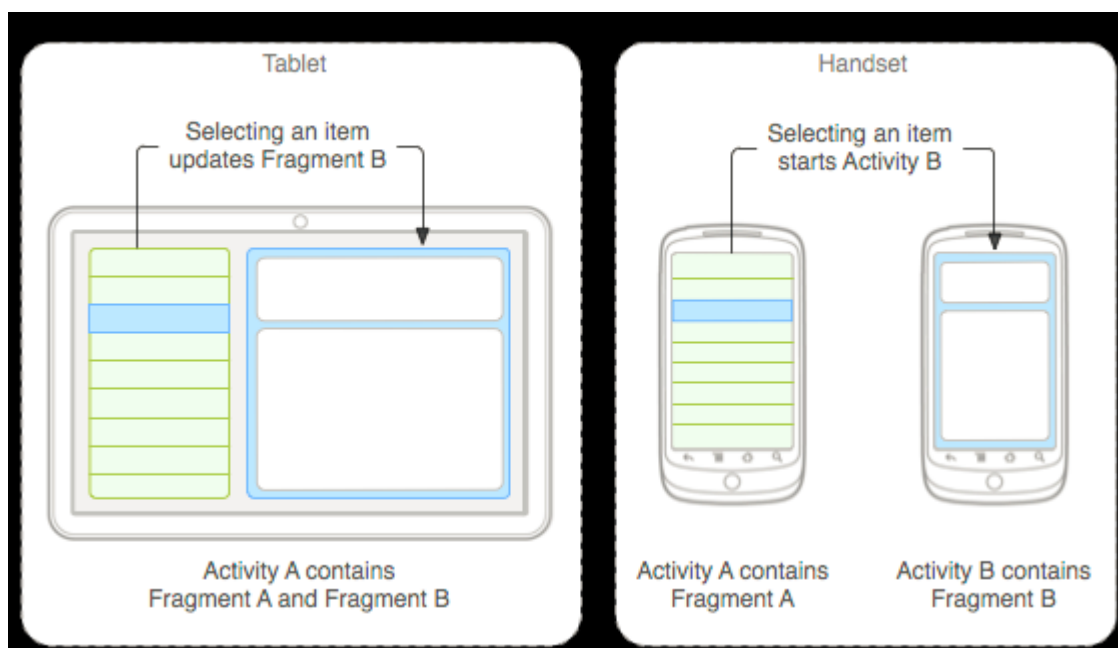
Fragmentový přístup

Přístup k vývoji aplikací pomocí *fragmentů* je trochu složitější než *nefragmentový*. Vznikl na základě vývoje nových mobilních zařízení, větších než jsou tzv. chytré telefony, jako například tablety a miniaturní počítače. Větší zařízení mají větší obrazovku a zjistilo se, že aplikace, která vyhlížela dobře na malé obrazovce mobilu, již na větším displeji dobře nevypadala. Z toho důvodu vznikly s nástupem Android verze 3.0 (API level 11) nové části aplikace *fragmenty*. [1]

Fragmenty jsou programové části aplikace, jež umožňují uchovat blok programátorského kódu. Umožňují nést části uživatelského rozhraní, které určují chování či vzhled aplikace. Tyto kousky kódu se pak dají umisťovat uvnitř bloku *activity* (prezentační vrstva aplikace, viz úvod kapitoly Vývoj aplikací pro Android). *Fragment* má definován svůj vlastní životní cyklus, jenž je jen trochu odlišný od životního cyklu *activity*. Tento životní cyklus *fragmentu* je ale přímo závislý na živ. cyklu *activity*, která ho obsahuje. Proto když je

activity pozastavena, je pozastaven i *fragment*. Když je *activity* zničena, je zničen i *fragment*. [1]

Každé *activity* je umožněno obsahovat hned několik *fragmentů*, přičemž může měnit jejich zobrazování, umístování, skrývat je nebo naopak ukazovat do popředí. Díky těmto možnostem se *fragmenty* dají používat pro odlišné zobrazování uživatelského rozhraní aplikace, či ke změnám chování *activity*, vzhledem k různým podmínkám které mohou na zařízení nastat. Uvádí se třeba příklad, že za předpokladu spuštění aplikace na velkém displeji se zobrazí uvnitř *activity* dva *fragmenty* nesoucí různé grafické uživatelské rozhraní vedle sebe, místo toho aby každému vytvářela zvlášť *activity* a přepínala je mezi sebou. Naopak na malém displeji by dvě *activity* pro *fragmenty* vytvořila a nechala uživatele tlačítkem přepínat mezi nimi, protože v jedné obrazovce by se tak velký obsah zobrazit nedal, nebo by provádění tolika úkonů v jedné aktivitě nebylo zcela efektivní. [23] Na následujícím obrázku je tento princip zobrazen.



Obrázek 9: Odlišnost v zobrazování dvou *fragmentů* v závislosti na velikosti displeje
Zdroj: <http://developer.android.com/images/fundamentals/fragments.png>, [vid. 2015-02-14]

Z toho plyne opravdu mnoho výhod, které může programátor využít při vytváření programu.

Nefragmentový přístup

Tento přístup je starší než výše zmíněný a používá se již od začátku. Jeho koncepce spočívá v používání součástí *activity* bez vytváření *fragmentů* jako nositele hlavního kódu. V nynější době se stále používá hodně, ale spíše pro aplikace určené pro menší displeje, tedy mobilní telefony. Někteří vývojáři ho ale stále používají i pro aplikace k větším displejům, vzhledem k tomu, že je jednodušší, nemusí pro *activity* vytvářet *fragments* a řešit jejich chování, nemusí se také zabývat prací s *Fragment Managerem*. Vývoj aplikací s *nefragmentovým* přístupem je obecně pohodlnější, nemusí být ale uživateli větších zařízení tak přijímán.

Vyhodnocení přístupů

Přístupy k vývoji aplikací pro platformu Android se stále mění. To je sice výhodou pro Google, OHA, jejich motivaci trhů, a také pro nové příležitosti a rozvoj této technologie, ale pro vývojáře to znamená, že aby udrželi krok s dobou, musí se stále učit. Je nutné sledovat vývoj platformy a zabývat se studii nové dokumentace k novým a novým verzím stavebnice Android. Nutno je ale také podotknout, že o žádné razantní změny nejde. Důležité je, že platforma se mění spíše okrajově a její vnitřní základ (Linuxové jádro) zůstává téměř stejný, verze vyvíjející se platformy spíše přidávají nové funkce či vylepšují stávající funkce, než aby je měnili a odstraňovali, a tím programátorovi ničili jeho stávající práci.

Ohledně výběru *fragmentových* a *nefragmentových* přístupů, je na každém programátorovi, který bude používat. Jestliže je ale aplikace vyvíjena pouze pro větší zařízení, nebo potřebuje nějakou speciální funkci (například zobrazování dialogů uvnitř *service*, apod.), měla by používat *fragments*.

Fragmentový přístup je samozřejmě obecně vhodnější pro dnešní zařízení. Jeho nevýhodou je o něco složitější vytváření a struktura aplikace, na druhou stranu ale přináší spoustu výhod, které umožňují zefektivnit *image* aplikace, vylepšit její chování a zpřístupnit jí mnoho nových možností. Přizpůsobivost aplikace k různým zařízením pomocí *fragmentů* je určitě velkou výhodou na trhu mobilních aplikací.

Jak už plyne z nevýhody *fragmentového* přístupu, výhodou *nefragmentového* je jeho jednoduchost, a v návaznosti na to větší rychlost implementace. Kód aplikace je v tomto případě čitelnější a programátor se nemusí starat o práci s *Fragment Managerem*. Nevýhodou je, jak již plyne z předešlého odstavce, zaostávající grafické uživatelské prostředí aplikace na větších zařízeních a nemožnost použití nějakých speciálních chování aplikace, které by v jeho aplikaci mohly být efektivní.

Dle názoru autora této bakalářské práce by měla aplikace sloužit jak na malých tak na velkých zařízeních a uspokojovat grafickým prostředím uživatele. Nejlépe by měla být téměř každá aplikace vyvíjena *fragmentovým* přístupem, aby žádného uživatele, který by si ji chtěl zobrazit na tabletu, nepřekvapila její grafická nedostatečnost. Pokud je však aplikace určena spíše pro malé displeje a uživatelské rozhraní je navrženo dobře, může být postavena i bez *fragmentů*.

4. Vývoj Android mobilní aplikace Plus4U Album nad platformou UU

Tato kapitola se zabývá praktickou částí bakalářské práce, je zde rozepsáno zadání, postup vypracování a vyhodnocení přínosu Android mobilní aplikace Plus4U Album nad platformou UU. Aplikace je určena mobilním zařízením s operačním systémem Android a svou funkcí poskytuje multimediální data (obrázky) mobilních zařízení informačním systémům postaveným na digitální stavebnici informačních systémů UU. Konkrétně aplikace umožňuje nahrávat obrázky obsažené v mobilních zařízeních do služby Plus4U postavené na platformě UU. Sama o sobě aplikace postavena na stavebnici UU není, ale svou funkcí pro ní slouží a je její součástí jako jeden prvek ze sady Unicorn Universe Applications, což jsou vyvinutá informační řešení, které se berou jako součást platformy UU, a jež slouží k rozšiřování funkcí informačních systémů postavených na stavebnici UU.

Vývoj aplikace nad stavebnicí UU znázorňuje, jak se dá stavebnice UU rozvinout nejen prostřednictvím vývoje aplikací z jejích vlastních „dílů“, ale také jak se dá její funkce lehce rozšířit spoluprací s jinou platformou, jinou technologií, jako je Android.

Při vytváření aplikace byla součinnost platformy Android s platformou UU umožněna díky již existujícímu aplikačnímu programovému rozhraní Unicorn Universe Application Programming Interface (uuAPI), které je postavené nad platformou UU, a které umožňuje volat a vytvářet příkazy nad stavebními prvky stavebnice UU. Příkazy uuAPI jsou určeny pro vývoj aplikací uuApps a jsou zpracované pro využití v programovacím jazyku Ruby, kdežto mobilní aplikace Plus4U Album je postavena za pomoci programovacího jazyka Java. Z toho důvodu je v aplikaci Plus4U Album použita knihovna, která umožňuje snadné volání uuAPI příkazů i pro tuto aplikaci. Použitá knihovna vznikla v jednotce iCom Vision společnosti Unicorn Systems a byla poskytnuta k dispozici pro řešení této aplikace.

Aplikace této bakalářské práce byla vypracována ve vývojovém prostředí Android Studio a pro její vývoj byl použit převážně *nefragmentový* přístup (viz kapitola 3.3.2 Způsoby programování), z důvodu jeho menší časové náročnosti a složitosti. Fragment byl použit pouze v jediném případě, jako výjimka při potřebě správného zobrazení výsledného dialogu (viz popis dále). Je třeba zde uvést, že aplikace je prototyp a v jednotce iCom

Vision ještě stále nebyla testována. Je otestována pouze řešitelem projektu (autorem) a její otestování, doladění a uvedení do provozu bude teprve provedeno za pomoci podniku, ve kterém byla vytvořena.

4.1 Zadání

Autorovi této práce byl zadán úkol vytvořit aplikaci, která propojí mobilní zařízení, na bázi operačního systému Android, s informačním systémem UU, a která umožní nahrávání obrázků z mobilního telefonu do obsahu artefaktu (základního nositele informace v UU). Toto zadání bylo k první etapě vývoje aplikace, v dalších etapách byly poskytnuty návrhy na vylepšení aplikace, jinak ale byli její modifikace ponechány na uvážení řešitele projektu.

4.2 První etapa vývoje aplikace

Zadání aplikace bylo řešeno a vypracováno postupně, dle potřeby. Tato kapitola popisuje vznik již fungující aplikace, dle chronologického zpracování a postupu její funkčnosti. Na začátek této kapitoly je v Tabulce 2 uvedena zjednodušená struktura aplikace pro lepší přehlednost. Níže lze vidět třídy, *activity* a *layouty* aplikace Plus4U Album v první etapě jejího vývoje.

Tabulka 2: Zjednodušená struktura aplikace Plus4U Album v první etapě vývoje

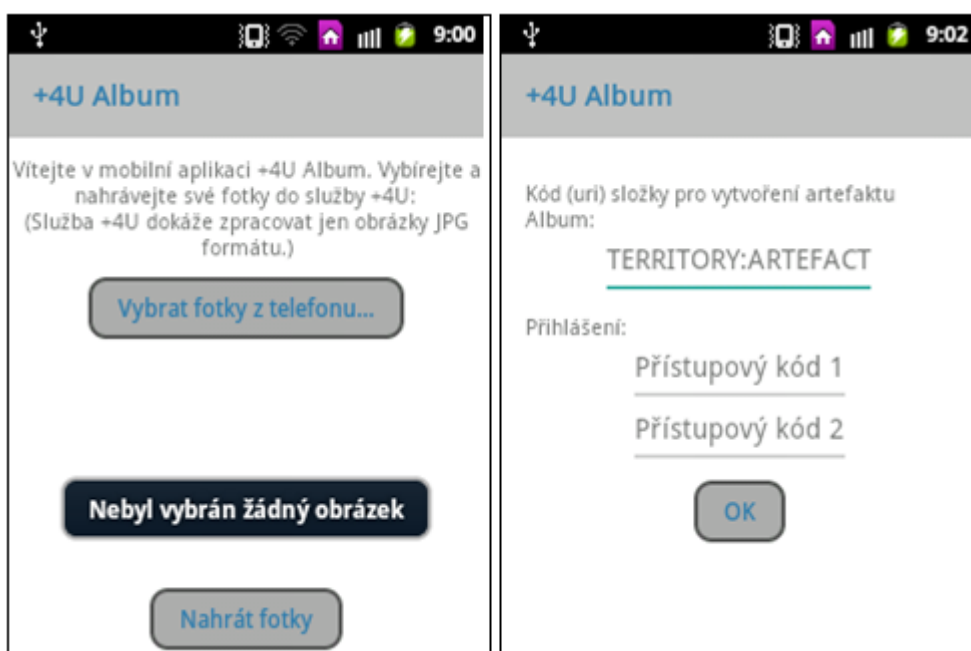
Třídy:	Layouty:
AlbumActivity.java	activity_album.xml
SaveImagesActivity.java	activity_save_images.xml
CreateThumbnailsClass.java	
CreateXmlFileClass.java	
UarchiveClass.java	

Zdroj: vlastní

Jako první byly při vývoji vytvořeny dvě *activity* s názvy AlbumActivity a SaveImagesActivity. Jsou to základní části aplikace. Jak už bylo řečeno v kapitole 3.1 *activity* nám umožňují zobrazování grafického uživatelského rozhraní a práci s ním, slouží

především jako prezentační vrstva aplikace. V *activity* se také kód vytváří nebo volá z jiných tříd.

Dále bylo potřeba vytvořit související XML soubory zvané *layout*, jež slouží k modelování grafického prostředí, jež je zobrazováno v *activity*. Vytváří se v nich menu, formuláře, tlačítka, texty, tabulky, apod. K *aktivitám* AlbumActivity a SaveImagesActivity byly vytvořeny dva související *layouty*: activity_album a activity_save_images. Na obrázku níže můžete vidět, jak se *layouty* vizuálně zobrazují pomocí *activity* na telefonu. Vlevo na obrázku je vidět první *layout* patřící k AlbumActivity a vpravo druhý, který je zobrazován pomocí SaveImagesActivity.



Obrázek 10: Zobrazení layoutů na telefonu
Zdroj: vlastní

V hlavní *activity* AlbumActivity (viz Obrázek 10 vlevo) vyzývá aplikace uživatele k vybrání fotek určených k nahrání do služby Plus4U. Po kliku na tlačítko pro vybrání fotek se spustí *intent*, což je jakýsi můstek mezi jednotlivými stavebními prvky platformy Android (např. *activity*, *services*, aplikacemi). V první etapě vývoje aplikace byl vytvořen *intent*, který spojuje aplikaci s interní aplikací v operačním systému Android: galerií. Galerie zobrazí veškeré obrázky v telefonu, z nichž se dá opětovně vybrat pouze jeden. Po vybrání obrázku se jeho data předají do aplikace +4U Album pomocí vytvořeného tzv. Content Provideru MediaStore, který se používá k sdílení dat mezi aplikacemi, a který

ukládá data o obrázcích (i dalších souborech) v tabulce. Po tomto kroku výběru aplikace přidá obrázek do vnitřního aplikačního pole obrázků vybraných uživatelem, vytvoří k němu náhled, a ten zobrazí pomocí grafického zobrazovacího prostředku typu *view* tzv. *ImageView* uživateli na obrazovce pomocí kódu vytvořenému v *AlbumActivity*. Níže lze vidět jak se obrázky prostřednictvím *layoutu* na telefonu zobrazí.



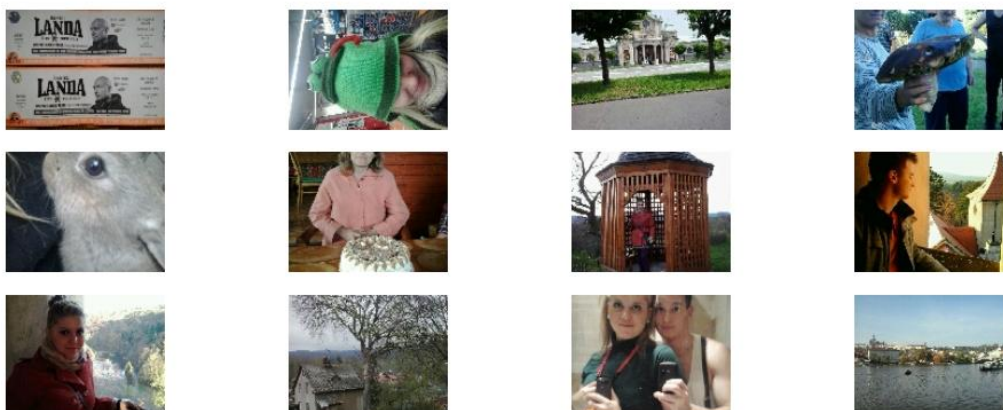
Obrázek 11: Zobrazení vybraných obrázků na telefonu
Zdroj: vlastní

Po tomto kroku lze z galerie po jednom vybírat další obrázky, jež se přidají také do vnitřního aplikačního pole, jinak jsou už ale obrázky připraveny k nahrávání. Kliknutím na tlačítko Nahrát fotky se pomocí dalšího vytvořeného *intentu* v *AlbumActivity*, kterému aplikace předá jako data pole obrázků, spustí *activity* *SaveImagesActivity*.

Activity *SaveImagesActivity* zobrazí uživateli formulář (viz Obrázek 10 vpravo). V něm uživatel vyplní cestu k artefaktu typu složka v prostředí Plus4U, v kterém se má Album vytvořit, a zadá své přihlašovací údaje. Potvrzením formuláře se údaje překontrolují a v případě správnosti se spustí postupně tři algoritmy, které byly vytvořeny pro zpracování obrázků, a poté asynchronní vlákno, v němž byl vytvořen kód pomocí knihovny s *uuAPI* příkazy pro nahrávání obrázků do služby.

První ze tří vyvinutých algoritmů, který je obsažen ve třídě `CreateThumbnailsClass` (viz Tabulka 2) projde pole obrázků, ke všem vytvoří další obrázky – náhledy a jim příslušná jména. Druhý algoritmus vytvoří XML soubor pomocí vytvořené třídy `CreateXmlFileClass`, v cílovém souboru se uvede několik dat o obrázcích a tabulka, která stanoví, jak se obrázky a náhledy budou zobrazovat v obsahu artefaktu v UU. Pomocí třídy `UarchiveClass` vloží třetí algoritmus obrázky, náhledy a XML soubor do jednoho pole, z kterého pak vytvoří archiv (komprimovaný soubor), jenž je typu `.uarchive` a je kompatibilní se stavebnicí UU.

Po těchto krocích se (stále v `SaveImagesActivity`) spustí asynchronní vlákno, které obsahuje kód pro komunikaci se službou Plus4U, jež je postavená na stavebnici UU. Zde je použita výše zmíněná knihovna pro práci s uuAPI příkazy. Knihovna je jakási další „nadstavba“ uuAPI příkazů. V kódu uuAPI příkazy spojí samotnou aplikaci se službou Plus4U přes internet pomocí autorizačních údajů uživatele. Poté se nalezne artefakt (složka), na kterém se má Album vytvořit. Z něj se přečtou určité informace, vytvoří se na něm nový artefakt typu dokument s algoritmem vygenerovaným unikátním systémovým kódem, na něm nový list, a posléze se do jeho obsahu nahraje archivní soubor typu `.uarchive`, který zde vytvoří cílové Album. Po ukončení nahrávání se zobrazí dialogové okno v *activity* `SaveImagesActivity`, které nese informaci jestli proběhlo nahrávání úspěšně či nikoliv.



Obrázek 12: Hotové album zobrazené v listu artefaktu
Zdroj: vlastní

Na obrázku výše lze vidět již vytvořené album na stránkách služby Plus4U. Album ukazuje náhledy obrázků, které jsou seřazené dle algoritmem vytvořené tabulky v XML souboru, který byl se souborem .uarchive nahrán. Po kliknutí na nějaký náhled se zobrazí celý příslušný obrázek. O operace rozbalení archivu, jeho převedení na takovéto album a zobrazování jeho obrázků se již stará samotná služba Plus4U.

4.3 Druhá etapa vývoje aplikace

Další etapa vývoje rozšířila funkci aplikace o další možnosti. V Tabulce 3 níže je opět znázorněna zjednodušená struktura projektu druhé etapy vývoje.

Tabulka 3: Zjednodušená struktura aplikace Plus4U Album v druhé etapě vývoje

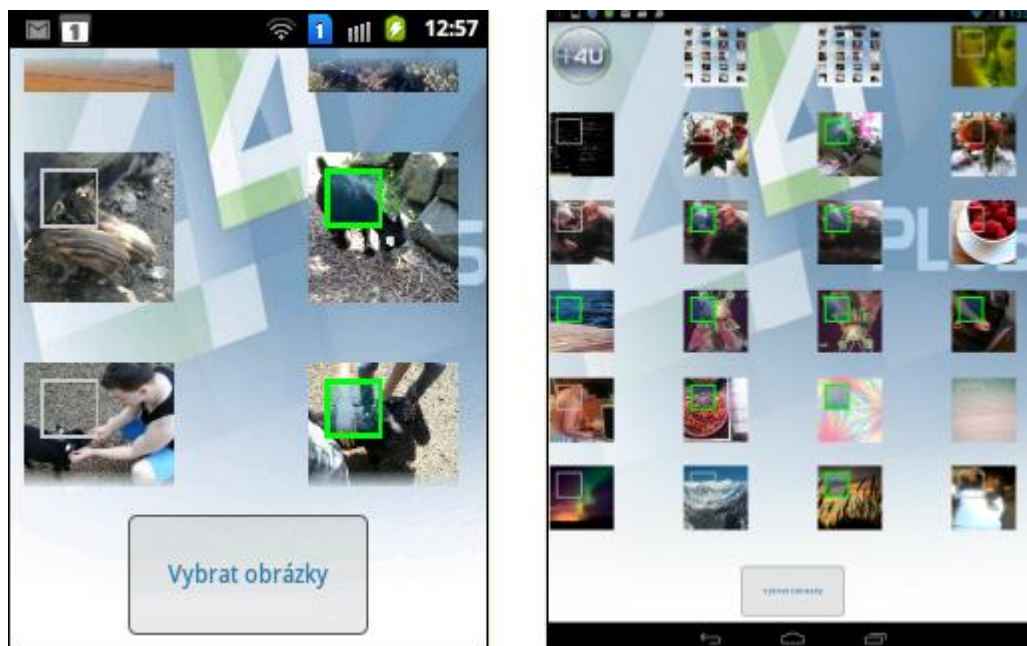
Třídy:	Layouty:
AlbumActivity.java	activity_album.xml
SaveImagesActivity.java	activity_save_images.xml
CreateThumbnailsClass.java	ac_image_grid.xml
CreateXmlFileClass.java	row_multiphoto_item.xml
UarchiveClass.java	
MultiPhotoSelectActivity.java	
BaseActivity.java	
AlertDialogFragment.java	

Zdroj: vlastní

Hlavním rozšířením aplikace bylo zvýšení efektivity výběru obrázků pro uživatele. Ve své první fázi aplikace spolupracovala s interní aplikací operačního systému Android Galerií, která umožňovala vybrat (kliknutím na tlačítko pro výběr) pouze jeden obrázek ze všech obrázků v mobilu. Jestliže tedy uživatel chtěl nahrát více obrázků do služby, musel stále opakovat operaci pro výběr obrázků, což bylo nevhodné. Z toho důvodu byla v projektu vytvořena nová galerie ve třídě MultiPhotoSelectActivity pomocí kódu dostupného na stránkách <http://www.technotalkative.com/android-select-multiple-photos-from-gallery/>. Tato nová galerie již vícenásobný výběr umožňuje. Problém s vícenásobným výběrem obrázků je v komunitě Android vývojářů znám a vzniklo již spoustu řešení, které se ním zabývají. [24] Vývoj vlastní aplikace galerie byl nejvýhodnější z důvodu spolehlivosti její funkce, jedná se ovšem o velmi obsáhlé téma. Její vývoj by mohl zabrat i měsíce. Z toho důvodu byl použit kód tohoto již vzniklého řešení a byl upraven dle vlastních požadavků. V nové aplikaci galerie byla také použita volně dostupná knihovna pro načítání obrázků telefonu s označením *universal-image-loader-1.9.0.jar*. [25]

V nové etapě byly tedy některé části aplikace upraveny a také byly vytvořeny nové. Nyní aplikace funguje tak, že se po kliknutí na tlačítko „Vybrat fotky z telefonu...“ spustí *intent*, který zapne do popředí *activity* MultiPhotoSelectActivity a sdělí jí, že požaduje

k navrácení nějaká data. MultiPhotoSelectActivity má k dispozici hned dva *layout* ac_image_grid.xml a row_multiphoto_item.xml. První *layout* obsahuje *view* zvané *GridView*, které umožňuje zobrazit více položek najednou a tlačítko pro potvrzení. Druhý *layout* obsahuje prvek tzv. *ImageView* pro zobrazení jednoho obrázku, *CheckBox* (zaškrtování políčko) pro možnost označení (výběru) obrázku, a slouží jako šablona, jenž určuje jak se má zobrazit jeden obrázek uvnitř grafického prvku *GridView*. Po zapnutí se tedy MultiPhotoSelectActivity spojí s úložištěm obrázků v telefonu pomocí *Content Provideru* a uloží adresy (url) obrázků do pole. Poté se spustí v *activity* další prvek Android aplikace – tzv. *Adapter*. Ten rozdělí adresy obrázků z pole na jednotlivé obrázkové položky (graficky se řídí podle dvou výše zmíněných *layoutů*) a pomocí metod z knihovny *universal-image-loader-1.9.0.jar* v rozdělených položkách zobrazí jednotlivé obrázky telefonu. Výsledné zobrazení je vidět na následujícím obrázku. Vlevo je zobrazena galerie na telefonu, vpravo na tabletu.



Obrázek 10: Zobrazení nové galerie na telefonu a tabletu
Zdroj: vlastní

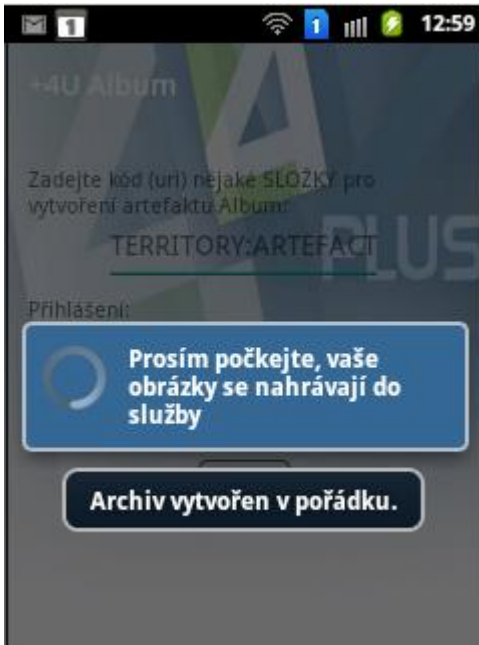
Takto se tedy uživateli zobrazí všechny obrázky z jeho telefonu. Díky *CheckBoxu*, který má každý obrázek překrytý přes sebe, může uživatel označit libovolný počet obrázků a tlačítkem „Vybrat obrázky“ výběr potvrdit. Kliknutím na potvrzující tlačítko „Vybrat obrázky“ se spustí nový *intent*, který přebere vybrané adresy obrázků, nastaví je jako

výsledek a *activity* MultiPhotoSelectActivity ukončí. Výsledné obrázky jsou zpracovány v AlbumActivity, kde se nachází s novou etapou nový algoritmus, který umožní zobrazit všechny vybrané obrázky uživateli na obrazovku najednou pomocí několika *ImageView*. Nyní má uživatel možnost provést zcela nový výběr obrázků nebo již nahrát obrázky do služby. Níže je znázorněno, jak aplikace vypadá před (vlevo) a po (vpravo) vybrání obrázků pomocí nové galerie. Lze si také všimnout, že s novou etapou byl pozměněn i *design* aplikace.



Obrázek 11: Zobrazení vybraných obrázků v AlbumActivity
Zdroj: vlastní

Když jsou již obrázky vybrané, je aplikace připravena pro nahrání obrázků do služby Plus4U. Po kliknutí na tlačítko „Nahrát“ se (stejně jako v první etapě vývoje aplikace) spustí *SaveImagesActivity*, která zobrazí formulář pro přihlášení a pro zadání cesty k složce, kde se má artefakt vytvořit. Dále po potvrzení údajů se vytvoří komprimovaný soubor *.uarchive*, který je nahrán do služby, a který zde vytvoří cílové album. Vytvořené album v systému vypadá stále stejně, jako na Obrázku 12 v předešlé kapitole. Na následujícím obrázku je znázorněno, jak vypadá aplikace při nahrávání fotek do služby Plus4U.

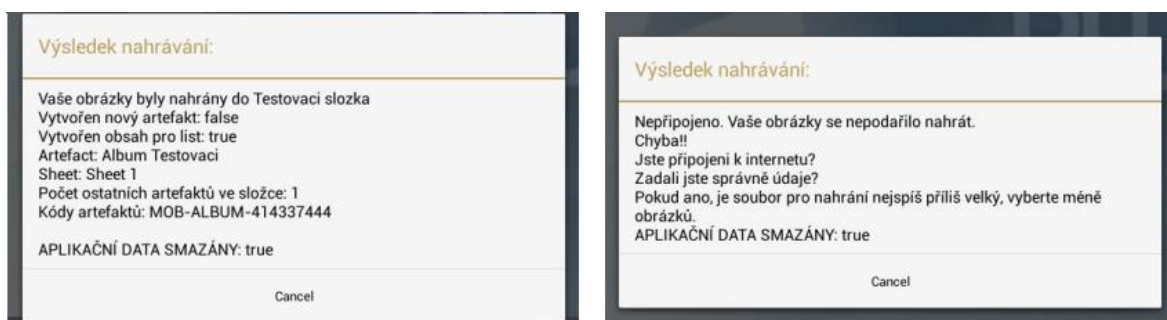


Obrázek 12: Nahrávání obrázků do služby
Zdroj: vlastní

V Tabulce 3, kde je uvedena struktura projektu, se nachází další dvě třídy, jež zde stále nebyly zmíněny. Jednou z nich je `BaseActivity`. Tato vytvořená třída je velmi jednoduchá, slouží spíše jako *service* (služba). Neobsahuje žádné grafické prostředí a nese pouze instanci tzv. `ImageLoaderu`, který je z knihovny `universal-image-loader-1.9.0.jar` použit v `MultiPhotoSelectActivity` k nahrávání obrázků do *layoutu* (`MultiPhotoSelectActivity` od `BaseActivity` dědí a získává tak instanci `ImageLoaderu`).

Poslední třídou z Tabulky 3 je `AlertDialogFragment`. Tato třída není typu *activity*, nýbrž typu *fragment*, který nám umožňuje nosit část kódu (viz. Kapitola 3.2.2). Jak lze ze stávajícího popisu aplikace `Plus4U Album` vidět, pro její vývoj byl převážně použit přístup *nefragmentový*, z důvodu méně složitého a časově náročného vývoje aplikace. V tomto případě ale byla potřeba udělat výjimku. V první etapě aplikace byl totiž vytvořený dialog (který se nachází na konci asynchronního vlákna pro nahrání obrázků do služby) stále problémový, často kvůli němu spadla celá aplikace. V asynchronním vlákně se zobrazování grafických prvků uživateli chovalo jinak než přímo uvnitř *activity*, a proto byl vytvořen nový *fragment* typu `AlertDialogFragment`, který nese grafické znázornění dialogu, zobrazuje se při ukončování asynchronního vlákna k nahrávání v `SaveImagesActivity` a obsahuje informace o úspěšném/neúspěšném nahrání souboru do

služby. *Fragment* nám umožňuje zobrazit okno jen přes část obrazovky. Níže lze vidět na obrázku zobrazení dialogu v případě úspěšného (vlevo) a neúspěšného (vpravo) nahrání.



Obrázek 13: Dialogová okna informující o výsledku nahrávání
Zdroj: vlastní

Velkým omezením pro aplikaci, který se s druhou etapou vývoje objevil, je velká paměťová náročnost aplikace na zařízení. Z důvodu aplikační práce s obrázky, k nimž se vytváří náhledy, a jež se zobrazují několikrát uživateli, začalo vznikat spousta problémů v aplikaci, které se týkaly překročení maximální alokovatelné paměti. V této etapě vývoje se problém také zhoršil s použitím nové galerie pro vícenásobný výběr obrázků, která je paměťově náročnější. Z tohoto důvodu bylo vytvořeno několik algoritmů pro odlehčení paměti. Aplikace byla paměťově více uvolněna a nyní se problémy s pamětí objevují méně. Nutnost je zde ale uvést informaci, že po testování aplikace na starším zařízení – mobilním telefonu Samsung Y DUOS s Android verzí 2.3.3, který není tak hardwarově výkonný, vznikaly tyto problémy stále, i když již ne tak často. Stále se tedy může stát, že v aplikaci skončí nahrávání obrázků neúspěchem. Paměťové problémy jsou v aplikaci ošetřeny, kdyby ovšem nastal nějaký, který by nebylo možno zachytit, jistě pomůže restart aplikace.

4.4 Vyhodnocení a přínos aplikace Plus4U Album

Aplikace splnila základní požadavky zadání a rozvinula svou efektivitu díky několika vylepšením, a to zejména díky nové galerii, jež byla přidána v druhé etapě jejího vývoje, a která umožňuje uživatelsky přívětivější výběr obrázků. S vývojem této galerie se bohužel zvýšila paměťová náročnost aplikace na zařízení, kvůli čemuž vzniklo nemálo problémů aplikace. Problémy aplikace byly ošetřeny a paměťová náročnost byla částečně snížena několika úpravami kódu a novými algoritmy pro uvolnění paměti. V druhé etapě bylo také

zjištěno omezení při nahrávání vyššího počtu obrázků do služby Plus4U. Na testovacím tabletu se to stávalo konkrétně při nahrávání 90 a více obrázků. Aplikace v takovém případě často vypíše zprávu, že nebylo možno takový objem dat zpracovat. Záleží samozřejmě na každém zařízení, jakou má výkonnost. U každého však existuje určitý maximální počet zpracovatelných obrázků, který nelze překročit.

Aplikace přináší možnost spolupráce Android mobilních zařízení s informačními systémy postavenými na UU. Tuto spolupráci řeší v kódu prostřednictvím asynchronního vlákna, v kterém je komunikace zařízení se systémem pomocí uuAPI příkazů obsažena. Díky tomu může nahrávání obrázků do služby běžet paralelně s jinými operacemi v telefonu, jejichž průběh není tak omezován.

Pro společnost Unicorn je důležité ulehčovat práci uživatelům informačních systémů postavených na UU. Vede to ke zvýšení efektivnosti práce, k zlepšení spokojenosti zaměstnanců a zákazníků, a ke kladnému vnímání informačního systému. Z toho důvodu vznikají nové funkce na platformě UU (samozřejmě i na jiných, jež s UU spolupracují), které se staví do služeb a usnadňují uživatelskou práci. Jednou z takových nových funkcí je právě mobilní aplikace Plus4U Album. Tato aplikace usnadní práci uživatelům a zvýší efektivnost práce. Je velmi uživatelsky příjemné propojit přímo své mobilní zařízení s informačním systémem a nahrávat na něj potřebné fotky. Ušetří to jistě spoustu času. Bez této aplikace by se muselo mobilní zařízení spojit s počítačem, nahrávat fotky na něj a poté do systému. Aplikace zároveň prezentuje svou funkcí a svým designem službu Plus4U a stavebnici UU, čímž může zvýšit kladnost uživatelova vztahu k samotnému informačnímu systému.

Ekonomický přínos aplikace není tak lehké změřit. Aplikace bude k dispozici uživatelům zdarma a nelze tedy vypočítat přímo tržební částku, kterou přinese. Aplikace ale není určena pro přímý výdělek, nýbrž pro rozšíření funkcí platformy UU a služeb na platformě postavených. Aplikace bude přínosem všem uživatelům, kteří služby Plus4U využívají. Mobilní aplikace Plus4U Album tedy nebude k dispozici pouze zákazníkům skupiny společností Unicorn, ale také (a možná především) jejím zaměstnancům, kteří v prostředí informačního systému UU pracují, a kde má každý z nich vytvořený svůj osobní prostor v My Territory. Tato aplikace jim umožní lehce zálohovat své obrázky, fotky z mobilního

zařízení do jejich prostoru, umožní jim vytvářet alba rodinných fotek v osobním prostoru My Territory, či nahrávat obchodní foto dokumenty do pracovního prostoru Business Territory. Spoustu zaměstnanců a zákazníků jistě uvítá novou aplikaci, která propojí jejich telefon, tablet, atd. se službou Plus4U, a která jim umožní snadno ukládat své obrázky do služby. Aplikace bude velkým přínosem pro platformu UU a společnost Unicorn.

Závěr

Hlavním cílem bakalářské práce bylo vytvořit konkrétní Android mobilní aplikaci nad digitální stavebnicí UU prostřednictvím vlastního řešení, což se podařilo pouze za pomoci zmíněné knihovny pro práci s uuAPI příkazy.

Teoretická část této práce splňuje stanovené dílčí cíle práce, jimiž je představení platformy UU společnosti Unicorn a poskytnutí náhledu na platformu Android společně s popsáním, rozebráním a vyhodnocením přístupů k vytváření mobilních aplikací na platformě.

Aplikace umožňuje vybírat a nahrávat obrázky z mobilního zařízení s operačním systémem Android do informačních systémů postavených na digitální stavebnici informačních systémů UU. Plynulost a funkce aplikace je bohužel omezena výkonností zařízení, na kterém je spuštěna. Na každé ze zařízení lze nahrát pouze určitý maximální počet obrázků (na testovacím tabletu byl max. počet 90 obr.). Ekonomické vyhodnocení přínosu aplikace je těžké změřit, protože bude poskytována zdarma, její návratnost ale tkví v ulehčení a zpříjemnění práce s informačním systémem uživatelům.

Hlavním aspektem, na který by měl být další směr vývoje aplikace zaměřen, je zvýšení maximálního možného počtu zpracovatelných obrázků k nahrávání do služby a umožnění nahrávání více obrázků do služby najednou i zařízením, které nedisponují příliš velkou pamětí a výkonným procesorem. Velmi dobrým výsledkem by bylo, kdyby se podařilo paměťové problémy aplikace úplně odstranit. Dalším návrhem na rozšíření funkcionality aplikace do budoucna je umožnění editace již existujících alb v informačním systému. S tímto vylepšením by bylo potřeba využít uuAPI příkazů k stažení obsahu artefaktu (obrázků) do aplikace. V tomto vylepšení jsou tedy hned dvě nové výhody. Uživatel by si mohl stahovat obrázky ze služby Plus4U přímo do mobilního telefonu a mít je tak k dispozici v zařízení, a také by si mohl upravovat svá alba. Toto vylepšení by bylo jistě velmi paměťově náročné a jeho řešení by bylo nelehké, ale v případě úspěchu by přineslo další přidanou hodnotu uživatelům. Pro pohodlnější práci uživatele v aplikaci by dále bylo výhodné přidat do aplikace práci s nějakým uložištěm dat. V něm by měl možnost ukládat identifikační kódy svých alb (tj. adresy složek v informačním systému) a své přihlašovací údaje (samozřejmě zašifrované) a zpětně je volat. Tím by měl ulehčenou práci při

nahrávání obrázků. Dobré by také bylo, kdyby aplikace umožňovala spravovat více účtů (ukládat více přihlašovacích údajů), mezi kterými by se dalo přepínat. Takto by aplikaci mohlo snadno využívat více uživatelů na jednom zařízení. Vhodným vylepšením aplikace by také bylo vložení asynchronního vlákna do notifikací telefonu (na pozadí), takže by nezabíralo obrazovku. V současné době totiž sice nebrání k výkonu jiným procesům na pozadí zařízení, omezuje ale práci uživatele zabráním displeje při nahrávání z důvodu zobrazování dialogu. Dále by v budoucnu také bylo dobré umožnit uživateli vybrané obrázky zobrazené na displeji odstraňovat v případě, že si jejich nahrání rozmyslí. Obrázky by odstraňoval pomocí nějakého dotykového gesta. Nakonec by snad nejvíce mohla aplikace svou funkcionalitu vylepšit, kdyby se v budoucnu její zaměření rozšířilo nejen na obrázky, ale také na ostatní data v telefonu. Velkou přidanou hodnotou pro uživatele by bylo, kdyby aplikace umožnila také nahrávání zvukových stop, videí, prezentací, apod. Tímto krokem by se efektivita aplikace zvýšila a ještě více by zpříjemnila práci uživatelům služby Plus4U.

Snad největším přínosem se aplikace stala pro jejího autora. Při práci na projektu mobilní aplikace Plus4U Album se autor této práce zejména naučil vytvářet aplikace pro platformu Android. Získal a stále získává mnoho znalostí o technologii Android, nové zkušenosti se stavebnicí Unicorn Universe a s programovacím jazykem Java.

Seznam použité literatury

Citace

- [1] MEIER, R. *Professional Android 4 application development*. Updated for Android 4. Indianapolis: John Wiley, 2012, xlii, 817 p. ISBN 978-111-8262-153.
- [2] Univerzální metodika pro řízení podniků a organizací. UNICORN A.S. *Unicorn Universe*. [online]. [vid. 2015-04-14]. Dostupné z: <https://unicornuniverse.eu/cz/uup.html>.
- [3] UNICORN A.S. *Unicorn Universe*. [online]. [vid. 2015-04-14]. Dostupné z: <https://unicornuniverse.eu/>.
- [4] MELOUNOVÁ, L. Unicorn Universe OS. Interní dokumenty UNICORN COLLEGE S.R.O. *Plus4U* [online]. 2012 [vid. 2015-04-14]. Dostupné z: <https://plus4u.net/>.
- [5] Příručka Plus4U. Interní dokumenty UNICORN TOP GUN ACADEMY. *Plus4U* [online]. 2014 [vid. 2015-04-14]. Dostupné z: <https://plus4u.net/>.
- [6] KOVÁŘ, V. Vladimírovo - prezentace. Interní dokumenty UNICORN TOP GUN ACADEMY. *Plus4U* [online]. 2011 [vid. 2015-04-14]. Dostupné z: <http://plus4u.net/>.
- [7] Unicorn Universe Process Documentation. Interní dokumenty VIGOUR A.S. *Plus4U* [online]. 2013 [vid. 2015-04-14]. Dostupné z: <http://plus4u.net/>.
- [8] Objektový operační systém virtuálního superpočítače. UNICORN A.S. *Unicorn Universe*. [online]. [vid. 2015-04-14]. Dostupné z: <https://unicornuniverse.eu/cz/uuos.html>.
- [9] Nástroj pro vizuální modelování a komunikaci. UNICORN A.S. *Unicorn Universe*. [online]. [vid. 2015-04-14]. Dostupné z: <https://unicornuniverse.eu/cz/uubml.html>.

- [10] LÍZNEROVÁ, Z., J. MĚŘINSKÝ. Unicorn Universe Applications. Interní dokumenty UNICORN COLLEGE S.R.O. *Plus4U* [online]. 2014 [vid. 2015-04-14]. Dostupné z: <https://plus4u.net/>.
- [11] VRÁNA, M. Tak se dělá software & Produktové portfolio. Interní dokumenty UNICORN TOP GUN ACADEMY. *Plus4U* [online]. 2013 [vid. 2015-04-14]. Dostupné z: <https://plus4u.net/>.
- [12] UJBÁNYAI, M. *Programujeme pro Android*. Vyd. 1. Praha: Grada, 2012, 187 s. Průvodce (Grada). ISBN 978-80-247-3995-3.
- [13] ALLEN, G. *Android 4: průvodce programováním mobilních aplikací*. 1. vyd. Překlad Jakub Mužík. Brno: Computer Press, 2013, 656 s. ISBN 978-80-251-3782-6.
- [14] SEMECKÝ, V. Android Studio – nové vývojové prostředí. *Zdrojak.cz* [online]. 2013 [vid. 2015-04-14]. Dostupné z: <http://www.zdrojak.cz/clanky/android-studio-nove-vyvojove-prostredi/>.
- [15] App Resources. *Developer.android.com* [online]. 2014 [vid. 2015-04-14]. Dostupné z: <http://developer.android.com/guide/topics/resources/index.html>.
- [16] ADT Plugin Release Notes. *Developer.android.com* [online]. 2014 [vid. 2015-04-14]. Dostupné z: <http://developer.android.com/tools/sdk/eclipse-adt.html>.
- [17] Android Studio Overview. *Developer.android.com* [online]. 2014 [vid. 2015-04-14]. Dostupné z: <http://developer.android.com/tools/studio/index.html>.
- [18] Differences between Android NetBEANS plugin and Eclipse plugin?. *Stackoverflow.com* [online]. 2011 [vid. 2015-04-14]. Dostupné z: <http://stackoverflow.com/questions/6756396/differences-between-android-netbeans-plugin-and-eclipse-plugin>.
- [19] Best IDE (Netbeans or eclipse) for android application development?. *Stackoverflow.com* [online]. 2013 [vid. 2015-04-14]. Dostupné z:

<http://stackoverflow.com/questions/18953910/best-idenetbeans-or-eclipse-for-android-application-development>.

- [20] List of IDEs for Android App Development, Which is Best for You?. *Tek Eye: Programming, Android and Computing Technology* [online]. 2014 [vid. 2015-04-14]. Dostupné z: <http://tekeye.biz/2014/list-of-android-app-development-ides>.
- [21] NetBeans. CANONICAL LTD. *Ubuntu.cz* [online]. 2012 [vid. 2015-04-14]. Dostupné z: <http://wiki.ubuntu.cz/programov%C3%A1n%C3%AD/netbeans>.
- [22] <uses-sdk> *Developer.android.com* [online]. 2015 [vid. 2015-04-14]. Dostupné z: <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>.
- [23] Fragments. GOOGLE INC. *Developer.android.com* [online]. 2015 [vid. 2015-04-14]. Dostupné z: <http://developer.android.com/guide/components/fragments.html>.
- [24] MAYANI, P. Android – Select multiple photos from Gallery. *TechnoTalkative* [online]. 2012 [vid. 2015-04-17]. Dostupné z: <http://www.technotalkative.com/android-select-multiple-photos-from-gallery/>.
- [25] TARASEVICH, S. Android-Universal-Image-Loader. GITHUB, INC. *GitHub* [online]. 2015 [vid. 2015-04-17]. Dostupné z: <https://github.com/nostra13/Android-Universal-Image-Loader>.

Bibliografie

CASTLEDINE, E., M. EFTOS and M. WHEELER. *Build mobile websites and apps for smart devices*. 1st ed. Collingwood, VIC, Australia: SitePoint, 2011. ISBN 09-870-9084-4.

SCHILDT, H. *Java 7: výukový kurz*. 1. vyd. Brno: Computer Press, 2012, 664 s. ISBN 978-80-251-3748-2.

Seznam příloh

Příloha A: zadní desky – CD s instalačním souborem