

**Univerzita Hradec Králové**  
**Fakulta informatiky a managementu**  
**Katedra informatiky a kvantitativních metod**

**Mobilní aplikace - IS pro správu a rezervaci inzerce**

Bakalářská práce

Autor: Jakub Beneš  
Studijní obor: Aplikovaná informatika

Vedoucí práce: Malý Filip, doc. Ing. Ph.D.

Hradec Králové

duben 2017

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 18.4.2017

.....  
Jakub Beneš

#### Poděkování:

Děkuji vedoucímu bakalářské práce Ing. Filipovi Malému, doc. Ph.D. za metodické vedené práce a cenné rady. Dále děkuji Karlovi Naimanovi Ph.D. za možnost podílet se na praktickém projektu. V neposlední řadě patří mé poděkování rodině a přátelům za obrovskou podporu.

## **Anotace**

Tato bakalářská práce se zabývá vývojem mobilních aplikací. Cílem bakalářské práce je poskytnout čtenáři kvalitní informace o mobilních aplikacích a o způsobech vývoje, které nám dnešní technologie umožňují.

V druhé části práce je zobrazen postup vývoje mobilní aplikace pro platformu Android. Tato aplikace byla vyvinuta autorem práce pro společnost Lineup Systems Ltd. a slouží jako mobilní verze již existujícího inzertního systému. Bakalářská práce obsahuje zadání aplikace, analýzu v podobě vývojových diagramů a ukázky kódů. Zároveň je v práci zobrazen postup výroby uživatelského rozhraní mobilní aplikace v programu Android Studio.

## **Annotation**

**Title:** Mobile Applications - IS for management and reservation of advertisings

Bachelor thesis deals with the development of mobile applications. The aim of the thesis is to provide readers quality informations on mobile applications and informations about ways of development that allow us today's technology.

The second part shows the progress of the development of mobile application for the Android platform. Application was developed by author of this bachelor thesis for company named Lineup Systems Ltd. and serves like mobile version of existing advertisement system. The bachelor thesis contains description of application, analysis, diagrams and code examples. This bachelor thesis also shows process of creating user interface of mobile application using Android Studio.

# Obsah

|       |                                    |    |
|-------|------------------------------------|----|
| 1     | Úvod.....                          | 1  |
| 1.1   | Cíl práce a metodika.....          | 1  |
| 1.2   | Současné mobilní telefony.....     | 2  |
| 2     | Operační systém.....               | 4  |
| 2.1   | Mobilní operační systém.....       | 5  |
| 2.1.1 | Android.....                       | 6  |
| 2.1.2 | iOS.....                           | 7  |
| 2.1.3 | Windows Phone.....                 | 7  |
| 3     | Mobilní aplikace.....              | 9  |
| 3.1   | Vývoj mobilních aplikací.....      | 9  |
| 3.1.1 | Nativní aplikace.....              | 10 |
| 3.1.2 | Webové aplikace.....               | 10 |
| 3.1.3 | Hybridní aplikace.....             | 11 |
| 4     | Objektově orientovaný přístup..... | 12 |
| 5     | Použitelné technologie.....        | 14 |
| 5.1   | Nativní technologie.....           | 14 |
| 5.1.1 | Java.....                          | 14 |
| 5.1.2 | Objective-C.....                   | 15 |
| 5.1.3 | C#.....                            | 15 |
| 5.2   | Webové technologie.....            | 16 |
| 5.2.1 | HTML.....                          | 16 |
| 5.2.2 | CSS.....                           | 17 |
| 5.2.3 | Javascript.....                    | 18 |
| 5.2.4 | Bootstrap.....                     | 20 |

|       |  |    |
|-------|--|----|
| 5.2.5 | ASP.NET.....                                       | 20 |
| 5.2.6 | PHP.....   | 21 |
| 5.3   | Multiplatformní mobilní frameworky.....            | 21 |
| 5.3.1 | PhoneGap.....                                      | 22 |
| 5.3.2 | Sencha Touch .....                                 | 23 |
| 5.3.3 | Xamarin .....                                      | 23 |
| 5.3.4 | Unity 3D.....                                      | 23 |
| 6     | Vývoj aplikace pro platformu Android v praxi ..... | 24 |
| 6.1   | Zadání .....                                       | 24 |
| 6.2   | Použité technologie .....                          | 25 |
| 6.3   | Naplánování vývoje .....                           | 25 |
| 6.4   | Instalace.....                                     | 26 |
| 6.4.1 | Nette .....  | 26 |
| 6.4.2 | MySQL.....   | 26 |
| 6.4.3 | Java.....  | 27 |
| 6.4.4 | Android Studio.....                                | 27 |
| 6.5   | Návrh databáze .....                               | 28 |
| 6.5.1 | Vytvoření tabulek.....                             | 29 |
| 6.5.2 | Naplnění tabulek daty.....                         | 32 |
| 6.6   | Serverová část aplikace.....                       | 33 |
| 6.6.1 | Application Programming Interface.....             | 33 |
| 6.7   | Vzhled aplikace .....                              | 34 |
| 6.8   | Aktivity .....                                     | 37 |
| 6.9   | Testování .....                                    | 40 |
|       | Shrnutí výsledků .....                             | 41 |
|       | Závěr .....  | 42 |

|                          |    |
|--------------------------|----|
| Seznam zkratk.....       | 43 |
| Seznam obrázků .....     | 48 |
| Seznam tabulek.....      | 49 |
| Seznam ukázek kódu ..... | 50 |



# 1 Úvod

Žijeme v době, kdy mít mobilní telefon, již neznamena pouze volat a psát textové zprávy. Internet, sociální sítě, GPS<sup>1</sup>, filmy, hudba, knihy nebo hry jsou dnes vybavením každého chytrého telefonu.

Již není nutné na autobusovém nádraží hledat časy odjezdů spojů v nepřehledných jízdnicích řádech. Lidé již nemusí prohledávat, každou kapsu kabátu a zjišťovat, kam vlastně zmizel jejich papírek s nákupem, nebo každé ráno navštěvovat novinový stánek, aby si mohli přečíst, jak předešlý den dopadl jejich oblíbený sportovní tým.

Chytré telefony nahradily obrovské množství starých způsobů a přinesly do našich životů jednoduchost a pohodu. Jak ale takovéto mobilní aplikace vznikají? Jaké jsou možnosti při vývoji mobilních aplikací. Tyto a další podobné otázky budou zodpovězeny v této bakalářské práci.

## 1.1 Cíl práce a metodika

V bakalářské práci je popsán vzhled dnešních mobilních telefonů a mobilní operační systémy, které jsou nejvíce využívány. Dále jsou vysvětleny způsoby vývoje mobilních aplikací a popsány konkrétní technologie, které mohou být při vývoji aplikací použity.

Mimo představení technologií, se práce zabývá i jejím praktickým využitím na konkrétním projektu.

V praktické části této bakalářské práce je představen postup vývoje mobilní aplikace pro správu inzerce. Aplikace je mobilní verzí již existujícího informačního systému a byla napsána v nativním jazyce pro platformu Android.

---

<sup>1</sup> Globální polohový systém, s jehož pomocí lze určit geografickou polohu přijímače

## 1.2 Současné mobilní telefony

Od roku 1992, kdy byl představen první smartphone<sup>2</sup> (tehdy se mu tak ještě neříkalo) s názvem Simon od společnosti IBM, uběhala již spousta času. [1] Neustálý vývoj nových technologií a konkurence zapříčinili, že všechny dnešní mobilní telefony vypadají téměř totožně i přestože tu máme ohromné množství výrobců mobilních telefonů.

První, kdo přišel s dnešní podobou chytrých telefonů, které poháněl operační systém s možností instalovat další aplikace přes speciální obchody, byla společnost Apple v roce 2007, když v lednu představila svůj smartphone iPhone.

Do iPhonu bylo možné instalovat aplikace pouze po jejich stažení z internetu, což na první pohled není příliš elegantní a přitažlivá vlastnost, ale to není důvod, proč byl iPhone tak unikátním zařízením. Zařízení od firmy Apple se vyznačovalo naprosto unikátním vzhledem. Na přední straně telefonu se nacházelo pouze jedno tlačítko (stejně jako dnes), které sloužilo pro návrat na hlavní obrazovku. Po stranách telefonu byly ještě tři další. Dvě pro ovládání hlasitosti a jedno, které sloužilo k zapínání a vypínání telefonu, nebo k přechodu do úsporného režimu. iPhone byl výjimečný i co se týče uživatelského prostředí operačního systému, které se stalo inspirací pro konkurenční telefony. [1]

Od té doby se podoba mobilních telefonů příliš nezměnila. Co se ale změnilo, je velikost. Původní telefon od firmy Apple měl rozměry 115 x 61 x 11,6 mm s velikostí displeje 3,5 palce, který zabíral 52 % celkové velikosti telefonu. Nejnovější model iPhone ze září roku 2016 má rozměry 158,2 x 7,9 x 7,3 mm s velikostí displeje 5,5 palce, který zabírá 67,7 % celkové velikosti telefonu. [2] Tento telefon ovšem zdaleka není největším zařízením. V tabulce níže jsou porovnány zajímavé rozměrové údaje o telefonech za poslední roky.

---

<sup>2</sup> Neboli chytrý telefon, je mobilní telefon, který používá pokročilý operační systém, díky kterému můžou uživatelé využívat velké množství pokročilých funkcí

| <b>Rok</b> | <b>Telefon</b>         | <b>Rozměry (mm)</b> | <b>Velikost displeje</b> |
|------------|------------------------|---------------------|--------------------------|
| 2007       | Apple iPhone           | 115 x 61 x 11,6     | 3,5 palce (52 %)         |
| 2014       | Sony Xperia Z1 Compact | 127 x 64,9 x 9,5    | 4,3 palce (61,8 %)       |
| 2016       | Samsung Galaxy S7 edge | 150,9 x 72,6 x 7,7  | 5,5 palce (76,1 %)       |
| 2016       | Apple iPhone 7         | 138,3 x 67,1 x 7,1  | 4,7 palce (65,6%)        |
| 2016       | Apple iPhone 7 Plus    | 158,2 x 77,9 x 7,3  | 5,5 palce (67,7 %)       |
| 2016       | Huawei Mate 9          | 156,9 x 78,9 x 7,9  | 5,9 palce (77,5 %)       |
| 2017       | Samsung Galaxy S8+     | 159,5 x 73,4 x 8,1  | 6,2 palce (84 %)         |

**Tabulka 1 Porovnání velikostí telefonů za poslední roky [2]**

## 2 Operační systém

V předchozí kapitole byly zmíněni někteří velcí výrobci mobilních telefonů. Pokud ale pomineme výrobce, můžeme dále mobilní telefony rozdělit do skupin podle druhu operačního systému.

Operační systém je základní softwarové vybavení počítače, které je spuštěno při jeho zapnutí a ukončeno až při jeho vypnutí. Dnešní počítače by se bez operačního systému jen stěží obešly, jelikož plní základní úkoly jako:

- Rozpoznávání uživatelských vstupů např. z klávesnice nebo myši
- Zobrazuje výsledky na monitoru
- Umožňuje využívat další periferie<sup>3</sup> jako jsou např. tiskárny
- Umožňuje spouštět další programy
- Umožňuje spravovat soubory na disku

Operačních systémů existuje velké množství, necelých 1300 [3] a jsou rozdělovány podle jejich použití. Některé jsou využívány na osobních počítačích pro domácnosti, kde uživatelé uvítají příjemné uživatelské rozhraní a jednoduchost nainstalovat a používat další programy. Jiné operační systémy najdou své místo na serverových počítačích, kde naopak uživatelské rozhraní hraje jen velmi malou roli a důraz je kladen na rychlost a efektivitu.

Mezi oblíbené systémy patří komerční systém Windows od firmy Microsoft. V roce 2015 vyšla poslední verze tohoto systému pro osobní počítače s označením Windows 10. Firma Microsoft se zaměřuje i na počítače serverové, poslední verze systému Windows Server vyšla v roce 2016.

Dalšími velice oblíbenými systémy jsou systémy Linuxové, které se vyznačují tím, že jsou dostupné zdarma a je možné dále upravovat jejich zdrojové kódy. Základem každého takového systému je linuxové jádro, které je dále podle druhu systému

---

<sup>3</sup> Zařízení, které lze připojit k počítači. Zpravidla slouží ke vstupu a výstupu dat do/z počítače.

rozšířeno o další programy. Mezi nejznámější linuxové distribuce lze zařadit Ubuntu, Debian, Fedoru nebo např. Slackware.

Jedním z dalších velice rozšířených operačních systémů, je systém macOS od americké firmy Apple. Tento systém je unikátní svojí nekompatibilitou s jinými druhy počítačů než s těmi od firmy Apple. Systém je plně přizpůsoben počítačům Macintosh, které firma vyrábí. Zajímavostí je, že firma Apple verze systému do roku 2012 pojmenovávala po kočkovitých šelmách, později přešla na geografická místa. Nejnovější verze systému vyšla na podzim roku 2016 a nese název macOS Sierra.

## **2.1 Mobilní operační systém**

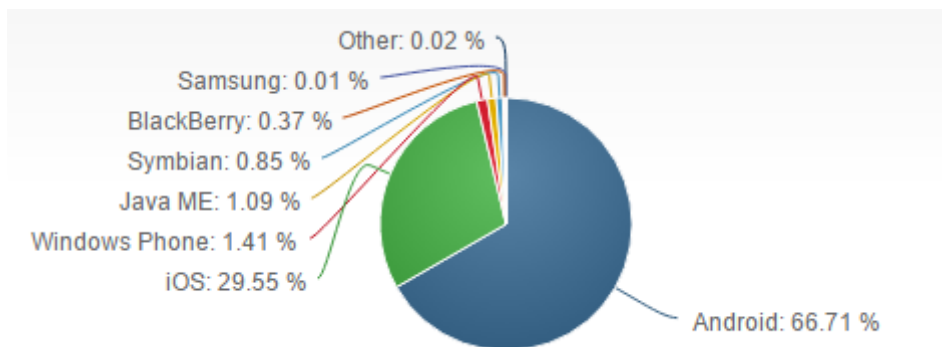
Mobilní operační systém je speciální systém určený pro zařízení jako jsou mobilní telefony, tablety, chytré televize nebo chytré hodinky. Systém je speciálně upraven pro potřeby těchto zařízení. Stejně jako typické počítačové operační systémy, nám i tyto mobilní umožňují ovládat zařízení nebo používat hardwarové vybavení<sup>4</sup> jako je fotoaparát, videokamera, GPS<sup>5</sup> nebo Wi-Fi<sup>6</sup>. Rovněž je přizpůsobeno uživatelské rozhraní systémů. Displej některých mobilních telefonů může být malý a je tedy nutné zařídit, aby i na těchto zařízeních byl uživateli poskytnut maximální komfort v ovládání. Samostatnou kapitolou jsou chytré hodinky, kde je uživatelské rozhraní extrémně omezováno velikostí displeje.

Mobilních operačních systémů opět existuje více, stejně jako je tomu u systémů počítačových. Mezi nejznámější patří Android od společnosti Google, iOS od firmy Apple a Windows Phone od firmy Microsoft. V grafu níže je znázorněn podíl trhu mobilních operačních systémů z února roku 2017.

---

<sup>4</sup> Veškeré fyzické vybavení počítače

<sup>6</sup> Označení pro bezdrátovou komunikaci v počítačových sítích



Obrázek 1 Podíl trhu mobilních operačních systémů z února roku 2017 [4]

### 2.1.1 Android

Android je open-source<sup>7</sup> mobilní operační systém, který je vyvíjen od roku 2003. V roce 2005 byl odkoupen od americké společnosti Google. [5]

Android dnes najdeme na většině mobilních telefonů a tabletů, ale můžeme se s ním setkat i na jiných elektronických zařízeních, jako jsou chytré hodinky nebo televize.

Důvodem, proč je Android nejvíce rozšířený je, že výrobci mobilních telefonů, nemusí Googlu za použití systému platit, tím se snižuje cena telefonu.

Android je vysoce přizpůsobitelný, lze měnit tapety, témata nebo použít různé aplikace, které uživatelské rozhraní telefonu změní k nepoznání. Android dovoluje propojení s Google účtem a dává tak možnost uživatelům zálohovat kontakty, fotky a vlastně veškerá data. Výhodou je i velikost cloudového úložiště<sup>8</sup>. Google nabízí svým uživatelům zdarma 15 GB volného místa na jejich data.

Aplikace jsou distribuovány skrze Google Play, kde se v březnu roku 2017, nacházelo 2,8 miliónů aplikací. [6] Většinu z nich je možné stáhnout zdarma. Cena placených aplikací se pohybuje přibližně od 15 Kč do přibližně 300 Kč.

---

<sup>7</sup> Jedná se o označení softwaru, jehož zdrojový kód je veřejně dostupný. Vývojáři ho mohou dále upravovat a vylepšovat.

<sup>8</sup> Jedná se o datové úložiště, které je poskytováno přes internet. K takovému úložišti mají uživatelé přístup odkudkoliv, kde se zrovna mohou připojit k internetu.

Android prošel mnoha verzemi a má v oblibě je pojmenovávat podle jídla. Poslední verzí systému je Android 6.0 Marshmallow, který vyšel v roce 2015.

### **2.1.2 iOS**

Jedná se o mobilní operační systém vyvinutý firmou Apple, který se nachází pouze na zařízeních této společnosti. Lze ho tedy nalézt na iPhoneu, iPadu, iPodu Touch nebo chytrých hodinkách Apple Watch.

První verze systému iOS byla představena v lednu roku 2007 společně s revolučním telefonem iPhone. Tato verze systému se s těmi dnešními nedá srovnávat, obsahovala pouze předinstalované aplikace bez možnosti stáhnout nové. Tato nepříjemnost byla ovšem rychle vyřešena. [7]

Jelikož je systém iOS vyvíjen přímo pro produkty firmy Apple, dosahují tyto zařízení obrovského výkonu, který bývá mnohdy větší, než mají hardwarově silnější zařízení s jiným operačním systémem.

Aplikace lze stahovat skrze obchod App Store, ve kterém se v březnu roku 2017 nacházelo 2,2 miliónů aplikací. [6] Ceny jsou podobné jako u konkurence a rovněž je zde možné nalézt aplikace zdarma.

### **2.1.3 Windows Phone**

V roce 2000 firma Microsoft představila svůj první mobilní systém Windows Mobile, někdy také označovaný jako Pocket PC 2000, který napodoboval vzhled počítačového operačního systému Windows 98. Vývoj systému pokračoval až do roku 2010, kdy byl ve verzi Windows Mobile 6.5.5 ukončen. [8]

Ve stejném roce Microsoft představil systém nový, který nesl název Windows Phone 7. Tento systém nebyl zpětně kompatibilní s Windows Mobile a vyznačoval se novým uživatelským rozhraním Metro.

Nyní se Windows Phone nachází ve verzi Windows Phone 10. Někdy se také označuje pouze jako Windows 10.

Aplikace jsou distribuovány skrze Windows Store, který obsahuje znatelně méně aplikací než konkurence. V březnu roku 2017 se v obchodě nacházelo necelých 700 000 aplikací. [6]



### 3 Mobilní aplikace

Mobilní aplikace je software navrhnutý speciálně pro mobilní zařízení. Jsou distribuovány přes speciální obchody, které se podle druhu platformy liší. Mezi největší takovéto obchody patří Google Play, kde se nachází aplikace pro systém Android. V obchodu App Store, je možné stahovat obsah pro systém iOS a do systému Windows Phone je možné nainstalovat aplikace přes obchod Microsoft Store.

Obvykle, při zakoupení chytrého telefonu nebo např. tabletu, již má zařízení předinstalovanou sadu základních aplikací. Pod jednoduchou aplikací je možné si představit kalendář, e-mailový klient nebo webový prohlížeč. Aplikace mohou využívat hardwarovou výbavu telefonu jako je Wi-Fi, GPS, fotoaparát či kameru.

V březnu roku 2017 se v obchodech nacházelo celkem přes 6,4 milionu aplikací. [6] Je možné je rozdělit do několika kategorií. Nejvíce zastoupeným druhem aplikací jsou hry. [9] Hry mohou být velice jednoduché, příliš nenáročné na výkon, ale najdeme zde i tituly, které plně využijí výkon silnějších zařízení. Mezi další kategorie je možné zařadit business aplikace, výukové, sociální sítě, různé přehrávače hudby nebo videí, sportovní aplikace a mnoho dalších.

Uživatelské rozhraní je navrženo tak, aby vyhovovalo velikosti zařízení. Ve většině případů je totiž možné aplikaci nainstalovat na více druhů zařízení zároveň. Uživatelé potom mohou např. svoji oblíbenou hru hrát v pohodlí domova na větším displeji tabletu a zároveň v autobuse cestou do práce na mobilním telefonu.

#### 3.1 Vývoj mobilních aplikací

Existuje několik cest, kterými je možné se při vývoji mobilní aplikace vydat. Vývojářovo rozhodnutí obvykle ovlivní druh aplikace, kterou se chystá vytvořit. Jak již bylo napsáno výše, existuje mnoho druhů aplikací. Pokud je vyvíjena aplikace náročnější na výkon, je vhodné sáhnout po jiném způsobu vývoje než u té, která například jen zobrazuje data.

Dalším faktorem, který může rozhodnutí o vývoji ovlivnit, je rozpočet. V případě, že má být aplikace spustitelná na více platformách, je vhodné sáhnout po

multiplatformním vývoji. Programovat aplikaci, pro každou platformu zvlášť si vyžaduje více znalostí, programátorů, času a tím samozřejmě i peněz.

Aplikace můžeme rozdělit do několika druhů, podle toho, jakým způsobem byly vytvářeny. Jedná se o nativní, hybridní a webové aplikace.

### **3.1.1 Nativní aplikace**

Tyto aplikace jsou vyvíjeny speciálně pro určitý druh operačního systému. Například to tedy znamená, že nativní aplikaci pro Android není možné spustit na zařízení, které pohání operační systém iOS. Nativní mobilní aplikace dokáží maximálně využít možnosti telefonu a výkonově předbíhají ostatní druhy.

Jejich velická nevýhoda spočívá právě ve zmíněných mobilních platformách, jelikož pro každou z nich, je nutné vyvíjet aplikaci v jiném programovacím jazyce. To sebou nese hned další nevýhody. Více programovacích jazyků znamená více znalostí, popřípadě programátorů. Budoucí vývoj, aktualizace, nové funkce, vše je nutné dělat pro každou platformu zvlášť.

Vývoj nativní aplikace pro Android vyžaduje znalost programovacího jazyka Java. Pro iOS je nutné ovládat jazyk Objective-C a aplikace pro Windows Phone, jsou napsány v jazyce C#.

### **3.1.2 Webové aplikace**

Webové aplikace jsou celkem jednoduché, ale přesto funkční řešení. Jedná se v podstatě o klasickou webovou stránku, která je spuštěna na mobilním zařízení. Jejich obrovskou výhodou je, že je možné je vytvořit pomocí typických webových technologií, kterými jsou HTML, CSS a Javascript. Poté stačí jen uložit odkaz na webovou stránku na pracovní plochu mobilního zařízení. V tu chvíli se web chová jako mobilní aplikace a funguje na všech mobilních platformách. Takováto stránka musí být responzivní, aby se její vzhled upravil při vstupu z mobilního zařízení, jinak by bylo velice nepříjemné s takovouto aplikací pracovat.

Samozřejmě v tomto způsobu vývoje lze nalézt i zásadní nevýhody. Takto vytvořené aplikace nemohou využívat hardware telefonu jako je fotoaparát nebo např. GPS. Další nevýhodou je, že bez přístupu k internetu uživatel zároveň ztrácí přístup

k aplikaci. Webové aplikace není možné distribuovat skrze obchody jako je Google Play, což může být silnou nevýhodou, pokud vývojáři cílí na velké množství uživatelů. V případě firemní aplikace je tato skutečnost zanedbatelná.

### **3.1.3 Hybridní aplikace**

Tento druh aplikací kombinuje oba předešlé druhy. Pro vývoj jsou využity webové technologie a výsledná aplikace může být nainstalována stejně jako nativní, s možností využívat hardwarovou výbavu zařízení. Aplikace je možné rovněž distribuovat skrze obchody aplikací.

Tento druh aplikace se může jevit jako dokonalý. Přináší jednotný vývoj pro všechny platformy a zároveň dovoluje využívat veškeré možnosti zařízení. Přesto lze nalézt jisté nevýhody. Hybridní aplikace nedosahují výkonů jako nativní. Pokud se tedy vývojáři chystají pracovat s 3D grafikou, vytvářet složitou hru, nahrávat či upravovat videa nebo fotografie je vhodnější sáhnout po aplikaci nativní.

## 4 Objektově orientovaný přístup

Objektově orientovaný přístup je způsob myšlení, které je využíváno při vývoji softwaru. Je založeno na abstrakci reálného světa. Na svět je pohlíženo jako na skupinu objektů, které mezi sebou komunikují, ukládají stavy a poskytují si navzájem různé operace.

Základním stavebním prvkem jsou objekty, které odpovídají entitám z reálného světa. Může jít například o konkrétní osobu nebo o automobil určité značky a barvy. Dalším velice důležitých pojmem je třída, která je určitým vzorem, podle kterého se objekty vytvářejí. Definuje jejich vlastnosti a schopnosti vykonávat určité operace.  
[10]

Pro každý objekt, jsou zaváděny čtyři charakteristické aspekty:

- Jedinečnost – Každý objekt je jedinečný, i když jeho vlastnosti mohou být totožné. Pro rozlišení takovýchto objektů se používají jednoznačná identifikační čísla.
- Zatřiditelnost – Objekty se stejnými vlastnostmi a chování jsou seskupovány do tříd.
- Mnohotvárnost – Stejné operace se můžou chovat rozdílně pro různé třídy objektů.
- Dědičnost – Jedná se o sdílení vlastností a chování mezi třídami. Každá třída může být potomkem třídy jiné. V takovém případě zahrnuje vlastnosti své nadřazené třídy, které může obohatit o vlastnosti nové.

Objektově orientovaný přístup je založen na principech, které vyjadřují jeho podstatu:

- Zobecnění – Při pohledu na objekty z reálného světa jsou vnímány pouze jejich podstatné vlastnosti.
- Zapouzdření – Objekty poskytují jiným objektům rozhraní, přes které poskytují přístup jen k vlastnostem a operacím, které si to vyžadují. Ostatní vlastnosti a operace jsou vnějšímu okolí skryty. Díky tomu může být

implementace objektů změněna, aniž by se změna dotkla dalších částí systému, kde je objekt použit.

## 5 Použitelné technologie

Technologií, které je možné použít při vývoji mobilní aplikace je mnoho. Níže jsou popsány technologie, které jsou nejvíce využívány. V kategorii webových technologií, lze nalézt i programovací jazyky, pomocí kterých jsou vyvíjeny serverové části aplikací. Mobilní aplikace serverové části využívají pro získávání potřebných dat.

### 5.1 Nativní technologie

Nativní technologie, nejsou příliš vhodné pro tvorbu multiplatformní mobilní aplikace. Vývoj za pomoci těchto technologií je složitější a zabere vývojářům více času. Naopak nám dávají možnosti, které nám jiné technologie nabídnout nemohou.

#### 5.1.1 Java

Java je objektově orientovaný programovací jazyk<sup>9</sup>, vytvořen firmou Sun Microsystems a představen v roce 1995. [11] Současným vlastníkem je firma Oracle, která jazyk získala v roce 2010.

Syntaxe Javy vychází z jazyka C++, ale je značně zjednodušený. Programátor se například díky automatické správě paměti, nemusí starat o její vyčištění, po již nepotřebných objektech. Java dále přináší jednoduchost při zápisu dat do souborů, nebo při jejich přenášení po síti. Bez použití dalších knihoven<sup>10</sup>, umožňuje tvořit uživatelské rozhraní, pracovat se soubory nebo s databází. [11]

Jedná se o jeden z nejpoužívanějších jazyků ve světě [12], a to hlavně díky tomu, že není závislý na architektuře počítače nebo jiného zařízení. Právě díky této skutečnosti je možné pomocí jazyka Javy, vyvíjet aplikace cílené na mobilní zařízení.

Z hlediska mobilních aplikací je Java používána při vývoji pro platformu Android.

---

<sup>9</sup> Programovací jazyk, používaný při objektovém programování. Cílem objektového programování je vytvářet malé, relativně samostatné a v různých aplikacích opakovaně použitelné jednotky (objekty)

<sup>10</sup> Soubor funkcí a procedur, který může být sdílen více počítačovými programy

```
public class AhojSvete {  
    public static void main(String[] args) {  
        System.out.println("Ahoj světe!");  
    }  
}
```

**Ukázka kódu 1 Jednoduchý program v jazyce Java [Zdroj: Autor práce]**

### 5.1.2 Objective-C

Jak již název napovídá, Objective-C je objektově orientovaný programovací jazyk založen na jazyce C. Byl vyvinut již v roce 1980, ale svoji popularitu získal až s vzestupem operačních systémů od firmy Apple. [13]

Objective-C může působit netradičně pro svou syntaxi. Namísto běžného volání metod, se kterým se programátor setkává u většiny objektově orientovaných programovacích jazyků, se zde posílají zprávy. Tento model zasílání zpráv byl převzat z programovacího jazyka Smalltalk.

```
// většina objektově orientovaných jazyků  
objekt.metoda();  
  
// v Objective-C jsou namísto běžného volání metod, posílány zprávy  
[příjemce zpráva];
```

**Ukázka kódu 2 Příklad posílání zpráv v jazyce Objective-C [Zdroj: autor práce]**

Stejně jako u Javy, ani zde se programátor nemusí starat o správu paměti. S příchodem iOS 4.0, Apple představil koncept nazývaný ARC, který toto dovoluje. Přesto je nutné, aby programátor věděl, kdy je paměť zabírána a uvolňována, aby se vyhnul komplikacím při vývoji aplikace. [13]

Objective-C je nativním jazykem při vývoji aplikací pro systém iOS.

### 5.1.3 C#

C# je dalším objektově orientovaným jazykem, který byl vytvořen firmou Microsoft. Jazyk je neoddělitelně spjat s platformou Microsoft .NET, která byla poprvé uvedena v roce 2000. Platforma řeší problémy týkající se vzájemné spolupráce jednotlivých programů a jejich integraci do větších celků.

Microsoft C# založil na jazycích C++ a Java. Syntaxí se pak jazyku Java velmi podobá, čemuž napovídá i fakt, že měl být jejím přímým konkurentem. [14]

Pomocí jazyka C# lze vytvářet webové, desktopové a nativní mobilní aplikace pro operační systém Windows Phone.

```
public class AhojSvete{  
    public static void Main(){  
        System.Console.WriteLine("Ahoj světe!");  
    }  
}
```

**Ukázka kódu 3 Jednoduchý program v jazyce C# [Zdroj: Auto práce]**

## 5.2 Webové technologie

Webové technologie jsou primárně určeny pro tvorbu webových stránek, ale v posledních letech jsou značně využívány i při tvorbě mobilních aplikací. V této kapitole lze nalézt popis technologií, které jsou označovány jako front-end technologie. Lze mezi ně zařadit HTML, CSS a Javascript.

Kapitola je rozšířena i o technologie označovány jako back-end technologie, mezi které můžeme zařadit ASP.NET nebo PHP. Nelze je použít přímo k tvorbě mobilních aplikací, ale můžeme s jejich pomocí vytvořit serverovou část, s kterou bude mobilní aplikace komunikovat, a pomocí které bude získávat potřebná data.

### 5.2.1 HTML

HTML je značkovací jazyk používaný primárně pro tvorbu webových stránek, jehož první definice se objevila v roce 1991. [15] HTML používá definované značky, někdy také elementy, pro vytváření různých prvků na stránce. Jak byl jazyk vyvíjen, přidávaly se do definice další značky, které dávaly programátorům stále více možností, jak webové stránky tvořit.

V roce 1999 byla vydána verze 4.01 a bylo oznámeno, že se jazyk nebude již dále vyvíjet. [15] Nástupcem jazyka HTML mělo být XHTML, které přineslo například uzavírání značek, nutnost psát značky malými písmeny nebo povinně vyplňovat hodnoty atributů.



V roce 2014 byla vydána nová verze HTML5, která je postavena na verzi 4.01, obohacená o zvyklosti z XHTML. [15] Dále HTML5 přineslo nové značky, které mnohem lépe popisují určité části webové stránky nebo značky multimediální či grafické.

Nové značky ve verzi HTML5.

```
<!-- Sématické značky -->
<header></header>, <footer></footer>, <article></article>,
<section></section>
<!-- Grafické značky -->
<svg></svg>, <canvas></canvas>
<!-- Multimediální značky -->
<audio></audio>, <video></video>
```

**Ukázka kódu 4 Ukázka nových značek ve verzi HTML5 [Zdroj: autor práce]**

## 5.2.2 CSS

Pomocí HTML je tvořen obsah webových stránek. Aby ale vývojáři dosáhli požadovaného vzhledu webu, je nutné obsah naformátovat. To je možné udělat dvěma způsoby.

Formátovat obsah lze pomocí HTML tagů, ale jejich použití dává jen velmi málo možností.

```
<b>Tučný text</b>
<u>Podtržený text</u>
<i>Kurzíva</i>
```

**Ukázka kódu 5 Možnosti základního formátování textu v HTML [Zdroj: autor práce]**

Mnohem více možností, jak webové stránky formátovat dávají programátorům takzvané kaskádové styly neboli CSS. První verze CSS byla vydána v roce 1996 a dnes se již nachází ve verzi CSS 3. [16] Tato verze přinesla mnoho novinek, které z jazyka CSS dělají opravdu silný nástroj pro tvorbu webových stránek.

Jednou z prvních velice užitečných novinek je možnost vybírat elementy na stránce pomocí pokročilých selektorů.

```
/* Vybere element, jehož atribut „name“ obsahuje slovo „ABC“ */  
E[name*="ABC"]  
  
/* Vybere element, jehož atribut „name“ začíná slovem „ABC“ */  
E[name^="ABC"]  
  
/* Vybere element, jehož atribut „name“ končí slovem „ABC“ */  
E[name$="ABC"]
```

#### Ukázka kódu 6 Pokročilé selektory v CSS3 [17]

Pomocí CSS3 je dále možné jednoduše tvořit přechody barev, zaoblené rohy rámečků, přidávat vlastní fonty, používat transformace a posuny elementů nebo vytvářet animace, které bylo dříve nutné dělat složitými způsoby pomocí Javascriptu.

### 5.2.3 Javascript

Javascript je objektově orientovaný programovací jazyk, který vykonává svoji práci na straně klienta, tedy v jeho webovém prohlížeči. V oblasti tvorby webových stránek najde své uplatnění v mnoha ohledech. Díky Javascriptu je možné webovou stránku rozpohybovat, nabízí se například tvorba dynamického menu, skrývání a odkrývání obsahu, zvětšování obrázků nebo zobrazování modálních oken. Lze ho ale využít i k složitějším úlohám jako je validace formulářů před jejich odesláním.

```
document.getElementById("pozdrav").innerHTML = "Ahoj světe!";
```

#### Ukázka kódu 7 Vložení textu do vybraného elementu podle jednoznačného identifikátoru [Zdroj: autor práce]

Javascript dále najde uplatnění v takzvaných Single page aplikacích, které jsou z největší části napsané právě v tomto jazyce. Jedná se o klasickou webovou stránku, která používá server pouze jako uložisko dat. Javascript poté data zobrazuje uživateli. Výhodou je, že se uživateli vždy překresluje jen část webové stránky, u které je to požadováno, což má kladný dopad na rychlost a celkově práce s takovouto aplikací je pro uživatele mnohem příjemnější.

O tom, že je Javascript opravdu silným nástrojem svědčí fakt, že je díky němu možné vyvíjet i desktopové aplikace. Tuto možnost vývojářům dává framework Electron, který využili vývojáři známých aplikací jako je Slack, Visual Studio Code nebo Atom.

### 5.2.3.1 jQuery

V souvislosti s Javascriptem je dobré zmínit pomocnou knihovnu jQuery, která je hojně používána. Jedná se o malou a rychlou knihovnu s jednoduchou syntaxí, která řeší problémy s nekompatibilitou mezi prohlížeči. Programátorovi dále ulehčuje práci v mnoha ohledech. Pomocí selektorů, které jsou podobné jako selektory CSS, lze lehce odkazovat na HTML elementy. Pomocí jQuery lze také lehce obsluhovat události jako je kliknutí na element, změna nějaké hodnoty apod.

```
/* Použití jednoduchého selektoru */  
  
$("#prvekFormulare").val();  
  
/* Kód reagující na změnu hodnoty v textovém poli */  
$("#textovePole").on("change", function(e) {  
    // Místo pro kód  
});
```

**Ukázka kódu 8 Použití jQuery [Zdroj: autor práce]**

### 5.2.3.2 AJAX

Další výhodou použití jQuery, je jednoduchost implementace ajaxového volání. Pomocí tzv. AJAXU, je možné asynchronně načítat pouze požadované části stránek, v závislosti na odpovědi ze serveru, bez nutnosti obnovení celé stránky.

```
$.ajax({  
    url: "/api/nactiPocasi",  
    data: {  
        mesto: "Hradec Králové"  
    },  
    success: function(result) {  
        $("#pocasiTrutnov").html(result.stupne);  
    }  
});
```

**Ukázka kódu 9 Načtení aktuální teploty v Hradci Králové [Zdroj: autor práce]**

### 5.2.3.3 JSON

Také je vhodné zmínit JSON, což je jednoduchý lehce čitelný textový formát pro přenos dat, který je často používán právě při asynchronním načítání částí stránek.

```
[
  {
    id: 1,
    jmeno: "Jakub",
    prijmeni: "Beneš"
  },
  {
    id: 2,
    jmeno: "Jan",
    prijmeni: "Novák"
  }
]
```

**Ukázka kódu 10 Formát JSON [Zdroj: autor práce]**

### **5.2.4 Bootstrap**

V dnešní době, kdy uživatelé na webové stránky přichází z různých zařízení, je nutné, aby byly tyto stránky tzv. responzivní. Taková stránka musí upravit svůj obsah tak, aby byl přizpůsoben velikosti displeje zařízení, na kterém je prohlížena. Právě tuto problematiku vývojářům pomáhá řešit framework Bootstrap.

Bootstrap je zdarma dostupná a nejoblíbenější knihovna pro tvorbu responzivních webů. Pomocí CSS tříd, které programátor přiřadí HTML elementům na stránce, již sama zařídí, aby byl obsah upraven při zobrazení na mobilním zařízení. Bootstrap obsahuje sady tříd pro různé prvky stránky jako jsou blokové elementy, formuláře, tlačítka, modální okna, menu a mnoho dalších.

### **5.2.5 ASP.NET**

ASP.NET je webový framework, který umožňuje tvorbu webových aplikací v jazyce C#. Je vytvořen firmou Microsoft a je nástupcem frameworku ASP.

Framework řeší za vývojáře velké množství základních problémů jako je bezpečnost, autentifikace uživatele, práce s databází nebo správu formulářů. [18]

S tvorbou webových stránek pomocí tohoto frameworku se setkáme spíše v korporátní sféře. Vlastníci malých webových portálů nebo běžných osobních stránek, se mu vyhýbají z důvodu vyšší ceny za hosting.

## 5.2.6 PHP

PHP je serverový skriptovací jazyk, který vznikl v roce 1995 a jedná se o velmi silný nástroj používaný pro tvorbu dynamických webů. Veškeré skripty jsou prováděny na straně serveru a uživateli jsou následně pouze zobrazovány výsledky operací.

PHP má velice jednoduchou syntaxi a stal se tak oblíbeným nástrojem pro tvorbu malých i velkých webových projektů.

```
<php  
echo "Ahoj světe!";  
>
```

### Ukázka kódu 11 Jednoduché použití jazyka PHP [Zdroj: autor práce]

Způsobů, jak přistupovat k programování v jazyce PHP je několik. Jazyk programátorům dává možnost kombinovat PHP přímo s elementy HTML. To může být vhodné u jednoduchých stránek. Pokud je ale tvořen složitější web, například internetový obchod, stal by se takovýto přístup velice nepřehledný a jeho rozšiřitelnost a celková správa by byla velmi složitá. Z tohoto důvodu se v praxi často PHP skripty a HTML stránky oddělují.

Pro programování v jazyce PHP byla vytvořena celá řada frameworků, díky kterým se vývojáři nemusí starat o celou řadu základních problémů jako je bezpečnost, rotování nebo o integraci dalších moderních technologií a pluginů. Mezi takovéto frameworky patří Laravel, Symphony2 nebo Nette od českých vývojářů.

## 5.3 Multiplatformní mobilní frameworky

V této kapitole budou popsány frameworky, používané pro vývoj hybridních mobilních aplikací. Takovýchto frameworků existuje obrovské množství, zde jsou popsány nejvíce používané.

Výběr správného frameworku pro vývoj konkrétní aplikace, opět ovlivní její zaměření. Některé jsou vhodné pro vývoj mobilních her, jiné kladou důraz na bezpečnost dat. Rozdílů je spousta a je jen na vývojáři, jaký framework je pro jeho aplikaci nejvhodnější.

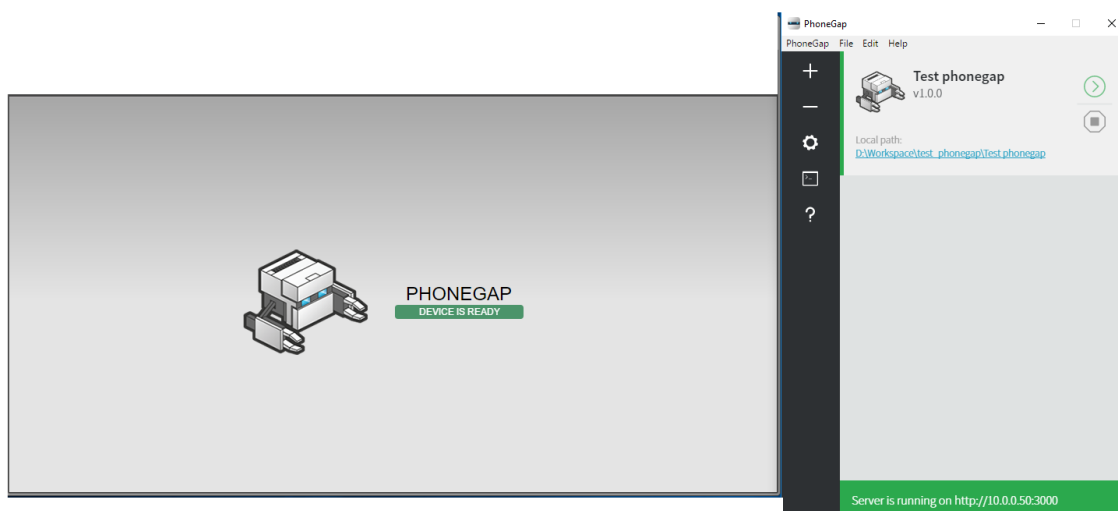
### 5.3.1 PhoneGap

PhoneGap je open source framework vlastní společnosti Adobe a založený na open source projektu zvaném Apache Cordova. [19] Umožňuje vývojářům tvořit mobilní aplikace pomocí HTML5, CSS a Javascriptu pro iOS, Android i Windows Phone.

Adobe také poskytuje cloudovou službu, která sama zkompiluje nahraný webový projekt. Službu je možné získat v několika licencích. Za placenou verzi vývojáři zaplatí 10 dolarů měsíčně a získají tak možnost vytvořit 25 privátních aplikací. Velikost jedné aplikace může být maximálně 100 MB. Za měsíční členství v Adobe Creative Cloud, vývojáři zaplatí 30 dolarů, získají tak možnost tvořit aplikace do velikosti až 1 GB, zároveň získají další výhody členství.

Adobe nabízí i použití služby zdarma. V této licenci je možné vytvořit pouze jednu aplikaci o velikosti maximálně 50 MB. [20]

Pro vývoj je možné stáhnout speciální vývojové prostředí nazývané PhoneGap Desktop, které ocení programátoři, kteří při vývoji preferují grafické uživatelské rozhraní.



Obrázek 2 PhoneGap Desktop a spuštění projektu v prohlížeči [Zdroj: autor práce]

### 5.3.2 Sencha Touch

Sencha Touch je javascriptový framework určený pro tvorbu multiplatformních mobilních aplikací pomocí webových technologií. Využívá hardwarové akcelerační techniky pro poskytnutí vysoce výkonných komponent uživatelského rozhraní.

S více jak 50 vestavěnými komponentami je možné tvořit aplikace pro Android, iOS i Windows Phone.

Sencha Touch je dostupná zdarma. [21]

### 5.3.3 Xamarin

Xamarin je open source framework vlastněný firmou Microsoft, který umožňuje vývojářům vytvořit aplikace pro všechny tři nejrozšířenější platformy pomocí programovacího jazyka C#.

Framework je postaven na projektu Mono a byl dříve spravován společností Novell. V únoru roku 2016 byl Xamarin koupen již zmíněnou společností Microsoft.

Xamarin disponuje silnými nástroji pro vývoj, mezi které patří Xamarin Test Cloud, Xamarin Profiler nebo Xamarin Insights. [22]

Co se týká licencí, je Xamarin pro malé týmy dostupný zdarma. Pro vývoj je možné použít vývojové prostředí Visual Studio, které je silným nástrojem i v jiných oblastech než právě v mobilních aplikacích. Dalším použitelným nástrojem je Xamarin Studio, vytvořené speciálně pro vývoj aplikací pomocí jazyka C#.

### 5.3.4 Unity 3D

Unity je framework, který je zaměřen speciálně na herní vývoj. Při jeho použití, vývojáři mohou využít znalosti jazyka C# nebo speciálního jazyka UnityScript, který se syntaxí podobá jazyku Javascript.

Pomocí unity lze vytvořit aplikaci pro více jak 25 různých platform. [23]

Unity je dostupné v celkem čtyřech licencích, mezi kterými lze nalézt i licenci dostupnou zdarma. S touto verzí přichází jistá omezení. Ostatní licence cenově začínají na 35 dolarech měsíčně.

## 6 Vývoj aplikace pro platformu Android v praxi

V této části práce je zobrazen postup vývoje nativní mobilní aplikace pro platformu Android. Autor vyvinul aplikaci ve spolupráci se společností Lineup Systems Ltd., která slouží jako mobilní verze již existujícího inzertního systému.

V následující kapitole je zároveň popsán obecný postup vytváření uživatelského rozhraní ve vývojovém prostředí Android Studio. Dále je vysvětlen základní princip fungování jednotlivých obrazovek nativní aplikace pro Android.

### 6.1 Zadání

Autor vytvoří mobilní nativní aplikaci pro platformu Android. Aplikace bude sloužit obchodníkům, kteří po přihlášení uvidí seznam svých zákazníků. Přihlášený obchodník bude mít možnost zobrazit detail zákazníka, kde uvidí jeho osobní údaje.

U každého zákazníka je možné spravovat aktivity, které fungují jako upomínky. Říkají obchodníkovi, kdy má s klientem naplánovanou schůzku, kdy ho má kontaktovat nebo např. kdy mu má zaslat důležité dokumenty. Obchodník může aktivity vytvářet, upravovat a mazat.

Dále je možné u zákazníka sledovat objednávky, které vytvořil. Objednávka je jen obálka pro skupinu inzerátů. Inzerátem se rozumí už jedna konkrétní položka, která má dle svého typu určené základní vlastnosti, např. pro tištěnou inzerci jsou to rozměry, sekce, jméno novin a datum, kdy má vyjít. Inzeráty jsou obecné, ale jsou děleny do sekcí jako jsou domácí zpravodajství, sport, počasí apod.

Autorovi bylo umožněno nahlédnout do již existujícího inzertního systému, odkud převzal nejdůležitější vlastnosti výše zmíněných entit.

Zadavatel práce se s autorem dohodl, že autor vytvoří i serverovou část aplikace, která bude obstarávat komunikaci s databází. Jaké technologie budou použity pro vývoj serverové části, bylo ponecháno na rozhodnutí autora.

Po dokončení spolupráce, si sám zadavatel již výslednou aplikaci upraví tak, aby byla schopná komunikovat s vlastním serverovým řešením.



## 6.2 Použité technologie

Jak již bylo popsáno výše, aplikace bude psána v nativním jazyce Java. Autor se rozhodl, že pro vývoj využije oficiální vývojové prostředí Android Studio, které přináší mnoho výhod. Mezi jednu z výhod patří jednoduchost instalace emulátorů, které budou použity pro testování aplikace. Aplikace bude dále testována na reálných zařízeních, konkrétně na chytrých telefonech Samsung Galaxy Alpha, Samsung S3 Neo a na tabletu Samsung Galaxy Tab E 9.6.

Pro serverovou část aplikace bude použit jazyk PHP, který je dostupný zdarma. Autor se takto rozhodl, jelikož již má s tímto jazykem zkušenosti. Pro zjednodušení práce a zpřehlednění zdrojového kódu bude dále použit framework Nette, pro který se autor rozhodl opět z důvodu zkušeností s tímto frameworkem. Jako vývojové prostředí bude použit program PHPStorm od firmy JetBrains s.r.o.

Pro ukládání dat byla zvolena zdarma dostupná databáze MySQL.

Aplikace bude pro komunikaci s databází využívat API, které autor sám vytvoří.

Aby bylo mobilní aplikaci možné jednoduše testovat i na reálných zařízeních, bude serverová část umístěna na veřejně dostupné doméně, kterou autor vlastní.

## 6.3 Naplánování vývoje

Autor nejdříve nainstaluje všechny technologie, které budou použity.

Pokračovat bude návrhem databáze a vytvořením všech potřebných tabulek. Pro testování bude nutné tabulky naplnit daty. Jelikož přes mobilní aplikaci bude možné přidávat pouze aktivity, vytvoří autor ostatní testovací data ručně pomocí SQL dotazů.

Pokračovat bude vytvořením serverové části aplikace a API, pomocí kterého bude aplikace komunikovat s databází.

Následovat bude design a layout již samotné mobilní aplikace, který autor nejdříve navrhne a poté pomocí nástrojů vývojového prostředí Android Studia vytvoří.

Autor dále naprogramuje logiku aplikace. Vytvoří potřebné aktivity, třídy a jejich vlastnosti a metody.

Nakonec bude aplikace otestována na již zmíněných reálných zařízeních.

## 6.4 Instalace

V následujících kapitolách bude popsán popis instalací technologií, které budou použity při vývoji aplikace.

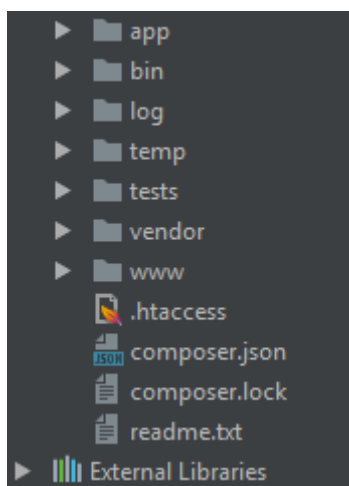
### 6.4.1 Nette

Autor začíná instalací prostřední pro serverovou část aplikace. Nejdříve vytvoří skeleton pro novou aplikaci pomocí Composeru<sup>11</sup> a následujícího příkazů.

```
composer create-project nette/web-project
```

#### Ukázka kódu 12 Příkaz pro vytvoření projektu Nette [24]

Po vytvoření projektu, vznikne následující souborová struktura.



Obrázek 3 Souborová struktura projektu Nette [Zdroj: autor práce]

### 6.4.2 MySQL

Jak již bylo popsáno výše, serverová část aplikace bude umístěna na veřejně dostupné doméně. Není tedy třeba instalovat a vytvářet novou databázi. Autor pouze získá od poskytovatele služeb potřebné údaje k připojení k databázi. Spravovat databázi bude přes nástroj phpMyAdmin.

---

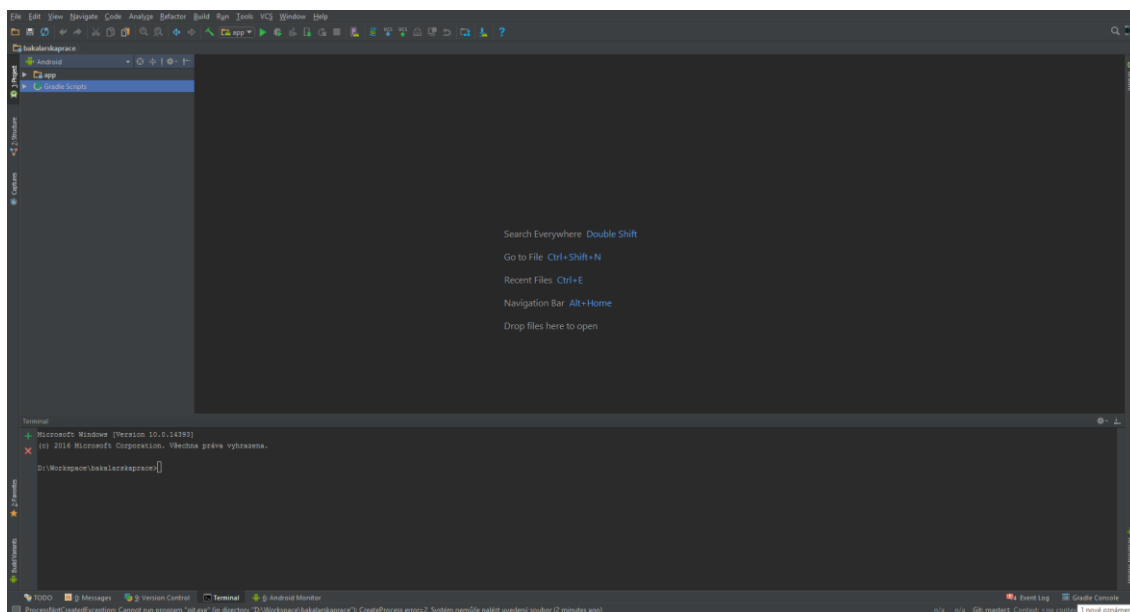
<sup>11</sup> Programová utilita určená pro správu knihoven a jiných zdrojů

### 6.4.3 Java

Aby bylo možné pustit se do vývoje, je nutné, aby měl vývojář nainstalovanou Javu. Zároveň je potřeba nainstalovat Java SE Development Kit, což je produkt, který obsahuje základní nástroje pro vývoj aplikací pro platformu Java. Tento produkt je možné zdarma stáhnout z oficiálního webu společnosti Oracle.

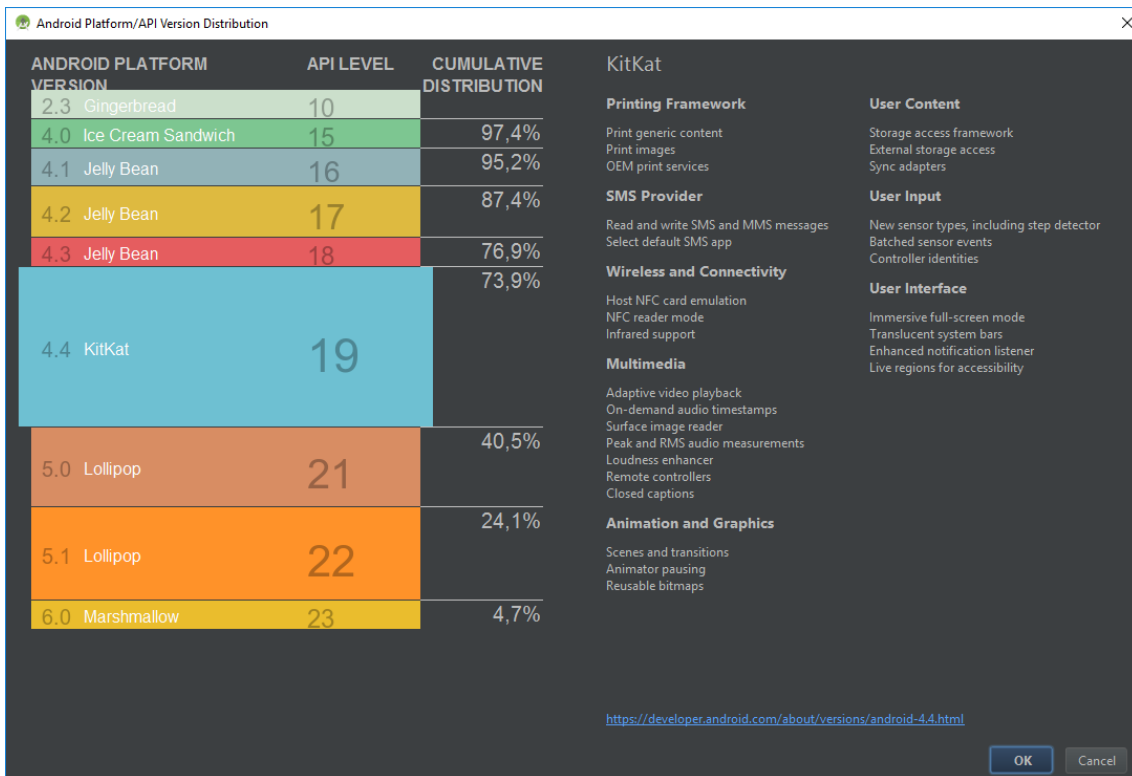
### 6.4.4 Android Studio

Android studio je možné zdarma stáhnout z oficiálního zdroje pro android vývojáře. Společně s vývojovým prostředím lze zároveň stáhnout i Android SDK. Jedná se o sadu softwarových vývojových nástrojů, které umožňují vyvíjet aplikaci právě pro platformu Android. Zároveň obsahuje ukázkové projekty se zdrojovými kódy a emulátor, který je možné použít pro testování při vývoji.



**Obrázek 4** Vývojové prostředí Android Studio [Zdroj: autor práce]

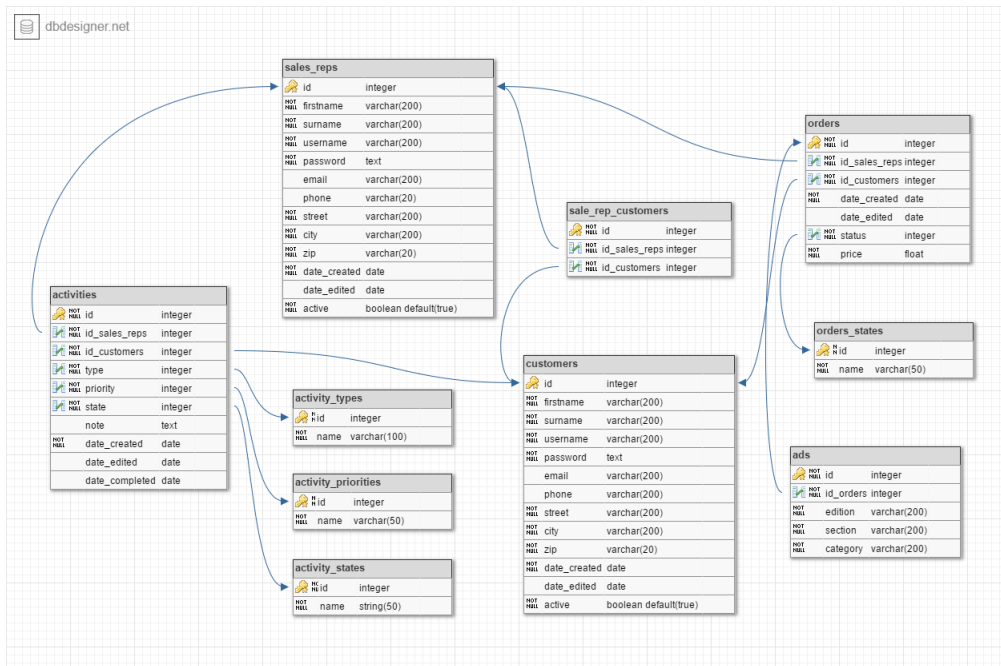
Při vytváření nového projektu, je možné zvolit cílenou verzi API. Když vývojář zvolí nižší verzi, půjde jeho aplikaci spustit na starších zařízeních, ale zároveň přijde o možnost využívat nové funkce, kterými disponují verze nové. Autor práce zvolil API verze 19.



Obrázek 5 Zvolení verze platformy [Zdroj: autor práce]

## 6.5 Návrh databáze

Autor vytvoří celkem 10 tabulek, které budou ukládat potřebná data.



Obrázek 6 Návrh databáze. [Zdroj: autor práce]

## 6.5.1 Vytvoření tabulek

Pro vytvoření tabulek byly použity následující SQL dotazy.

```
CREATE TABLE IF NOT EXISTS `activities` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `id_sales_reps` int(11) NOT NULL,  
  `id_customers` int(11) NOT NULL,  
  `type` int(11) NOT NULL,  
  `priority` int(11) NOT NULL,  
  `state` int(11) NOT NULL,  
  `note` longtext COLLATE utf8_czech_ci NOT NULL,  
  `date_created` date NOT NULL,  
  `date_edited` date NOT NULL,  
  `date_completed` date NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
CREATE TABLE IF NOT EXISTS `activity_priorities` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(200) COLLATE utf8_czech_ci NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
CREATE TABLE IF NOT EXISTS `activity_states` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(200) COLLATE utf8_czech_ci NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
CREATE TABLE IF NOT EXISTS `activity_types` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(200) COLLATE utf8_czech_ci NOT NULL,  
  PRIMARY KEY (`id`)  
);  
  
CREATE TABLE IF NOT EXISTS `ads` (  
  `id` int(11) NOT NULL AUTO_INCREMENT,  
  `name` varchar(200) COLLATE utf8_czech_ci NOT NULL,  
  PRIMARY KEY (`id`)  
);
```

```

`id` int(11) NOT NULL AUTO_INCREMENT,
`id_orders` int(11) NOT NULL,
`edition` varchar(200) COLLATE utf8_czech_ci NOT NULL,
`section` varchar(200) COLLATE utf8_czech_ci NOT NULL,
`category` varchar(200) COLLATE utf8_czech_ci NOT NULL,
PRIMARY KEY (`id`)
);

CREATE TABLE IF NOT EXISTS `customers` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `firstname` varchar(200) COLLATE utf8_czech_ci NOT NULL,
  `surname` varchar(200) COLLATE utf8_czech_ci NOT NULL,
  `username` varchar(200) COLLATE utf8_czech_ci NOT NULL,
  `password` longtext COLLATE utf8_czech_ci NOT NULL,
  `email` varchar(200) COLLATE utf8_czech_ci DEFAULT NULL,
  `phone` varchar(20) COLLATE utf8_czech_ci NOT NULL,
  `street` varchar(200) COLLATE utf8_czech_ci NOT NULL,
  `city` varchar(200) COLLATE utf8_czech_ci NOT NULL,
  `zip` varchar(20) COLLATE utf8_czech_ci NOT NULL,
  `date_created` date NOT NULL,
  `date_edited` date DEFAULT NULL,
  `active` tinyint(1) NOT NULL DEFAULT '1',
  PRIMARY KEY (`id`)
);

CREATE TABLE IF NOT EXISTS `orders` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_sales_reps` int(11) NOT NULL,
  `id_customers` int(11) NOT NULL,
  `date_created` date NOT NULL,
  `date_edited` date NOT NULL,
  `status` int(11) NOT NULL,
  `price` float NOT NULL,

```

```

    PRIMARY KEY (`id`)
);
CREATE TABLE IF NOT EXISTS `orders_states` (
    `id` int(11) NOT NULL,
    `name` varchar(200) COLLATE utf8_czech_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_czech_ci;
CREATE TABLE IF NOT EXISTS `sales_reps` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `firstname` varchar(200) COLLATE utf8_czech_ci NOT NULL,
    `surname` varchar(200) COLLATE utf8_czech_ci NOT NULL,
    `username` varchar(200) COLLATE utf8_czech_ci NOT NULL,
    `password` longtext COLLATE utf8_czech_ci NOT NULL,
    `email` varchar(200) COLLATE utf8_czech_ci DEFAULT NULL,
    `phone` varchar(20) COLLATE utf8_czech_ci NOT NULL,
    `street` varchar(200) COLLATE utf8_czech_ci NOT NULL,
    `city` varchar(200) COLLATE utf8_czech_ci NOT NULL,
    `zip` varchar(20) COLLATE utf8_czech_ci NOT NULL,
    `date_created` date NOT NULL,
    `date_edited` date DEFAULT NULL,
    `active` tinyint(1) NOT NULL DEFAULT '1',
    PRIMARY KEY (`id`)
);
CREATE TABLE IF NOT EXISTS `sale_rep_customers` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `id_sales_reps` int(11) NOT NULL,
    `id_customers` int(11) NOT NULL,
    PRIMARY KEY (`id`)
);

```

**Ukázka kódu 13 SQL dotazy pro vytvoření tabulek [Zdroj: autor práce]**

## 6.5.2 Naplnění tabulek daty

Aby bylo možné aplikaci testovat, je nutné, aby se v databázi nacházela potřebná data. Autor data do databáze vloží ručně pomocí SQL dotazů.

```
/* activity_types, activity_priorities, activity_states, orders_states
*/
INSERT INTO activity_priorities (name) VALUES ('Low');
INSERT INTO activity_priorities (name) VALUES ('Normal');
INSERT INTO activity_priorities (name) VALUES ('High');

INSERT INTO activity_types (name) VALUES ('Call');
INSERT INTO activity_types (name) VALUES ('Call back');
INSERT INTO activity_types (name) VALUES ('Document Storage');
INSERT INTO activity_types (name) VALUES ('Invoice task');
INSERT INTO activity_types (name) VALUES ('Mail');
INSERT INTO activity_types (name) VALUES ('Meeting');
INSERT INTO activity_types (name) VALUES ('Other');

INSERT INTO activity_states (name) VALUES ('ToDo');
INSERT INTO activity_states (name) VALUES ('Done');
INSERT INTO activity_states (name) VALUES ('Cancelled');

INSERT INTO orders_states (name) VALUES ('Active');
INSERT INTO orders_states (name) VALUES ('Proposal');
INSERT INTO orders_states (name) VALUES ('Reservation');
INSERT INTO orders_states (name) VALUES ('Confirmed');

/* Vložení obchodníků */
INSERT INTO sales_reps (firstname, surname, username, password, email,
phone, street, city, zip, date_created, date_edited, active) VALUES
('Jan', 'Novák', 'novakja', 'test', 'jan.novak@email.com', '+420 123
456 789', 'Za lesem 56', 'Hradec Králové', '425 56', '2017-02-15',
NULL, true);

/* Vložení zákazníků */
INSERT INTO customers (firstname, surname, username, password, email,
phone, street, city, zip, date_created, date_edited, active) VALUES
('Jakub', 'Svoboda', 'svobodaja', 'test', 'jakub.svoboda@email.cz',
'+420 456 789 123', 'Na náměstí 12', 'Brno', '123 25', '2017-02-17',
'2017-02-20', true);

INSERT INTO customers (firstname, surname, username, password, email,
phone, street, city, zip, date_created, date_edited, active) VALUES
('Pavel', 'Peterka', 'peterkapa', 'test', 'pavel.peterka@email.cz',
'+420 789 456 124', 'Pampelišková 360', 'Trutnov', '879 45', '2017-02-
26', NULL, true);

INSERT INTO customers (firstname, surname, username, password, email,
phone, street, city, zip, date_created, date_edited, active) VALUES
('David', 'Dlouhý', 'dlouhyda', 'test', 'david.dlouhy@email.cz', '+420
987 456 321', 'Za rohem 1', 'Hradec Králové', '425 56', '2017-02-20',
'2017-02-21', false);

/* Naplnění tabulky přiřazující zákazníky k obchodníkům */
INSERT INTO sale_rep_customers (id_sales_reps, id_customers) VALUES
(1, 1);
```



```

INSERT INTO sale_rep_customers (id_sales_reps, id_customers) VALUES
(1, 2);
INSERT INTO sale_rep_customers (id_sales_reps, id_customers) VALUES
(1, 3);

/* Naplnění tabulky objednávek */
INSERT INTO orders (id, id_sales_reps, id_customers, date_created,
date_edited, status, price) VALUES (1, 1, '2017-03-21', '2017-03-21',
2, 15000);
INSERT INTO orders (id, id_sales_reps, id_customers, date_created,
date_edited, status, price) VALUES (1, 2, '2017-03-21', NULL, 3,
3000);
INSERT INTO orders (id, id_sales_reps, id_customers, date_created,
date_edited, status, price) VALUES (1, 2, '2017-03-30', '2017-03-31',
3, 6000);

INSERT INTO ads (id, id_orders, edition, section, category) VALUES (1,
'The Sun - Scotland', 'Features', 'Elektronika');
INSERT INTO ads (id, id_orders, edition, section, category) VALUES (1,
'The Sun - N.Ireland', 'Features', 'Koníčky');
INSERT INTO ads (id, id_orders, edition, section, category) VALUES (3,
'The Sun - N.Ireland', 'Business Opportunities', 'Business');
INSERT INTO ads (id, id_orders, edition, section, category) VALUES (4,
'The Sun - UK', 'Business Opportunities', 'Zábava');
INSERT INTO ads (id, id_orders, edition, section, category) VALUES (4,
'The Sun - UK', 'Business Opportunities', 'Zábava');
INSERT INTO ads (id, id_orders, edition, section, category) VALUES (4,
'The Sun - UK', 'Business Opportunities', 'Zábava');

```

**Ukázka kódu 14 SQL dotazy pro naplnění tabulek daty [Zdroj: autor práce]**

## 6.6 Serverová část aplikace

Serverová část aplikace bude mobilnímu klientovi poskytovat API, přes které bude moci komunikovat s databází. Aplikace bude získávat data ve formátu JSON.

### 6.6.1 Application Programming Interface

Application Programming Interface neboli API, je rozhraní, pomocí kterého může aplikace komunikovat s aplikací jinou.

Při vývoji této mobilní aplikace, bude použita architektura REST, která je oproti ostatním jako je například SOAP, orientována datově, nikoliv procedurálně.

Rozhraní REST je použitelné pro jednotný a snadný přístup ke zdrojům. Všechny zdroje mají vlastní identifikátor URI a REST definuje čtyři základní metody pro přístup k nim. Tyto čtyři základní metody jsou označovány jako CRUD. Vytvoření dat (Create), získání dat (Retrieve), změna dat (Update) a jejich mazání (Delete). [25]

Serverová část aplikace, kterou autor vytvořil, bude poskytovat následující metody.

```
http://bakalar.webovkanamiru.cz/api/login/  
http://bakalar.webovkanamiru.cz/api/customers/  
http://bakalar.webovkanamiru.cz/api/customer/  
http://bakalar.webovkanamiru.cz/api/activities/  
http://bakalar.webovkanamiru.cz/api/activity/  
http://bakalar.webovkanamiru.cz/api/orders/  
http://bakalar.webovkanamiru.cz/api/order/
```

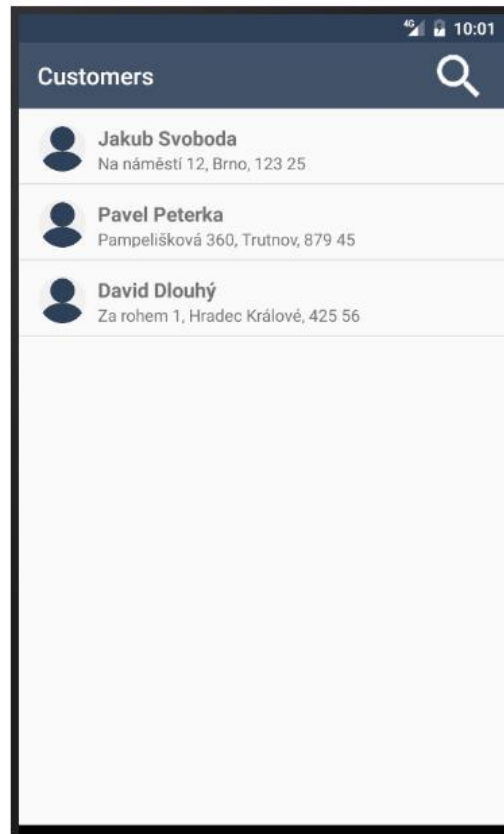
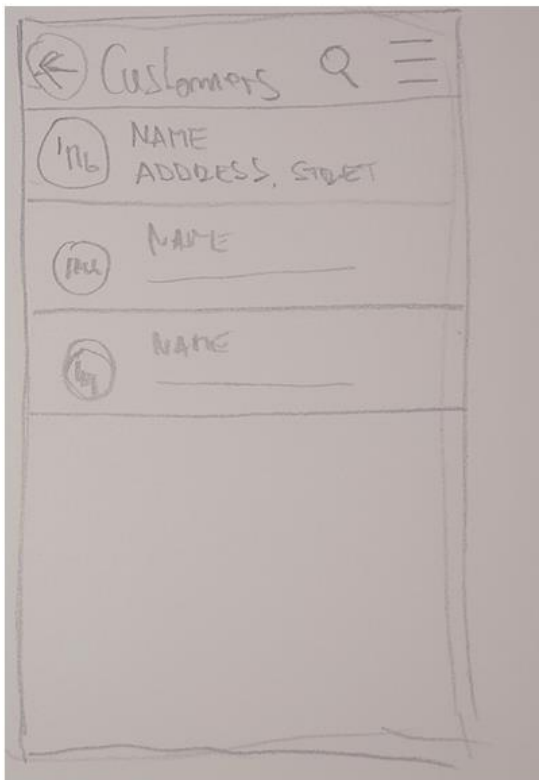
#### **Ukázka kódu 15 Konkrétní URI vracející data [Zdroj: autor práce]**

Mimo různých parametrů, pomocí kterých jsou získávána konkrétní data, je v každém požadavku poslán šifrovaný kód. Pomocí tohoto kódu serverová část aplikace pozná, že komunikuje právě s mobilní aplikací, pro kterou byla vytvořena. V případě neposlání žádného nebo poslání chybného kódu, nejsou vrácena žádná data.

## **6.7 Vzhled aplikace**

Autor nejdříve navrhne layout aplikace. Uživatel bude moci přepínat mezi celkem 6 obrazovkami. První obrazovka, kterou uživatel uvidí při spuštění aplikace, je obrazovka pro přihlášení, kde bude nucen zadat uživatelské jméno a heslo.

Po přihlášení do aplikace, ho přivítá obrazovka se seznamem jeho zákazníků. Po kliknutí na položku v seznamu, bude přesměrován na obrazovku s detailními informacemi o zákazníkovi. Zde uvidí více osobních informací a seznam nejbližších pěti aktivit. Na této obrazovce se budou dále nacházet dvě tlačítka pro zobrazení všech aktivit a objednávek.

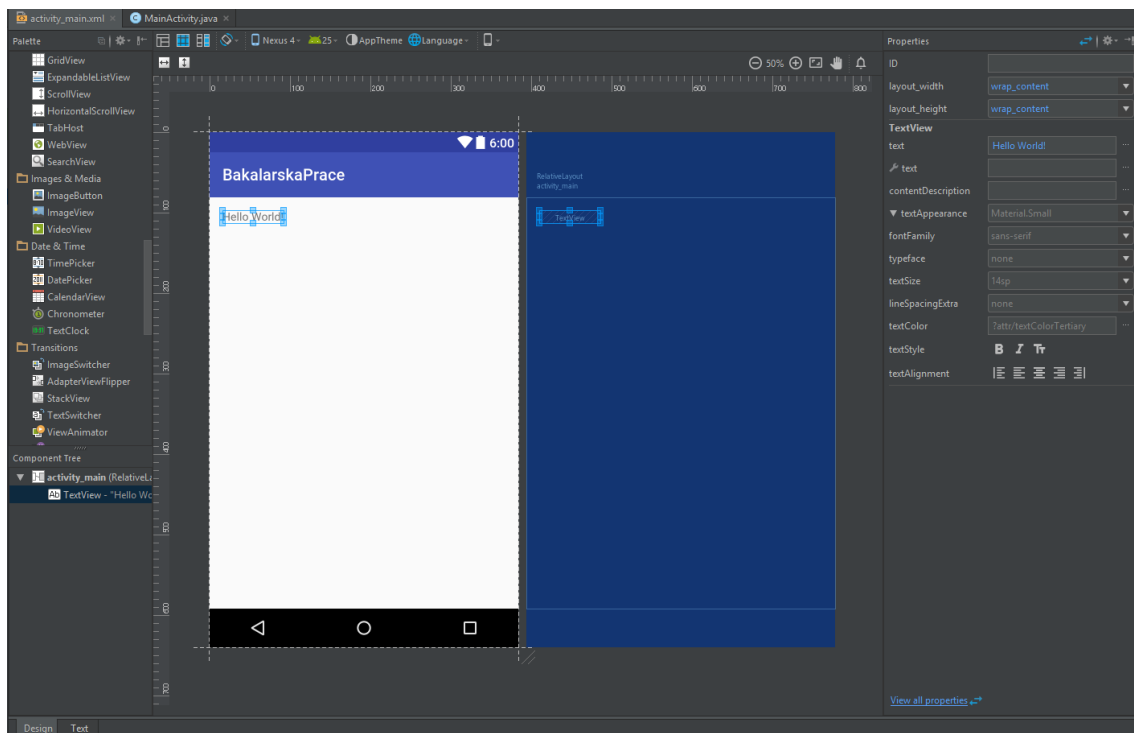


**Obrázek 7** Vytvoření designu z návrhu [Zdroj: autor práce (pokračování - Příloha 1)]

Vzhled jednotlivých obrazovek lze v Android Studiu upravovat dvěma způsoby. Jednak disponuje jednoduchým editorem, ve kterém může vývojář lehce vkládat různé ovládací prvky pouze pomocí myši. Na výběr jsou různé widgety<sup>12</sup>, textová pole, kontejnery, obrázky, tlačítka a mnoho dalších. Po vybrání konkrétního prvku, je možné editovat jeho vlastnosti jako jsou odsazení, identifikátory, barvy, text a mnoho dalších.

---

<sup>12</sup> Všeobecné označení pro ovládací prvek



**Obrázek 8 Editor designu ve vývojovém prostředí Android Studio [Zdroj: autor práce]**

Layout jednotlivých obrazovek, je ukládán jako soubor ve formátu XML. Tento soubor, je možné upravovat ručně. Jakákoliv změna je okamžitě promítnuta do editoru.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.jakubbene.bakalarskaprace.MainActivity">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Ahoj světe!"
    android:textStyle="bold" />
</RelativeLayout>
```

#### **Ukázka kódu 16 XML soubor definující vzhled obrazovky [Zdroj: autor práce]**

V souboru se mimo základního layoutu nachází i všechny komponenty, které se v dané obrazovce nacházejí. V tomto konkrétním případě jde o textové pole s určitým tučným textem.

Může se stát, že se určité části, texty nebo barvy opakují. V takovém případě je vhodné, definovat je pouze na jednom místě. K tomuto účelu slouží další XML soubory. Na definice v těchto souborech se lze jednoduše odkazovat. Jakákoliv případná budoucí změna, ušetří vývojáři mnoho času.

```
<resources>
    <string name="app_name">BakalarskaPrace</string>
</resources>
```

#### **Ukázka kódu 17 Definice textových řetězců v souboru strings.xml [Zdroj: autor práce]**

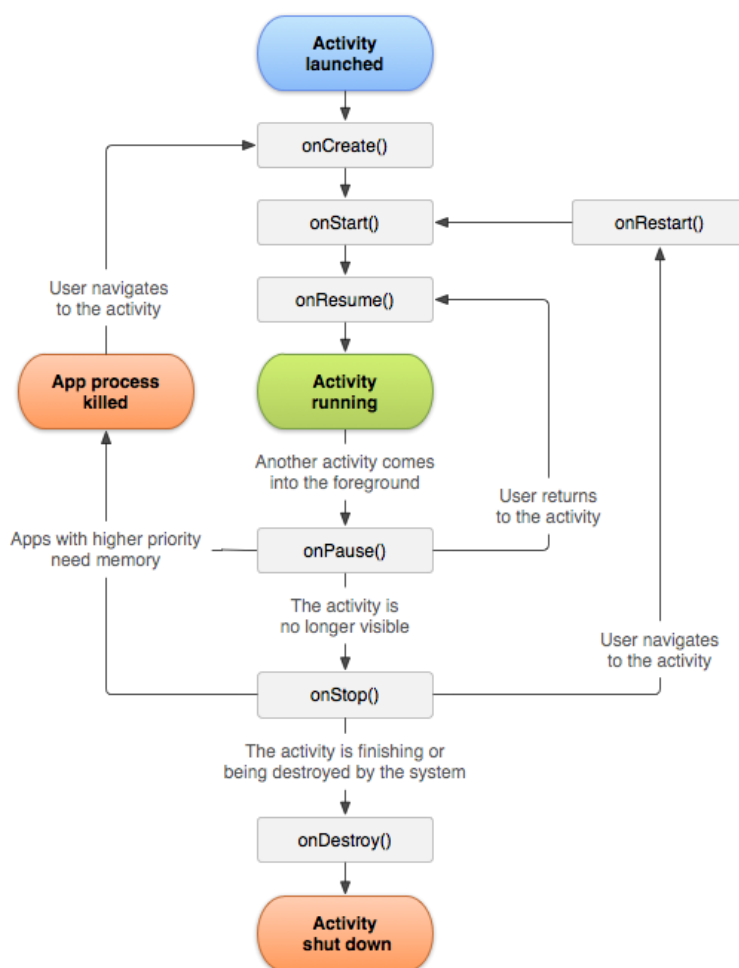
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#415269</color>
    <color name="colorPrimaryDark">#2F4159</color>
    <color name="colorAccent">#3AD9F6</color>
</resources>
```

#### **Ukázka kódu 18 Definice barev v souboru colors.xml [Zdroj: autor práce]**

## **6.8 Aktivity**

Hlavní částí mobilní aplikace, které jsou používány při programování pro platformu Android, jsou tzv. Aktivity. Každá aktivita zařizuje, aby se data, která byla získána od nižších vrstev aplikace, správně zobrazila uživateli. Každá z nich reprezentuje jednu obrazovku, s kterou může uživatel pracovat. Pro každou aktivitu je vytvořeno uživatelské rozhraní, což znamená, že může obsahovat různá tlačítka, textová pole, seznamy nebo další ovládací prvky.

Každá aktivita se může nacházet v různých stavech, které popisuje následující diagram životního cyklu aktivity.



**Obrázek 9 Diagram životního cyklu aktivity [26]**

Aktivity jsou v systému spravovány jako zásobník. Jakmile je aktivita spuštěna, dostává se na vrchol zásobníku a stává se z ní běžící aktivita. Předchozí spuštěná aktivita se posouvá v zásobníku níže a nedostává se do popředí, dokud aktivita nová existuje.

Jak již bylo popsáno, aktivita může nabývat různých stavů, konkrétně se jedná o následující stavy:

- Pokud je aktivita na popředí obrazovky, tedy na vrcholu zásobníku, jedná se o spuštěnou aktivitu.

- Pokud je aktivita částečně překryta jinou aktivitou nebo se nad ní nachází aktivita průhledná, jedná se o pozastavenou aktivitu. Taková je stále spuštěná a dále se v ní uchovávají všechny informace a stavy. Pokud dojde k extrémnímu nedostatku paměti, může být tato aktivita systémem ukončena.
- Pokud je aktivita kompletně překryta aktivitou jinou, nachází se ve stavu zastavená. Stále uchovává informace a stavy, ale může být v případě potřeby ukončena systémem.
- Jestliže je aktivita ve stavu pozastavená nebo zastavená, může být systémem ukončena. Pokud se uživatel k takovéto aktivitě vrátí, musí být znovu spuštěna.

Celý životní cyklus aktivit je definovaný následujícími metodami aktivit. Každá z metod je přepisovatelná a může být upravena programátorem.

```
public class Activity extends ApplicationContext {
    protected void onCreate(Bundle savedInstanceState);
    protected void onStart();
    protected void onRestart();
    protected void onResume();
    protected void onPause();
    protected void onStop();
    protected void onDestroy();
}
```

**Ukázka kódu 19 Aktivita implementující metody definující životní cyklus aktivity [26]**

- onCreate – Metoda, která je volána při vytvoření aktivity. V této metodě by měly být vytvářeny pohledy, přiřazována data seznamům apod. Metoda je vždy následována metodou onStart.
- onRestart – Metoda je volána ve chvíli, kdy aktivita přechází ze stavu pozastavená do stavu spuštěná. Je vždy následována metodou onStart.

- `onStart` – Je volána ve chvíli, kdy se aktivita stává viditelnou pro uživatele. V této metodě se provádí všechny úkony, které je třeba provést, aby byla daná aktivita zobrazena.
- `onResume` – Ve chvíli, kdy dojde k interakci mezi uživatelem a aktivitou, je volána právě tato metoda. Vždy následuje metoda `onPause`.
- `onPause` – Volána ve chvíli, kdy se z dané aktivity přechází do aktivity jiné. V této metodě se typicky provádí uložení změn, zastavení animací a zastavení dalších aktivit, které mohou využívat procesor. Provedení této metody musí být rychle, jelikož nová aktivita není zobrazena, dokud tato metoda neskončí.
- `onStop` – Metoda je volána ve chvíli, kdy se pro uživatele stává neviditelnou, tedy při překrytí jinou aktivitou.
- `onDestroy` – Poslední aktivita je volána předtím, než je zničena. K její zničení může dojít ve chvíli, když je zavolána metoda `finish`. Zničena může být také systémem, když je vyžadována uvolnění paměti.

## 6.9 Testování

Autor během vývoje testoval aplikaci pomocí virtuálního zařízení. Nakonec vyexportoval spustitelný soubor a otestoval aplikaci na reálných zařízeních. Při testování byla aplikace svižná a nedošlo k žádnému jejímu nečekanému ukončení. Jelikož je aplikace závislá na připojení k internetu, byl při testování kladen důraz na případné výpadky připojení a změnu způsobu připojení k internetu.

Uživatelské rozhraní bylo na všech zařízeních přehledné a nedocházelo k zakrytí žádných z prvků.



## Shrnutí výsledků

V práci byl krátce popsán vývoj mobilních telefonů do dnešní podoby. Bylo vysvětleno, co je operační systém a jaké jsou nejvíce zastoupené systémy na mobilních zařízeních.

Práce se dále zabývala mobilními aplikacemi a tím, jak se staly nedílnou součástí lidského života. Z toho důvodu je vývoj mobilních aplikací velmi důležitý. Uživatelé automaticky počítají s možností využívat mobilní verze webových služeb a systémů. I běžné webové stránky či internetové obchody se již bez optimalizace pro mobilní zařízení neobejdou, a to právě z důvodu častého přístupu právě z mobilních telefonů či tabletů.

V práci byly dále popsány možné postupy při vývoji mobilních aplikací. Práce se zabývala jejich výhodami a nevýhodami a v jakých situacích je vhodné dané postupy zvolit. Text práce dále popsal technologie, které jsou využívány při vývoji mobilních aplikací, ale i technologie, pomocí kterých jsou tvořeny webové stránky. Tyto technologie jsou úzce spjaty právě s mobilním vývojem. Některé z nich je možné využívat přímo pro vývoj, pomocí jiný lze tvořit webová rozhraní pro umožnění komunikace.

Práce nakonec popsala, jak autor postupoval při vývoji nativní aplikace pro platformu Android. Zabývala se instalací technologií, které využíval, návrhem databázové struktury a následným vytvořením jednotlivých tabulek a testovacích dat. Popsala vzhled vytvářené aplikace a její jednotlivé obrazovky. Zároveň popsala, jak je pomocí vývojového prostřední Android Studio možné vytvářet uživatelské rozhraní.

Během vývoje bylo zjištěno, že by bylo mnohem vhodnější zvolit vývoji hybridní mobilní aplikace, zvláště pokud bude rozšířena na více mobilních platform, což lze předpokládat.

## Závěr

V rámci této práce byla vytvořena nativní mobilní aplikace pro platformu Android. Práce bude zcela jistě v budoucnu rozšířena o celou řadu nových funkcí, a to z důvodu, že aplikace je mobilní verzí již existujícího informačního systému a do mobilní podoby, byla přenesena jen malá část funkcí.

Vzhledem k nynějším funkcím aplikace, by bylo vhodnější vybrat vývoj hybridní mobilní aplikace namísto nativní. V současné době aplikace nevyužívá žádnou z výhod nativní aplikace. Vzhledem k tomu, že byla dokončena pouze její část, lze počítat s jejím rozšířením, které ale pravděpodobně nebude takové, aby byl využit potenciál nativních aplikací.

Jelikož lze předpokládat, že obchodníci vlastní různá zařízení, na kterých budou aplikaci využívat, bude ji nutné distribuovat na více platform. Bude tedy nutné aplikaci zvlášť vyvinout pro systém iOS, Windows Phone a případně další systémy. Vývoj hybridní mobilní aplikace by tento problém vyřešil a ušetřil tak společnosti čas i peníze.

## Seznam zkratek

|             |   |
|-------------|---|
| GPS .....   | Global Positioning System                   |
| IBM .....   | International Business Machines Corporation |
| APP .....   | Application                                 |
| PC .....    | Personal computer                           |
| HTML .....  | HyperText Markup Language                   |
| XHTML ..... | Extensible hypertext markup language        |
| XML .....   | Extensible Markup Language                  |
| CSS .....   | Cascading Style Sheets                      |
| JS .....    | JavaScript                                  |
| 3D .....    | Three-dimensional                           |
| ARC .....   | Automatic Reference Counting                |
| AJAX .....  | Asynchronous JavaScript and XML             |
| JSON .....  | JavaScript Object Notation                  |
| PHP .....   | Hypertext Preprocessor                      |
| API .....   | Application Programming Interface           |
| SQL .....   | Structured Query Language                   |
| SDK .....   | Software development kit                    |
| REST .....  | Representational state transfer             |
| URI .....   | Uniform Resource Identifier                 |
| SOAP .....  | Simple Object Access                        |

## Seznam použitých zdrojů

- [1] M. Taylor, „The evolution of the smartphone,“ [Online]. Available: <http://pocketnow.com/2014/07/28/the-evolution-of-the-smartphone>. [Přístup získán 22 Prosinec 2016].
- [2] GSMARNA, „Webová stránka poskytující informace o parametrech mobilních telefonů,“ [Online]. Available: <http://www.gsmarena.com>. [Přístup získán 18 Leden 2017].
- [3] „operating-system.org,“ [Online]. Available: [http://www.operating-system.org/betriebssystem/\\_english/os-liste.htm](http://www.operating-system.org/betriebssystem/_english/os-liste.htm). [Přístup získán 15 Únor 2017].
- [4] netmarketshare.com, „Mobile/Tablet Operating System Market Share,“ [Online]. Available: [netmarketshare.com](http://netmarketshare.com). [Přístup získán 15 Březen 2017].
- [5] K. Kilián, „Historie Androidu v kostce aneb Od verze 1.0 až po Android M,“ [Online]. Available: <https://www.svetandroida.cz/historie-androidu-201506>. [Přístup získán 10 Leden 2017].
- [6] statista.com, „Number of apps available in leading app stores as of March 2017,“ [Online]. Available: <https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>. [Přístup získán 30 Březen 2017].
- [7] D. Grebeň, „Kompletní historie iOS: od prvního iPhone až po iOS 9,“ [Online]. Available: <https://www.letemsvetemapplem.eu/2016/03/06/kompletni-historie-ios/>. [Přístup získán 15 Březen 2017].
- [8] J. Smrček, „OS Windows Mobile/Phone: strmá cesta historií,“ [Online]. Available: <https://www.cnews.cz/os-windows-mobilephone-strma-cesta-historii/>. [Přístup získán 15 březem 2017].

- [9] statista.com, „Most popular Google Play app store categories in 4th quarter 2015 and 2016, by number of downloads (in millions),“ [Online]. Available: <https://www.statista.com/statistics/256772/most-popular-app-categories-in-the-google-play-store/>. [Přístup získán 25 Březen 2017].
- [10] D. Čápka, „1. díl - Úvod do objektově orientovaného programování v C#,“ [Online]. Available: <http://www.itnetwork.cz/csharp/oop/c-sharp-tutorial-uvod-do-objektove-orientovaneho-programovani>. [Přístup získán 30 Březen 2017].
- [11] T. Bosák, „Úvod do programovacího jazyka Java,“ [Online]. Available: <http://programujte.com/clanek/2006041804-uvod-do-programovacieho-jazyka-java/>. [Přístup získán 25 Březen 2017].
- [12] S. CASS, „The 2016 Top Programming Languages,“ [Online]. Available: <http://spectrum.ieee.org/computing/software/the-2016-top-programming-languages>. [Přístup získán 25 Březen 2017].
- [13] B. Jakuben, „The Beginner’s Guide to Objective-C: Language and Variables,“ [Online]. Available: <http://blog.teamtreehouse.com/the-beginners-guide-to-objective-c-language-and-variables>. [Přístup získán 24 Březen 2017].
- [14] P. Černohorský, „Historie a vývoj jazyka C (od C až po C#),“ [Online]. Available: <http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xcernoh1.htm>. [Přístup získán 30 Březen 2017].
- [15] J. Kosek, „Historie a vývoj HTML,“ [Online]. Available: <http://htmlguru.cz/uvod-historie.html>. [Přístup získán 29 Březen 2017].
- [16] A. E. Hissom, „History of HTML and CSS,“ [Online]. Available: <http://amyhissom.com/HTML5-CSS3/history.html#4>. [Přístup získán 29 Březen 2017].

- [17] W3C, „Selectors Level 3,“ [Online]. Available: <https://www.w3.org/TR/selectors/#selectors>. [Přístup získán 30 Březen 2017].
- [18] D. Čápka, „1. díl - Úvod do ASP.NET,“ [Online]. Available: <http://www.itnetwork.cz/csharp/asp-net/tutorial-uvod-do-asp-dot-net>. [Přístup získán 30 Březen 2017].
- [19] B. LeRoux, „PhoneGap, Cordova, and what’s in a name?,“ [Online]. Available: <http://phonegap.com/blog/2012/03/19/phonegap-cordova-and-whate28099s-in-a-name/>. [Přístup získán 30 Březen 2017].
- [20] Adobe, „Adobe® PhoneGap™ Build. Package mobile apps in the cloud,“ [Online]. Available: <https://build.phonegap.com/>. [Přístup získán 30 Březen 2017].
- [21] Sencha, „Create native-looking HTML5 apps using JavaScript,“ [Online]. Available: <https://www.sencha.com/products/touch/#overview>. [Přístup získán 30 Březen 2017].
- [22] V. Mádr, „Xamarin: Představujeme nástroj pro multiplatformní vývoj mobilních aplikací (díl 1),“ [Online]. Available: <https://www.eman.cz/blog/xamarin-predstavujeme-nastroj-pro-multiplatformni-vyvoj-mobilnich-aplikaci-dil-1/Xamarin:> Představujeme nástroj pro multiplatformní vývoj mobilních aplikací (díl 1). [Přístup získán 26 Březen 2017].
- [23] Unity, „Build once, deploy anywhere,“ [Online]. Available: <https://unity3d.com/unity/multiplatform>. [Přístup získán 30 Březen 2017].
- [24] Nette, „Dokumentace instalace,“ [Online]. Available: <https://nette.org/cs/download>. [Přístup získán 25 Březen 2017].

- [25] M. Malý, „REST: architektura pro webové API,“ [Online]. Available: <https://www.zdrojak.cz/clanky/rest-architektura-pro-webove-api/>. [Přístup získán 27 Březen 2017].
- [26] A. dokumentace, „Activity,“ [Online]. Available: <https://developer.android.com/reference/android/app/Activity.html>. [Přístup získán 27 Březen 2017].

## Seznam obrázků

|  |    |
|--|----|
| Obrázek 1 Podíl trhu mobilních operačních systémů z února roku 2017 [4].....                 | 6  |
| Obrázek 2 PhoneGap Desktop a spuštění projektu v prohlížeči [Zdroj: autor práce]<br>.....    | 22 |
| Obrázek 3 Souborová struktura projektu Nette [Zdroj: autor práce] .....                      | 26 |
| Obrázek 4 Vývojové prostředí Android Studio [Zdroj: autor práce] .....                       | 27 |
| Obrázek 5 Zvolení verze platformy [Zdroj: autor práce] .....                                 | 28 |
| Obrázek 6 Návrh databáze. [Zdroj: autor práce] .....   | 28 |
| Obrázek 7 Vytvoření designu z návrhu [Zdroj: autor práce (pokračování – Příloha<br>1)] ..... | 35 |
| Obrázek 8 Editor designu ve vývojovém prostředí Android Studio [Zdroj: autor<br>práce] ..... | 36 |
| Obrázek 9 Diagram životního cyklu aktivity [26] .....  | 38 |

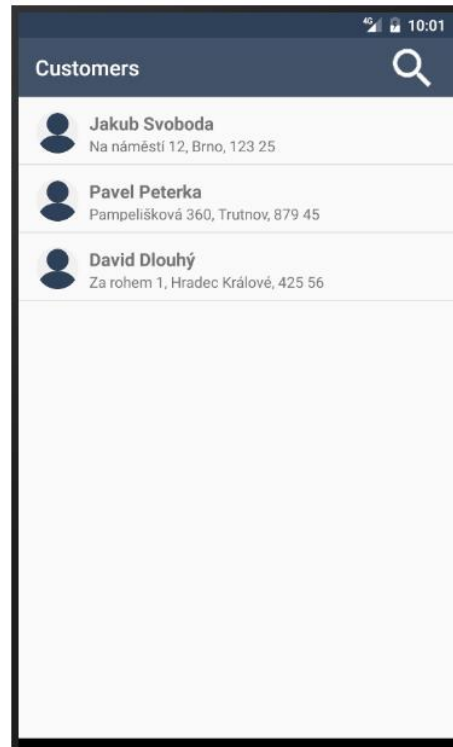
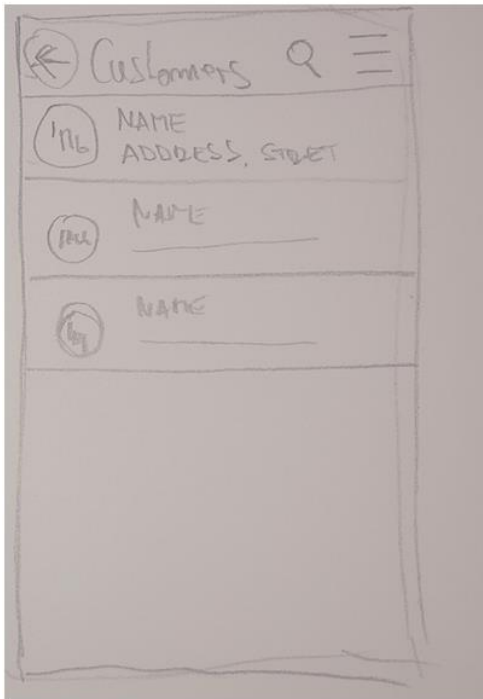


## Seznam tabulek

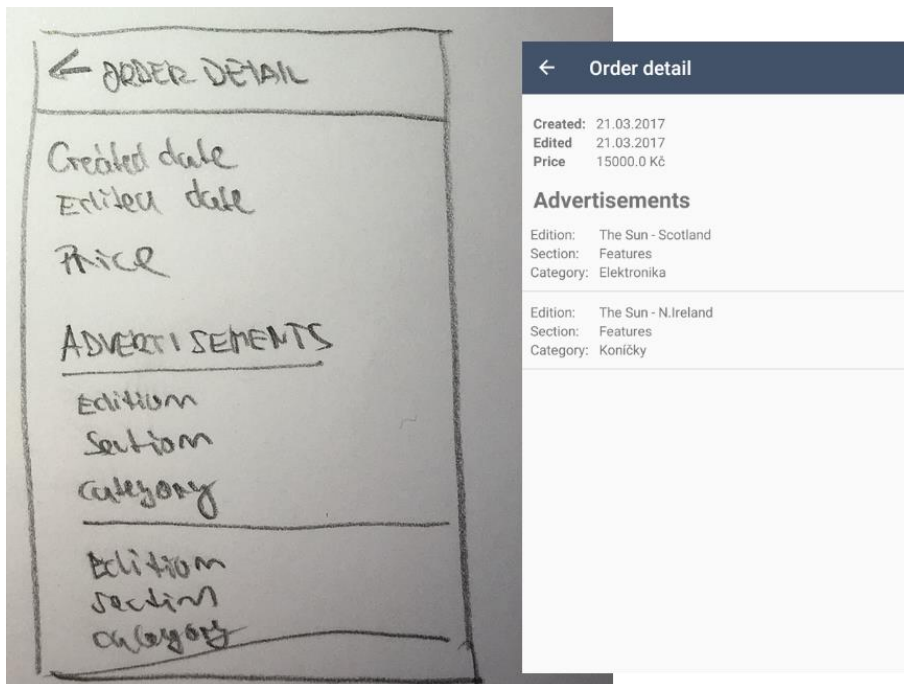
|  |   |
|--|---|
| Tabulka 1 Porovnání velikostí telefonů za poslední roky [2]..... | 3 |
|--|---|

## Seznam ukázek kódu

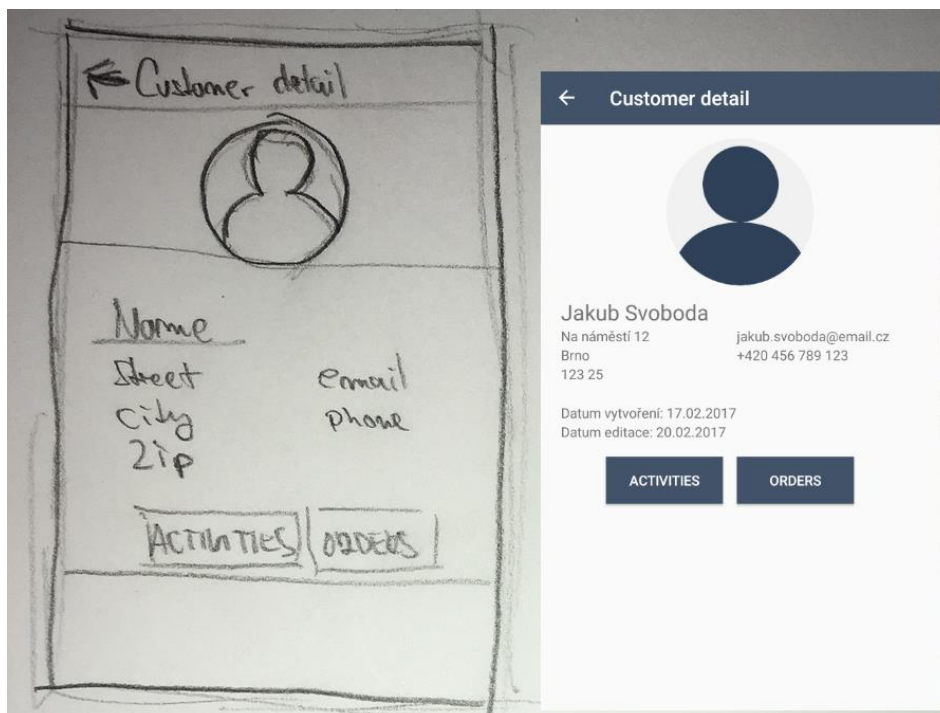
|  |    |
|--|----|
| Ukázka kódu 1 Jednoduchý program v jazyce Java [Zdroj: Autor práce].....   | 15 |
| Ukázka kódu 2 Příklad posílání zpráv v jazyce Objective-C [Zdroj: autor práce].....                                | 15 |
| Ukázka kódu 3 Jednoduchý program v jazyce C# [Zdroj: Auto práce] .....   | 16 |
| Ukázka kódu 4 Ukázka nových značek ve verzi HTML5 [Zdroj: autor práce].....  | 17 |
| Ukázka kódu 5 Možnosti základního formátování textu v HTML [Zdroj: autor<br>práce] .....                           | 17 |
| Ukázka kódu 6 Pokročilé selektory v CSS3 [17] .....  | 18 |
| Ukázka kódu 7 Vložení textu do vybraného elementu podle jednoznačného<br>identifikátoru [Zdroj: autor práce] ..... | 18 |
| Ukázka kódu 8 Použití jQuery [Zdroj: autor práce].....   | 19 |
| Ukázka kódu 9 Načtení aktuální teploty v Hradci Králové [Zdroj: autor práce].....                                  | 19 |
| Ukázka kódu 10 Formát JSON [Zdroj: autor práce].....   | 20 |
| Ukázka kódu 11 Jednoduché použití jazyka PHP [Zdroj: autor práce] .....  | 21 |
| Ukázka kódu 12 Příkaz pro vytvoření projektu Nette [24] .....  | 26 |
| Ukázka kódu 13 SQL dotazy pro vytvoření tabulek [Zdroj: autor práce] .....   | 31 |
| Ukázka kódu 14 SQL dotazy pro naplnění tabulek daty [Zdroj: autor práce].....                                      | 33 |
| Ukázka kódu 15 Konkrétní URI vracející data [Zdroj: autor práce].....  | 34 |
| Ukázka kódu 16 XML soubor definující vzhled obrazovky [Zdroj: autor práce] .....                                   | 37 |
| Ukázka kódu 17 Definice textových řetězců v souboru strings.xml [Zdroj: autor<br>práce] .....                      | 37 |
| Ukázka kódu 18 Definice barev v souboru colors.xml [Zdroj: autor práce] .....                                      | 37 |
| Ukázka kódu 19 Aktivita implementující metody definující životní cyklus aktivity<br>[26].....                      | 39 |



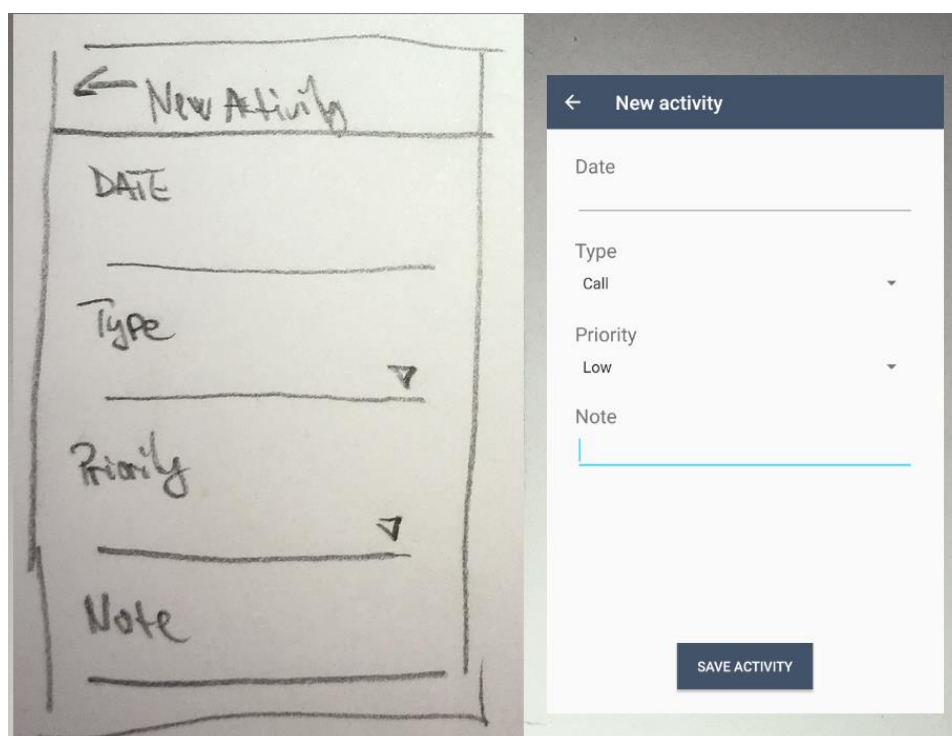
**Příloha 1 Vytvoření designu z návrhu - seznam zákazníků**



**Příloha 2 Vytvoření designu z návrhu - detail objednávky**



Příloha 3 Vytvoření designu z návrhu - detail zákazníka



Příloha 4 Vytvoření designu z návrhu - nová aktivita

Univerzita Hradec Králové  
Fakulta informatiky a managementu  
Akademický rok: 2016/2017

Studijní program: Aplikovaná informatika  
Forma: Prezenční  
Obor/komb.: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

| PŘEDKLÁDÁ:  | ADRESA                           | OSOBNÍ ČÍSLO |
|-------------|----------------------------------|--------------|
| Beneš Jakub | Dolní Promenáda 76, Janské Lázně | I1301363     |

**TÉMA ČESKY:**

Mobilní aplikace - IS pro správu a rezervaci inzerce

**TÉMA ANGLICKY:**

Mobile Applications - IS for management and reservation of advertisings

**VEDOUCÍ PRÁCE:**

doc. Ing. Filip Malý, Ph.D. - KIKM

**ZÁSADY PRO VYPRACOVÁNÍ:**

Cíl práce:

Prozkoumat možnosti v oblasti mobilních technologií a vytvořit mobilní aplikaci pro platformu Android pro společnost Lineup Systems.

Aplikace bude mobilní verzí již existujícího systému pro správu a rezervaci inzerce. Bude sloužit prodejcům a umožní v telefonu zpracovat nabídku na základě dat v systému.

Osnova:

- Úvod
- Mobilní a webové aplikace
- Rešerše multiplatformních frameworků
- Analýza, návrh, implementace aplikace
- Shrnutí, výsledky, závěr

**SEZNAM DOPORUČENÉ LITERATURY:**

Podpis studenta:



Datum:



Podpis vedoucího práce:



Datum:

