



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ
ÚSTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT
INSTITUTE OF INFORMATICS

NÁVRH A REALIZACE MOBILNÍ APLIKACE PRO ZAŘÍZENÍ ANDROID

DESIGN AND IMPLEMENTATION OF MOBILE APPLICATION FOR ANDROID DEVICES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. TOMÁŠ BOHÁČ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR DYDOWICZ, Ph.D.

BRNO 2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

Boháč Tomáš, Bc.

Informační management (6209T015)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává diplomovou práci s názvem:

Návrh a realizace mobilní aplikace pro zařízení Android

v anglickém jazyce:

Design and Implementation of Mobile Application for Android Devices

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrh řešení, přínos práce

Závěr

Seznam použité literatury

Seznam odborné literatury:

GARGENTA, Marko. Learning Android. 1st ed. Sebastopol, Calif.: O'Reilly, c2011, xvii, 245 p. ISBN 14-493-9050-1.

LEE, W. a M. Beginning Android application development. Indianapolis, IN: Wiley Pub., 2011. 428 s. Wrox beginning guides. ISBN 978-111-8087-800.

MARTIŠEK, D. Algoritmizace a programování v Delphi. 1. vyd. Brno: Littera, 2007. 230 s. ISBN 978-80-85763-37-9.

UJBÁNYAI, M. Programujeme pro Android. 1. vyd. Praha: Grada, 2012. 187 s. Průvodce (Grada). ISBN 978-80-247-3995-3.

VELTE, A., T. VELTE a R. ELSENPETER. Cloud Computing: praktický průvodce. 1. vyd. Brno: Computer Press, 2011. 344 s. ISBN 978-80-251-3333-0.

Vedoucí diplomové práce: Ing. Petr Dydowicz, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2014/2015.

L.S.

doc. RNDr. Bedřich Půža, CSc.
Ředitel ústavu

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
Děkan fakulty

V Brně, dne 28.2.2015

Abstrakt

Diplomová práce se zabývá návrhem a vývojem mobilní aplikace pro operační systém Android. K realizaci jsou použity metody a nástroje pro vývoj mobilních aplikací. Aplikace je implementována v programovacím jazyku Java s použitím knihovny Jsoup. Hlavním cílem aplikace je zpřístupnit funkce informačního systému Vysokého učení technického v Brně i na mobilních zařízeních.

Abstract

This diploma thesis is focused on mobile application design and development for Android operating system. Mobile application development methods and tools were used in order to accomplish set goals. The application is implemented in Java programming language with use of Jsoup library. The main purpose of the application is to allow access to Brno University of technology's information system on mobile devices.

Klíčová slova

Mobilní aplikace, Java, Jsoup, informační systém, Android

Keywords

Mobile application, Java, Jsoup, information system, Android

Bibliografická citace

BOHÁČ, T. *Návrh a realizace mobilní aplikace pro zařízení Android*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2015. 71 s. Vedoucí diplomové práce Ing. Petr Dydowicz, Ph.D.

Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně, dne 29. května 2015

.....

Poděkování

Děkuji vedoucímu práce Ing. Petru Dydowiczovi, Ph.D., za odborné vedení, cenné připomínky, náměty a konzultace, kterými přispěl k vypracování této diplomové práce.

Obsah

Úvod	10
1 Cíle práce, metody a postupy zpracování.....	12
2 Teoretická východiska práce	13
2.1 Vývoj trhu s mobilními zařízeními.....	13
2.2 Mobilní operační systémy	14
2.3 Životní cyklus Android aplikace	17
2.4 Architektura OS Android.....	18
2.5 Základní prvky grafického rozhraní	20
2.6 Java	21
2.7 API.....	23
2.8 Metodiky vývoje software	23
2.9 Životní cyklus softwaru	25
2.10 Fáze tvorby SW produktu.....	26
2.11 Vývojové nástroje.....	26
2.12 Jsoup	28
3 Analýza současného stavu	29
3.1 Informační systém Fakulty podnikatelské	29
3.2 Průzkum typu používaných OS	33
3.3 Porovnání doby přístupu do IS VUT z různých zařízení.....	34
3.4 Mobilní aplikace českých univerzit.....	36
3.5 UniApps v2 Beta	37
3.6 SWOT analýza IS VUT Fakulty Podnikatelské	38

4	Vlastní návrhy řešení.....	40
4.1	Identifikace potřeb studentů, požadavků na aplikaci.....	40
4.2	Volba operačního systému.....	40
4.3	Popis návrhu a architektury vytvořené aplikace.....	41
4.4	Jednotlivé funkční části aplikace.....	53
4.5	Porovnání doby přístupu k informacím.....	60
4.6	Možnosti propagace aplikace.....	62
4.7	Reálné přínosy.....	63
4.8	Prognóza do budoucnosti, možnosti rozšíření.....	63
	Závěr.....	65
	Seznam použitých zdrojů.....	67
	Seznam obrázků.....	70
	Seznam tabulek.....	71

Úvod

Tématem mé diplomové práce je návrh mobilní aplikace pro studenty Vysokého učení technického (VUT), Fakulty podnikatelské v Brně. Nápad iniciovaly dvě věci – první z nich byla neexistence mobilní aplikace či mobilní verze webových stránek informačního systému, přes který se studenti přihlašují na zkoušky, nahlízejí na svoje rozvrhy, hledají informace o zaměstnancích VUT a jiné potřebné informace. Druhým podnětem byl systém vypisování zkoušek. Každý předmět má sice vypsáno více termínů zkoušek, ale pokud není student přítomen prakticky pořád u počítače a nekontroluje jejich vypsání, může velmi snadno šanci registrace na požadovaný termín propásnout a nezbývá mu nic jiného, než se přihlásit pouze na pozdější termín. Proto jsem si položil otázku: „Jak se dozvědět co nejdříve o vypsání nového termínu zkoušky bez nutnosti neustálé kontroly informačního systému?“

První myšlenka vedla k napsání skriptu, který bude nové termíny kontrolovat a v případě nového termínu pošle oznamovací email. Velikou nevýhodou by ovšem byla nutnost neustálého běhu tohoto programu na nějakém serveru. A co když nebudu zrovna u počítače? Budu se muset znovu přihlašovat prostřednictvím telefonu do systému a registrovat. To není zrovna ideální a pohodlné řešení hodné 21. století. Proč tedy nenechat telefon, aby to dělal za mě a já se pak jen jedním dotykem displeje na zkoušku přihlásil? Ano, to je to pravé řešení! Tímto myšlenkovým pochodem jsem tedy došel k tématu své práce a doufám, že bude užitečná i pro mé kolegy, ostatní studenty.

System bude navržený pro chytré telefony s operačním systémem Android, jelikož tyto jsou u nás nejrozšířenější. Aplikace bude zaměřená především na řešení problematiky přihlašování na zkoušky, ovšem budou implementovány i další základní funkce, aby se mohla stát plnohodnotnou univerzitní aplikací každého studenta. V prvních fázích je nutno udržet koncentraci a soustředit se pouze na klíčové funkce a postupně implementovat další. Mezi základní funkce lze počítat zobrazování aktualit z předmětu, prohlížení elektronického indexu, zobrazování svého rozvrhu a již zmíněné přihlašování na zkoušky.

Každá tato činnost bude doprovázena funkcí hlídání změn provedených na daném modulu, kterou současný informační systém nedisponuje, přestože její implementace pravděpodobně není nijak zvlášť složitá. Nicméně pro realizování tohoto nápadu je to spíše výhoda, jelikož většina studentů by tuto funkci ocenila a proto věřím, že aplikace neupadne v zapomnění a bude hojně využívána.

Pokud se předchozí uvedené funkcionality podaří dosáhnout, aplikace má šanci stát se komplexním pomocníkem každého studenta Fakulty podnikatelské. V případě úspěchu lze aplikaci portovat i na ostatní u nás rozšířené platformy, zejména pak přichází v úvahu velmi rozšířený iOS.

1 Cíle práce, metody a postupy zpracování

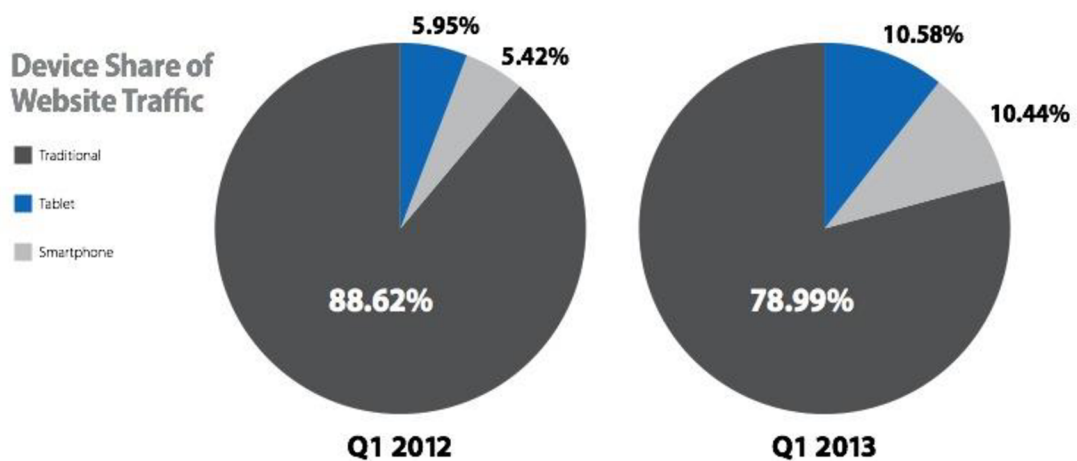
Cílem této práce je navrhnout a realizovat mobilní aplikaci pro studenty Fakulty podnikatelské. Prostřednictvím vytvořené aplikace bude možno přistupovat k různým částem informačního systému a student tak bude moci provádět určité úkony z mobilního telefonu. Součástí práce bude i analýza současného stavu a ostatních aplikací na trhu určených pro univerzity. Aplikace bude vyvíjena pro operační systém Android, implementace bude řešena v jazyce Java s pomocí knihovny Jsoup.

Aplikace se zaměří na čtyři klíčové prvky informačního systému – prohlížení aktualit z předmětu, možnost zkontrolovat elektronický index, schopnost přihlašovat se na zkoušky a prohlížet vlastní rozvrh. Nejdůležitější funkcí aplikace však bude upozorňování studenta na změny provedené v informačním systému. Těmito změnami se myslí například vystavení nové aktuality, zapsání známky do indexu nebo vypsání nových termínů zkoušek. Tyto funkce současný informační systém nenabízí a bude tak klíčovou a jedinečnou funkcí aplikace.

2 Teoretická východiska práce

2.1 Vývoj trhu s mobilními zařízeními

Před samotným začátkem vývoje je potřeba udělat alespoň základní průzkum trhu, který napoví, zda bude mít navržená aplikace možnost získat potenciální uživatele. V tomto případě bylo snahou zjistit, jak je rozvinutý trh s mobilními zařízeními, konkrétně z jakého typu zařízení uživatelé přistupují na web. Z grafu níže lze vyčíst jasný trend, kdy v průběhu jednoho roku se počet uživatelů přistupujících z mobilních zařízení zdvojnásobil (6).



Obrázek 1: Přístupy na web dle typu zařízení, Zdroj: (6)

Zajímavé je také pozorovat konkrétní vývoj typů mobilních zařízení přistupujících na web. Kupříkladu je důležité si všimnout silné pozice iPadu na trhu tabletů, ovšem neopomenout stoupající tendenci přístupů tabletů s Androidem, který zaznamenal dvojnásobné zvýšení. Z tabulky je možno také vidět, jak některé hodnoty oscilují kolem určité hodnoty, což je dáno metodikou měření, respektive jejími nepřesnostmi (6).

Tabulka 1: Přístupy na web dle konkrétních zařízení, Zdroj: (6)

Website Visits by Device		Q1 2012	Q2 2012	Q3 2012	Q4 2012	Q1 2013
OVERALL	Traditional	88.62%	85.27%	82.61%	80.74%	78.99%
	Tablet	5.95%	7.09%	8.48%	8.93%	10.58%
	Smartphone	5.42%	7.64%	8.91%	10.34%	10.44%
TABLET	iPad	91.68%	90.82%	90.17%	89.49%	89.28%
	Kindle Fire	3.58%	4.06%	4.08%	3.41%	2.51%
	Android	4.75%	5.12%	5.75%	7.10%	8.20%
SMARTPHONE	iPhone	58.20%	61.48%	59.78%	59.75%	63.20%
	Android	40.16%	36.93%	38.51%	38.81%	35.52%
	Windows	0.90%	0.92%	0.92%	0.94%	1.04%
	Other	0.74%	0.66%	0.80%	0.50%	0.25%

2.2 Mobilní operační systémy

Na mobilních zařízeních zákazníků po celém světě lze nalézt nespočet typů operačních systémů. V tabulce níže je přehled těch nejrozšířenějších společně s jejich tržním podílem a srovnáním roku 2012 s rokem 2013. Tento pohled bere v potaz celosvětový podíl, tudíž se nezabývá rozdělením operačních systémů na jednotlivých trzích. Na prvním místě se umístil operační systém Android, který drží prvenství téměř ve všech zemích Evropy. Na druhém místě je pak operační systém od firmy Apple – iOS, který má dominantní zastoupení na trzích v USA a Kanadě. Po malých krůčcích vzrůstá i tržní podíl operačního systému od společnosti Microsoft - Windows Phone. Výrazný úpadek v poslední době zaznamenává systém BlackBerry, který do svého systému dlouho neimplementoval nové funkce a nedržel tak krok s tahouny v čele s Androidem a iOS, což mělo za následek odliv zákazníků (3).

Pro ilustraci bude ještě uveden tržní podíl v České republice: v čele je znovu Android s podílem 81% následován iOS s 12,9%, za nimi se následuje s 3,6% Windows Phone a na posledním místě BlackBerry s 1,7%. Zbytek trhu mají rozebrány ostatní mobilní operační systémy, jichž je na trhu nespočet a není tudíž nutné je vyjmenovávat. Jen pro úplnost je potřeba uvést, že se jedná o data z třetího kvartálu roku 2013 (15).

Tabulka 2: Tržní podíl mobilních operačních systémů, Zdroj: (3)

Global Smartphone Operating System Marketshare %	Q3 '12	Q3 '13
Android	75.0%	81.3%
Apple	15.6%	13.4%
Microsoft	2.1%	4.1%
BlackBerry	4.3%	1.0%
Others	3.0%	0.2%
Total	100.0%	100.0%

2.2.1 Android

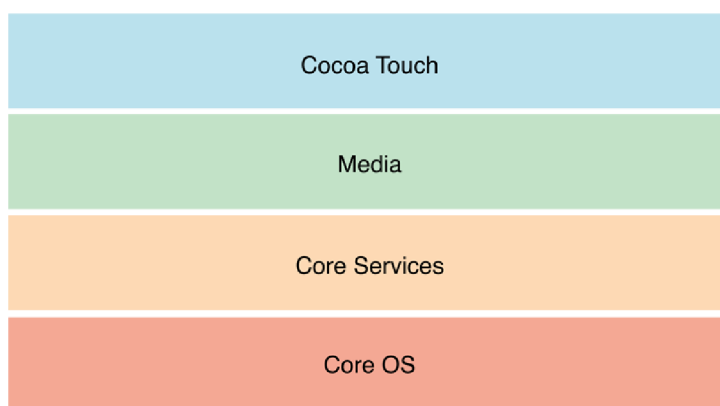
Android je rozsáhlý operační systém vytvořený společností Google, založený na open source platformě, tedy jedná se o počítačový software s otevřeným zdrojovým kódem. Otevřený kód zde znamená, že uživatel jej může při splnění jistých podmínek využívat zadarmo a tato licenční politika mu také umožňuje přistoupit ke zdrojovým kódům, které následně využívá nebo upravuje podle svých potřeb. OS je založen na Linuxovém jádře 2.6 různých verzí, které zajišťuje zabezpečení systému jako celku, správu paměti, správu procesů, přístup k síti a ovladačům všech vnitřních senzorů a komponent. Jednotlivé aplikace k funkcím jádra nepřístupují přímo, ale prostřednictvím Android API (17, str. 13).

„Android je tedy progresivní operační systém primárně vyvíjen jako platforma převážně pro PDA, tablety a tzv. chytré telefony. Byl postaven od základu, který umožní vývojářům vytvářet působivé mobilní aplikace, jež mohou plně využívat všech vlastností, které telefon nabízí, jako např. základní funkce telefonu (obsluha telefonních hovorů, posílání textových zpráv (SMS), nebo využívání fotoaparátu). Takto vybudovaný systém umožňuje vývojářům vytvářet bohatší a soudržnější zážitky pro uživatele. Android je postaven na otevřeném jádře Linux a používá vlastní virtuální stroj, který byl navržen tak, aby optimalizoval paměť a hardwarové prostředky v mobilním prostředí. Tato platforma se bude dále vyvíjet, protože vývojářská komunita pracuje společně na vytváření inovativních mobilních aplikací.“ (17, str. 13)

2.2.2 iOS

Největším konkurentem Androidu je iOS. Jedná se o mobilní operační systém od firmy Apple Inc., dříve také známý jako iPhone OS. Tento systém byl původně navržen pro použití v chytrém telefonu od stejné firmy – dnes již všem velmi dobře známého iPhone. Postupem času se však rozšířil i do dalších zařízení jako hudební přehrávač iPod Touch nebo tablet iPad (2).

Stejně jako systém Android je architektura iOS tvořena jednotlivými vrstvami, konkrétně jsou čtyři – Core OS, Core Services, Media, Cocoa Touch. Podstata je prakticky stejná – vývojář spolupracuje pokud možno s co nejvyšší vrstvou, a při potřebě využít funkci nižší vrstvy, musí použít frameworky na různých úrovních. Nejzajímavější vrstvou je pak Cocoa Touch. Tato vrstva obsahuje frameworky pro vytváření aplikací, definuje vzhled aplikace, poskytuje základní infrastrukturu a podporu klíčových technologií jako je např. multitasking, notifikace, dotykový vstup a další (2).



Obrázek 2: Vrstvy systému iOS, Zdroj: (2)

2.2.3 Windows Phone

Windows Phone je mobilní operační systém od společnosti Microsoft. Tento systém nahrazuje předchozí Windows Mobile a na rozdíl od něj je cílený na uživatele jako takového a ne na firemní sféru. S příchodem Windows Phone Microsoft představil nové uživatelské rozhraní Metro, které vzbudilo vlnu nadšení i značnou nevoli u konzervativních fanoušků Microsoftu. Důležitým milníkem pro Windows Phone bylo

oznámení výrobce mobilních telefonů Nokia použít Windows Phone jako operační systém pro všechny vyráběné smartphony (22).

2.2.4 Blackberry

Systém BlackBerry je z uvedené čtveřice mobilních operačních systémů ten nejméně známý, tedy alespoň v našich končinách. Původním záměrem tvůrců (Mike Lazaridis a Douglas Fregin) bylo sestrojít jednoduché, bezpečné a efektivní zařízení, které umožní zaměstnancům psát, přijímat a odesílat emaily mimo kancelář. Toto zařízení bylo pojmenováno BlackBerry (20).

BlackBerry OS je, jak již dává tušit, proprietární operační systém tohoto zařízení. BlackBerry OS disponuje mnoha speciálními funkcemi – především trackball, trackwheel, trackpad a dotykovou obrazovkou. BlackBerry nabízí nativní podporu pro firemní email pomocí MIDP, který přináší snadnou synchronizaci s Microsoft Exchange účtem, Lotus Domino a emailem, kontakty, poznámkami a dalšími organizačními prvky. Na tento systém mohou vývojáři také díky uvolněným API programovat aplikace (20).

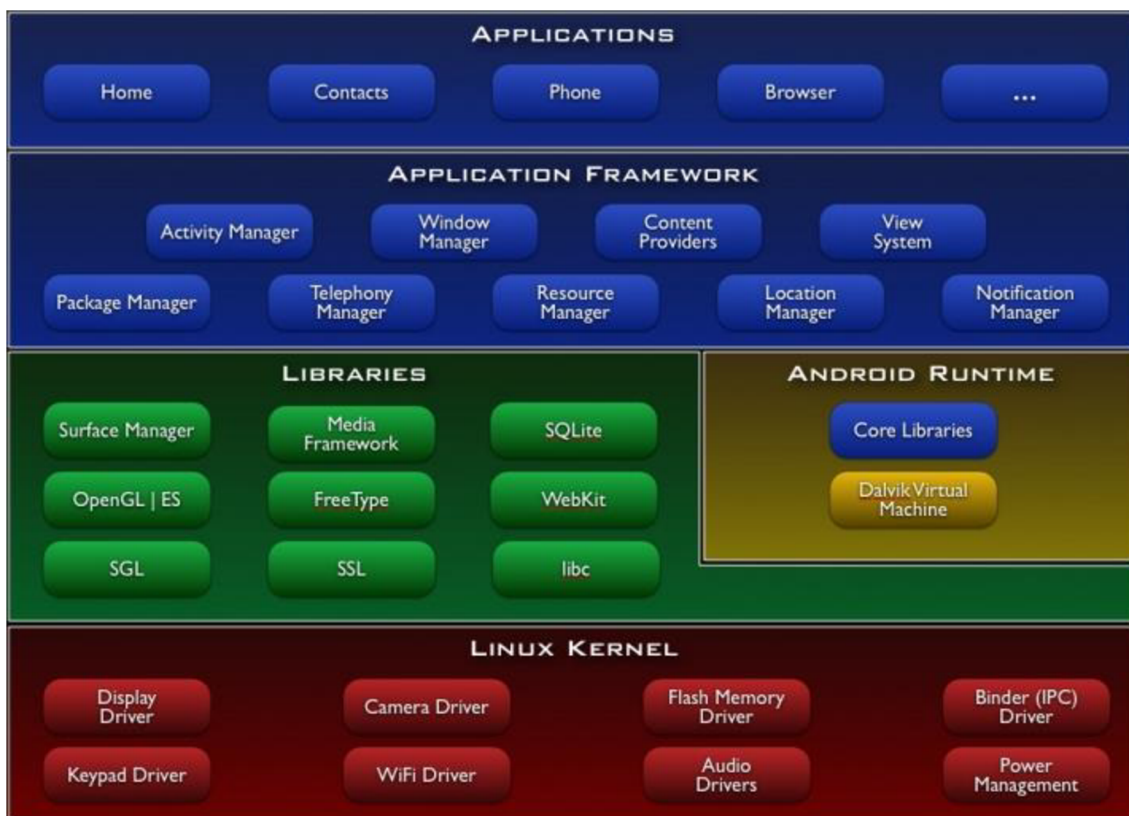
2.3 Životní cyklus Android aplikace

Operační systém Android nemá, na rozdíl od jiných prostředí, žádnou kontrolu nad svým vlastním životním cyklem. Z tohoto důvodu musí být součástí každé aplikace naslouchání na změny stavu aplikace a jejich adekvátní řešení, přičemž zvláštní pozornost je třeba věnovat jejímu případnému předčasnému ukončení (17, str. 39).

Při výchozím nastavení běží každá Android aplikace ve svém vlastním procesu, který je spuštěn samostatnou instancí virtuálního stroje Dalvik. Za běhu aplikace je také řešeno zacházení s pamětí a řízení procesů. Jelikož je každá aplikace samostatný proces, úzce tedy souvisí s životním cyklem procesu. Proces aplikace je řízen systémem v závislosti na aktuálním stavu systémové paměti. Pokud nastane taková situace (a nastává často), že systém má nedostatek paměti, je Android nucen ukončit některé méně důležité procesy. Rozhodnutí o ukončení daného procesu je spojeno se stavem jednotlivých komponent procesu (17, str. 40).

2.4 Architektura OS Android

Architektura systému je znázorněna na obrázku níže. Celkem se skládá z pěti vrstev. Každá vrstva provádí různé operace a vystupuje víceméně samostatně. V praxi však dochází ke spolupráci jednotlivých částí a vrstvy tímto nejsou mezi sebou striktně odděleny (17, str. 17).



Obrázek 3: Architektura systému Android, Zdroj: (10)

2.4.1 Bootloader

Tato část netvoří přímo součást operačního systému, a proto ani není znázorněna na obrázku. Pro spuštění systému je však naprosto klíčová. Jedná se o samostatný program sloužící primárně k nahrání jádra operačního systému do operační paměti. Kromě tohoto může taky zvládat jiné operace (10).

2.4.2 ROM

Ani ROM paměť není součástí operačního systému. Je to část paměti, do které lze zapisovat pouze ve zvláštním režimu a je zde uložen operační systém. Má několik důležitých součástí (10):

- Radio ROM – zde jsou uloženy informace o operátorovi a základní ovladače hardwaru, zejména GSM čipu
- Extended ROM – zde jsou uloženy zvláštní programy či úpravy od výrobce
- CID Lock (Carrier ID) – mechanismus, který brání nahrání neoficiální ROM – tedy neoficiálně upraveného operačního systému

2.4.3 Kernel

Kernel je nejnižší vrstva architektury a představuje jádro operačního systému. Jeho hlavní úkol je implementace abstrakce mezi použitým hardwarem a softwarem. Má na starosti také kontrolu nad systémem, správu procesů, správu paměti, správu sítí apod. Součástí jsou také ovladače. Operační systém Android nepoužívá své vlastní jádro, ale Linuxové jádro (10; 17, str. 17 - 18).

2.4.4 Libraries

Knihovny zabezpečují základní funkce systému. Lze tu nalézt tzv. Android API, tedy rozhraní pro vývoj aplikací (android.util, android.os. apod.). Mimo těchto zde existují také knihovny napsané v jazyce C/C++, které jsou vývojáři poskytnuty prostřednictvím Android Application Framework. Jedná se například o dobře známé OpenGL (pro práci s grafikou), FreeType (vykreslování písma), SQLite (ukládání a práce s daty) a další (10; 17, str. 18).

2.4.5 Runtime

Tato vrstva slouží pro běh samotných aplikací. Vrstva obsahuje virtuální stroj tzv. DVM (Dalvik Virtual Machine) a současně základní Java knihovny. Virtuální stroj slouží k převodu kódu, ve kterém jsou psané aplikace, na nativní kód. Programovací jazyk Java má samozřejmě svůj vlastní virtuální stroj, ten však nemohl být použit z licenčních důvodů – nebyl vyvíjen jako open source projekt (10; 17, str. 19).

2.4.6 Application Framework

Tato vrstva je pro vývojáře nejdůležitější, obsahuje totiž další knihovny, které jsou tentokrát napsané v Javě a umožňují vývojáři přistupovat k nejrůznějším službám, které pak mohou být využity přímo v aplikacích, a ty jim následně dovolí např. přistoupit na prvky graficko-uživatelského rozhraní, používat hardware zařízení, spouštět jiné aplikace na pozadí apod. (10; 17, str. 19 - 20).

2.5 Základní prvky grafického rozhraní

2.5.1 Aktivita

Aktivita reprezentuje jednu obrazovku uživatelského rozhraní. Uvažujme například aplikaci emailového klienta - jedna aktivita by zobrazovala seznam přijatých emailů, další aktivita by sloužila k vytvoření nového emailu a další například pro čtení daného emailu (5).

Přestože aktivity společně tvoří komplexní uživatelské prostředí, nejsou na sobě vzájemně závislé. Aktivity mohou být navíc spuštěny i jakoukoli jinou aplikací, tedy pokud programátor tuto možnost povolil. Kupříkladu, aplikace Fotoaparát může po pořízení snímku vyvolat emailovou aplikaci za účelem odeslání pořízeného obrázku emailem (5).

2.5.2 Fragment

Fragment reprezentuje chování určité části uživatelského rozhraní aktivity. V rámci jedné aktivity lze zobrazit několik fragmentů a vytvořit tak uživatelské rozhraní s několika panely, zároveň lze jeden fragment použít ve více aktivitách (7).

Fragment si lze představit jako modulární sekci aktivity, která má svůj vlastní životní cyklus, dokáže zpracovávat své vlastní vstupní události a může být přidána či odebrána za běhu aktivity (7).

Fragment musí být vždy přiřazen k aktivitě, jeho životní cyklus úzce závisí na životním cyklu hostované aktivity. Pokud je například aktivita pozastavena (je volána metoda

onPause()), jsou pozastaveny i všechny její fragmenty. Pokud je však aktivita v běžícím stavu, s každým jejím fragmentem lze manipulovat nezávisle (7).

2.5.3 Služba

Služba je komponenta běžící v pozadí aplikace, sloužící pro zpracování náročnějších, dlouho trvajících operací (long runtime), nebo zpracování úkolů vzdálených procesů. Služba nemá žádné uživatelské rozhraní (5).

Základním příkladem může být služba pro přehrávání hudby na pozadí, zatímco uživatel používá jinou aplikaci. Typický způsob použití je také získávání dat z internetu, bez toho aniž by bylo zabráněno uživateli v interakci s právě spuštěnou aktivitou. Služba může být spuštěna komponentou, jako je například aktivita. Služba pak bude provádět svoji činnost, pro kterou byla navržena nebo může být s aktivitou svázána a provádět určitou interakci (5).

2.5.4 Broadcast receiver

Broadcast receiver je komponenta, která odpovídá na systémové broadcasty. Tyto broadcasty mohou být šířeny mnoha komponentami, jejich velké množství generuje samotný systém – například broadcast oznamující vypnutí obrazovky zařízení nebo slabou baterii (5).

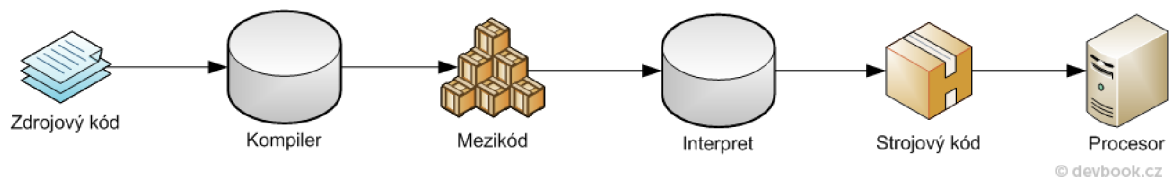
Aplikace mohou vytvářet svoje vlastní broadcasty, například oznámení o úspěšném stažení požadovaných dat a jejich dostupnosti k dalšímu zpracování. Ačkoli Broadcast receiver nedisponuje uživatelským rozhraním, může vytvořit notifikaci v oznamovací oblasti a informovat tak uživatele o nastání určitého stavu. Ve většině případů je Broadcast receiver použit pouze na iniciování další akce, jako je například spuštění služby (5).

2.6 Java

Jelikož je vývoj aplikací pro Android realizován pomocí jazyka Java, je potřeba si uvést základní informace o tomto programovacím jazyku. Jedná se o programovací jazyk

s takzvaným „virtuálním strojem“. Představuje nejmodernější podobu programovacího jazyka, která v sobě spojuje výhody kompilovaného a interpretovaného jazyka (1).

“Zdrojový kód je nejprve přeložen do tzv. mezikódu, kterému se u Javy říká bytecode. Jedná se v podstatě o strojový (binární) kód, který má ale o poznání jednodušší instrukční sadu a přímo podporuje objektové programování. Tento mezikód je potom díky jednoduchosti relativně rychle interpretovatelný tzv. virtuálním strojem (tedy interpretem, v případě Javy je to tzv. JVM - Java Virtual Machine). Výsledkem je strojový kód pro náš procesor.” (1)



Obrázek 4: Překlad zdrojového kódu v jazyce Java, Zdroj: (1)

Tento způsob překlada má mnoho výhod (1):

- **Odhalení chyb ve zdrojovém kódu** – Chyby lze jednoduše odhalit díky kompilaci do bytekódu
- **Stabilita** - Díky tomu, že interpret kódu rozumí, zastaví provádění programu před vykonáním nebezpečné operace a na chybu upozorní
- **Jednoduchý vývoj** – K dispozici jsou moderní datové struktury a knihovny, správu paměti má na starosti garbage collector
- **Rychlost** - Rychlost se u virtuálního stroje pohybuje mezi interpretem a kompilerem. Virtuální stroj již výsledky své práce po použití nezahazuje, ale dokáže je cachovat, sám se tedy optimalizuje při čtenějších výpočtech a může dosahovat až rychlosti kompilera. Start programu bývá pomalejší, protože stroj překládá společně využívané knihovny

- **Málo zranitelný kód** - Aplikace se šíří jako zdrojový kód v bytekódu, není tedy úplně jednoduše lidsky čitelná
- **Přenositelnost** – Program lze spustit na každém počítači, na kterém se nachází virtuální stroj

2.7 API

„API je zkratka anglických slov application programming interface, což znamená rozhraní pro programování aplikací. Tento termín používá softwarové inženýrství v programování. Jde o sbírku procedur, funkcí či tříd nějaké knihovny (ale třeba i jiného programu nebo jádra operačního systému), které může využívat programátor, který knihovnu využívá. API určuje, jakým způsobem se funkce knihovny mají volat ze zdrojového kódu programu; rozhraní knihovny, které se využívá po přeložení programu do binární podoby a během jeho běhu, se nazývá ABI.“(4)

2.8 Metodiky vývoje software

Software lze vyvíjet pomocí několika metod, které je možno rozdělit mezi klasické a agilní. Agilní metodiky jsou navíc v poslední době čím dál více populárnější a jejich správná implementace má za následek dosažení velmi uspokojivých výsledků (16).

Klasické metodiky jsou zaměřeny především na procesy, mají přesné plány a podle těch se řídí. Integrace a testování u těchto metodik se provádí až na konci vývoje, což mnohdy způsobuje různé problémy. Často jsou zaměřeny na tvorbu dokumentů bez přidané hodnoty a jejich revize (16).

Agilní metodiky jsou zaměřené především na lidi. Velice důležitým aspektem je zde komunikace a motivace. Podrobné plány jsou tvořeny pouze pro jednotlivé iterace – kratší časové úseky (maximálně 2 měsíce). Tato metodika je integrována a testována průběžně, počítá se změnami (16).

2.8.1 Klasické metodiky

- **Moderní strukturovaná analýza (MSA)** – metodika používá klasický vodopádový model. Při analýze je vytvářen analytický model, který se skládá z datového modelu a funkční hierarchie – sada diagramů datových toků (diagramů aktivit). Návrh pak bere v úvahu výstup analýzy a podrobí ji transakční a transformační analýze (16).
- **Unifikovaný proces vývoje (UP)** – jedná se o průmyslový standard kostry vývojového procesu. Používá UML notaci (Unified Modeling Language), je řízen požadavky, případy užití a riziky. Tato metodika musí být uzpůsobena pro danou firmu (16).

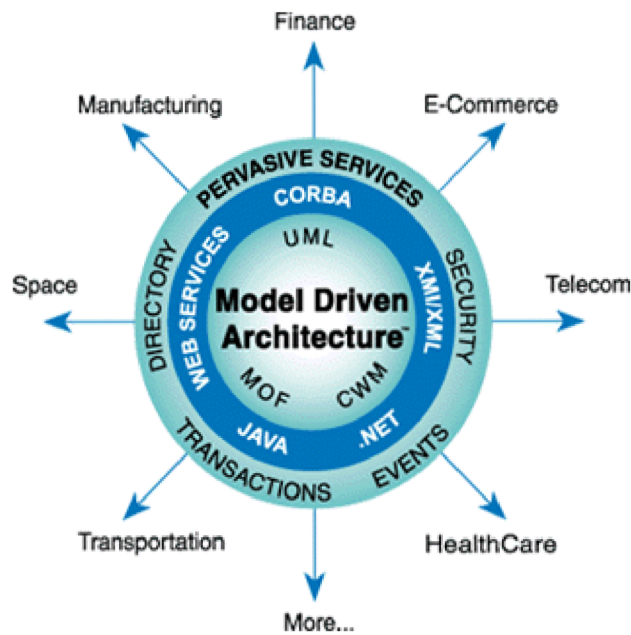
2.8.2 Agilní metodiky

- **Extrémní programování (XP)** – upřednostňuje fungující a intuitivní software před rozsáhlou dokumentací. Tato metodika se snaží o dodržování co nejjednoduššího návrhu a okamžitého testování. Navíc se snaží uživatele dostat do vývojového týmu a získat tak co nejrychlejší zpětnou vazbu (16)
- **SCRUM** – tato metodika nemá žádné týmové hierarchie. Tým musí být sestaven z počtu maximálně 7 osob, ze kterých se volí šéf týmu, tzv. SCRUM master; založen na velmi krátkých iteracích (sprinty), před každou iterací je realizována plánovací schůzka a po každé iteraci retrospekce (16)

2.8.3 Modelem řízený vývoj (MDD, MDA)

MDA = Model Driven Architecture (altern. MDD = Model Driven Development)

Jde o nový přístup k vývoji. Hlavní myšlenkou tohoto modelu je oddělení podnikové a aplikační logiky od technologické platformy. Platformově nezávislé modely aplikací nebo integrovaných podnikových funkcionalit (založeny na UML a dalších OMG modelovacích standardech). Mohou být realizovány použitím MDA virtuálně na jakékoli platformě, ať už otevřené či uzavřené, včetně Web Services, .NET, CORBA, J2EE a dalších (14).



Obrázek 5: Modelem řízený vývoj, Zdroj: (14)

2.9 Životní cyklus softwaru

Softwarové aplikace prochází následujícími fázemi (14):

- Nápad
- Neformální specifikace
- Katalog požadavků
- Formální specifikace
- Dekompozice
- Řešení komponent (modulů)
- Implementace komponent
- Testování komponent
- Integrace komponent do celku
- Testování celku (akceptační test)
- Instalace
- Provoz a údržba

2.10 Fáze tvorby SW produktu

Tvorba SW produktu má následující fáze (14):

- **Úvodní studie** – v této části jsou sbírány veškeré požadavky. Součástí je i feasibility study (studie proveditelnosti); takto provedená studie slouží jako podklad ke konečnému rozhodnutí, zda má projekt smysl.
- **Analýza (specifikace)** – tato fáze je velice důležitá a neměla by být opomenuta, cílem je vytvoření co nejpřesnější specifikace produktu.
- **Návrh (design)** – tato část se zabývá dekompozicí systému na komponenty, které lze naprogramovat.
- **Implementace (konstrukce)** - v této fázi již dochází k samotné realizaci komponent definovaných v návrhu a jejich sestavení do výsledného produktu.

2.11 Vývojové nástroje

Vývoj aplikací pro Android probíhá v takzvaném Host-Target vývojovém prostředí. To znamená, že prostředí, ve kterém je aplikace vyvíjena a prostředí, ve kterém je nakonec spuštěna jsou naprosto odlišná. Vývoj aplikace probíhá na počítači, který obsahuje vývojové prostředí a testování aplikace probíhá již na mobilním zařízení (17, str. 23).

Aplikace mohou být testovány na přístroji opatřeným systémem Android nebo v jeho emulátoru. Použití emulátoru usnadňuje vývoj a je tak pro mnoho vývojářů základním testovacím zařízením. Závěrečné testování probíhá na reálném zařízení se systémem Android (17, str. 23).

Pro vývoj aplikací pro Android jsou potřeba následující nástroje (17, str. 23):

- Java Development Kit (JDK)
- Android Software Development Kit (SDK)
- Vývojové prostředí
- Android Development Tool (ADT)

2.11.1 Java Development Kit (JDK)

JDK je soubor základních nástrojů a knihoven pro vývoj aplikací a appletů pro platformu Java, a jelikož se aplikace pro Android programují v jazyce Java, je potřeba tento nástroj nainstalovat. Základní součástí JDK je Java Runtime Environment (JRE), jenž slouží pro spuštění aplikací i vývojových nástrojů, dále překladač, debugger atd. (17, str. 23).

2.11.2 Android Software Development Kit (SDK)

SDK je balíček vývojových nástrojů pro vytváření aplikací pro určité operační systémy, hardware platformy nebo herní konzole. SDK obsahují knihovny API, dokumentaci, ukázky využití spolu se zdrojovými kódy atd. Obsah Android SDK je totožný s jinými SDK a navíc zde např. najdeme knihovny Javy potřebné pro tvorbu výkonných mobilních aplikací pro OS Android, další nástroje pro vývoj a ladění aplikací a také emulátor – virtuální mobilní zařízení fungující na stolním počítači (17, str. 25).

2.11.3 Vývojové prostředí

Pro naprogramování aplikace pro Android postačí klasický textový editor nebo jakékoli prostředí umožňující programovat v Javě (Java IDE). Mezi takové se řadí například NetBeans, Oracle JDeveloper, BlueJ nebo Eclipse. Pro efektivitu a komfort vývojáře je potřeba zvolit si to správné prostředí, které bude pokud možno po všech stránkách vyhovovat (17, str. 29 – 30).

Vývojové prostředí Eclipse je primárně určeno pro programování v Javě, díky jeho výborné vlastnosti je možno jej rozšířit o podporu dalších programovacích jazyků nebo o vizuální nástroj pro tvorbu graficko-uživatelského rozhraní (17, str. 29 – 30).

2.11.4 Android Development Tool (ADT)

Vývojové prostředí ovšem nestačí pouze vybrat a nainstalovat, je potřeba jej také správně nastavit. Pro potřeby vývoje pro Android je nutné propojit prostředí Eclipse s Android SDK. Propojení je reprezentováno pluginem Android Development Tool (ADT), jenž rozšiřuje možnosti Eclipse a umožňuje tím rychlou tvorbu Android projektů. Díky ADT programátor získá výkonné integrované prostředí s editorem

vizuálních aplikací, vlastními XML editory, ladícími panely a tvorbou APK balíčků pro distribuci aplikace (17, str. 32).

2.12 Jsoup

Jsoup je knihovna programovacího jazyka Java určená pro práci s HTML kódem. Jsoup nabízí velmi příjemné API pro parsování a extrahování dat z HTML stránek za pomoci DOM, CSS a jQuery metod (9).

Současně implementuje HTML5 specifikaci a parsuje kód do stejného DOM formátu jako moderní webové prohlížeče (9).

Jsoup může být použito konkrétně pro (9):

- Parsování HTML stránek z internetu (URL), ze souboru nebo z proměnné (String)
- Pomocí Jsoup lze najít a extrahovat data z HTML kódu, při použití DOM a CSS selektorů
- Manipulaci s HTML elementy, atributy a textem
- Pročištění HTML kódu za účelem zabránění cross-site scripting útoku
- Čištění HTML kódu pro jeho správné zobrazení

Jsoup je šířena pod MIT licenci a lze ji zcela volně používat, modifikovat či vylepšovat.

„Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:“(9)

3 Analýza současného stavu

Tato kapitola je zaměřena na analýzu informačního systému Fakulty podnikatelské a porovnání časů potřebných k přihlášení do informačního systému školy skrze webový prohlížeč na PC a mobilním telefonu. V rámci rozhodování pro který operační systém je nejvhodnější aplikaci navrhnout, byl proveden průzkum popularity mobilních OS a jeho výsledky jsou analyzovány v této kapitole.

V této části práce najdete také analýzu nabídky mobilních aplikací jiných univerzit v České republice, porovnání jejich funkční a uživatelské stránky (user interface) a analýzu populárnosti těchto aplikací mezi studenty. Výsledkem analýzy bude stanovení požadavků na mobilní aplikaci pro studenty FBM, tedy jaké funkce by měla aplikace obsahovat, na základě požadavků studentů a výše provedených analýz.

3.1 Informační systém Fakulty podnikatelské

Informační systém Fakulty podnikatelské je z hlediska designu na první pohled účelový a přehledný. Horní a levá část stránky slouží jako navigace, přičemž v horní části jsou umístěny záložky pro přístup k různým komponentám. Levá strana pak poskytuje přístup k jednotlivým položkám zvolené komponenty. Toto rozdělení je velmi přehledné a intuitivní.

Obsahová stránka systému je rozdělena do šesti částí, které jsou zobrazeny jako karty (záložky). Z pohledu studenta Fakulty podnikatelské je nejdůležitější kartou "Student", kde najde přehled o všech informacích souvisejících s jeho studiem. Nachází se zde především nejnovější aktuality z různých předmětů, elektronický index nebo sekce pojmenovaná "Registrace termínů", která slouží pro přihlašování na jednotlivé termíny zkoušek a rozvrhy. Tyto záložky představují nejdůležitější části informačního systému a ten k nim přistupuje mnohem častěji než k ostatním.

3.1.1 Navigační prvky

Pokud student zvolí jednu ze záložek, oba navigační panely se zbarví dle podkladové barvy záložky a v levém navigačním menu se zobrazí jednotlivé položky dostupné pro zvolenou komponentu.



Obrázek 6: Hlavní navigační panel, Zdroj: (21)



Obrázek 7: Postranní navigační panel, Zdroj: (21)

3.1.2 Rozvrhy

Sekce rozvrhů je zobrazena pomocí tabulky. Každý záznam v rozvrhu tvoří zkratka předmětu, jméno učitele, číslo místnosti, typ (C = cvičení, P = přednáška), údaj o pravidelnosti předmětu, tedy zda je vyučován v lichém nebo sudém týdnu, či každý týden. Student na první pohled pozná, zda se jedná o cvičení nebo přednášku. Přednášky jsou podbarveny zelenou barvou, zatímco cvičení mají podbarvení žluté či modré, podle toho zda jsou cvičení vyučována v počítačové učebně nebo v klasické.

den (FP)	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	7:00 7:50	8:00 8:50	9:00 9:50	10:00 10:50	11:00 11:50	12:00 12:50	13:00 13:50	14:00 14:50	15:00 15:50	16:00 16:50	17:00 17:50	18:00 18:50	19:00 19:50	20:00 20:50
Po	IrmP (P) E342 V prof. Ing. Karel Rais, CSc., MBA, dr. h. c. (1-10L) - V									IrmP (C1) P157 L Mgr. Vladěna Štěpánková (1-10L) - V				
Út					IciP (P) P381 V doc. Ing. František Bartes, CSc. (1-10L) - V					IciP (C1) P284 S Ing. Jaroslav Rompelt (1-10L) - V				
St	IbiP (P) P381 V Ing. Jiří Kříž, Ph.D., Ph.D. (1-10L) - V						IbiP (C2a) P263 V Mgr. Vladěna Štěpánková (1-10L) - V							
Čt					IimP (P) E342 V doc. Ing. Miloš Koch, CSc. (1-10L) - V								IimP (C1) P157 V Ing. Romana Křížová (1-10L) - V	



Obrázek 8: Zobrazení rozvrhu, Zdroj: (21)

3.1.3 Aktuality z předmětu

Pomocí aktualit sdílejí učitelé dokumenty potřebné pro výuku, hodnocení dílčích úloh, či ostatní informace pro studenty, jako je například absence učitele na příští hodině. Prezentace dat je opět řešena pomocí tabulky, do které jsou "dynamicky" přidávány řádky a sloupce Javascriptem (respektive všechny řádky i sloupce jsou načteny již při prvním zobrazení stránky, ovšem jsou zobrazeny až poté, co student klikne na jeden z předmětů).

letní semestr

zkratka	předmět	dostupné aktuality
[IbiP]	Business Intelligence	0
[IciP]	Competitive Intelligence	3

Název aktuality: **Zadání seminární práce 1**
Datum vystavení: 09.02.2015 Datum platnosti: 30.06.2015
Příloha:
Povinná osnova seminární práce,
 Povinná osnova_seminární práce.docx (129896 b)
 VZOR - CI_VAS_makro.doc (409088 b)
>> zobrazit celou aktualitu >>
doc. Ing. František Bartes, CSc.

Obrázek 9: Aktuality z předmětu, Zdroj: (21)

3.1.4 Registrace termínů - zkoušek

Registrace termínů je z hlediska studenta klíčová součást systému, jelikož v průběhu zkouškového období se svádí veliký boj o nejlepší termíny na zkoušku. Každý termín má vlastní div element s třídou definovanou na základě stavu. Jedná-li se například o propadlý termín, předchozí registrovaný apod. Z pohledu studenta Fakulty podnikatelské je největším problémem absence oznamování vypsání nových termínů, což může mít za následek, že se student nestihne včas přihlásit na požadovaný termín.

IfmP - Finanční management pro informatiky - závěrečná zkouška	
čt 22. ledna 2015 15:00 až 17:00 - Termín IM 3	✓
hodnocení termínu: známka D (čt 22. ledna 2015, prof. Ing. Mária Režňáková, CSc.)	
zkoušející: prof. Ing. Mária Režňáková, CSc., místnost: P384 [?] , přihlášení: min. 0, reg. 43, max. 50	
ImasP - Metody aplikované statistiky - závěrečná zkouška	
st 14. ledna 2015 12:00 až 14:00 - řádný termín zkoušky	✓
hodnocení termínu: známka D (st 14. ledna 2015, Ing. Karel Doubravský, Ph.D.)	
zkoušející: Ing. Karel Doubravský, Ph.D., místnost: IO/P381 [?] , přihlášení: min. 0, reg. 9, max. 50	
IsmanP - Strategický management - závěrečná zkouška	
čt 8. ledna 2015 11:00 až 13:00 - termín	✓
hodnocení termínu: známka D (čt 8. ledna 2015, doc. Ing. Robert Zich, Ph.D.)	
zkoušející: doc. Ing. Robert Zich, Ph.D., místnost: IO/P381 [?] , přihlášení: min. 0, reg. 19, max. 100	

Obrázek 10: Registrace termínů, Zdroj: (21)

3.1.5 Elektronický index

Poslední analyzovanou částí je elektronický index. Jedná se o přehled všech předmětů, které má student v aktuálním semestru zapsány. Zobrazení dat je řešeno pomocí tabulky.

předmět										hodnocení			
zkr.	název předmětu	jazyk	typ	kr.	uk.	eL.	záp.	datum záp.	body	zn.	datum zn.	termín	absl.
Akademický rok: 2014/2015													
zimní semestr, 2. ročník, 2. stupeň													
IfmP ⁰²	Finanční management pro informatiky ⁰²	cs	P	3	zk,zá		✓	17.12.2014		D	22.01.2015	2	✓
ImasP ⁰²	Metody aplikované statistiky ⁰²	cs	P	5	zk,zá	fm	✓	15.12.2014		D	14.01.2015	1	✓
IsmanP ⁰²	Strategický management ⁰²	cs	P	3	zk,zá		✓	16.12.2014		D	08.01.2015	1	✓
letní semestr, 2. ročník, 2. stupeň													
IdsP ⁰²	Diplomní seminář ⁰²	cs	P	10	kl								✗

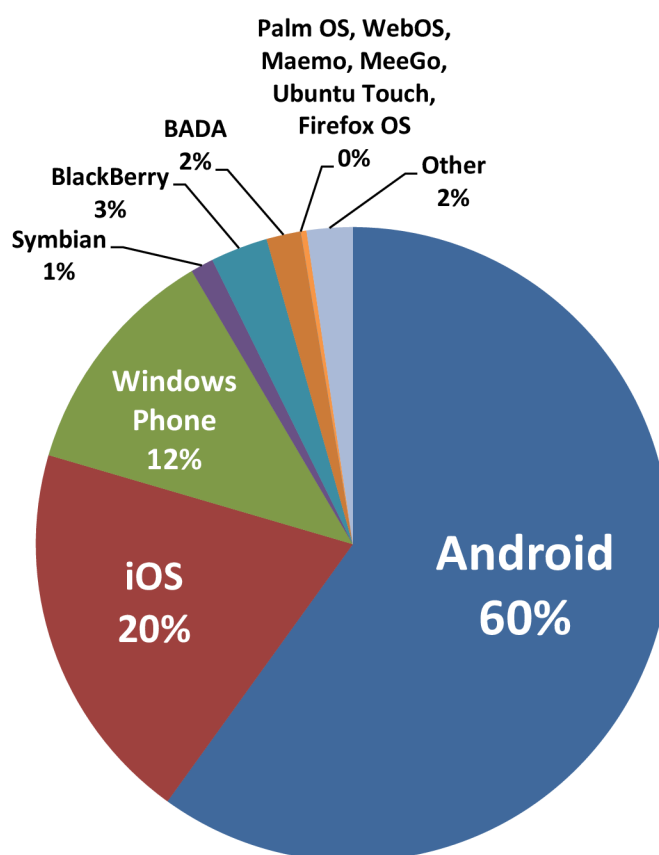
Obrázek 11: Elektronický index, Zdroj: (21)

3.2 Průzkum typu používaných OS

Pro získání konkrétní představy o populárnosti operačního systému Android mezi studenty Fakulty podnikatelské byl proveden průzkum. Do tohoto průzkumu se mohli zapojit všichni studenti FP, kteří byli vyzváni k vyplnění jednoduchého dotazníku, publikovaném na stránce fakulty na sociální síti Facebook.

V dotazníku byla položena jediná otázka, a to jaký operační systém má dotazovaný nainstalovaný na svém mobilním zařízení. Dotazovaný byl vyzván k zaškrtnutí typu OS na všech vlastněných zařízeních, jelikož spousta studentů může vlastnit buď více mobilních telefonů, nebo mobilní telefon a navíc tablet.

Dotazník byl vytvořen pomocí Google Forms a odpovědělo na něj celkem 312 studentů. Průzkum byl proveden v rámci jednoho týdne, konkrétně 11. až 17. května.



Obrázek 12: Podíl jednotlivých typů OS mezi studenty FP, Zdroj: vlastní zpracování

Výsledky tohoto průzkumu prokazují, že většina studentů Fakulty podnikatelské ve svých zařízeních používá operační systém Android. Na druhém místě se umístil iOS, jenž je statisticky nainstalován na každém pátém zařízení. Nema­lý podíl má i Windows Phone, celých 12%, což je vcelku překvapivý výsledek. Systém BlackBerry používá 3% dotazovaných uživatelů. Velkým překvapením je BADA, který je jen o jeden procentní bod za BlackBerry. Dále nevyvíjený operační systém Symbian ještě žije na některých zařízeních, ale pouze u 1 % uživatelů. Ostatní uvedené systémy měly tak malé zastoupení, že nezískaly ani jeden procentní bod. Skupina „Other“ s velkou pravděpodobností reprezentuje zařízení, které se neřadí mezi „chytře“, tedy nemají nainstalován žádný z uvedených operačních systémů.

Tabulka 3: Výsledky provedeného průzkumu popularity jednotlivých OS mezi studenty FP, Zdroj: vlastní zpracování

OS	Počet uživatelů	%
Android	205	60%
iOS	67	20%
Windows Phone	41	12%
Symbian	4	1%
BlackBerry	10	3%
BADA	6	2%
Palm OS	0	0%
WebOS	0	0%
Maemo	0	0%
MeeGo	0	0%
Ubuntu Touch	0	0%
Firefox OS	1	0%
Other	8	2%
Celkem	342	100%

3.3 Porovnání doby přístupu do IS VUT z různých zařízení

Informační systém fakulty podnikatelské je přístupný z adresy <https://www.vutbr.cz/login/login>. Po úspěšném ověření studentova identifikačního čísla a hesla je možné prohlížet elektronický index, rozvrh, termíny zkoušek, přistupovat

k aktualitám z předmětu. Tento proces na běžném PC netrvá příliš dlouho, ovšem pokud je přístup realizován z mobilního telefonu, může zabrat až minuty.

Přitom jde většinou o stále stejnou proceduru – zadat identifikační číslo a heslo, rychle přejít na elektronický index a zkontrolovat zda jsou vypsané termíny zkoušek, či zkontrolovat známku. Student se chce co nejrychleji dostat k potřebným informacím, ale je zdržován načítáním obrázků a dalších prvků stránky, opakovaným zadáváním svého hesla a znovu načítáním stránky, což znemožňuje rychlou cestu k informacím, které požaduje. Účelem této analýzy je porovnání časové a úkonové náročnosti při přihlášení z PC a z mobilního telefonu. Tyto výsledky budou posléze porovnány s dobou přihlášení a přístupem k požadovaným informacím za použití vytvořené aplikace.

3.3.1 Postup testování

Při provádění testu byl dodržen vždy stejný sled následujících úkonů, simulující studenta, který chce zjistit známku z absolvované zkoušky:

- Zadání adresy do adresního řádku
- Vyplnění VUT loginu a hesla
- Přihlášení
- Přejít do IS kliknutím na "Intraportál"
- Přepnutí na kartu "Student"
- Zvolení "Elektronického indexu"

Test má pouze orientační charakter, proto nebudou brány v úvahu žádné statistické ukazatele. Z uvedených úkonů je zřejmé, že uživatel musí podstoupit nejméně 8 úkonů (interakcí), než se dostane k požadovaným informacím. Každé měření bylo provedeno třikrát, konečný výsledek v tabulce 4 je dán zprůměrováním těchto hodnot.

Jelikož se (zejména u mobilních zařízení) odezva systému a celkový výkon zařízení odráží v počtu spuštěných aplikací, procesů, probíhajících operací apod., bylo měření provedeno vždy po restartu daného zařízení s jedinou potřebnou spuštěnou aplikací - webovým prohlížečem.

3.3.2 Použitá zařízení

PC - klasický stolní počítač s dvoujádrovým procesorem a 4GB RAM paměti, připojený kabelem napřímo do modemu UPC sítě s garantovaným připojením 40Mbps / 10Mbps

ZTE Skate Pro - jedná se o telefon s jednojádrovým procesorem a 512 MB RAM paměti

Gigabyte GSmart Roma R2 - telefon s dvoujádrovým procesorem a 1 GB RAM paměti

3.3.3 Výsledky testu

Tabulka 4: Porovnání doby přístupu, Zdroj: vlastní zpracování

Zařízení	Výsledek testu [s]
PC	28
ZTE Skate Pro (Wifi)	92
ZTE Skate Pro (HSPDA)	98
ZTE Skate Pro (Edge)	114
Gigabyte GSmart Roma R2 (Wifi)	72
Gigabyte GSmart Roma R2 (HSPDA)	76
Gigabyte GSmart Roma R2 (Edge)	88

Z výsledků vyplývá, že doba potřebná k získání stejných informací pomocí telefonu je výrazně vyšší, než doba potřebná pro získání těchto informací za použití PC. Při použití low-end telefonu tato doba může být i trojnásobně vyšší. Je nutno také brát v úvahu počet interakcí, které musí uživatel podstoupit, aby se dostal k požadovaným informacím, ve výše uvedeném příkladu je to nejméně 8. Tento postup tak vyžaduje pozornost studenta a doba jeho trvání se může výrazně lišit.

3.4 Mobilní aplikace českých univerzit

Při prvotním průzkumu trhu to vypadalo, že tato část ani nevznikne, protože mobilních aplikací českých univerzit je jako šafránu. Prakticky jedinou aplikací, která by se mohla

nazývat plnohodnotnou aplikací pro studenta, je UniApps. Ostatní aplikace jsou většinou zaměřené velmi jednostranně, na jednu jedinou operaci. Mezi tyto aplikace patří například objednávání jídel v menze, přihlašování na zkoušky nebo přihlašování do školní sítě. Pokud tedy lze hovořit o opravdu komplexní aplikaci, kterou by student očekával, je zde pouze UniApps.

3.5 UniApps v2 Beta

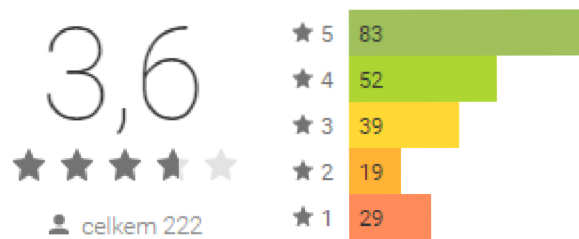
Tato aplikace je rozšířením univerzitních informačních systémů AiS2, eVzdělávání, UIS a IS/STAG. Je tedy zřejmé, že pouze univerzity s těmito informačními systémy mohou používat tuto aplikaci. V rámci České republiky jde o následující univerzity (18):

- Vysoká škola obchodní v Praze
- Škoda Auto vysoká škola
- Mendelova univerzita v Brně

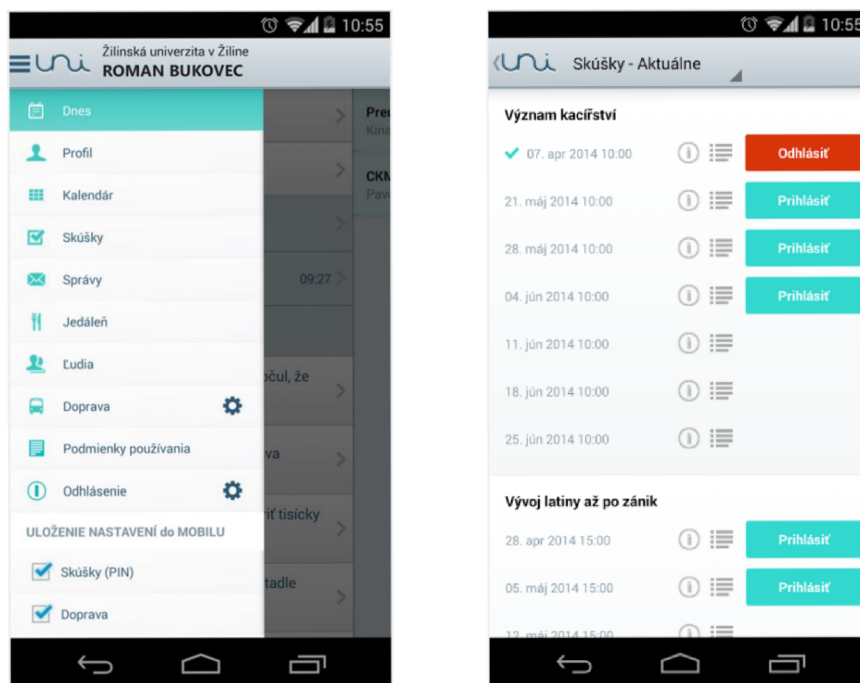
Aplikace na první pohled zaujme jednoduchým, přehledným a velmi povedeným designem. Navigace je řešena pomocí Navigation Drawer, tedy vysunovací panel z levé strany. Tento způsob navigace preferuje většina moderních aplikací díky své nenáročnosti, zpětné podpoře pro starší zařízení a elegantnosti (18).

Do funkční výbavy této aplikace patří prohlížení rozvrhu, přihlašování/odhlašování ze zkoušek, kalendář, vyhledávání osob spojených s fakultou. Vedle těchto funkcí pro studenty UniApps nabízí i komunikačního klienta a vyhledávání cesty do školy za použití hromadných dopravních prostředků (18).

Dle hodnocení uživatelů jde o úspěšnou aplikaci, která ovšem nefunguje zcela bezchybně při spolupráci se všemi informačními systémy. Objevují se zde tedy negativní komentáře poukazující na nefunkčnost některého z modulů pro danou univerzitu (18).



Obrázek 13: Hodnocení uživatelů UniApps na Google play, Zdroj: (18)



Obrázek 14: Aplikace UniApps v2 Beta, Zdroj: (18)

3.6 SWOT analýza IS VUT Fakulty Podnikatelské

3.6.1 Silné stránky

- Přehlednost systému
- Logické členění navigačních prvků
- Přiměřená podrobnost informací

3.6.2 Slabé stránky

- Absence mobilní verze
- Absence notifikací při změně v systému

3.6.3 Příležitosti

- Implementování mobilní verze
- Odlišení od konkurenčních univerzit vytvořením vlastní mobilní aplikace

3.6.4 Hrozby

- Žádné hrozby plynoucí z provedené analýzy nebyly identifikovány, avšak absence mobilní aplikace může vést v delším časovém horizontu k zaostání za konkurencí

4 Vlastní návrhy řešení

Z předchozí analýzy poměrně zřetelně vyplývá, že současný systém je dostačující, ovšem nevhodný pro přístup z mobilního zařízení. Pokud budeme uvažovat přístup z tabletu, není to velký problém, avšak pro přístup z mobilního telefonu se již musíme obrnit trpělivostí. Jde o neustálé opakované zadávání přihlašovacích údajů, přibližování a oddalování pohledu obrazovky a hledání požadovaného modulu v navigaci.

4.1 Identifikace potřeb studentů, požadavků na aplikaci

Z hlediska studentů lze každodenně používané funkce shrnout do několika bodů. Nejčastěji jsou využívány následující moduly:

- Zobrazení rozvrhu
- Registrace na zkoušky
- Kontrola elektronického indexu
- Kontrola aktualit z předmětu

Tyto moduly jsou bezpochyby nejpoužívanější na celém informačním systému fakulty podnikatelské. Samozřejmě nelze tvrdit, že ostatní moduly nemají pro studenta informační hodnotu, jsou stejně tak důležité jako ty ostatní, jen je zkrátka student nevyužívá tak často.

Příkladem takového modulu může být registrace závěrečné práce. Je nesmírně důležitý pro studenty v posledním ročníku, ale rozhodně nepatří mezi ty, které by musel mít student vždy po ruce a přistupovat k nim z mobilního telefonu.

Pokud se tedy zaměříme na aktuální potřeby studenta, výše uvedené 4 moduly postačují k jeho spokojenosti.

4.2 Volba operačního systému

Trh s mobilními zařízeními je rozmanitý a potenciální zákazník má na výběr z mnoha modelů mobilních zařízení v různých cenových kategoriích, v různých variantách barev, velikostí, ale i operačních systémů. Volba operačního systému, pro který bude aplikace

realizována, je nesmírně důležitým krokem. Úspěch aplikace je měřen počtem jejich uživatelů, a proto je nejrozumnější zvolit nejrozšířenější mobilní operační systém používaný na Fakultě podnikatelské.

Dle provedené analýzy je tímto systémem Android od firmy Google. Výhoda tohoto systému (a zároveň nevýhoda) je jeho modifikovatelnost, čehož výrobci telefonů hojně využívají a do základního systému doinstalují vlastní aplikace, či aplikace partnerských společností. Výjimkou není ani grafická úprava samotného systému, případně implementace dodatečných funkcí. Pro výrobce je nasazení Androidu do svých telefonů výhodným krokem, jedná se totiž o velmi oblíbený a vyspělý systém, který je navíc zdarma (přestože výrobci musí dodržovat určitá pravidla).

Nevýhoda takové otevřenosti je však v nejednotnosti systému na zařízeních od různých výrobců. Stejná verze tedy nemusí nutně znamenat stejnou záruku funkčnosti. Systém je navíc velmi složitě aktualizován na novější verzi, protože výrobce často nepočítá s vydáním této nové verze pro daný model telefonu (nelze použít verzi od Googlu, výrobci často provádí modifikace za účelem propagace vlastních aplikací). Uživatel má tak často smůlu a jediným řešením je koupě nového telefonu (nebo nahrání novějšího firmware neoficiální cestou).

I přes zmíněné nevýhody, se systém Android u nás těší největší oblibě a aplikace tak bude vyvíjena právě pro tento operační systém.

4.3 Popis návrhu a architektury vytvořené aplikace

4.3.1 Barevná paleta

Barevná paleta aplikace se řídí dle nejnovější specifikace navrhování vzhledu Material Design, kterou vytvořila společnost Google. Tento koncept je založen na dvou základních barvách – Primary Color a Accent Color. Každá z těchto barev je jednou z několika elementárních barev barevné palety, přičemž musí být dodrženo, že vybrané barvy musí být vzájemně dostatečně kontrastní.

Každá z těchto barev má přesně definován účel, na který by měla být použita a neměla by být použita žádným jiným způsobem. Pokud toto pravidlo není dodrženo, design neodpovídá specifikacím a aplikace ztrácí svůj glanc.

- **Dark Primary Color** – od verze Android 5.0 Lollipop je barva oznamovací oblasti (status bar) nahrazena Dark Primary Color danou aplikací (v tomto případě tedy temně fialová). S nižší verzí API zůstává barva oznamovací oblasti dle systémového nastavení.
- **Primary Color** – barva by měla být použita na střední a větší prvky uživatelského rozhraní. Jedná se o základní barvu aplikace, do této barvy je také vyveden Action Bar.
- **Light Primary Color** – světlejší odstín Primary Color může být použit jako podkladová barva pro některé prvky, přičemž světlejší odstín evokuje menší důležitost zobrazené informace nebo prvku.
- **Text / Icons** – barva textu v tmavých ikonách.
- **Accent Color** – barva ze základní palety, která musí být dostatečně kontrastní vůči Primary Color. Tato barva je určena pro použití především u primárních komponent uživatelského rozhraní, jako jsou tlačítka, přepínače apod. Barva by však neměla být použita na větší prvky nebo větší bloky textu.
- **Primary Text** – barva základního textu v aplikaci.
- **Secondary Text** – světlejší odstín barvy základního textu, vyjadřuje menší důležitost informace či větší podrobnost.
- **Divider Color** – barva „oddělovače“, např. oddělení řádku v ListView.

Barevná paleta použitá v aplikaci:

#512DA8 DARK PRIMARY COLOR	#673AB7 PRIMARY COLOR	#D1C4E9 LIGHT PRIMARY COLOR	#FFFFFF TEXT / ICONS
#4CAF50 ACCENT COLOR	#212121 PRIMARY TEXT	#727272 SECONDARY TEXT	#B6B6B6 DIVIDER COLOR

Obrázek 15: Barevná paleta použitá v aplikaci, Zdroj: (13)

4.3.2 Oprávnění (Permissions)

Každá aplikace před instalací do mobilního zařízení musí získat od uživatele oprávnění k použití jednotlivých funkcí telefonu. Pokud toto oprávnění od uživatele nezíská, nebude moci být zaručena její úplná funkcionální, a tudíž nebude ani nainstalována. Tento systém byl zaveden za účelem větší kontroly uživatele nad všemi nainstalovanými aplikacemi v telefonu a jejich přístupu k citlivým údajům, internetu apod. S rostoucí popularitou mobilní platformy vždy rostou i potenciální hrozby, jelikož se stává atraktivnější i pro možné útočníky.

Navržená aplikace potřebuje pro svůj bezproblémový chod následující oprávnění:

- INTERNET
- ACCESS_NETWORK_STATE
- WAKE_LOCK
- VIBRATE
- RECEIVE_BOOT_COMPLETED
- WRITE_EXTERNAL_STORAGE

Aplikace stahuje aktuální data z internetu, a tedy potřebuje oprávnění `ACCESS_NETWORK_STATE`, které slouží k testování, zda je k dispozici síťové připojení a samozřejmě také `INTERNET` pro samotné stahování dat z internetu.

Pro správné fungování služeb na pozadí aplikace žádá oprávnění `WAKE_LOCK`, které dané službě umožní zabránit přechodu zařízení do úsporného režimu dříve, než dokončí svůj proces. Zároveň umožňuje probudit telefon z úsporného režimu a provést zadaný úkol vyvolaný nastaveným alarmem.

Oprávnění `VIBRATE` je použito při upozorňování uživatele na změny v aktualitách, indexu nebo termínech, tedy při vytváření notifikace.

`WRITE_EXTERNAL_STORAGE` je použito při stahování příloh z aktualit, jelikož tyto nemají omezenou velikost, jsou stahovány na externí uložení.

Posledním potřebným oprávněním je `RECEIVE_BOOT_COMPLETED`. Toto oprávnění je nezbytné pro automatickou registraci alarmu po restartování zařízení.

Aplikace tedy po restartu nemusí být znovu spuštěna a uživatel přesto bude upozorňován na změny.

4.3.1 Online / Offline mód

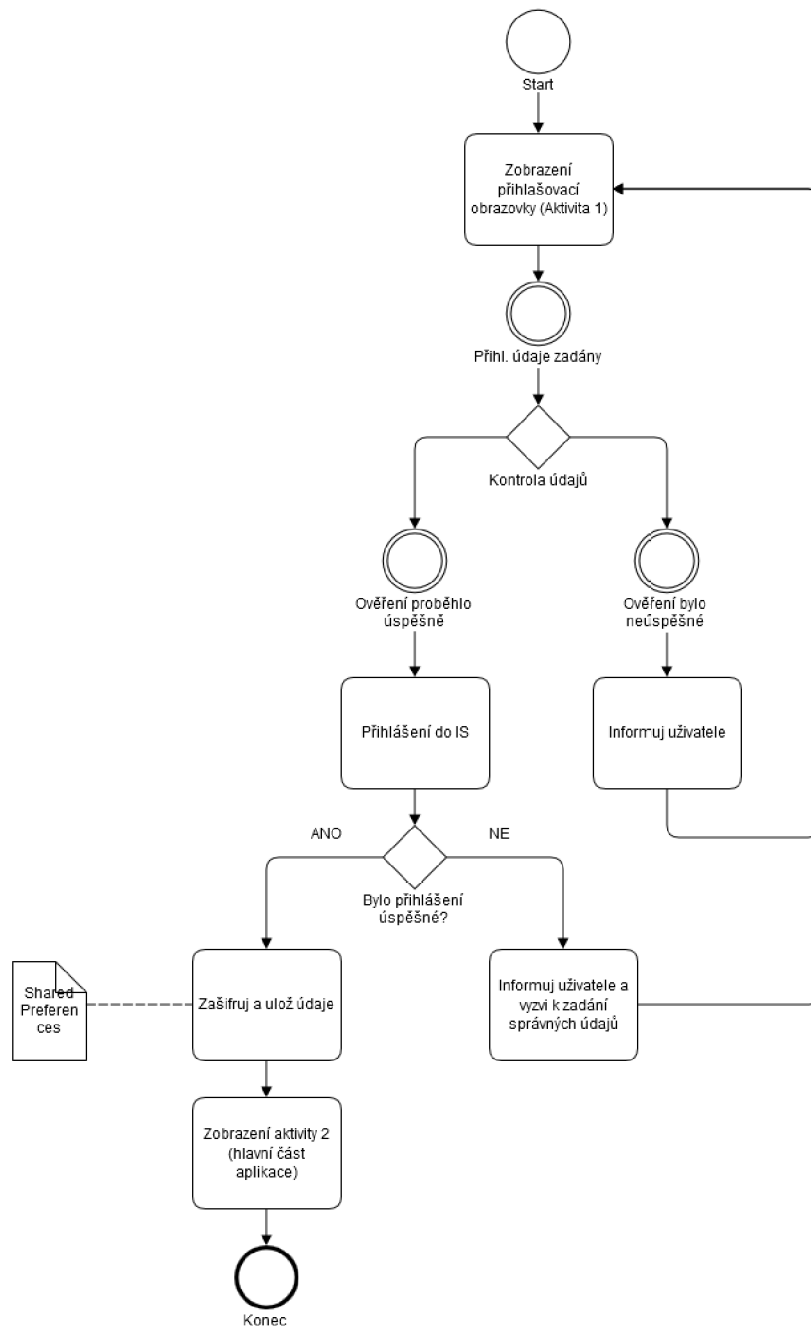
Při návrhu aplikace je potřeba rozhodnout, zda bude vyžadováno k jejímu běhu internetové připojení či nikoliv. Tomuto rozhodnutí je totiž nutno přizpůsobit její architekturu, která se bude lišit u výhradně offline a výhradně online aplikace.

Při navrhování aplikace byl kladen důraz na možnost použití i v případě, že uživatel právě není připojen k žádné síti. V těchto situacích jsou použita data získaná v předchozím online běhu. Uživatel tak má přístup ke všem funkcím, které nevyžadují připojení do IS VUT. Je zřejmé, že uživatel nebude moci stahovat přílohy v aktualitách, přihlašovat se na zkoušky nebo kontrolovat vystavení nových aktualit. Bude však schopen zjistit svůj rozvrh, podívat se do svého indexu nebo na seznam zkoušek.

Všechny moduly jsou zpřístupněny i bez internetového připojení a ve stavovém řádku (status bar, viz další kapitoly) je vždy uveden datum a čas poslední aktualizace. Při pokusu o aktualizaci dat je uživateli předložena zpráva o chybějícím internetovém připojení.

4.3.2 Proces přihlášení

Na níže uvedeném diagramu je znázorněn proces ověření údajů při přihlášení. Po spuštění aplikace je uživatel vyzván k zadání přihlašovacích údajů. Po jejich zadání a stisku tlačítka „Přihlásit“ nejprve dochází k prvnímu kroku ověření údajů, kdy systém kontroluje, zda nejsou pole prázdná. Pokud toto ověření dopadne úspěšně, systém se pokusí se zadanými údaji přihlásit do informačního systému VUT. Toto ověření probíhá na základě analýzy přijatých cookies, jež se liší v případě neúspěšného přihlášení. Pokud tedy aplikace obdrží ty správné cookies, zašifruje přihlašovací údaje a uloží je do privátního souboru pomocí SharedPreferences, což je knihovna určená právě pro ukládání nastavení aplikace. Uživatel již tedy nikdy nebude muset zadávat údaje znovu, pokud je bude aplikace potřebovat, stačí je rozšifrovat a jsou připraveny k použití. Po uložení údajů je spuštěna druhá aktivita, která tvoří hlavní větev programu.

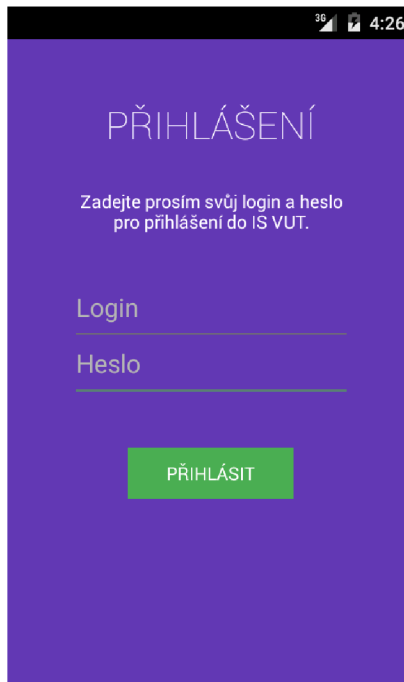


Obrázek 16: Procesní diagram přihlášení uživatele, Zdroj: vlastní zpracování

4.3.1 Přihlašovací obrazovka

Cílem přihlašovací obrazovky je jednoduchost a přehlednost přihlášení, což vede ke snadné orientaci. Obsahuje pouze dvě pole, pro zadání loginu uživatele a hesla. Pod těmito poli je tlačítko, kterým se uživatel přihlásí. Samozřejmostí je ověření,

zda nejsou pole prázdná a následný pokus o přihlášení pro zjištění správnosti zadaných údajů.



Obrázek 17: Přihlašovací obrazovka, Zdroj: vlastní zpracování

4.3.2 Základní stavební prvky aplikace

Každá aplikace pro operační systém Android může být implementována dvěma základními způsoby. První způsob používá pro každou uživatelskou obrazovku specifickou aktivitu, která má svůj vlastní layout. Aktivita je základní prvek uživatelského rozhraní, který má svůj životní cyklus, svoje dispoziční rozmístění (layout) a obsahuje všechny ostatní prvky uživatelského rozhraní. Způsob tvorby aplikace pomocí několika na sebe navazujících aktivit ovšem již neodpovídá současným požadavkům na aplikaci.

Od API verze 11 (odpovídá Android 3.0) se pro zobrazení jednotlivých obrazovek distribuovaných uživateli používají fragmenty. Fragment je samostatné View (prvek uživatelského rozhraní), který má svůj životní cyklus (více o fragmentech lze nalézt v kapitole Teoretická východiska). Fragmenty mají především využití při používání aplikace na zařízeních s různými velikostmi obrazovek – např. na tabletech, které mají

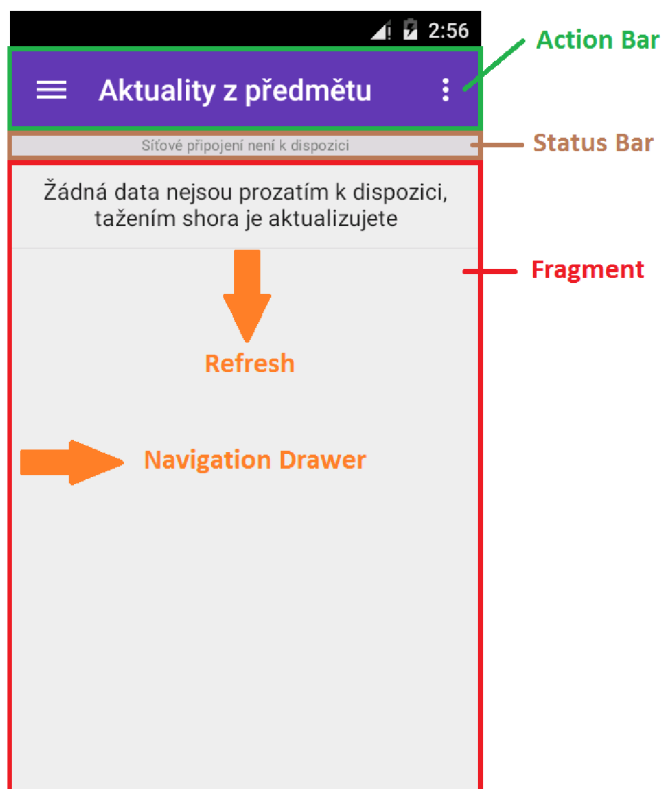
mnohem větší displej než mobilní telefony. Pokud je k dispozici dostatek místa, může být zobrazeno i několik fragmentů současně, což je na rozdíl od aktivit velká výhoda.

Navržená aplikace obsahuje celkem dvě aktivity a několik fragmentů. Při prvním spuštění aplikace se uživateli zobrazí první aktivita, tedy přihlašovací obrazovka, kde musí uživatel vyplnit svoje uživatelské jméno a heslo. Přihlašovací obrazovka neobsahuje žádné fragmenty, má vlastní statický layout, který je aplikován přímo na aktivitu. Z této aktivity se spouští aktivita druhá, podstatně bohatší na počet uživatelských prvků. Ta obsahuje navigační prvek v podobě vysouvacího postranního panelu (Navigation Drawer), který se používá v moderních mobilních aplikacích. Tento panel je po celou dobu aplikace skrytý a nezabírá žádné místo na obrazovce, uživatel jej může zobrazit pomocí gesta – přejetím prstu z levého okraje do středu obrazovky, nebo pomocí ikony v levé horní části aplikace. Podobný způsob navigace se používá v mnoha aplikacích, takže většině uživatelů je tento způsob ovládání velmi dobře znám.

Tato aktivita obsahuje i tzv. Action Bar, což je horní panel (Toolbar) vyplněný barvou Primary Color. Tento panel je viditelný po celou dobu životního cyklu aplikace, tedy alespoň pokud je na popředí, a nese název fragmentu, který je právě zobrazen. Pod tímto panelem je úzký pruh, tzv. Status bar, který obsahuje jednoduché textové pole pro informování uživatele o procesech na pozadí. Toto pole je využíváno například aktualizací službou, která zde zobrazuje svoji návratovou hodnotu. Status bar také využívají fragmenty, které zobrazují datum a čas poslední aktualizace dat.

Pod tímto polem je místo vyhrazené pro jednotlivé fragmenty. Každý z fragmentů má svůj vlastní layout uzpůsobený pro informaci, kterou chce uživateli poskytnout. Většina fragmentů v aplikaci je typu ListFragment, tedy zobrazuje položky v jednotlivých řádcích pod sebou.

Na obrázku jsou vyobrazeny výše popisované části uživatelského rozhraní. Oranžovou šipkou jsou znázorněny akce, které budou vyvolány po provedení gesta ve směru šipky. Při potažení levého okraje do středu obrazovky se zobrazí postranní navigační panel, tzv. Navigation Drawer. Druhým gestem je potažení fragmentu shora dolů, tato akce vyvolá aktualizaci dat a jejich následné zobrazení. Tato gesta budou podrobněji popsána v následujících kapitolách.



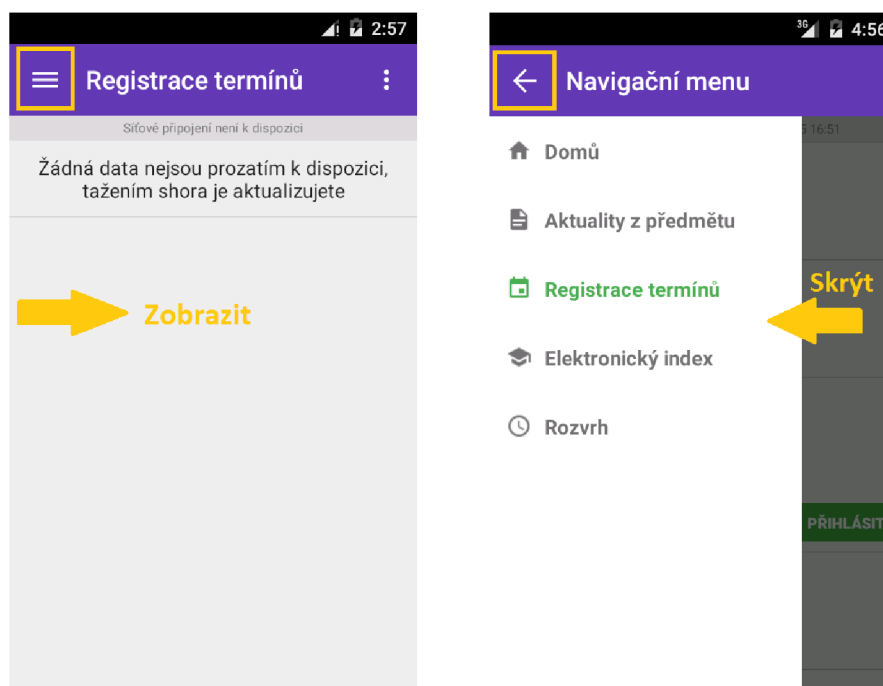
Obrázek 18: Popis jednotlivých komponent a gest, Zdroj: vlastní zpracování

4.3.3 Navigační panel (Navigation Drawer)

Hlavní navigace aplikace je řešena pomocí Navigation Drawer panelu. Tento panel je po celou dobu běhu aplikace neviditelný a nezabírá na obrazovce žádné místo. Zobrazit jej lze dvěma způsoby – pomocí gesta tažením z levého okraje do středu, nebo klepnutím na ikonu „hamburgeru“ v levé horní části aplikace. V obou případech se zobrazí stejný navigační panel, jenž je zobrazen na obrázku č. 19.

Každá položka navigace má vedle popisu i vlastní ikonu. Aktivní položka je zvýrazněna barvou Accent Color, tuto barvu přejímá i ikona dané položky. Po zvolení některé z položek se panel automaticky skryje a načte se odpovídající fragment. Skrytí celého panelu lze udělat opět dvěma způsoby – klepnutím na šipku v levé horní části nebo gestem přejetím z pravého okraje panelu k levému okraji obrazovky, tak jak je to naznačeno na obrázku č. 19.

Tento způsob navigace je standardem ve většině moderních aplikací. Pro většinu uživatelů tento způsob ovládání není ničím novým, a tudíž není potřeba žádného vysvětlování či tutoriálu při prvním spuštění.



Obrázek 19: Navigační panel, Zdroj: vlastní zpracování

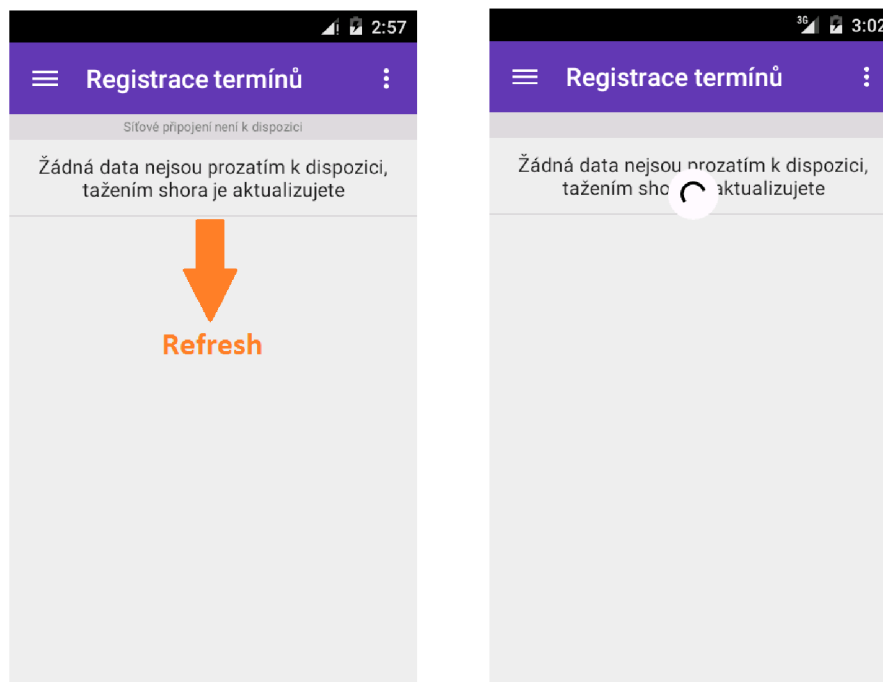
4.3.4 Aktualizace dat

Operace týkající se připojování k webu a stahování nových dat nemohou být prováděny v hlavní vlákne programu, jelikož doba potřebná k připojení ke stránce se může velmi lišit v závislosti na rychlosti připojení. Pokud by byla prováděna v hlavní vlákne, program by musel stále na něco čekat, což z uživatelského hlediska rozhodně není správný přístup.

Z tohoto důvodu jsou všechny operace související se síťovou aktivitou prováděny pomocí služby (konkrétně `IntentService`) nezávislé na samotné aplikaci, což prakticky znamená, že služba může být spuštěna i přesto, že aplikace samotná spuštěná není. Tento přístup je výhodný zvláště v případě, kdy je potřeba provádět některé operace nezávisle na aplikaci, jako je tomu v tomto případě. Aplikace v pravidelných intervalech spouští tuto službu za účelem kontroly změn provedených v portále, aby mohla uživatele v případě změny informovat.

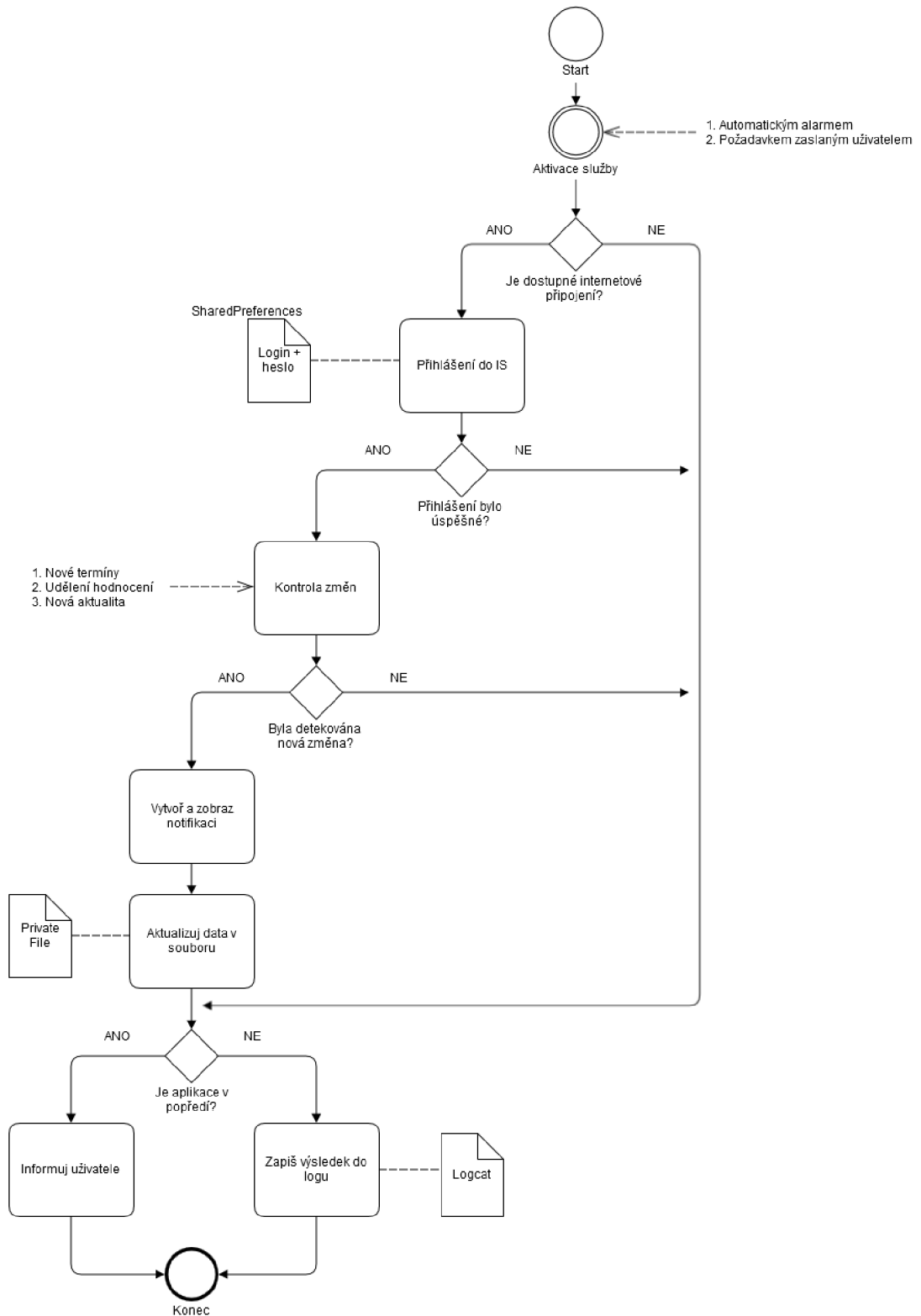
Další výhodou vyplývající z nezávislosti služby na životním cyklu aplikace je dokončení právě prováděných aktivit i přesto, že uživatel aplikaci náhle ukončí. Mohou nastat situace, kdy uživatel zahájí proces aktualizace dat, ale například kvůli pomalému připojení aplikaci náhle ukončí. Pokud by byla aktualizace prováděna v hlavním běhu programu, aktualizace by byla ukončena v nedefinovaném stavu. Díky této službě je ovšem zaručeno dokončení operace i přes ukončení aplikace. Uživatel může tak být dodatečně informován o změnách, zatímco prováděl úplně jiné úkony v jiné aplikaci.

Aktualizace dat může být spuštěna dvěma způsoby – manuálně uživatelem pomocí gesta, nebo automaticky pomocí nastaveného alarmu. Nezávisle na způsobu spuštění je aktualizace VŽDY prováděna pomocí služby.



Obrázek 20: Aktualizace dat, Zdroj: vlastní zpracování

4.3.5 Služba na pozadí



Obrázek 21: Procesní diagram aktualizací služby, Zdroj: vlastní zpracování

Na diagramu je popsána služba na pozadí, která obstarává veškeré aktualizace dat. K aktivaci služby může dojít buď při požadavku uživatele o aktualizaci dat, nebo po aktivaci alarmem. V rámci svého běhu služba nejdříve kontroluje, zda je dostupné internetové připojení. Pokud totiž nemá uživatel zapnutá data nebo není připojen k bezdrátové síti, služba se ukončí. Pokud je připojení k dispozici, služba rozšifruje přihlašovací údaje a pokusí se s nimi přihlásit. Po úspěšném přihlášení jsou kontrolovány změny, a to konkrétně: kontrola nových aktualit, kontrola nově vypsáných termínů a kontrola změny v elektronickém indexu.

Pokud je nalezena jakákoli změna, je vytvořena notifikace (viz následující kapitola) a zobrazena uživateli. Notifikace obsahuje jméno předmětu, ve kterém byla změna nalezena, např. „Byly vypsány nové termíny z předmětu IfmP“ (obrázek č. 22). Zároveň jsou zapsány všechny změny do souboru a jsou tak ihned dostupné uživateli, aniž by musel data stahovat znovu.

V posledním kroku se uživateli zobrazí zpráva o úspěšném / neúspěšném pokusu o aktualizaci a případně příčině chyby. Aplikace rozlišuje chyby při připojení k webové stránce (nejčastěji timeout), chyby při parsování (např. stránka s rozvrhem neobsahuje žádné předměty) a chyby při přihlášení (špatný login nebo heslo). Tato zpráva se uživateli zobrazí v části Status bar, ale pouze pokud je aplikace v popředí, tzn. je spuštěna a uživatel s ní pracuje.

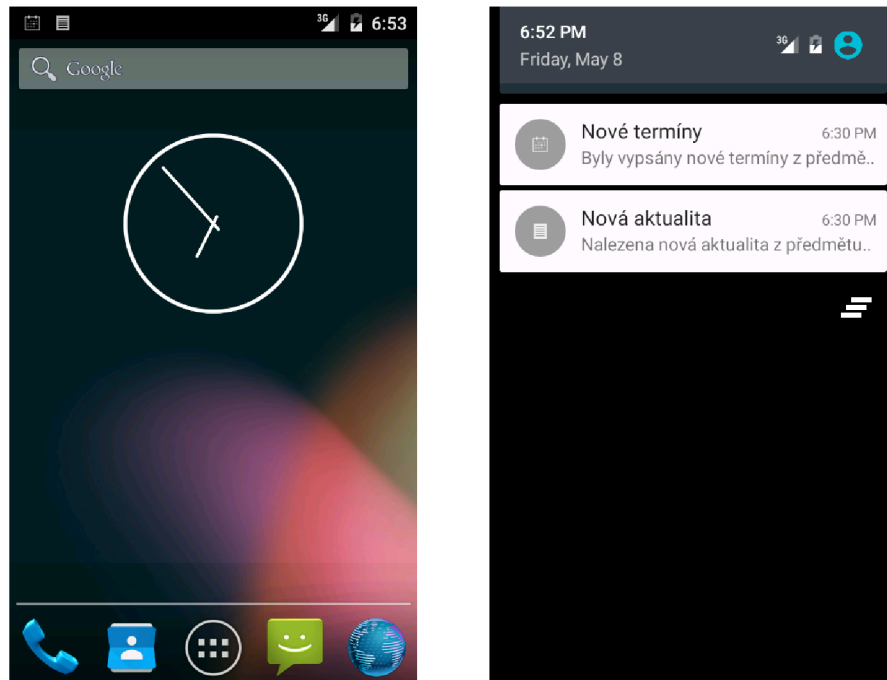
Pokud aplikace v popředí není, informace se zapíše pouze do logu a služba se ukončí.

4.3.6 Notifikace

Navržená aplikace se v pravidelných intervalech automaticky připojuje do informačního systému VUT za účelem zjištění změn od posledního přihlášení. Tento automatický systém sledování změn funguje pro moduly aktuality z předmětu, registrace termínů a elektronický index. Student se tak nemusí několikrát za den ručně připojovat do informačního systému pro kontrolu aktualit a na začátku zkouškového období bude automaticky upozorněn na nově vypsané termíny a rovnou z aplikace má možnost se i přihlásit. Zároveň se ihned dozví o udělení zápočtu a hodnocení zkoušky, to vše bez nutnosti jakékoli interakce. Vše co potřebuje je mít povolené internetové připojení.

Notifikace jsou spouštěny alarmem, který je zaregistrován v AndroidManifest souboru aplikace (soubor se základními parametry aplikace). Tento alarm má nastavené opakování po určité době, takže se automaticky spustí několikrát za den po uplynutí stanoveného intervalu. Pro šetření baterie není opakování nastaveno na přesně definovaný časový úsek, ale spouští se společně s ostatními alarmy, čímž se minimalizuje počet „probuzení“ (wake up events) a tedy i spotřeba celého zařízení.

Alarm spustí službu v pozadí, která zkontroluje změny, případně je zapíše do souboru a vytvoří notifikaci. Pokud je aplikace na popředí, pošle zobrazené aktivitě zprávu o výsledku, aby mohla být ihned zobrazena uživateli.



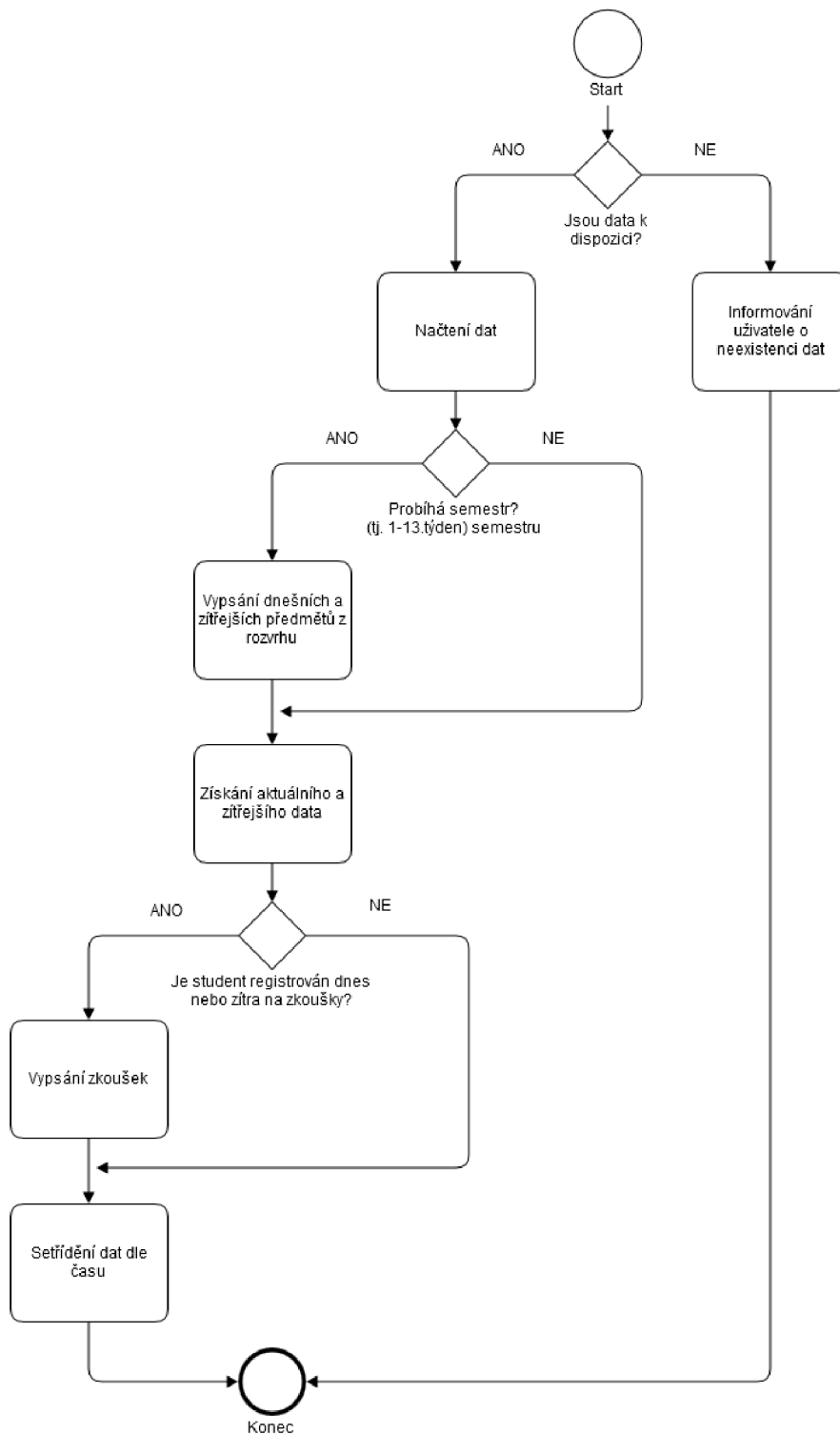
Obrázek 22: Ukázka notifikace, Zdroj: vlastní zpracování

4.4 Jednotlivé funkční části aplikace

4.4.1 Aktivity (obrazovka Domů)

Aktivity jsou základní fragment, který je zobrazen ihned při prvním startu aplikace. Tato obrazovka je základním přehledem o nadcházejících aktivitách, přičemž aktivita zde může být chápána jako účast na přednášce, cvičení či plánovaná zkouška. Cílem

tohoto modulu je poskytnutí základních informací, které student pravděpodobně potřebuje mít po ruce, aniž by se potřeboval někam připojovat.



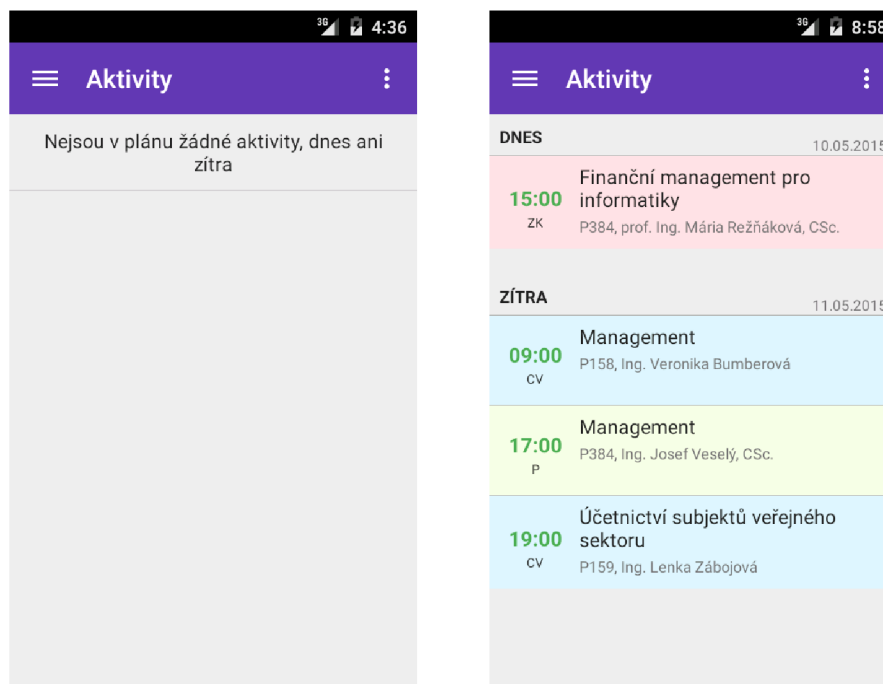
Obrázek 23: Procesní diagram úvodní obrazovky, Zdroj: vlastní zpracování

Aplikace nejdříve zkontroluje, zda již byla v minulosti stažena nějaká data, ze kterých by se daly nejbližší aktivity studenta přečíst. Pokud jsou k dispozici, proběhne jejich načtení a kontrola, zda aktuální datum odpovídá rozmezí začátku a konce semestru, jinak řečeno, zda semestr probíhá. Při probíhajícím semestru jsou přidány do pole dat předměty, které odpovídají aktuálnímu nebo zítřejšímu datu. Dále je získáno aktuální a bezprostředně následující datum za účelem kontroly registrace studenta na zkoušky. V případě, že jsou některé zkoušky zaregistrovány, jsou také přidány do pole dat. V posledním kroku jsou už jen data seříděna dle času a předložena uživateli, jejich konečná podoba je zobrazena na obrázku č. 24.

Aktivity jsou tvořeny na základě již stažených dat, takže pro zobrazení není potřeba žádné internetové připojení. V listu jsou zobrazeny všechny aktivity pro aktuální a nadcházející den. Zahrnují všechny předměty z rozvrhu a přihlášené zkoušky.

Každý záznam obsahuje několik informací. V levé části je vždy zvýrazněný barvou Accent Color čas, kdy daná aktivita začíná. Pod časovým údajem je zkratka aktivity udávající o jaký typ jde – **Z**Kouška, **C**Vičení, **P**řednáška. Na základě tohoto typu je také zvolena barva pozadí pro danou aktivitu. Student tak na první pohled pozná, o jakou aktivitu v daném případě jde. Napravo od časového údaje je název předmětu, kterého se aktivita týká. Pod názvem jsou organizační informace, tedy číslo místnosti a jméno vyučujícího (obrázek č. 24).

Tento modul najde využitelnost především na cestě do školy, kdy je student tlačěn časem a nemusí dlouze procházet informační systém za účelem zjištění čísla místnosti, kde se koná zkouška či cvičení. Důležité je také zobrazení pouze aktivit pro aktuální a nadcházející den. Přispívá to k přehlednosti tohoto modulu a rychlé orientaci. Navíc samozřejmě snižuje počet stažených dat a je dostupný i v offline režimu.



Obrázek 24: Úvodní obrazovka, Zdroj: vlastní zpracování

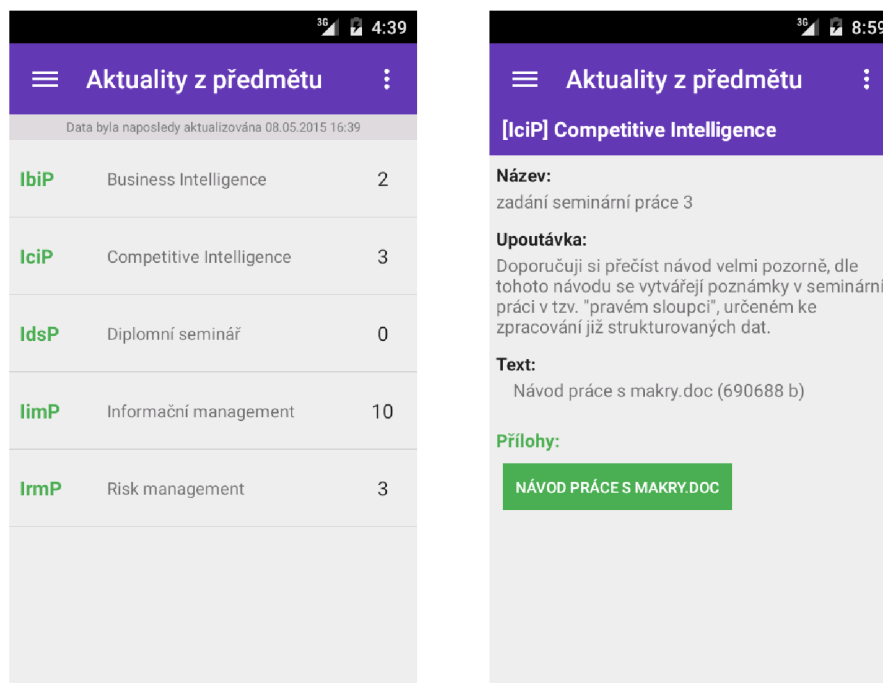
4.4.2 Aktuality z předmětu

Aktuality z předmětu jsou velmi používaný modul v informačním systému. Pomocí něj lze studentům předat důležité informace, například o změně místnosti na nadcházející cvičení nebo zrušení příštího cvičení z důvodu nemoci. U tohoto modulu je tedy důležité, aby se informace ke studentovi dostala včas – o to se v aplikaci stará automatická kontrola nových aktualit.

Struktura modulu v aplikaci téměř odpovídá struktuře v informačním systému. Na úvodní obrazovce je zobrazen seznam předmětů v aktuálním semestru společně s jejich názvem a aktuálním počtem aktualit v každém z nich (viz obrázek č. 25). Po vybrání jednoho z předmětů se zobrazí další fragment, ve kterém jsou zobrazeny názvy jednotlivých aktualit, datum jejich vystavení a indikátor, zda aktualita obsahuje přílohu.

Po vybrání jakékoli z aktualit se její obsah začne načítat z internetu. Tato funkce tedy není dostupná bez funkčního připojení. Pokud má uživatel zakázaná mobilní data, stahování nebude úspěšné a uživatel je upozorněn na chybějící připojení.

Pokud je připojení k dispozici, aktualita stáhne všechny informace do dalšího fragmentu, který již obsahuje celou aktualitu, včetně možnosti stažení příloh do mobilního zařízení. Tyto přílohy mohou být staženy po klepnutí na tlačítko s názvem souboru (viz obrázek č. 25 níže).



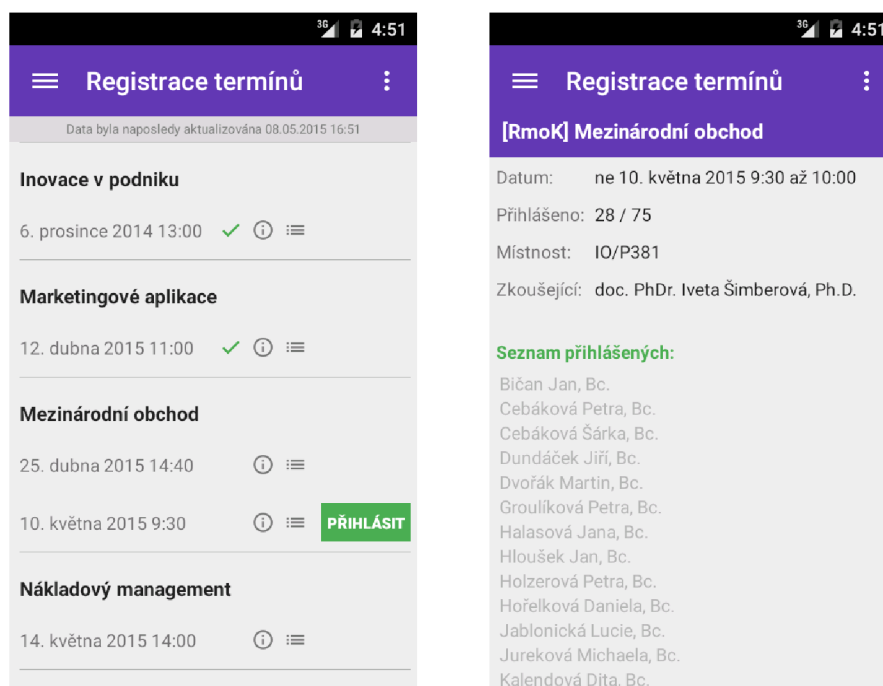
Obrázek 25: Aktuality z předmětu, Zdroj: vlastní zpracování

4.4.3 Registrace termínů

Registrace termínů představuje modul pro registrování zkoušek. Po načtení fragmentu se zobrazují všechny termíny pro aktuální semestr. Na prvním řádku je vždy uveden název předmětu, pro který je zobrazen seznam termínů. Pro lepší přehlednost je u splněných termínů zobrazena ikona fajfky nebo naopak křížku u nesplněných termínů (viz obrázek č. 26).

Na každém řádku jsou také dvě další ikony, napovídající o možnosti zobrazení více informací o daném termínu po klepnutí na daný řádek. Na konci řádku je u některých termínů zobrazeno tlačítko pro přihlášení, respektive odhlášení z termínu, pokud je to možné. Pokud uživatel klikne na toto tlačítko, aplikace se pokusí na zadaný termín přihlásit. Pro tuto akci je tedy potřeba internetového připojení.

Při zobrazení doplňujících informací k termínu jsou načteny informace o přesném času, počtu přihlášených studentů, místnosti, zkoušejícím a ve spodní části fragmentu je také seznam přihlášených studentů, který lze procházet.

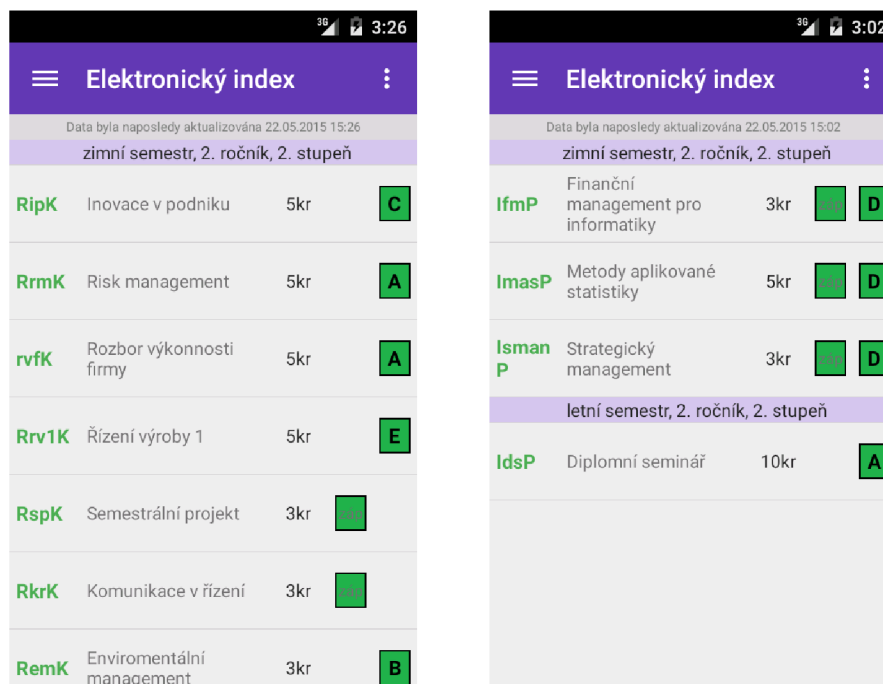


Obrázek 26: Registrace termínů, Zdroj: vlastní zpracování

4.4.4 Elektronický index

U tohoto fragmentu je výmluvný již jeho název. Jedná se o velmi jednoduchý přehled předmětů, udělených zápočtů a výsledků zkoušek. Předměty jsou seřazeny v listu pod sebou, takže student může vidět svoje výsledky z obou semestrů školního roku (viz obrázek č. 27).

Každý řádek začíná zkratkou předmětu, poté jeho celým názvem, počtem kreditů a dvou boxů, které představují zápočet a zkoušku. Pokud je box prázdný, tj. bílý, nebylo prozatím žádné hodnocení uděleno. Pokud již student nějaké hodnocení obdržel, je box červený nebo zelený dle toho, zda byl zápočet či zkouška úspěšně složena. V boxu zkoušky je po udělení hodnocení zobrazena ještě výsledná známka.



Obrázek 27: Elektronický index, Zdroj: vlastní zpracování

4.4.5 Rozvrh

Posledním modulem je rozvrh přihlášeného studenta. Rozvrh je přehledně rozdělen do jednotlivých dnů, a pokud má jeden řádek více předmětů, než se vejde na obrazovku, lze v něm horizontálně posouvat (viz obrázek č. 28). Každý záznam obsahuje zkratku předmětu uvedenou v závorce s typem záznamu, tedy jedná-li o cvičení nebo přednášku. Ve spodní části boxu jsou pak informace o přesném času, místnosti a údaj v jakém týdnu je předmět vyučován (Sudý, Lichý, V - oba).

Rozvrh může být manuálně uživatelem aktualizován gestem pro refresh, tak jako u ostatních fragmentů.

Rozvrh hodin

3G 5:29

Pondělí

UmanP (CV) 9:00 - 11:00 P158, S	UmanP (P) 17:00 - 19:00 P384, V	UusvsP (CV) 19:00 - 21:00 P159, V
--	--	--

Úterý

UfapP (CV) 9:00 - 11:00 P163, L	UlogP (CV) 11:00 - 13:00 P255, S	UlogP (P) 15:00 - 17:00 P158, V
--	---	--

Středa

UzpoP (CV) 15:00 - 17:00 P287, V

Čtvrtek

(P) 1:00 V	UfapP (P) 11:00 - 13:00 P384, V	UzpoP (P) 13:00 - 15:00 E342, V	UusvsP (P) 15:00 - 17:00 E342, V
-------------------------	--	--	---

Pátek

AOP2 (CV) 9:00 - 11:00

Rozvrh hodin

3G 9:35

Data byla úspěšně aktualizována

Pondělí

Kma2P (P) 11:00 - 13:00 P384, V	KuceP (P) 13:00 - 15:00 P384, V
--	--

Úterý

Kmik1P (CV) 13:00 - 15:00 P292, V	Kma2P (CV) 15:00 - 17:00 P287, V	KuceP (CV) 17:00 - 19:00 P283, V
--	---	---

Středa

Kmik1P (P) 07:00 - 9:00 P384, V	KzpP (P) 09:00 - 11:00 P384, V
--	---

Čtvrtek

KvfP (P) 09:00 - 11:00 P384, V	HA2P (CV) 13:00 - 15:00 E215, V	KvfP (CV) 15:00 - 17:00 P287, L
---	--	--

Obrázek 28: Rozvrh hodin, Zdroj: vlastní zpracování

4.5 Porovnání doby přístupu k informacím

V návaznosti na testy provedené v části „Analýza současného stavu“ bude nyní porovnáno získání úplně stejných informací, avšak tentokrát za použití navržené mobilní aplikace.

Postup testu bude totožný, jako je popsán v již zmíněné kapitole. Test bude proveden na různých zařízeních, které jsou každé z jiné kategorie. ZTE Skate Pro je zástupcem low-end kategorie, disponuje pouze nízkým výpočetním výkonem a omezenou pamětí RAM s velikostí 512 MB. Druhým zařízením je modernější GSmart Roma R2 s již dvou jádrovým procesorem a 1 GB pamětí RAM.

Tabulka 5: Porovnání doby přístupu, Zdroj: vlastní zpracování

Zařízení	Výsledek testu [s]	Předchozí výsledek [s]
ZTE Skate Pro (Wifi) – se zadáváním údajů	28	92
ZTE Skate Pro (Edge) - se zadáváním údajů	38	114
ZTE Skate Pro (HSPDA) – se zadáváním údajů	35	98
ZTE Skate Pro (Wifi) – bez zadávání údajů	11	92
ZTE Skate Pro (Edge) – bez zadávání údajů	17	114
ZTE Skate Pro (HSPDA) – bez zadávání údajů	15	98
Gigabyte GSmart Roma R2 (Wifi) – se zadáváním údajů	25	72
Gigabyte GSmart Roma R2 (Edge) – se zadáváním údajů	32	88
Gigabyte GSmart Roma R2 (HSPDA) – se zadáváním údajů	31	76
Gigabyte GSmart Roma R2 (Wifi) – bez zadávání údajů	11	72
Gigabyte GSmart Roma R2 (Edge) – bez zadávání údajů	16	88
Gigabyte GSmart Roma R2 (HSPDA) – bez zadávání údajů	16	76

Z tabulky č. 5 je zřejmé zjednodušení celého procesu získávání informací, které student požaduje. Potřebný čas se výrazně zkrátil na téměř desetinu původního. V tabulce jsou vždy uvedeny případy se zadáváním přihlašovacích údajů a bez jejich zadávání. V prvním případě test simuluje prvotní spuštění aplikace, jelikož v žádném dalším spuštění není potřeba zadávat přihlašovací údaje znovu.

V příkladu byla opět simulována snaha zobrazit elektronický index, ale velmi podobné časy lze očekávat i v ostatních modulech, které jsou v aplikaci dostupné.

Zajímavý je i velmi nepatrný rozdíl mezi typem připojení – tedy WIFI, EDGE, HSPDA. Jelikož jsou stažená data omezena pouze na text a nejsou stahovány žádné obrázky ani jiné grafické prvky, představuje každé přihlášení úsporu stažených dat. Jedno stažení požadované stránky představuje přenesená data v jednotkách, maximálně desítkách kilobajtů.

V předchozím případě byl tento rozdíl mnohem významnější, zejména při připojení pomocí EDGE. Tento rozdíl byl pravděpodobně způsoben stahováním nejen textového obsahu, ale i obrázků a ostatních prvků na stránce.

4.6 Možnosti propagace aplikace

V této kapitole bych chtěl nastínit možné způsoby propagace aplikace. Aplikace má podle prvních ohlasů potenciál stát se velmi populární mezi studenty i bez přílišné propagace. Poskytuje totiž něco, co studenti již dlouho postrádají a co nelze v současné době jiným způsobem získat – upozorňování na změny provedené v informačním systému VUT.

Z pohledu studenta sám dobře znám boj začínající na konci semestru před začátkem zkouškového období, kdy studenti horlivě aktualizují stránky VUT, čekajíc na nově vypsané termíny za účelem přihlášení na ten nejlepší z nich. Tato časově náročná aktivita by mohla studentům nových ročníků používáním aplikace úplně odpadnout.

Co se týče konkrétních kroků na propagaci aplikace, hlavním pilířem by měly být sociální sítě, tedy Facebook. Existuje mnoho stránek jednotlivých skupin, oborů, fakulty, kde lze aplikaci zdarma a velmi účinně zpropagovat, vzhledem k velkému počtu pravidelných návštěvníků. Facebookové stránky jsou totiž pro mnoho studentů neoficiálním zdrojem informací, kde studenti mohou diskutovat a sdílet dokumenty a jsou používány většinou studentů. Vystavení informace na zeď podobných stránek by tak mohlo mít velký vliv na počty stažení.

Z analýzy současného stavu popsané v kapitole 3.4 vyplývá, že aplikace je jedinečná mezi většinou českých univerzit. Takřka jediná podobná aplikace je dostupná pouze pro pár vysokých škol. Ostatní univerzity prozatím mobilní aplikace nemají. Existence této aplikace může vést ke zvýšení prestiže Fakulty podnikatelské.

4.7 Reálné přínosy

Největší přínos navrženého řešení spočívá v ušetření času vynaloženého na získání požadovaných informací. Navržená aplikace navíc nabízí doplňkovou funkcionalitu k informačnímu systému, kterou spousta studentů postrádá. Touto funkcionalitou je upozorňování na provedené změny. Aplikace poskytuje kvalitní základ pro ovládání informačního systému při použití mobilního zařízení, nehledě na množství možných rozšíření, které může v budoucnu implementovat.

Přestože přínosy nemohou být ekonomicky kvantifikovány, aplikace poskytuje reálný přínos pro studenty a aplikuje moderní způsob přístupu k informacím. Na domácím trhu patří k něčemu neobvyklému, drtivá většina univerzit žádnou podobnou aplikaci nenabízí. Je tedy přínosem i pro samotnou Fakultu podnikatelskou, která se zařadí mezi těch několik málo univerzit v ČR, které aplikaci nabízejí.

Teze o nemožnosti ekonomického vyjádření platí i v případě nákladů vynaložených na vývoj aplikace. Na vývoji jsem se podílel pouze já, čili nikdo jiný do něj nezasahoval. Mým nákladem byl čas strávený nad vývojem, zároveň se však tento náklad stal pro mne přínosem, jelikož jsem pročetl mnoho oficiální dokumentace, získal mnoho nápadů a naučil se vytvářet mobilní aplikace pro systém Android dle „Best Practices“ (souhrn doporučených postupů a metod pro vytváření aplikací). Získané vědomosti hodlám dále prohlubovat a zaměřit se na vývoj dalších aplikací pro systém Android.

4.8 Prognóza do budoucnosti, možnosti rozšíření

Aplikace je navržena s jasným cílem usnadnit studentům běžné úkony spojené s informačním systémem a ušetření času. Pokrývá tak základní potřeby studenta, ovšem je zde spousta modulů, které by stály za zapracování do aplikace. Prozatímni vývoj se soustředil na stabilitu a zvládnutí klíčových oblastí. Pokud budu uvažovat konkrétněji, dalším modulem pro rozšíření by bylo vyhledávání osob, jejich rozvrhů, kontaktů, konzultačních hodin apod. Důležitou funkcí je také posílání a přijímání VUT zpráv, tento modul by určitě také našel své využití. Další postup je tedy jasně daný, postupně implementovat chybějící části informačního systému do navržené aplikace.

Budoucí vývoj bude samozřejmě směřovat také na portaci aplikace pro jiné operační systémy, konkrétně pro iOS nebo Windows Phone, což jsou po Androidu nejrozšířenější systémy u nás. Aktuální fáze je ověření funkčnosti, odladění chyb a otestování funkčnosti zvolené architektury, poté se lze zaměřit na zmíněnou portaci aplikace pro jiné platformy.

Závěr

Hlavním cílem této diplomové práce bylo navržení a implementace mobilní aplikace pro zařízení s operačním systémem Android, jejímž cílem je poskytnout studentům možnost provádět některé úkony informačního systému v mobilním telefonu.

V prvním kroku byla provedena analýza informačního systému VUT z pohledu přístupu z mobilního telefonu a byly zjištěny nedostatky, především chybějící verze webových stránek pro mobilní zařízení. Pro studenta je tak velmi nepohodlné pracovat s informačním systémem přes svůj mobilní telefon a tablet, navíc časově nevhodné. Jako další nedostatek byla identifikována absence systému pro upozorňování na provedené změny, jako jsou například vypsání nových termínů nebo zveřejnění nové aktuality.

Oba zmíněné nedostatky byly řešeny navržením mobilní aplikace, která poskytuje studentovi jednoduchý a rychlý přístup ke klíčovým funkcím informačního systému. Mezi tyto klíčové funkce patří prohlížení aktualit z předmětu, kompletní modul registrace termínů, včetně přihlašování na zkoušky a seznamu přihlášených studentů. Dalšími funkcemi je prohlížení elektronického indexu a vlastního rozvrhu.

Aplikace poskytuje i nové funkce. Jednou z nich je přehled plánovaných aktivit, jako jsou plánované přednášky, cvičení, zkoušky, to vše na jedné obrazovce, velmi jednoduchou a přehlednou formou a informacemi aktualizovanými k současnému datu.

Klíčovou funkcí aplikace je automatická kontrola nových aktualit, nově vypsáných zkoušek a udělených hodnocení. Jedná se tedy přesně o funkci, která je v současném informačním systému studenty postrádána. Díky jedinečnosti této funkce a reálné potřebnosti má aplikace velký potenciál rychle se mezi studenty rozšířit.

Při analyzování současného informačního systému VUT byla provedena měření doby potřebné pro přihlášení do IS pomocí webového prohlížeče a získání požadované informace. Tato měření byla následně zopakována při testování vytvořené aplikace a výsledky naplnily, či možná dokonce předčily očekávání. Doba se zkrátila na téměř desetinu původní hodnoty, což je neočekávaně dobrý výsledek. Lze tvrdit, že aplikace je opravdu jednoduchá a přehledná. Její design a architektura byly koncipovány pro rychlý

a snadný přístup k požadovaným informacím, ovládání koresponduje s ostatními aplikacemi pro Android a je tak pro uživatele naprosto intuitivní.

Pro vývoj aplikace byly použity nástroje doporučené samotným výrobcem tohoto operačního systému, firmou Google. Jako vývojové prostředí bylo použito Android Studio, které poskytuje programátorovi v novějších verzích již velký komfort. Aplikace je napsána v programovacím jazyce Java. Při vývoji byly použity postupy a metody doporučené dle „Best Practices“, což zaručuje podporovaný a nejlepší způsob implementace daných řešení. Samotné testování probíhalo na několika fyzických i virtuálních zařízeních, za použití vícero účtů studentů (jimž tímto děkuji).

Největším přínosem aplikace je ušetření času a získávání stále aktuálních informací bez nutnosti interakce uživatele. Pro samotnou univerzitu a fakultu aplikace přináší zvýšení prestiže, svojí aplikací se může pochlubit jen několik málo univerzit (slovy tři) v České republice. Aplikace navíc šetří přenesená data, jelikož nejsou stahovány žádné obrázky ani dodatečné prvky webové stránky.

Jako tvůrce aplikace doufám, že jsem jejím vytvořením zbavil a ještě zbavím vrásek spoustu studentů, kteří se potýkají se stejnými problémy jako já na začátku studia. Navíc doufám, že tato práce neupadne v zapomnění a bude příkladem při rozhodování o tématu pro ostatní studenty. Diplomová práce totiž nemusí být ihned po obhajobě zahozena a nenávratně zapomenuta, ale může opravdu přinášet něco užitečného a inovativního.

Seznam použitých zdrojů

- (1) 1. díl: Úvod do jazyka Java. In: *Devbook.cz: Programátorská sociální síť* [online]. © 2014 [cit. 2014-02-04]. Dostupné z: <http://www.devbook.cz/java-tutorial-uvod-do-jazyka-java>
- (2) About the iOS Technologies. In: *iOS Developer Library* [online]. 2013, 18.9.2013 [cit. 2014-02-04]. Dostupné z: <https://developer.apple.com/library/ios/documentation/miscellaneous/conceptual/iphoneostechoverview/Introduction/Introduction.html>
- (3) Android Captures Record 81 Percent Share of Global Smartphone Shipments in Q3 2013. In: *Strategy Analytics: Wireless Smartphone Strategies* [online]. 2013 [cit. 2014-02-04]. Dostupné z: <http://blogs.strategyanalytics.com/WSS/post/2013/10/31/Android-Captures-Record-81-Percent-Share-of-Global-Smartphone-Shipments-in-Q3-2013.aspx>
- (4) API. In: *IT Biz: Vaše jednička mezi mlami* [online]. © 2014 [cit. 2014-02-04]. Dostupné z: <http://www.itbiz.cz/slovník/informacni-technologie-it/api>
- (5) Application Fundamentals. [2015]. *Android Developers* [online]. [cit. 2015-05-15]. Dostupné z: <http://developer.android.com/guide/components/fundamentals.html>
- (6) BOSOMWOTTH, Danyl. Mobile Marketing Statistics 2013: Statistics on mobile usage and adoption to inform your mobile marketing strategy. *Smart Insights* [online]. 2013 [cit. 2014-02-04]. Dostupné z: <http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>
- (7) Fragments. [2015]. *Android Developers* [online]. [cit. 2015-05-15]. Dostupné z: <http://developer.android.com/guide/components/fragments.html>
- (8) GARGENTA, Marko. Learning Android. 1st ed. Sebastopol, Calif.: O'Reilly, c2011, xvii, 245 p. ISBN 14-493-9050-1.

- (9) HEDLEY, Jonathan. 2015. *Jsoup: Java HTML Parser* [online]. [cit. 2015-05-15]. Dostupné z: <http://jsoup.org/>
- (10) HOUŠKA, Lukáš. Jak vypadá Android uvnitř aneb co je ROM, kernel, bootloader a další?. In: *Android Market* [online]. 2011 [cit. 2014-02-04]. Dostupné z: <http://www.androidmarket.cz/android/jak-vypada-android-uvnitř-aneb-co-je-rom-kernel-bootloader-a-dalsi/>
- (11) LEE, W.,M. Beginning Android application development. Indianapolis, IN: Wiley Pub., 2011. 428 s. Wrox beginning guides. ISBN 978-111-8087-800.
- (12) MARTIŠEK, D. Algoritmizace a programování v Delphi. 1. vyd. Brno: Littera, 2007. 230 s. ISBN 978-80-85763-37-9.
- (13) *Material Design Color Palette Generator: Material Palette* [online]. [2014] [cit. 2015-05-17]. Dostupné z: <http://www.materialpalette.com/>
- (14) MDA: The Architecture of Choice for a Changing World. *OMG: Object Management Group* [online]. 2013, 23.4.2013 [cit. 2014-02-04]. Dostupné z: <http://www.omg.org/mda/>
- (15) PODZIMEK, David. IDC: Android překonal 80% tržní podíl, jak se dařilo ostatním platformám?. In: *Smart Mania* [online]. 2013 [cit. 2014-02-04]. Dostupné z: <http://smartmania.cz/clanky/idc-android-prekonal-80-trzni-podil-jak-se-darilo-ostatnim-platformam-6179>
- (16) RICHTA, Karel. Metodiky vývoje software, MDA. In: *ČVUT* [online]. 2011 [cit. 2014-02-04]. Dostupné z: <https://edux.fit.cvut.cz/oppa/BI-SI1/prednasky/BI-SI1-P10m.pdf>
- (17) UJBÁNYAI, M. Programujeme pro Android. 1. vyd. Praha: Grada, 2012. 187 s. Průvodce (Grada). ISBN 978-80-247-3995-3.
- (18) UniApps v2 Beta. In: *Google play* [online]. © 2015 [cit. 2015-03-15]. Dostupné z: <https://play.google.com/store/apps/details?id=sk.prosoft.uniapps&hl=cs>

- (19) VELTE, A., T. VELTE a R. ELSENPETER. *Cloud Computing: praktický průvodce*. 1. vyd. Brno: Computer Press, 2011. 344 s. ISBN 978-80-251-3333-0.
- (20) VISWANATHAN, Priya. What is the BlackBerry OS?. In: *About.com* [online]. © 2014 [cit. 2014-02-04]. Dostupné z: <http://mobiledevices.about.com/od/glossary/g/What-Is-The-Blackberry-Os.htm>
- (21) *Vysoké učení technické v Brně* [online]. © 2004-2015 [cit. 2015-03-15]. Dostupné z: <https://www.vutbr.cz/intra>
- (22) Windows Phone OS. In: *GSM Arena* [online]. © 2000-2014 [cit. 2014-02-04]. Dostupné z: <http://www.gsmarena.com/glossary.php3?term=windows-phone-os>

Seznam obrázků

Obrázek 1: Přístupy na web dle typu zařízení, Zdroj: (6).....	13
Obrázek 2: Vrstvy systému iOS, Zdroj: (2).....	16
Obrázek 3: Architektura systému Android, Zdroj: (10)	18
Obrázek 4: Překlad zdrojového kódu v jazyce Java, Zdroj: (1).....	22
Obrázek 5: Modelem řízený vývoj, Zdroj: (14).....	25
Obrázek 6: Hlavní navigační panel, Zdroj: (21).....	30
Obrázek 7: Postranní navigační panel, Zdroj: (21).....	30
Obrázek 8: Zobrazení rozvrhu, Zdroj: (21)	31
Obrázek 9: Aktuality z předmětu, Zdroj: (21)	31
Obrázek 10: Registrace termínů, Zdroj: (21).....	32
Obrázek 11: Elektronický index, Zdroj: (21).....	32
Obrázek 12: Podíl jednotlivých typů OS mezi studenty FP, Zdroj: vlastní zpracování .	33
Obrázek 13: Hodnocení uživatelů na Google play, Zdroj: (18)	38
Obrázek 14: Aplikace UniApps v2 Beta, Zdroj: (18).....	38
Obrázek 15: Barevná paleta použitá v aplikaci, Zdroj: (13).....	42
Obrázek 16: Procesní diagram přihlášení uživatele, Zdroj: vlastní zpracování	45
Obrázek 17: Přihlašovací obrazovka, Zdroj: vlastní zpracování	46
Obrázek 18: Popis jednotlivých komponent a gest, Zdroj: vlastní zpracování	48
Obrázek 19: Navigační panel, Zdroj: vlastní zpracování	49
Obrázek 20: Aktualizace dat, Zdroj: vlastní zpracování.....	50
Obrázek 21: Procesní diagram aktualizací služby, Zdroj: vlastní zpracování	51
Obrázek 22: Ukázka notifikace, Zdroj: vlastní zpracování	53
Obrázek 23: Procesní diagram úvodní obrazovky, Zdroj: vlastní zpracování.....	54
Obrázek 24: Úvodní obrazovka, Zdroj: vlastní zpracování	56

Obrázek 25: Aktuality z předmětu, Zdroj: vlastní zpracování.....	57
Obrázek 26: Registrace termínů, Zdroj: vlastní zpracování	58
Obrázek 27: Elektronický index, Zdroj: vlastní zpracování	59
Obrázek 28: Rozvrh hodin, Zdroj: vlastní zpracování.....	60

Seznam tabulek

Tabulka 1: Přístupy na web dle konkrétních zařízení, Zdroj: (6)	14
Tabulka 2: Tržní podíl mobilních operačních systémů, Zdroj: (3).....	15
Tabulka 3: Výsledky provedeného průzkumu popularity jednotlivých OS mezi studenty FP, Zdroj: vlastní zpracování	34
Tabulka 4: Porovnání doby přístupu, Zdroj: vlastní zpracování	36
Tabulka 5: Porovnání doby přístupu, Zdroj: vlastní zpracování	61