



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	FITCom - Mobilní aplikace pro komunikaci na VUT FIT, OS Android
Student:	Jan Panchártek
Vedoucí:	Ing. Ji í Hunka
Studijní program:	Informatika
Studijní obor:	Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2015/16

Pokyny pro vypracování

Cílem této práce je vytvořit mobilní aplikaci v OS Android, která usnadní a urychlí komunikaci mezi studenty a vyu učícími na Fakult ě Informa ních technologií VUT. Hlavním p ínosem má být snadné a rychlé spojení dle aktuálního rozvrhu vyu učících a student ů.

- Prostudujte již existující komunika ní zam ěné aplikace na naší fakult ě - pr ůzkum platform ů neomezujte.
- Navrhn ěte vhodné ešení, které umožní student ům snadné kontaktování vyu učících aktuálních p edm ět a vyu učícím umožní jednoduché kontaktování celé paralelky i konkrétního studenta.
- P í návrhu se zam ěte také na uživatelské rozhraní aplikace.
- Jako zdroj dat využijte dostupné zdroje v rámci infrastruktury FIT.
- Implementaci i grafické rozhraní ůd ů otestujte.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
ř íd ěn

V Praze dne 1. února 2015

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



Bakalářská práce

FITCom - Mobilní aplikace pro komunikaci na ČVUT FIT, OS Android

Jan Panchártek

Vedoucí práce: Ing. Jiří Hunka

12. května 2016

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu Ing. Jiřímu Hunkovi za konzultace a rady při realizaci této práce. Dále bych rád poděkoval přítelkyni, rodině a přátelům za nekonečnou podporu.

Prohlášení

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

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

V Praze dne 12. května 2016

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2016 Jan Panchártek. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Panchártek, Jan. *FITCom - Mobilní aplikace pro komunikaci na ČVUT FIT, OS Android*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Tato bakalářská práce pojednává o návrhu a implementaci mobilní aplikace sloužící pro komunikaci mezi studenty a vyučujícími na Fakultě informačních technologií ČVUT. Na základě analýzy existujících aplikací je navržena nová mobilní aplikace pro operační systém Android. Výsledkem implementační části je hotová mobilní aplikace.

Klíčová slova ČVUT FIT, OS Android, mobilní aplikace, komunikace studentů a vyučujících

Abstract

This thesis discusses the design and implementation of a mobile application to be used for communication between students and teachers at the Faculty of Information Technology on CTU. Based on analysis of existing applications, a new mobile application will be proposed for the Android operating system. The result is a complete mobile applications.

Keywords CTU FIT, OS Android, mobile application, communication between students and teachers

Obsah

Úvod	1
1 Úvodní analýza	3
1.1 Mobilní aplikace pro ČVUT FIT	3
1.2 Mobilní e-mailoví klienti pro OS Android	6
1.3 E-mailové webové aplikace	10
1.4 Nyní používaná řešení	11
1.5 Platforma Android	14
1.6 Nástroje	16
2 Návrh řešení	19
2.1 Doménový model	19
2.2 Specifikace požadavků	20
2.3 Příklad užití	22
2.4 Databáze	24
2.5 Platforma Android	25
2.6 Návrh uživatelského rozhraní	26
3 Implementace	33
3.1 Přihlašování	33
3.2 Stahování dat	36
3.3 Parsování XML	37
3.4 Práce na pozadí	39
3.5 Ukládání dat	41
3.6 Uživatelské rozhraní a jeho prvky	43
3.7 Knihovny pro práci s e-mailem	43
3.8 Zprávy uživateli	44
3.9 Logování a debug	44
4 Testování	45

4.1	Testování vývojářem	45
4.2	Uživatelské testování	45
5	Možné rozšíření aplikace	49
5.1	Synchronizace s IMAP	49
5.2	Koncepty	49
5.3	Našptávání e-mailů	49
5.4	Rozšíření vztahů odesílatele a příjemce	50
5.5	Posílání a přijímání příloh	50
5.6	Kontakty	50
	Závěr	53
	Literatura	55
	A Seznam použitých zkratk	59
	B Obsah příloženého CD	61

Seznam obrázků

1.1	Rozložení používání operačních systémů[1]	4
1.2	KOSeek beta	5
1.3	Gmail	7
1.4	HTC	7
1.5	Mailbox	9
1.6	Rozložení používání e-mailových klientů[2]	11
1.7	Ukázka dotazníku	12
1.8	Graf rozložení řešení na ČVUT FIT	12
1.9	Reklama v aplikaci MailDroid	14
1.10	Používání verzí systému Android k 4.4.2016 [3]	15
2.1	Doménový model	20
2.2	Diagram aktivit	23
2.3	Doručené	27
2.4	Detail e-mailu	27
2.5	Nový e-mail	28
2.6	Profil	28
2.7	Studentovi hodiny	29
2.8	Hodiny vyučujícího	29
2.9	Detail hodiny	30
3.1	AuthorizationActivity	35
3.2	LoginActivity	35
3.3	Navigation Drawer	44

Seznam tabulek

1.1	Rozložení používání operačních systémů[1]	3
1.2	Funkce mobilních aplikací pro ČVUT	4
1.3	Funkce mobilních e-mailových klientů	6
1.4	Funkce webových aplikací s funkcí e-mailového klienta	10
1.5	Rozložení e-mailových účtů mezi servery [2]	10
1.6	Rozdělení existujících řešení	11
1.7	Používání verzí systému Android k 4.4.2016 [3]	15
4.1	Testeři	45

Úvod

Rychlost, čas, umění komunikace, znalost potřeb klientů - to jsou hesla, která se v současné době často používají. Jejich správné pochopení a využití je předpokladem pro úspěch v dnešním životě.

Čas je velmi cennou součástí života a většina z nás se snaží o jeho efektivní využití. Rychlý vývoj moderní techniky nám to částečně umožňuje. Většina lidí dnes má v kapse a na každém svém kroku "chytrý" mobilní telefon s přístupem na internet. Díky tomu zažívá obrovský "boom" vývoj aplikací pro tato zařízení.

Ve své bakalářské práci se nejprve budu zabývat průzkumem již existujících mobilních aplikací umožňujících komunikaci studentů a vyučujících na Fakultě informačních technologií. Na základě výsledků tohoto průzkumu a po zvážení nejvhodnějších možností řešení navrhnu elegantní a efektivní mobilní aplikaci, která co nejlépe splní požadavky jejich budoucích uživatelů. Dále ji implementuji a otestuji pro operační systém Android.

Mnou navržená mobilní aplikace bude umožňovat používání školního e-mailu. Její hlavní výhodou je rychlá komunikace bez zdlouhavého hledání kontaktů na jednotlivé vyučující, popřípadě na studenty, kteří jsou zapsáni do jednotlivých paralelek. Toho dosáhnu pomocí propojení s rozvrhem přihlášeného uživatele. Samozřejmostí je, že aplikace bude umět odesílat a přijímat e-maily. Další vítanou funkcí bude zobrazení důležitých údajů o odesílateli e-mailu přihlášenému uživateli. To jistě uvítají jak vyučující, tak studenti především prvních ročníků.

Úvodní analýza

1.1 Mobilní aplikace pro ČVUT FIT

Prvním směrem mého výzkumu jsou mobilní aplikace se zaměřením přímo na naši fakultu, tj. ČVUT FIT. Průzkum jsem dělal multiplatformní, proto jsem se zaměřil na tři operační systémy (Android, iOS, Windows phone), které zabírají 99,3%^[1] chytrých telefonů. Jejich přesné rozložení můžeme vidět v tabulce 1.1, nebo v přehledném koláčovém grafu na obrázku 1.1. V tabulce 1.2 vidíme funkčnost jednotlivých zkoumaných mobilních aplikací pro ČVUT.

1.1.1 KOSeek beta

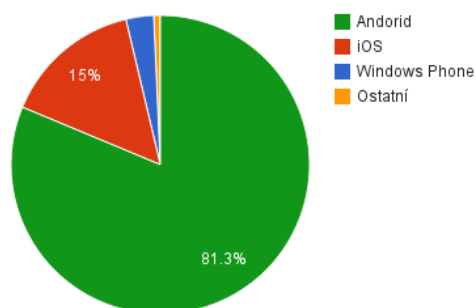
KOSeek je multiplatformní aplikace pro chytré telefony s operačním systémem Android, iOS a Windows phone. Tato aplikace je k dostání od druhé poloviny roku 2014 a vyrobil ji kolega Ing. Jan Šedivý CSc. Aplikace se může pochlubit pěkným grafickým zpracováním, a přestože obsahuje velkou řadou funkcí, je velmi přehledná.

Jak vidíme v tabulce 1.2, s porovnáním mobilních aplikací pro ČVUT FIT, funkčnost této je obrovská. Přihlášenému uživateli umožňuje prohlížet svůj profil, rozvrh i průchod studiem. Dále lze vyhledávat jakoukoli osobu na naší fakultě a nezáleží na tom, jestli vyhledávaná osoba je studentem, vyučujícím nebo dalším zaměstnancem školy. Také máme možnost kontaktovat

Tabulka 1.1: Rozložení používání operačních systémů^[1]

Operační systém	%
Android	82.30
iOS	15.00
Windows Phone	3.00
ostatní	0.70

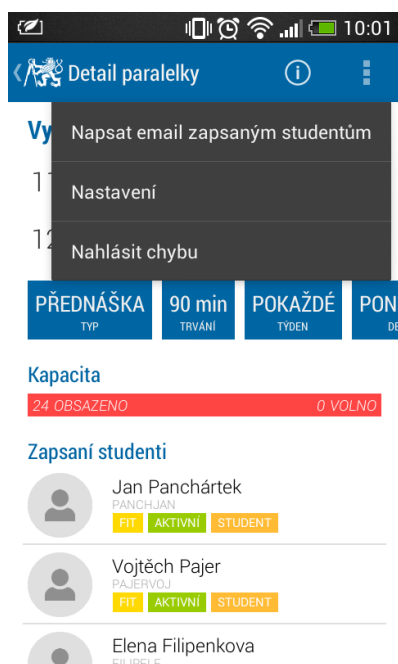
1. ÚVODNÍ ANALÝZA



Obrázek 1.1: Rozložení používání operačních systémů[1]

Tabulka 1.2: Funkce mobilních aplikací pro ČVUT

Funkce	KOSeek	FIT Checker	Menzy
Přihlašování	Ano	Ano	Ne
Profil studenta	Ano	Ne	Ne
Informace o studiu	Ano	Ne	Ne
Hledání kontaktů	Ano	Ne	Ne
Aktuality	Ano	Ne	Ne
Kontakt na vedení školy	Ano	Ne	Ne
Psaní e-mailů	Ne	Ne	Ne
Úřední hodiny SO	Ano	Ne	Ne
Rozvrh uživatele	Ano	Ne	Ne
Rozvrh ostatních	Ano	Ne	Ne
Vyučující + kontakty	Ano	Ne	Ne
Psaní e-mailu	Ano*	Ne	Ne
Psaní e-mailu paralelce	Ano*	Ne	Ne
Import předmětů z KOSu	Ne	Ano	Ne
Edux studijní výsledky	Ne	Ano	Ne
Termíny zkoušek	Ne	Ano	Ne
Otevírací doba menz	Ne	Ne	Ano
Jídlo v menzách	Ano*	Ne	Ano
Stav konta ČVUT	Ne	Ne	Ano*
Pro Android	Ano	Ano	Ano
Pro Windows phone	Ano	Ano	Ano
Pro iOS	Ne	Ano	Ano
* pomocí jiné aplikace v zařízení			



Obrázek 1.2: KOSeek beta

osobu nebo všechny osoby v paralelce našeho rozvrhu, jak vidíme na obrázku 1.2. Tato funkce je zde realizována pomocí otevření externí aplikace, která komunikaci umožňuje.

Závěrem zkoumání KOSeek beta můžeme konstatovat, že aplikace je dobře navržená a implementovaná, ale neumožňuje nám přímou komunikaci mezi vyučujícími a studenty.

1.1.2 ČVUT FIT Checker

Aplikace FIT Checker je jednoduchou mobilní aplikací, která za nás kontroluje školní webový portál Edux, do kterého vyučující zadávají body jednotlivým studentům. V případě změny nás informuje pomocí upozornění v našem mobilním telefonu. Druhou funkcí této aplikace je upozornění uživatele na nově vypsany zkouškový termín v KOSu u kteréhokoliv z jeho zapsaných předmětů. Předměty si uživatel buď importuje z KOSu podle rozvrhu všechny najednou, nebo si může přidávat předměty postupně pomocí zadání jejich kódu.

Z testu vyplývá, že FIT Checker je plně funkční mobilní aplikace, ale nezajišťuje komunikaci studentů a vyučujících na naší fakultě.

1.1.3 Menzy ČVUT

Pod České Vysoké Učení Technické náleží několik menz. Pro rychlou a s snadnou informovanost studentů i ostatních lidí patřících pod ČVUT existuje tato

Tabulka 1.3: Funkce mobilních e-mailových klientů

Funkce	Gmail	Email.cz	HTC	Droid	Mailbox
Přihlašování	Ano	Ano	Ano	Ano	Ano
Poslání e-mailu	Ano	Ano	Ano	Ano	Ano
Našeptávání	Ano	Ano	Ne	Ne	Ano
Import kontaktů	Ne	Ne	Ne	Ne	Ne
Připojení souborů	Ano	Ano	Ano	Ano	Ano
Čtení e-mailů	Ano	Ano	Ano	Ano	Ano
Vyhledávání e-mailů	Ano	Ano	Ano	Ano	Ano
Štítkování	Ano	Ne	Ne	Ano	Ne
Oznámení příchozího e-mailu	Ano	Ano	Ano	Ano	Ano
Grupování konverzací	Ano	Ne	Ano	Ne	Ano
Kontakty	Ne	Ne	Ne	Ne	Ne
Složkování	Ne	Ano	Ne	Ne	Ne
Reklama	Ne	Ne	Ne	Ano	Ne
Filtrování	Ne	Ne	Ne	Ano	Ne
Propojení více účtů	Ne	Ne	Ano	Ano	Ano*
Odložení upozornění	Ne	Ne	Ne	Ne	Ano
Jazyk Čeština	Ano	Ano	Ano	Ne	Ne
Pro Android	Ano	Ano	Ano	Ano	Ano
Pro Windows phone	Ne	Ano	Ano	Ne	Ne
Pro iOS	Ano	Ne	Ne	Ne	Ano

jednoduchá aplikace Menzy ČVUT. Tato aplikace zobrazuje aktuální nabídku jídel v jednotlivých menzách ČVUT včetně jejich cen a otevíracích dob jídelen.

Tato mobilní aplikace, jako již dříve zmíněné, nepodporuje komunikaci lidí na ČVUT.

1.2 Mobilní e-mailoví klienti pro OS Android

Druhým směrem mého výzkumu jsou mobilní aplikace reprezentující e-mailového klienta v "chytrých" mobilních telefonech. Pomocí těchto aplikací můžeme snadno a rychle komunikovat prostřednictvím e-mailů. Zkoumal jsem funkčnost známých a denně používaných aplikací, ale i méně známé e-mailové klienty. Veškerou funkčnost si můžeme prohlédnout v tabulce 1.3. Funkce mobilních e-mailových klientů.

1.2.1 GMail

Mobilní e-mailový klient od společnosti Google, který je automaticky nainstalován na všech mobilních zařízeních s operačním systémem Android. Jelikož

1.2. Mobilní e-mailoví klienti pro OS Android

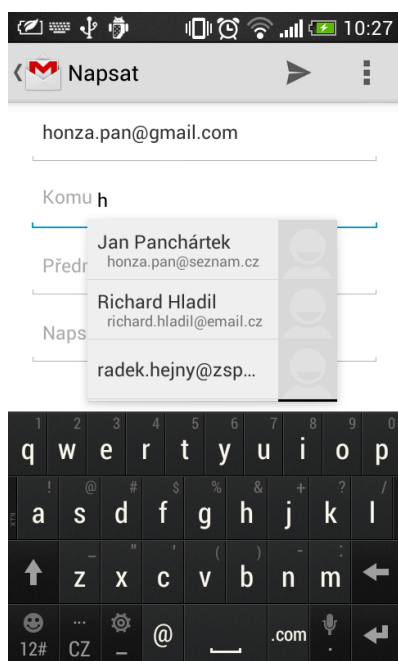
mobilní aplikace je plně podporována vývojáři z Googlu, umí veškeré vymoženosti včetně štitkování, barvení, třídění do konverzací nebo filtrování.

Velkou výhodou této oficiální aplikace je perfektně provedená synchronizace, díky které můžeme utvořit koncept e-mailu v telefonu a všechno v pohodlí dopsat a odeslat z webové aplikace. Což u ostatních mobilních e-mailových aplikací většinou nelze.

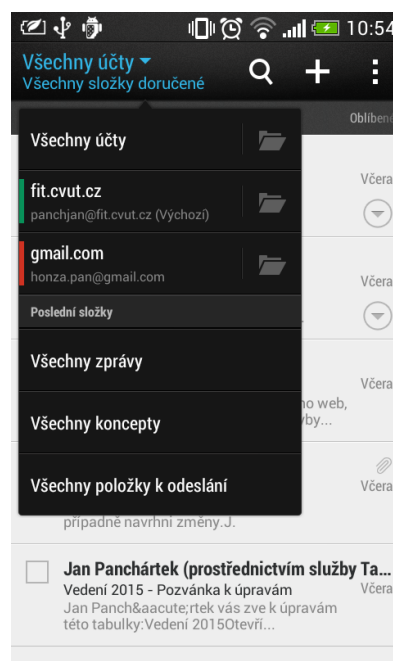
Jedna z mála nevýhod této aplikace je omezení pouze na e-mail od Googlu. Nelze do ní, na rozdíl od jiných e-mailových, klientů připojit více e-mailů. Druhá věc, která může někomu vadit je, že vzhled aplikace nelze skoro vůbec přizpůsobit podle svých představ.[4]

Závěrem lze o této aplikaci říci, že je perfektní, ale pro komunikaci vyučujících a studentů na naší fakultě nevhodná, a to zejména ze dvou důvodů. Prvním důvodem je nemožnost importování kontaktů, druhou již zmíněná nemožnost připojení jiného e-mailu než je Gmail.

Vzhledem k její dobré funkčnosti a preciznímu zpracování mi ale může velmi dobře posloužit jako inspirace při návrhu mé mobilní aplikace. Například rozložení tlačítek je povedené, funkce našeptávání kontaktu při psaní e-mailu je také vynikající. Pro větší názornost ukázaná na obrázku 1.3.



Obrázek 1.3: Gmail



Obrázek 1.4: HTC

1.2.2 Email.cz

Aplikace Email.cz byla vytvořena programátory českého portálu Seznam.cz, který je u nás velkým konkurentem celosvětového fenoménu Google. V dnešní době tuto mobilní aplikaci podporují dva mobilní operační systémy, a to Android a Windows phone.

Samotná aplikace nabízí přehledné zobrazení příchozích zpráv a všechny funkce snadno dosažitelné na obrazovce. Buď jako tlačítka, popřípadě pomocí gest. Například rychlé smazání zprávy je zde umožněno pomocí přejetí prstem po displeji zleva doprava. Aktualizování stavu schránky pomocí gesta tažením prstu dolů. Další funkcí této mobilní aplikace je podpora vyhledávání e-mailů. Můžeme vyhledávat podle textu, předmětu, ale i dle odesílatele nebo příjemce.[5]

Nakonec musím konstatovat, že mobilní aplikace od společnosti Seznam.cz nabízí pro běžného uživatele používajícího jejich e-mail vše, co může potřebovat, ale stejně jako v aplikaci gmail od Googlu nelze importovat kontakty a připojit jiný než jejich domovský e-mail. Proto je tato aplikace také nevhodná pro komunikaci studentů a vyučujících na naší fakultě.

1.2.3 HTC výchozí e-mailový klient

Společnost HTC v každém telefonu nabízí standardní aplikaci pro správu e-mailových účtů. Tato mobilní aplikace disponuje funkcemi jako většina e-mailových klientů, tj: psaní e-mailů, vyhledávání ve zprávách či propojení více e-mailových účtů do jedné aplikace. Toto můžeme vidět na obrázku aplikace 1.4.

Dále se může pochlubit pěkným grafickým uživatelským rozhraním s dobrou dostupností všech zmíněných funkcí. Bohužel i tato aplikace má své nedostatky, jako například nemožnost importování kontaktů nebo použití pouze v telefonech společnosti HTC.

Mé hodnocení: tato mobilní aplikace je velmi pěkně navržená a dobře implementovaná, ale bohužel se nedá použít pro náš problém, tj. komunikace mezi studenty a vyučujícími na naší fakultě, vzhledem k nemožnosti importování kontaktů.

1.2.4 Mail Droid

Mail Droid je univerzální mobilní aplikace zajišťující e-mailové služby. Můžeme ji najít pouze v obchodě od Googlu. Což znamená, že si ji nainstalujeme jenom na mobilní telefon s operačním systémem Android. Aplikace nám nabízí plnou funkčnost reprezentovanou čtením a odesíláním e-mailů, dále samozřejmě vyhledávání e-mailů či štítkování.

Jelikož tato aplikace není zastoupena žádnou velkou společností, jako Google, HTC nebo další, můžeme zde vidět částečné nedostatky. Například špatný

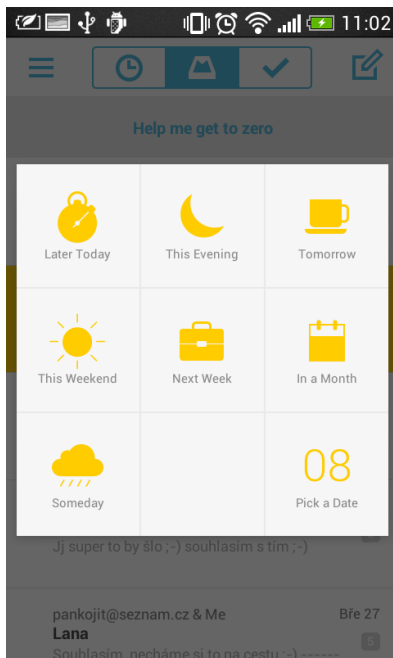
překlad do češtiny, kdy některé části aplikace jsou v češtině, některé v angličtině a některé kombinací obou. Dále mi dost chybí našeptávání při psaní nové zprávy.

1.2.5 Mailbox

Mailbox je novější mobilní aplikace od vývojářů ze společnosti Dropbox. Tento mobilní e-mailový klient má navíc další funkce, které ostatní konkurenti nemají. Zároveň neumí některé funkce, které ostatní umí. Například jeden z nedostatků je nemožnost propojení jiného účtu než Google nebo iCloud. Druhou nevýhodou je nutnost mít v mobilu nainstalovanou aplikaci Mailbox, ale i Dropbox, do kterého se přichází a odchází e-maily ukládají. Aplikace mají 53MB potažmo 34MB, což je docela velké místo, které zabírají v uživatelově zařízení.

Nové funkce, se kterými vývojáři přišli, spočívají hlavně v jednoduchém třídění zpráv a to pomocí pouhého tažení doleva či doprava. Přetažením zprávy doprava zařadíme e-mail do seznamu důležitých zpráv. Když zprávu přetáhneme naopak doleva, docílíme tím, že na tento přichází e-mail budeme znovu upozorněni a z přehledné tabulky si vybereme, kdy má toto upozornění přijít. Jestli večer, zítra nebo až za týden. Tabulku si můžeme prohlédnout na obrázku 1.5.[4]

Tato mobilní aplikace se může pyšnit pěkným grafickým rozhraním a myslím, že nově implementované funkce jsou dobře promyšlené a mají budoucnost.



Obrázek 1.5: Mailbox

1.3 E-mailové webové aplikace

E-mailové webové aplikace jsou primárně používány každým z nás pro správu e-mailového účtu nebo účtů. Funkčnost jednotlivých zkoumaných webových aplikací si můžeme prohlédnout v tabulce 1.4. Tyto e-mailové aplikace mají výhodu v tom, že nepotřebují žádnou instalaci a jsou zdarma. Stačí nám jakýkoliv počítač s přístupem na internet a webovým prohlížečem. Celosvětový fenomén Google a jeho @gmail.com je v naší republice drcen konkurencí v podobě tuzemského vyhledávače Seznam, který nabízí hned tři domény. A to: @seznam.cz, @email.cz, @post.cz a zabírá 70% používaných e-mailů. Přesná čísla můžeme vidět v tabulce 1.5, popřípadě na grafu 1.6. Druhým naším vyhledávačem poskytujícím e-mailové služby, zabírajícím 10% e-mailových schránek, je Centrum. Google je až třetím poskytovatelem s 5% [2].

Tabulka 1.4: Funkce webových aplikací s funkcí e-mailového klienta

Funkce	Google	Seznam	imap.fit.cvut.cz	Centrum
Přijímání e-mailů	Ano	Ano	Ano	Ano
Odesílání e-mailů	Ano	Ano	Ano	Ano
Kontakty	Ano	Ano	Ano	Ano
Import kontaktů	Ano	Ne	Ano	Ano
Složky	Ne	Ano	Ne	Ano
Štítkování	Ano	Ano	Ano	Ano
Našeptávání	Ano	Ano	Ano	Ano
Posílání souborů	Ano	Ano	Ano	Ano
Automatická oprava chyb	Ano	Ano	Ano	Ne
Export kontaktů	Ano	Ne	Ano	Ano
Více rozepsaných e-mailů	Ano	Ne	Ne	Ne

Tabulka 1.5: Rozložení e-mailových účtů mezi servery [2]

E-mail	%
Seznam	63.60
Centrum	9.10
Email	5.60
Gmail	5.20
Volný	2.60
Atlas	1.40
Tiscali	1.30
Post	0.80
Quick	0.40
Ostatní	9.90

Tyto velké rozsáhlé aplikace s plnou funkcí můžeme spustit a nahlížet na e-mailový účet i pomocí svého "chytrého" telefonu, ale má to svoje nevýhody jako například pomalé načítání. V některých případech i špatné zobrazení, protože na ně nahlížíme pomocí nainstalovaného prohlížeče, který máme ve svém telefonu.



Obrázek 1.6: Rozložení používání e-mailových klientů[2]

1.4 Nyní používaná řešení

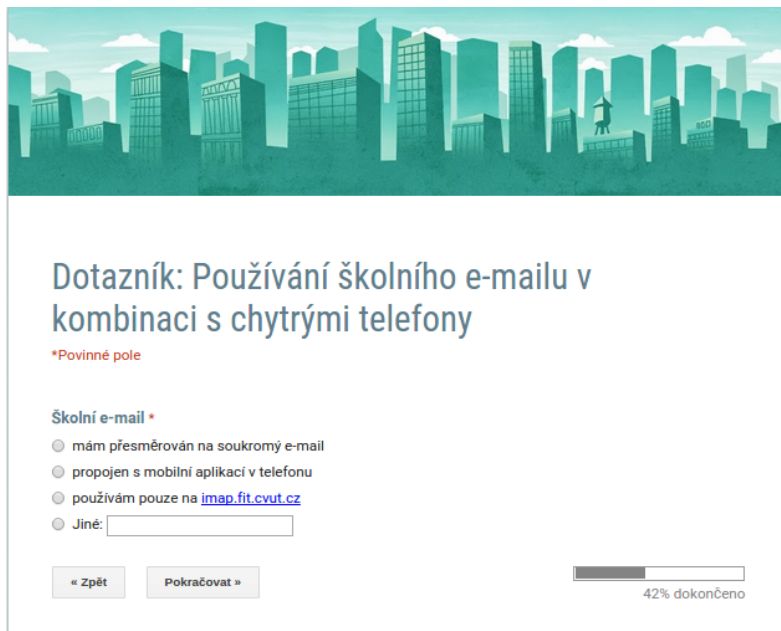
Abych zjistil, jak tento problém řeší v současnosti studenti i vyučující, vytvořil jsem krátký dotazník pomocí google dokumentů. Na obrázku 1.7 můžeme vidět ukázkou jedné otázky. Požádal jsem o vyplnění studenty i vyučující naší fakulty. Dotazník vyplnilo 250 respondentů a to v přesném poměru 22:3 (studenti:vyučující).

Výsledky jsem vyhodnotil podle několika kritérií. První vyhodnocená data můžeme vidět v tabulce 1.6 nebo v grafu 1.8. Tyto výsledky reprezentují, jak tento problém řeší v dnešní době lidé na naší fakultě. Možnosti řešení jsou pouze tři a jsou rovnoměrně rozloženy v našich řadách.

Tabulka 1.6: Rozdělení existujících řešení

Řešení	Počet	%
Přesměrování	86	35.54
Propojení s jinou mobilní aplikací	77	31.82
Použití pouze imap.fit.cvut.cz	76	31.40
Ostatní	3	1.24

1. ÚVODNÍ ANALÝZA



Dotazník: Používání školního e-mailu v kombinaci s chytrými telefony

*Povinné pole

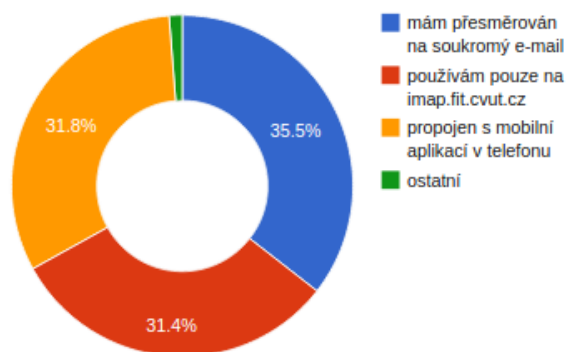
Školní e-mail *

- mám přesměrován na soukromý e-mail
- propojen s mobilní aplikací v telefonu
- používám pouze na imap.fit.cvut.cz
- Jiné:

« Zpět Pokračovat »

42% dokončeno

Obrázek 1.7: Ukázka dotazníku



Obrázek 1.8: Graf rozložení řešení na ČVUT FIT

1.4.1 Přesměrování

Problém neexistující mobilní aplikace pro ČVUT FIT řeší 36% studentů a zaměstnanců naší fakulty pomocí přesměrovávání. V tomto případě si nechávají všechny příchozí e-maily na školní e-mail přeposílat na svoje osobní e-mailové účty (gmail.com, seznam.cz nebo další poštovní adresy). Pomocí přesměrování docílují toho, že při přijmutí nového e-mailu jim "chytrý" telefon zobrazí upozornění na novou zprávu. Mají tak přehled o novinkách a dalších důležitých věcech, jako například otevření zkuškového termínu pro přihlášení.

Velkou nevýhodou tohoto řešení je nemožnost přímo v aplikaci využívat tlačítko odpovědět a pohodlně odepsat. Nejdříve se musí přihlásit, nejlépe z počítače, do webové aplikace `imap.fit.cvut.cz`, a zde na tento e-mail odpovědět. Druhým problémem, který zde vidím, je nepropojenost s uživatelským rozvrhem. Musí tedy hledat každý kontakt zvlášť, což mu zabere čas, který by mohl věnovat jiné práci.

1.4.2 Propojení s jinou mobilní aplikací

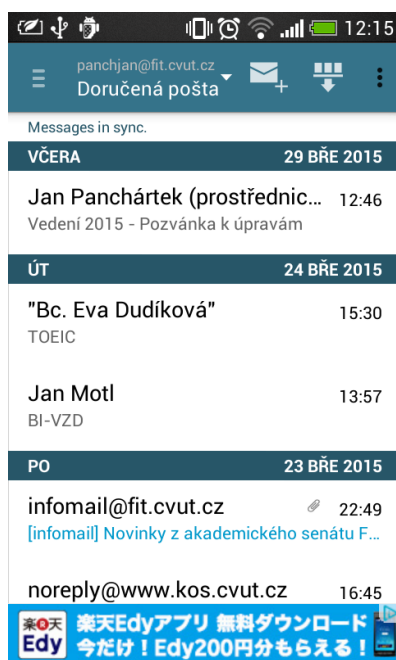
Druhá třetina kolektivu ČVUT FIT si školní e-mail propojí s jinou mobilní aplikací pro správu e-mailu. Pomocí této aplikace jsou e-maily stahované ze serveru a uživatel dostává upozornění na novou zprávu ihned. Může na ni pomocí této aplikace odpovědět.

Jelikož je většina těchto aplikací volně stažitelná, obsahuje reklamy, které zabírají část displeje 1.9. Po náhodném kliknutí na tuto reklamní část se nám otevře bez upozornění výchozí prohlížeč telefonu, což může být vysoce "otravné" a znepríjemňující práci. Dále tyto aplikace neumí propojení se školním rozvrhem a celkově s kontakty z naší fakulty. Proto občas vůbec nevíme, kdo nám vlastně píše a zjistíme to až při čtení samotného obsahu e-mailu.

1.4.3 Používá pouze `imap.fit.cvut.cz`

Třetí početná část studentů a zaměstnanců tento problém neřeší, protože jim nevyhovuje ani jedna z předchozích možností. Používají pouze webovou aplikaci IMAP. Toto řešení jeho uživatelům dopřává nevědomost, protože o nově příchozích e-mailech nemají žádné upozornění a dozvídají se o nich, až když si je sami vyhledají.

Obrovskou nevýhodou pro studenty, kteří nesedí a nekontrolují IMAP neustále je, že se o otevření zkuškových termínů dozvídají později, kdy už bývají nejlukrativnější termíny zabrané. Další nevýhodou studentů je, že pokud některý z vyučujících využívá pouze IMAP, jeho odpovědi na případné dotazy nejsou většinou tak rychlé jako u vyučujících, kteří používají propojení do mobilní e-mailové aplikace nebo mají školní e-mail přesměrován.



Obrázek 1.9: Reklama v aplikaci MailDroid

1.5 Platforma Android

1.5.1 Obecné informace

Android je operační systém, který je vyvíjen konsorciem Open Handset Alliance. Toto konsorcium je seskupení předních firem v oblasti IT a mobilních zařízení, jako jsou například Google, HTC, Samsung a mnoho dalších. Androidní platforma využívá jádro postavené na Linuxu a stejně jako tento operační systém je open source.

Dnes je nejvíce využíván v mobilních chytrých telefonech, ale i dalších zařízeních, kterých je používáno stovky milionů ve více než 190 zemích po celém světě. Operační systém Android je nejrozšířenější mobilní platforma a dále rychle roste. Každý den spousta uživatelů spouští poprvé své zařízení, využívající tento systém, a začíná hledat aplikace, hry a další digitální obsah. Android je také prvotřídní platformou pro vytváření aplikací a her, což ocení uživatelé po celém světě, protože vzniká velké množství různých aplikací.

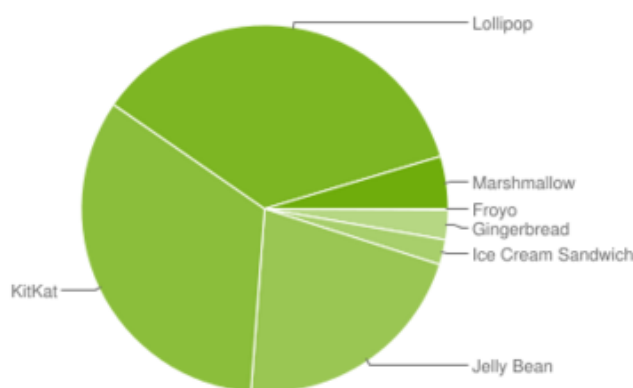
Dříve chytré telefony disponovaly obrazovkou o velikosti okolo tří palců, dnes na trhu nalezneme zařízení s dvojnásobnou hodnotou. V roce 2010 se na trh dostaly první tablety s Androidem, jako reakce na uvedení tabletu iPad od společnost Apple. Tyto tablety disponují velikostí obrazovky od sedmi do deseti palců. Proto vývojáři museli přijít s novou verzí Androidu, která umožňuje práci s těmito velkými obrazovkami.

1.5.2 Historie verzí

Všechny dosud vydané verze systému jsou pojmenovány podle sladkostí v angličtině (Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat, Lollipop a Marshmallow). Verze 2.2 Froyo - 2.3.7 Gingerbread byly uzpůsobeny pouze na malé displeje tj. maximálně okolo tří palců. Verze 3.0 Honeycomb je platforma přizpůsobena na větší úhlopříčky, takže se s ní můžeme setkat například v tabletech. Až od verze Androidu 4.0 Ice Cream Sandwich byla platforma upravena tak, že zvládne jakoukoli velikost obrazovky. Poté následovaly další verze 4.X, které jsou velice úspěšné a v dnešní době je najdeme na velké většině "chytrých" mobilních telefonů s operačním systémem Android. Přesné rozložení si můžeme prohlédnout v tabulce 1.7 nebo na grafu 1.10. Podíl je vyjádřen v procentech.

Tabulka 1.7: Používání verzí systému Android k 4.4.2016 [3]

Verze	Kódové jméno	API	Podíl
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.2%
4.1.X	Jelly Bean	16	7.8%
4.2.X	Jelly Bean	17	10.5%
4.3	Jelly Bean	18	3.0%
4.4	KitKat	19	33.4%
5.0	Lollipop	21	16.4%
5.1	Lollipop	22	19.4%
6.0	Marshmallow	23	4.6%



Obrázek 1.10: Používání verzí systému Android k 4.4.2016 [3]

Nižší verze než 2.2 Froyo zde nejsou uvedeny pro svůj zanedbatelný podíl na trhu. V době psaní této práce byla nejaktuálnější verze systému Android 6.0 s kódovým označením Marshmallow, která byla vydána 5.října 2015. Novinkami v této verzi jsou rozpoznávání otisku prstů a připojení přes USB Typu C.

1.6 Nástroje

1.6.1 Android SDK

Android SDK rozděluje nástroje, platformy a další součásti do balíčků. Tyto si snadno můžeme stáhnout pomocí SDK Manageru. Ten obsahuje například balíčky SDK Tools, které nám napomáhají v debugování a testování vyvíjené aplikace. Balík SDK platform je k dostání pro každou z verzí systému Android a nalezneme v něm zkompilevané knihovny. Dále balíček Documentation je dobrý pomocník, protože obsahuje aktuální dokumentaci k dané verzi [6].

1.6.2 Implementační prostředí IDE

Programy pro operační systém Android se dají psát, tak jako všechny jiné programy, v jakémkoli editoru. A to v úplně obyčejném, nebo ve více inteligentním programu navrženém přímo pro vývoj aplikací určených pro OS Android.

- **Textové editory** - Všechny programy, včetně těch pro OS Android, můžeme psát v textových editorech bez žádné podpory v podobě našeptávání nebo zbarvování klíčových slov. Můžeme tedy používat Gedit, Vim nebo Nano na Linuxu nebo na operačním systému Windows klasický poznámkový blok (Notepad).
- **Eclipse** - Je jedno z nejoblíbenějších vývojových prostředí v dnešní době. Můžeme v něm vyvíjet programy v jakémkoli jazyce, proto ani vývoj aplikací pro OS Android není výjimkou. Musíme pro to udělat pouze jedinou věc, a to doinstalovat si plugin, který tento vývoj podporuje. Další možností je stáhnout si kompletní balíček vývojového prostředí již s Android SDK.
- **IntelliJ IDEA** - Vývojové prostředí od společnosti JetBrains je zárukou kvality, stejně jako ostatní vývojová prostředí od této firmy. Například pro PHP PhpStorm nebo pro skriptovací jazyky Python PyCharm. Pro vývoj aplikací na Android můžeme používat buď plnou placenou verzi, a nebo neúplnou verzi, která je ke stažení zdarma.
- **Android Studio** - Android Studio je vyvíjeno společností Google a je nabízeno zdarma, přestože je postaveno na výše zmíněném IntelliJ IDEA [7] [8].

1.6.3 Umlet

Pro tvorbu návrhových diagramů je používán jednoduchý grafický editor Umlet. Tento program je založen na předpřipravených paletách pro různé diagramy. V jednotlivých paletách nalezneme všechny možné připravené tvary od šipek, bublinek, čtverečků, koleček až po nejrůznější tvary, které jednotlivé diagramy vyžadují. Licence programu Umlet je volně šiřitelná. Sice nenabízí plnou podporu jako jiné návrhové programy, které částečně za vás myslí a nedovolí vám namodelovat nesmyslné. To Umlet neumí, dovolí prakticky cokoli a myslet musíte vy.

1.6.4 Gimp

Gimp je grafický program pro tvorbu grafiky, který používám jak pro vytvoření loga aplikace, tak pro vytvoření obrázků, které následně využívá aplikace. Jedná se o obrázky puntíků k jednotlivým hodinám podle jejich typu.

1.6.5 Git a Gitlab

Git je nástroj pro verzování. Pomocí tohoto nástroje ukládám a zálohuji hlavně kód aplikace, ale i další součásti bakalářské práce. Veškeré zálohování a ukládání je prováděno na fakultní datový server s názvem Gitlab.

Gitlab je poskytován každému studentovi nebo vyučujícímu na naší fakultě informačních technologií. Dá se použít jako datový server pro veškeré projekty a semestrální práce na naší fakultě, jelikož povoluje u projektů nastavení ochrany soukromí. Projekty mohou být soukromé (privátní), společné pro vybrané uživatele používající Gitlab, anebo plně veřejné.

Návrh řešení

Kapitola Návrh řešení popisuje, jak bude má aplikace vnitřně navržena v podobě doménového modelu, s vymezením rozsahu pomocí funkčních a nefunkčních požadavků a také jakou bude mít funkcionalitu.

2.1 Doménový model

Na obrázku 2.1 je vykreslen model aplikace, jehož základem jsou tři hlavní třídy. Konkrétně **E-mail**, **Kontakt** a **Hodina**. Potom jsou zde zakresleny i podpůrné třídy a to **Čas a místo**, **Student** a **Vyučující**.

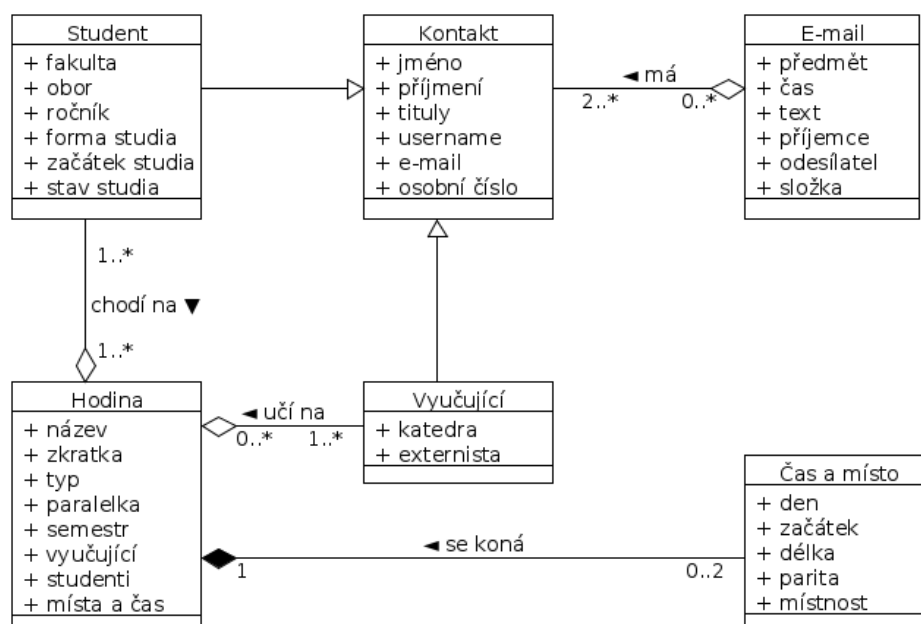
V diagramu je zakomponováno několik věcí. Jsou tam popisky atributů jednotlivých tříd, různé typy vztahů mezi třídami a také je zde zakreslena kardinalita vztahů jednotlivých tříd.

Na diagramu jsou použity tři typy vztahů. Prvním typem je **Generalizace**, která je zakreslována pomocí plné čáry zakončené prázdnou uzavřenou šipkou. Vztah je následně implementován jako dědění. Šipka je zakreslena u třídy, z které je děděno. Na mém diagramu třídy **Student** a **Vyučující** dědí atributy a metody z třídy **Kontakt**.

Dalším vztahem je **Agregace**. Tento vztah reprezentuje typ celek a část. V kódu ho implementujeme jako kolekci (pole) tříd část, kterou si držíme jako atribut třídy celek. Vztah označujeme plnou čarou, zakončenou prázdným kosočtvercem u třídy celek. Na mém diagramu je toto nejpoužívanější vztah. Například třída **Hodiny** využívá tento vztah hned třikrát a vždy je v roli celku.

Posledním použitým vztahem je **Kompozice**. Tento vztah je silnější agregace. Jinak řečeno, část nemá bez celku smysl. Implementuje se stejně jako předchozí vztah a to kolekci v celku. Označujeme ji plnou čarou ukončenou plným kosočtvercem opět u třídy celku.[9]

2. NÁVRH ŘEŠENÍ



Obrázek 2.1: Doménový model

2.2 Specifikace požadavků

V této podkapitole jsou rozepsány jednotlivé požadavky v textové podobě s případným detailním popisem.

2.2.1 Nefunkční požadavky

- **Podpora verze Androidu 4.0 a výš** - aplikace bude podporována od verze Androidu 4.0 a výše.
- **Bezpečnost** - aplikace bude zabezpečena ve smyslu ochrana přihlašovacími údaji, jelikož přihlašování bude probíhat prostřednictvím školního username a hesla.
- **Komunikace s IMAP** - komunikace a posílání e-mailů bude fungovat přes školní server IMAP.
- **Offline režim** - aplikace bude funkční i v době, kdy nebude k dispozici internetové připojení. V tuto dobu bude mít uživatel možnost pročítat dříve příchozí e-maily.
- **Uzpůsobení návrhu pro migraci na jiné platformy** - návrh nebude platformě závislý a bude přenositelný na ostatní platformy (iOS, Windows phone, popřípadě další).

- **Získání dat z Kosapi** - aplikace bude získávat data o přihlášeném uživateli z datového serveru Kosapi.
- **Možnost rozšíření a modifikace** - aplikace bude mít jednoduchý objektový návrh, díky kterému bude v budoucnu snadno rozšiřitelná a případně modifikovatelná.
- **Intuitivní uživatelské rozhraní** - uživatelské rozhraní bude jednoduché, příjemné a intuitivní.

2.2.2 Funkční požadavky

- **Autorizace uživatele** - uživatel se bude muset přihlásit do aplikace pomocí školních přihlašovacích údajů.
- **Přidání účtu do zařízení** - jestliže uživatel chce používat komunikaci pomocí e-mailů, musí si přidat účet do zařízení, přičemž dojde k bezpečnému uložení uživatelského jména a hesla.
- **Informace o hodinách studentů** - student bude mít možnost zobrazit si své veškeré hodiny za celé studium. Následně student bude mít k dispozici detail jednotlivých hodin, kde se dozví veškeré informace o hodině.
- **Informace o hodinách vyučujících** - vyučující si může zobrazit hodiny, které vyučuje nebo v minulosti vyučoval na FIT a FEL. Poté bude mít k dispozici detail jednotlivých hodin, stejně jako student.
- **Odesílání e-mailů** - aplikace bude umět odesílat e-maily jednotlivcům nebo celé paralelce z detailu hodiny, bez jakékoliv znalosti e-mailu(ů).
- **Vztah uživatele s odesílatelem e-mailu** - v detailu jednotlivého e-mailu přihlášený uživatel uvidí, jaký má vztah s odesílatelem e-mailu, který mu přišel. Studentovi se zobrazí, jestli daný odesílatel je jeho vyučující, v současné době i v minulosti. Vyučujícímu se zobrazí, jestli daný odesílatel je student, kterého někdy vyučoval nebo cvičil.
- **Manuální aktualizace doručených e-mailů** - v seznamu příchozích e-mailů bude mít uživatel možnost manuální aktualizaci schránky.
- **Zobrazovat profil uživatele** - aplikace bude umět přehledně zobrazovat detail přihlášeného uživatele, v kterém budou vykresleny veškeré podrobné informace o něm.
- **Odhlášení z aplikace** - systém bude poskytovat uživateli možnost odhlášení, při kterém dojde k smazání veškerých stažených dat, aby aplikace byla připravena pro nové přihlášení jiného uživatele.

2.3 Příklad užití

V této podkapitole jsou popsány jednotlivé role v aplikaci a případy užití aplikace studenty i vyučujícími. Jednotlivé případy jsou popsány v scénářích.

2.3.1 Seznam účastníků

2.3.1.1 Vyučující

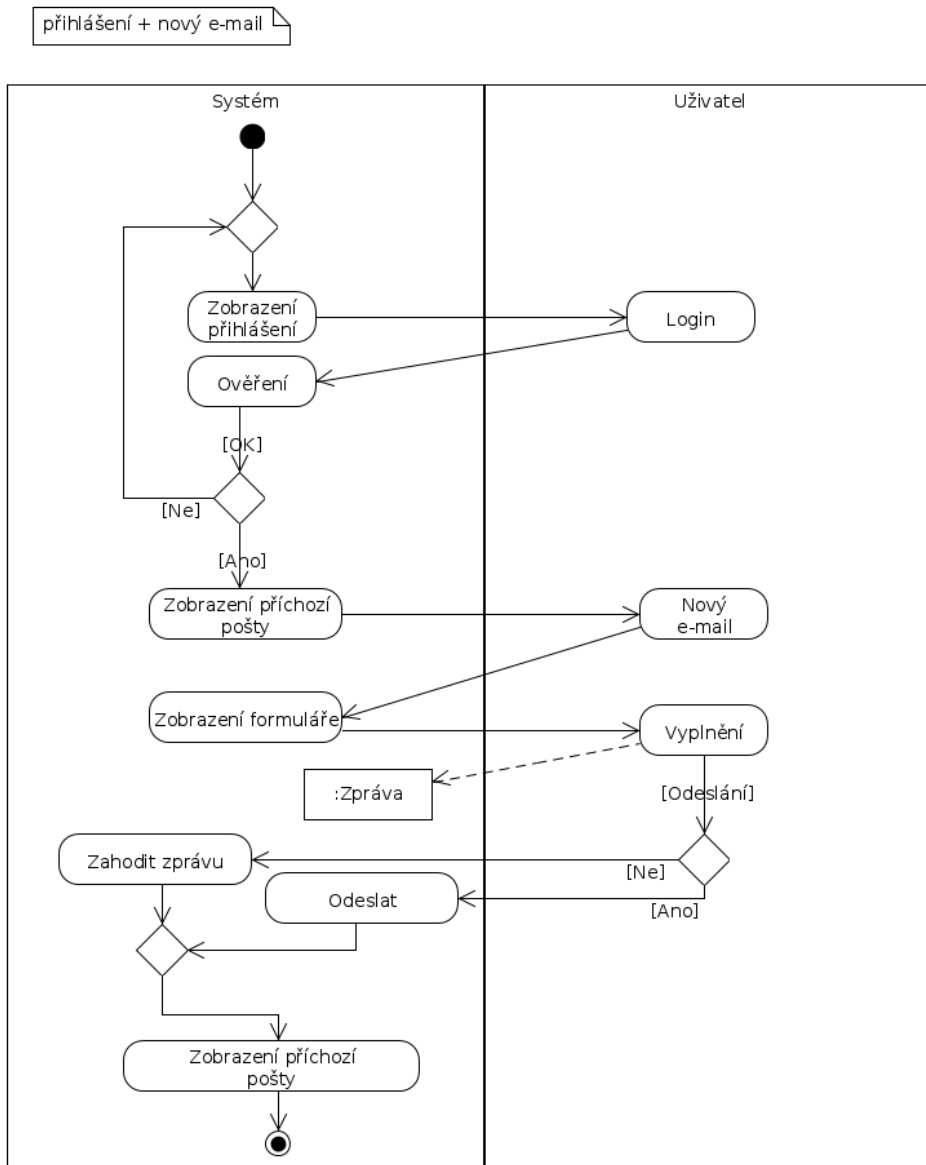
Vyučující má k dispozici plnou funkčnost aplikace. Může rozesílat e-maily celé paralelce. Dále je v roli vyučující plně funkční podbarvování, řeší se zde vztah student/vyučující, ale i vyučující/vyučující, kteří vyučují ve stejné hodině.

2.3.1.2 Student

Uživatel s rolí studenta není oprávněn k plné funkčnosti aplikace, nemá možnost rozesílání e-mailu celé paralelce a dále e-maily od ostatních studentů nejsou podbarvovány a neřeší se jejich vzájemné vztahy příjemce a odesílatele.

2.3.2 Scénáře

- **Autorizace** - uživatel se přihlásí pomocí username a školního hesla a bude školním serverem ověřen.
- **Přihlášení do zařízení** - uživatelé se přihlásí vyplněním svých přihlašovacích údajů. Jedná se o školní username a školní heslo (viz obrázek 2.2).
- **Inicializace zařízení** - probíhá na vyžádání uživatele. V případě, že zařízení je připojeno k internetu, dojde ke stažení údajů o uživateli, jeho hodinách za celé dosavadní studium a jeho poště z imap.fit.cvut.cz.
- **Přijmutí e-mailu** - aplikace stáhne a následně zobrazí všechny přijaté e-maily. Uživateli se po kliknutí na příslušný e-mail tato zpráva zobrazí na celé obrazovce a bude pohodlně k přečtení.
- **Odeslání nového e-mailu** - aplikace zobrazí formulář pro vyplnění zprávy. Uživatel vyplní příjemce, předmět a text. Po kliknutí na tlačítko se e-mail odešle. Jestliže uživatel ukončí psaní zprávy před odesláním, e-mail bude zahozen (viz obrázek 2.2).
- **Přeposlání e-mailu** - aplikace bude umožňovat přeposílání e-mailu, v jeho detailu bude umístěno tlačítko v horní liště. Jestliže uživatel vybere tuto možnost, aplikace ho přesměruje na předvyplněný formulář nového e-mailu.



Obrázek 2.2: Diagram aktivit

- **Odpověď na e-mail** - aplikace bude umožňovat odpověď na e-mail, v jeho detailu bude umístěno tlačítko v horní liště. Jestliže uživatel vybere tuto možnost, aplikace ho přesměruje na předvyplněný formulář nového e-mailu.
- **Zobrazení hodin** - aplikace bude také umožňovat zobrazení seznamu hodin přihlášeného uživatele za celé jeho dosavadní studium. Následně i detail jednotlivé hodiny, kde jsou zobrazeny veškeré informace.
- **Odhlášení** - jestliže se uživatel odhlásí, dojde k odstranění veškerých stažených dat a vyčištění aplikace pro nového uživatele.

2.4 Databáze

V požadavcích výše bylo uvedeno, že aplikace si musí stažená data uložit do vlastní databáze v zařízení, aby mohla pracovat offline a aby nezdržovala uživatele stahováním při používání. Data musí být uložena v databázi a uživatel má právo je jen číst a doplňovat, nikoli upravovat.

Nejefektivnější způsob, jak data uložit do paměti zařízení je, použít relační databázi. V každém zařízení OS Android je k dispozici databáze SQLite, proto jsem se tuto databázi rozhodl použít. Každá aplikace má svou vlastní databázi, kterou si vytvoří. [10]

2.4.1 Datové typy

Databáze SQLite nám nabízí několik datových typů pro jednotlivé sloupce tabulek.

- **NULL** - značí prázdnou hodnotu
- **INTEGER** - celé kladné nebo záporné číslo
- **REAL** - desetinné číslo s plovoucí čárkou
- **TEXT** - řetězec znaků

2.4.2 Tabulky

Návrh je přizpůsoben staženým datům z datového serveru Kosapi a e-mailového serveru imap.fit.cvut.cz.

- **Contact** - tabulka uchovává společná data pro jednotlivé osoby, ať již vyučující tak studenty. Je používána například při sestavování detailu hodiny, kde jsou vždy podrobné informace o vyučujících.
- **Teacher** - v tabulce "Teacher" jsou uložena doplňující data o každém z vyučujících.

- **Teaching** - podpůrná tabulka, která pomáhá realizovat vztah "N ku M", jelikož každá hodina může mít více vyučujících a zároveň každý vyučující může učit na více hodinách
- **Student** - v tabulce "Student" jsou uložena doplňující data o každém ze studentů.
- **Studying** - podpůrná tabulka, která pomáhá realizovat vztah "N ku M", jelikož každá hodina má více studentů a zároveň každý student dochází na více hodin.
- **Subject** - v tabulce "Subject" uchovává stažená data o jednotlivých hodinách, které uživatel za celou dobu studia navštěvoval nebo které vyučující učil.
- **TimeTableSlot** - toto je podpůrná tabulka k tabulce "Subject", který reprezentuje čas a místo konání jednotlivé hodiny.
- **E-mail** - tabulka uchovává veškerou odchozí a příchozí poštu.
- **Sending** - podpůrná tabulka, která pomáhá realizovat vztah "N ku M", jelikož každý e-mail může mít více odesílatelů i příjemců.

2.4.3 Dotazy

Databáze SQLite umí pracovat se standardními SQL dotazy za pomoci SQLite adaptéru (popsán v části implementace 3.5). Adaptér můžeme použít k vytvoření, naplnění, změnění i smazání jednotlivých tabulek. Pomocí adaptéru můžeme také spouštět přímo SQL dotazy nad databází, čehož aplikace hojně využívá a komunikuje s databází pouze pomocí tohoto adaptéru.

2.5 Platforma Android

2.5.1 Obecné informace

Po původním průzkumu platformy Android v kapitole 1.5 jsem se rozhodl, že aplikace bude podporovat verze Androidu 4.0 a vyšší. Tímto krokem pokryji více jak 97% mobilních telefonů, jak vyplývá z tabulky 1.7, a do budoucna se toto procento bude dále zvyšovat s tím, jak budou vyměňována dosluhující zařízení s nižší verzí Androidu. Kompatibilita s vyššími verzemi je zaručena již od tvůrců operačního systému Android a nebudu se jí muset dále zabývat.

2.5.2 Aktivity

Aktivitou je nazývána každá jednotlivá obrazovka aplikace, kterou vidí uživatel. Tato obrazovka obsahuje grafické uživatelské rozhraní, se kterým následně

samotný uživatel pracuje. Uživatel při práci s aplikací prochází jednotlivé aktivity, které se zobrazují a následně mizí. Tyto aktivity si mohou při uživatelském průchodu aplikací předávat informace a díky tomu mezi sebou komunikovat. Aktivity mají svůj životní cyklus, který je rozdělen do několika stavů.

- **Start** - aktivita je vytvořena a inicializována.
- **Running** - aktivita, se kterou uživatel komunikuje. V tomto stavu je pouze jedna aktivita a to ta, která je zobrazena na displeji zařízení.
- **Stop** - aktivita již není vidět na displeji zařízení, nepřijímá uživatelský vstup, ale není zničena. Je zde předpoklad, že uživatel se k ní ještě bude chtít vrátit.
- **Destroy** - aktivita byla zničena "Activity Managerem", který ji ukončil buď z důvodů nedostatku místa nebo na základě kódu napsaného vývojářem.

Při změně stavů aktivity se volají jednotlivé "override" metody. Pomocí těchto metod vývojář určuje, co se má s aktivitou dít v jednotlivých přechodech mezi stavy [11] [12].

2.5.3 Fragments

Fragment je prvek grafického rozhraní, který obaluje nějaký uzavřený celek. Aktivita může obsahovat více těchto fragmentů. Životní cyklus fragmentu je stejný jako životní cyklus aktivity. Tudíž stejně jako aktivita se po přechodu z fragmentu pryč, uloží do paměti a po následném opětovném otevření se již nemusí pracně vytvářet, ale pouze se načte.

Fragmenty využívám v prvku "Navigation Drawer" (práce s ním je popsána v kapitole 3.6.1). Zde je fragment roven jedné obrazovce menu.

2.6 Návrh uživatelského rozhraní

Při návrhu uživatelského rozhraní jsem se snažil aplikaci navrhnout co nej-jednodušeji a nejprívětivěji pro uživatele. Inspiroval jsem se jinými aplikacemi pro OS Android, které podporují práci s e-maily.

2.6.1 Doručené a Odeslané

Obrazovka Doručené se zobrazuje při zapnutí aplikace a je to první věc, kterou uživatel uvidí. Správné navržení a fungování úvodní obrazovky je velmi důležité, protože často rozhoduje o tom, zda uživatel bude aplikaci nadále používat. Automaticky se zde zobrazí seznam přijatých e-mailů. Vidíme, kdy byl e-mail přijat, dále se zobrazuje předmět a odesílatel. Odesílatel e-mailu je zobrazen

buď pouze jako e-mailová adresa, jestliže není tento kontakt na vyučujícího, případně studenta uložen v databázi. Pokud je, zobrazí se celé jméno odesílatele e-mailu. Dále zde uživatel uvidí svůj vztah s odesílatelem. Tuto funkci ocení hlavně uživatelé z řad vyučujících, kteří ještě před otevřením samotného e-mailu zjistí, zda se jedná o studenta, kterého cvičí nebo mu přednáší (screenshot fragmentu na obrázku 2.3). Po kliknutí na jednotlivý e-mail ze seznamu se otevře Detail e-mailu popsany dále.

V horní části na "ActionBar"(horní lišta) je přidáno tlačítko aktualizace složky Doručené. Při kliknutí na něj se aplikace připojí k serveru a pokusí se stáhnout nové e-maily.

Odeslané používají stejnou šablonu jako Doručené, jelikož se jedná o zobrazení e-mailů, ale tentokrát z jiné složky. Po kliknutí na vybraný e-mail se otevře aktivita Detail e-mailu.

Obě tyto obrazovky jsou realizovány pomocí Fragmentu, jelikož jsou napojeny na Main aktivitu, pomocí které je realizováno vyjždějící menu "NavigationDrawer".



Obrázek 2.3: Doručené



Obrázek 2.4: Detail e-mailu

2.6.2 Detail e-mailu

Aktivita Detail e-mailu se otevře na vyžádání uživatele. Obsahuje všechny informace o e-mailu (příjemce, odesílatel, předmět, čas odeslání, samotný text

2. NÁVRH ŘEŠENÍ

e-mailu) i vztah, který mezi sebou odesílatel a příjemce mají. V aktivitě je použito "ScrollView"(realizuje skrolování na stránce, jestliže obsahuje více informací než se na displej vejde) a to tak, že je aplikováno jenom na část obsahu emailu. Tudíž hlavička se základními informacemi je stále zobrazena na displeji.

Zde jsou také do "ActionBar" přidána další dvě tlačítka. Konkrétně pro přeposlání e-mailu a odpověď na e-mail. Po kliknutí na jakékoliv z tlačítek se otevře aktivita Nový e-mail s předvyplněnými informacemi.

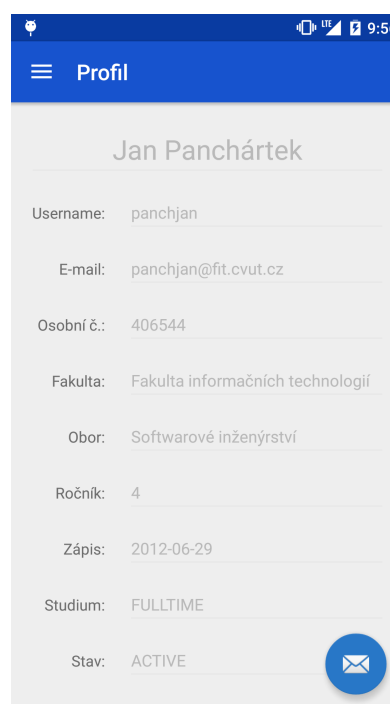
Tato obrazovka je ukázána na obrázku 2.4.

2.6.3 Nový e-mail

Do aktivity Nový e-mail se uživatel může dostat hned několika způsoby. První je pomocí tlačítka v Main aktivitě umístěného v pravém dolním rohu, zobrazeného na každém fragmentu této aktivity. Další dvě možnosti jsou zmíněny již v předchozí podkapitole a jsou to přeposlání e-mailu a odpověď na e-mail. Poslední možností, jak se dostat do této aktivity je přes fragment Hodiny a zde kliknutím na konkrétního vyučujícího, studenta nebo celou paralelku. U této možnosti se předvyplní příjemce a předmět e-mailu (jak je vidět na obrázku 2.5).



Obrázek 2.5: Nový e-mail



Obrázek 2.6: Profil

Tentokrát je v "ActionBar" umístěno pouze jedno tlačítko a to na samotné odeslání e-mailu. Po uživatelské kliknutí aplikace udělá kontrolu, jestli byla vyplněna pole příjemce a předmět a dále jestli má uživatel přidán účet do zařízení a tudíž nastavené heslo pro práci s e-mailu.

2.6.4 Profil

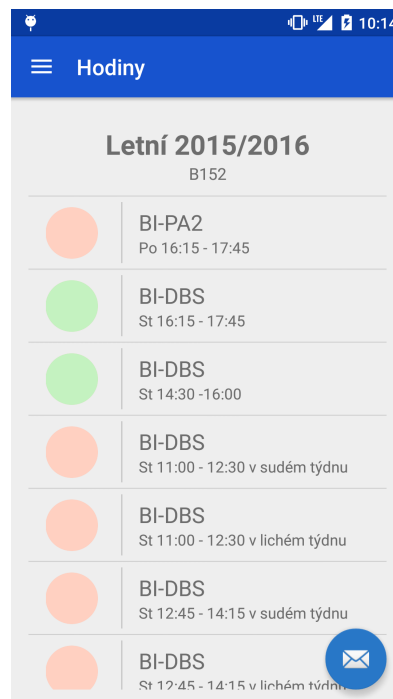
Profil je první z fragmentů, který je upraven podle uživatelské role. Již z návrhu aplikace vyplývá, že o vyučujících jsou uchovávána jiná data než o studentech. Například přihlášenému studentovi aplikace zobrazuje začátek studia, ročník nebo formu studia, kdežto vyučující má informace jiné. Ukázka obrazovky studentova profilu je na obrázku 2.6.

2.6.5 Seznam hodin a jejich detail

Seznam hodin je dalším fragmentem spadajícím do Main aktivity. Tento fragment je různý pro vyučující a studenty, ale jsou zde jen drobné úpravy. Student má zapsaný předmět jedenkrát v semestru a proto jej aplikace nemusí odlišovat. Naproti tomu vyučující, ve většině případů, vyučuje předmět vícekrát a proto zde muselo dojít k odlišení těchto stejných předmětů. To bylo docíleno pomocí času, kdy se konkrétní hodina vyučuje 2.8. Porovnání obou obrazovek je vidět níže.



Obrázek 2.7: Studentovi hodiny



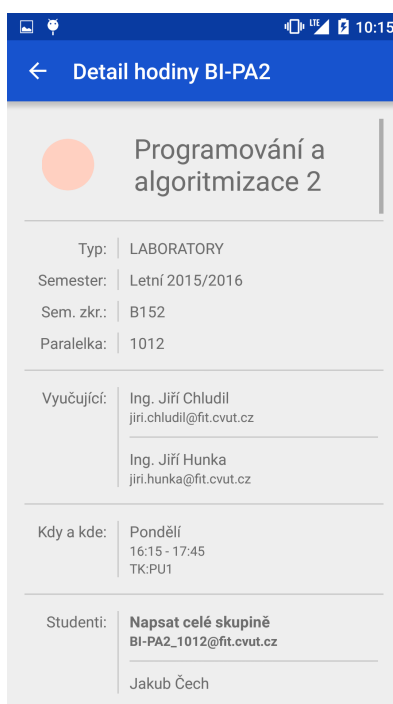
Obrázek 2.8: Hodiny vyučujícího

2. NÁVRH ŘEŠENÍ

Tento fragment je reprezentován seznamem hodin, které měl student zapsané za celou dobu studia nebo které vyučující vyučoval po dobu působení na fakultě. Každá hodina je zobrazena jako jeden řádek, v kterém je zobrazen typ hodiny, jméno a zkratka hodiny. Typ hodiny je zobrazen pomocí barevných puntíků. Barvy jsou stejné jako ve školním portálu timetable.fit.cvut.cz.

Detail hodiny je další aktivita, která je vytvořena po uživatelské kliknutí na hodinu ze zobrazeného seznamu. V tomto detailu jsou zobrazeny veškeré informace o dané hodině. Vyučující zde mají kompletní seznam studentů, seznam kolegů vyučujících, ale i veškeré informace o tom, kdy se daná hodina koná (na obrázku 2.9). Student má tento detail hodiny částečně upraven. Není zobrazován seznam ostatních studentů, kteří na tuto paralelku dochází. To je z důvodu zrychlení aplikace pro studenty. Většina studentů informace o svých spolužácích vůbec nevyužije a tato data by jenom zdržovala rychlost aplikace.

Po kliknutí v Detailu hodiny na jednotlivé osoby nebo v případě vyučujícího kliknutí na celou paralelku je vytvořena nová aktivita Nového e-mailu s předdefinovaným předmětem a příjemcem. Díky této funkci studenti nemusí zjišťovat jméno a e-mail vyučujícího, který jim přednáší daný předmět, když mu chtějí napsat.



Obrázek 2.9: Detail hodiny

2.6.6 Nastavení

Fragment Nastavení je podpůrná obrazovka, v které uživatel může přidat účet do zařízení, přihlásit se k školnímu datovému serveru, nebo vyvolat stahování dat.

Implementace

Tato kapitola pojednává o použitých nástrojích pro implementaci, jejich možnosti a samotné použití v mé aplikaci.

3.1 Přihlašování

V každé aplikaci je přihlašování jednou z nejdůležitějších částí a každý by se měl nejdříve zamyslet nad tím, kam se přihlásí pomocí svých přihlašovacích údajů a zdali je to bezpečné. Asi většina z nás přemýšlí jak spravovat e-maily. Jestli mít pro každý typ účtu oficiální prověřenou aplikaci, a tudíž pracovat například s třemi různými aplikacemi, nebo sáhnout po nějaké neoficiální, kde nás může napadat otázka: "Co vlastně s našimi přihlašovacími údaji dělá? Jak si je ukládá? Neodesílá je někam?".

Mnou vytvořená aplikace potřebuje dvojitě přihlášení kvůli vysoké míře bezpečnosti. Pro možnost stahování informací o uživateli z datových serverů, je vyžadováno přihlášení k portálu AppsManager, spravovaným ČVUT. Následné stahování je realizováno pomocí autorizačního tokenu (dále popsáno v podkapitole 3.1.1). Pro práci s e-mailem potřebuje aplikace přímo uživatelské heslo, jeho uchovávání v zařízení a práce s ním je popsána v podkapitole 3.1.2.

3.1.1 Autorizace

Při prvním spuštění vyžaduje vytvořená aplikace přihlášení, na jehož základě ověří, zda dotyčný uživatel je zaregistrován v databázi naší fakulty. Pokud ano, tak následně provede stažení základních informací vztahujících se k dotyčnému uživateli. Poté si on může stáhnout všechny své rozvrhy hodin za dosavadní studium. Ke každé hodině tak získá veškeré informace včetně kontaktů na vyučující, případně i kontakty na studenty zapsané v dané hodině, jestliže přihlášený uživatel je vyučujícím. Toto se již děje pod jeho identitou (autorizační token), jelikož je toto stahování zabezpečené.

3.1.1.1 AppsManager

Autorizace pro školní datové servery je vyřešena velice elegantní cestou a to přes AppsManager. Tuto webovou aplikaci spravuje ČVUT. Vývojáři mobilních aplikací se zde přihlásí a zaregistrují si svou aplikaci. Získají ID aplikace a vyberou si jaké API chtějí ve své aplikaci používat (KOSapi, Sirius, UsernameAPI, VVVSapi).

Následně již stačí vytvořit v aplikaci aktivitu ve které přesměrujeme uživatele do přihlášení k AppsManageru, který udělá většinu práce. Ověří zadané údaje, v případě špatného přihlášení uživatele informuje a pouze při správném přihlášení předá aplikaci uživatelské jméno a autorizační token. Tento token slouží jako klíč do výše zmíněných datových serverů. V autorizaci se tímto způsobem využívá OAuth standardu.

3.1.1.2 OAuth 2.0

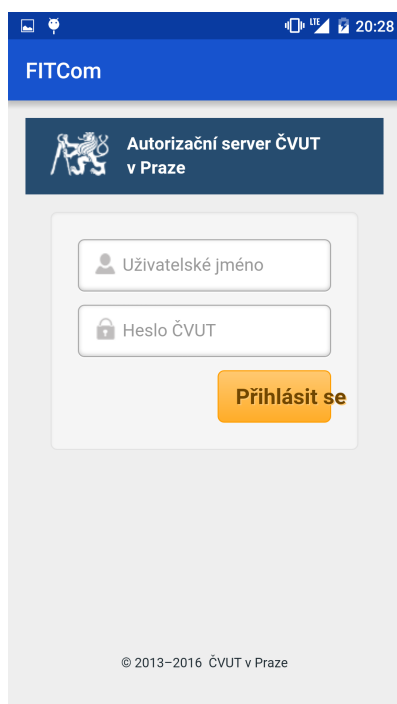
OAuth je protokol poskytující bezpečnou autorizaci pro mobilní aplikace. Velkou výhodou je, že uživatel neposkytuje své přihlašovací údaje přímo aplikaci, ale webové službě (v našem případě AppsManager). Toto je založené na přesměrování do AppsManageru, který uživatele vyzve k přihlášení. To již neprobíhá v rámci mobilní aplikace, ale v rámci webové služby, tudíž není způsob, jak tuto autorizaci narušit. Po správném přihlášení je aplikace opět přesměrována na libovolně zvolenou URL adresu. Zde se z ní vyčte autorizační token, uživatelské jméno a kdy daný token expiruje. Tokeny mají omezenou platnost, po kterou s nimi lze ze školních datových serverů stahovat data. Zpravidla se tato doba rovná jedné hodině. Poté pro další stahování musí dojít k opětovnému přihlášení. Díky těmto postupům je tato autorizace bezpečná [13].

3.1.1.3 Použití

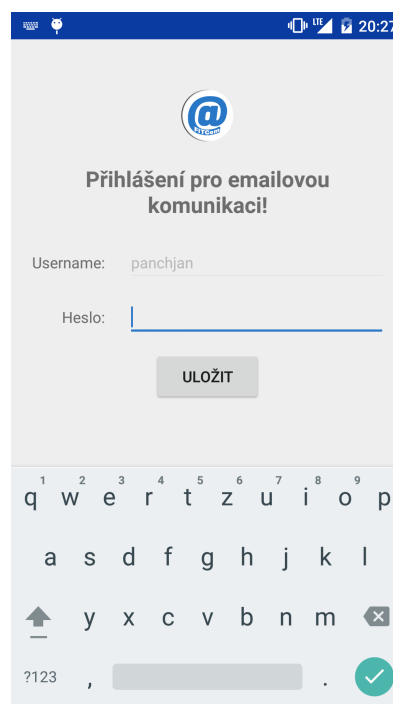
Proces autorizace a získání autorizačního tokenu probíhá v aktivitě AuthorizationActivity (viz obrázek 3.1). Tato aktivita obsahuje pouze prvek "WebView", v kterém se uživateli zobrazuje přesměrovaná stránka AppsManageru. Obsluhu objektu typu "WebView" zajišťuje metoda setWebViewClient(), která jako parametr vyžaduje instanci třídy "WebVewClient". V této třídě dochází k implementování dvou override metod. A to shouldOverrideUrlLoading() a onPageFinished(). V těchto metodách dochází k samotnému přesměrování a následnému získání výše zmíněných údajů o přihlášeném uživateli.

3.1.2 Účet v zařízení

V další řadě potřebuje aplikace přímo heslo a username uživatele, aby mohla přistupovat ke školnímu e-mailovému serveru IMAP. Problém s uchováváním těchto citlivých údajů je řešen pomocí třídy "AccountManager". Přihlášení probíhá v aktivitě "LoginActivity" (viz obrázek 3.2), kde uživatel vyplní pouze



Obrázek 3.1: AutorizationActivity



Obrázek 3.2: LoginActivity

heslo. Username je již předdefinováno podle přihlášeného uživatele. Tato aktivita může být vyvolána z nastavení aplikace, nebo přímo z nastavení zařízení ze sekce účty, kde si jej ale musí uživatel manuálně vytvořit.

3.1.2.1 AccountManager

Tato třída spravuje účet aplikace v zařízení a zajišťuje celkovou bezpečnost. Pomocí jednoduché konstrukce se účet v zařízení vytvoří, uloží se uživatelské jméno, heslo a typ účtu [14].

3.1.2.2 Vytvoření účtu

Pro vytvoření účtu používám podpůrnou třídu "Account". Vytvořením této třídy získám instanci objektu, ve kterém je uloženo username a unikátní řetězec charakterizující aplikaci.

Dále již jednoduše pomocí metody `addAccountExplicitly()`, nad třídou "AccountManager", účet v telefonu vytvořím. Parametry této metody jsou výše vytvořený objekt třídy "Account" a uživatelské heslo (jak je vidět v kódu níže).

3. IMPLEMENTACE

```
1 AccountManager accMg = AccountManager.get(getApplicationContext());
2 Account account = new Account(username,
   "cz.cvut.fit.panchjan.fitcom");
3 accMg.addAccountExplicitly(account, password, null);
```

3.1.2.3 Získání hesla

Při práci s e-maily (odesílání nebo stahování e-mailů) je potřeba heslo. Uživatelovo heslo aplikace získá z již vytvořeného účtu v zařízení a to díky metodě `getPassword()`, kterou zavoláme nad proměnnou třídy "AccountManager" a jako parametr vložíme objekt třídy "Account". Metoda nám vrátí heslo jako základní datový typ "String".

3.1.2.4 Odstranění účtu

K odstranění účtu ze zařízení dochází v době, kdy se uživatel odhlásí. Vše jde opět velice jednoduše. Nad proměnnou třídy "AccountManager" zavolám metodu `removeAccountExplicitly()`, která účet ze zařízení odstraní. Parametr této metody je opět objekt třídy "Account", který musí být naprosto stejný jako ten při zakládání účtu.

3.2 Stahování dat

Veškerá data použitá v aplikaci jsou stahována z informačního serveru KOS, pomocí API Kosapi. Kosapi poskytuje veškerá data týkající se rozvrhů, studentů, vyučujících, místností, paralelek a další užitečné informace.

Data stahovaná z Kosapi jsou ve formátu XML, což je obecný značkovací jazyk. Tento jazyk byl vyvinut právě na výměnu dat mezi aplikacemi, kde záleží na věcném obsahu a nezabývá se vzhledem. Obsah souboru je tvořen pomocí tagů (značek), které slouží pro vyznačování významu a obsahu [15].

3.2.1 Parametry

Stahování dat je realizováno pomocí GET dotazu. Každý tento dotaz následuje skupina parametrů, které upravují výsledek dotazu. Tyto parametry jsou vkládány do URL adresy za otazník ("?"). Odděluje je znak ampersand("&") a jsou až na jednu výjimku nepovinné. Jediný povinný parametr je autorizační token [16].

Kosapi podporuje mnoho parametrů. Z nich já využívám v GET dotazech tyto tři:

- **accessToken** - povinný parametr, jedná se o autorizační token, přes který dojde k ověření.

- **limit** - upravuje počet výsledků, a defaultně je nastaven na hodnotu 20 a omezen shora hodnotou 1000.
- **sem** - upravuje, které semestry uvidíme. Jestliže parametr neuvedeme, zobrazí se pouze aktuální semestr. Pokud jej uvedeme, musíme zvolit kód semestrů, které chceme dostat. Potřebujeme-li všechny semestry, hodnota parametru se musí rovnat "none".

3.2.2 URL pro GET požadavky

Kosapi nabízí velké množství URL adres, na které jde přistupovat pomocí GET požadavků. Moje aplikace využívá jenom zlomek z nich. Konkrétně jich je pět a jsou ještě následně různě parametrizovány (viz parametry výše 3.2.1) [17].

- `/students/{username}/parallels` - pod tímto požadavkem vypíše Kosapi všechny zapsané hodiny studenta podle jeho username.
- `/teachers/{username}/parallels` - obdobně, jako o řádek výše požadavek vrací všechny hodiny, které vyučují, přednáší nebo cvičí.
- `/parallels/{ID}/students` - tento požadavek vypíše z Kosapi informace o studentech v dané hodině (paralelce) podle jejího ID.
- `/students/{username}` - pod tímto požadavkem aplikace dostane výsledek informace o daném studentovi podle jeho uživatelského jména.
- `/teachers/{username}` - obdobně, jako o položku výše, požadavek vrací informace o daném vyučujícím podle jeho uživatelského jména.

Jednotlivé výsledky jsou ve formě XML a aplikace je rozparsuje (viz podkapitola Parsování XML 3.3) a následně vytvoří pole objektů Student, Teacher, Subject, a další. Tyto třídy jsou vytvořeny a navrženy podle dat, které vrací Kosapi. Data jsou následně uložena do databáze (viz podkapitola Ukládání dat 3.5).

3.3 Parsování XML

Parsování XML souborů vytváří ze stažených textových souborů pole objektů v paměti zařízení. Tato data jsou následně ukládána do databáze a některá popřípadě do třídy "SharedPreferences". Pro parsování dat aplikace využívá třídy "XmlPullParser". S touto třídou je zmíněné převedení textového souboru na objekty jednoduché.

3.3.1 XmlPullParser

Nejprve po vytvoření proměnné typu "XmlPullParser" je objekt nastaven pomocí metod `setFeature()` a `setInput()`.

Dále již jednoduše procházíme kód pomocí metody `next()`. Názvy tagů získáváme metodou `getName()`. Data, ať už text nebo čísla mezi tagy, již jednoduše dostaneme pomocí metody `getText()`. Čísla musíme následně upravit do číselné podoby konstrukcí `"Integer.parseInt(parser.getText())"` [18].

V ukázce je vidět část XML souboru, který je stažen aplikací z datového serveru Kosapi:

```
1 <atom:content atom:type="xml" xsi:type="teacher">
2   <firstName>Jiri</firstName>
3   <lastName>Hunka</lastName>
4   <personalNumber>330145</personalNumber>
5   <titlesPre>Ing.</titlesPre>
6   <username>hunkajir</username>
7   <division xlink:href="divisions/18102/">katedra
      softwaroveho inzenyrstvi</division>
8   <email>jiri.hunka@fit.cvut.cz</email>
9   <extern>>false</extern>
10 </atom:content>
```

V ukázce kódu níže je znázorněno parsování XML z ukázky výše. Metoda vrací objekt typu `Teacher`, obsahující data o konkrétním vyučujícím:

```
1 private Teacher readTeacher(XmlPullParser parser) throws
      XmlPullParserException, IOException {
2
3   parser.require(XmlPullParser.START_TAG, ns, "atom:entry");
4
5   while (parser.next() != XmlPullParser.END_TAG) {
6
7     String name = parser.getName();
8     if (name.equals("atom:content")) {
9       parser.require(XmlPullParser.START_TAG, ns, "atom:content");
10
11      while(parser.next() != XmlPullParser.END_TAG){
12        String subName = parser.getName();
13        if(subName.equals("firstName")){
14          firstName = readText(parser, "firstName");
15        }
16        else if(subName.equals("lastName")){
17          lastName = readText(parser, "lastName");
18        }
19        else if(subName.equals("username")){
20          username = readText(parser, "username");
21        }
22      }
23    }
24  }
25 }
```

```
22     ...
23     else{
24         skip(parser);
25     }
26 }
27 } else {
28     skip(parser);
29 }
30 }
31
32 Teacher teacher = new Teacher(firstName, lastName, titlePre,
33                               titlePost, email, username, personalNumber, division,
34                               shortCutDivision, extern);
33 return teacher;
34 }
```

3.4 Práce na pozadí

Práce na pozadí je jedna z nejdůležitějších součástí programování mobilních aplikací, kterou realizujeme pomocí implementace vedlejšího vlákna zděděním třídy "AsyncTask". Hlavní zobrazovací vlákno tudíž nemusí dělat složité výpočty a operace, například stahování dat z internetu. Kdybychom toto stahování realizovali pomocí hlavního vlákna aplikace "zamrzne" a uživatel není schopen nic dělat, dokud nebude stahování dokončené.

3.4.1 AsyncTask

Třída "AsyncTask" vytváří vedlejší vlákno a pomáhá hlavnímu zobrazovacímu vláknu realizovat složité a časově náročné operace, které by hlavní vlákno samo nebylo schopno efektivně udělat. Tato třída implementuje čtyři důležité metody "onPreExecute", "onInBackground", "onProgressUpdate" a "onPostExecute", pomocí kterých vykonává svou činnost a informuje hlavní zobrazovací vlákno o své činnosti [18] [19].

3.4.1.1 onPreExecute

Metoda "onPreExecute" se vyvolá před samotnou prací na pozadí a často se v ní zobrazuje dialogové okno s informací o průběhu prací na pozadí. V další řadě se často "deaktivují klikatelné" prvky na obrazovce, pomocí kterých by uživatel mohl aplikaci zacyklit, například opakovaným stahováním stejných informací.

V ukázce vidíme metodu z vedlejšího vlákna stahující e-maily. Zde se uživateli zobrazuje pouze progress bar:

3. IMPLEMENTACE

```
1 @Override
2 protected void onPreExecute() {
3     if(progress != null){
4         progress.setVisibility(View.VISIBLE);
5     }
6 }
```

3.4.1.2 onInBackground

V této metodě se realizuje samotný výpočet, například již zmíněné stahování dat z internetu. Tato metoda musí být vždy definována, jestliže chceme dědit třídu "AsyncTask".

Zde je ukázka jak vedlejší vlákno stahuje e-maily:

```
1 @Override
2 protected List<Email> doInBackground(String... params) {
3
4     int downloadCountBefore = Integer.parseInt(params[0]);
5     int downloadCountNow = Integer.parseInt(params[1]);
6     String userName = params[2];
7     String password = params[3];
8
9     EmailReceiver receiver = new EmailReceiver();
10    List<Email> inboxEmails = receiver.downloadEmails("imap",
11        "imap.fit.cvut.cz", "993", userName, password, "INBOX",
12        downloadCountBefore, downloadCountNow);
13
14    return inboxEmails;
15 }
```

3.4.1.3 onProgressUpdate

Metoda, kterou volá vlákno spuštěné na pozadí a informuje v ní hlavní zobrazovací vlákno o pokroku výpočtu. Díky tomu hlavní vlákno může zobrazovat například průběh stahování v progress baru, případně jiném dialogovém okně.

3.4.1.4 onPostExecute

Tato metoda se vyvolá po skončení výpočtu, tj. po dokončení činnosti v metodě "onInBackground". Používá se pro informování hlavního vlákna o dokončení práce a zavírá dialogové okno, které informovalo uživatele o jejím průběhu.

3.4.1.5 Použití třídy AsyncTask

Třídu "AsyncTask" v aplikaci používám na několika místech. Konkrétně při autorizaci, přihlášení, stahování dat o uživateli a jeho hodinách a hlavně pro pří-

jímání a odesílání e-mailů.

Každé spuštění vedlejšího vlákna se naváže na aktivitu nebo fragment, z kterého je vlákno spuštěno. Tudíž, když je aktivita nebo fragment zničen, dojde k zastavení i vedlejšího vlákna. Výpočet není dokončen, není zavolána metoda "onPostExecute" a vlákno nedojde k očekávanému výsledku.

Tento problém řeším u stahování dat o uživateli a jeho hodin v dosavadním studiu. Fragment, ze kterého je stahování vyvoláno, je disablován a uživateli je hlavním vláknem vykreslen progress bar (točící se modré kolečko), ale jestliže uživatel přes vyjíždějící menu otevře jiný fragment, bude stahování přerušeno. Proto je do fragmentů, kde dochází k vytváření vedlejšího vlákna vložena metoda `setRetainInstance(true)`, díky které nedojde ke zničení vedlejšího vlákna a fragmentu. Tento nezničený fragment se uloží a již nebude znovu vytvářen, ale pouze znovu zobrazen.

3.5 Ukládání dat

Data se ukládají do zařízení ze dvou důvodů. Prvním důvodem je, aby po každé aplikaci nemusela stahovat množství dat o uživateli, jeho hodinách a e-mailech. Kdyby se tak dělo, aplikace by byla pomalejší při každém jejím opětovném otevření. Druhým důvodem je to, že aplikace funguje a zobrazuje již dříve stažená data i v době, kdy nemáme možnost připojení k internetu. K uložení dat v zařízení využívá databázi. Ta obsahuje informace o samotném přihlášeném uživateli, jeho hodinách za celé studium, vyučujících na jednotlivých hodinách, ale i samotné e-maily. Po prvním přihlášení do aplikace se databáze naplní daty.

3.5.1 SQLite databáze a práce s ní

Databáze je nejpoužívanějším nástrojem pro ukládání dat, které jsou strukturované a často se opakují jejich čtení. V androidích aplikacích se využívá databáze SQLite, která je Open Source. Tato databáze je obsažena v každém zařízení s Androidem a díky tomu její využití nevyžaduje žádné nastavení. Není to plnohodnotná databáze, jako například MySQL nebo PostgreSQL, protože se jedná pouze o malou knihovnu, která neběží samostatně a zprostředkovává otevírání datových souborů a práci s nimi pomocí jazyka SQL. Databáze vyžaduje pouze definování SQL příkazů pro vytvoření a aktualizaci.[20]

Všechny třídy, které potřebujeme k vytvoření, naplnění, aktualizaci a další práci s databází, jsou umístěny v balíčku "android.database". Ten je standardní součástí projektu typu Android. Aplikace používá tři třídy a to "SQLiteOpenHelper", "SQLiteDatabase" a "Cursor".

3.5.1.1 SQLiteOpenHelper

Třída "SQLiteOpenHelper" je využita pro vytvoření a aktualizaci databáze pomocí abstraktní vrstvy a to tak, že třídu zdědíme a tudíž musíme implementovat metody "onCreate" a "onUpgrade".

3.5.1.2 SQLiteDatabase

Třída "SQLiteDatabase" je základní nástroj pro samotnou práci s databází v zařízení. Poskytuje základní metody pro práci s databází, vložení (insert()), aktualizace (update()), smazání (delete()), dotazování (rawQuery()) a zavírání databáze (close()).

3.5.1.3 Cursor

Při vytvoření třídy "Cursor" pomocí metody rawQuery() se do jejího objektu uloží výsledek jakéhokoli SQL dotazu. Poté lze manipulovat s jednotlivými sloupci v řádcích. Před procházením výsledků se pomocí metody moveToFirst() posuneme na začátek prvního výsledku, další posuny realizujeme metodou moveToNext(), který posune cursor na další řádek výsledné tabulky. Dále, jestliže chceme z cursoru dostat výsledek, použijeme metody getInt(), getLong() nebo getString(), kterými určíme jaký datový typ je uložen ve sloupci, který chceme číst. Samotný sloupec označíme metodou getColumnIndex(). [21]

Zde získávám ID kontaktu podle e-mailu, za pomocí metody rawQuery():

```
1 public int getIDContact(String email) {
2
3     String query = "SELECT * FROM " + TABLE_NAME_CONTACT + " WHERE " +
4         EMAIL + " = '" + email + "'";
5
6     Cursor cursor = db.rawQuery(query, null);
7
8     if (cursor.getCount() == 1) {
9         cursor.moveToFirst();
10        return cursor.getInt(cursor.getColumnIndex(ID_CONTACT));
11    } else {
12        return -1;
13    }
```

3.5.1.4 Insert

Pro uložení dat aplikace využívá třídu "ContentValue", v které pomocí metody put() definuji dvojici klíč a hodnota. Vytvořený objekt této metody reprezentuje jeden řádek tabulky a následně ho metodou insert() nad objektem SQLiteDatabase vložíme do databáze.

V ukázce kódu níže je vidět vložení dat o vyučujícím do databáze:

```
1 public int insertTeacher(Teacher teacher) {
2
3     int idContact = insertContact(teacher);
4
5     ContentValues content = new ContentValues();
6     content.put(ID_CONTACT, idContact);
7     content.put(DIVISION, teacher.getDivision());
8     content.put(SHORT_CUT_DIVISION, teacher.getShortCutDivision());
9     content.put(EXTERN, teacher.getExtern());
10
11     long idTeacher = db.insert(TABLE_NAME_TEACHER, null, content);
12     return (int) idTeacher;
13 }
```

3.5.2 Shared Preferences

Třída "SharedPreferences" je další nástroj pro ukládání dat. Tento způsob uložení je založen na principu klíč a hodnota. Třída podporuje základní datové typy a to integer, long, float, string a boolean. Data zůstávají uložena i po zavření aplikace, pro jejich smazání musím původní data se stejným klíčem přemazat. Třidu "Shared Preferences" využívám pro uložení dat o přihlášeném uživateli a jeho nastavení aplikace [22].

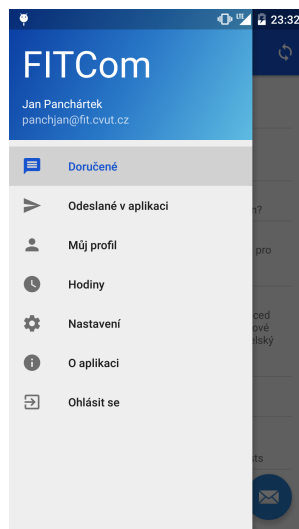
3.6 Uživatelské rozhraní a jeho prvky

3.6.1 Navigation Drawer

"Navigation Drawer" slouží jako jedna z možností, jak zobrazit uživateli menu v Android aplikaci. Prvek je zobrazen jako vyjíždějící panel z levé strany displeje. Uživatel může prvek "Navigation Drawer" použít dvěma způsoby. Buď kliknutím na tlačítko v levém horním rohu, a nebo pomocí gesta (přejetí prstem(y) po displeji) horizontálně od levé hrany displeje. Ukázku použití tohoto prvku můžeme vidět na obrázku 3.3 [18].

3.7 Knihovny pro práci s e-mailem

Pro práci s e-mailem aplikace využívá tři externí java knihovny, které zprostředkovávají odesílání a přijímání e-mailů: mail.jar, activation.jar a additional.jar. Tyto knihovny jsou "openSource" a byly do projektu přidány.



Obrázek 3.3: Navigation Drawer

3.8 Zprávy uživateli

Zprávy uživateli v aplikaci realizují pomocí třídy "SnackBar". Tato třída při vyvolání zobrazuje uživateli na displeji zprávy, které jsou ukázány v dolní části obrazovky v černém poloprůsvitném řádku. Tento řádek po zvoleném časovém intervalu opět zmizí.

Zde je ukázka konstrukce třídy "SnackBar":

```
SnackBar.make(this.findViewById(android.R.id.content), "zprava pro  
uživatele", SnackBar.LENGTH_LONG).setAction("Action",  
null).show();
```

3.9 Logování a debug

Při tvorbě programu je důležitá kontrola, co aplikace v který moment "dělá" a s jakými daty pracuje. Pro toto Android poskytuje třídu "Log", která má několik předdefinovaných metod. Jejich název označuje stupeň logování.

Pro vývoj aplikace jsem nejvíce používám metodu "d", což je označení pro metodu s úrovní debug. Metody mají vždy dva parametry. První z nich je tag a druhý samotná zpráva k vypsání, jak je vidět v ukázce níže.

```
int i = 10/2;  
Log.d("Debug", "to je zprava s hodnotou promenne " + i);
```

Testování

V této kapitole popíši způsoby testování aplikace. Moje aplikace byla podrobená dvojímu testování. První bylo provedeno na kompatibilitu na různých zařízeních. Druhé pak bylo testování s uživateli.

4.1 Testování vývojářem

Po celou část implementace byla aplikace průběžně testována za pomoci třídy "Log" 3.9. Pomocí kontrolních výpisů, jsem otestoval, každou nově naprogramovanou funkci. Testování bylo převážně prováděno pomocí zařízení One Plus X s verzí OS Android 5.1 a uhlopříčkou displeje 5 palců, popřípadě zpětná kompatibilita byla testována na zařízení Huawei G510 s verzí androidu 4.1.1 a uhlopříčkou displeje 4.3 palce.

4.2 Uživatelské testování

4.2.1 Testeři

V tabulce 4.1 jsou zobrazeny informace o jednotlivých uživateli, kteří testovali aplikaci a o zařízeních na kterých testování prováděli.

Tabulka 4.1: Testeři

Telefon	Velikost displeje	Verze OS Andorid	Role
Samsung S6	5.1"	6.0.1	Student
Nexus 5	1080*1920	5.0.0	Student
Asus Zenfone	5"	4.4.2	Student
Huawei Honor 7	5.2"	5.0.2	Student
Lenovo A820	4.5"	4.1.2	Vyučující
Samsung Galaxy S5	4.4"	4.4.2	Vyučující

4.2.2 Scénář

Testovací scénář obsahuje zadání úkolu a jeho předpokládané řešení.

1. Stáhněte si informace o hodinách.
Ideální řešení: Přejděte do fragmentu Nastavení, zde zadejte stahovat data o hodinách. Jestliže platnost autorizačního tokenu již vypršela, musíte se přihlásit.
2. Vyhledejte jméno(a) vyučujícího(ch), popřípadě studentů, v libovolné přednášce minulého semestru.
Ideální řešení: Přeprňte se do fragmentu Hodiny. Zde vyberte libovolnou hodinu označenou žlutým kolečkem (značícím přednášku) z minulého semestru. Po kliknutí se otevře detail hodiny. Zde jsou již data vidět.
3. Pošlete e-mail svému cvičícímu (kolegovi) na libovolný předmět v letošním semestru.
Ideální řešení: Přejděte do fragmentu Hodiny, zde vyberte libovolný předmět označený zelenou tečkou (značící cvičení) z letošního semestru. Zde kliknout na jméno vyučujícího. Vyplňte předmět a obsah e-mailu a odešlete ho.
4. Aktualizujte inbox složku
Ideální řešení: Přejděte do fragmentu Doručené a zde klikněte na tlačítko aktualizovat v pravém horním rohu.
5. Zjistěte svoje osobní číslo
Ideální řešení: Otevřete fragment Můj profil a zde najděte položku osobní číslo.
6. Najděte e-mail víc jak měsíc starý
Ideální řešení: Přejděte do fragmentu Doručené a zde skrolujte dolů až najdete požadovaný e-mail. Jestliže zde není, kliknutím načtete další e-maily a to opakujte dokud se vám nenačtete hledaný e-mail.
7. Odhlaste se
Ideální řešení: Otevřete menu tlačítkem v levém horním rohu, zde vyberte možnost Odhlásit se.

4.2.3 Výsledky testu

Aplikace byla podrobena testování s několika uživateli, kteří zkoušeli různé chování a funkčnost aplikace podle scénáře 4.2.2. Během testování byly nalezeny drobné chyby, které se povedlo snadno odstranit. Největší problém dělala testerům úvodní inicializace zařízení (nastavení hesla pro e-mailovou komunikaci, stažení e-mailů a dat o hodinách), která byla prováděna jen v nastavení

aplikace. Tento problém jsem se rozhodl vyřešit tak, že jsem možnost inicializace přidal i do obrazovek týkajících se jednotlivých funkcí a také napsal instalační a inicializační příručku, která je umístěna na příloženém CD.

Některé účastníky testu také zajímalo, jak je nakládáno s heslem pro e-mailovou komunikaci a jestli s ním aplikace nakládá bezpečně. Proto jsem se rozhodl přidat do aplikace informační text, který vysvětluje, že je vytvořen účet v samotném zařízení a heslo není uloženo přímo v aplikaci (podrobně popsáno v 3.1.2).

Uživatelé měli také další náměty na možné rozšíření aplikace. Nejčastěji zmiňovali synchronizace s poštovním serverem IMAP a s tím spojené upozornění neboli notifikace na nově příchozí e-mail. Další zajímavý nápad je rozšířit aplikaci o anglickou jazykovou verzi a tím usnadnit začátek studia na fakultě zahraničním studentům.

Možné rozšíření aplikace

Vzhledem k tomu, že aplikaci lze dále rozšiřovat, navrhnu v této kapitole možné doplnění dalších funkcí do aplikace. Veškeré zdrojové kódy a informace o projektu jsou nahrány na školním Gitlabu ve veřejném projektu, takže kdokoli z fakulty může na tuto práci dále navázat.

5.1 Synchronizace s IMAP

První funkcí, kterou by bylo vhodné do aplikace doimplementovat, je synchronizace školního e-mailového serveru s aplikací. K synchronizaci by docházelo u všech složek tj. Odeslané a Doručené. Díky tomu by se v aplikaci odstraněné e-maily automaticky mazaly i ze školního poštovního serveru.

Upozornění na nově přichozí e-mail, neboli notifikace, je s tímto problémem úzce spojena. Synchronizace by umožnila lepší práci s e-mailovým serverem, a proto by mohlo být doimplementováno do aplikace upozornění na nově přichozí e-maily neboli "push notifikace".

5.2 Koncepty

Při vytvoření nového e-mailu, který není uživatelem odeslán, by se tento koncept mohl ukládat do databáze a následně uživateli umožnit jeho opětovné vyvolání, zobrazení, dopsání a odeslání.

5.3 Našeptávání e-mailů

Funkce našeptávání usnadňuje výběr příjemce při psaní nové zprávy. Po napsání prvních tří písmen z křestního jména nebo příjmení nebo e-mailové adresy se zobrazí nabídka, ze které si uživatel klepnutím lehce vybere požadovaného adresáta.

V aplikaci by mohlo být našeptávání doplněno o možnost napovídání přes zkratku předmětu nebo celý název předmětu mimo výše zmíněných možností. Díky této rozšířené funkci o školní informace, ke kterým běžná aplikace pro spravování e-mailu nemá přístup, bude komunikace mezi studenty a vyučujícími rychlejší.

Vyučující by mohli pomocí efektivního našeptávání vybrat celou paralelku, a to tak, že napíše kód předmětu, popřípadě začátek názvu předmětu a zobrazí se jim jednotlivé paralelky s informací o dnu a času kdy probíhají.

5.4 Rozšíření vztahů odesílatele a příjemce

Doplněním dalších vztahů do aplikace by se zlepšila informovanost uživatelů. Již existující vztahy student/vyučující a vyučující/vyučující by byly doplněny o další: garant/vyučující, garant/student nebo student/student. Pro realizaci vztahů by musely být staženy další informace ze školního datového serveru; například garant předmětu, ostatní vyučující nebo všichni studenti, kteří mají tento předmět zapsán.

Dalšími užitečnými údaji, které by mohly pomoci v informovanosti uživatelů, jsou data o jednotlivých katedrách fakulty. Pomocí těchto dat by mohl nový student na fakultě zjistit důležitost e-mailu i bez hledání svého vztahu s odesílatelem, jelikož u každého nově příchozího e-mailu by se mu tento vztah automaticky zobrazil.

5.5 Posílání a přijímání příloh

Další funkcí, která by aplikaci rozšířila, je možnost připnutí přílohy k odesílanému e-mailu nebo přijetí a uložení přílohy z příchozího e-mailu do zařízení.

Rozšíření aplikace o tuto funkci by se týkalo většiny obrazovek, které realizují práci s e-maily tj.: Nový e-mail, Detail e-mailu, Doručené, Odeslané.

5.6 Kontakty

Poslední zmíněnou funkcí pro rozšíření aplikace je adresář kontaktů. Zde by uživatel mohl ukládat e-mailové adresy na vyučující nebo žáky, se kterými nemá žádnou vazbu. Do kontaktů by si mohl uložit i osoby, které nemají s fakultou nic společného, ale chce s nimi pomocí školního e-mailu komunikovat.

Do adresáře kontaktů by samozřejmě byly vloženy již stažené kontakty na vyučující, případně studenty, které aplikace získala při stahování dat o hodinách.

Realizace této funkce by se mohla projevit na dvou místech. První je přidání položky do vyjíždějího menu. Zde by uživatel mohl spravovat nebo zobrazovat své kontakty, přidávat nové a také upravovat stávající. Druhým místem, kde by aplikace byla změněna, je obrazovka psaní nového e-mailu,

kde by si uživatel mohl vyvolat seznam kontaktů a z něho si již vybrat příjemce.

Závěr

Cílem práce bylo vytvořit aplikaci pro operační systém Android, která bude podporovat komunikaci studentů a vyučujících na naší fakultě. Zároveň také bude pracovat se školními daty, jako jsou rozvrhy, e-mailové adresy a další data.

Před samotnou implementací jsem provedl počáteční analýzu, ve které jsem se zaměřil na aplikace pro jednotlivé operační systémy podporující e-mailovou komunikaci. Dále na aplikace vytvářené přímo pro naši fakultu a na možnosti řešení komunikace vyučujících a studentů na naší fakultě. Po této analýze jsem provedl návrh aplikace s hlavním zaměřením na dva aspekty. Tím prvním bylo to, jak bude samotná aplikace pracovat na pozadí a tím druhým uživatelské rozhraní. Při tvorbě tohoto rozhraní jsem uplatnil poznatky z analýzy a snažil se jej navrhnout tak, aby bylo jednoduché a přívětivé k jeho budoucím uživatelům. V další řadě jsem provedl implementaci samotné aplikace a průběžně ji sám testoval. Po dokončení práce jsem ji nechal vyzkoušet osloveným uživatelům. Nakonec, nad rámec zadání, bude práce umístěna na Google Play a poskytnuta studentům a vyučujícím volně ke stažení.

Díky této práci jsem se naučil vyvíjet mobilní aplikace pro operační systém Android a obohatil si znalosti při tvorbě analýzy a návrhu softwaru. Byla to pro mě do budoucna velmi užitečná zkušenost, vyzkoušet si realizaci celého projektu od zadání až po fungující aplikaci.

Programování mobilních aplikací na OS Android je rychle expandující obor IT a v budoucnu bych se mu rád věnoval.

Literatura

- [1] Michal Pavlíček: Android drží spolu s iOS více než 96% podíl mezi operačními systémy [online]. 30.1.2015, [Citováno 2015-03-15]. Dostupné z: <http://mobilenet.cz/clanky/android-spolu-s-ios-drzi-vice-nez-96-podil-mezi-operacnimi-systemy-18872>
- [2] Lucie Bílá: Největší statistika českých a slovenských freemailů. Kdo vládne e-mailové komunikaci? [online]. 1.7.2014, [Citováno 2015-03-24]. Dostupné z: <http://www.mail-komplet.cz/blog/nejvetsi-statistika-ceskych-a-slovenskych-freemailu>
- [3] Android Developers: Dashboards [online]. 4.4.2016, [Citováno 2016-04-17]. Dostupné z: <http://developer.android.com/about/dashboards/index.html>
- [4] Pět mailových klientů pro váš mobil [online]. [Citováno 2015-03-15]. Dostupné z: <http://www.aplikace.t-mobile.cz/clanky/pet-mailovych-klientu-pro-vas-mobil?nomobile=1>
- [5] Jan Dolejš: Aplikace Email.cz – jednoduchý a rychlý klient pro e-mail na Seznamu [online]. 25.6.2014, [Citováno 2015-03-15]. Dostupné z: <http://www.svetandroida.cz/email-cz-jednoduchy-rychly-klient-e-mail-seznamu-201406>
- [6] Android Developers: SDK Manager [online]. [Citováno 2016-04-17]. Dostupné z: <http://developer.android.com/tools/help/sdk-manager.html>
- [7] Vojtěch Semecký: Android Studio – nové vývojové prostředí [online]. 4.11.2013, [Citováno 2016-04-17]. Dostupné z: <https://www.zdrojak.cz/clanky/android-studio-nove-vyvojove-prostredi/>

- [8] Android Developers: Android Studio Overview [online]. [Citováno 2016-04-17]. Dostupné z: <http://developer.android.com/tools/studio/index.html>
- [9] David Čápka: UML - Doménový model [online]. [Citováno 2016-04-20]. Dostupné z: <http://www.itnetwork.cz/navrhove-vzory/uml/uml-domenovy-model-diagram>
- [10] Prof. RNDr. Jaroslav Pokorný, I. M. V. P., CSc.: *Databázové systémy*. Praha: Česká technika - nakladatelství ČVUT, 2013, ISBN 978-80-01-05212-9.
- [11] Matěj Konečný: Vytváříme pro Android: Dialogy a activity [online]. 17.8.2012, [Citováno 2016-04-23]. Dostupné z: <https://www.zdrojak.cz/clanky/vyvijime-pro-android-dialogy-a-activity/>
- [12] Android Developers: Managing the Activity Lifecycle [online]. [Citováno 2016-04-23]. Dostupné z: <http://developer.android.com/training/basics/activity-lifecycle/index.html>
- [13] Jakub Jirůtka: OAuth 2.0 [online]. 30.3.2016, [Citováno 2016-04-17]. Dostupné z: <https://rozvoj.fit.cvut.cz/Main/oauth2>
- [14] Android Developers: AccountManager [online]. [Citováno 2016-04-17]. Dostupné z: <http://developer.android.com/reference/android/accounts/AccountManager.html>
- [15] Kosapi [online]. [Citováno 2016-04-17]. Dostupné z: <https://kosapi.fit.cvut.cz/projects/kosapi/wiki>
- [16] Kosapi URL parametry [online]. [Citováno 2016-04-17]. Dostupné z: <https://kosapi.fit.cvut.cz/projects/kosapi/wiki/URLParameters>
- [17] Kosapi RESTful zdroje [online]. [Citováno 2016-04-17]. Dostupné z: <https://kosapi.fit.cvut.cz/projects/kosapi/wiki/Resources>
- [18] Allen, G.: *Android 4: Průvodce programováním mobilních aplikací*. Brno: Computer Press, 2013, ISBN 978-80-251-3782-6.
- [19] Android Developers: AsyncTask [online]. [Citováno 2016-04-17]. Dostupné z: <http://developer.android.com/reference/android/os/AsyncTask.html>
- [20] About SQLite [online]. [Citováno 2016-04-16]. Dostupné z: <https://www.sqlite.org/about.html>
- [21] Android Developers: Cursor [online]. [Citováno 2016-04-16]. Dostupné z: <http://developer.android.com/reference/android/database/Cursor.html>

- [22] Android Developers: SharedPreferences [online]. [Citováno 2016-04-17]. Dostupné z: <http://developer.android.com/reference/android/content/SharedPreferences.html>

Seznam použitých zkratk

- GUI** Graphical user interface
- XML** Extensible markup language
- UI** User interface
- OS** Operační systém
- ID** Identifikační číslo
- API** Application Programming Interface
- ČVUT** České vysoké učení technické
- FIT** Fakulta informačních technologií
- URI** Uniform Resource Location
- iOS** iPhone Operation System
- SDK** Software Developer Kit
- SQL** Structured Query Language
- IDE** Integrated Development Environment
- CTU** Czech Technical University

Obsah přiloženého CD

installationManual.txt	popis instalace a inicializace aplikace
readme.txt	stručný popis obsahu CD
apk	adresář s instalovatelnou formou aplikace
├─ FITCom.apk	instalační soubor aplikace
src		
├─ documentation	JavaDoc dokumentace
│ └─ index.html	spustitelný soubor
├─ impl	zdrojové kódy implementace
├─ thesis	zdrojová forma práce ve formátu \LaTeX
│ └─ thesis.tex	zdrojový soubor \LaTeX
│ └─ images	adresář s obrázky
│ └─ diagram	adresář s diagramy
text	text práce
└─ thesis.pdf	text práce ve formátu PDF