

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

MOBILNÍ CRM APLIKACE PRO OBCHODNÍ ZÁSTUPCE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MAREK NEVŘELA

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

MOBILNÍ CRM APLIKACE PRO OBCHODNÍ ZÁSTUPCE

MOBILE CRM APPLICATION FOR SALES REPRESENTATIVES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MAREK NEVŘELA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. RADEK KOČÍ, Ph.D.

BRNO 2015

Abstrakt

Bakalářská práce popisuje vývoj mobilní CRM aplikace pro obchodní zástupce. Cílem je vytvořit aplikaci, která umožňuje plánování událostí a online přístup k databázi, která obsahuje informace o zákaznících. Práce popisuje základní principy CRM, které byly použity při vývoji. Aplikace je vyvíjena pro mobilní telefony s operačním systémem Android, který je v této práci stručně představen. Součástí práce je analýza podobných, již existujících aplikací a analýza požadavků ze strany potenciálních uživatelů. V dalších částech je popsán návrh a postup implementace jednotlivých částí. V závěru je popsáno testování aplikace z několika různých hledisek a popsána zpětná vazba od uživatelů.

Abstract

Bachelor's thesis describes development of mobile CRM application for sales representatives. The aim is creating an application that helps with scheduling of events and online access to the database. The database contains information about customers. The thesis describes basic principles of CRM, which were used during development. Application is developed for mobile phones with operating system Android, which is briefly introduced in the thesis. The thesis consists of an analysis of similar, existing applications and analysis of requests from potential users. The next section portrays the design and implementation process of the parts. At the end, the thesis depicts testing from several different points of view and describes the feedback from users.

Klíčová slova

Android, CRM, řízení vztahů se zákazníky, mobilní aplikace, obchodní zástupce, online CRM

Keywords

Android, CRM, customer relationship management, mobile application, sales representative, online CRM

Citace

Marek Nevřela: Mobilní CRM aplikace pro obchodní zástupce, bakalářská práce, Brno, FIT VUT v Brně, 2015

Mobilní CRM aplikace pro obchodní zástupce

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Radka Kočího, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Marek Nevřela
20. května 2015

Poděkování

Rád bych poděkoval vedoucímu práce, Ing. Radku Kočímu, Ph.D., za cenné rady, připomínky a ochotu při tvorbě technické zprávy. Dále bych rád poděkoval Tomáši Bulawovi za odborné konzultace při vývoji aplikace.

© Marek Nevřela, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	3
2	CRM	5
2.1	Historie	5
2.2	Prvky CRM	6
2.3	Funkce CRM	6
2.4	Online systém CRM	7
3	Platforma Android	8
3.1	Historie a vývoj	8
3.2	Architektura systému	9
3.3	Základní části Android aplikace	11
3.4	Problematika vývoje pro OS Android	13
4	Analýza a návrh	15
4.1	Existující aplikace	15
4.2	Specifikace požadavků	16
4.3	Vývojové prostředí	18
4.4	Návrh databázové části	19
4.5	Návrh grafického uživatelského rozhraní	20
5	Implementace	23
5.1	Implementace částí aplikace	24
5.1.1	Denní přehled	24
5.1.2	Manažerská část	25
5.1.3	Kalendář	26
5.1.4	Adresář	27
5.1.5	Detailní pohled	28
5.1.6	Úkoly	28
5.1.7	Vytváření události a úkolu	29
5.2	Datová úložiště a komunikace	30
5.3	Speciální komponenty	30
5.3.1	FragmentManagerAdapter	30
5.3.2	Tabs	31
5.3.3	PIN	31
5.3.4	SlidingUpPanelLayout	32
5.3.5	DetailCards	33
5.4	AndroidManifest, služby, přijímače	33

5.5 Použité knihovny	34
6 Testování	36
6.1 Testování funkčnosti aplikace	36
6.2 Testování uživatelského rozhraní	37
7 Závěr	39
A Obsah CD	42
B Dotazník k testování uživatelského rozhraní	43

Kapitola 1

Úvod

Informační technologie jsou v dnešní době jednou z nejrozšířenějších technických odvětví na světě. Tyto technologie zasahují do všech odvětví průmyslu a v každém z nich mají svou nepopiratelnou hodnotu. Aktuálním trendem jsou mobilní zařízení, ať už se jedná o notebooky, mobilní telefony, tablety, či čtečky. Zejména mobilní telefony dalece přesáhly svůj původní záměr, jímž byla skutečnost umožňovat telefonní hovory bez závislosti na dostupnosti telefonního kabelu. V zaměstnání jsou mobilní telefony nejčastěji používány zejména na komunikaci, ale často také jako zdroj informací. V současné době se rozrůstá počet lidí, kteří využívají mobilní internet. Během roku 2014 stoupl počet jeho uživatelů téměř o polovinu [14]. Do budoucna se dá očekávat, že tento trend bude pokračovat a během nejbližších let bude mobilní internet používán téměř u všech uživatelů chytrých mobilních telefonů.

Jedním z odvětví, které využívají moderní informační technologie, je terciární sektor, který má v ekonomicky vyspělých zemích dominantní postavení v hospodářství, a je tak ukazatelem jejich vyspělosti. Tento sektor představuje veškeré služby včetně obchodu. Vzhledem k neustálému růstu konkurence v tomto sektoru je pro podniky nutné, aby si sami hledali zákazníky, udržovali s nimi dobré vztahy a tímto spravovali svůj marketing. Zejména na těchto bodech stojí zákaznický orientovaný management nazývaný CRM.¹ V dnešní době je již poměrně vyspělým systémem, který se orientuje na potřeby zákazníka. Mnoho firem tento systém již používá, většinou v podobě desktopových aplikací. V posledních letech je v informačních technologiích kladen důraz na mobilitu a neustálou aktuálnost dat a také proto tyto aplikace expandují na mobilní zařízení.

Cílem této práce je vytvořit mobilní aplikaci ve spolupráci s firmou Nextis s.r.o., která bude online nástrojem obchodních zástupců. Tito lidé využívají již existující desktopovou aplikaci Nextis Project od zmíněné firmy. Aplikace bude zaměřena na plánování a řízení změn událostí. Obchodní zástupci budou mít skrze aplikaci neustálý přístup k aktuálním informacím o svých zákaznících, neboť tato data budou zpřístupněna skrze síť internetu. Cílovou platformou je operační systém Android, který je momentálně nejrozšířenější mobilním operačním systémem vůbec.

Druhá kapitola je zaměřena na CRM. Zmíněny zde budou základní prvky této strategie, její cíle a klíčové části online CRM systémů. Další kapitola obsahuje informace o operačním systému Android a problematice vývoje pro tuto platformu. Čtvrtá kapitola popisuje již existující aplikace určené k plánování a správě událostí, jejich klíčové funkce a zhodnocení vzhledem k požadavkům této práce. Tato část nabízí také detailní specifikaci požadavků a

¹Customer Relationship Managment - Řízení vztahů se zákazníky

návrh databázové části a grafického rozhraní. V páté kapitole jsou zmíněny implementační detaily jednotlivých částí aplikace, datová komunikace a knihovny, které byly použity. Šestá kapitola pojednává o testování aplikace na různých zařízeních, testech uživatelů a jejich zpětné vazbě. V poslední kapitole se nachází závěrečné shrnutí a možnosti rozšíření aplikace.

Kapitola 2

CRM

Customer Relationship Management (CRM), nebo-li řízení vztahů se zákazníkem, je způsob jakým identifikovat, získat a udržet si zákazníky, kteří jsou největším obchodním aktivem společnosti [15]. Tato definice popisuje CRM jako obchodní strategii. CRM však nemusí být chápána jako pouhá strategie. V mnoha definicích nabývá i širšího významu než jen způsobu vedení společnosti. Řízení vztahu se zákazníky zahrnuje pracovníky, podnikové procesy a technologii IS/ICT s cílem maximalizovat loajalitu zákazníků a v důsledku toho i ziskovost podniku. Technologicky stále více využívá potenciál a možností internetu [10]. Vztahy, které vznikají mezi zákazníky a firmami, jsou prospěšné pro obě strany, vzniká tak situace dvou vítězů.

2.1 Historie

Počátky CRM jako marketingové strategie či filosofie sahají do 80. - 90. let 20. století.

Už v 80. letech 20. století začínalo být nutností využití databázových systémů v obchodování, což směřovalo k vytváření oddělení péče o zákazníky. U klíčových zákazníků šlo o důležitý nástroj, který zajišťoval komunikaci mezi firmou a zákazníkem a umožňoval vyrábět řešení na míru. V případě menších zákazníků byl přínos minimální, protože informace o velkém množství zákazníků hlavně zatěžovali databáze a profit z těchto zákazníků nebyl tak velký. Postupem času byly zjištěny klíčové informace, které postačily ve většině případů: zaměření klienta, zboží, které pravidelně kupuje a jeho útrata.

V 90. letech se u CRM vyvinulo obousměrné využití. Oproti původnímu sbírání informací začaly společnosti své zákazníky odměňovat nejen v podobě lepší péče, ale i skrze dárky a další výhody upevňujících jejich vztah. To byl začátek v dnešní době známých věrnostních programů. V roce 1995 se definitivně ustálilo názvosloví a ujalo se označení CRM po předchozím sjednocení CIM¹ a CIS² [18].

Na začátku 21. století se strhl konkurenční boj mezi poskytovateli CRM systémů, a to vyústilo ke snížení cen. Důsledkem této akce bylo rozšíření těchto systémů i do menších společností. IT firmy vyvíjely novější a pokročilejší řešení, které umožňovaly přizpůsobení v závislosti na typu podniku, kde mají být používány, což umožnilo lépe využívat získané informace.

Velkým přínosem k vývoji CRM systémů bylo umožnění využívat vzdálená úložiště dat. Internet nabídl tak novou možnost přístupu k těmto datům a výrobci se začali zaměřovat

¹CIM - Customer Information Management - správa informací o zákaznících

²CIS - Customer Information System - systém informací o zákaznících

na internetová řešení. Tímto byl umožněn vývoj nových obchodních strategií a lepší sdílení informací o zákazníkovi. Všechny tyto aspekty směřovali ke zvýšení spokojenosti zákazníka [20].

2.2 Prvky CRM

Řízení vztahů se zákazníky znamená interakci se zákazníkem. Tuto interakci musí zajistit čtyři na sobě závislé prvky CRM [22]:

- **Lidé** - zaměstnanci společnosti. Schopnosti a znalosti těchto lidí jsou klíčem k uspokojení potřeb zákazníka. Nekvalifikovaní zaměstnanci mohou škodit nejen zákazníkovi, ale v konečném důsledku zejména společnosti. Proto je důležité získat pouze kvalifikované zaměstnance a dále je školit a vzdělávat.
- **Procesy** - komunikace a vyřizování požadavků zákazníků. Tyto procesy musí mít jasný postup, jak vyřídit požadavky zákazníka v nejkratším možném termínu.
- **Data** - informace o zákaznících. Aby byla činnost společnosti úspěšná, je potřeba vědět kdy a jaké přání má zákazník a v pravý okamžik mu je splnit. Získaná data musí být neustále aktualizována a musí být dostupná odpovědnému zaměstnanci, který je ve styku se zákazníkem, aby mohla být použita ve vhodný okamžik.
- **Technologie** - informační technologie, které usnadňují řízení vztahů s větším množstvím zákazníků. Jedná se o softwarové produkty, umožňující automatizaci procesů a analýzu, databázové systémy, bezpečnostní prvky a aplikace, které spojují všechny tyto prvky.

Samotné zavedení CRM ve společnosti je nejnáročnější fáze. Jedná se o zprovoznění koloběhu činností, které spolu úzce souvisí. Jednotlivé prvky CRM musí být vzájemně propojeny, aby byl zisk společnosti při jednání se zákazníkem maximální.

2.3 Funkce CRM

Klíčovým úkolem CRM je podpora obchodních procesů a vytváření dlouhodobých a úspěšných vztahů se zákazníky a zvyšování jejich loajality. Mezi hlavní funkce CRM patří [4]:

- Průběžné sledování zákaznických požadavků a chování, evidence a hodnocení současných obchodních kontaktů.
- Vytvoření nových obchodních příležitostí s využitím zmíněných zákaznických informací.
- Aktivitty vedoucí k vytváření dlouhodobých a ekonomicky hodnotných vztahů se zákazníky.
- Náročné analýzy zákazníků podle nejrůznějších hledisek.
- Řízení marketingových kampaní s využitím výsledků zákaznických analýz a jejich požadavků.

2.4 Online systém CRM

Důležitou částí pro využití potenciálu CRM strategie je automatizace prodejních činností. Tato automatizace není zaměřena na prodej výrobků nebo zaměření na produktivitu, ale pomáhá při vytváření informací o zákaznících. Online systém by měl umožňovat všem obchodníkům, případně externím obchodním partnerům, rychlý a efektivní přístup k důležitým informacím o zákaznících, plánování pracovního programu, využití znalostníchází a dalších prodejních nástrojů zahrnutých v systému. Hlavními funkcemi takového online systému jsou [4]:

- Správa kontaktů a plánovač času.
- Správa nových, dosud neidentifikovaných kontaktů - zahrnuje distribuci nových kontaktů zodpovědným obchodníkům a zároveň sledování a zpětnou vazbu další komunikace s potenciálními klienty.
- Nástroje tvorby souhrnných zpráv pro management - umožňují manažerům udržet si trvalý přehled o zákaznících ve zhuštěné a efektivní podobě.
- Nástroje vzdáleného přístupu k systému - zajištění rychlého a efektivního přístupu k systému mimo kancelář.
- Správa obchodních partnerů - zajištění efektivní spolupráce s distributory a dalšími obchodními partnery pracujícími s firemními produkty.

Kapitola 3

Platforma Android

Android je open source platforma určená především pro mobilní zařízení jako jsou chytré telefony, tablety nebo nositelná zařízení (hodinky, náramky). V roce 2014 bylo představeno rozšíření tohoto systému také pro televize a auta [17].

Tato platforma zahrnuje operační systém založený na linuxovém jádře, uživatelské rozhraní, základní aplikace a middleware.¹

3.1 Historie a vývoj

Společnost Android Inc. byla založena roku 2003 v Kalifornii. V roce 2005 byla společnost odkoupena firmou Google Inc. a ta z ní vytvořila svou dceřinou společnost. Klíčové pro tento obchod bylo setrvání hlavních zakladatelů Android Inc.

Tým vedený Andy Rubinem, jedním ze zakladatelů, vyvinul platformu pro mobilní zařízení, založenou na linuxovém jádře, a v roce 2007 získal několik patentů v oblasti mobilních technologií. Na začátku listopadu téhož roku vzniklo sdružení firem *Open Handset Alliance* (OHA) z oblastí mobilních operátorů, softwarových společností a výrobců mobilních telefonů. Tato firma při oznámení svého vzniku představila Android jako svého partnera a zasloužila se o jeho další vývoj [21]. K dnešnímu dni toto sdružení zahrnuje 84 firem (původně 34) [3].

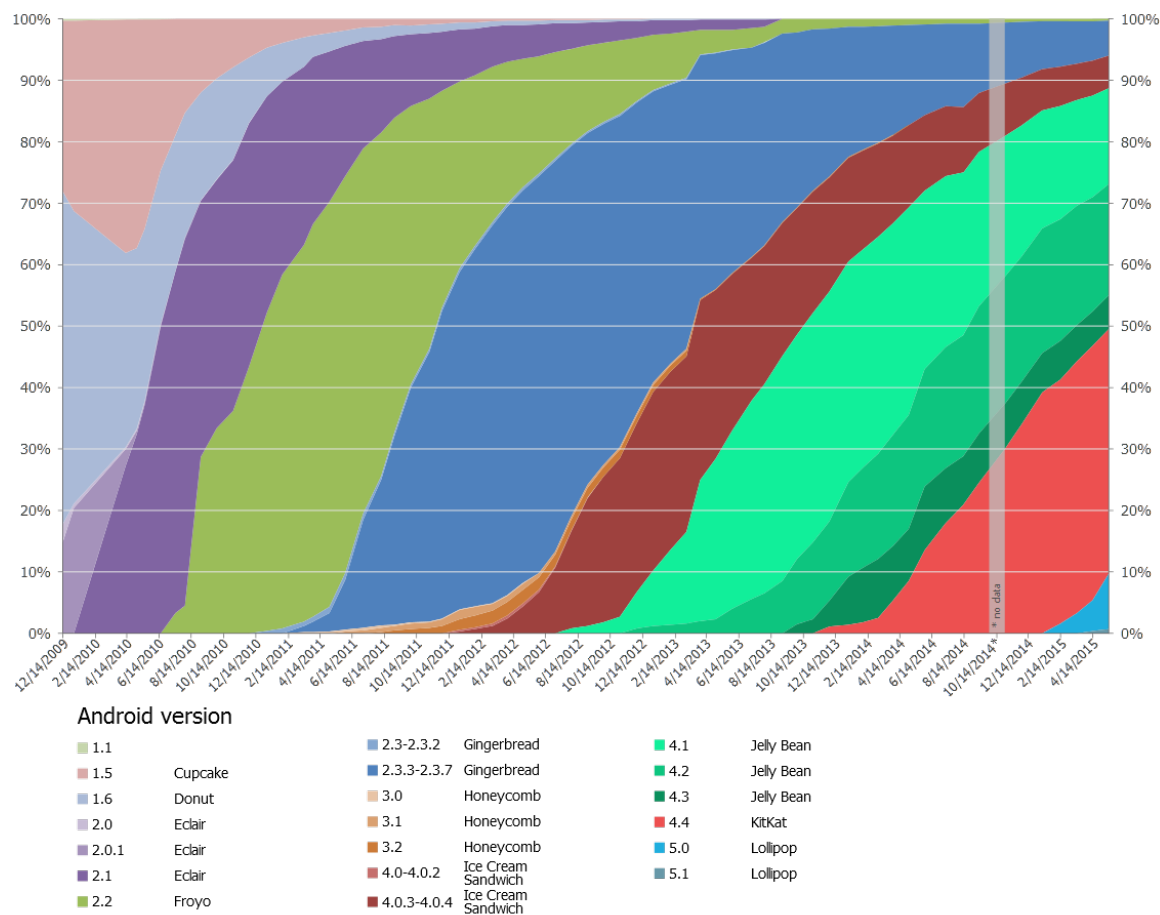
V září 2008 byl představen první mobilní telefon s operačním systémem Android - HTC Dream. Na tomto zařízení byla nainstalována nepojmenovaná verze Android 1.0. Od verze Android 1.5 Cupcake jsou pojmenovávány všechny verze v abecedním pořadí podle sladkostí. Každá aktualizace systému přináší nové API,² které většinou rozšiřuje předešlou verzi o nové komponenty či funkcionality.

První tablet s operačním systémem Android byl představen v roce 2010. Jednalo se o Android 2.2 Froyo. Další rok Android představil svou novou verzi systému Android 3.0 Honeycomb. Tato verze byla určena pouze pro tablety, avšak zařízení s těmito verzemi začalo velmi rychle ubývat potom, co byla vypuštěna verze následující. Po tabletech Android expandoval také na hodinky a v roce 2013 představil Samsung své první chytré hodinky s operačním systémem Android 4.3 Jelly Bean. Poslední novinky představil Google na své konferenci Google I/O v červnu 2014, kdy představil svou poslední velkou aktualizaci na verzi Android 5.0 Lollipop. Tato verze představuje použití nejen pro chytré telefony, tablety a hodinky, ale také pro televize a auta. Vývoj procentuálního zastoupení systémových verzí

¹middleware - softwarová vrstva ležící mezi operačním systémem a aplikacemi

²API - Application Programming Interface - aplikační programové rozhraní

můžete vidět na obrázku 3.1. Během této konference Google oznámil, že zařízení s operačním systémem Android již používá více než 1 miliarda uživatelů, což je oproti předchozímu roku téměř dvojnásobek [13].



Obrázek 3.1: Zastoupení verzí OS Android od konce roku 2009 do současnosti³

3.2 Architektura systému

Architektura operačního systému je složena z pěti vrstev. Tyto vrstvy jsou zobrazeny na obrázku 3.2.

Linuxové jádro

Nejnižší vrstva založená na upraveném linuxovém jádru. Vytváří abstraktní vrstvu mezi hardwarem a softwarem. První verze Androidu používaly verzi jádra 2.6, dnes se používají verze novější. Jádro zajišťuje správu paměti, procesů a správu sítě.

³Převzato z http://en.wikipedia.org/wiki/Android_version_history

⁴Převzato z <http://developer.android.com/images/system-architecture.jpg>



Obrázek 3.2: Vrstvy architektury systému Android⁴

Knihovny

Tato vrstva poskytuje skupinu knihoven. Android nabízí své základní API, které je k dispozici pro všechna zařízení. Kromě Android API zde existuje skupina knihoven, psaných v jazyce C/C++, které jsou využívány různými komponenty systému.

Běhové prostředí

Tato vrstva obsahuje virtuální stroj Dalvik a základní Java knihovny pro běh aplikací. Tento virtuální stroj nahrazuje JVM⁵ známý z PC, který slouží pro překlad programu do spustitelné podoby. Jeden z hlavních důvodů vytvoření virtuálního stroje Dalvik byla skutečnost, že knihovny Androidu jsou licencovány jako open source, avšak JVM volně šiřitelný není. Dalším důvodem byla potřeba optimalizovat výkon a spotřebu energie.

Aplikace pro Android jsou programovány v jazyce Java. Při překladu dochází k vytvoření Java byte kódu a poté dochází ke kompilaci pomocí Dalvik kompilátoru na Dalvik byte kód. Tento byte kód je při spuštění aplikace přeložen do nativního strojového kódu zařízení na virtuálním stroji Dalvik. Po dokončení překladu je aplikace spuštěna. Všechny spuštěné Android aplikace běží ve svém vlastním procesu. Od verze Androidu 4.4 se používá dopředná kompilace kódu, kdy překlad z Dalvik byte kódu na nativní strojový kód probíhá již při instalaci aplikace do zařízení. Díky tomuto přístupu je spouštění aplikací rychlejší, procesor je při spouštění aplikací méně vytěžován a důsledkem je i menší spotřeba energie [11].

⁵JVM - Java Virtual Machine

Aplikační framework

Nejdůležitější vrstva pro vývoj aplikací, která zprostředkovává přístup ke službám využívaných v aplikacích a hardwaru zařízení. Mezi základní služby aplikačního rámce patří:

- **View System** - sada prvků, které jsou používány k sestavení uživatelského rozhraní. Součástí této sady jsou tlačítka, textová pole, zaškrtačací políčka a jiné.
- **Notification Manager** - zajišťuje zobrazení notifikací ve stavovém řádku
- **Content Providers** - umožňuje používat data jiných aplikací. Například takto můžeme přistupovat ke kontaktům.
- **Activity Manager** - řídí životní cyklus aktivit aplikací
- **Resource Manager** - umožňuje přístup k „nekódovým“ zdrojům aplikace, např. obrázky, řetězce.

Aplikace

Nejvyšší vrstva architektury, která je tvořena aplikacemi uživatele. Základní aplikace jsou již předinstalovány, jako např. internetový prohlížeč, telefonní seznam atd. Další aplikace lze stáhnout přes obchod Google Play.

3.3 Základní části Android aplikace

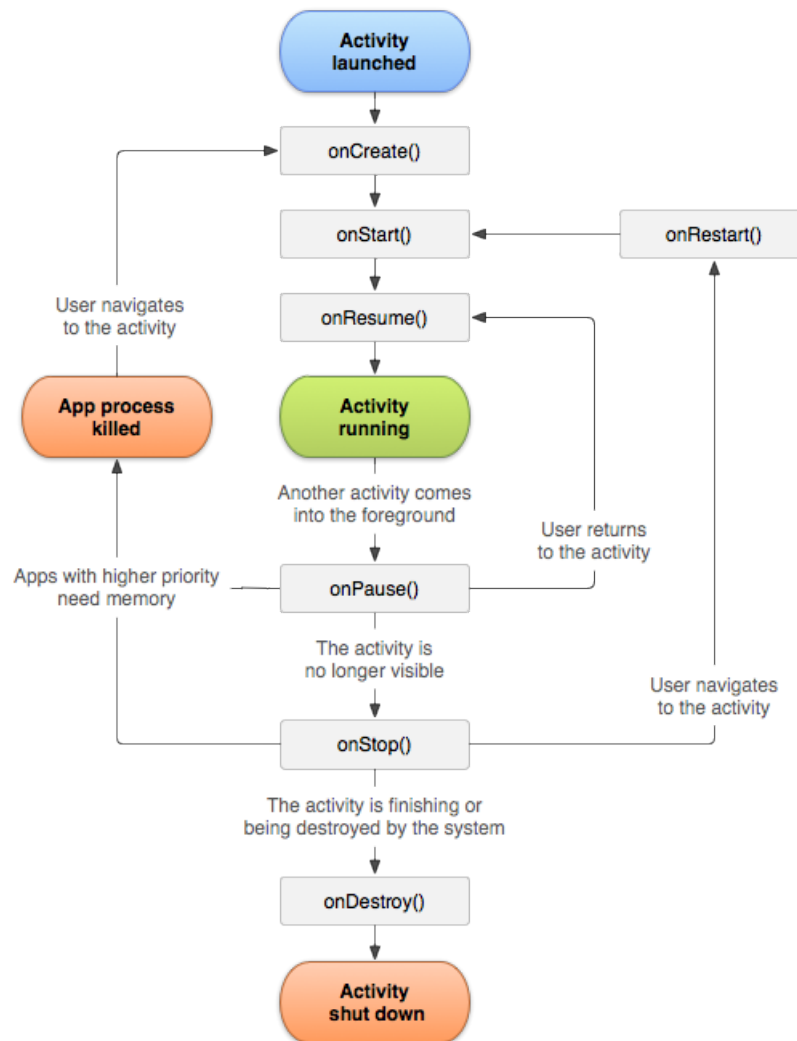
Aplikační komponenty jsou základními stavebními kameny aplikací. Každá komponenta je vstupní bod, skrze který lze do aplikace vstoupit. Jednotlivé komponenty mohou mezi sebou komunikovat pomocí zpráv (instance třídy **Intent**). Existují celkem čtyři různé typy aplikačních komponent [8]:

Aktivita

Aktivita je komponenta, která reprezentuje jednu obrazovku aplikace. Součástí aktivity je grafické uživatelské rozhraní, pomocí kterého aplikace interaguje s uživatelem. Aplikace je většinou složena z více aktivit. Každá z aktivit má svůj vlastní životní cyklus. Životní cyklus aktivity je znázorněn na obrázku 3.3. Aktivita se může v jeden moment nacházet v jednom z těchto stavů:

- **Běžící** - aktivita je spuštěna, běží na popředí a je možná plná interakce uživatele s aplikací.
- **Pozastavená** - aktivita je spuštěná a viditelná, ale na popředí je zobrazena jiná. Většinou se jedná o upozornění, např. příchozí hovor.
- **Zastavená** - aktivita byla překryta jinou aktivitou, není viditelná a běží na pozadí. Je stále uložena v paměti, ale při nedostatku paměti systém může aktivitu ukončit.
- **Ukončená** - aktivita byla ukončena a její obsah byl uvolněn z paměti.

⁶Převzato z http://developer.android.com/images/activity_lifecycle.png



Obrázek 3.3: Životní cyklus aktivity⁶

Služby

Služba neposkytuje uživatelské rozhraní, zajišťuje běh dlouhotrvajících operací na pozadí a je nezávislá na aktivitách. Každá služba běží jako samostatný proces a má vlastní životní cyklus.

Poskytovatel obsahu

Poskytovatel obsahu je aplikační rozhraní pro sdílení dat mezi aplikacemi. Zajišťuje přístup k datům a jejich zabezpečení.

Přijímače

Přijímač je komponenta, která slouží k naslouchání zpráv zaslaných systémem nebo jinými aplikacemi. Poté na ně může dle jejich obsahu reagovat, například vytvořením notifikace nebo spuštěním jiné komponenty. Stejně jako služba neposkytuje uživatelské rozhraní.

3.4 Problematika vývoje pro OS Android

Android je operační systém s poměrně rychlým vývojem. Každý rok je vydána velká aktualizace systému. Během roku jsou potom obvyklé menší aktualizace, které opravují drobné chyby. Tyto aktualizace přinášejí pozitiva i negativa. Většina chyb z předchozích verzí je odstraněna, je přidána nová funkcionality a často také i nové vizuální prvky systému. Některé starší prvky mohou být označeny jako zastaralé a nejsou doporučeny používat.

Poslední velká aktualizace systému na verzi Android 5.0 Lollipop proběhla v listopadu 2014 a přinesla nejzásadnější změny za několik posledních let. Byl vytvořen nový styl uživatelského rozhraní *Material Design* [2]. Obecně každý větší update systému přináší nové ovládací prvky a rozšiřování rozhraní starších tříd. Tyto prvky a nová funkcionality je však použitelná pouze v nových verzích. Tímto nastává problém s vývojem aplikací pro zařízení, která aktualizaci nedostanou. Do této skupiny většinou patří starší zařízení nebo ta, která nesplňují hardwarové požadavky. Vývojáři aplikací musí vyvíjet software, který je kompatibilní s novými i staršími systémy. Částečným řešením je použití zpětně kompatibilních knihoven, které poskytuje oficiálně Google. Tyto knihovny obsahují některé nové prvky, které lze použít i pro starší verze Androidu.

Časté problémy nastávají také při vytváření vzhledu jednotlivých obrazovek aplikace nebo jejich částí. Při zobrazení těchto obrazovek hrají roli tyto faktory:

- Velikost obrazovky
- Hustota zobrazovacích bodů obrazovky
- Orientace obrazovky

Tyto hodnoty mohou být pro různá zařízení velmi rozdílné, protože se týkají jak nejmenších chytrých telefonů (velikost 3 palce), tak největších tabletů (velikost 10 palců). Zařízení lze rozdělit do několika skupin podle hustoty zobrazovacích bodů a velikosti obrazovky podle tabulky 3.1.

velikost	hustota	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Součet
malé		4.1 %						4.1 %
normální			7.6 %	0.1 %	39.5 %	19.8 %	16.2 %	83.2 %
velké		0.4 %	4.8 %	2.2 %	0.6 %	0.6 %		8.6 %
extra velké			3.2 %		0.3 %	0.6 %		4.1 %
Součet		4.5 %	15.6 %	2.3 %	40.4 %	21.0 %	16.2 %	

Tabulka 3.1: Tabulka zastoupení zařízení podle hustoty zobrazovacích bodů a velikosti obrazovky [5]

Operační systém Android podporuje vykreslování vizuálních prvků a jejich rozmístění pro více typů displejů. Někdy se však může stát, že prvky nejsou rozmístěny přesně dle představ vývojáře nebo nejsou požadované velikosti. Pokud tedy vývojář chce zajistit, že obrazovky budou vypadat přesně dle návrhu na každém typu displeje, pak musí dodržet následující body [9]:

- Explicitně určit v souboru `AndroidManifest`, které velikosti obrazovek jsou podporovány

- Vytvořit různé rozložení obrazovek pro různé velikosti obrazovek
- Poskytnout rozdílné velikosti bitmap pro obrazovky s rozdílnou hustotou zobrazovacích bodů

Kapitola 4

Analýza a návrh

V této kapitole jsou popsány vybrané existující aplikace pro správu a plánování událostí a úkolů. Zaměřena bude zejména na funkcionalitu, kterou jednotlivé aplikace poskytují. Dále jsou specifikovány požadavky na funkčnost aplikace a popsán návrh databázové části, grafického rozhraní a rozložení jednotlivých logických částí.

4.1 Existující aplikace

V této kapitole budou popsány vybrané existující aplikace pro správu a plánování událostí a úkolů. Zaměřena bude zejména na funkcionalitu, kterou jednotlivé aplikace poskytují.

Jako výchozí zdroj aplikací je pro zařízení s OS Android služba Google Play.¹ Tato služba byla využita při hledání aplikací s výše popsaným zaměřením. Těchto aplikací je k dispozici na službě Google Play poměrně dost. Proto bylo vybráno několik aplikací s největším počtem stažení a nejlepším hodnocením. Po prozkoumání aplikací byly aplikace rozděleny do dvou kategorií. První kategorie tvoří aplikace typu kalendář, druhou poznámkové bloky a „úkolníčky“.

Kalendáře

Zástupci této kategorie jsou aplikace: Google Kalendář, aCalendar, Sunrise Calendar a DigiCal.

Tyto kalendáře používají několik módů zobrazení - agenda dne, týdenní a měsíční mód. Tyto rozdílné módy jsou důležité pro zobrazení událostí podle situace. Pokud chce uživatel vědět, v jakém pořadí má naplánovány schůzky, zvolí denní pohled. Pro přehled obsazenosti jednotlivých dnů je možné zvolit jeden ze zbývajících dvou módů. Výjimku tvoří aplikace Sunrise Calendar, která má dva módy - smíšený týdenní / měsíční s denní agendou a podrobný třídní. Důležité pro práci s kalendářem je obrazovka vytváření událostí. Všechny tyto aplikace nabízejí sdílení událostí pro více lidí, možnost upozornění, opakování událostí, barevné rozlišení nebo podpora více kalendářů. Avšak nejdůležitějšími položkami u každé události je název, začátek, případně konec schůzky a poznámka.

Tyto aplikace nabízejí funkčně téměř stejné možnosti. Liší se většinou jen grafickým zpracováním jednotlivých částí. Důraz se klade především na přehlednost aplikace a jednoduchost ovládání.

¹Google Play - <https://play.google.com/>

Poznámkové bloky a „úkolníčky“

Zástupci této kategorie jsou aplikace: Evernote, OmniNotes, Google Keep a WunderList.

Všechny tyto aplikace nabízí prostor ke správě poznámek a přístupu přes widgety.² Tyto aplikace také podporují zaškrtačací seznamy, plánování notifikací na určitou dobu nebo sdílení s dalšími osobami. Mezi zajímavé funkce patří například přidávání multimediálních příloh k vybrané poznámce nebo označení poznámek štítky či řazení poznámek do skupin. Avšak oproti kalendářům nemají agendu, ani kalendářové náhledy.

Tyto aplikace jsou opět velmi podobné a liší se pouze ve specifických funkcích. Např. Evernote poskytuje pro placený účet Premium bezpečnostní PIN obrazovku, Google Keep je specifický svým zobrazením barevných poznámek, připomínající nástěnku.

4.2 Specifikace požadavků

Cílem této práce je vytvořit mobilní aplikaci, která bude sloužit jako nástroj zejména pro obchodní zástupce firem, ale také pro jejich manažery. Specifikace vycházejí z principů popsaných v kapitole 2.4 a konzultací s potenciálními uživateli aplikace. Aplikace bude rozdělena do dvou hlavních částí - uživatelské a manažerské. Případy užití pro uživatele a manažera jsou uvedeny v diagramu 4.1.

Uživatelská část

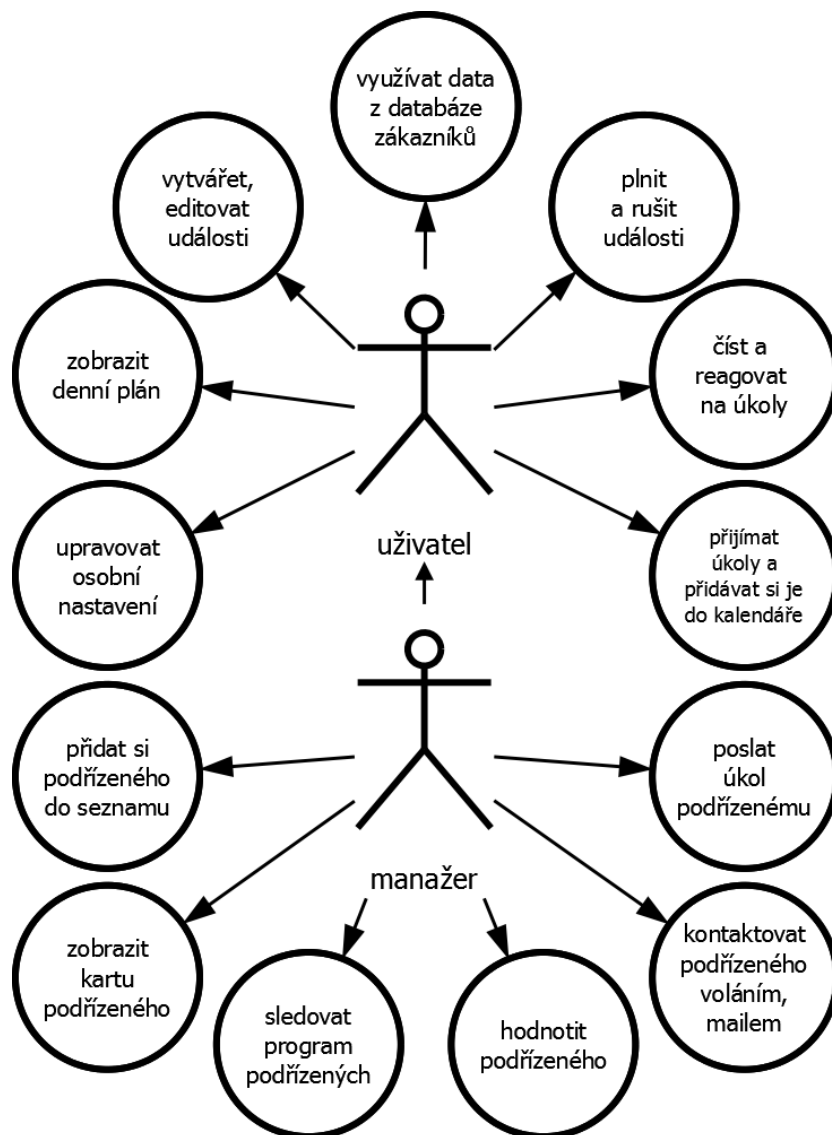
Do této části budou mít přístup všichni uživatelé, kteří aplikaci budou používat, včetně manažerů.

1. Denní přehled - tato část zobrazuje přehled událostí uživatele pro aktuální den. Jednotlivé položky lze postupně plnit a odškrtačovat jako položky seznamu. Každá z událostí je specifikována svým typem.
2. Kalendář - část poskytující přehled naplánovaných událostí uživatele. Nabízí procházení kalendářem ve více pohledech, nejdůležitější jsou však pro danou skupinu uživatelů nejbližší dny, maximálně týdny. V této části lze také vytvářet nové události.
3. Adresář - v této části lze vyhledávat mezi partnery. V detailním pohledu vybraného partnera jsou k dispozici údaje o nákupech, prodeji a spolupráci s firmou. Odpovídající data jsou pro lepší přehlednost reprezentována grafy. Detail bude také obsahovat výpis kontaktů, CRM modulu a událostí spojených s partnerem.
4. Úkoly - přehled úkolů zadaných nadřízeným uživatelem. V této části lze úkol přijmout nebo odmítnout. V kladném případě se tento úkol automaticky zařadí do kalendáře uživatele.
5. Nastavení - sekce poskytuje možnost upravovat osobní nastavení k právě přihlášenému účtu.

²widget - ovládací prvek aplikace, který lze umístit na základní obrazovku zařízení a přistupovat tak přímo k jeho datům a funkcím, aniž by byla aplikace spuštěna

Manažerská část

Tato část je přístupná pouze pro uživatele specifikované vyššími přístupovými právy (manažery). V této sekci bude přehled jednotlivých uživatelů, které má daná osoba na starosti. V této části lze také prohlížet profilové části těchto uživatelů, jejich plány, hodnotit jejich události a vytvářet jim úkoly.



Obrázek 4.1: Diagram případu užití

K této aplikaci je navržena externí databáze, kterou bude aplikace využívat k získání stále aktuálních dat. K získání těchto dat a práci s aplikací bude potřeba trvalý přístup k internetu.

4.3 Vývojové prostředí

Aplikace pro Android jsou ve většině případů vyvíjeny v programovacím jazyce Java. Proto je nezbytné mít nainstalovaný Java Development Kit (JDK),³ který poskytuje základní nástroje pro vývoj aplikací v programovacím jazyce Java. Pro vývoj aplikací je vhodné použít vývojové prostředí (IDE),⁴ které by mělo urychlit práci. Mezi dvě nejpoužívanější prostředí patří Android Studio⁵ a Eclipse.⁶

K využívání Eclipse IDE pro vývoj aplikací pro OS Android je potřeba samostatně nainstalovat Android Development Tool (ADT) plugin. Další nutností je mít nainstalovaný Android Software Development Kit (SDK),⁷ který obsahuje nástroje pro vývoj aplikací pro platformu Android.

Při instalaci Android Studia musíte mít nainstalovaný jen JDK, další nástroje jako Android SDK, kompilátor Android nebo základní emulátory s plnohodnotným systémem jsou součástí instalačního balíčku.

Android Studio

Android Studio jsem zvolil nejen z důvodu jednodušší instalace, ale i díky tomu, že jsem měl v minulosti možnost vyzkoušet práci s oběma vývojovými prostředími.

Android Studio je poměrně nové vývojové prostředí, které bylo představeno v květnu roku 2013. Je založené na IntelliJ IDEA, což je vývojové prostředí, zaměřené zejména na programování v jazyce Java. Tento software získal několik ocenění a je uznávaným po celém světě [12]. V následujících sekcích jsou shrnuty výhody a novinky Android Studia, které přináší oproti Eclipse.

Gradle

Android Studio spojil svůj další vývoj s nástrojem Gradle, což je automatizační nástroj, který staví na návrhu Apache Ant a Apache Maven [1]. Gradle byl navržen pro překlad velkých projektů, sestavených z více menších částí. Narozdíl od svých předchůdců používá orientovaný acyklický graf k určení pořadí, ve kterém budou jednotlivé úlohy spuštěny. Tento nástroj tak řeší problémy, které nastávaly u Apache Ant při sestavování větších projektů [16]. Umožňuje také velmi jednoduché nalinkování knihoven z různých webových repozitářů, což v Eclipse je možné jen se speciálními pluginy.

GUI designer

Pro návrh designu jednotlivých rozložení(layoutů) stránek je zde připraven speciální návrhář, který předčil svého předchůdce v Eclipse. Stejně můžete tvořit návrh v XML nebo v Design módu, avšak Android Studio nabízí navíc hybridní pohled. Ten využijete při psaní XML, kde se Vám v reálném čase zobrazuje Váš návrh. Speciální funkcí je zobrazení obrazovky na více zařízeních najednou s různou velikostí displeje a různým rozlišením [19].

Kromě těchto předností má Android Studio spoustu dalších drobných předností, jako zobrazení barev a ikon v bočním panelu odpovídajícího řádku nebo velmi rychlý a chytrý našeptávač. Celkově působí rychleji než Eclipse a má menší paměťové nároky.

³Dostupné z <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

⁴IDE - Integrated Development Environment

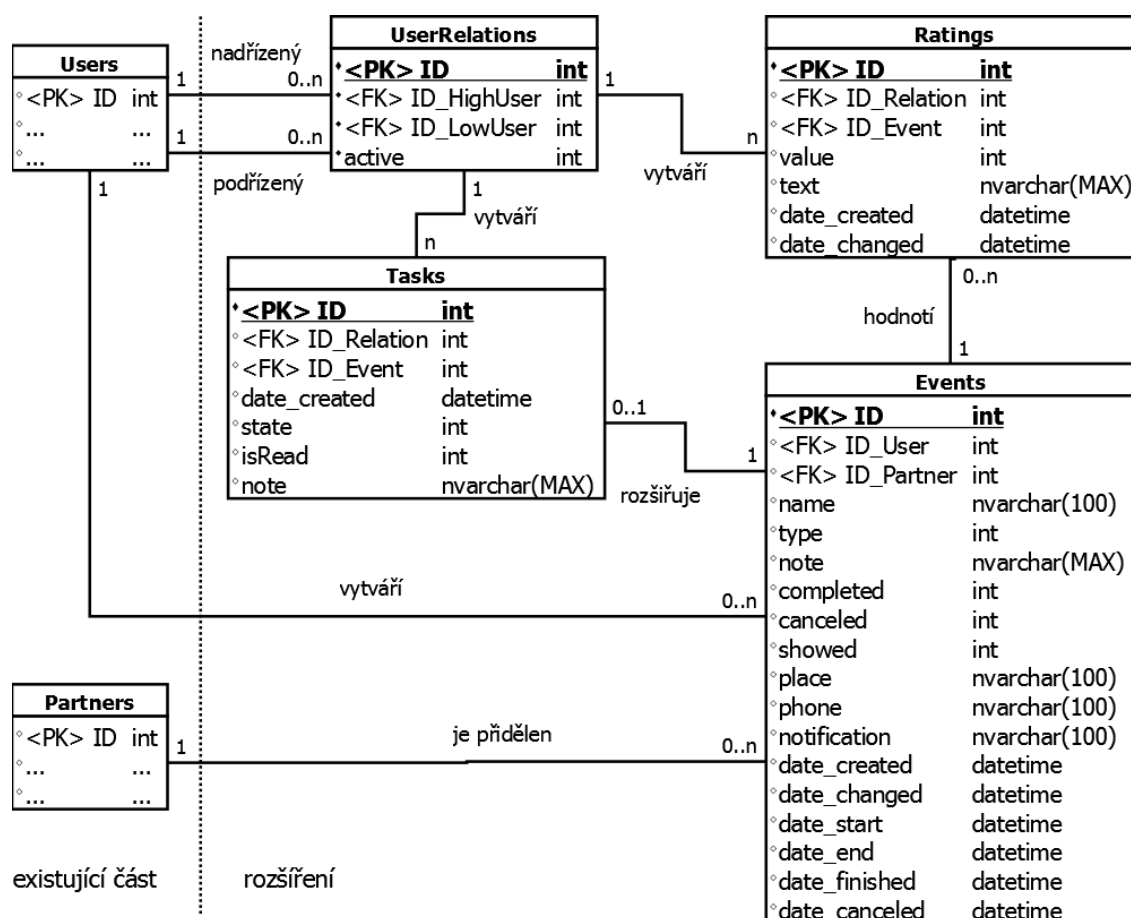
⁵Dostupné z <http://developer.android.com/sdk/index.html>

⁶Dostupné z <https://www.eclipse.org/downloads/>

⁷Dostupné z <https://developer.android.com/sdk/index.html>

4.4 Návrh databázové části

Pro práci s daty bylo potřeba vytvořit tabulky databáze. Pro užívání aplikace byly navrženy tabulky, které budou uchovávat vytvořené události, úkoly, hodnocení a vztahy jednotlivých zaměstnanců. Databáze s tabulkami, které uchovávají informace o uživateli, zákaznících a informace s nimi spojené již existuje, tudíž tuto databázi pouze rozšíříme o výše zmíněné tabulky. Tato databáze je typu MySQL. Součástí této práce byl návrh databázových tabulek, implementaci zajistila firma Nextis s.r.o., která poskytla i testovací data a server pro komunikaci s aplikací. Návrh databáze můžete vidět na obrázku 4.2. Všechna data jsou uložena na centrálním serveru, který je zpřístupněn aplikací skrze síť internetu. Interní databáze mobilního zařízení zůstává nevyužita.



Obrázek 4.2: Entity-relationship diagram

Tabulka *Events* obsahuje mnoho sloupců s údaji o každé položce, např. název, typ události, nebo na kdy je událost naplánována. Ke každé události je přiřazena právě jedna zodpovědná osoba z tabulky uživatelů a právě jeden partner, ke kterému je událost vázána, z tabulky partnerů.

Tabulka *UserRelations* zaznamenává, který uživatel je podřízen a komu. Každý záznam v tabulce značí vztah nadřizovaný - podřízený. Pokud se vazba vytváří nová, vytvoří se i nový záznam v této tabulce. Pokud však určitou vazbu chceme zrušit, pak měníme pouze hodnotu v sloupci *active*, abychom zachovali konzistenci databáze. Pokud nastane znovu situace, kdy chceme přidat vazbu, která již existovala, pak pouze vazbu vyhledáme a změníme její

hodnotu v sloupci *active*.

Tabulka *Tasks* obsahuje záznamy úkolů. Tyto úkoly jsou vytvářeny nadřízeným uživatelem a přidělovány konkrétnímu podřízenému uživateli. Proto je využita vazba s tabulkou *UserRelations*. Samotný záznam úkolu je pouze rozšířením události a označením, že událost byla vytvořena nadřízeným uživatelem. Pro každý záznam v této tabulce tedy musí existovat právě jeden záznam v tabulce *Events*.

Tabulka *Ratings* obsahuje hodnocení událostí podřízených lidí od nadřízených. Je tedy také vázána na konkrétní záznam v tabulce *UserRelations* a záznam v tabulce *Events*.

4.5 Návrh grafického uživatelského rozhraní

Návrh uživatelského rozhraní byl rozdělen do několika logických částí, kterým se budeme v této kapitole jednotlivě věnovat. Návrhy vzhledu ikon byly součástí návrhu grafického rozhraní, avšak jejich konečnou podobu zpracovala a poskytla firma Nextis s.r.o.

Denní přehled

V této části můžeme vidět události, které jsou naplánovány pro současný den. Jednotlivé události jsou graficky vyznačeny jako milníky na časové ose. Každý milník je vyznačen ikonou, která reprezentuje, o jaký typ události se jedná. Tyto položky budou seřazeny podle času začátku události, který bude vyznačen u každého bodu na časové ose. Takto bude uživateli vždy nabízena událost, která by měla být v nejbližší době splněna. Ke každému bodu je také vázána karta, která zobrazuje základní údaje o naplánované události. Zobrazení detailu je pak podmíněno kliknutím na kartu. Při splnění určité události se osa opticky posune, ukončená položka zmizí a do čela řady se dostává další v pořadí. Při kliknutí na milník časové osy se dle typu události vykoná akce s tímto typem spojená. Například typ schůzka je spojen s cestou na místo schůzky a je spuštěna navigace.

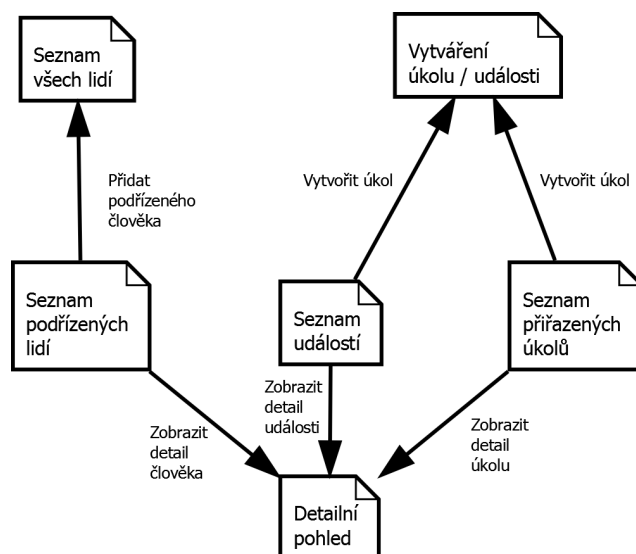
Manažerská část

V této sekci se nacházejí tři seznamy, které jsou zobrazeny na samostatných obrazovkách. První z nich zobrazuje seznam lidí označených jako podřízené. Z tohoto seznamu lze použít ikonu rychlé volby volání a kontaktovat vybraného člověka. Při kliknutí na daného člověka lze zobrazit obrazovku s jeho kontaktními údaji, denním plánem a posledním hodnocením, které bylo provedeno konkrétním uživatelem. V této části je také možné podřízenou osobu přiřadit do svého seznamu, ale také odstranit.

Další dva seznamy mají přímou souvislost s prvním. Druhý seznam obsahuje přehled událostí, naplánovaných pro podřízené lidi, zobrazuje přiřazení k dané osobě, začátek události, jejich stav a přiřazeného partnera.

Třetí seznam obsahuje položky událostí, ale pouze ty, které byly zadány jako úkol. Lze zde vidět stejné informace jako u seznamu předchozího.

Tyto seznamy lze filtrovat podle nejbližších dní. V obou seznamech se při kliku na položku zobrazí podrobnější informace. Kromě sledování lze z této sekce také úkoly zadávat. V celé manažerské části lze použít vyhledávání pro filtrování zobrazených položek v seznamech. Tento návrh je zobrazen na obrázku 4.3. Každá položka představuje vlastní obrazovku a šipky přechody, za jakým účelem chceme obrazovku zobrazit.



Obrázek 4.3: Návrh rozložení manažerské části aplikace

Kalendář

V této části jsou zobrazeny veškeré události, které uživatel vytvořil nebo mu byly přiděleny a schváleny. Pro každý den existuje jejich samostatný seznam a lze prohlížet události, které jsou naplánovány do budoucna, i ty, které již proběhly. Prohlížení kalendáře je tak možné na úrovni dne, týdne i měsíce. Při vybrání konkrétní události se zobrazí její detailní pohled. K aktuálně vybranému dni lze také vytvořit událost.

Adresář

V adresáři může uživatel zobrazit veškeré informace o partnerovi. Po vyhledání a vybrání partnera ze seznamu jsou zobrazeny detailní informace o partnerovi. Součástí těchto informací jsou kontakty, obchodní informace, zápisy CRM a události spojené s tímto partnerem.

Úkoly

Tato část obsahuje přehled úkolů, které jsou rozdělené do dvou skupin - nové a přečtené. Při obdržení nového úkolu se zobrazí položka v odpovídajícím seznamu. Po kliknutí se dostává uživatel do detailního pohledu. V této části se uživatel může rozhodnout, zda úkol přijme či ne.

Nastavení

V této sekci uživatel může upravit své priority při používání aplikace. Mezi tyto možnosti patří nastavení PIN kódu, časový interval automatického zamknutí, povolení notifikací o obdržení úkolu a interval jejich synchronizace.

Detailní pohled

Detailní pohled je část, která je zobrazena při vybrání položky jakéhokoliv seznamu v aplikaci. Každý pohled zobrazuje data, která jsou relevantní k vybrané položce a přiřazenému

stupni oprávnění. Tyto pohledy jsou v mnoha případech interaktivní. Většina údajů obsahuje ikonku, která reprezentuje stav či akci dat, které jsou zobrazeny. Při vybrání specifických položek se provede odpovídající akce. Tyto akce budou detailněji popsány v kapitole 5.

Vytváření událostí

Vytváření a plánování událostí je jedna z nejdůležitějších částí této aplikace. Proto byl při návrhu kladen důraz na zjednodušení práce s jednotlivými položkami. Při vytváření akce do kalendáře jsou zde jen tři povinná pole pro vyplnění: typ vytvářené události, partner se kterým je spojená, a časový údaj o plánovaném začátku. Dalšími volitelnými údaji jsou například název, poznámka, kontakt či místo. Tyto údaje mohou být v době, kdy má být událost provedena, využity k doplňkovým funkcím, např. navigaci na místo určení.

Těchto polí k vyplnění je poměrně dost a při výběru pole a psaní každého textového údaje je poměrně zdlouhavé. Navíc hledání některých údajů, např. kontaktu, může být i nemožné (neuložené číslo v telefonním seznamu). Proto byl navrhnut systém inteligentních dialogů, které budou pomáhat s vyplňováním údajů. K vyplnění budou nabízet data relevantní k danému partnerovi.

Předem definované dialogy jsou uživateli postupně nabízeny k vyplnění. Uživateli tak odpadá nutnost klikat na každou položku zvlášť. Je samozřejmostí, že toto chování nemusí vyhovovat každému uživateli. Proto je v nastavení možnost tuto funkci vypnout či upravit podle potřeb uživatele.

Kapitola 5

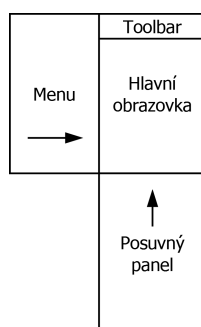
Implementace

Tato kapitola pojednává o celkovém konceptu aplikace a o tom, jak bylo postupováno při implementaci jejich jednotlivých částí. Součástí kapitoly je část, která rozebírá datovou komunikaci mezi mobilním zařízením a serverem. Závěr kapitoly je zaměřen na použité knihovny a jejich využití v aplikaci.

Aplikace byla implementována v programovacím jazyce Java. Podpora byla vybrána pro verze 4.1 Jelly Bean (verze API 16) a vyšší. Zařízení od této verze až po nejnovější verzi 5.1 Lollipop tvoří 88,7 % všech aktivních zařízení s operačním systémem Android [5].

Pro vytvoření grafického rozhraní byly použity komponenty jak z knihoven operačního systému Android, tak z externích knihoven zmíněných v kapitole 5.5. Při vytváření rozvržení jednotlivých obrazovek byly upřednostňována schémata systému Android s relativním rozmístěním a podporující vrstvení komponent, zejména třídy `RelativeLayout` a `FrameLayout`. V minulosti byla používána zejména schémata `LinearLayout`, které podle typu orientace umísťují své prvky do horizontálních či vertikálních řad. Při tomto přístupu však často dochází k přílišnému větvení stromu prvků, a to je poté příčinou pomalého vykreslování a velké paměťové náročnosti aplikace.

Při skládání komponent do jednotlivých obrazovek jsem se řídil zásadami Material Designu [2]. Vzor pro zobrazení všech logických částí můžete vidět na obrázku 5.1. *Menu* je možné zobrazit gestem přesunutí z levého okraje obrazovky směrem doprava. *Posuvný panel* již není uzpůsoben k posunu gestem, lze jej zobrazit pouze programově. *Toolbar* je prvek, který obsahuje aktivní ikonu, vysouvající menu, název stránky, na které se nacházíme, a případně ikonky doplňující ovládání aktivní obrazovky. Hlavní myšlenkou tohoto rozložení je vkládání jednotlivých logických částí, tak, jak byly představeny v kapitole 4.2, do oblasti *hlavní obrazovky*.



Obrázek 5.1: Rozvržení částí hlavní aktivity

5.1 Implementace částí aplikace

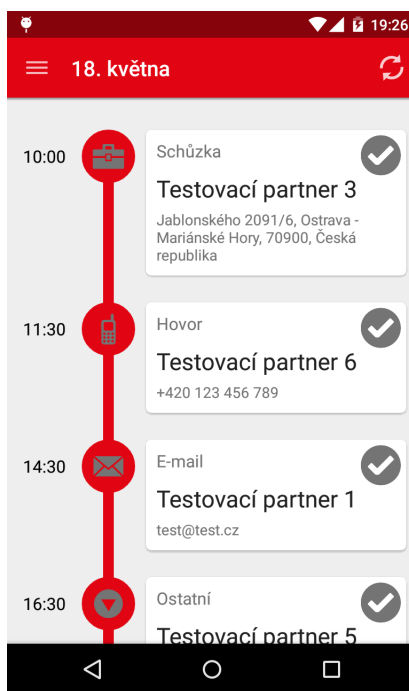
Celý tento návrh představuje hlavní aktivitu, která je spuštěna a zobrazena po celou dobu běhu aplikace. Do prostoru hlavní obrazovky jsou vkládány jednotlivé části, které jsou implementovány jako potomci třídy `Fragment`. Tyto fragmenty jsou vkládány do hlavní aktivity a přes rozhraní instance této třídy mají přístup k *toolbaru* i k *posuvnému panelu*.

5.1.1 Denní přehled

V této části je zobrazen denní přehled událostí, jak je popsán v kapitole 4.5. Pro obecný seznam je nejlepší použít prvek `ListView`. Tento prvek umožňuje posouvání v seznamu položek s optimálním využitím paměti. Je určen zejména pro větší seznamy, které by brzdily aplikaci. Tento princip je založen na recyklaci jednotlivých položek seznamu a vychází z myšlenky, že všechny položky vypadají stejně a je jen obsahově, např. textově nebo obrazově, upravován obsah objektů a jejich velikost se nemění.

V tomto případě jsem zvolil kombinaci prvků `ScrollView` a `LinearLayout` s vertikální orientací. K tomuto řešení mě vedly dvě skutečnosti. Tento seznam je určen pouze pro seznam událostí, určených pro jeden den. Během tohoto časového úseku nebude prvků v seznamu tolik, aby měly vliv na výkon aplikace. Druhý důvod byl ten, že karty zobrazující detailní data události, mohou být velmi rozdílné z pohledu délky textů. Pro tento přehled je důležité, aby tyto texty byly zobrazeny vždy celé. To má vliv na velikost karet, ve kterých jsou tyto texty umístěny. Proto by použití prvku `ListView` nebylo vhodné.

Karty jsou implementovány jako objekty instance třídy `CardView`, které obsahují základní prvky systému Android - textová pole (`TextView`) a obrázek (`ImageView`), zobrazující tlačítko na splnění události. Při vytvoření obrazovky a zavolání metody `onCreateView` jsou nejprve staženy události pomocí metody `downloadData` a v rámci této metody `fillList` vytvoří jednotlivé položky událostí.



Obrázek 5.2: Vzhled denního plánu

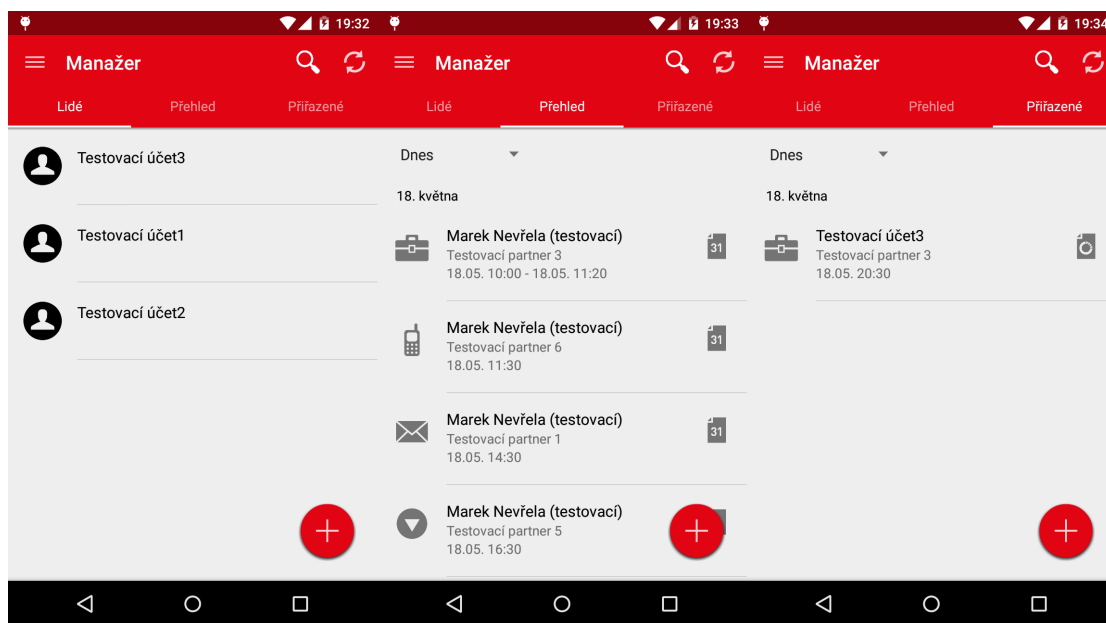
Po stisknutí tlačítka pro splnění události je událost ukončena a nabídnuta možnost přidat CRM záznam do databáze. V případě typu události schůzka je tento prvek povinný, v ostatních typech je volitelný. Náhled této části můžete vidět na obrázku 5.2.

5.1.2 Manažerská část

Tato část zobrazuje celkově tři seznamy. Tyto seznamy jsou zobrazeny v prvku `ViewPager`, což je komponenta, která je schopna zobrazovat jednotlivé instance třídy `Fragment` horizontálně umístěné vedle sebe a umožnit jejich procházení pomocí gest. Tento prvek je poměrně používaný, protože pro uživatele je pohodlnější než výběr z klasické kontextové nabídky nebo menu.

Ke správnému použití prvku `ViewPager` je potřeba adaptér, který pomáhá vytvořit a naplnit tento prvek instancemi třídy `Fragment`. Adaptér je v tomto případě instancí třídy `Tabs`, která je detailněji rozebrána v kapitole 5.3.2. Data jsou stahována samostatně v jednotlivých instancích.

Ukázku z této části můžete vidět na obrázku 5.3, kde můžeme vidět 3 seznamy položek. Všechny jsou implementovány pomocí prvku `ListView` a adaptéru položek, které jsou implementovány dvouřádkovým (vlevo) nebo třířádkovým (uprostřed a vpravo) vzorem podle stylu Material Design.



Obrázek 5.3: Vzhled manažerské části aplikace

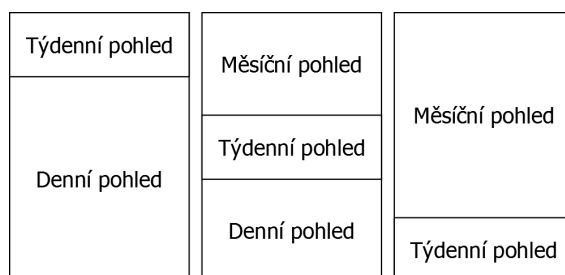
U každé z této stránek lze použít filtr vyhledávání. V horní liště `ToolBar` je po zvolení ikony lupy rozbaleno textové pole. Při postupném psaní textu do tohoto pole je volána metoda `filtr(String text)`, kde tento řetězec je vyhledáván v datech každé položky seznamu a v případě shody je zobrazen. V první části lze hledat uživatele podle jména, mailu a telefonního čísla. V částech seznamů událostí / úkolů lze filtrovat položky podle osoby zodpovědné za úkol nebo partnera přiřazeného k události. Filtr je zrušen v případě vymazání textového pole, zavření filtru nebo přechodu na jinou obrazovku.

U každého seznamu je použito plovoucí tlačítko `CircleButton` z knihovny `MaterialWidget` (kapitola 5.5), které je využito k zobrazení nabídky uživatelů pro přidání do seznamu

podřízených v případě prvního seznamu. U druhého a třetího seznamu zobrazuje obrazovku pro vytvoření úkolu. Vzhledem k vrstevnatosti zobrazení jednotlivých prvků (plovoucí tlačítko je umístěno nad seznamem položek) může při posunu seznamu na konec dojít k překrytí posledního prvku seznamu tímto tlačítkem. Proto je u každého seznamu prvků, u kterého je viditelné toto plovoucí tlačítko, použito zápatí. Toto zápatí je implementováno pomocí metody `addFooterView`. Tato metoda na konec seznamu prvků v komponentě `ListView` přidá prvek `View` a tím posune původně poslední prvek seznamu do viditelné pozice.

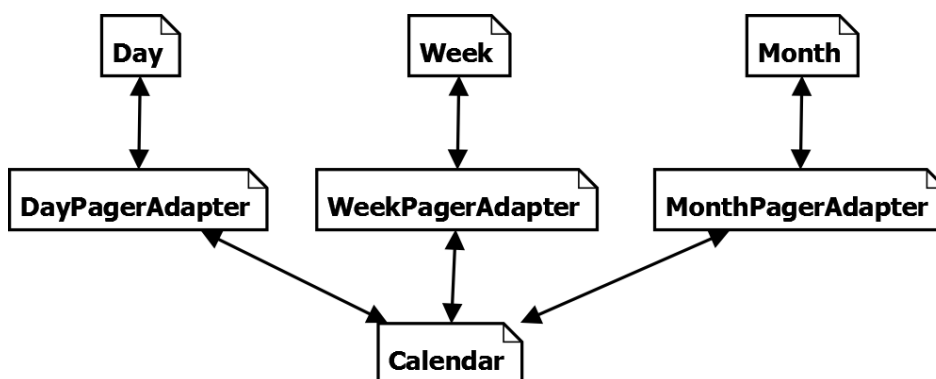
5.1.3 Kalendář

Tato část je rozdělena do tří sekcí. Každá sekce reprezentuje jeden z pohledů kalendáře (měsíční, týdenní a denní). Rozvržení této části využívá prvek `SlidingUpPanelLayout`, podrobněji zmíněný v kapitole 5.3.4. Klíčovou vlastností tohoto prvku je vertikálně posunovací lišta, která nám umožňuje překrýt jedno rozložení obrazovky druhým. Nákres použití v této části můžete vidět na obrázku 5.4.



Obrázek 5.4: Návrh rozložení kalendáře

Na první obrazovce (vlevo) lze vidět výchozí stav kalendáře. V tomto případě je zobrazena vyšší vrstva. *Týdenní pohled* je posouvateľný, takže jej můžete přesunout do spodní pozice, jak je zobrazeno na třetí obrazovce (vpravo). Mezistav při posouvání je zobrazen uprostřed. Při stavu plně zobrazeného měsíčního pohledu je zobrazen i pohled týdenní. Ten je však součástí vrstvy která měsíční pohled překrývá.

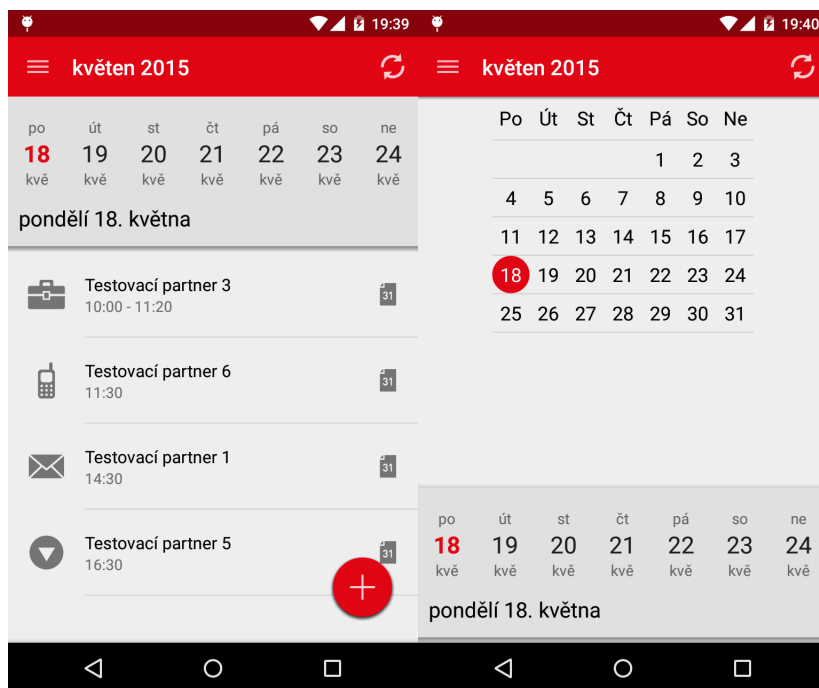


Obrázek 5.5: Komunikace mezi třídami kalendáře

Všechny tyto sekce jsou implementovány pomocí prvků `ViewPager`. Lze se mezi nimi tedy pohybovat horizontálními gesty. Vlastní třídy `Day`, `Month` a `Week` jsou potomci třídy `Fragment`, a jsou spravovány odpovídajícími adaptéry. Adaptéry jsou implementovány jako

potomci třídy `FragmentStatePagerAdapter` (kapitola 5.3.1). Všechny tyto vrstvy spolu vzájemně komunikují, jak je znázorněno na obrázku 5.5.

Procházení (listování) týdenním či měsíčním pohledem nemá vliv na denní pohled. Až při samotném vybrání konkrétního dne v daném pohledu je volána metoda `setDate` vlastní třídy `Calendar`, která nastaví aktuální datum a aktualizuje ostatní vrstvy kalendáře. Oproti tomu při procházení denní pohledem jsou ostatní vrstvy aktualizovány okamžitě. Právě až v jednotlivých instancích třídy `Day` dochází ke stahování dat. Tímto způsobem je stahováno pouze minimální množství dat.



Obrázek 5.6: Vzhled kalendáře

Pro vytvoření rozvržení dne byl opět použit prvek `ListView` se zápatím a třířádkovým vzhledem pro položku stylu Material Design. Ukázku můžete vidět na obrázku 5.6. Po výběru konkrétní položky se zobrazí její detail. Z této obrazovky lze přejít do sekce vytváření událostí opět pomocí plovoucího tlačítka.

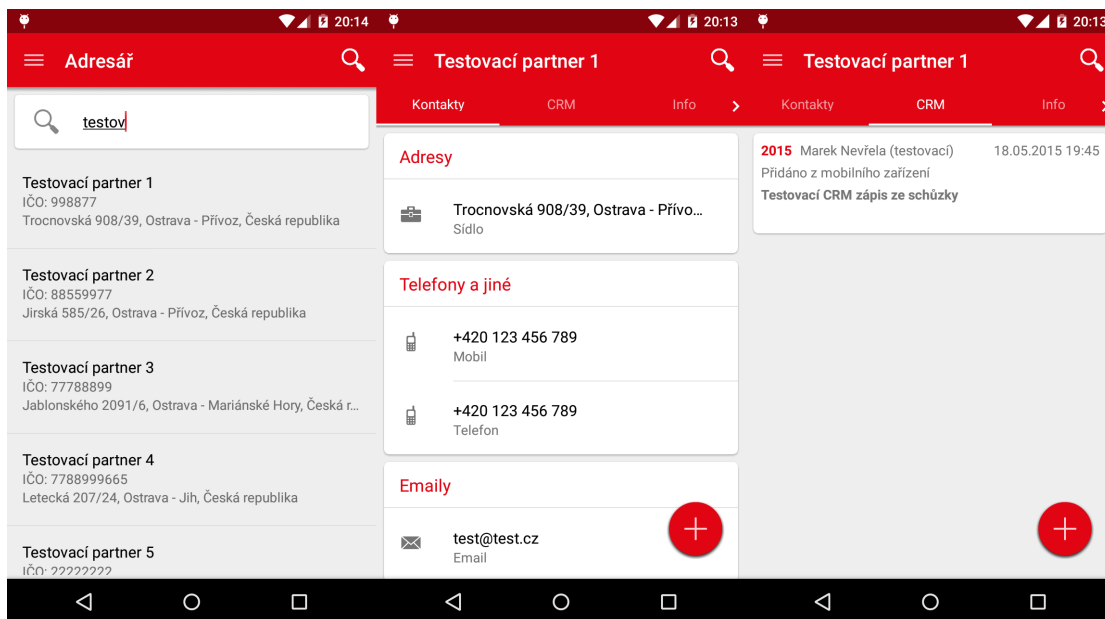
5.1.4 Adresář

Úvodní stránkou této části je vyhledávací pole, ve kterém lze partnery vyhledávat pomocí názvu firmy i jejich identifikačního čísla (IČO). Při zadání části názvu nebo čísla jsou vyhledáni všichni partneři, kteří obsahují tuto část a zobrazení v seznamu. Po vybrání konkrétního partnera je zobrazen jeho detail.

Tato část je rozdělena do čtyř stránek. Tyto stránky jsou zobrazeny jako instance třídy `Fragment` v prvku `ViewPager`. Adaptér je použit stejný jako v manažerské části - komponenta `Tabs`, která je detailněji popsána v kapitole 5.3.2.

Rozvržení této části můžete vidět na obrázku 5.7. Na levém obrázku je zobrazena úvodní stránka vyhledávání partnerů. Další dva obrázky jsou náhledy stránek kontaktů a CRM zápisu.

Sekce kontakty je rozdělena do tří částí - adresy, telefony a e-maily. Každá z těchto částí



Obrázek 5.7: Vzhled adresáře

nabízí položky, které po jejich vybrání aktivují příslušnou funkci (navigaci, volání, mailový klient). V CRM sekci jsou zobrazeny CRM zápisy k příslušnému partnerovi seřazené od nejnovější položky po nejstarší. V této sekci lze přidávat další položky skrze dialog. Tato možnost je podmíněna právy pro zápis CRM. V další části jsou zobrazena obchodní data a rozdělena do skupin. V poslední záložce je seznam událostí, vázaných k danému partnerovi. Pokud má uživatel práva manažera, pak jsou zobrazeny všechny události k tomuto partnerovi. Jinak jsou viditelné pouze události, vytvořené přihlášeným uživatelem.

5.1.5 Detailní pohled

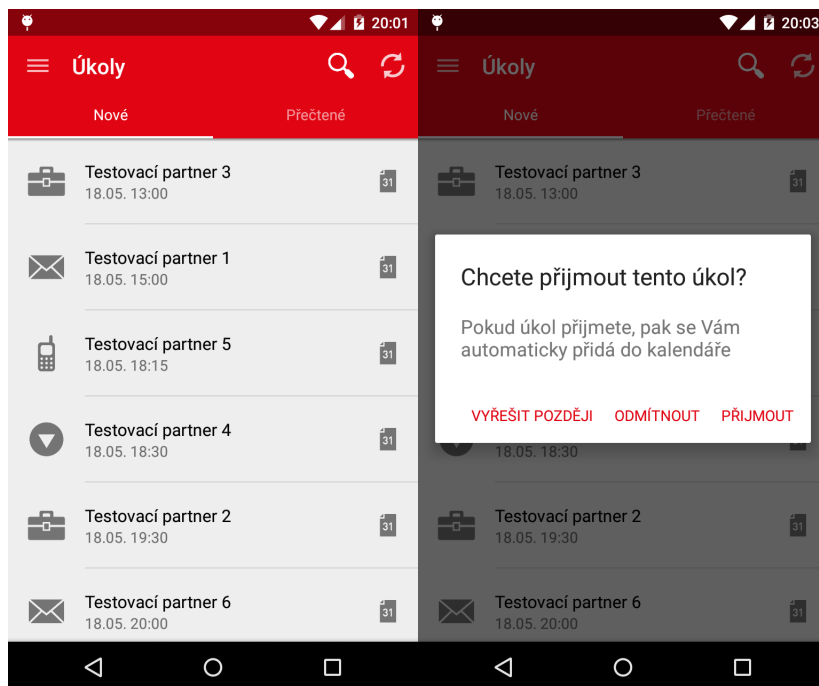
Tato část je vytvořená velmi specifickým způsobem. Využit je *posuvný panel* zmíněný na začátku kapitoly 5. Prostor této části rozložení je možné využít pro každou obrazovku (Fragment) této aplikace přes rozhraní hlavní aktivity Main. Proto tato část byla implementována jako instance třídy `FrameLayout` s myšlenkou vkládání detailních pohledů, či jiných instancí třídy `Fragment`.

Při spuštění aktivity je do této části vložena instance třídy `DetailCards`, která je detailně popsána v kapitole 5.3.5. Při vybrání položky jakékoliv seznamu je prostřednictvím hlavní aktivity zpřístupněn tento objekt a skrze své rozhraní je nastaven do takové podoby, aby odpovídal datům zvolené položky. Tento objekt je zde umístěn po celou dobu běhu programu, pouze je měněn jeho obsah. Po nastavení tohoto zobrazení je *posuvný panel* posunut do pozice, kdy překrývá celou *hlavní obrazovku*, včetně *Toolbaru*, a je tak zobrazen uživateli.

5.1.6 Úkoly

Tato sekce je rozdělena do dvou částí. První seznam zobrazuje úkoly nové a dosud nepřečtené. V druhém seznamu jsou umístěny všechny ostatní úkoly. Tyto seznamy jsou implementovány pomocí třídy `ListView` a využít je stejný adaptér jako v manažerské části a

použito je opět dvouřádkové rozložení položky. Náhled můžete vidět na obrázku 5.8.



Obrázek 5.8: Vzhled části úkoly

Při kliku na konkrétní úkol je zobrazen detail úkolu. Pokud je tento úkol nový, pak je v databázi označen jako přečtený. V detailním pohledu je možnost tento úkol přijmout či odmítnout. Pokud tak není učiněno a úkol je označen stále jako „bez odezvy“, pak je tato možnost nabídnuta po odejití z tohoto pohledu pomocí dialogu.

5.1.7 Vytváření události a úkolu

Vytváření událostí a úkolů zajišťuje vlastní třída `EventFragment`, která je také potomkem třídy `Fragment`. Tento objekt je přidán jako další vrstva do posuvného panelu a překrývá tak v době používání detailní pohled. Při ukončení vytváření či editace je vrstva smazána.

Tato obrazovka je pro přidávání událostí dostupná z manažerské části a z kalendáře. Pro obě se však liší. Pokud je událost vytvářena z kalendáře, pak si uživatel plánuje svůj denní program, tudíž není nabídnuta možnost přiřazení úkolu osobě a je zvolena implicitně. V případě manažerské části je tato možnost nabídnuta a je povinná pro vyplnění. Uživatel tak vybírá člověka ze seznamu podřízených a vytváří tak úkol.

Balíčky (instance třídy `Bundle`) jsou používány k předávání informací mezi fragmenty nebo aktivitami. V tomto případě jsou použity k určení, o jaký typ obrazovky se bude jednat - nová událost, nový úkol nebo editace události či úkolu.

Pro vyplňování položek lze nastavit tzv. průvodce, který postupně nabízí dialogy k vyplnění jednotlivých položek. Tohoto průvodce lze přizpůsobit v nastavení aplikace, případně tuto volbu úplně zakázat. V případě aktivního průvodce jsou v nastavení zvoleny položky, které mají být v průvodci zobrazeny. Pořadí prvků je předem určeno podle pořadí v rozložení obrazovky. Tyto prvky jsou při vytvoření instance třídy `Fragment` uloženy do seznamu podle zvoleného nastavení. Při potvrzení dat a zavření dialogu je otevřen dialog, uložený

v seznamu jako následující. Pokud chce uživatel průvodce ukončit, stačí uzavřít dialog bez uložení, např. kliknutím mimo dialog nebo klávesou zpět.

Nutností pro vytvoření události či úkolu je přiřazení partnera. V dialogu pro vyhledání a vybrání partnera jsou stažena data s ním spojená a poté jsou uživateli nabízena jako prioritní při výběru místa schůzky či kontaktu. Samozřejmostí je však i možnost pro zadání těchto údajů ručně. V případě, že chce uživatel zadat místo či kontakt dříve, než je zvolen partner, pak je mu nabídnut nejprve výběr partnera a poté až tyto položky.

5.2 Datová úložiště a komunikace

Tato kapitola popisuje postupy při datové komunikaci mezi zařízením a serverem. Všechna data zobrazená v aplikaci jsou přenášena přes internet z centrální databáze.

Jediná data, která jsou uložena v zařízení, jsou osobní nastavení uživatele. Tato data jsou uložena pomocí třídy `SharedPreferences`. Instance této třídy je zpřístupněna skrze vlastní třídu `UserSetting` a její statické metody. Data jsou tudíž jednoduše přístupná v celé aplikaci. V instanci třídy `SharedPreferences` jsou uloženy takové údaje, jako například id uživatele, přihlašovací jméno a heslo, pin, práva uživatele a další. Tyto údaje jsou vždy při přihlášení aktualizovány.

Komunikace se serverem probíhá asynchronně pomocí knihovny `WSClient` (kapitola 5.5). Zařízení komunikuje se serverem pomocí webových metod. Rozhraní webových metod bylo navrženo v rámci bakalářské práce. Jejich implementace však byla realizována firmou Nextis s.r.o.

Ke stahování dat je využívána třída `AsyncTask`. Tato třída umožňuje provádět operace v pozadí a výsledky těchto operací poté zveřejnit v uživatelském rozhraní. Při vytváření instance třídy `AsyncTask` jsou implementovány metody `onPreExecute` a `onPostExecute`, které využívají vlákno uživatelského rozhraní a metoda `doInBackground`, která běží ve zvláštním vlákne v pozadí.

V metodě `onPreExecute` je zpravidla skryt prvek, který má být naplněn daty, např. seznam, a zobrazen indikátor načítání (komponenta `CircularProgress`). Metoda `doInBackground` zajišťuje vytvoření instance třídy, která je použita pro vzdálenou komunikaci se serverem. Po ukončení stahování je volena metoda `onPostExecute`, v níž je skryt indikátor stahování. Pokud byla data v pořádku stažena, tak jsou poté zobrazena v odpovídajících prvcích.

5.3 Speciální komponenty

V této kapitole jsou popsány speciální komponenty, které byly použity při implementaci. Jsou to prvky vytvořené pro zvláštní účel buď rozšířením prvků knihovny, nebo úplně nové.

5.3.1 `FragmentStatePagerAdapter`

Tento prvek je rozšířením třídy `PagerAdapter`, která spravuje instance třídy `Fragment` jednotlivých stránek komponenty `ViewPager`. Tato třída také ovládá ukládání a obnovování stavu instance třídy `Fragment`. Tento prvek je mnohem více užitečnější pokud je použito větší množství stránek v komponentě `ViewPager`. Pokud stránky nejsou pro uživatele viditelné, pak je jejich obsah zničen a je uložen pouze jejich stav.

Tento proces ovlivňuje navíc vlastnost třídy `ViewPager` `OffscreenPageLimit`. Tato hodnota určuje, kolik stránek od aktivní stránky na každou stranu bude drženo v paměti.

Výchozí a zároveň minimální hodnotou je 1. Toto nastavení je kvůli možnosti animace při přechodu z aktivní stránky na stránku vedlejší.

Tento způsob práce s objekty třídy **Fragment** je mnohem více úspornější z hlediska paměťové vytíženosti prvku a celé aplikace [6]. Tento prvek má využití zejména v části kalendáře, kde může být vytvářeno v každé instanci třídy **ViewPager** mnoho instancí třídy **Fragment**. Instance třídy je vytvořena, avšak zabírá minimum paměti. V případě adaptéru denního pohledu je využit i životní cyklus jednotlivých instancí třídy **Fragment** a stahování jednotlivých dat probíhá až v metodě **onCreateView**, která je volána až při jeho zobrazení.

5.3.2 Tabs

Tato třída je založena na třídě **FragmentStatePagerAdapter** a rozšiřuje ji o možnost zobrazení záložek. Tato komponenta je použita v manažerské části, adresáři a úkolech. Ukázku můžete vidět na obrázku 5.9.



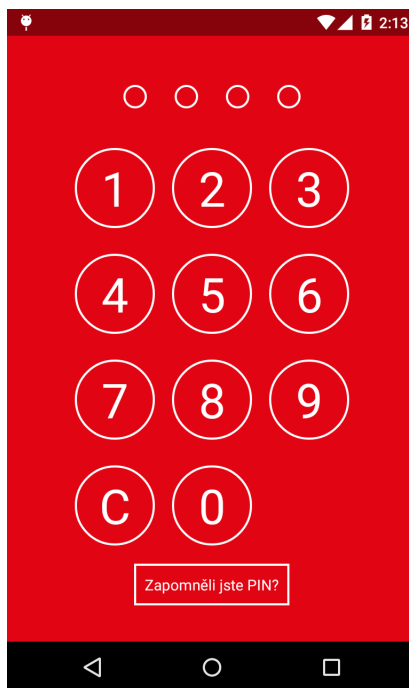
Obrázek 5.9: Vzhled komponenty Tabs

Při vytváření tohoto adaptéru je nastaven počet stran zobrazovaných instancí třídy **ViewPager** a rozložení, kde budou zobrazeny záložky. Podle počtu stran jsou vypočítány velikosti pro jednotlivé záložky a lištu, která identifikuje aktivní záložku. Pokud počet položek překračuje předem určený počet (v našem případě tři), pak je nastaven právě tento počet záložek. Záložky jsou zobrazeny ve vlastní komponentě **HorizontalScrollViewTabs**, která rozšiřuje třídu **HorizontalScrollView** a zajišťuje možnost posouvat se záložkami. Kromě toho zobrazuje na krajích obrázek šipky, pokud nejsou viditelné všechny záložky. Metoda **computeScroll** vypočítá pozici, do které se musí přesunout komponenta **HorizontalScrollViewTabs**, aby aktivní záložka byla vždy plně viditelná a zarovnaná do nejbližší pozice.

5.3.3 PIN

Tato vlastní komponenta byla vytvořena kvůli zabezpečenému a rychlému přístupu do aplikace. Pokud uživatel spustí aplikaci poprvé, pak se zobrazí tři textová pole pro přihlášení do aplikace, které je nutné vyplnit (server, přihlašovací jméno a heslo). Abychom nemuseli při každém spuštění aplikace vyplňovat všechny tyto údaje, byla vytvořena tato komponenta.

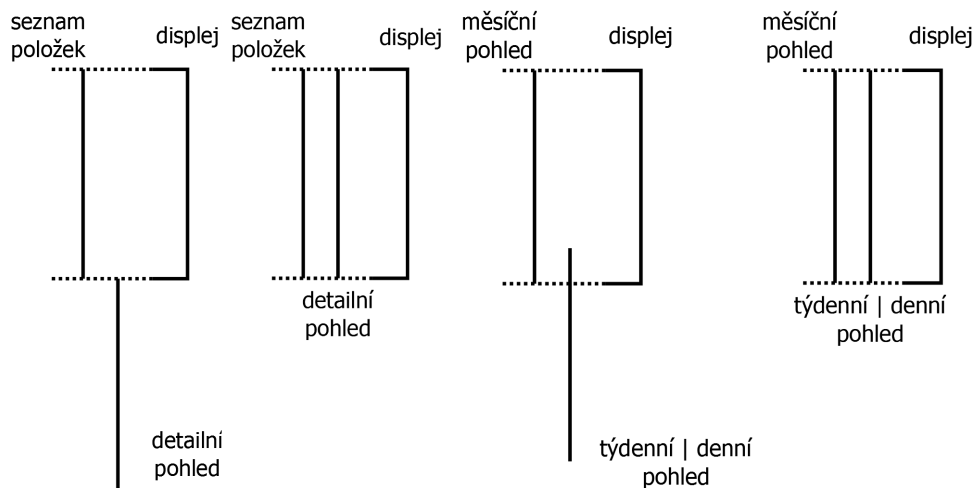
Po prvním přihlášení je přístup pomocí PINu deaktivován. Aktivace je možná v nastavení. Při zaškrtnutí této volby jste vyzváni k zadání PINu, který budete používat. PIN obrazovka je vytvořena tak, že reaguje na životní cyklus aktivity (obrázek 3.3). Využívány jsou metody hlavní aktivity **onResume** a **onPause**. Pro tuto obrazovku je nastavitelná doba aktivace zámku. Metoda **onPause** je volána při přechodu aktivity do pozadí. V této metodě je spuštěn časovač, který po zadané době aktivuje proměnou zámku. Při opětovném zobrazení je volána metoda **onResume** a pokud již vypršel limit pro aktivaci zámku, je zobrazena tato obrazovka. V případě, že uživatel PIN zapomene nebo zvolí PIN pětikrát špatně, pak je automaticky odhlášen a musí se přihlásit pomocí přihlašovacích údajů. PIN obrazovku můžete vidět na obrázku 5.10.



Obrázek 5.10: Vzhled PIN obrazovky

5.3.4 SlidingUpPanelLayout

Specifikem této komponenty je možnost vrstvení rozložení (layoutů) v rámci jedné obrazovky. Tyto komponenty lze vrstvit, takže může být překryto přes sebe několik vrstev. Tento prvek je součástí knihovny `AndroidSlidingUpPanel` (kapitola 5.5).



Obrázek 5.11: Náhled použití prvku `SlidingUpPanelLayout` v detailním pohledu

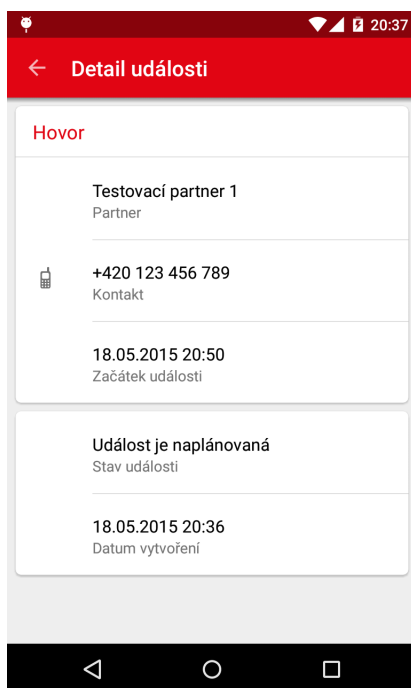
Náhled fungování je zobrazen na obrázku 5.11. Tento prvek je v aplikaci použit při dvou specifických situacích. První je vybrání položky jakéhokoliv seznamu a zobrazení jeho detailu (první a druhý obrázek zleva). Druhý případ je v sekci kalendáře. V této části překrývá týdenní a denní pohled vrstvu měsíčního pohledu. Na obrázku je zobrazen řez

vrstvami z profilu. Pohled uživatele je myšlen u každého obrázku zprava. Vodicí tečkované čáry vymezují viditelnou oblast pro uživatele aplikace.

Tato komponenta poskytuje mnoho možností modifikace. Lze jí nastavit prvek, kterým lze posouvat vrstvy, jeho výšku, průhlednost tohoto prvku či počáteční stav vrstev.

5.3.5 DetailCards

Tato komponenta byla navržena k zobrazení detailních dat. Tato třída je potomkem třídy **Fragment**. Její jediná instance byla vytvořena ihned po vytvoření hlavní aktivity **Main** a vložena na pozici *posuvného panelu* (viz obrázek 5.1). Při výběru položky v jakémkoliv seznamu aplikace je instance této třídy naplněna daty vztaženými ke zvolené položce a *posuvný panel* je vysunut do viditelné polohy.



Obrázek 5.12: Vzhled DetailCards komponenty

Tato třída poskytuje jednoduché rozhraní pro vytvoření detailního přehledu dat v jednotném stylu Material Designu. Související data jsou sloučena do skupin a každá skupina je reprezentována kartou (prvkem **CardView**). Každá karta může mít nadpis a sama je složena z položek. Pomocí metody **newItem** lze vytvořit několik druhů položek. Parametr **ListenerType** u této metody určuje, jakou funkci bude vykonávat při vybrání položky, např. navigace, volání nebo psaní mailu. Vzhled jednotlivých karet a položek můžete vidět na obrázku 5.12.

5.4 AndroidManifest, služby, přijímače

AndroidManifest je soubor ve formátu XML, popisující aplikaci. Součástí tohoto souboru je název balíčku aplikace nebo speciální oprávnění, které aplikace obsahuje. Obsahuje také popis použitých komponent aplikace, jako jsou např. aktivity, přijímače nebo služby. U každé

aktivity lze nastavit, jak bude reagovat na změnu orientace (překlopení) displeje. Pro tuto aplikaci byla zvolena pouze orientace *Portrait* (na výšku).

Povolení

Aplikace obsahuje povolení přístupu k internetu. Toto povolení je využíváno při každé komunikaci se serverem a získávání dat. Další povolení povoluje aplikaci uskutečňovat telefonní hovory. Poslední povolení umožňuje aplikaci naslouchat, kdy systém dokončí své spuštění. Toto povolení je při instalaci deaktivováno.

Přijímače

Tato aplikace používá dva přijímače. **AlarmReceiver** je potomek třídy **BroadcastReceiver** a zajišťuje opakované spuštění služby **AlarmService** v určitých intervalech. Při nastavování opakovaného alarmu, který aktivuje tuto službu, je také aktivováno nastavení povolení, které aktivuje naslouchání restartu zařízení. Opakování v časových intervalech je zajištěno pomocí instance třídy **AlarmManager**.

BootReceiver je třída, která je také potomkem třídy **BroadcastReceiver** a její metoda **onReceive** je volána v případě, když systém pomocí broadcastového volání oznámí, že spuštění systému bylo dokončeno. Toto platí pouze v případě, že tento přijímač byl aktivován. Implementací metody **onReceive** je nastavení alarmu, který zajišťuje aktivaci služby **AlarmService**.

Služby

V aplikaci je využita pouze služba třídy **AlarmService**, která rozšiřuje třídu **IntentService**. Tato služba zjišťuje, zda existuje v databázi úkol pro daného uživatele, který by byl nepřetčený. V tomto případě by to byly ty úkoly, které byly nově zadány od nadřízeného pracovníka. Pokud takový úkol existuje, pak je zobrazeno upozornění v notifikační liště telefonu. Po kliknutí na notifikaci se uživatel dostává do části úkoly.

5.5 Použité knihovny

Android Support Library

Tento balíček obsahuje několik knihoven [7]. Některé knihovny zajišťují zpětnou kompatibilitu komponent pro starší systémy a jiné rozšiřují původní komponenty a snaží se tak ulehčit vývojářům práci s těmito prvky. V této aplikaci jsou použity knihovny *v4 Support Library*, *v7 appcompat library* a *v7 cardview library*. Tyto knihovny jsou využívány zejména kvůli prvkům **Fragment**, **ViewPager**, **Toolbar**, **SearchView** a **CardView**. Komponenty z těchto knihoven byly použity v rámci celé aplikace. Knihovna je distribuována pod licencí Apache License 2.0.

DateTimePicker

Tato knihovna¹ poskytuje prvky pro výběr data a času. Prvky této knihovny jsou zobrazeny v dialogích a poskytují jednotný vzhled výběru ve všech podporovaných verzích OS Android. Pomocí této komponenty byla vytvořena třída **DateTimePicker**, která je využívána při

¹Dostupné z <https://github.com/flavienlaurent/datetimetypepicker>

vytváření události a při filtrování dle data v manažerské části. Knihovna je distribuována pod licenci Apache License 2.0.

MaterialWidget

Knihovna MaterialWidget² poskytuje základní komponenty, které jsou vzhledově stejné jako prvky Material Designu v OS Android 5.0 a vyšším. Použity jsou obdoby prvků `CheckBox`, `EditText` nebo `RadioButton`. Navíc jsou použity prvky `CircularProgress` a `CircleButton`. Tyto prvky jsou použity skrze celou aplikaci místo původních systémových. U této knihovny není uvedena licence.

JodaTime

Tato knihovna³ je určena pro práci s daty a časem. Oproti klasické třídě `Calendar` z knihovny *java.util* poskytuje mnohem širší API, ať už se jedná o vytváření nových objektů, získávání jednotlivých údajů, nebo aritmetice s daty. Třídy této knihovny jsou preferovány v celé aplikaci při práci s daty. Knihovna je distribuována pod licenci Apache License 2.0.

AndroidSlidingUpPanel

Knihovna AndroidSlidingUpPanel⁴ poskytuje prvek, který umožňuje posouvání vrstev rozložení (layoutu) přes sebe (kapitola 5.3.4). Knihovna je distribuována pod licenci Apache License 2.0.

Material-drawer

Tato knihovna⁵ je určena k vytváření (generování) bočního vysouvacího menu. Toto menu je generováno dle standardů Material Designu. Knihovna je distribuována pod licenci Apache License 2.0.

WSClient

Knihovna WSClient je prostředník v komunikaci mezi zařízením a serverem. Webová služba je systém umožňující komunikaci zařízení na síti. Tento druh služby je popsán ve strojově zpracovatelném jazyku WSDL.⁶ Služba komunikuje pomocí protokolu SOAP, který je založen na formátu XML. Knihovna tedy při odeslání požadavku na server převede volání metod na XML strukturu a tu odesílá serveru. Při obdržení odpovědi je postup opačný - XML struktura je za použití parseru analyzována a je vytvořen odpovídající objekt odpovědi. Knihovna tedy provádí tuto konverzi a zajišťuje komunikaci. Tato knihovna byla distribuována spolu se softwarem WSClient++, zakoupené a poskytnuté firmou Nextis s.r.o.

²Dostupné z <https://github.com/keithellis/MaterialWidget>

³Dostupné z <http://www.joda.org/joda-time>

⁴Dostupné z <https://github.com/umano/AndroidSlidingUpPanel>

⁵Dostupné z <https://github.com/HeinrichReimer/material-drawer>

⁶WSDL - Web Services Description Language, zapisuje se ve formátu XML

Kapitola 6

Testování

V této kapitole bude popsáno jakými způsoby byla aplikace testována. V první části bude zmíněno testování při vývoji aplikace. Druhá část se věnuje testování uživatelského rozhraní nezávislými uživateli formou interakce s aplikací a vyplněním dotazníku.

6.1 Testování funkčnosti aplikace

Testování aplikace probíhalo po celou dobu vývoje na dvou zařízeních vyznačených v tabulce 6.1. Nejprve bylo testováno grafické uživatelské rozhraní aplikace na těchto zařízeních a odladěny nedostatky týkající se odlišných velikostí obrazovek. Poté byla testována funkčnost jednotlivých kontrolních prvků aplikace. Pozornost byla věnována také plynulosti aplikace. Nejnáročnější obrazovkou, z pohledu využití procesoru a paměti, je kalendář. Tato obrazovka obsahuje i při běžném používání mnoho objektů, které reprezentují jednotlivé dny, týdny a měsíce a ovládací prvky v každém z nich. Pokud přecházíme gesty mezi dny, pak je každý den aktivován a paměť zařízení pro něj a grafické prvky, které obsahuje, alokuje místo. Navíc v každém dnu jsou vytvářeny instance třídy `AsyncTask`. Tyto instance vytvářejí nová vlákna, která zajišťují stahování dat ze serveru. Na těchto testovacích zařízeních však vizuálně nebylo zjištěno žádné zpomalení, pouze nástroje pro vývoj aplikací bylo možné sledovat nárůst využití těchto prostředků. Pro testování správnosti komunikace se serverem a operacemi nad databází byla vytvořena sada testů. Tyto testy jsou součástí zdrojových kódů aplikace a ověřují správnost implementace. Tyto testy jsou zaměřeny zejména na práci s tabulkami databáze, které byly navrženy jako rozšíření stávající databáze.

Značka Model	Obrazovka	Procesor	Frekvence	Počet jader	RAM	Verze
LG Nexus 4	4.7" 768x1280	Qualcomm Snapdragon	1.5 GHz	4	2048 MB	5.1
Asus Zenfone 4	4.5" 480x854	Intel Atom	1.2 GHz	2	1024 MB	4.4.2

Tabulka 6.1: Zařízení použita při testování aplikace

6.2 Testování uživatelského rozhraní

V této části je popsáno testování uživatelského prostředí. Toto testování proběhlo na vzorku osmi lidí. Tito lidé dostali k dispozici aplikaci a úkoly, které reprezentovaly základní operace s aplikací. Formou ankety poté hodnotili náročnost jednotlivých úkolů. Anketa je rozdělena do tří částí. První část obsahuje pět úkolů, které byly určeny zejména k seznámení s aplikací. Ve všech úkolech bylo naznačeno, v které části aplikace lze úkol vykonat. U těchto úkolů se předpokládalo jeho splnění a po uživateli se požadovalo pouze ohodnocení náročnosti úkolu, případně označení, pokud se úkol nepovedlo provést. V druhé části byly testovány náročnější úlohy, jejichž cílem bylo najít informace specifikované v každém úkolu. V této části uživatel u každého úkolu vyplňoval získanou informaci a opět náročnost plněného úkolu. Náročnost úkolu je určována v rozsahu 1 - 5 (1 = nejméně náročná, 5 = nejvíce náročná). V třetí části je zjišťován subjektivní názor uživatele na aplikaci. Výsledky testování prvních dvou částí jsou zobrazeny v tabulce 6.2. Celý dotazník je umístěn v příloze B.

číslo úkolů	hodnocení náročnosti	číslo úkolů	hodnocení náročnosti
1	1	6	1.13
2	1.5	7	1.5
3	2	8	1.5
4	1	9	2.13
5	2.25		

Tabulka 6.2: Průměrné hodnocení náročnosti jednotlivých úloh

Úkoly 1 - 5 byly úkoly poměrně triviální. Mezi nejjednodušší patřily úkoly č. 1, 2 a 4. V těchto úkolech bylo ke splnění potřeba pouze několik kliknutí a kroky, které vedly ke splnění, byly jasně viditelné. Úkol č. 3 a 5 byly průměrně hodnoceny stupněm 2 a 2.25. Toto hodnocení přičítám délce úkolu, který mohl být pro uživatele, který pracuje s aplikací poprvé náročnější než ostatní úkoly ve skupině.

V druhé části jsou zahrnuty úkoly složitější než v první části. U těchto úkolů byla brána v úvahu i správnost odpovědi. Všichni respondenti otázky zodpověděli správně, tudíž jediným kritériem pro vyhodnocení je bodování složitosti úloh. Úkoly č. 6 a 7 byly zaměřeny na vyhledání informací o zákazníkovi. Tyto úkoly nebyly problémové, i když s touto sekci uživatelé neměli zkušenosti z předchozích úkolů. Problémem nebylo ani zjištění telefonního čísla uživatele u úlohy č. 8. Vyšší ohodnocení získala pouze č. 9, kde bylo požadováno vyhledání schůzky podřízeného uživatele ze zadaného dne. V této úloze bylo klíčové přejít v manažerské části na obrazovku přehledu událostí a vybrat požadovaný den. Ve zpětné vazbě se poté často objevoval názor, že tato událost byly nejdříve hledána v detailu konkrétního uživatele, až poté v přehledu úkolů. Proto by stálo za úvahu, zda neposkytnout tuto možnost právě v detailním pohledu.

Poslední část dotazníku se zabývá komplexním hodnocením aplikace. Celková intuitivnost a přehlednost aplikace byla ohodnocena číslem 1.88. Toto číslo mohla negativně ovlivnit úloha č. 9, která byla zmíněna výše a uživatelé s ní měli problémy. Celkové ohodnocení aplikace má hodnotu 1.38.

Ve zpětné vazbě se nejčastěji objevovala připomínka k dialogům, které nabízejí vyplnění dalších položek při vytváření úkolu / události. Tuto možnost lze v nastavení aplikace vypnout. Automatické nabízení dialogů je však při výchozím nastavení aplikace aktivní a uživatelé informaci o možnosti vypnutí neměli. Jedním z návrhů vylepšení byla možnost

Značka Model	Obrazovka	Procesor	Frekvence	Počet jader	RAM	Verze
LG Optimus L5	4" 480x800	Mediatek MT6575	1 GHz	1	512 MB	4.1.2
Samsung Galaxy S4	5" 1080x1920	Qualcomm Snapdragon	1.9 GHz	4	2048 MB	5.0
Sony Xperia M	4" 480x854	Qualcomm Snapdragon	1 GHz	2	1024 MB	4.3
Sony Xperia SP	4.6" 720x1280	Qualcomm Snapdragon	1.7 GHz	2	1024 MB	4.3
Sony Xperia Z1	5" 1080x1920	Qualcomm Snapdragon	2.2 GHz	4	2048 MB	4.4.4

Tabulka 6.3: Zařízení použita při testování uživatelského rozhraní

přidání funkce, která umožní výběr SIM karty před voláním z aplikace. Tato připomínka se týká pouze velmi specifické skupiny telefonů. Systém Android poskytuje vlastní podporu pro práci s dvěma SIM kartami až od verze 5.1. Předchozí podpora byla zajištěna nadstavbami systému poskytnutých od konkrétního výrobce. Proto je pro vývojáře příliš obtížné až nemožné zajistit podporu této funkce u všech druhů telefonů. Poslední připomínka se týkala příliš ostrých barev. Tato poznámka se objevila pouze u jednoho uživatele, konkrétně u zařízení s displejem Super AMOLED. Tyto displeje zobrazují všechny barvy s příliš velkým kontrastem, což mohlo být důvodem této připomínky na rozdíl od ostatních uživatelů, kterým barva nevadila. Uživatelské testy probíhaly na mobilních zařízeních specifikovaných v tabulce 6.3 a uživatelé neupozornili na žádné neobvyklé chování aplikace.

Kapitola 7

Závěr

Cílem práce bylo vytvořit aplikaci pro obchodní zástupce firem, která bude nástrojem při vytváření a organizování pracovního programu. Hlavní přínos aplikace tkví ve spojení kalendáře pro plánování událostí uživatele a poskytování neustále aktuálních informací o zákaznících, které jsou pro práci obchodního zástupce klíčové. Tato aplikace je rozšířením desktopové aplikace Nextis Project od firmy Nextis s.r.o. na mobilní zařízení. Navíc byl tento informační systém, zaměřený na zákazníky, rozšířen o možnost správy denního programu uživatele a možnosti delegace úkolů mezi nimi.

V úvodu práce bylo popsáno CRM a operační systém Android, který je cílovou platformou aplikace. V práci jsou také specifikovány požadavky obchodních zástupců a návrh, který vychází z CRM strategie. Druhá polovina práce je věnována implementaci aplikace a jejímu testování.

Rozšíření aplikace by se mohlo týkat detailních pohledů uživatelů nebo událostí. Po kliknutí na zákazníka by mohla být zobrazeny informace o něm. Podobným způsobem by bylo možné v detailním pohledu uživatele vyhledávat události v rámci dnů nebo zobrazovat jejich detailní pohledy. Takto by mohly být jednotlivé části aplikace více provázány.

Nutností pro práci s aplikací je neustálý přístup k internetu. Zde se nabízí možnost rozšíření i pro stav, kdy internetová síť nebude dostupná. Pro tuto situaci by muselo zařízení uchovávat data. Toto řešení by bylo nutné navrhnout a implementovat synchronizaci pouze potřebných dat. Například určených pouze pro aktuální den. Tato data jsou samozřejmě citlivá a uchovávat je v zařízení nemusí být bezpečné. Součástí návrhu by proto muselo být také šifrování těchto dat.

Aplikace byla vzhledově primárně směřována pro chytré mobilní telefony, tzn. zařízení s obrazovkou o velikosti 4 - 5". Pro používání aplikace na tabletech by mohl být potenciál velkého displeje využit lépe a to zejména v rozmístění ovládacích prvků a množství zobrazených informací. Využit by se dala také k lepším statistikám, např. zobrazení informací pomocí grafů.

Literatura

- [1] Gradle [online]. 2015 [cit. 2015-04-28].
URL <http://gradle.org/getting-started-android/>
- [2] Material Design [online]. 2015 [cit. 2015-04-28].
URL <http://www.google.com/design/spec/material-design/introduction.html>
- [3] Open Handset Alliance. 2015 [cit. 2015-05-13].
URL <http://www.openhandsetalliance.com/>
- [4] Al-Haj-Epraheem, Y.: *Systém CRM (řízení vztahu se zákazníky) ve společnosti Johnson and Johnson, s.r.o.* bakalářská práce, FIS VŠE v Praze, 2010.
- [5] Android Developers: Dashboards [online]. 2015 [cit. 2015-05-05].
URL <https://developer.android.com/about/dashboards/index.html>
- [6] Android Developers: FragmentStatePagerAdapter [online]. 2015 [cit. 2015-05-05].
URL <http://developer.android.com/reference/android/support/v4/app/FragmentStatePagerAdapter.html>
- [7] Android Developers: Android Support Library [online]. 2015 [cit. 2015-05-11].
URL <http://developer.android.com/tools/support-library/index.html>
- [8] Android Developers: Application Fundamentals [online]. 2015 [cit. 2015-05-14].
URL <http://developer.android.com/guide/components/fundamentals.html>
- [9] Android Developers: Supporting Multiple Screens [online]. 2015 [cit. 2015-05-14].
URL http://developer.android.com/guide/practices/screens_support.html
- [10] Dohnal, J.: *Řízení vztahů se zákazníky: procesy, pracovníci, technologie*. Grada, 2002 [cit. 2015-05-15], iISBN 978-80-247-0401-2.
- [11] DSL.sk: Android dostane rychlejší runtime už v nejbližší verzi [online]. 2014 [cit. 2015-05-14].
URL <http://dsl.sk/article.php?article=15739>
- [12] JetBrains: Industry Recognition and Awards [online]. 2015 [cit. 2015-04-28].
URL <https://www.jetbrains.com/company/press/awards.html>
- [13] Karásek, J.: Android aktivně používá miliarda uživatelů + další zajímavá čísla z Google I/O [online]. 2014 [cit. 2015-05-14].
URL <http://smartmania.cz/bleskovky/android-aktivne-pouziva-miliarda-uzivatelu-a-dalsi-zajimava-cisla-z-google-i-o-812>

- [14] Marketing Journal: Počet uživatelů mobilního internetu se během roku 2014 zvýšil téměř o 50 % [online]. 2015 [cit. 2015-03-30].
URL http://www.m-journal.cz/cs/aktuality/pocet-uzivatelu-mobilniho-internetu-se-behem-roku-2014-zvysil-temer-o-50--_s288x11109.html
- [15] Petersen, R.: 21 experts define CRM in their own words and pictures [online]. 2012 [cit. 2015-05-15].
URL <http://barnraisersllc.com/2012/06/21-experts-define-crm-words-pictures/>
- [16] Plesac, Z.: Eclipse is dead for Android development and I helped kill it [online]. 2014 [cit. 2015-04-28].
URL <https://www.infinum.co/the-capsized-eight/articles/eclipse-is-dead-for-android-development-and-i-helped-kill-it>
- [17] Polesný, D.: Android bude všude: Google představil hromadu novinek [online]. 2014 [cit. 2015-05-13].
URL <http://www.zive.cz/clanky/android-bude-vsude-google-predstavil-hromadu-novinek/sc-3-a-174296/default.aspx>
- [18] Roberts, L. P.: The History of CRM – Moving Beyond the Customer Database [online]. 2004 [cit. 2015-05-16].
URL <http://customerservicezone.com/customerserviceguest/crmhistory.htm>
- [19] Semecký, V.: Android Studio - nové vývojové prostředí [online]. 2013 [cit. 2015-04-28].
URL <http://www.zdrojak.cz/clanky/android-studio-nove-vyvojove-prostredi/>
- [20] Štosek, J.: *Analýza a srovnání Dynamics CRM 4.0 a 2011 s Dynamics CRM Online*. Diplomová práce, Masarykova univerzita, 2011.
- [21] Ujbányai, M.: *Programujeme pro Android*. Grada Publishing a.s, 2012, iSBN 978-80-247-3995-3.
- [22] Wessling, H.: *Aktivní vztah k zákazníkům pomocí CRM*. Grada Publishing, 2002, 80-247-0569-9.

Příloha A

Obsah CD

Příložené CD obsahuje následující položky:

- **NextisPlanner.apk** - instalační soubor aplikace
- **Mobilní CRM aplikace pro obchodní zástupce.pdf** - bakalářská práce v elektronické podobě
- **LatexSource/** - složka se zdrojovými kódy bakalářské práce
- **NextisPlanner/** - složka se zdrojovými kódy aplikace a soubory potřebnými ke spuštění v prostředí Android Studio

Příloha B

Dotazník k testování uživatelského rozhraní

První část

1. Nastavte PIN kód pro zabezpečený vstup do aplikace
2. Ze sekce kalendář vytvořte událost na dnešní den. Typ události - schůzka, partner - Nextis Services s.r.o. Událost bude probíhat od 20:00 do 22:00. Pro místo události zvolte sídlo partnera.
3. V denním přehledu (1. záložka v menu) spusťte navigaci na místo určení schůzky. Poté se vraťte do aplikace a potvrďte splnění této události. Do CRM zápisu vepište "testovací zápis" a potvrďte.
4. Z pozice manažera si přidejte uživatele Marek Nevřela do svého seznamu lidí a zavolejte mu.
5. Vytvořte z pozice manažera uživateli Marek Nevřela libovolný úkol.

Druhá část

6. Zjistěte, kolik je k dispozici emailových kontaktů u firmy Testovací partner 5.
7. Najděte CRM zápis u firmy Testovací partner 3 z 28. 4. 2015, 10:09. Do odpovědi přepište text zápisu.
8. Zjistěte telefonní číslo uživatele Marek Nevřela.
9. Zjistěte jméno partnera, s kterým měl uživatel Marek Nevřela dne 27. 4. 2015 naplánovanou schůzku

Hodnocení aplikace

10. Ohodnoťte intuitivnost a přehlednost aplikace
11. Ohodnoťte aplikaci jako celek

12. Pokud máte nějaký nápad na zlepšení, pak jej napište zde.
13. Naskytl se nějaký problém při používání aplikace? Pokud ano, jaký?
14. Napište typ Vašeho mobilního telefonu a verzi OS.
15. Zde se můžete volně vyjádřit k čemukoliv ohledně aplikace.