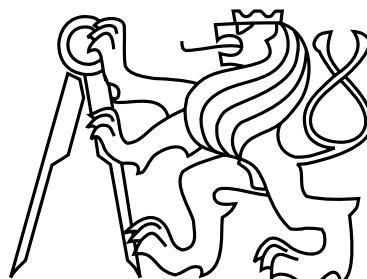


Na tomto místě bude oficiální zadání vaší práce

- Toto zadání je podepsané děkanem a vedoucím katedry,
- musíte si ho vyzvednout na studijním oddělení Katedry počítačů na Karlově náměstí,
- v jedné odevzdané práci bude originál tohoto zadání (originál zůstává po obhajobě na katedře),
- ve druhé bude na stejném místě neověřená kopie tohoto dokumentu (tato se vám vrátí po obhajobě).

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce

Inteligentní kuchařka pro Android

Michal Koutný

Vedoucí práce: Ing. Martin Balík, Ph.D.

Studijní program: Softwarové technologie a management, dobíhající, Bakalářský

Obor: Softwarové inženýrství

26. května 2017

Poděkování

Děkuji vedoucímu bakalářské práce, panu Ing. Martinu Balíkovi, Ph.D. za vedení během práce a cenné rady.

Prohlášení

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu §60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 26. 5. 2017

.....

Abstract

The theme of this bachelor thesis is development of the Cookbook application for Android operating system. This application will allow user to keep a list of materials which are available to him. It will also show him recipes that will be prepared from available materials.

Abstrakt

Tématem bakalářské práce je vývoj aplikace Kuchařka pro operační systém Android. Tato aplikace bude uživateli umožňovat vést seznam surovin, které má k dispozici. Dále bude zobrazovat recepty, které půjdou z dostupných surovin připravit.

Obsah

| | | |
|----------|---|-----------|
| 1 | Úvod | 1 |
| 2 | Mobilní operační systémy a aplikace | 3 |
| 2.1 | Výhody nativní aplikace | 3 |
| 2.2 | Mobilní operační systémy | 3 |
| 3 | Úvod do Androidu | 5 |
| 3.1 | Definice Androidu | 5 |
| 3.2 | Historie Androidu | 5 |
| 3.3 | Verze Android OS | 5 |
| 3.4 | Zařízení s Androidem | 7 |
| 3.5 | Struktura/Architektura systému | 7 |
| 3.6 | Struktura aplikace | 10 |
| 3.7 | Vývojová prostředí | 10 |
| 4 | Adaptivní hypermediální systémy - AHS | 11 |
| 4.1 | Komplexní adaptivní systémy | 11 |
| 4.2 | AHS | 11 |
| 4.3 | Adaptive System Framework - ASF | 11 |
| 5 | Analýza a návrh řešení | 13 |
| 5.1 | Volba vývojového prostředí | 13 |
| 5.2 | Již dostupné stejně zaměřené aplikace | 13 |
| 5.3 | Požadavky na aplikaci Inteligentní kuchařka | 16 |
| 5.4 | Případy užití | 16 |
| 6 | Realizace | 21 |
| 6.1 | Výběr API | 21 |
| 6.2 | Použité prvky | 21 |
| 6.3 | Databáze | 22 |
| 6.4 | Uživatelské rozhraní - Návrh | 23 |
| 6.5 | Třídy projektu | 23 |
| 7 | Testování | 27 |
| 7.1 | Testovací zařízení | 27 |
| 7.2 | Testování během vývoje | 27 |

| | | |
|----------|---|-----------|
| 7.3 | Závěrečné testování | 27 |
| 7.4 | Výsledky testování | 28 |
| 8 | Závěr | 29 |
| | Literatura | 31 |
| A | Android Studio | 33 |
| A.1 | Popis Android Studia | 33 |
| A.2 | Užitečné nástroje v Android Studiu | 34 |
| A.3 | Založení nového projektu v Android Studiu | 34 |
| A.4 | Nastavení Android Studia | 35 |
| B | Instalační příručka | 37 |
| C | Obsah příloženého CD | 39 |

Seznam obrázků

| | | |
|-----|---|----|
| 2.1 | Zastoupení operačních systémů | 4 |
| 3.1 | Zastoupení jednotlivých verzí Android OS | 6 |
| 3.2 | Zastoupení výrobců mobilních zařízení | 7 |
| 3.3 | Architektura Android OS | 9 |
| 4.1 | Architektura ASF | 12 |
| 5.1 | Aplikace Vaření.cz - nejlepší recepty | 14 |
| 5.2 | Aplikace Recepty doma | 15 |
| 5.3 | Aplikace Nejlepší recepty (offline) | 15 |
| 5.4 | Případy užití | 20 |
| 6.1 | Struktura databáze | 22 |
| 6.2 | Struktura tabulky Material | 22 |
| 6.3 | Příkaz na vytvoření databáze | 23 |
| 6.4 | Zobrazuje přehled receptů, detail receptu a seznam surovin | 24 |
| 6.5 | Zdrojový kód pro zobrazení surovin použitých pro přípravu pokrmu ze souboru activity_add_cooked.xml | 26 |
| 6.6 | Výsledek zpracování zdrojového kódu, uvedeného v obrázku 6.5 z pohledu uživatel | 26 |
| A.1 | Uživatelské rozhraní | 33 |
| A.2 | Výběr SDK | 35 |
| A.3 | Soubor AndroidManifest.xml | 36 |
| C.1 | Soubor AndroidManifest.xml | 39 |

SEZNAM OBRÁZKŮ

Seznam tabulek

| | | |
|-----|---------------------------------|----|
| 3.1 | Verze systému Android | 6 |
| 7.1 | Testovací zařízení | 27 |

SEZNAM TABULEK

Kapitola 1

Úvod

Mobilní telefony a tablety s pokročilým operačním systémem jsou stále rozšířenější a denně je používají miliony uživatelů. Pokročilý operační systém umožňuje současný běh více procesů, pokročilou správu operační paměti nebo instalaci aplikací třetích stran. Nejrozšířenějším operačním systémem je Android a na jeho aplikačním portále Google Play jsou miliony aplikací a mezi nimi i mnoho kuchařek. Avšak dnes běžně dostupné kuchařky pro mobilní telefony a tablety zobrazují recepty bez jakéhokoliv ohledu na dostupné množství surovin, které má uživatel během vaření k dispozici. Proto se tato bakalářská práce zabývá vývojem aplikace *Inteligentní kuchařka* s pokročilou správou databáze surovin a jejich množství. Aplikace je vyvinuta pro operační systém Android. Její hlavní výhodou je provázanost nabízených receptů s databází surovin. Aplikace dále v sobě implementuje ASF framework a SQLite databázi. Uživatel sám při nákupu nových surovin zadá jejich množství do tak zvané Spižírny. Spižírna je v podstatě databáze se surovinami a jejich množstvím. V případě již uvařeného receptu se použité množství surovin ze Spižírny odečte. Tím je zajištěna aktuálnost databáze se surovinami a díky této funkcionalitě lze předejít zjištění nedostatku surovin až v době přípravy jídla.

Kapitola 2

Mobilní operační systémy a aplikace

První kapitola se zabývá výhodami aplikace napsané přímo pro mobilní zařízení. V druhé části jsou uvedeny nejrozšířenější pokročilé mobilní operační systémy a jejich zastoupení na trhu.

2.1 Výhody nativní aplikace

Hlavní výhody aplikace napsané pro daný operační systém na rozdíl od webové služby jsou:

- Aplikace může mít data uložena v paměti zařízení, proto připojení k internetu nemusí být vyžadováno. Aplikace bude fungovat i v místech, kde není možnost přístupu k bezdrátovým sítím wifi nebo není pokrytí bezdrátovým připojením poskytovaným mobilním operátorem.
- Stejný vzhled aplikace, protože bude přizpůsobena vzhledu zbytku systému, například použití systémových grafických prvků, neměl by pro uživatele být problém se v aplikaci zorientovat. Je zaručený podobný vzhled na displejích s různým rozlišením.
- Možnost distribuovat aplikaci přes aplikační portály a následné vylepšování aplikace díky zpětné vazbě od uživatelů přes recenze.

2.2 Mobilní operační systémy

- Android

Výrobce: Open Handset Alliance

Zařízení: mobilní telefony, tablety, hodinky, televize multimedialní přehrávače a další

Datum vydání: 23. září 2008

- iOS

Výrobce: Apple Inc.

Zařízení: mobilní telefony, tablety, hodinky

Datum vydání: 29. června 2007 (ještě pod názvem iPhone OS)

- Windows Phone

Výrobce: Microsoft

Zařízení: mobilní telefony

Datum vydání: 21. října 2010

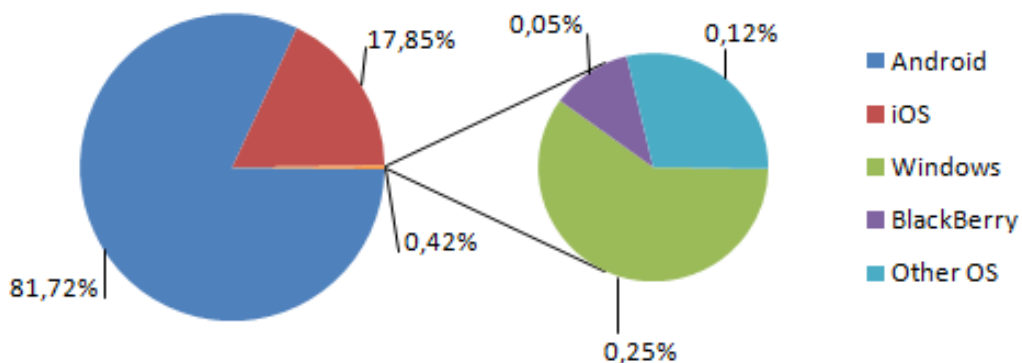
- BlackBerry 10

Výrobce: Research in Motion

Zařízení: mobilní telefony

Datum vydání: 30. ledna 2013

Následující graf 2.1 zobrazuje zastoupení jednotlivých operačních systémů prodaných koncovým uživatelům za čtvrtý kvartál roku 2016[10]. Operační systéme Android byl ke konci roku 2016 s téměř 82% nejprodávanějším mobilním operačním systémem. Druhým nejprodávanějším s necelými 18% byl operační systém iOS od společnosti Apple. A zároveň tyto dva operační systémy měly na trhu více jak 99% podíl v mobilních telefonech s pokročilým operačním systémem.



Obrázek 2.1: Zastoupení operačních systémů

Kapitola 3

Úvod do Androidu

V této kapitole je popsán operační systém Android, jeho historie a verze. Dále se kapitola zaměřuje na popis struktury operačního systému Android a také struktury aplikací. V poslední části kapitoly je popsáno vývojové prostředí Android Studio.

3.1 Definice Androidu

Android je moderní operační systém určený pro mobilní telefony, tablety, autorádia, televize a další zařízení. Android je vyvíjený konsorciem Open Handset Alliance a je dostupný jako open source (otevřený software). Systém je vývojářům přístupný zdarma, mohou ho upravovat, ale pokud provedou větší úpravy, než je povoleno licenčními podmínkami, musí změnit i jméno systému.

3.2 Historie Androidu

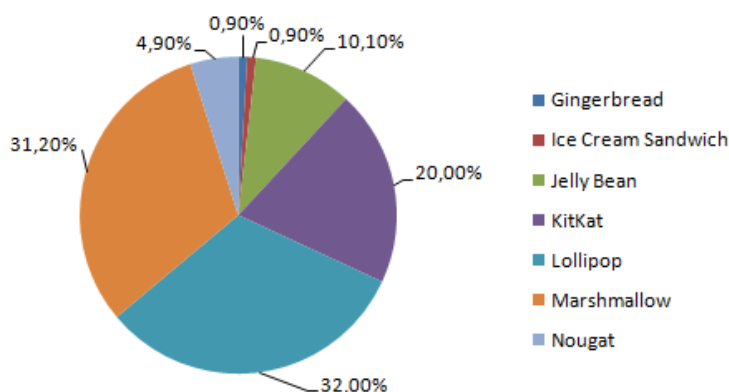
Společnost Android Inc založil Andy Rubin, Rich Miner, Nick Sears a Chris White v roce 2003, tu v roce 2005 koupila společnost Google Inc. Konsorcium Open Handset Alliance (dále jen OHA) založilo 34 firem 5. listopadu 2007, dnes je v konsorciu OHA 84 technologických společností. První mobilní telefon s operačním systémem Android byl T-Mobile G1 vyrobený společností HTC. Při uvedení na trh byl v telefonu nainstalován Android ve verzi 1.0 a uvedení proběhlo 23. září 2008. V dalších letech vydává OHA aktualizované verze, které od verze 1.5 pojmenovává podle amerických sladkých zákusků v abecedním pořadí.

3.3 Verze Android OS

První veřejně dostupný Android byl ve verzi 1.0 v zařízení T-Mobile G1. Tato verze už obsahovala aplikaci Android Market, kde byly poskytovány další aplikace. Android ve verzi 3 Honeycomb byl určen a optimalizován pro použití v tabletech. Tato verze systému Android se nikdy masově nerozšířila, proto OHA vývoj specializovaného Androidu ukončila. Od verze 4.0 probíhá vývoj pouze jedné verze Androidu, která je schopná běžet na všech typech zařízení. S verzí 4.4 KitKat byl do Androidu přidán k testování virtuální stroj Android

| Verze | Jméno | Datum vydání |
|---------|--------------------|--------------------|
| 1.5 | Cupcake | 27. dubna 2009 |
| 1.6 | Donut | 15. září 2009 |
| 2.0-2.1 | Eclair | 26. říjen 2009 |
| 2.2 | Froyo | 20. květen 2010 |
| 2.3 | Gingerbread | 6. prosince 2010 |
| 3.0-3.2 | Honeycomb | 22. února 2011 |
| 4.0 | Ice Cream Sandwich | 19. října 2011 |
| 4.1-4.3 | Jelly Bean | 9. července 2012 |
| 4.4 | KitKat | 31. října 2013 |
| 5.0-5.1 | Lollipop | 12. listopadu 2014 |
| 6.0 | Marshmallow | 5. října 2015 |
| 7.0-7.1 | Nougat | 22. srpna 2016 |

Tabulka 3.1: Verze systému Android



Obrázek 3.1: Zastoupení jednotlivých verzí Android OS

Runtime (dále jen ART), který měl vylepšit často kritizovanou rychlost systému. Od verze 5.0 Lollipop byl Dalvik¹ plně nahrazen ART [6]. S verzí 6.0 Marshmallow získal uživatel možnost instalovaným aplikacím udělovat pouze vybraná oprávnění k systémovým funkcím [9].

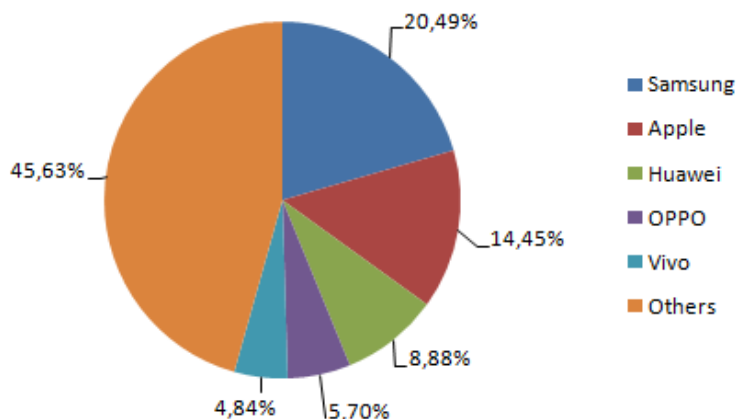
Tabulka 3.1 uvádí přehled všech dosud vydaných verzí operačního systému Android

Graf 3.1 zobrazuje zastoupení jednotlivých verzí operačního systému Android ze dne 3. 4. 2017 [11].

¹Dalvik je virtuální stroj, na kterém běží programy napsané v jazyce Java v operačním systému Android

3.4 Zařízení s Androidem

Systém Android je dnes používán v mobilních telefonech, tabletech, televizích, hodinkách, multimediálních přehrávačích, autorádiích a dalších zařízeních. Největším světovým výrobcem zařízení s Androidem je společnost Samsung viz graf 3.2.



Obrázek 3.2: Zastoupení výrobců mobilních zařízení

Graf 3.2 zobrazuje zastoupení výrobců mobilních telefonů v roce 2016.

3.5 Struktura/Architektura systému

Operační systém Android je rozdělen do pěti hlavních vrstev.

- Linuxové jádro - Linux Kernel

Základem platformy Android je jádro operačního systému Linux. Linuxové jádro se využívá zejména z důvodu poměrně snadného sestavení pro konkrétní hardware a bezpečnostních funkcí. Jádro umožňuje zbytku softwaru komunikaci s hardwarem pomocí ovladačů. Dále se jádro stará o správu paměti, síťovou komunikaci, správu procesů a další.

- Hardwarová Abstraktní Vrstva - Hardware Abstraction Layer (HAL)

Definuje standardizované rozhraní pro výrobce hardwaru a umožňuje aby Android nebyl závislý na implementaci nízkourovňových ovladačů hardwaru. Díky existenci HAL mohou výrobci hardwaru implementovat funkčnost bez nutnosti upravovat nebo rozšiřovat systém vyšší úrovně.

- Android Runtime

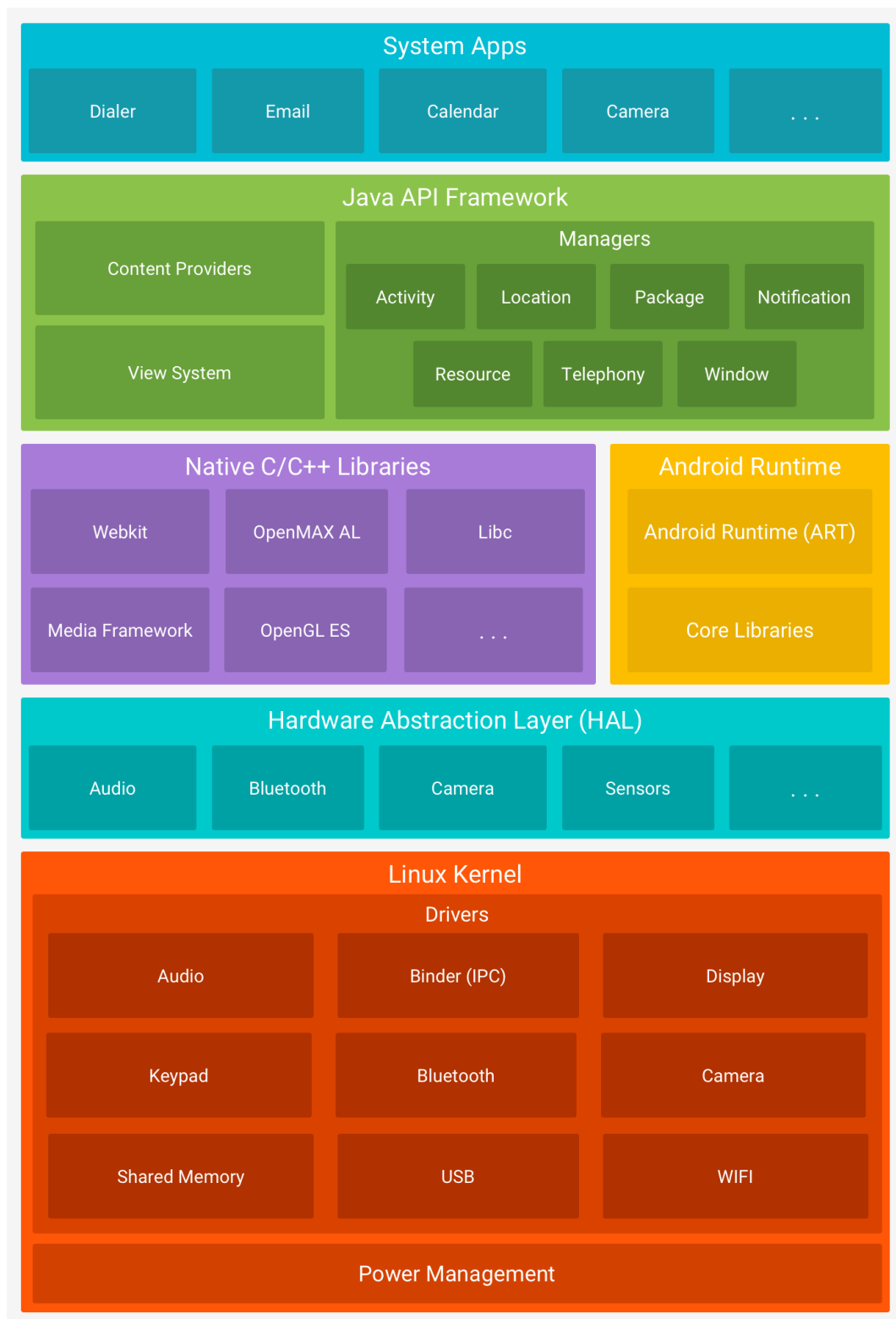
Tato vrstva obsahuje virtuální stroj, na kterém se spouštějí aplikace napsané v jazyce Java. Do verze 4.3 byl využíván Dalvik, ve verzi 4.4 bylo možné pro testovací účely zvolit místo Dalvika Android Runtime, od verze 5.0 je využíván pouze Android Runtime.

- Nativní C/C++ knihovny - Native C/C++ Libraries

V této vrstvě jsou knihovny napsané v jazyce C a C++, které jsou využívány systémovými komponentami, například ART a HAL. Funkcionalita některých knihoven z této vrstvy je přes vrstvu Java API framework poskytována systémovým a uživatelským aplikacím.

- Java API Framework

Vrstva Java API framework poskytuje pomocí API, napsaných v jazyce Java, řadu funkcí systému Android. Tato API se využívají při vytváření aplikací pro Android, kdy tento přístup usnadňuje znovupoužití komponent a služeb systému.



Obrázek 3.3: Architektura Android OS

3.6 Struktura aplikace

- Activities

Activita je základní prvek interakce mezi uživatelem a aplikací, reprezentuje jednotlivé obrazovky uživatelského rozhraní. Příklad takové aktivity je obrazovka, která uživateli v aplikaci *Inteligentní kuchařka* zobrazí přehled receptů a druhá aktivita je detail receptu.

- Services

Servisy (Services) běží na pozadí a vykonávají dlouhodobé operace nebo práci pro vzdálené procesy. Neposkytují uživatelské rozhraní. Jedná se například o přehrávání hudby, když je uživatel v jiné aplikaci, nebo načítání dat přes síť.

- Broadcast receivers

Broadcast receivers slouží k zachycení událostí, například když systém začne hlásit slabou baterii, tak některé aplikace, které mají na tento stav zareagovat, mohou ukončit nedůležité operace na pozadí.

- Content providers

Content providers je možnost, jak sdílet data mezi aplikacemi, pokud k tomu mají oprávnění.

3.7 Vývojová prostředí

- Android Studio

Android Studio je oficiální vývojářské prostředí pro programování aplikací pro operační systém Android. Bylo oznámeno 15. května 2013 na Google I/O a je založeno na vývojovém prostředí IntelliJ IDEA Community Edition od společnosti JetBrains. Android Studio ve verzi 1.0 bylo vydáno v prosinci 2014, je zdarma dostupné pro operační systémy Mac OS, Linux a Windows[4].

- NetBeans

Ve vývojovém prostředí NetBeans je možné programovat aplikace pro Android po nainstalování zásuvného modulu NBAndroid nebo Codename One.

- Eclipse

V IDE Eclipse je možné vyvíjet aplikace pro Android po doinstalování Android SDK a ADT (Android Development Tool) doplňku. Na konci roku 2015 byla ukončena oficiální podpora pro ADT doplněk a vývojářům bylo doporučeno přejít na Android Studio.[3]

Kapitola 4

Adaptivní hypermediální systémy - AHS

4.1 Komplexní adaptivní systémy

Je systém složený z mnoha nezávislých částí, které spolupracují. Tato spolupráce vede k výstupům, které je často velmi těžké, až nemožné předvídat pohledem na jednotlivé části odděleně. Tyto části se řídí jednoduchými pravidly a všechny mají stejné pravomoci, žádná jednotka nemá vedoucí úlohu[7].

4.2 AHS

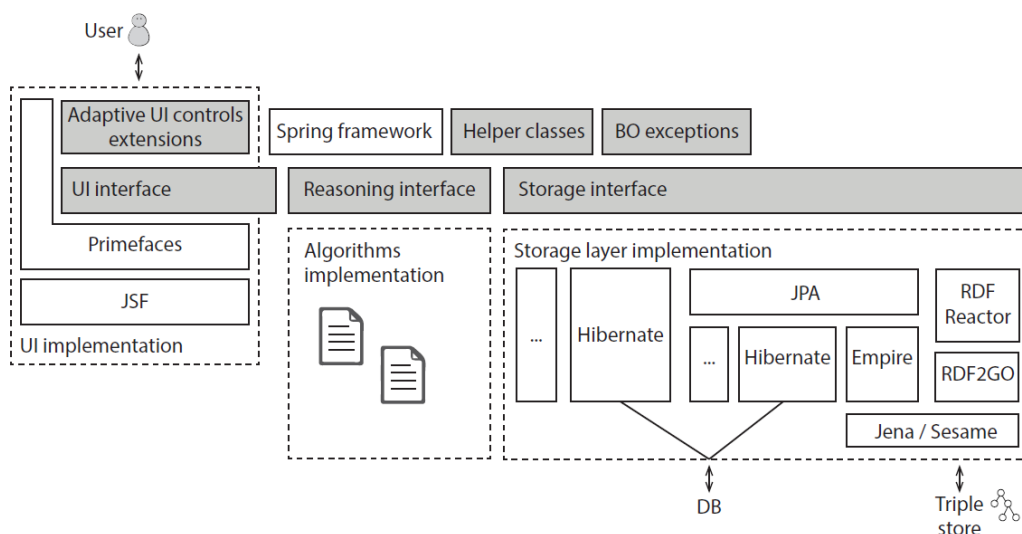
Adaptivní znamená schopný přizpůsobit se. Hypermediální systémy se dají chápat jako rozšíření konceptu hypertextu, kdy audiovizuální obsah, který se může skládat například z videa, hudby, obrazů nebo textu. Propojení do jednoho nelineárního média je mezi jednotlivými komponentami realizováno přes hyperlinky. AHS jsou schopny se přizpůsobit na základě vnějších podnětů, které jsou o uživateli zjištěny pozorováním. AHS systémy přizpůsobují výstup zobrazovaný uživateli, na základě informací, které o něm získá[2].

4.3 Adaptive System Framework - ASF

Cílem ASF je zjednodušit a zrychlit vývoj AHS. Je založen na teoretickém modelu Generic Ontological Model for Adaptive Web Environments (GOMAWE).

Struktura ASF je založena na komponentách, které jsou na sobě navzájem nezávislé a mají vlastní zodpovědnost. Model ASF je zobrazen na obrázku 4.1.

V ASF jsou data o uživateli rozdělena na dvě části, na uživatelský profil a uživatelský model. Data o uživateli se dají sbírat dvěma způsoby. První způsob je, že data o sobě sdělí sám uživatel, například vyplněním dotazníku či odpověďmi na zadané otázky. Druhou možností je monitorování uživatele. Touto cestou aplikace o uživateli získá data spojená s určitými komponentami, například jak dlouho strávil na nějaké stránce nebo jaké má znalosti o zadané problematice.



Obrázek 4.1: Architektura ASF

Přístup k profilu uživatele a jeho modelu je zajištěn přes `AdaptationManager`, ten je založen na návrhových vzorech `Singleton` a `Factory`. `Singleton` je v aplikaci použit pro poskytování instance správce. Návrhový vzor `factory` je využíván pro vytváření konkrétních modelů systému pro jednotlivé uživatele.

V ASF je dále definováno úložiště pro adaptující pravidla. Tato pravidla mění obsah aplikace za využití adaptačních algoritmů. Pravidla jsou vytvářena automaticky nebo tvůrcem obsahu.[1]

Kapitola 5

Analýza a návrh řešení

Kapitola popisuje intuitivní vývojové prostředí Android Studio, které bylo použito pro samotnou tvorbu aplikace Inteligentní kuchařka.

5.1 Volba vývojového prostředí

Cílem bakalářské práce bylo vyvinout aplikaci pro operační systém Android, proto bylo zvoleno oficiální vývojové prostředí Android Studio. Na internetu je mnoho návodů, jak s Android Studiem pracovat, veškerá důležitá nastavení, ať vývojového prostředí nebo samotného projektu, jsou velmi dobře zdokumentována a proto i začínající vývojáři nemají problém se rychle zorientovat. Mezi velmi často využívané zdroje informací patří webové stránky <http://stackoverflow.com/> a <https://developer.android.com/index.html>. Nalézají se zde řešení od úplně základních problémů, se kterými se potýkají začátečníci, až po ty velmi složité.

5.2 Již dostupné stejně zaměřené aplikace

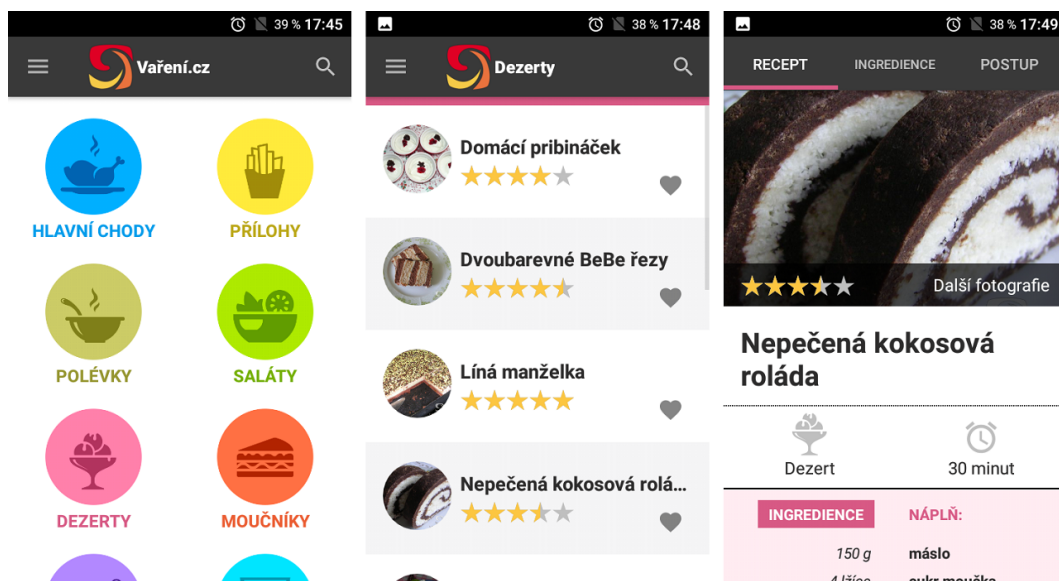
Na aplikačním portálu Google Play byly nalezeny pouze tři české offline aplikace kuchařka, které poskytují recepty bez připojení k internetu, avšak nedisponují veškerou funkcionalitou kterou nabízí aplikace vyvinutá pro tuto bakalářskou práci.

- *Vaření.cz - nejlepší recepty* Stažení mezi 100 000 - 500 000, průměrné hodnocení 4.7 z 5, je aplikace s přehledným a graficky pěkně zpracovaným uživatelským rozhraním, možností stahovat si další balíčky receptů z webového portálu www.varime.cz. Tento program ve srovnání s vyvíjenou aplikací nedisponuje následujícími funkcemi:

Nelze recepty upravovat přímo v mobilní aplikaci, ani pro lokální použití

Nelze přidávat vlastní recepty přes mobilní aplikaci

Aplikace nedisponuje databází dostupných surovin



Obrázek 5.1: Aplikace Vaření.cz - nejlepší recepty

- *Recepty doma* Stažení mezi 10 000 - 50 000, průměrné hodnocení 3.7 z 5, je aplikace s nepřehlednou úvodní obrazovkou. Dalším nevyhovujícím faktorem je rozdělení receptů do více jak třiceti kategorií. Některé funkce jsou dostupné pouze při připojení k internetu a po registraci. Největší nedostatky aplikace jsou následující:

Pro větší nabídku receptů je nutná registrace

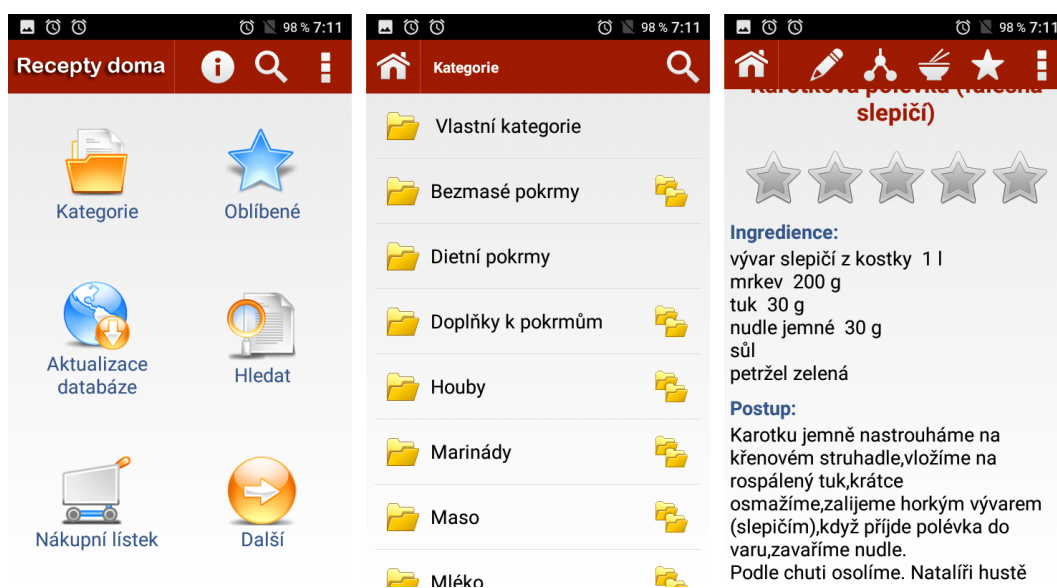
Nemá funkci vedení databáze surovin

- *Nejlepší recepty (offline)* Stažení mezi 500 - 1000, průměrné hodnocení 4.5 z 5, je velmi špatně optimalizovaná aplikace, u které musí uživatel dlouho čekat na odezvu.

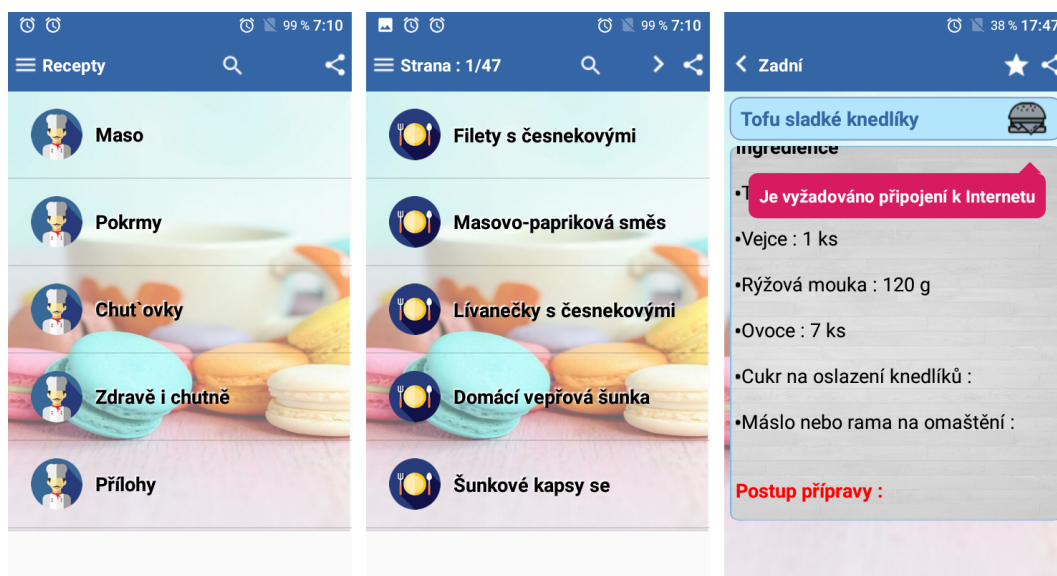
Nelogické rozdělení kategorií pokrmů

Aplikace je velmi náročná na výkon mobilního zařízení

Zobrazení fotografie pokrmu, pouze při připojení k internetu. Fotografie není součástí detailu receptu, ale se zobrazuje na nové obrazovce.



Obrázek 5.2: Aplikace Recepty doma



Obrázek 5.3: Aplikace Nejlepší recepty (offline)

5.3 Požadavky na aplikaci Inteligentní kuchařka

Hlavní výhodou aplikace Inteligentní kuchařka, oproti výše zmíněným aplikacím, je vedený seznam surovin. Uživatel sám při nákupu surovin zadá jejich množství do takzvané spížírny. Spížirna je databáze surovin a jejich množství, které má uživatel k dispozici. Databáze surovin a jejich množství je zcela ve správě uživatele. Aplikace může uživateli zobrazovat pouze recepty, pro jejichž přípravu bude mít dostatek surovin. Díky této funkcionalitě lze předejít zjištění nedostatku surovin, až v době přípravy pokrmu. V případě již uvařeného pokrmu se spotřebované množství surovin ze spížírny odečte.

Používání aplikace musí být pro uživatele intuitivní. Při spuštění aplikace si uživatel zvolí pro kolik osob bude chtít připravit pokrm. Dalším krokem je výběr kategorie, ze které si uživatel přeje připravit jídlo. Kategorie jsou následující:

- Hlavní chod
- Polévka
- Salát
- Dezert
- Předkrm
- Snídaně

Podle zvolené kategorie aplikace nabídne seznam receptů, pro které má uživatel dostatečné množství surovin. Seznam bude řazen s ohledem na to, kdy uživatel vybraná jídla naposledy připravoval. Pokud by se chtěl uživatel například inspirovat, co by chtěl vařit příště, bude mít možnost prohlédnout si všechny recepty, bez ohledu na stav dostupných surovin a podle toho si naplánovat nákup. Další funkcionalitou, dostupnou pro uživatele, bude možnost libovolně upravovat základní databázi receptů a surovin, přidávat nové recepty i suroviny. Pokud uživatel zadá, že byl recept uvařen, aplikace odečte ze seznamu surovin množství, které se pro přípravu pokrmu využilo.

5.4 Případy užití

V této kapitole budou popsány případy užití uvedené na obrázku 5.4. Případy užití (anglicky Use cases) popisují funkcionalitu poskytovanou systémem, která bude mít pro uživatele viditelný výsledek. Systém je v tomto případě vyvinutá mobilní aplikace.

- Zvolit počet porcí
 1. Uživatel zvolí možnost změnit počet porcí
 2. Systém zobrazí uživateli formulář pro změnu počtu porcí
 3. Uživatel zadá novou hodnotu
 - 4.1. IF Uživatel zvolí možnost uložit novou hodnotu

4.1.1. THEN Systém zavře formulář a uloží si novou hodnotu

4.2. ELSE Uživatel zvolí možnost zrušit zadání nové hodnoty

4.2.1 Systém zavře formulář

5. Podle počtu porcí bude systém přepočítávat potřebné suroviny pro přípravu receptů

- Vyhledat a zobrazit recept:
 1. Systém uživateli zobrazí seznam kategorií pokrmů
 2. Uživatel si vybere kategorii
 3. Systém uživateli zobrazí seznam receptů. Recepty budou vybrány s ohledem na dostatečné množství dostupných surovin v databázi.
 - 4.1. IF Uživatel najde mezi nabízenými recepty takový, který ho zaujme
 - 4.1.1. THEN Uživatel zvolí možnost prohlédnout detail receptu
 - 4.2. ELSE Uživatel zvolí možnost zobrazit všechny recepty v kategorii
 - 4.2.1. Systém zobrazí uživateli seznam všech receptů z kategorie
 - 4.2.2. IF Uživatel najde mezi nabízenými recepty takový, který ho zaujme
 - 4.2.3.1. THEN Uživatel zvolí možnost prohlédnout detail receptu
 5. Systém uživateli zobrazí detail receptu
 - Upravit recept: Možné pokračování scénáře - Vyhledat a zobrazit recept
 1. Uživatel zvolí v detailu receptu možnost upravit recept.
 2. Systém uživateli zobrazí editační formulář
 3. Uživatel v tomto formuláři může upravit jméno receptu, popis receptu a postup přípravy
 4. IF Uživatel vybere možnost odstranit vybranou surovinu
 - 4.1. THEN Systém zobrazí uživateli potvrzovací okno, zda chce opravdu zvolenou surovinu odstranit
 - 4.2. IF Uživatel vybere možnost zrušit vymazání
 - 4.2.1. THEN Systém zavře potvrzovací okno a neprovede žádnou změnu
 - 4.3. ELSE uživatel zvolí možnost vymazat surovinu
 - 4.3.1. Systém zavře potvrzovací okno a odstraní surovinu ze seznamu
 5. IF Uživatel chce smazat další surovinu
 - 5.1. THEN Pokračuje krokem 4
 6. V dalším kroku, má uživatel možnost přidat do receptu novou surovinu
 7. IF Uživatel zvolí přidání nové suroviny
 - 7.1. THEN Systém zobrazí formulář s dostupnými surovinami
 - 7.1.1. Uživatel u vybrané suroviny zadá požadované množství

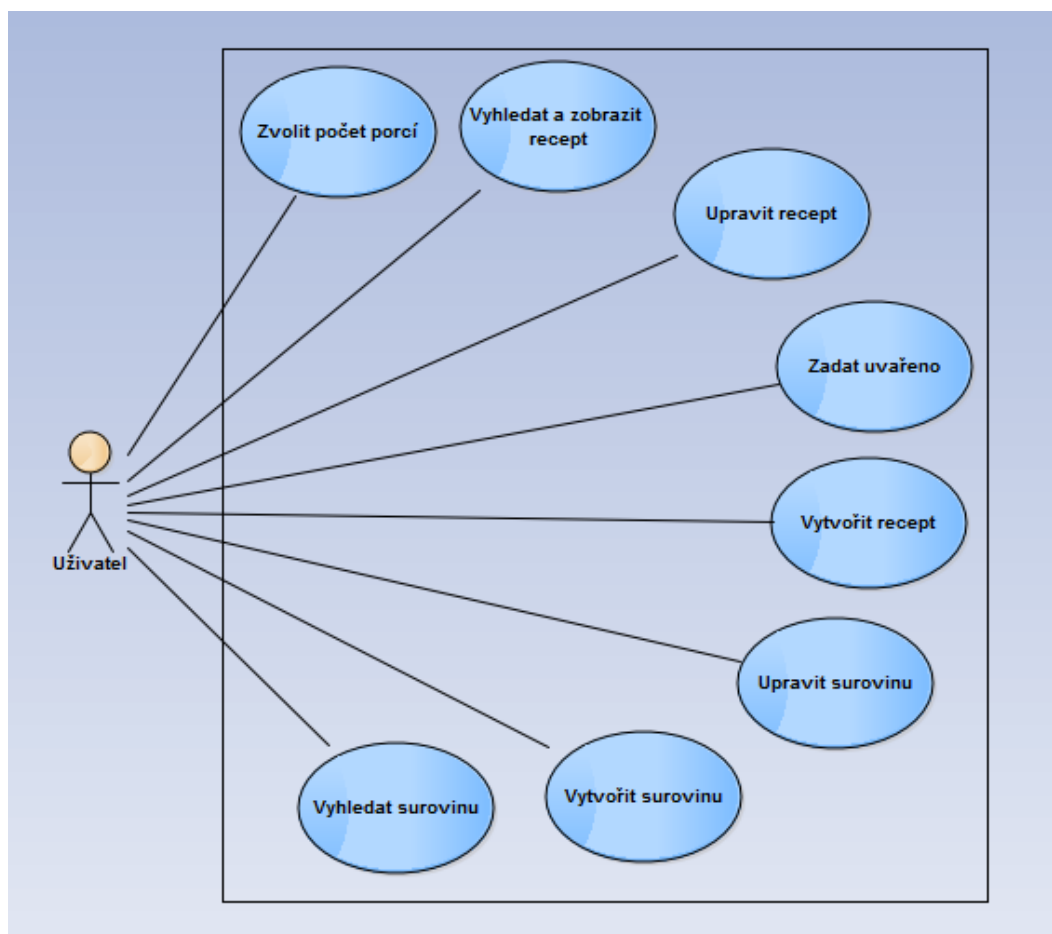
- 7.1.2. Uživatel vybere volbu přidat surovinu
- 7.1.3. Systém přidá surovinu na seznam
- 7.2. IF Uživatel chce přidat další surovinu
 - 7.2.1 THEN Uživatel pokračuje krokem 7.1.1.
- 7.3. ELSE Uživatel zvolí možnost ukončit přidávání surovin
- 8. Systém zobrazí uživateli editační formulář receptu
- 9. V dalším kroku si uživatel vybere, zda chce provedené změny uložit do databáze nebo je zrušit
 - 10. IF uživatel vybere uložit změny
 - 10.1. THEN Systém provedené změny uloží do databáze
 - 11. ELSE Systém provedené změny "zapomene"
- Zadat uvařeno: Druhé možné pokračování scénáře - Vyhledat a zobrazit recept
 - 1. Uživatel zvolí možnost uvařeno
 - 2. Systém uživateli zobrazí formulář s přehledem surovin, které jsou potřeba k přípravě receptu
 - 3. IF Uživatel vybere odstranit surovinu ze seznamu
 - 3.1. THEN Systém zobrazí potvrzovací okno
 - 3.2. IF Uživatel vybral potvrzení odstranění odstranit ze seznamu
 - 3.2.1. THEN Systém vybranou surovinu odstraní ze seznamu
 - 3.3. ELSE Uživatel vybral zrušit odstranění
 - 3.3.1 Systém zavře potvrzovací obrazovku
 - 4. ELSE Uživatel vybere přidat surovinu, kterou použil navíc
 - 4.1. Systém uživateli zobrazí seznam surovin
 - 4.2. Uživatel k vybrané surovině zadá spotřebované množství a zvolí přidat do seznamu
 - 4.3. Systém přidá surovinu seznamu spotřebovaných surovin
 - 5. IF Uživatel chce přidat další surovinu
 - 5.1. THEN Uživatel pokračuje krokem 4.2.
 - 6. ELSE Uživatel zvolí možnost ukončit přidávání surovin
 - 7. Systém zobrazí editační formulář s nově upraveným seznamem použitých surovin
 - 8. IF Uživatel použil některých surovin jiné množství, než bylo v receptu uvedeno
 - 8.1. THEN Uživatel opraví množství použitých surovin v seznamu
 - 9. IF uživatel zvolí potvrdit uvaření
 - 9.1. Systém odečte spotřebované množství surovin z databáze
 - 9.2 Systém uloží do databáze údaj, kdy byl tento recept uvařen

10. ELSE Uživatel zvolí zrušit operaci

10.1. Systém provedené změny zapomene

Další případy jsou uvedeny pouze ve zkratce.

- Vytvořit recept: Pokud bude uživatel chtít, bude mít možnost přidat libovolný recept do databáze. Při vytváření receptu, uživatel vyplní jméno receptu, popis, postup přípravy. Dalším krokem je výběr surovin z nabízeného seznamu, uživatel také musí zadat potřebné množství. Pokud by uživateli některá surovina chyběla, má možnost si ji vytvořit, až by musel zrušit vytváření receptu.
- Upravit surovinu: Uživatel bude mít možnost upravovat dostupné množství surovin.
- Vytvořit surovinu: Pokud bude uživateli určitá surovina v databázi chybět, bude mít možnost si surovinu do databáze přidat. Jako první zadá jméno suroviny, potom jaké množství chce přidat do databáze a poslední položkou jsou jednotky, na výběr má z gramů, kilogramu, mililitrů, litrů a kusů. Pak už pouze zvolí tlačítko uložit surovinu.



Obrázek 5.4: Případy užití

Kapitola 6

Realizace

6.1 Výběr API

V době vypracování této práce je k dispozici nejnovější API verze 25, s téměř 5% podílem mezi přístroji s OS Android. Pro tuto bakalářskou práci bylo zvoleno cílové API 22 minimální podporované je API 15. Minimální API verze 15 bylo vybráno z důvodu co nejširší podpory zařízení, která je přes 98% viz graf 3.1, ale také s ohledem na nové technologie, které přišly s novějšími API. Jde zejména o podporu nových grafických prvků a dalších komponent pro tvorbu uživatelského rozhraní.

6.2 Použité prvky

V bakalářské práci byly využity prvky standardních knihoven jazyka Java, které obsahuje Android Studio a také rozšiřující knihovny třetích stran popsane níže.

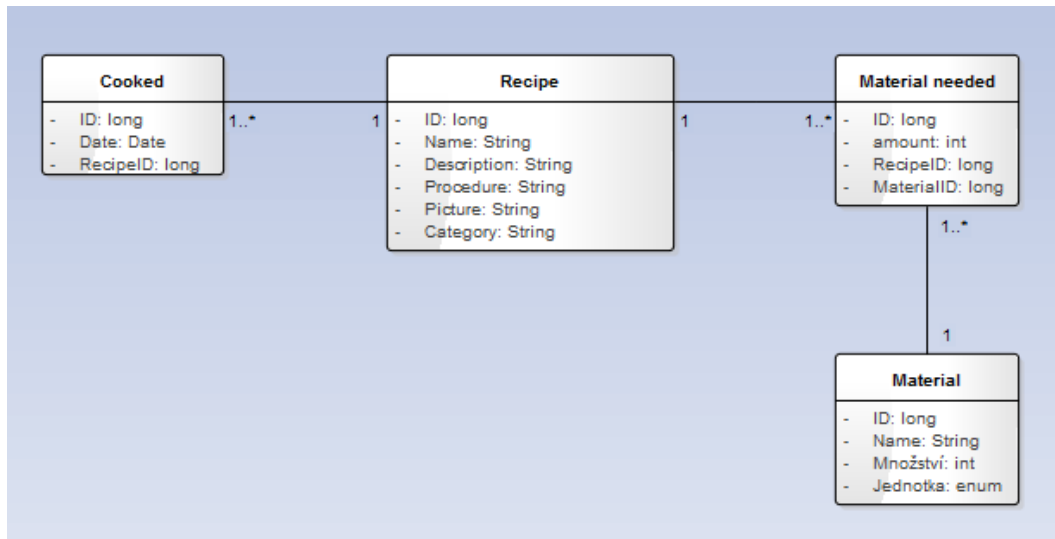
Recycler View Jedná se rozšíření klasického ListView s pokročilejšími možnostmi a celkovou optimalizací. Pro využití prvku RecyclerView je potřeba přidat do projektu knihovnu 'com.android.support.recyclerview-v7:XX.X.X'.

SQLite Je relační databázový systém, který na rozdíl od klasických databázových systému jako například MySQL, PostgreSQL neběží jako samostatná služba, ale pouze jako proces přidružený k aplikaci. SQLite většinou na disku vytvoří soubor s koncovkou .db, kde je uložena databáze. Pro snazší využití služeb SQLite jsou k projektu přidány knihovny ormlite-android-4.48.jar a ormlite-core4.48.jar

ASF Adaptive system framework, je projekt, který vznikl pod vedením pana inženýra Martina Balíka, Ph.D. na Fakultě elektrotechnické, při ČVUT v Praze. Tento projekt má za cíl vytvoření standardů použitelných v adaptivních metodách programování na různých platformách.

6.3 Databáze

S ohledem na zaměření této bakalářské práce bylo vybráno jako vhodné řešení pro ukládání dat, použití SQLite databáze, která je nenáročná na systémové prostředky a je součástí operačního systému Android. Pro usnadnění práce s databází je využito knihoven, zmíněných v předchozí kapitole. Výsledná struktura databáze je zachycena na obrázku 6.1.



Obrázek 6.1: Struktura databáze

Databáze byla přímo vygenerována v aplikaci.

```

@DatabaseTable
public class Material extends AbstractPersistable implements IUser<Long> {

    private static final long serialVersionUID = -222864131214757024L;

    @DatabaseField(columnName = "material_name", unique = true)
    private String materialName;

    @DatabaseField(columnName = "material_amount")
    private Integer materialAmount;

    @DatabaseField(columnName = "material_unit")
    private String materialUnit;
  
```

Obrázek 6.2: Struktura tabulky Material

Struktura tabulky Material je zobrazena na obrázku 6.2. Anotace @DatabaseTable určuje, že třída Material se namapuje do databáze jako tabulka se jménem material. Anotace @DatabaseField(columnName = "material_name", unique = true) říká že se v tabulce vytvoří

sloupeček, který se bude jmenovat "material_name" a hodnoty v něm uložené budou jedinečné.

```

@Override
public void onCreate(SQLiteDatabase sqLiteDatabase, ConnectionSource cs) {
    try {
        TableUtils.createTable(connectionSource, DatabaseTableConfig.fromClass(connectionSource, Recipe.class));
        TableUtils.createTable(connectionSource, DatabaseTableConfig.fromClass(connectionSource, Material.class));
        TableUtils.createTable(connectionSource, DatabaseTableConfig.fromClass(connectionSource, MaterialNeeded.class));
        TableUtils.createTable(connectionSource, DatabaseTableConfig.fromClass(connectionSource, Category.class));
        TableUtils.createTable(connectionSource, DatabaseTableConfig.fromClass(connectionSource, Tool.class));
        TableUtils.createTable(connectionSource, DatabaseTableConfig.fromClass(connectionSource, ToolRequirements.class));

    } catch (SQLException e) {
        Log.e(DatabaseHelper.class.getName(), "Unable to create databases", e);
        LOGGER.error("Unable to create databases", e);
    }
}

```

Obrázek 6.3: Příkaz na vytvoření databáze

Obrázek 6.3 ukazuje jeden z možných postupů jak vytvořit databázi na zařízeních s OS Android. Příkaz `DatabaseTableConfig.fromClass(connectionSource, Material.class)` vytvoří konfigurační data, na jejichž základě příkaz `TableUtils.createTable(connectionSource, DatabaseTableConfig.fromClass(connectionSource, Material.class))`; vytvoří databázovou tabulku pro uchování seznamu surovin a jejich množství.

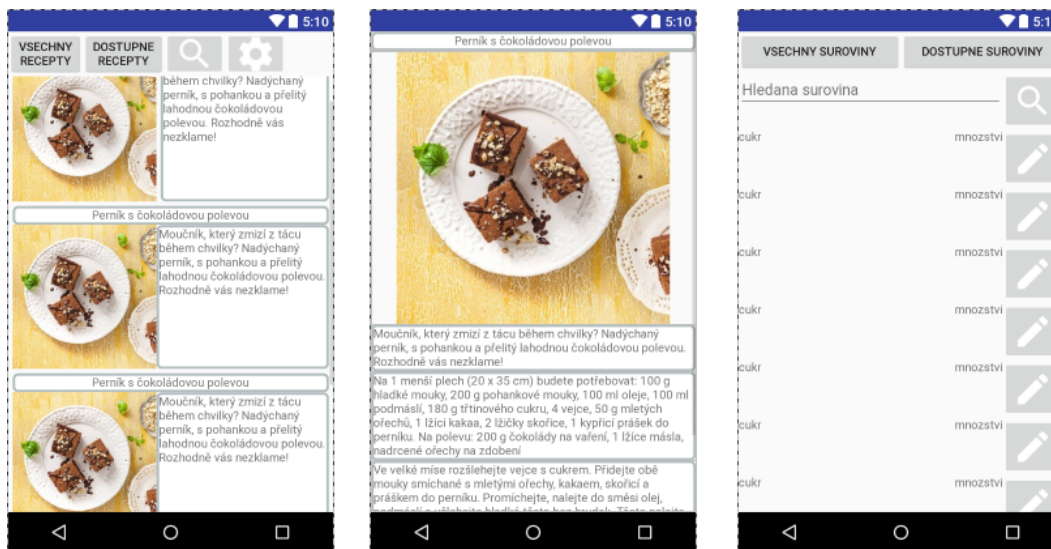
6.4 Uživatelské rozhraní - Návrh

Návrh uživatelského rozhraní vychází z analýzy již existujících řešení, funkčních požadavků a případů užití. Návrh byl vytvořen v Android Studiu, pouze z grafických komponent bez další funkčnosti, jako vzor pro vývojovou fázi. Připomínky lidí, kteří během vývoje aplikaci testovali, byly postupně zapracovávány, proto se výsledné uživatelské rozhraní bude lišit od tohoto návrhu. Návrhy tří obrazovek jsou zobrazeny na obrázku 6.4

6.5 Třídy projektu

V této kapitole bude popsána struktura projektu a hlavní komponenty. Projekt je rozdělen do tří vrstev.

- Datová vrstva je nejnižší vrstva a zajišťuje práci s databází. Jak bylo zmíněno v kapitole 6.2, je pro databázi využito nástroje SQLite.
- Aplikační vrstva je prostřední vrstvou projektu a zpracovává data, která dostane z datové vrstvy nebo je datové vrstvě předává. Na této vrstvě jsou prováděny veškeré výpočty a operace s daty. V této vrstvě probíhá většina operací s daty.
- Prezentační vrstva je nejvyšší vrstvou, která je jako jediná viditelná pro uživatele. Její funkcí je, aby data z datové vrstvy zobrazila uživateli, pomocí grafického uživatelského rozhraní a kontrolovala vstupy od uživatele. Neprobíhají zde žádné výpočty.



Obrázek 6.4: Zobrazuje přehled receptů, detail receptu a seznam surovin

Datová vrstva

Business objekty

Každá třída ze skupiny business objektů je namapována na jednu tabulku databáze. Všechny třídy jsou potomkem třídy `AbstractPersistable` a implementují rozhraní `IUser<Long>`. Ve skupině jsou tyto třídy:

- `Cooked` - třída uchovávající informace o tom kdy byl který recept uvařen. Obsahuje parametry `id`, `date`, `id_recipe`. Atribut `id` je unikátní vygenerované číslo, které jednoznačně identifikuje objekt v databázové tabulce. Pole `id` obsahují všechny business třídy a má stejnou funkčnost jako v této třídě, proto nebudou dále uváděny. Položka `date` uchovává informaci, kdy byl recept, který je uveden v položce `id_recipe` uvařen. Položka `id_recipe` uchovává odkaz na recept, ke kterému patří, v databázi, je toto pole využito jako cizí klíč odkazující na objekt v tabulce `recipe`. Položku `id_<jméno třídy>` obsahují i další třídy z business objektů, a mají stejnou funkci jako zde, odkazují na `id` objektu třídy, jejíž jméno je uvedeno za podtržítkem, proto nebudou dále vysvětlovány.
- `Material` - třída reprezentující jednotlivé suroviny a jejich množství. Její atributy jsou `id`, `name`, `amount`, `unit`. Atribut `amount` uchovává kolik dané suroviny má uživatel k dispozici. Atribut `unit` udává jednotku suroviny, zda jde o suroviny, které se váží, měří na objem nebo počítají na kusy.
- `Material Needed` - třída uchovává informace, jaké suroviny jsou potřeba v konkrétním receptu. Má tyto parametry `id`, `amount`, `id_recipe`, `id_material`. Položka `amount` udává kolik je potřeba dané suroviny pro přípravu receptu, ke kterému patří.
- `Recipe` - třída reprezentující recepty. Parametry tohoto objektu jsou `id`, `name`, `description`, `procedure`, `picture`, `category`. `Description` je popis receptu, `procedure` je postup

přípravy a v picture bude uložen název obrázku, který se pro něj bude zobrazovat. Položka category uchovává informaci, do které kategorie recept patří.

Struktura objektů a jaké jsou mezi nimi vazby je zachyceno na obrázku 6.1.

DAO objekty

Data Access Object(DAO) poskytuje aplikační vrstvě, přes veřejné metody, rozhraní pro práci s databází. Tato skupina tříd se přímo stará o komunikaci s databází. Třídy jsou potomkem AbstractDAO a implementují rozhraní ICanDelete. Každá třída z business objektů má svoji DAO třídu. Metody této třídy umožňují vytváření, čtení, mazání, editaci záznamů v databázi.

DatabaseHelper.class

Tato třída má na starosti vytvoření databáze. Pokud při prvním dotazu není ještě databáze vytvořena, zajistí její vytvoření. Její další funkcí je zajištění DAO třídám spojení s databází. Třída je potomkem OrmLiteSqliteOpenHelper třídy a je v aplikaci použita jako singleton.

Pro nahrání předpřipravené databáze do nově nainstalované aplikace byl využit script, který je umístěn v souboru DatabaseUpdate.class. Byl převzat z webového zdroje [8].

Aplikační vrstva

Servis objekty

Objekty této třídy využívají metod poskytovaných DAO objekty. Všechny ostatní třídy z aplikační nebo prezenční vrstvy komunikují s databází přes servis objekty.

Activity objekty

Activity objekty zpracovávají data, která dostala z databáze přes servis objekty. Tato data mohou předat ke zobrazení prezentační vrstvě. Každý objekt z této skupiny má na starost pouze jeden view objekt. Všechny objekty z této skupiny, musí být registrované v Android-Manifest.xml souboru. MainActivity je první aktivita, která se po spuštění aplikace zobrazí a vyvolá spuštění úvodní obrazovky.

Prezentační vrstva

Struktura UI

Základem grafického rozhraní v OS Android jsou ViewGroup, které v sobě mohou obsahovat další ViewGroup nebo View. Příkladem ViewGroup je třeba LinearLayout. LinearLayout je kontejner, do kterého se přidávají jednotlivé grafické prvky. Pro tento layout musí programátor nastavit orientaci, zda vodorovně nebo svisle, a nové komponenty se řadí postupně pod sebe nebo vedle sebe. Příkladem View je například TextView, které se využívá pro zobrazení textu. Tuto strukturu si uživatel nadefinuje v souborech <jméno souboru>.xml které jsou umístěné v `\app\src\main\res\layout`. Příklad takového souboru je uveden na obrázku 6.5 a to co uvidí uživatel na obrazovce zařízení je na obrázku 6.6


```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:paddingLeft="5dp"
    android:paddingRight="5dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:padding="5dp">

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="8"
            android:gravity="center"
            android:text="Použite suroviny"
            android:textSize="20dp" />

        <TextView
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="3"
            android:gravity="center"
            android:textSize="17dp"
            android:text="Porci" />

        <TextView
            android:id="@+id/cooked_add_text_view_servings"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="2"
            android:clickable="true"
            android:gravity="center"
            android:hint="4"
            android:textSize="20dp"
            android:onClick="editServingPopupWindow" />

    </LinearLayout>

    <android.support.v7.widget.RecyclerView
        android:id="@+id/material_cooked_list"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:scrollbars="vertical" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Extra suroviny"
        android:textSize="20dp" />

    <android.support.v7.widget.RecyclerView
        android:id="@+id/material_cooked_list_extra"
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:scrollbars="vertical" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">

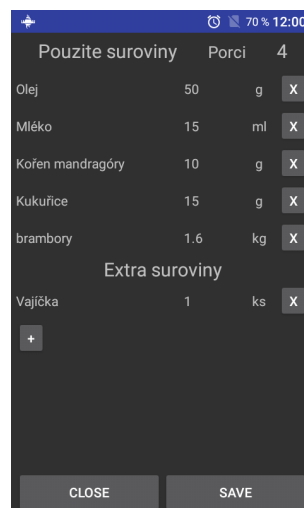
        <Button
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="closeCooked"
            android:text="Close" />

        <Button
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:onClick="saveCooked"
            android:text="Save" />

    </LinearLayout>
</LinearLayout>

```

Obrázek 6.5: Zdrojový kód pro zobrazení surovin použitých pro přípravu pokrmu ze souboru activity_add_cooked.xml



Obrázek 6.6: Výsledek zpracování zdrojového kódu, uvedeného v obrázku 6.5 z pohledu uživatele

Kapitola 7

Testování

Testování probíhalo ve dvou fázích. První byla během vývoje aplikace, kdy uživatelé vznášeli své připomínky k rozmístění ovládacích prvků, vzhledu a k některým funkcím. Druhá fáze byla testování s uživateli, kteří se neúčastnili první fáze testování. Této druhé skupině byl zadán testovací scénář, který měli projít.

7.1 Testovací zařízení

Testování probíhalo na zařízeních uvedených v tabulce [7.1](#)

| Zařízení | Rozlišení displeje (pixels) | Verze OS Android |
|----------------------|-----------------------------|------------------|
| HTC Wildfire S | 320x480 | 4.4.4 |
| Samsung Galaxy Ace 4 | 480x800 | 4.4.4 |
| Samsung Galaxy S II | 480x800 | 7.1.2 |
| Nexus 7 (2012) | 1280x800 | 5.1.1 |

Tabulka 7.1: Testovací zařízení

7.2 Testování během vývoje

Při testování použitelnosti během vývoje byly uživateli vzneseny připomínky k logické struktuře aplikace, které byly následně zapracovány do aplikace. Jedna připomínka se například týkala překrývání skupin jednotlivých kategorií jídel, kdy jedno jídlo mohlo být ve více kategoriích. Na tento podnět byla upravena databáze tak, že každé jídlo může patřit pouze do jedné kategorie. Naprogramovaný kód pro podporu více kategorií u jednoho jídla zůstal v aplikaci zachován, pokud by se v budoucnu našel důvod pro využití.

7.3 Závěrečné testování

Závěrečné testování probíhalo podle scénáře, kdy byla testovací zařízení v tabulce [7.1](#) rozdána uživatelům, kteří se neúčastnili testování během vývoje a byly jim zadány tyto úkoly:

- 1 - Změnit počet porcí, které chtějí připravit
- 2 - Otevřít seznam všech surovin, u některé změnit množství.
- 3 - Vytvořit novou surovinu a zadat množství a surovinu uložit do databáze.
- 4 - Otevřít detail libovolného receptu.
- 5 - Tento recept upravit, provést změnu, jména a postupu při vaření, odstranit libovolnou surovinu z receptu a do receptu přidat novou surovinu.
- 6 - Vytvořit nový recept a přidat do něj alespoň dvě suroviny.
- 7 - Otevřít libovolný recept a zadat, že byl uvařen.

Výsledky testování jsou následující:

- 1 - Tento krok nečinil žádnému uživateli problém.
- 2 - Děma ze čtyř uživatelů chvíli trvalo, než zjistili, které tlačítko použít na otevření menu, kde lze otevřít databázi surovin. Následný krok, úpravy suroviny, nečinil potíže žádnému uživateli. Řešením tohoto stavu může být změna ikony pro toto menu.
- 3 - Tento krok nečinil potíže žádnému uživateli.
- 4 - Tento krok nečinil potíže žádnému uživateli.
- 5 - Provést úpravy jména receptu a postupu při vaření, nečinilo problémy, žádnému uživateli. Jeden uživatel měl problém s přidáním nové suroviny, kdy si nebyl jistý, co se stalo, když zmáčkl tlačítko pro přidání suroviny. Řešením tohoto problému může být zobrazením informace, že položka byla přidána do receptu.
- 6 - Po splnění předchozích kroků tento úkol nebyl pro žádného uživatele problém.
- 7 - Tento krok nečinil potíže žádnému uživateli.

7.4 Výsledky testování

Aplikace na všech testovacích zařízeních běžela bez problému a až na drobné problémy s grafickým uživatelským rozhraním, kdy texty byly občas odskočené na nový řádek, nebyly žádným z uživatelů zaznamenány problémy s ovládním. Uživatelé po závěrečném testování navrhli, pár změn pro intuitivnější ovládním, zejména popisky na některých ovládacích prvcích, či změnu velikosti některých popisků. Protože je aplikace v neveřejném testování a není volně dostupná, byly pro testovací účely, použity recepty z webového portálu www.vareni.cz a www.prostreno.cz.

Kapitola 8

Závěr

Cílem bakalářské práce bylo vyvinout aplikaci Inteligentní kuchařka pro operační systém Android, která bude uživateli nabízet recepty, pro jejichž přípravu má dostupné suroviny. Na aplikačním portálu Google Play se nepovedlo nalézt aplikaci, která by obsahovala tuto funkcionalitu. Vyvinutá aplikace integruje databázi surovin s databází receptů. Komunikace mezi databází a datovou vrstvou probíhá s využitím komponent ASF rozhraní.

Seznam surovin a jejich dostupné množství je vedené v takzvané spižárně. Při návštěvě spižárny má uživatel možnost suroviny a jejich množství upravovat, nebo přidávat nové. Dalším požadavkem na aplikaci byla správa receptů. Recepty jsou přehledně rozděleny do šesti kategorií a uživatel může libovolný recept upravit, tak aby mu vyhovoval, nebo si vytvořit úplně nový recept. Pokud během procesu úpravy nebo vytváření receptu zjistí, že ve spižárně není surovina, kterou potřebuje, může si ji do spižárny přidat, aniž by musel přerušit vytváření nebo úpravu receptu. Další funkcionalitou integrovanou do této aplikace je automatická správa surovin ve spižárně. Pokud uživatel zvolí, že byl vybraný recept připraven, potřebné množství surovin se automaticky odečte ze spižárny pro zadaný počet porcí. Aplikace může uživateli zobrazit pouze recepty, pro jejichž přípravu má dostatek surovin, nebo pokud by se chtěl například při nákupu inspirovat co má přidat do košíku, může si nechat zobrazit všechny recepty.

Testování aplikace probíhalo od raných fází vývoje a připomínky uživatelů byli postupně zapracovány do výsledné aplikace. V závěrečné fázi bylo provedeno testování se skupinou uživatelů a s různými přístroji s odlišnými verzemi OS Android, kteří o aplikaci věděli pouze základní informace, ale neměli s ní žádnou zkušenost. Z tohoto testování vzešlo několik podnětů pro budoucí vylepšení aplikace.

Literatura

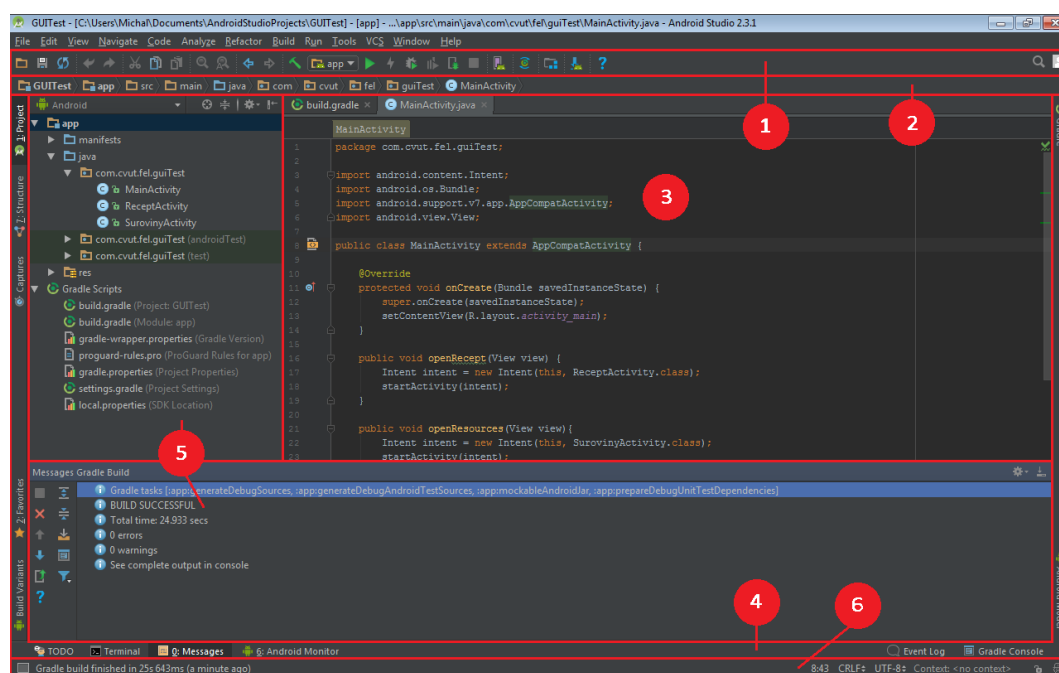
- [1] I. Balík, M.; Jelínek. Adaptive system framework: A way to a simple development of adaptive hypermedia systems. *ADAPTIVE 2013 : The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications*, pages 20–25, 2013.
- [2] M. Raszková. *Výukové adaptivní hypermediální systémy*. Mendelova zemědělská a lesnická univerzita v Brně. Provozne ekonomická fakulta, Brno, 2008.
- [3] Ukončení podpory pro adt doplnku do eclipse ide.
<https://android-developers.googleblog.com/2015/06/an-update-on-eclipse-android-developers.html>, stav z 15. 6. 2015.
- [4] Oznámení nového vývojového prostředí android studio.
<https://android-developers.googleblog.com/2013/05/android-at-google-io-2013-keynote-wrapup.html>, stav z 15. 5. 2013.
- [5] Popis vývojového prostředí android studio.
https://developer.android.com/studio/intro/index.html#the_user_interface, stav z 15. 5. 2013.
- [6] Co je nového v Android 5.0 Lollipop.
<https://liliputing.com/2014/10/whats-new-android-5-0-lollipop.html>, stav ze 15. 10. 2014.
- [7] Komplexní adaptivní systémy.
https://code.org/curriculum/science/files/CS_in_Science_Background_papers.pdf/, stav z 26. 5. 2017.
- [8] Načtení připravené databáze do nově nainstalované aplikace.
<https://blog.reigndesign.com/blog/using-your-own-sqlite-database-in-android-applications/>, stav z 26. 5. 2017.
- [9] Co je nového v Android 6.0 Marshmallow.
<http://www.trustedreviews.com/opinions/android-m-features>, stav ze 5. 10. 2015.
- [10] Prodej mobilních telefonů v roce 2016.
<http://www.gartner.com/newsroom/id/3609817>, stav z 15. 2. 2017.
- [11] Zastoupení jednotlivých verzí android os.
<https://developer.android.com/about/dashboards/index.html>, stav ze 3. 4. 2017.

Příloha A

Android Studio

A.1 Popis Android Studia

Android studio se skládá z několika logických celků zobrazených na obrázku A.1. [5]



Obrázek A.1: Uživatelské rozhraní

- 1 - Panel nástrojů umožňuje uživateli provádět celou řadu funkcí, včetně spuštění aplikace.
- 2 - Navigační lišta umožňuje uživateli procházet strukturou projektu a otevírat soubory pro úpravu.

- 3 - V editačním okně se vytváří a upravuje kód, v závislosti na otevřeném okně se může změnit editační okno.
- 4 - Panel nástrojů se spouští kolem okna IDE a umožňuje zavírat a otvírat jednotlivá okna nástrojů.
- 5 - Okna nástrojů umožňují přístup k úlohám, například řízení projektů, verzování a další.
- 6 - Stavový řádek zobrazuje stav projektu, varování nebo jiné zprávy.

A.2 Užitečné nástroje v Android Studiu

Android studio obsahuje řadu podpůrných nástrojů pro vývoj a testování aplikací. Mezi tyto nástroje patří například:

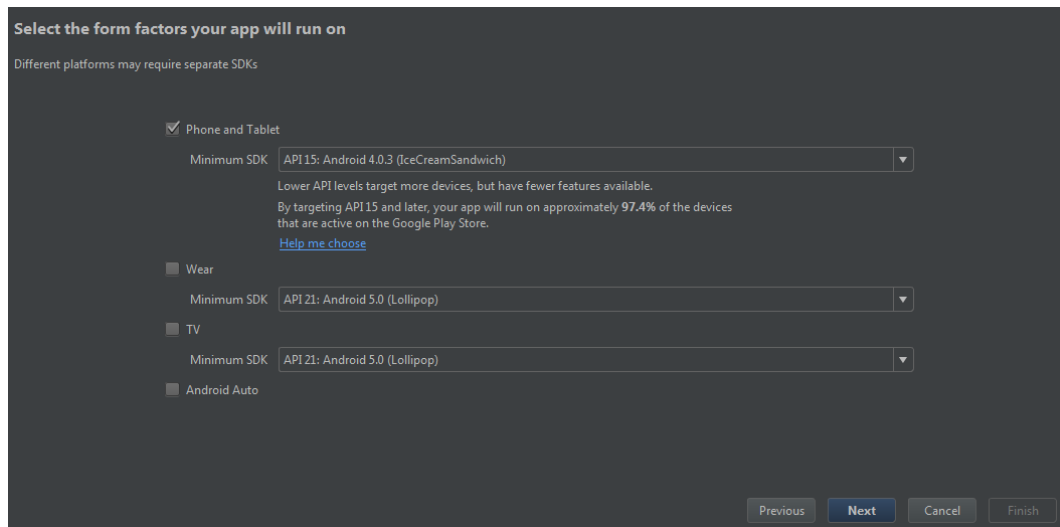
- Android Virtual Device (AVD) Manager umožňuje definovat telefony, tablety, televize nebo nositelnou elektroniku (například hodinky) s operačním systémem Android, které je možné spouštět v Android emulátoru.
- Android Emulátor napodobuje všechny softwarové a hardwarové funkce reálných zařízení s operačním systémem Android, jedinou výjimkou je uskutečňování telefonních hovorů. Emulátor umožňuje vývojářům testovat aplikace na různých zařízeních bez toho, aby dané zařízení museli vlastnit.
- Android monitor umožňuje sledovat prostředky, které aplikace vyžaduje po zařízení, na kterém je spuštěna, ať se jedná o reálné nebo virtuální zařízení, spuštěné v Android Emulátoru. Komponenta dovoluje sledovat využití operační paměti, procesoru a grafického procesoru, zaznamenávat zprávy definované uživatelem nebo systémem, dále nabízí u fyzických zařízení sledovat síťový provoz.

A.3 Založení nového projektu v Android Studiu

Založením projektu uživatele provede v IDE Android Studio grafický průvodce. V prvním kroku uživatel zadá jméno projektu, doménu společnosti a umístění projektu. Doména společnosti se používá k sestavení jména balíčku. Jméno balíčku je unikátní identifikátor aplikace v aplikačním portálu Google Play.

V druhém kroku si uživatel vybere na jakých zařízeních půjde aplikace spustit a nejstarší SDK, která bude aplikace podporovat viz obrázek A.2. Software Development kit (SDK) je sada nástrojů, která může být využita pro vývoj aplikace zaměřené na specifickou platformu. SDK obsahuje nástroje, dokumentaci, ukázkové kódy, knihovny, které pomáhají programátorům ve vývoji aplikací.

Dalším krokem je výběr Activity. Na výběr je z dvanácti předpřipravených Activit a volba bez Activity, kde je prvotní nastavení ponecháno uživateli. Vybranou Activitu Android Studio nastaví jako výchozí pro nový projekt. Toto nastavení lze kdykoliv později změnit v souboru `AndroidManifest.xml`.



Obrázek A.2: Výběr SDK

Posledním krokem je pojmenování nové Activity a k ní přiřazenému souboru xml. V tomto xml souboru je definováno rozmístění grafických prvků na obrazovce zařízení. Ve výchozím nastavení je Activity pojmenována MainActivity, ale je plně na rozhodnutí uživatele, jak ji pojmenuje. Přiřazený soubor je pojmenován `activity_main.xml` a opět lze změnit pojmenování.

Po vytvoření projektu se zobrazí uživatelské rozhraní popsané v kapitole [A.1](#).

A.4 Nastavení Android Studia

Na obrázku [A.3](#) je konfigurační soubor projektu v Android Studiu. Položka `android:icon="@mipmap/ic_launcher"` udává, jakou bude mít aplikace ikonu. Další položka `android:label="@string/app_name"` určuje, jak se bude aplikace jmenovat. Položky na řádcích 11 - 17 definují, že aktivita `cz.cvut.fel.cookbook.MainActivity` bude první aktivita, která se načte po spuštění aplikace a další doplňující informace. Posledními položkami jsou registrované activity. Každá activity, která bude programem využita, musí být na tom to místě uvedena. Pokud zde nebude zaregistrována, dojde při pokusu o její spuštění k chybě a následnému pádu aplikace.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="cz.cvut.fel.cookbook">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Cookbook"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name="cz.cvut.fel.cookbook.MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".RecipeDetailActivity"></activity>
        <activity android:name=".RecipeAddActivity"></activity>
        <activity android:name=".MaterialListActivity"></activity>
        <activity android:name=".MaterialCreateActivity"></activity>
        <activity android:name=".RecipeUpdateActivity"></activity>
        <activity android:name=".RecipeAddMaterialActivity"></activity>
        <activity android:name=".RecipeNewMaterialActivity"></activity>
        <activity android:name=".CookedAddActivity"></activity>
    </application>
</manifest>
```

Obrázek A.3: Soubor AndroidManifest.xml

Příloha B

Instalační příručka

Instalace

Souborapk umístíme na do mobilního zařízení, do kterého aplikaci chceme nainstalovat. V nastavení zařízení musíme nejdříve povolit položku *Neznámé zdroje*. V zařízení zvolíme Nastavení → Zabezpečení → Neznámé zdroje. Ve druhém kroku najdeme přes správce souborů, kam jsme apk soubor umístili. Když na soubor klikneme, začne instalace. Po dokončení instalace, se v menu aplikací objeví ikona aplikace Kuchařka. Aplikaci můžeme používat.

Příloha C

Obsah přiloženého CD

```
CD-CookBook-app
    .gitignore
    gradle
    gradlew
    gradlew.bat
    settings.gradle
LaTeX-hyphen
    k336_thesis_macros
    Koutny-thesis.pdf
    Koutny-thesis
    Makefile
    reference
cookBook.apk
```

Obrázek C.1: Soubor AndroidManifest.xml