



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH A TVORBA MOBILNÍ APLIKACE PRO PLATFORMY ANDROID A IOS

DESIGN AND CREATION OF MOBILE APPLICATION FOR PLATFORMS ANDROID AND IOS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. HANA RYBNÍKÁŘOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. JAN LUHAN, Ph.D., MSc

BRNO 2021

Zadání diplomové práce

Ústav: Ústav informatiky
Studentka: **Bc. Hana Rybníkářová**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Informační management
Vedoucí práce: **Ing. Jan Luhan, Ph.D., MSc**
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Návrh a tvorba mobilní aplikace pro platformy Android a iOS

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Zaměřením práce je návrh a tvorba mobilní aplikace pro platformy Android a iOS, včetně analýzy vhodných implementačních nástrojů, s cílem vytvoření prototypu pro ukládání a vhodné zobrazení statistik z oblasti prodeje rukodělných výrobků.

Základní literární prameny:

GENADINIK, A. MobileApp Marketing and Monetization. 5th ed. CreateSpace Independent Publishing Platform, 2014. 149 p. ISBN 978-1502383822.

JACOBS, J. and B. RUDIS. Data-Driven Security: Analysis, Visualization and Dashboards. 1st ed. Indianapolis: John Wiley & Sons, 2014. 352 p. ISBN 978-1-118-79372-5.

KOZEL, R., L. MYNÁŘOVÁ a H. SVOBODOVÁ. Moderní metody a techniky marketingového výzkumu. 1. vyd. Praha: Grada, 2011. 304 s. ISBN 978-80-247-3527-6.

LACKO, L. a M. HERODEK. Vývoj aplikací pro iOS. 1. vyd. Brno: Computer Press, 2018. 480 s. ISBN 978-80-251-4942-3.

LEE, W-M. Beginning Android Application Development. 1st ed. Indianapolis: Wiley Publishing Inc., 2011. 448 p. ISBN 978-1-118-08780-0.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2020/21

V Brně dne 28.2.2021

L. S.

Mgr. Veronika Novotná, Ph.D.
ředitel

doc. Ing. Vojtěch Bartoš, Ph.D.
děkan

Abstrakt

Diplomová práce se zabývá návrhem a vytvořením prototypu mobilní aplikace pro operační systémy Android a iOS. K realizaci byl vybrán framework React Native, který umožňuje multiplatformní vývoj. Aplikace je zaměřena na vhodné zobrazení dat z prodeje rukodělných výrobků na webovém portálu Fler.

Abstract

The thesis deals with design and implementation of prototype of mobile application for Android and iOS operating systems. Framework React Native was chosen for realization because it allows multiplatform development. Application focuses on data visualization from sales of handmade products on Fler web portal.

Klíčová slova

Android, iOS, React Native, mobilní aplikace, návrh, grafy

Keywords

Android, iOS, React Native, mobile application, design, charts

Bibliografická citace

RYBNIKÁŘOVÁ, Hana. *Návrh a tvorba mobilní aplikace pro platformy Android a iOS*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/135114>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jan Luhan.

Čestné prohlášení

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

V Brně dne 16. 5. 2021

.....

podpis autora

Poděkování

Na tomto místě bych ráda poděkovala vedoucímu práce Ing. Janu Luhanovi, Ph.D., MSc. za rady a trpělivost při vytváření práce. Také bych chtěla poděkovat svému příteli a rodině za velkou podporu během celého studia.

Obsah

1	Úvod	9
2	Vymezení problému a cíle práce	10
3	Teoretická východiska práce	11
3.1	Operační systém Android	11
3.1.1	Architektura	12
3.1.2	Životní cyklus aplikace	14
3.1.3	Vývojové prostředí	17
3.2	Operační systém iOS	17
3.2.1	Architektura	18
3.2.2	Vývoj aplikace	20
3.3	React Native	22
3.3.1	Multiplatformní nástroje	23
3.3.2	Architektura React Native	24
3.3.3	HTML, Javascript a CSS	24
3.3.4	React	26
3.3.5	React Redux	26
3.3.6	HTTP Request, API	27
3.3.7	React Native komponenty	28
3.3.8	React Navigace	29
3.3.9	Vývojové prostředí	29
3.4	Principy vývoje mobilní aplikace	30
3.4.1	Postup vývoje	30
3.4.2	Vzhled mobilní aplikace	33
3.4.3	Marketingový výzkum	37
3.4.4	Monetizace aplikace	38
4	Analýza problému a současné situace	40
4.1	Současná situace	40
4.1.1	Portál Etsy	40
4.1.2	Portál Fler	41

4.2	Dotazník	43
4.2.1	Cílová skupina	47
4.3	Požadavky na mobilní aplikaci	47
4.4	Plán vývoje	48
4.4.1	Analýza rizik	48
4.4.2	Časová analýza	51
4.5	Zhodnocení	52
5	Vlastní návrh řešení	54
5.1	Wireframe	54
5.1.1	Přihlašovací stránka	54
5.1.2	Domovská stránka	54
5.1.3	Stránka s aktuálními přehledy	56
5.1.4	Stránka s historií	56
5.1.5	Další podstránky	57
5.2	Design aplikace	58
5.3	Implementace	59
5.3.1	Prototyp aplikace	59
5.3.2	Použité technologie	62
5.3.3	Ukázky kódu	63
5.4	Možné rozšíření aplikace	66
6	Zhodnocení	67
7	Závěr	68
8	Seznam použitých zkratk a symbolů	73
9	Přílohy	74

1. Úvod

Chytré telefony se staly nedílnou součástí našeho života a využívají je také prodejci rukodělných výrobků. Tato práce se bude zabývat analýzou a návrhem mobilní aplikace, která jim bude zobrazovat podrobné statistiky prodejů a oblíbenosti produktů. Mobilní aplikace bude vytvořena pro platformy Android a iOS a bude zobrazovat informace z portálu Fler.cz, který sdružuje velké množství prodejců.

V úvodu se práce zabývá teoretickými východisky, kde budou porovnány platformy Android a iOS a popisovány možnosti, jak pro ně lze vyvíjet mobilní aplikace. Následně bude popsán framework React Native, díky kterému je možné vyvíjet pro obě platformy zároveň. Také se bude část práce zaměřovat na postup vývoje mobilní aplikace a jejího vzhledu. Následuje popis marketingovému výzkumu a způsobů monetizace aplikace.

Další kapitola se zaměřuje na analýzu současné situace. Budou charakterizovány možnosti prodeje rukodělných výrobků a na základě dotazníku definována cílová skupina a zpracovány požadavky na mobilní aplikaci. Také bude vytvořen časový plán vývoje metodou PERT.

Poslední část se bude zabývat návrhem řešení, kdy bude vytvořen drátěný model aplikace a návrh stylů. Následně bude popsána implementace prototypu aplikace, jaké technologie byly využity a budou představeny ukázky kódu. V závěru budou popsány nápady na rozšíření do budoucna a zhodnocen výsledný prototyp aplikace.

2. Vymezení problému a cíle práce

Prodej rukodělných výrobků online se stále více rozmáhá. Někteří prodejci nabízejí své výrobky na sociálních sítích, jiní mají vlastní eshopy a mnoho z nich se v rámci České republiky sdružuje na portálu Fler.

Tento portál vznikl již v roce 2008 a seskupuje přibližně 16 tisíc aktivních prodejců. Ti své nabízené zboží spravují přímo na webových stránkách portálu. Také již vznikla mobilní aplikace, která umožňuje prodejcům spravovat jejich obchod, ale má omezené funkce.

Na webových stránkách Fleru ani v mobilní aplikaci není možné zobrazit podrobnější statistiky o prodeji, návštěvnosti nebo oblíbenosti obchodu. Je možné najít, kdy se které zboží prodalo, ale tyto údaje jsou zobrazeny pouze v tabulce. Není možné s daty více pracovat, jako např. zobrazovat je v grafech, porovnávat s oblíbeností produktů a podobně.

Aby si prodejce mohl tato data přehledně zobrazit a vyčíst z nich, kterým směrem se má dále ubírat jeho obchodní strategie, bude navržena mobilní aplikace pro podporu správy statistik z prodeje. Tato aplikace umožní prodejci pohled na data jiným způsobem. Bude moci například zjistit, proč se určitému druhu zboží daří více než jinému. Nebo uvidí přehledy nejprodávanějšího a nejpopulárnějšího zboží.

Proč právě mobilní aplikace? Mobilní aplikace by měla být vytvořena pro obě dnes populární platformy Android a iOS. Chytrý telefon máme po ruce pořád n rozdíl od počítače. Když bude prodejce například někde v dílně vyrábět nové zboží, těžko se u toho podívá na počítač, jak se daří jeho dalším produktům.

Aby byl vývoj aplikace rychlejší a jednodušší, bylo by dobré ji vyvíjet pro obě platformy zároveň v nástroji, který toto umožňuje. Cílem práce je tedy návrh a tvorba prototypu aplikace, která bude zobrazovat statistické přehledy o prodejích na portálu Fler, a bude fungovat na obou platformách.

3. Teoretická východiska práce

Tato kapitola se bude zaměřovat na platformy Android a iOS a porovnávat jejich rozdíly. Proč právě tyto dvě platformy? Podle serveru GlobalStats mají aktuálně mobilní zařízení se systémem Android až 71% zastoupení. Na druhém místě jsou právě systémy s iOS, které mají kolem 27 %. [11]

V další části budou charakterizovány možnosti vývoje aplikace pro obě platformy zároveň. Pro tento účel byl vybrán framework React Native, který je blíže popsán.

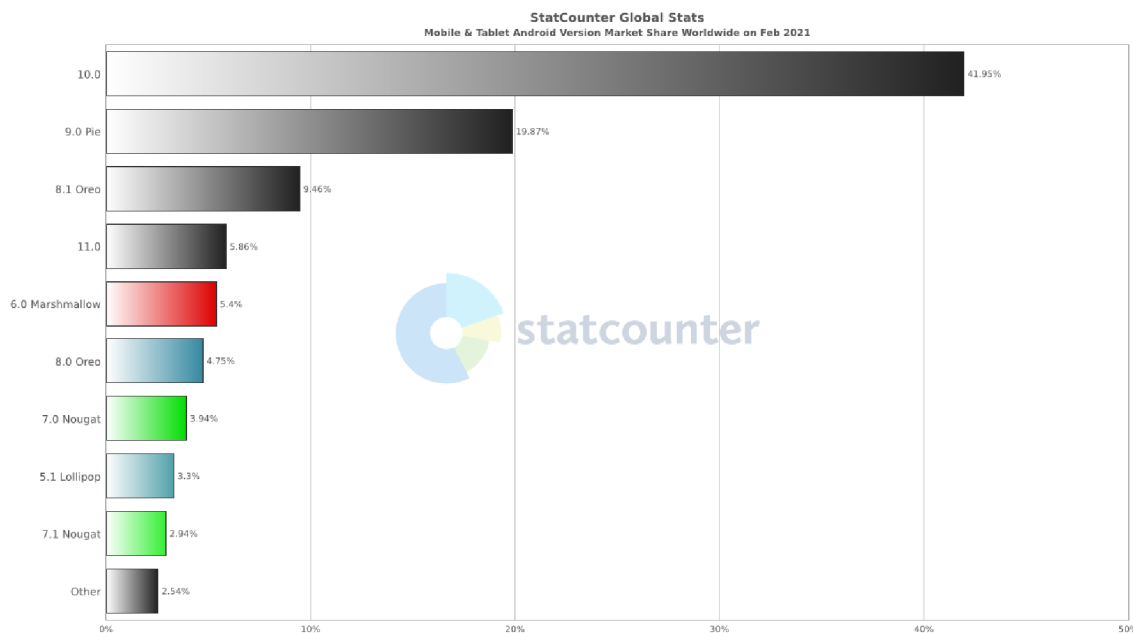
Následně se bude kapitola zabývat principy vývoje mobilní aplikace, vhodným vzhledem aplikace a výběrem grafických prvků. Nakonec je řešeno, jak probíhá marketingový výzkum a jak je možné aplikaci monetizovat.

3.1. Operační systém Android

Operační systém Android je open source software, který pracuje na jádru Linuxu. O jeho vývoj se stará společnost Google a využívají ji různí výrobci mobilních zařízení. Výrobci mohou za dodržení určitých podmínek operační systém upravovat podle svého a odlišit své produkty od konkurence. Díky tomu, že je software volně dostupný, disponuje miliony aplikací, které využívají uživatelé pro každodenní činnosti. [22]

Historie Androidu začíná již v roce 2003, kdy jej založil Palo Alto z Californie. Vytvořil operační systém pro tehdejší chytré telefony. O dva roky později jej koupila společnost Google a v roce 2007 vydala první veřejnou beta verzi 1.0 pod open-source Apache Licencí, což znamenalo, že každý si mohl stáhnout zdrojový kód zdarma. O rok později již vyšel první mobil s Androidem s názvem HTC Dream, který nijak úspěšný nebyl. Měl méně jak 1% zastoupení. V roce 2009 byl vydán Android ve verzi 1.5 Cupcake, čímž začalo pojmenovávání verzí po sladkostech. Následovaly verze se jmény: Donut, Eclair, Froyo, Gingerbread a další. [2]

Postupně se mobilní zařízení se systémem Android dostaly do popředí a v roce 2019 bylo až 2,5 miliardy aktivních uživatelů [35]. Zatím poslední verzí je Android 11, který byl vydán v září 2020. V únoru 2021 byl celosvětově nejrozšířenější Android verze 10.0 viz graf 3.1.



Obrázek 3.1: Zastoupení verzí Androidu v únoru 2021¹

3.1.1. Architektura

Operační systém Android se dělí do několika vrstev. Začíná se jádrem, které poskytuje služby vyšším vrstvám a tak dále. Jednotlivé vrstvy jsou znázorněny na obrázku 3.2 a budou dále popsány. [3]

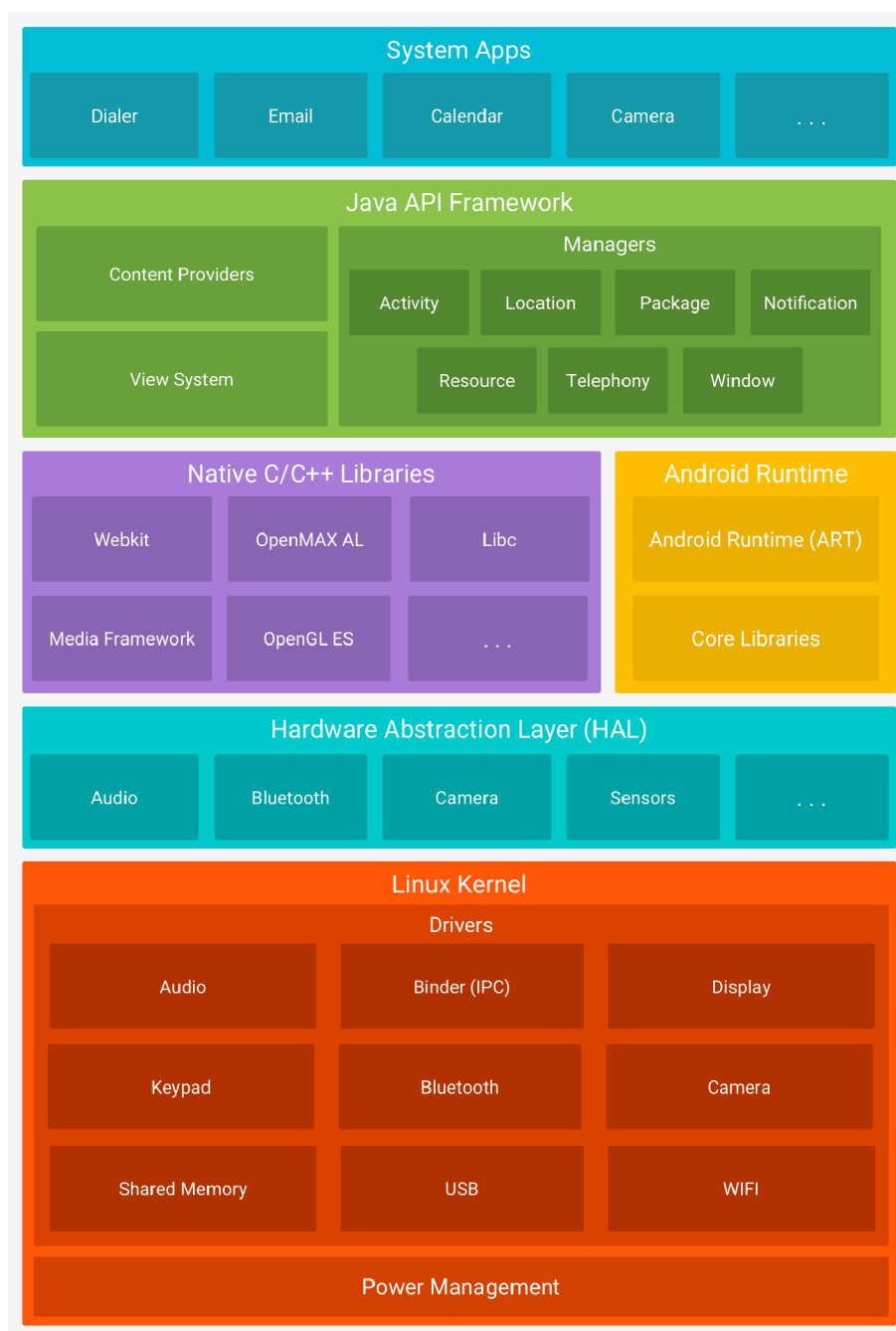
Linux Kernel

Základem architektury a důležitou součástí softwaru je Linux Kernel, což je linuxové jádro. Obsluhuje hardwarové komponenty mobilního zařízení a poskytuje základní operace s nimi. [22]

Hardware Abstraction Layer (HAL)

Vrstva HAL poskytuje standardní rozhraní, která pomáhají vyšším vrstvám s komunikací s hardwarem. Vrstva obsahuje několik knihoven, jež se zabývají jednotlivými komponentami z hardwaru, např. obsluha kamery nebo wi-fi. [3]

¹Převzato z <https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide/#monthly-202102-202102-bar>



Obrázek 3.2: Architektura Androidu [3]

Native C/C++ Libraries

Nativní knihovny napsané v jazycích C a C++ jsou základem pro mnoho komponent a služeb Androidu. Jejich funkcionalita se používá díky Java framework API. Ovládat se dají i přímo z kódu pomocí Android NDK. [3]

Android Runtime

Od verzi Androidu 5.0 a vyšších má každá běžící aplikace svou instanci v Android Runtime, které obsluhuje všechny běžící aplikace.

Java API Framework

Všechny funkce a vlastnosti systému Android jsou dostupné díky API frameworku, který je rozdělen do následujících komponent: [3]

- **View System** se používá pro tvorbu UI (uživatelských rozhraní).
- **Resource Manager** poskytuje přístup například k obrázkům.
- **Notification Manager** umožňuje ovládání upozornění.
- **Activity Manager** řeší životní cyklus aplikace a navigaci.
- **Content Providers** umožňují aplikacím získávat data z jiných aplikací (např. kontakty).

System Apps

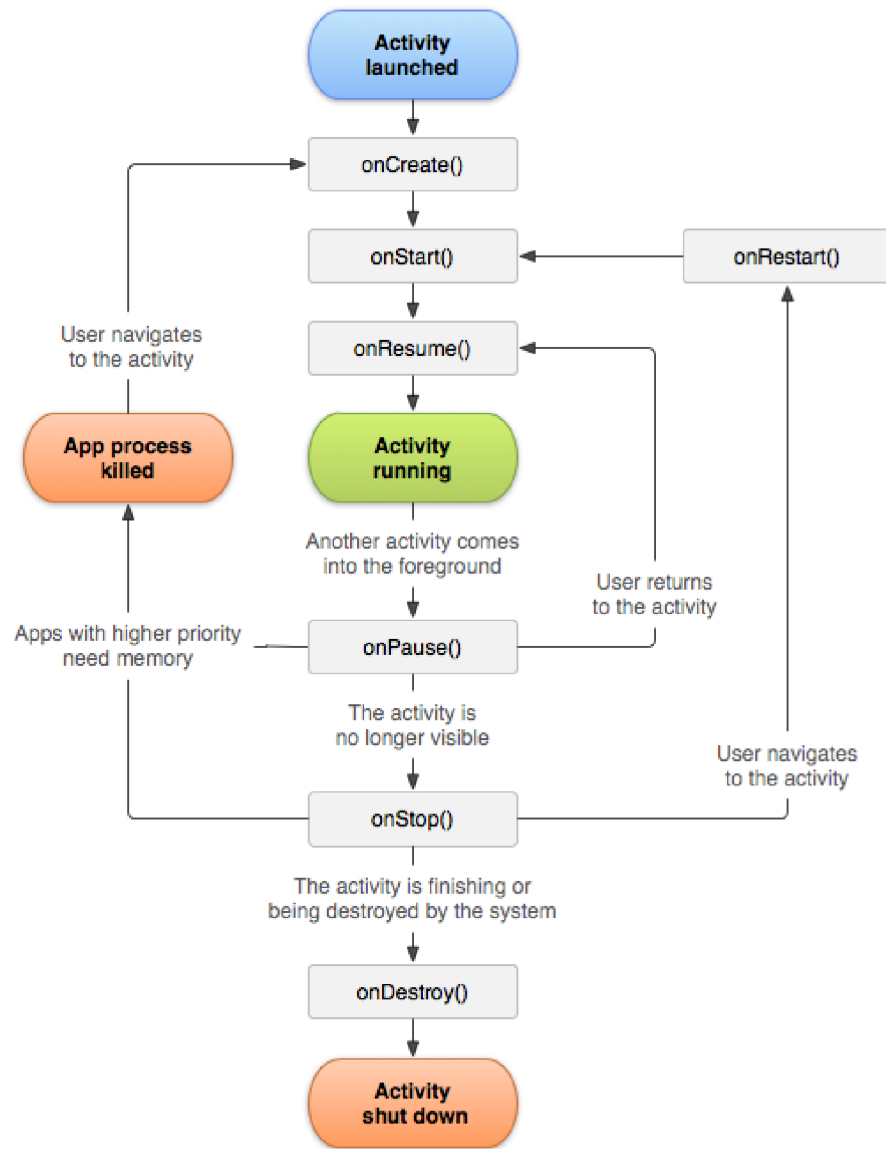
Systémové aplikace jsou základní aplikace přímo od Androidu. Zahrnují email, SMS, kalendář, kontakty, internetový prohlížeč a další. Aplikace se sice nedají ze systému smazat, ale nemusejí být nastaveny jako výchozí. Uživatel si může stáhnout jiné, pro něj lepší aplikace, a ty základní od Androidu vůbec nepoužívat. [3]

3.1.2. Životní cyklus aplikace

Aplikace v Androidu se skládají z přesně definovaných metod, které se spouštějí v předem daném pořadí tak, aby aplikace správně zobrazovala dané prvky pro uživatele. Každý zobrazený stav, ve kterém se aplikace může nacházet, se nazývá aktivita. Může se jednat například o úvodní obrazovku aplikace, zobrazení menu nebo jiné stránky. Celý proces změny aktivit se nazývá životní cyklus aktivit, neboli Activity Lifecycle, a stará se o vytvoření, pozastavení, obnovení nebo zrušení procesu, který aktivitu ovládá. [3]

Díky funkcím životního cyklu je možné v aplikaci naprogramovat, jak se má zachovat, pokud je přerušena příchozím hovorem nebo přechodem na jinou aplikaci.

Může například pozastavit právě běžící video a následně jej obnovit. Následující funkce jsou životním cyklem volány a aplikace na ně reaguje. Zjednodušená vizualizace životního cyklu aplikace je na obrázku 3.3.



Obrázek 3.3: Životní cyklus aktivit [3]

`onCreate()`

Funkci `onCreate()` je nutné v aplikaci naprogramovat, jelikož se vyvolá, když systém spouští danou aktivitu. V metodě se řeší první nastavení aktivity a měla by se volat právě jednou. Jedná se například o získání dat, konfigurace proměnných a objektů

a nastavení jejich zobrazení. Vše zatím probíhá na pozadí a aktivita není viditelná pro uživatele.

onStart()

Hned po metodě `onCreate()` je volána metoda `onStart()`, díky které je aktivita už zobrazena uživateli, ale ještě s ním nemůže interagovat.

onResume()

I tato metoda je zavolána ihned po předchozí. V této části je aktivita přesunuta do popředí a uživatel s ní již může interagovat. V tomto stavu aktivita setrvává až do doby, kdy je něčím přerušena. Například právě příchozím hovorem, změnou aktivity (uživatel se přesune na jinou aplikaci), nebo vypnutím obrazovky.

Metoda `onResume()` může být volána také po metodě `onPause()`, kdy se opět dostává do popředí.

onPause()

Když uživatel opouští danou aktivitu, volá se metoda `onPause()`, což značí, že aktivita již není v popředí a není možné interagovat s uživatelem. V tomto stavu aktivita pozastaví veškeré funkce a čeká na metodu `onResume()`. Nicméně je stále možné, že uživatel aktivitu vidí. Může ji mít jen z části překrytou jinou aplikací, proto se nedoporučuje zásadně funkce stopovat nebo zobrazovat něco jiného.

onStop()

Metoda `onStop()` je volána, když uživatel danou aktivitu už vůbec nevidí. Mohla ji například překrýt nová aktivita nebo původní aktivita už sama končí. V tomto stavu by měla původní aktivita ukončit veškeré své funkce. Je ale stále možné, že bude aktivita znovu vyvolána metodou `onRestart()`, která pokračuje metodou `onStart()`. Aktivita tak začíná téměř od začátku.

onDestroy()

Nakonec je volána funkce `onDestroy()`, díky které může původní aktivita po sobě tzv. uklidit před svým zničením. Následně systém může opět volat `onCreate()` jiné aktivity.

3.1.3. Vývojové prostředí

Oficiálním vývojovým prostředím od společnosti Google pro vývoj aplikací pro Android je Android Studio. Je založené na softwaru IntelliJ IDEA od společnosti JetBrains, která se specializuje na vývoj software pro vývojáře a projektové manažery. [17]

Android Studio bylo poprvé představeno v prosinci 2014 a jeho aktuálně poslední verze 4.1 je z října 2020. Je volně dostupné pod Apache Licence 2.0. Oproti klasickému vývojovému prostředí od JetBrains obsahuje několik dalších vlastností vhodných pro vývoj v Androidu [3]:

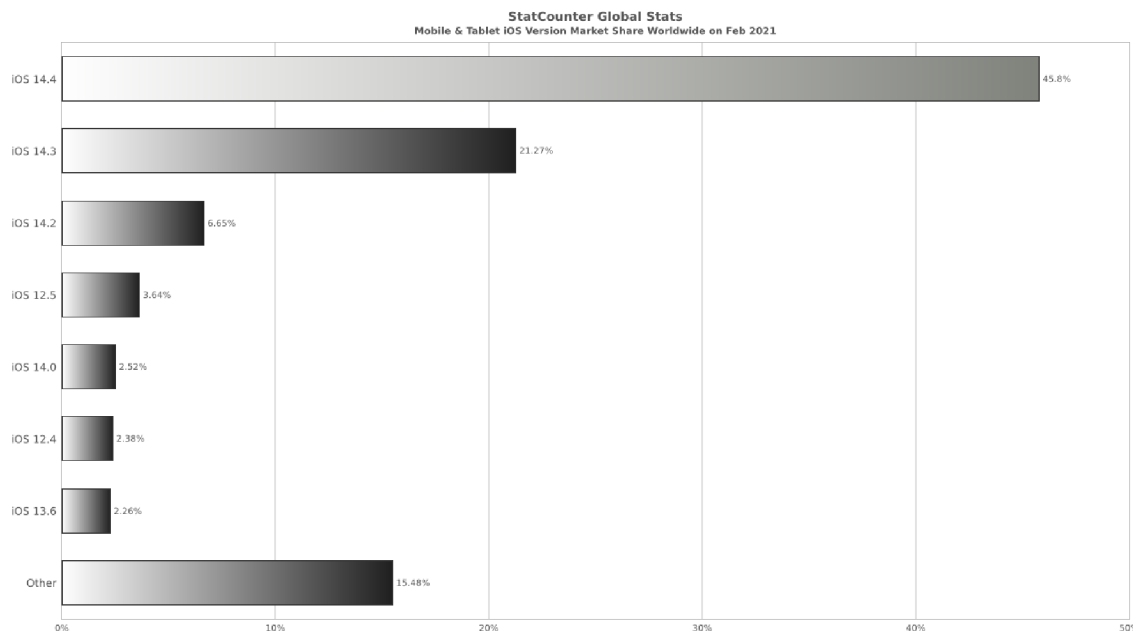
- Flexibilní systém pro kompletaci založený na Gradle.
- Rychlý emulátor s mnoha vlastnostmi.
- Sjednocené prostředí, kde je možné vyvíjet pro všechna Android zařízení.
- Šablony kódu a integrace GitHubu.
- Rozšířené nástroje pro testování a frameworky.
- Nástroje pro kontrolu přehlednosti kódu, využití a kompatibilitu verzí.
- Vestavěná podpora pro Google Cloud Platform, která usnadňuje integraci Google Cloud Messaging a App Engine.

3.2. Operační systém iOS

Operační systém iOS pro mobilní zařízení je vyvíjený společností Apple Inc. exkluzivně pro jejich zařízení, jako jsou iPhone (mobilní telefon) nebo iPad (tablet). Jak už bylo zmíněno výše, jedná se o druhý nejvíce využívaný software v mobilních zařízeních. Apple samozřejmě cílí i na další segmenty a jejich software se objevuje v chytrých televizích (Apple TV) nebo chytrých hodinkách (Apple Watch).

Tento operační systém není volně dostupný oproti Androidu, je proprietární, což znamená, že jeho kód je uzavřený a není možné jej získat, upravovat nebo využívat. Nicméně jsou určité části, které spadají pod open source Apple Public Source Licence. [19]

Historie telefonů od této společnosti sahá do roku 2007, kdy byl představen iPhone první generace. Čísla verzí se mění přibližně jednou za rok. V dnešní době je nejnovější verzí iOS 14.4, která byla vydána v lednu 2021. Celosvětové zastoupení verzí je zobrazeno na obrázku 3.4.

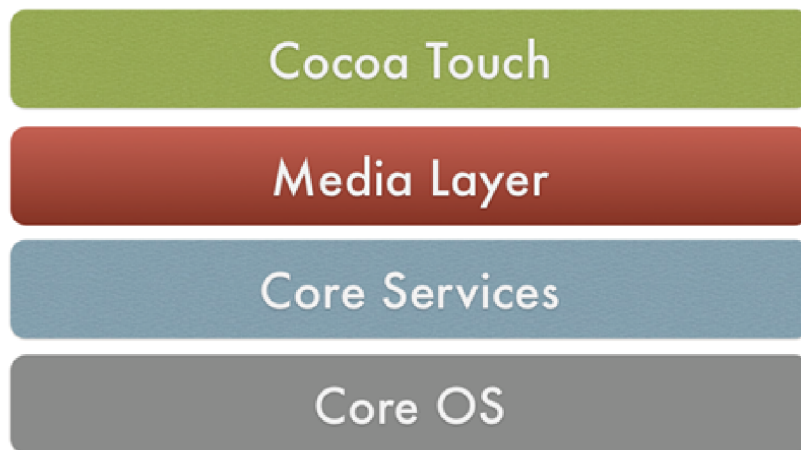


Obrázek 3.4: Zastoupení verzí iOS v únoru 2021²

3.2.1. Architektura

Operační systém iOS, který byl vyvinut pro přenosná zařízení, vznikl podle systému macOS. Jeho architektura je rozdělena do čtyř vrstev: Core OS, Core Services a vyšších vrstev Media a Cocoa Touch. Nad těmito vrstvami jsou ještě samotné aplikace. Pro jejich vývoj se doporučuje používat dvě vyšší vrstvy, které umožňují programovat pomocí lepšího objektově orientovaného přístupu. Pokud ale vrstvy neobsahují požadované funkce, je možné ovládat i nižší vrstvy. Jednotlivé vrstvy budou popsány směrem od nejnižší (jádra) až po tu nejbližší k aplikacím. Pro zajímavost si můžeme všimnout, jak jednoduše je tato architektura (obrázek 3.5) strukturovaná oproti architektuře Androidu (obrázek 3.2). [19]

²Převzato z <https://gs.statcounter.com/os-version-market-share/ios/mobile-tablet/worldwide#monthly-202102-202102-bar>



Obrázek 3.5: Architektura operačního systému iOS [13]

Core OS

Jádro obsahuje funkce, které dále využívají vyšší vrstvy. Většinou se při vývoji aplikací žádné funkce jádra nepoužívají. Najdou se zde frameworky pro ovládání jádra Bluetooth, externích zařízení nebo služby pro bezpečnost a autentifikaci. [19]

Core Services

Tato vrstva obsahuje základní funkce pro aplikace, jako například framework Core Foundation, kde jsou definovány základní typy, framework Core Location, který podává informace o lokaci, nebo Homekit framework, který dokáže komunikovat s dalšími připojenými zařízeními v domácnosti.

Jsou zde umístěny i rozhraní pro komunikaci se službou iCloud, webovými službami, sociálními sítěmi nebo telefonováním a posíláním SMS. [19]

Media Layer

Vrsta Media obsahuje funkce pro obsluhu grafických zobrazení, videa nebo zvuku. Jsou zde umístěny grafické knihovny, jako například Core Graphics, OpenGL, Metal, Photos Library, Media player nebo Core Audio. [13]

Cocoa Touch

Mezi nejdůležitější vrstvu pro vývojáře patří Cocoa Touch, která obsahuje všechny důležité frameworky pro tvorbu aplikací. Jedná se například o EventKit, který se

stará o události v kalendáři. Framework GameKit zase obsahuje podporu Game Center a stará se o sdílení úspěchu ve hrách. Framework iAd obsluhuje reklamy a framework UIKit podporuje zobrazení grafiky, multitasking, notifikace a další. [13]

3.2.2. Vývoj aplikace

Pro vývoj aplikace je dobré znát architekturu MVC, kterou využívají i frameworky. Architektura nám pomáhá rozdělit aplikace do lépe od sebe oddělitelných částí. Dále je potřeba znát objektově orientované programování a jazyk Swift, jenž byl vyvinut jako nástupce předchozího jazyka Objective-C. Pro psaní kódu bylo od společnosti Apple vyvinuto vývojové prostředí Xcode, které umožňuje snadný vývoj a simulaci aplikace.

MVC

Model-View-Controller je softwarová architektura aplikace a využívá se i u vývoje iOS aplikací. Jedná se o rozdělení funkcí do tří komponent tak, aby změna jedné z nich neměla téměř žádný vliv na ostatní komponenty [21]:

- **Model** poskytuje a edituje informace v databázi.
- **View** vykresluje data uživateli.
- **Controller** reaguje na změny v aplikaci, aktualizuje data přes Model a následně je zobrazuje přes View.

Proč používat MVC? Mnoho iOS frameworků MVC využívá (např. UIKit), navíc je kód lépe čitelný, snadno se v něm dělají změny a většina programátorů jej zná.

Objective-C

Tento objektově orientovaný jazyk je rozšířením jazyka C a byl primárně používán pro vývoj aplikací pro iOS, než jej nahradil Swift. Byl vyvinut již v osmdesátých letech a programování v něm bylo na dnešní dobu poměrně náročné a zdlouhavé. [4]

Swift

Programovací jazyk Swift byl poprvé představen v roce 2014 jako lepší verze jazyků C a Objective-C. Jelikož se zrušila kompatibilita s jazykem C, jazyk Swift nemá tolik omezení, využívá moderní přístupy a je proto více flexibilní a méně náročný. Říká se mu také „Objective-C without the C“ a byl vytvořen jako primární jazyk pro vývoj v iOS. [4]

Velkou výhodou jazyka Swift je, že je vytvářen jako open source, proto se neustále vyvíjí a je hojně podporován pro vývoj nových aplikací pro iOS. Také je rychlejší, bezpečnější a jednodušší. Nicméně projekt se dá stále programovat z části v jazyce Objective-C a z části v jazyce Swift, takže nenutí vývojáře přepisovat starší kód. [4]

Vývojové prostředí

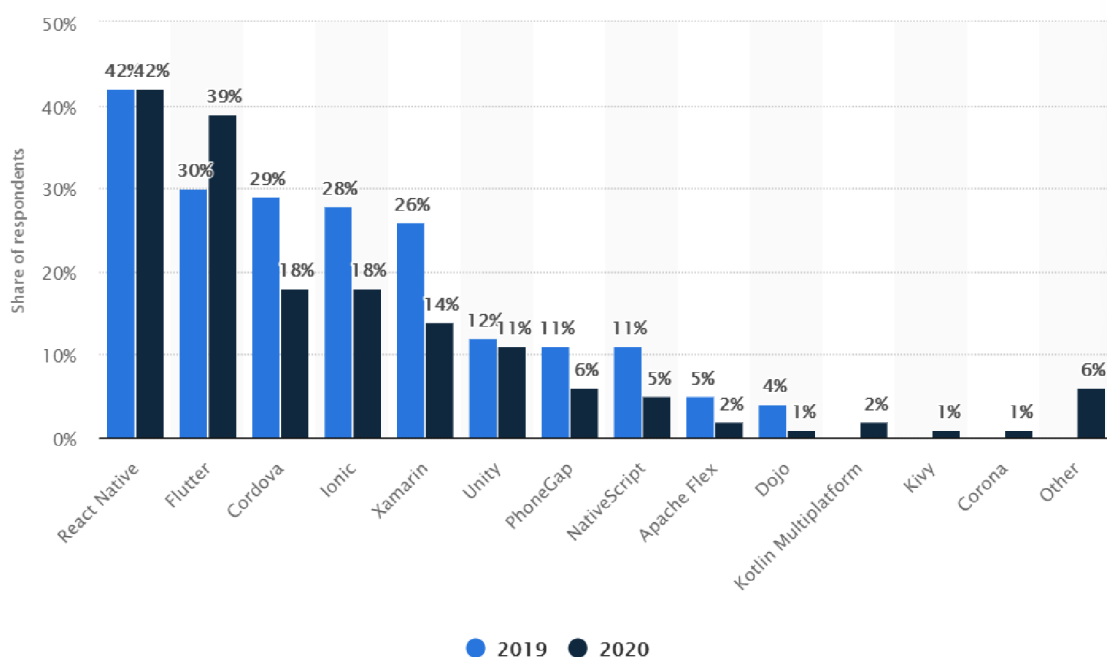
Xcode je IDE (Integrated Development Environment = vývojové prostředí) poskytnuté společností Apple pro všechny programátory vyvíjející aplikace pro iOS. Programuje se zde ve výše popsaných jazycích Objective-C a Swift. Xcode obsahuje následující nástroje pro vývoj [20]:

- **Code Editor** je textový editor pro psaní kódu.
- **Compiler** je program, který kód překládá do nativního jazyku pro procesor, aby byl spustitelný.
- **Debugger** pomáhá při hledání chyb.
- **Interface Builder** se využívá pro tvorbu UI (User Interface = uživatelské rozhraní).
- **Application Bundler** vytváří aplikační balíček, který se dá například nahrát do App Store.
- **Simulator** je program, který simuluje chování aplikace v mobilním zařízení a je možné aplikaci testovat.

3.3. React Native

Aplikaci je potřeba využívat v obou zařízeních Android i iOS. Pro kratší a levnější vývoj by bylo dobré najít nástroj, který dokáže vytvářet aplikace pro obě platformy zároveň. V dnešní době již existuje mnoho takových nástrojů. Mezi nejpopulárnější patří React Native, Flutter a Cordova, jak je vidět v grafu 3.6.

Aktuálně nejpopulárnější framework React Native byl vyvinut společností Facebook na základě klasického frameworku React, který je založen na jazyku Javascript a vytváří se s ním dynamické webové stránky a aplikace. React Native dodržuje principy Reactu a umožňuje podobným způsobem programovat aplikace pro mobilní zařízení, jako jsou Android nebo iOS. Je vydáván pod open source licencí, tudíž je volně dostupný a na jeho vývoji se podílí velká komunita programátorů.



Obrázek 3.6: Multiplatformní nástroje využívané v letech 2019 a 2020³

Výhodou frameworku React Native je využitelnost kódu ve více platformách až na 80 %. Framework také umožňuje simulovat výslednou aplikaci hned při vývoji, což jej značně urychluje. React Native se také zaměřuje na responzivní zobrazení aplikace, což znamená, že se umí přizpůsobit různým velikostem obrazovek mobil-

³Převzato z <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>

ních telefonů. Poslední výhodou může být, že umí využívat nativní kameru nebo akceleraci. [28]

Mezi nevýhody frameworku patří určitá omezení při využívání, například právě kamery, kdy je potřeba tuto část kódu programovat nativně a psát ji proto pro každý operační systém zvlášť. Také vývoj frameworku je občas pozadu za vývojem nových nativních funkcí iOS a Android. [28]

Znamé aplikace vyvíjené pomocí tohoto frameworku jsou např. Instagram, Pinterest nebo Skype.

3.3.1. Multiplatformní nástroje

Multiplatformní nástroje (Cross-Platform frameworks) mají řadu výhod. Převážně umožňují vyvíjet mobilní aplikaci, která bude kompatibilní s operačními systémy iOS i Android zároveň. Aplikace se programuje pouze v jednom prostředí, což šetří čas a urychluje i nasazování dalších verzí. [28]

Mezi vývojem v nativním prostředí a vývojem s multiplatformními nástroji se již po několik let vedou dohady o tom, co je lepší. Stále se používají a vyvíjí obě možnosti. Multiplatformní nástroje byly dříve vyvinuty převážně pro tvorbu jednodušších aplikací, ale s postupem času a vývojem technologií se to změnilo a je možné vytvářet i složitější a mocnější aplikace. Nicméně multiplatformní nástroje stále čelí určitým omezením, jako jsou výkyvy ve výkonu nebo méně zabezpečená data. [28]

Mezi výhody multiplatformního vývoje patří:

1. **Maximální dosah k uživatelům** - Díky jednoduššímu vývoji pro více zařízení zároveň je potenciální dosah mnohem vyšší, než kdyby se aplikace objevila např. jen v App Store.
2. **Snížení ceny za vývoj** - Jelikož se může aplikace psát jen jednou a využívat ve více platformách, snižuje to i cenu za vývoj.
3. **Jednodušší údržba** - Vývoj i údržba aplikace je pro programátory jednodušší, jelikož nemusí řešit více variant, ale programují jen v jednom nástroji.
4. **Rychlejší vývoj** - Jelikož je vývoj jednodušší, je zároveň i rychlejší.

5. **Znovu použitelný kód** - Kód se může opakovaně používat pro více druhů zařízení.
6. **Rychlejší nasazování** - S jednodušším vývojem souvisí i jednodušší nasazování verzí do obchodů.
7. **Sjednocený vzhled** - Jelikož se aplikace píše jen v jednom prostředí, vzhled aplikace se v různých platformách nemění.

3.3.2. Architektura React Native

Základní architektura React Native vypadá následovně: Nejprve je vyvolána událost (např. klik na tlačítko v menu), následně jsou zpracována data (jaké to bylo tlačítko), potom je událost zpracována v Javascriptu, který zavolá nativní metody. Metody požadavek zpracují a zobrazí aktualizovanou verzi aplikace (správnou obrazovku z menu). [28]

Popsané rozdělení je znázorněno na obrázku 3.7. V části Native jsou funkce, které řeší zobrazení aplikace uživatelům a události. Javascript je ta část, kterou programujeme a která vykonává všechny funkce v pozadí. Bridge se stará o spojení těchto dvou částí, právě kvůli tomu, že část Native může sloužit pro Android i iOS.

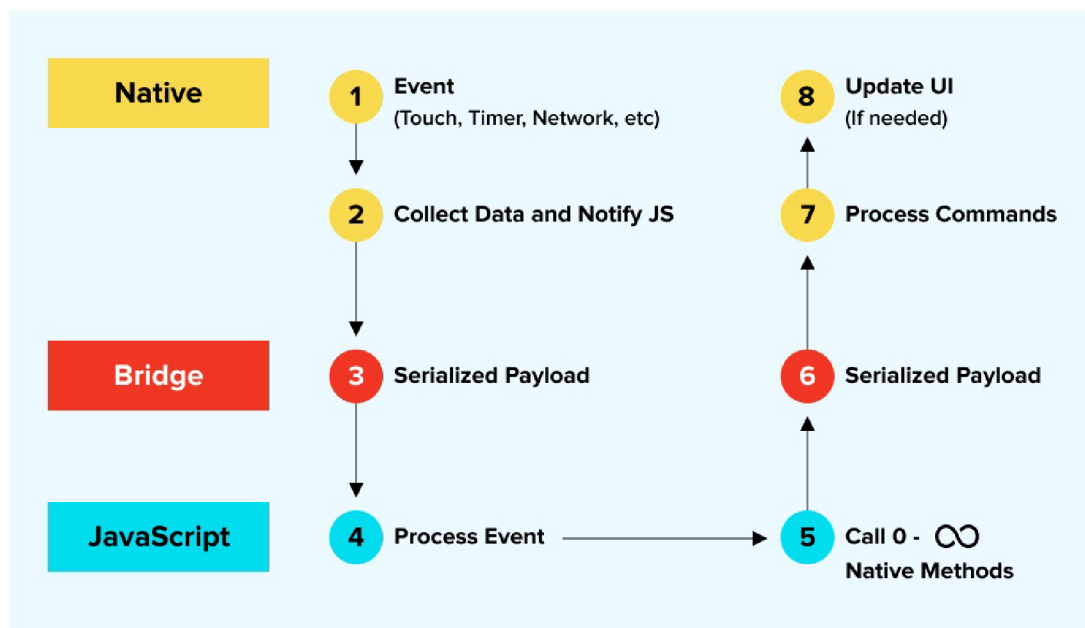
3.3.3. HTML, Javascript a CSS

Framework React Native staví na základech jazyků HTML a Javascript, které se běžně využívají pro vývoj webových stránek a aplikací. CSS slouží pro definování stylů, jako jsou barvy písma nebo velikosti obrázků.

Jazyk HTML (Hyper Text Markup Language) je značkovací jazyk, díky němuž je možné vkládat na webovou stránku texty, obrázky, videa, tabulky, tlačítka a další. Každý prvek (element) je definován pomocí značek (tagů) a může obsahovat různé atributy. [29]

Skriptovací jazyk Javascript se používá pro dynamické změny webové stránky a existuje již od devadesátých let 20. století. Dynamickými změnami se myslí různé animace obrázků, interakce s tlačítky, výběry nebo textová pole. Je možné je psát přímo do HTML kódu a program se vykonává až na straně prohlížeče u klienta. Javascript je interpretovaný jazyk, tudíž se nemusí kompilovat. Využívá objektové

Basic Overview of React Native Architecture



net solutions

Obrázek 3.7: Architektura React Native [28]

programování a je case sensitive, což znamená, že závisí na velikosti písmen. Syntaxí je podobný jazykům C a Java, ale to je vše, co má s jazykem Java společné, i když mají podobné názvy. [30]

Jeho velkou nevýhodou je, že typy proměnných přiřazuje sám dynamicky a programátora proto nehlídá, aby například řetězec náhodou nepoužil při sčítání. Programování v samotném Javascriptu je tak poměrně náročné. Kvůli typování proměnných, definici tříd a rozhraní byl vyvinut Typescript, což je nástavba Javascriptu. Typescript pomáhá k přehlednosti kódu a menší chybovosti. Jelikož je to nástavba, musí se program následně kompilovat zpět do Javascriptu, aby jej prohlížeče uměly spustit. [15]

3.3.4. React

React byl vyvinut společností Facebook a je to framework založený na Javascriptu. Podobně jako React Native je open source a vytváří se s ním frontend webových stránek, což znamená to, jak stránky vypadají a jak se dynamicky mění.

V Reactu se kódují klasické dynamické funkce jako v Javascriptu, ale zároveň se díky němu vytváří i celá HTML stránka, všechny její elementy a styly. Elementy se zapisují v tzv. JSX, což je podobný zápis jako klasické HTML a Javascript, ale umožňuje funkce navíc.

Kód v Reactu se skládá převážně z komponent, které generují většinou určitý element. Může to být třeba jen tlačítko, ale také celá tabulka nebo seznam jiných komponent. Každá komponenta může mít své proměnné, které se ukládají do tzv. *state* a při jeho změně se sama automaticky překresluje. Pro nastavení, kdy se komponenta překreslí, slouží následující funkce:

- `shouldComponentUpdate()` umožňuje nastavit, za jakých podmínek se má komponenta překreslit. Nemusí to být totiž pokaždé, když je hodnota proměnné změněna.
- `componentDidMount()` je funkce, která je zavolána ve chvíli, když se komponenta objeví pro uživatele. Využívá se např. pro zavolání funkcí na načtení dat z API.
- `componentWillUnmount()` je funkce, která je zavolána těsně předtím, než je komponenta překreslena. To umožňuje např. ukončit funkce, které můžou ještě běžet a něco vykonávat.
- `render()` nastavuje, co se vykresluje. Zde se musí vložit veškeré prvky, které mají být zobrazeny. Tato funkce je povinná pro každou komponentu a pokud není definováno jinak, volá se při každé změně *state* (proměnných).

3.3.5. React Redux

Redux vznikl jak nástavba Reactu a slouží pro správu dat v programu. V klasickém Reactu může mít každá komponenta své proměnné ve *state* a ještě je sdílet s jinými

komponentami. Přehlednost dat a ovládání změn proto může způsobovat problémy. Redux všechna data v aplikaci ukládá na jedno místo, do tzv. *store*, a tím umožňuje ovládat změny dat jednodušeji a praktičtěji. Jelikož ale s touto změnou přichází i trochu komplikovanější programování, doporučuje se Redux používat spíše u větších aplikací. [23]

Redux rozděluje správu dat do tří částí, které se o jejich změny starají:

- **Actions** slouží pro správu událostí v aplikaci. Kdykoliv se v aplikaci něco stane, např. uživatel klikne na tlačítko, zavolá se příslušná událost z Actions.
- **Reducers** podle příslušné události z Actions upravují data ve Store.
- **Store** je místo, kde jsou uložena všechna aktuální data aplikace.

3.3.6. HTTP Request, API

Pro získávání dat ze serveru je dobré znát HTTP protokol (Hypertext Transfer Protocol), který slouží pro komunikaci s webovými servery. Umožňuje přenášet data ve formátech HTML nebo XML a obvykle používá pro komunikaci TCP protokol. Funguje na jednoduchém principu, kdy po přijetí dotazu pošle odpověď. Protokol se využívá pro načítání celých webových stránek. [26]

Velkým využitím tohoto protokolu je i komunikace s API (Application Programming Interface), což označuje rozhraní, které obsahuje určité funkce, jež se dají využívat externě. Často se jedná o poskytování dat.

HTTP definuje určité metody, které je možné využívat [26]:

- **GET** je metoda určená pro klasické získání dat.
- **HEAD** se využívá podobně jako metoda GET, ale vrátí se jen hlavička odpovědi bez dat.
- **POST** posílá data na server.
- **PUT** je metoda podobná jako POST, ale je idempotentní, což znamená, že v případě, kdy přijde více stejných požadavků za sebou, výsledek bude vždy stejný.

- **PATCH** je opět podobná jako metody PUT a POST, ale s rozdílem, že data jen částečně upravuje.
- **DELETE** je metoda, která může mazat určitá data.
- **CONNECT** vytváří síťové spojení k serveru, které se nazývá tunel.
- **TRACE** je metoda pro testování spojení a zjišťování, přes jaké servery se k danému serveru dostáváme.

3.3.7. React Native komponenty

Kód v React Native se skládá z komponent, které fungují na stejném principu jako v klasickém Reactu. Každá komponenta může být třeba jen tlačítko nebo může obsahovat více různých komponent. React Native komponenty ale upravuje podle operačního systému, pro který aplikaci právě kompiluje. Komponenty, které se dají využít právě pro obě platformy, se nazývají *Core Components*. Několik základních komponent a jejich nativní alternativy jsou uvedeny v tabulce 3.1. Je možné také vytvářet vlastní nativní komponenty nebo využít komponenty vytvořené jinými vývojáři. [32]

Popis	React Native	Android	iOS
Základní bloková komponenta	<View>	<ViewGroup>	<UIView>
Textová komponenta	<Text>	<TextView>	<UITextView>
Zobrazení obrázků	<Image>	<ImageView>	<UIImageView>
Sdružení více komponent	<ScrollView>	<ScrollView>	<UIScrollView>
Vložení textu uživatelem	<TextInput>	<EditText>	<UITextField>

Tabulka 3.1: Příklad základních komponent v React Native, Android a iOS [32]

Stejně jako React, také React Native využívá syntaxi JSX, ve které se vytvářejí veškeré komponenty. Nechybí ani již známý State, do kterého se ukládají data. Díky této funkci se při jakékoliv změně dat komponenta sama aktualizuje a zobrazí se podle aktuálního stavu. Změna State se volá pomocí funkce `useState()`. [32]

Design aplikace a jejích komponent se dá také zapisovat přímo u daných komponent v JSX. Využívá se k tomu atribut *style* a jeho vlastnosti jsou založené na jazyku CSS. Například známá CSS vlastnost `background-color` se tady nazývá `backgroundColor`, což je stejný název, jen zapsán ve stylu „camel case“. Pokud je

stylů víc, jdou hromadně napsat pomocí funkce `StyleSheet.create()`, kde se styly vypíší podobně jako v CSS. [32]

Také je potřeba při vytváření layoutu aplikace myslet na responzivitě. Jedná se o vlastnost, která zaručuje, že aplikace bude na všech různých velikostech mobilních zařízení vypadat dobře. Nesmí se tudíž např. výška a šířka aplikace nastavovat v pixelech, ale v procentech. Kvůli tomu, aby se komponenty zobrazovaly na všech zařízeních dobře, se využívá *Flexbox*. [32]

Flexbox je skupina CSS vlastností, kterými se dá nastavit, aby layout splňoval požadavky na responzivní zobrazení. Dá se nastavit, jak budou jednotlivé části velké oproti ostatním, jak půjdou za sebou, jestli se budou rovnat doleva, na střed nebo doprava a další. [6]

3.3.8. React Navigace

K React Native patří také knihovna, která obsahuje funkce pro navigaci v aplikaci, která obvykle potřebuje zobrazovat více stránek neboli obrazovek. React Navigation poskytuje základní funkce pro nastavení více obrazovek. Vytvoří se tzv. *NavigationContainer*, ve kterém se nadefinují jednotlivé obrazovky, jež dále obsahují vlastní komponenty. [32]

Umožňuje nastavit navigaci přes záložky, které jsou zpravidla zobrazeny ve spodní části obrazovky. Nebo je možné vytvořit klasické menu, jež vyjíždí ze strany. Navigace funguje na obou platformách Android i iOS. [32]

3.3.9. Vývojové prostředí

K programování je možné využívat jakýkoliv editor kódu. Pro kompilování je potřeba mít Node.js, což je knihovna pro kompilování Javascriptu, a využívá se například i pro kompilování klasického Reactu. Následně je potřeba nainstalovat balíček Expo CLI, díky němuž je možné aplikaci simulovat přímo ve webovém prohlížeči, ale dá se také jednoduše zobrazit v mobilním telefonu. Zde je potřeba nainstalovat aplikaci Expo Go, která se přes bezdrátovou síť připojí k počítači a díky QR kódu danou aplikaci zobrazí. [32]

3.4. Principy vývoje mobilní aplikace

Vývoj mobilní aplikace zahrnuje úspěšné zvládnutí postupu vývoje, který obsahuje strategii, analýzy a plánování vývoje. Dále návrh designu aplikace, který má také několik fází od návrhu dat a barev až po prototyp. Následně se aplikace implementuje a testuje a po úspěšném dokončení všech oprav konečně nasazuje.

3.4.1. Postup vývoje

Aby byla aplikace úspěšná, je dobré dodržovat postup vývoje, který zahrnuje následujících 6 fází:

1. Strategie

První fází je definování strategie, jak vytvořit úspěšnou aplikaci. Je potřeba převážně určit, na jaké uživatele bude aplikace cílit. K tomu slouží různé metody sběru dat, které budou více rozvedeny v další části. Také je potřeba zjistit, jaká je konkurence na trhu a k čemu bude nová aplikace sloužit, jaké budou hlavní cíle. Nakonec je potřebné zvolit, pro jaké platformy bude aplikace dostupná. [14]

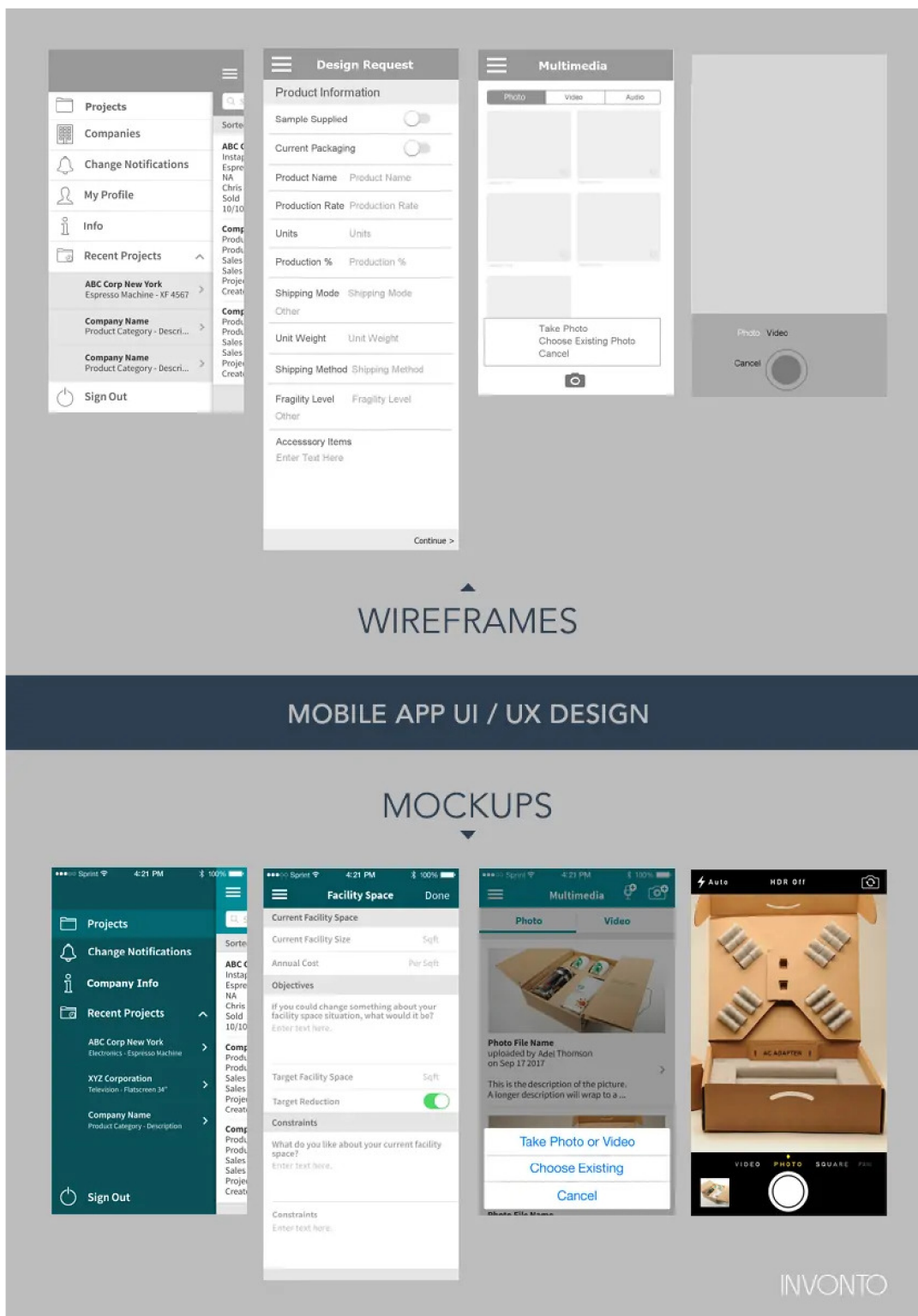
2. Analýza a plánování

V této fázi se naplánuje veškerý proces výroby aplikace. Od definování minimálního produktu, který již bude moci být spuštěn do produkce, až po plánování jednotlivých částí vývoje. Bylo by dobré již mít jméno budoucí aplikace, tým vývojářů, který bude aplikaci programovat, a samozřejmě časovou analýzu a analýzu rizik. [14]

3. UX Design

Jak bude aplikace vypadat a jak budou rozvrženy její komponenty, patří mezi nejdůležitější, protože to ovlivňuje, jestli ji uživatelé budou chtít používat. Zkratka UX znamená User Experience a řeší právě to, jestli je aplikace intuitivní a uživatelsky přívětivá.

Postup pro vývoj designu aplikace je následující. Nejprve je potřeba vymyslet, jaká data se budou uživateli zobrazovat a co s nimi bude možné provádět. Využívají se k tomu vývojové diagramy, které přesně popisují postup, jak se v aplikaci bude uživatel pohybovat a co bude moci dělat. [14]



Obrázek 3.8: Převedení Wireframe na Mockup [14]

Následně se vytváří tzv. **Wireframe** neboli drátěný model, kdy se jednotlivé komponenty aplikace rozmístí. Neřeší se zde vůbec barvy a design jednotlivých prvků, ale pouze jejich velikost a rozvržení na obrazovce. Jedná se proto jen o di-

gitální podobu nákresů aplikace. Wireframe by měl být vytvořen i pro více velikostí obrazovek. Aplikace bude jinak vypadat na malém telefonu a jinak na velkém tabletu. [14]

Pro definování stylu aplikace se určuje tzv. **Style Guide**, což je návod, jak se bude aplikace stylovat. Zde se definuje například styl písma nebo barevné schéma aplikace. [14]

Další fází je převedení Wireframe do **Mockup**, kde se už jednotlivé prvky stylují podle Style Guide. Stále jde pouze o to, jak aplikace bude vypadat a s jednotlivými prvky se nedá nijak interagovat. Jak to může vypadat, je znázorněno na obrázku 3.8. [14]

Poslední částí je vytvoření **prototypu**. Díky němu je už možné simulovat, jak se aplikace bude ovládat. Tvorba prototypu bývá časově náročnější, ale může se vyplatit. Prototyp už je možné testovat a odhalit tak nedostatky aplikace ještě před samotným vývojem. [14]

4. Implementace aplikace

Před samotným programováním aplikace je potřeba vytvořit plán, který by měl obsahovat, jaké technologie se budou využívat, v čem se aplikace bude programovat a hlavně milníky vývoje, které se budou sledovat. Klasická aplikace se skládá ze tří částí: serverové části (backend), zobrazované části zákazníkovi (frontend) a API nebo metody, jak spolu předchozí dvě části budou komunikovat. [14]

Přední část aplikace, to co uvidí klient, by měla umět pracovat i bez internetu, tudíž bez přístupu k serveru a mít uložena určitá data i lokálně. Jak už bylo zmíněno, je mnoho variant, jak tuto část programovat. Ať už v Javě pro Android, v jazyce Swift pro iOS nebo v multiplatformních jazycích jako např. React Native (kapitola 3.3).

5. Testování

Testování může probíhat již po vytvoření prototypu a následně i při vývoji aplikace, kdy mohou být testovány jednotlivé milníky a již vytvořené části aplikace. Pro správné testování je nutné mít vytvořené testovací scénáře, kde by mělo být popsáno, jak se má v dané situaci aplikace zachovat. [14]

Dále je dobré zařadit testování přímo na uživateli, které se většinou provádí až po vytvoření celé aplikace. Náhodný uživatel z cílové skupiny zkouší aplikaci a popisuje, čemu rozumí a čemu ne.

Testování je možné rozdělit na několik druhů [14]:

- **Funkcionální testování** je možné již při vývoji aplikace, kdy se testují jednotlivé části a funkce aplikace. Můžou se zjistit chyby při vyplňování formulářů nebo zadávání jiných dat od uživatele. Pokud se aplikace vytváří pro platformy Android i iOS, je potřeba také testovat, zda všechny funkce pracují a zobrazují se na obou platformách stejně.
- **Testování výkonu** aplikace je potřebné pro zjištění, jak dobře bude aplikace zvládat nápor více uživatelů zároveň. Jak rychle se bude načítat, jestli se nebude sekát nebo kolik místa zabírá v paměti.
- **Testování bezpečnosti** zjišťuje, zda je možné se do aplikace nabourat, získat důvěrná data nebo je podvrhnout. Aplikace s přihlašování uživatele by měly mít tuto část řádně zabezpečenou.
- **Testování na různých zařízeních** zjišťuje, jak aplikace funguje na různých telefonech a platformách. Mobilní operační systémy se aktualizují téměř každý měsíc, takže je potřeba aplikaci udržovat.

6. Nasazení a podpora

Poslední fází je nasazení nové aplikace do produkce. Mobilní aplikace se dávají do obchodů Apple App Store a Google Play Store, ze kterých si je uživatelé můžou stáhnout. Následně se i přes tyto obchody mohou vydávat další verze a aktualizace aplikací. Pro nasazení aplikace do obchodu jsou vyžadovány informace jako název aplikace, popis, kategorie, klíčová slova, ikona a výstřižky z aplikace. [14]

3.4.2. Vzhled mobilní aplikace

Nedílnou součástí vývoje mobilní aplikace je návrh jejího vzhledu. Návrh barev v aplikaci, stylu písma, menu, tlačítek, textových polí, tabulek, grafů a dalších prvků.

Jak už bylo zmíněno v předchozí části, definice všech stylů se zapisují do Style Guide, který slouží jako dokumentace.

Barvy

Vhodně použité barevné schéma je velmi důležité, jelikož každá barva na člověka působí jinak. Vnímání barev ovlivňuje mnoho faktorů a můžeme je dělit do čtyř kategorií: fyzikální, fyziologické, psychologické a vizuální. [7]

Z fyzikálního hlediska je barva elektromagnetické vlnění, které lidské oko vnímá v rozmezí přibližně 390 nm až 700 nm. Fyziologické vnímání určuje účinnost světla na lidský organismus a jaký vliv následně mají. Některé pozitivní a jiné negativní. Z psychologického hlediska se zahrnují asociace, které jsou v lidech vyvolány na základě barvy. Například červená barva symbolizuje krev, oheň a srdce, což může vyvolat pozitivní emoce jako lásku nebo energii, ale naopak i negativní emoce jako nebezpečí a válku. Posledním vizuálním vnímáním se myslí, jak barvy vnímáme na velké ploše. Teplejší barvy většinou vystupují do popředí a studené zase ustupují. [7]

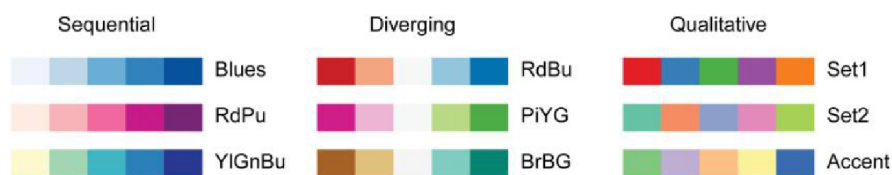


Obrázek 3.9: Iluze vnímání barev vůči pozadí [16]

Důležitým aspektem je také fakt, že barvy jsou vnímány odlišně v závislosti na okolní barvy. Je možné se o tom přesvědčit na obrázku 3.9, kde jsou zobrazeny dvě řady obdélníků stejné barvy, ale i když víme, že každá řada obdélníků má stejnou barvu, vzhledem k měnícímu se pozadí stejné nevypadají. Tohoto aspektu se dá využít, když chceme určité nadpisy nebo části grafů zvýraznit. Například červená

barva bude oproti světle modrému pozadí vyčnívat, ale mezi růžovými odstíny se téměř ztratí. [16]

Pro vytváření barevných kombinací a palet se barvy můžou rozlišovat podle sytosti, jasu a odstínu. Když se paleta převede do odstínů šedi, barvy by stále měly být od sebe rozeznatelné. Pro tvorbu takových palet na internetu existuje mnoho nástrojů, například ColorBrewer⁴ palety rozděluje podle typu dat na sekvenční, divergentní a kvalitativní (viz obrázek 3.10). [16]



Obrázek 3.10: Rozdělení palet podle ColorBrewer [16]

Design

Jak budou jednotlivé prvky a části v aplikaci vypadat, se definuje v Style Guide. Tento návod pro vytvoření uživatelského rozhraní obsahuje pravidla, jak mají vypadat grafické prvky. Využívá se jak pro návrh webových a mobilních aplikací, tak pro styly značky nebo například v časopisech.

Dobrý návod stylů by měl být velmi specifický a detailně zpracovaný, organizovaný, stále aktuální a dobře pochopitelný. Style Guide je důležitý pro návrh aplikací z několika důvodů. Zaprvé by měl udržovat grafickou reprezentaci aplikace stejnou, i když se aplikace bude postupně vyvíjet. Měl by obsahovat instrukce, jak vytvářet nová zobrazení a jaké postupy dodržet, aby nový obsah stále pasoval ke starším částem. Také se může měnit tým vývojářů, kteří aplikaci spravují, tudíž návod stylů by měli používat i noví pracovníci a udržet aplikaci ve stejném stylu. A v neposlední řadě aplikace často obsahují mnoho různých stavů, do kterých se můžou dostat. Není tedy možné vytvořit mockupy pro všechny stavy, a proto by měl Style Guide obsahovat návody, jak by měly jednotlivé prvky vypadat. [5]

Mezi prvky, které by měl návod obsahovat, patří [5]:

- **Fonty** aneb styl písma, jeho velikosti a použití.

⁴Dostupné na <http://colorbrewer2.org>

- **Barvy**, jejich RGB hodnoty a různé odstíny bare. Často se využívají dvě barevná schémata pro světlý a tmavý vzhled aplikace.
- **Layouty**, které zahrnují rozdělení stránek na menu, sloupce nebo různé lišty.
- **Grafické prvky**, jako jsou ikony, grafy nebo animace.
- **Komponenty**, do kterých řadíme např. tlačítka, posuvníky, textová pole, tabulky atd.
- **Vyskakovací okna**, jako jsou dialogová okna nebo upozornění.

Jako základ pro vytvoření návodu stylů se dá využít Material Design, což je minimalistický návrhový systém vyvinutý společností Google. Material Design definuje rámec, který je potřeba dodržovat, a radí, jak navrhovat styly, rozvržení, komponenty a vzory. Výraz material vyjadřuje metaforu, že zobrazení na obrazovce by stále měla být realistická. Téměř jako papír a inkoust. [25]

Pro využití Material designu v programovacím jazyku React Native je volně dostupná např. knihovna React Native Paper. Je vyvíjena jako open source a je jednoduchá na použití. Jelikož je zpracována pro React Native, samozřejmě se umí přizpůsobit oběma platformám Android i iOS. Zároveň je možné využívat tmavý a světlý režim aplikace a nadefinovat si vlastní barvy, fonty nebo animace. [33]

Grafy

Jelikož by se v aplikaci mělo objevovat větší množství dat, která by měl uživatel využívat, bylo by dobré je vizualizovat a zobrazovat v grafech, když to bude možné. Grafická reprezentace je účinnější než textové zobrazení, jelikož může zobrazovat více hodnot a uživatel může díky tomu vnímat kontext dat a porovnávat hodnoty mezi sebou. Často je jasně dané, jaký typ grafu bude nejvhodnější, ale někdy to vyžaduje určitý stupeň kreativity. [36]

Grafy můžeme řadit do několika typů, mezi nejpoužívanější patří:

- Sloupcové grafy
- Skládané grafy

- Spojnicové grafy
- Bodové grafy
- Koláčové grafy
- Bullet grafy
- Sparkline
- Box plot
- Treemap

3.4.3. Marketingový výzkum

Marketingový výzkum se zabývá průzkumem potřeb zákazníků. Jelikož v dnešní době se poptávka neustále mění, je potřeba rychle reagovat na změny a získávat nové informace. Ve vývoji mobilní aplikace díky tomuto výzkumu je možné identifikovat cílové uživatele a jejich požadavky. Marketingový výzkum obsahuje několik fází. Tou první je rozpoznání problémů a cílů výzkumu, na základě kterých jsou v další fázi vytvořeny hypotézy. Ty jsou následně díky výzkumu potvrzeny nebo vyvráceny. Každý výzkum je jedinečný, jelikož pokaždé zkoumá jiné problémy. [18]

Dotazník

Mezi metody sběru dat řadíme dotazování, pozorování nebo experiment. Jednou z nejpoužívanějších metod, jak získat kvantitativní data o uživateli, je dotazník, který slouží pro ověření hypotéz. Obsahuje převážně uzavřené otázky, které se snadno vyhodnocují. Někdy ale může obsahovat i otevřené otázky, které už spadají pod kvalitativní výzkum. Výhodou dotazníku je rychlý sběr dat a velký počet respondentů, jelikož se dnes dá posílat i vyplňovat přes internet. [12]

Při vytváření dotazníku je potřeba zohlednit časovou náročnost. Doporučuje se maximálně dvacet minut, jelikož poté je velmi pravděpodobné, že respondent dotazník nedokončí. Také je potřeba zhodnotit, jaké otázky v dotazníku použít. Otevřené otázky nenabízí žádné odpovědi a respondent má prostor na přemýšlení a není nijak omezený. Naopak ale odpovědi na tyto otázky vyžadují složitější zpracování a

interpretaci. Uzavřené otázky jsou rychlé na vyplnění, zobrazené varianty pomáhají respondentovi lépe formulovat názor a jsou jednoduché na další zpracování. Problémem zase můžou být skutečnosti, že otázky mohou vnucovat odpovědi a být monotónní. [18]

Data je potřeba při zpracování dotazníku vhodně zobrazovat. Využívají se k tomu tabulky a různé grafy. Díky tomu je možné z těchto informací pochopit souvislosti.

Persona

Na základě proběhnutého výzkumu je možné identifikovat cílové zákazníky a jejich potřeby. Pro lepší představu o zákazníkovi je možné vytvořit personu. Persona je fiktivní osoba, která představuje typické uživatele aplikace a pomáhá lépe chápat jejich potřeby a vžít se do uživatelské role. Řadíme ji do kvalitativního výzkumu. Persona je obvykle jednostránková vizitka o fiktivním uživateli, kde jsou rozepsány jeho osobní informace, cíle, zkušenosti, zájmy nebo dovednosti. [1]

3.4.4. Monetizace aplikace

Při vývoji mobilní aplikace je také potřeba přemýšlet o její monetizaci, což je proces, během kterého se produkt stává výdělečným. Vývoj aplikace a následné udržování a podpora nejsou levnou záležitostí, proto by si na ně aplikace měla vydělat, pokud není financována jinak. Existuje mnoho způsobů, jak na mobilní aplikaci vydělávat, a to například zpoplatnění stažení nebo vložení reklamních bannerů. Každý způsob je jinak efektivní, záleží na monetizační strategii. Před vytvořením monetizační strategie je potřeba znát cílovou skupinu uživatelů a zjistit, jaký má aplikace pro uživatele přínos, jaká je konkurence a jak naši cílovou skupinu oslovit. [10]

Zpoplatnění stažení aplikace

Nejjednodušším způsobem, jak monetizovat aplikaci, je přímo zpoplatnit její stažení. Bohužel je zaznamenán stálý nárůst bezplatných aplikací, a je proto obtížné se s placenou aplikací prosadit. Tyto aplikace musejí nabízet opravdu něco navíc než konkurence a mít dobře postavenou propagaci. Zisk z těchto aplikací je přímý, ale pouze jednorázový a nemusí být tolik výhodný. Uživatel také nemusí být spokojen s tím, že si aplikaci před zaplacením nemůže ani vyzkoušet. Řešením může být

poskytování dvou verzí aplikace, kdy jedna bude zdarma a nabídne jen základní funkce, a druhá placená bude nabízet další prémiové funkce. [31]

Reklamy v aplikaci

Druhou velice využívanou možností je zobrazení reklam přímo v aplikaci. Reklamy je možné zobrazovat pomocí bannerů, které zabírají menší plochu v horní nebo dolní části aplikace a jsou zobrazeny po celou dobu využívání aplikace. Druhou možností je využití vsunutých reklam, které se na krátkou chvíli objeví přes celou obrazovku. U herních aplikací je možné zobrazit reklamu s nabídkou odměny, kdy uživatel za zhlédnutí reklamy dostane něco, co mu pomůže při hraní hry. Reklamy se také mohou objevovat přímo v aplikaci mezi ostatním obsahem, například u sociálních sítí se zobrazují přímo mezi příspěvky. [31]

Výdělek z reklam se poté odvíjí podle počtu zhlédnutí a případných kliknutí. Pokud má aplikace dost uživatelů, je možné vydělat mnohem více, než při využití zpoplatnění stažení. Pro zobrazení reklam v aplikaci existují tzv. reklamní platformy, které reklamy poskytují. Mezi nejznámější patří Google AdMob, Media.net nebo Unity Ads. Nevýhodou využívání reklam je rušení uživatelského zážitku. Pokud je v aplikaci už moc reklam, uživatele to může odrazovat od jejího používání. [31]

Nákupy v aplikaci

Dalším způsobem, jak získat peníze, je nabízet například v herních aplikacích různé speciální produkty, které si mohou uživatelé dokoupit za reálnou měnu. Kromě herních aplikací takové služby najdeme i u aplikací pro editaci fotografií, kdy je možné si dokupovat různé filtry. [31]

Sponzoring

Aplikaci je možné monetizovat i pomocí sponzoringu, kdy určitý sponzor nebo partner pomáhá s financováním aplikace. Na oplátku je v aplikaci sponzor zmíněn a tím způsobem propagován. [31]

4. Analýza problému a současné situace

Tato kapitola se zabývá analýzou současné situace v oblasti prodeje rukodělných výrobků. Na jakých portálech nebo e-shopech se takové výrobky prodávají a jestli tyto portály nabízí přehledy prodejů nebo statistiky, které by prodejci mohli využívat pro strategické rozhodování.

V další části bude zhodnocen dotazník, který byl vytvořen pro prodejce rukodělných výrobků a zjišťoval, kde výrobky prodávají a jaké služby a aplikace využívají. Z analýzy existujících řešení a dotazníku bude definována cílová skupina uživatelů a budou zpracovány požadavky na novou mobilní aplikaci. Následně bude vytvořen plán vývoje aplikace, kdy bude provedena analýza rizik a časová analýza.

4.1. Současná situace

Rukodělná tvorba a hlavně její prodej se v posledních letech hodně rozrostl. Někteří prodejci se této práci věnují na plno, jiní to mají jen jako koníček při práci. Světový portál Etsy.com, který sdružuje prodejce ručně vyrobených produktů a starobylých věcí, se v roce 2020 dostal až na 4,3 miliony aktivních prodávajících. O rok předtím to bylo jen 2,7 milionů, portál se tedy stále rozrůstá. V České republice je neznámější a největší portál Fler.cz, který registruje až 73 tisíc prodejců. [34] [9]

Někteří prodejci rukodělných výrobků ale nechtějí být součástí Fleru nebo jiného velkého portálu a zakládají si vlastní internetové obchody. Je spousta možností, jak si vlastní e-shop pořídít za poměrně nízké ceny. Nevýhodou vlastních stránek a e-shopů však může být malá šance, že je nový zákazník objeví. Vlastní internetové stránky prodejci proto propagují na sociálních sítích jako Facebook nebo Instagram, přes které mohou jednodušeji získávat nové zákazníky. Prodejci také využívají osobních prodejů na různých jarmarcích, které se konají o svátcích, hodech nebo u jiných příležitostech. [27]

4.1.1. Portál Etsy

Světovým portálem, který vytváří online tržiště pro prodejce unikátních a kreativních výrobků, je Etsy.com. Nabízí pozoruhodné rukodělné produkty a také vintage

produkty, což jsou starodávné věci. Své zboží tu nabízí prodejci z celého světa a je jich více než 4 miliony. Kupujících je naopak až jeden milion. Portál Etsy byl založen v Americe již v roce 2005. [8]

Etsy má také dvě mobilní aplikace. Jednu pro kupující, která umožňuje zboží nakupovat, ukládat do oblíbených, zobrazovat informace o dopravě nebo si psát s prodejci. Druhá aplikace je určena pro prodejce a umožňuje spravovat objednávky a odpovídat na zprávy. Také obsahuje notifikace a prodejce může zboží vytvářet, editovat a publikovat. Aplikace prodejcům také zobrazuje statistiky. Jaká byla návštěvnost jejich obchodu, kolik objednávek a jaký je obrat. Data se snaží zobrazovat i v grafech.

Portál také nabízí pro své uživatele přehledné statistiky. Mezi hlavní sledované ukazatele patří návštěvnost, objednávky a obrat. Tyto charakteristiky jsou zobrazeny přehledně v grafech na časové ose a nechybí ani srovnání s předchozím obdobím. Také je možné sledovat, odkud návštěvníci na web obchodu přicházejí. Ať už přímo ze stránek Etsy a jejich vyhledávání, nebo z cizích stránek, což mohou být přímo odkazy na zboží např. ze sociálních sítí. V neposlední řadě také nabízí možnost propagace zboží za poplatky. [24]

4.1.2. Portál Fler

Prvním a největším portálem, který shromažďuje české a slovenské prodejce ručně-dělných výrobků, je Fler.cz, který vznikl již v roce 2008 a stále je velmi populární. Můžeme zde najít až 16 tisíc prodejců, kteří mají aktuálně zboží v nabídce. Fler denně registruje až 80 000 návštěv a zákazníci denně nakoupí až 4 000 produktů. Každý den je také prodejci vloženo až 5 tisíc nových produktů. [9]

Prodej na Fleru není náročný a vystavování zboží je zdarma. Kromě handmade výrobků se zde nabízí také materiál k jejich výrobě. Prodejci platí Fleru 11 % provizi až z prodaných výrobků. Pro prodejce Fler připravil i mobilní aplikaci, která umožňuje spravovat zboží a objednávky, ale většinu funkcí je stále potřeba dělat přes webovou stránku. Prodejci mohou své zboží také propagovat za určité poplatky, aby se více zobrazovalo ve vyhledávání. Také se zde vytvářejí FlerBoxy, což jsou kolekce zboží většinou k určitému tématu. Vytvářet je může každý registrovaný uživatel a

následně z nich tvořit Top výběry, které se mohou zobrazit na úvodní stránce Fleru, pokud jsou vybrány. [9]

Mobilní aplikace umožňuje základní ovládání obchodu. Hned na začátku jsou v horní liště zobrazena aktuální čísla, jako počet návštěv, obrát v Kč, počet objednávek, srdíček nebo nových fanoušků za poslední týden. Pod těmito informacemi jsou již tzv. události, kde jsou zobrazeny informace o tom, že se někomu líbí produkt, že uživatel vytvořil objednávku nebo že si obchod někdo přidal do oblíbených. Také jsou zde implementovány zprávy a zobrazení informací o profilu. Dále je možné přijímat objednávky a přesouvat je do stavů zaplacené a odeslané, což používá hodně prodejců, jelikož nemusí objednávky potvrzovat přes webovou stránku. Je možné i skrývat či zobrazovat zboží a vytvářet nové.

Problémem může být, že je na Fleru poměrně složité najít statistiky ohledně prodeje. V mobilní aplikaci je možné jen zobrazit čísla v horní liště za tento den, týden, měsíc nebo rok, ale to je všechno. Na webové stránce je možné zobrazit, jaké zboží bylo prodáno v jakých měsících, a je seřazeno podle nejprodávanějšího. Ještě je možné zobrazit nejvíce zobrazované zboží nebo přehled prodeje po dnech. Už ale není možné si prodané zboží rozdělit do kategorií a například zjišťovat, která kategorie je oblíbená. Chybí také zobrazení dat přehledněji v grafech a ne jen v tabulkách.

Fler API

Pro komunikaci mobilní aplikace se serverem Fleru vytvořili vývojáři veřejné API, které můžou využívat i sami prodejci například pro hromadné úpravy zboží, které na webové stránce nejsou možné. Využívání API ale vyžaduje znalosti programování. V další části bude představen přehled funkcí, které API umožňuje využívat. [9]

- **Účet uživatele** poskytuje základní informace o účtu, jako datum vytvoření nebo typ účtu.
- **Zboží prodejce** umožňuje vypsát seznam zboží, editovat jej, vkládat nebo mazat. Dále umožňuje upravovat klíčová slova, dobu dodání nebo zobrazovat a vkládat varianty zboží.
- **Objednávky prodejce** je možné vypsát nebo si zobrazit detail objednávky. Dále je možné přijmout a odmítnout objednávku, nebo ji označit jako zapla-

cenou a odeslanou. Také umožňuje řešit slevové kódy a psát zprávy k objednávkám.

- **Veřejné seznamy** jsou pomocné seznamy kategorií, barev, termínů dodání, zboží a sazeb DPH.
- **Seznamy pro prodejce** obsahují seznam objednávek a jejich stavů, seznam způsobů dopravy prodejce, seznam dopravních služeb, platebních metod a filtrů. Dále seznam zboží a jejich stavů, seznam prodejních kategorií a podkategorií.
- **Statistiky prodejce** obsahují počet návštěv, objednávek, srdíček, fanoušků, jaký je obrat a hodnocení obchodu. Při získávání těchto hodnot se definuje časové období.
- **Faktury prodejce** je možné vypsat i s detaily. Jsou k dispozici faktury, které prodejce vydává zákazníkům, a faktury za provize, které prodejce platí Fleru.
- **Události** jsou např. informace o novém fanouškovi, srdíčku u zboží nebo nové objednávce.
- **Fler pošta** zahrnuje veškerou komunikaci na Fleru. Vytváření zpráv, zobrazení, práci s vlákny nebo štítky.
- **Nástroje** umožňují vyhledávat uživatele.
- **Nahrávání souborů** je možné např. jako příloha ke zprávě v poště nebo fotografie u zboží.

4.2. Dotazník

Pro kvantitativní zjišťování, kde prodejci rukodělných výrobků prodávají své zboží a jaké služby k tomu využívají, byl vytvořen dotazník, který byl rozeslán prodejcům na portálu Fler a dalším prostřednictvím sociálních sítí Facebook a Instagram. Otázky byly především zaměřeny na prodejce Fleru, ale pro případ, když se k dotazníku dostal prodejce z jiných portálů, byly vytvořeny otázky i pro ně. Všichni respondenti byli dotazováni, kde všude své výrobky prodávají a jaký mají vztah ke statistickým

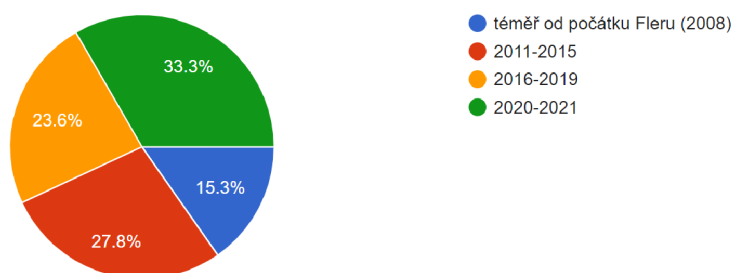
údajům z prodejů. Uživatelé Fleru byli navíc tázáni na jejich obchod na Fleru, jestli využívají Fler aplikaci a jestli na Fleru našli a využívají statistiky.

Dotazník vyplnilo 83 respondentů, z nichž 96 % byly ženy. Věk respondentů byl v rozmezí převážně 31 až 40 let (41 %). Na druhém místě bylo rozmezí 21 až 30 let (33 %). Hned zpočátku byla položena otázka, zda uživatel prodává své výrobky na Fleru, na kterou odpovědělo 87 % kladně.

Prodejci na Fleru byli následně tázáni na dobu, kdy začali na Fleru prodávat. Na obrázku 4.1 je možné vidět, že až třetina začala prodávat teprve nedávno. Nicméně ostatní kategorie byly také zastoupeny téměř stejně. Dále byla položena otázka, zda je prodej na Fleru hlavní činností respondenta a pokud ne, měl vybrat, co je jeho hlavní činností. Jen 23 % uvedlo, že je Fler jejich práce. Zbýlých 77 % prodává výrobky v rámci vedlejší činnosti nebo jen koníčku. Ti následně uvedli, že jsou zaměstnaní (45 %), na mateřské (24 %), nebo studenti (17 %).

Kdy jste začal/a na Fleru prodávat?

72 responses



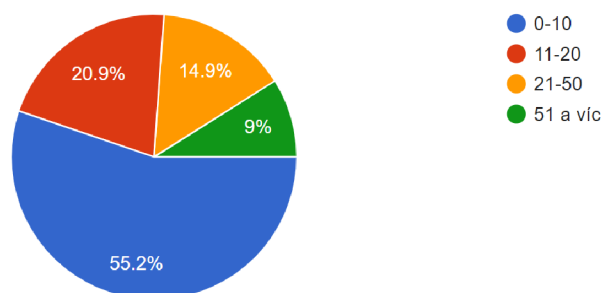
Obrázek 4.1: Graf, kdy začali prodejci na Fleru prodávat (vlastní zpracování)

Další otázky se zaměřovaly na aplikaci od Fleru a zda ji prodejci využívají. Neustále ji využívá asi 46 %, občas asi 22 %, ale 24 % ji ani vůbec nemá. Využívají tedy webovou stránku. Nicméně ti, co aplikaci využívají, s ní mají určité problémy. Nejvíce jim vadí omezené funkce při správě zboží, objednávek nebo zasílání zpráv. Využívají proto i mobilní rozhraní webové stránky, aby se ke všem funkcím dostali.

Následně bylo zkoumáno, kolik kusů zboží prodejci za měsíc zhruba prodají. Na obrázku 4.2 je vidět, že většina prodá jen 0-10 kusů.

Kolik kusů zboží na Fleru prodáte zhruba za měsíc?

67 responses

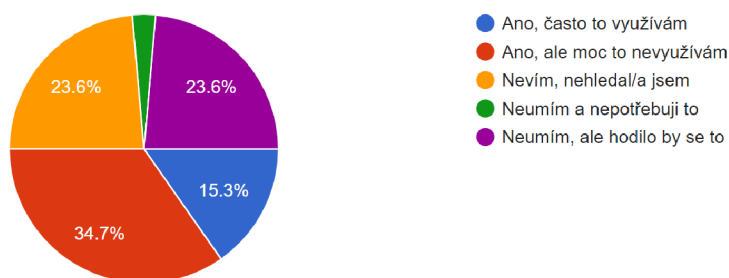


Obrázek 4.2: Graf, kolik kusů zboží je prodáno (vlastní zpracování)

Další část se věnovala statistikám. Prodejci na Fleru byli tázáni, zda si umí na webových stránkách najít, jaké zboží prodali v jednotlivých měsících. Graf odpovědí je na obrázku 4.3. Zajímavé je, že téměř 24 % statistiky nenašli, ale hodily by se jim. Další otázky byly pro všechny respondenty. Na otázku, zda je statistiky zajímaví, odpovědělo 66 % respondentů kladně. Téměř 80 % sleduje, jaké výrobky se nejvíce prodávají, a 71 % se snaží nabízet takové zboží, které se nejvíce prodává.

Umíte si najít nějaké statistiky na Fleru? Např. Jaké výrobky jste prodali v jednotlivých měsících?

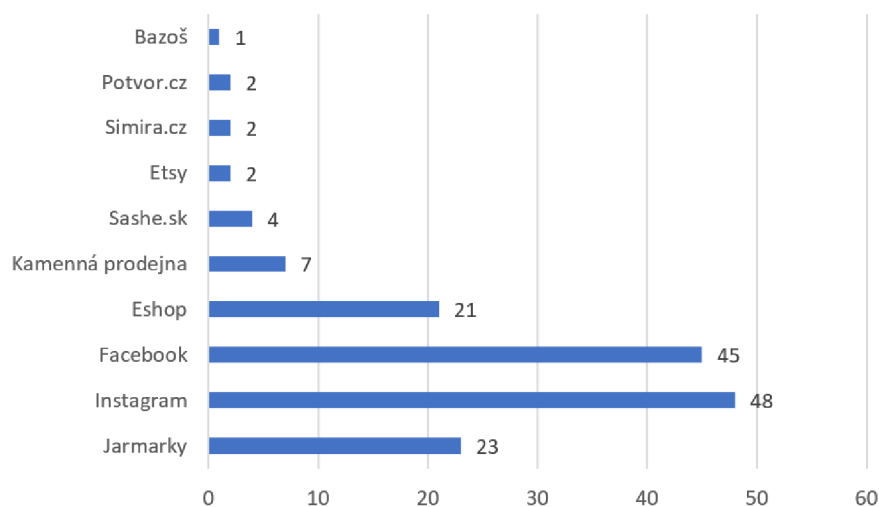
72 responses



Obrázek 4.3: Graf, zda prodejci využívají statistiky (vlastní zpracování)

Všichni respondenti byli také tázáni, kde všude své výrobky prodávají. Jestli jen na Fleru nebo i na sociálních sítích, vlastním eshopu nebo jinde. Tato otázka nabízela odpovědi, ale umožňovala i zapsání vlastní odpovědi, proto byly výsledky

následně zpracovány pomocí grafu v programu Excel (obrázek 4.4). Nejvíce prodávají na sociálních sítích, jarmarcích nebo vlastním eshopu.

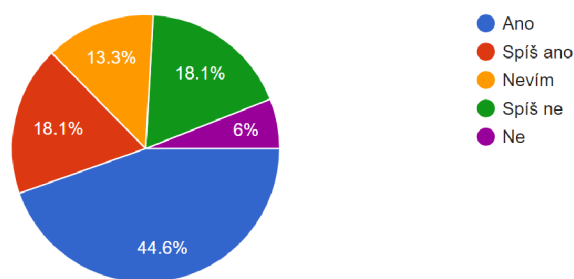


Obrázek 4.4: Místa, kde také prodejci prodávají výrobky (vlastní zpracování)

Na závěr bylo respondentům sděleno, že se tato diplomová práce bude zabývat návrhem aplikace, která bude zobrazovat statistiky z prodejů. Otázkou bylo, zda by o takovou aplikaci měli zájem. Odpovědi jsou na obrázku 4.5. Až 62 % by o aplikaci mělo zájem.

Mobilní aplikace, kterou navrhuji, by měla zobrazovat převážně statistické údaje a grafy ohledně prodeje zboží (jaké zboží se nejvíce prodává, jaké je už dlouho na skladě, jaké je potřeba doplnit atd.). Myslíte, že byste takovou aplikaci využívali?

83 responses



Obrázek 4.5: Graf, zda by měli prodejci zájem o aplikaci (vlastní zpracování)

ohledně jeho prodejů, aby mohl zhodnotit svou úspěšnost a podle dosavadních statistik rozhodovat, kam bude se svým obchodem mířit dál. Může například zjistit, jaká kategorie jeho zboží je nejúspěšnější a nejoblíbenější, a díky tomu se na ni více zaměřit. Aplikace by mu také měla poskytnout přehled o tom, jak se jeho prodeje a návštěvnost vyvíjely v čase. Nemělo by chybět ani upozornění na zboží, kterého je již v nabídce málo a bylo by dobré jej doplnit.

Základní charakteristiky, které by měla aplikace zobrazovat, jsou následující:

- **Prodej zboží** - jak se vyvíjel, jaké zboží se nejvíce prodává, v jakých kategoriích zboží je nebo jaký byl obrat.
- **Nabídka zboží** - počet nabízeného zboží dle kategorií.
- **Návštěvnost** - jak se vyvíjela a jestli se zlepšuje.
- **Oblíbenost** - počet fanoušků obchodu, počet srdíček u zboží a jak se vyvíjel.
- **Objednávky** - jaké bylo poslední objednané zboží, přehled a počet objednávek.
- **Hodnocení** - jak byly objednávky hodnoceny, jaký je průměr.

4.4. Plán vývoje

V rámci plánu vývoje aplikace byla vytvořena analýza rizik a časová analýza.

4.4.1. Analýza rizik

Pro analýzu rizik a jejich opatření byla použita skórovací metoda. Bylo definováno devět rizik, které je nutné sledovat. V tabulce 4.7 je definována stupnice pravděpodobností a dopadu rizik a následně v tabulce 4.8 jsou uvedena rizika, jejich pravděpodobnosti (P), dopady (D) a výsledné hodnoty (H).

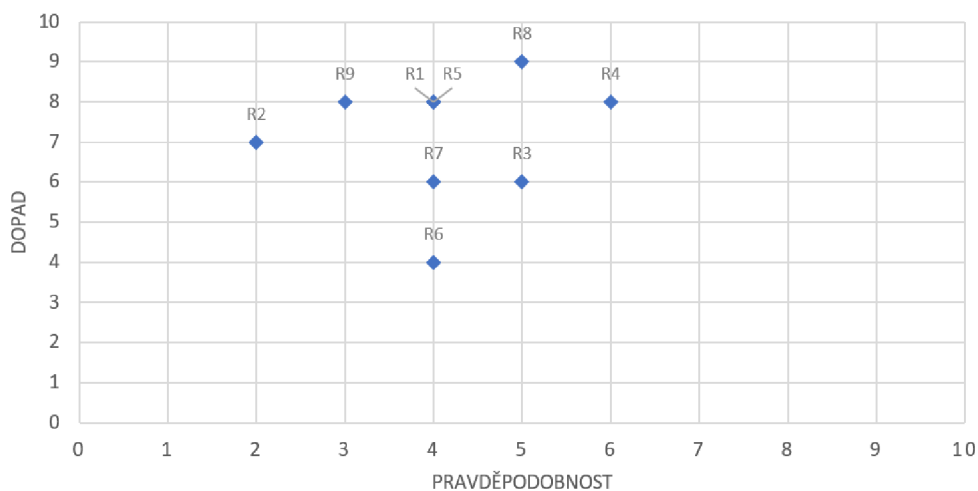
Kritickým rizikem je riziko 4, nedostatek pracovníků pro vývoj, a riziko 8, změny požadavků na funkčnost aplikace. Za závažná rizika můžeme označit riziko 1, kdy je možné, že se API od Fleru změní, nebo riziko 5, že produkt nebude využívaný uživateli. Také riziko 9, kdy by odešli nepostradatelní vývojáři. Mapa rizik je na obrázku 4.9.

Hodnota pravděpodobnosti	Pravděpodobnost vzniku rizika	Pravděpodobnost v %	Hodnota dopadu	Dopad na projekt
1-2	Žádná	0-20	1-2	Téměř žádný
3-4	Nízká	21-40	3-4	Nízký
5-6	Střední	41-60	5-6	Střední
7-8	Spíše vysoká	61-80	7-8	Závažný
9-10	Vysoká	81-100	9-10	Kritický

Obrázek 4.7: Stupnice pravděpodobností a dopadu rizika (vlastní zpracování)

ID	Hrozba	Scénář	P	D	H
R1	Změna Fler API	Nefunkční aplikace	4	8	32
R2	Nedostatek financí pro vývoj produktu	Nemožnost hradit výdaje	2	7	14
R3	Nedostatek času pro vývoj produktu	Pozastavení vývoje	5	6	30
R4	Nedostatek pracovníků pro vývoj produktu	Pozastavení vývoje	6	8	48
R5	Nevyužívání produktu	Zbytečně vynaložené náklady	4	8	32
R6	Podcenění časového odhadu vývoje	Prodloužení vývoje	4	4	16
R7	Neúplná analýza	Prodloužení vývoje	4	6	24
R8	Změny požadavků na funkčnost	Prodloužení vývoje nebo zrušení	5	9	45
R9	Odchod nepostradatelných zaměstnanců	Zrušení vývoje	3	8	24

Obrázek 4.8: Tabulka rizik s hodnocením (vlastní zpracování)

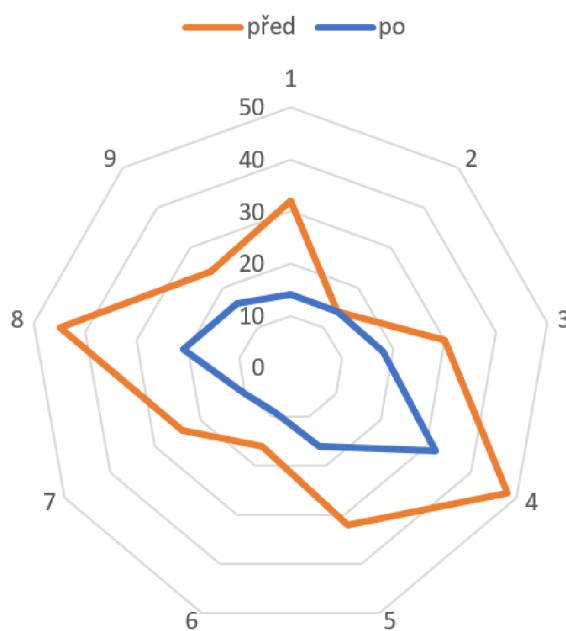


Obrázek 4.9: Mapa rizik před opatřením (vlastní zpracování)

V tabulce 4.10 jsou zobrazeny rizika již s jejich navrženými opatřeními a opět vypočítány výsledné hodnoty. Následuje pavučinový graf (obrázek 4.11), který reflektuje změny hodnot rizik po zavedených opatřeních.

ID	Hrozba	Opatření	P	D	H
R1	Změna Fler API	Komunikace s Flerem ohledně změn API	2	7	14
R2	Nedostatek financí pro vývoj produktu	Dobrá finanční a časová analýza	2	7	14
R3	Nedostatek času pro vývoj produktu	Analýza naplánované práce v období realizace a zohlednění časové rezervy	3	6	18
R4	Nedostatek pracovníků pro vývoj produktu	Analýza naplánované práce, komunikace se zaměstnanci, řešení personálních změn včas	4	8	32
R5	Nevyužívání produktu	Seznámení zaměstnanců s produktem a předložení výhod práce s ním	2	8	16
R6	Podcenění časového odhadu vývoje	Analýza naplánované práce, komunikace s vývojáři o časovém odhadu	3	3	9
R7	Neúplná analýza	Komunikace s budoucími uživateli systému, analýza jejich požadavků	2	5	10
R8	Změny požadavků na funkčnost	Komunikace s budoucími uživateli systému, analýza současného stavu	3	7	21
R9	Odchod nepostradatelných zaměstnanců	Komunikace se zaměstnanci, řešení jejich požadavků	2	8	16

Obrázek 4.10: Tabulka rizik a opatření (vlastní zpracování)



Obrázek 4.11: Pavučinový graf rizik před a po zavedení opatření (vlastní zpracování)

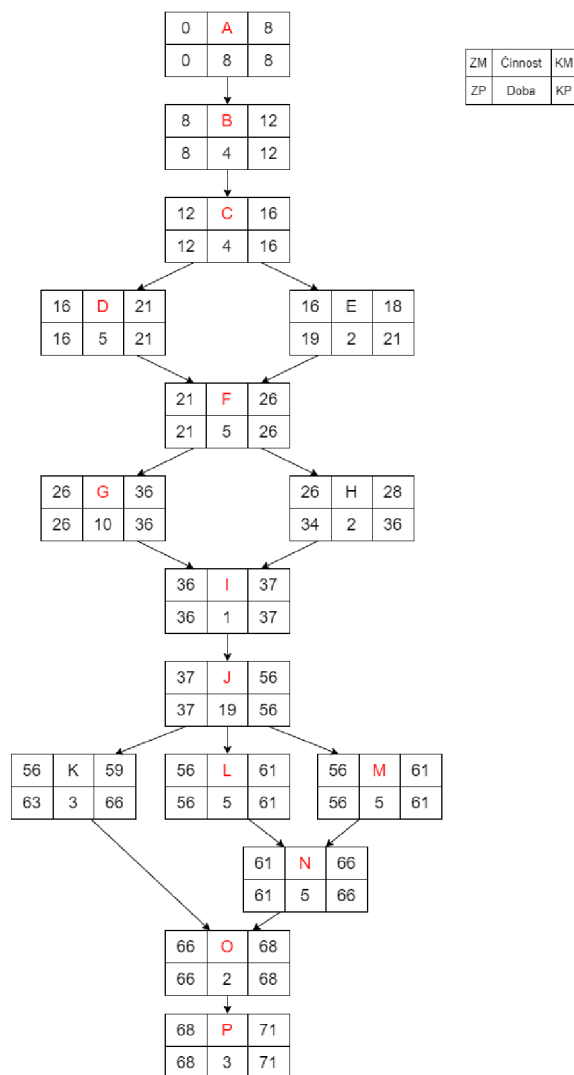
4.4.2. Časová analýza

Časová analýza byla vytvořena pomocí metody PERT a definuje 16 činností, které je potřeba vykonat. Nejprve bude proveden průzkum aktuálního stavu a budou zpracovány požadavky na novou aplikaci. Následně budou definovány funkce, které by aplikace měla splňovat, a vytvořen Wireframe a Style Guide. Následuje mockup a vytvoření prototypu. Poté by měly být definovány potřebné technologie na implementaci a vytvořeny milníky. Po implementaci bude potřeba testovat aplikaci na platformách Android i iOS a připravovat marketing a monetizaci. Následně budou probíhat případné opravy a poté již bude moct jít aplikace do produkce, po níž se může řešit marketing a monetizace.

Tabulka s trváním činností je na obrázku 4.12 a síťový graf je na obrázku 4.13. Kritická cesta zahrnuje činnosti: A-B-C-D-F-G-I-J-L-M-N-O-P a celkem by měl vývoj trvat 71 pracovních dnů. Časové odhady jednotlivých činností byly konzultovány s analytikem ve firmě, která se vývojem software a mobilních aplikací zabývá.

Údaje o posloupnosti činností projektu				Trvání (dny)				Statistické uk.		Termíny zahájení a ukončení					
Číslo	Označení	Popis činnosti	i	j	aij	mij	bij	tij	oyij	σ2yij	ZM	KM	ZP	KP	RC
1	A	Průzkum aktuálního stavu	-	B	6	8	10	8	0.67	0.44	0	8	0	8	0
2	B	Analýza požadavků na novou aplikaci	A	C	2	4	6	4	0.67	0.44	8	12	8	12	0
3	C	Definice funkcí aplikace	B	D,E	3	4	5	4	0.33	0.11	12	16	12	16	0
4	D	Wireframe	C	F	3	5	7	5	0.67	0.44	16	21	16	21	0
5	E	Style Guide	C	F	1	2	3	2	0.33	0.11	16	18	19	21	3
6	F	Mockup	D,E	G,H	3	5	7	5	0.67	0.44	21	26	21	26	0
7	G	Vytvoření prototypu	F	I	8	10	12	10	0.67	0.44	26	36	26	36	0
8	H	Výběr technologií	F	I	1	2	3	2	0.33	0.11	26	28	34	36	8
9	I	Naplánování milníků a postupu práce	G,H	J	1	1	2	1	0.17	0.03	36	37	36	37	0
10	J	Implementace aplikace	I	K,L,M	14	18	28	19	2.33	5.44	37	56	37	56	0
11	K	Příprava marketingu a monetizace	J	O	1	3	5	3	0.67	0.44	56	59	63	66	7
12	L	Testování Android	J	N	3	5	7	5	0.67	0.44	56	61	56	61	0
13	M	Testování iOS	J	N	3	5	7	5	0.67	0.44	56	61	56	61	0
14	N	Opravy	L,M	O	1	5	9	5	1.33	1.78	61	66	61	66	0
15	O	Nasazení	K,N	P	1	2	4	2	0.50	0.25	66	68	66	68	0
16	P	Marketing a monetizace	O	-	2	3	4	3	0.33	0.11	68	71	68	71	0

Obrázek 4.12: Tabulka činností PERT (vlastní zpracování)



Obrázek 4.13: Síťový graf PERT (vlastní zpracování)

4.5. Zhodnocení

V této části byla provedena analýza současné situace, kdy prodejci převážně prodávají na portálu Fler a využívají jejich webové stránky a aplikaci. Nikde ale nejsou zobrazeny podrobnější informace o návštěvnosti obchodu, oblíbenosti zboží a hlavně prodeji zboží. Je možné dohledat, jaké zboží se kdy prodalo, ale není to nikde zobrazeno přehledně, například v grafech. Fler byl srovnáván s portálem Etsy, který obsahuje více statistik.

Byla definována cílová skupina uživatelů, kteří by novou mobilní aplikaci, jež by přehledně statistiky zobrazovala, využili. A to na základně dotazníku, který vyplnilo

83 respondentů. Díky tomu byly specifikovány požadavky na mobilní aplikaci a následně provedena analýza rizik a časová analýza.

Při vývoji aplikace se může objevit mnoho problémů, ale díky analýze rizik se jim snažíme předejít. Nicméně stále je možné, že aplikace po vytvoření a nasazení nebude mít dostatečnou reklamu nebo se uživatelům nebude líbit. To by mělo odhalit i následné testování aplikace ještě před nasazením do produkce.

5. Vlastní návrh řešení

Mobilní aplikace bude sloužit výhradně prodejcům na Fleru a zobrazovat statistiky z prodeje na tomto portálu. Prodejce by měl aplikaci využívat jako doplněk k již stávající aplikaci od Fleru a webovým stránkám. Tato aplikace bude poskytovat určité přehledy a prodejce by měl díky ní mít větší přehled o svém obchodu. Jak se mu daří aktuálně a jestli se jeho obchod vyvíjí tím správným směrem. Aplikace jej bude i upozorňovat, pokud určité zboží nebo celá kategorie zboží již neobsahuje moc položek v nabídce. Prodejci by proto měla pomáhat řešit, jaké zboží je potřeba doplnit a co se zákazníkům líbí nejvíc.

V této kapitole bude navržen drátěný model aplikace, který bude následně převeden do funkčního prototypu, který jasně představí, jak bude aplikace vypadat. Budou také popsány použité technologie pro implementaci a ukázány části kódu. Poslední část se věnuje nápadům na rozšíření aplikace.

5.1. Wireframe

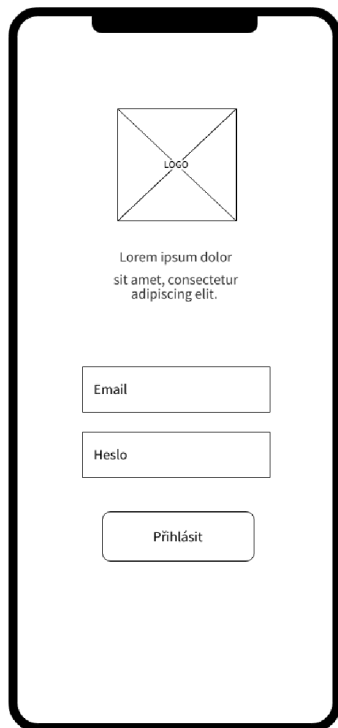
Pro definici základních charakteristik, které by aplikace měla zobrazovat, byl navržen drátěný model (wireframe), který obsahuje několik obrazovek. Wireframe byl zpracováván ve webové online aplikaci na stránce [Mockflow.com](https://www.mockflow.com), která umožňuje vytvoření několika stránek zdarma.

5.1.1. Přihlašovací stránka

První stránkou, na kterou se uživatel dostane hned po otevření aplikace, je přihlašování na Fler. Obrazovka obsahuje logo aplikace, krátký text, dvě textová pole pro email a heslo a tlačítko pro přihlášení (viz obrázek 5.1). Uživatelské rozhraní aplikace po přihlášení je rozvrženo do tří základních obrazovek: domovská stránka, stránka s aktuálními přehledy a stránka s historií. Pro navigaci mezi nimi je využita spodní lišta se třemi ikonami jako tlačítka.

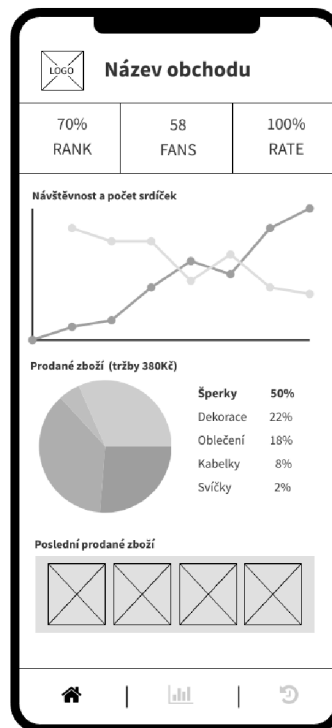
5.1.2. Domovská stránka

Na domovské stránce (viz obrázek 5.2) je uživateli poskytnut základní přehled o aktuálním dění v jeho obchodu. Dozví se, jaký má rank, což je speciální hodnocení



Obrázek 5.1:

Přihlašovací stránka (vlastní zpracování)



Obrázek 5.2:

Domovská stránka (vlastní zpracování)

obchodů na Fleru a pohybuje se v rozmezí 0 - 100, kdy 100 je nejlepší možný. Přesný algoritmus výpočtu není veřejný, ale odvíjí se od množství prodaného zboží, ceny, oblíbenosti prodejce, počet zobrazení zboží, hodnocení, množství zboží ve FlerBo-xech nebo Top výběrech [9]. Také je zde zobrazen počet aktuálních fanoušků, což mohou být jiní prodejci nebo registrovaní kupující, kteří si obchod přidají do oblíbených. Třetí položkou je procentuální hodnocení prodejce uživateli, kteří už zde nakoupili.

V další části domovské obrazovky se uživatel dozví, jaká je návštěvnost obchodu za poslední týden a jaký je počet srdíček u zboží. Uživatelé Fleru mohou každé zboží označovat jako oblíbené. Díky tomuto grafu může uživatel zjistit, jestli je jeho stránka s obchodem navštěvována a pokud ne, může ji zkusit jinak propagovat.

Poslední část domovské obrazovky je zaměřena na prodané zboží za poslední týden. Je zde zobrazen koláčový graf prodaného zboží podle kategorií a vedle přehledně sepsány jednotlivé kategorie, jejich procentuální hodnoty a případně i absolutní hodnoty. Dále jsou ještě zobrazeny fotografie posledního prodaného zboží, které by měly vést na podstránku s informacemi o daném zboží.

5.1.3. Stránka s aktuálními přehledy

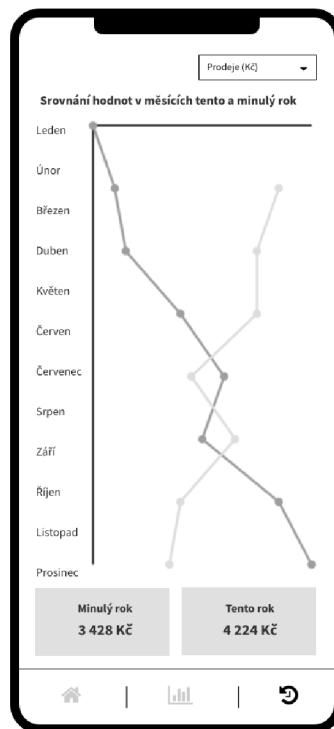
Na druhé obrazovce jsou grafy z aktuálního dění a jedná se o rozšíření grafu z prodejů z domovské stránky (viz obrázek 5.3). Této obrazovce dominuje graf, který rozděluje zboží podle kategorií a ukazuje, kolik bylo prodáno a kolik zboží je v dané kategorii v nabídce. Mělo by se tedy jednat o graf s dvěma hodnotami, aby si prodejce mohl porovnat, zda nabízí dostatek zboží v každé kategorii vůči již prodanému zboží.

V druhé části jsou zvýrazněny přímo položky zboží. Jedná se o nejprodávanější zboží, nejoblíbenější a nejvíce zobrazované zboží. Všechny tři položky by měly vést na podstránky, které zobrazí celý seznam zboží seřazený podle dané vlastnosti.

Celá tato obrazovka ještě obsahuje v horní části záložky, díky kterým je možné si všechny informace zobrazit buď za poslední týden, měsíc nebo rok.



Obrázek 5.3:
Aktuální přehledy (vlastní zpracování)



Obrázek 5.4:
Historie (vlastní zpracování)

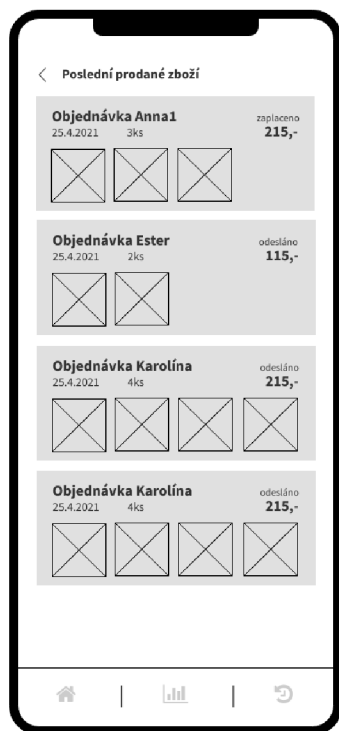
5.1.4. Stránka s historií

Třetí obrazovka se zaměřuje na data z historického pohledu (viz obrázek 5.4). Je zde zobrazen graf, který ukazuje např. prodané zboží v korunách podle měsíců za

tento a minulý rok. Díky tomu je možné data srovnávat s minulým rokem a zjišťovat, jestli se prodeje zlepšily. Pod grafem je navíc srovnání celkového počtu za tento a minulý rok. Aby měl uživatel více možností než jen zobrazovat obrat, je v horní části možnost výběru. Měl by mít možnost si zobrazit např. počet prodaného zboží, návštěvnost obchodu či počet srdíček.

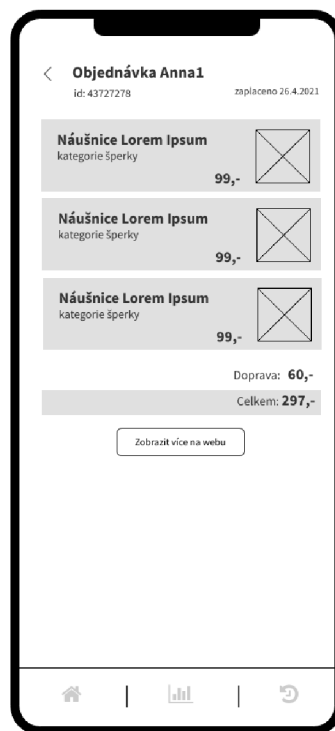
5.1.5. Další podstránky

Z domovské stránky bude možné přejít na seznam objednávek se zbožím a to ze sekce s posledním prodaným zbožím. Podstránka by měla vypadat jako seznam objednávek, které dále bude možné zobrazit detailně, jak je vidět na obrázcích 5.5 a 5.6. Z detailu objednávky bude možné přejít na webovou stránku Fleru, kde má uživatel všechny další informace.



Obrázek 5.5:

Poslední objednávky (vlastní zpracování)

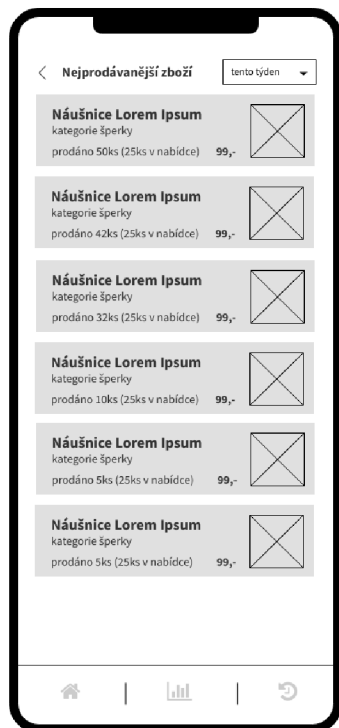


Obrázek 5.6:

Detail objednávky (vlastní zpracování)

Ze stránky s aktuálními přehledy bude možné zobrazit detailně seznamy se zbožím seřazené podle počtu koupených kusů, srdíček u zboží a podle navštěvovanosti zboží (viz obrázek 5.7). Bude zde opět možnost zobrazit statistiky za poslední týden, měsíc nebo rok. V seznamu budou o zboží zobrazeny základní informace, jako

název, kategorie, cena, počet prodaných kusů a kusů v nabídce a fotografie. Dané zboží by mělo dále odkazovat na podstránku s detailními informacemi, jak je vidět na obrázku 5.8. Zde jsou zobrazeny všechny fotografie u zboží, popis, cena a následně informace o kategorii, prodaných kusech, počtu zobrazení a srdíček. Opět bude možné zobrazit zboží na webu Fleru.



Obrázek 5.7:
Seznam zboží (vlastní zpracování)



Obrázek 5.8:
Detail zboží (vlastní zpracování)

5.2. Design aplikace

S vytvořenými drátěnými modely již bylo snadnější vymyslet, jak se aplikace bude jmenovat a v jakých barvách by měla být vyobrazena. Název aplikace byl navržen jako „Creative Stats“, což je anglický název, ale do češtiny snadno přeložitelný. Název by měl v uživateli vzbuzovat skutečnost, že se bude jednat o statistiky. Slovo kreativní je použito z toho důvodu, že je aplikace určena pro prodejce rukodělných výrobků, u jejichž výroby je kreativita zapotřebí.

Barvy aplikace byly navrženy pomocí nástroje Paletton¹. Hlavní barvou aplikace byla navržena světle hnědá okrová barva. Jelikož bude v aplikaci větší množství

¹Dostupné na <https://paletton.com/>

grafů, které by měly být stěžejní, bylo rozhodnuto, že tyto grafy budou barevné, aby uživatel mohl lépe porovnávat informace. Z toho důvodu byla zvolena právě okrová neutrální barva, která k ostatním barevným prvkům bude ladit.

Styly grafických prvků, jako jsou tlačítka, textová pole, záložky, ikony nebo výběrová tlačítka, budou implementovány pomocí knihovny React Native Paper, která je zobrazuje podle stylu Material Design.

5.3. Implementace

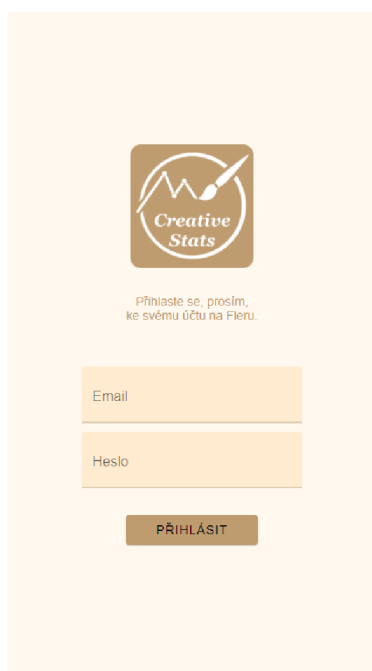
V rámci implementace byl vytvořen funkční prototyp aplikace s testovacími daty. Také bylo testováno reálné připojení na Fler API a získávání dat.

5.3.1. Prototyp aplikace

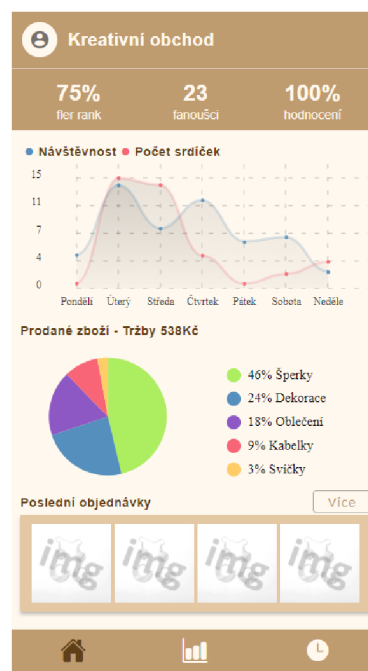
Prototyp aplikace, který již zobrazuje výsledný design a zároveň umožňuje uživateli přecházet mezi jednotlivými stránkami, byl vytvářen přímo v React Native a testován převážně na webovém prohlížeči. Prototyp zatím zobrazuje jen testovací data, ale je díky němu znatelné, jak by měla aplikace vypadat a jaké funkce bude obsahovat.

Přihlašovací stránka (obrázek 5.9) zobrazuje logo aplikace, má stejně rozmístěné prvky jako na drátěném modelu a dovoluje uživateli vyplnit údaje. Po kliknutí na tlačítko Přihlásit je uživatel přesměrován na domovskou stránku (obrázek 5.10). Zde jsou zobrazeny základní údaje o obchodu, spojnicový graf s návštěvností a srdíčkami za poslední týden, prodeje za poslední týden v rámci koláčového grafu podle kategorie zboží a obrázky posledního prodaného zboží. Zde přibyl tlačítko s nápisem „Více“, které uživatele přesměruje na podstránku se seznamem posledních objednávek (obrázek 5.11). Každá objednávka v tomto seznamu dále funguje jako odkaz na detail objednávky (obrázek 5.12).

Stránka s aktuálními přehledy umožňuje v záhlaví aplikace přepínat mezi týdnem, měsícem nebo rokem. Ukazuje prodané zboží za daný čas versus aktuálně nabízené zboží. Byl vybrán sloupcový graf, který je otočen o 90 stupňů, jelikož se předpokládá, že kategorií může být více. V dolní části pak dominují tři produkty, které vévodí žebříčkům co se týče prodeje, oblíbenosti a návštěvnosti. Každý box je



Obrázek 5.9:
Přihlašovací stránka (vlastní zpracování)



Obrázek 5.10:
Domovská stránka (vlastní zpracování)

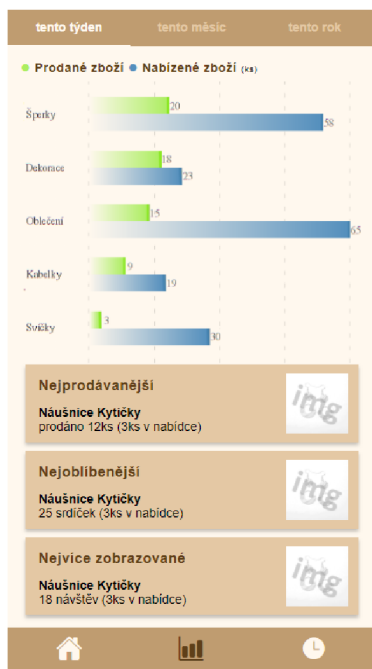


Obrázek 5.11:
Poslední objednávky (vlastní zpracování)

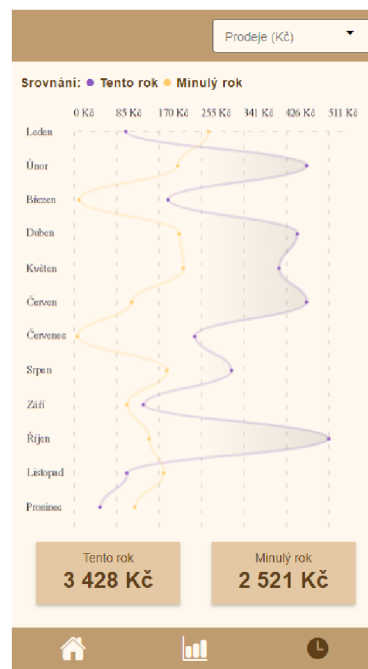


Obrázek 5.12:
Detail objednávky (vlastní zpracování)

zároveň odkazem na podstránku se seznamem zboží (obrázek 5.15), které je seřazeno podle příslušného typu. Každý produkt dále odkazuje na detail (obrázek 5.16), kde je zobrazeno více informací o produktu.



Obrázek 5.13:
Aktuální přehledy (vlastní zpracování)



Obrázek 5.14:
Historie (vlastní zpracování)



Obrázek 5.15:
Seznam zboží (vlastní zpracování)



Obrázek 5.16:
Detail zboží (vlastní zpracování)

Stránka s historií (obrázek 5.14) obsahuje spojnicový graf s hodnotami za poslední rok srovnanými s minulým rokem. Zde má prodejce možnost porovnat hodnoty z dlouhodobého hlediska. V dolní části jsou ještě srovnány celkové hodnoty za oba

roky. V záhlaví aplikace je možné měnit zobrazované hodnoty na prodeje v korunách, prodeje v kusech, návštěvnost obchodu a počet srdíček.

5.3.2. Použité technologie

Při programování prototypu aplikace byl využíván framework React Native a další knihovny pro správu stylů, designu aplikace a pro zobrazení grafických prvků, jako jsou grafy. Také byly využívány HTTP požadavky (metody GET a POST), které umožnily komunikaci s Fler API.

React Native

Framework React Native [32] byl již představen v teoretické části. Pomocí npm serveru byl framework nainstalován společně s knihovnou Expo CLI, která se stará o celé vývojové prostředí a simulaci aplikace na webovém prohlížeči. K editování kódu byl využit program PhpStorm od firmy JetBrains [17], který zvýrazňuje proměnné a také upozorňuje na chyby ještě před kompilací.

React Native Paper

Knihovna React Native Paper [33] již také byla představena v teoretické části práce. Definuje určité grafické prvky, které je následně snadné použít a není potřeba řešit, jak budou vypadat. Knihovna umožňuje definovat vlastní barvy, čehož bylo využito.

Mezi grafické prvky, které byly použity, patří tlačítko, textové pole, záložky, výběrové pole nebo ikony.

React Native Chart Kit

Grafy byly vykresleny pomocí React Native Chart Kit, což je open source knihovna, která umožňuje jednoduché vytvoření grafů. Mezi ně patří spojnicové grafy, sloupcové grafy, skládané grafy, koláčové grafy a contribution grafy (grafy zobrazující mřížku kalendáře a každý den je zbarvený podle počtu). Byl využit spojnicový, sloupcový a koláčový graf.

5.3.3. Ukázky kódu

V této části práce bude představeno několik ukázek kódu v React Native. Jak vypadá nastavení komponent pro zobrazení textových polí a tlačítek, jak probíhá komunikace se serverem Fleru a jak se nastavuje zobrazení grafu.

Přihlášení na Fler

Na začátku aplikace je uživatel vyzván, aby se přihlásil ke svému účtu na Fleru. Musí vyplnit email a heslo a kliknout na tlačítko Přihlásit. Poté je volána funkce `handleSubmit()`, která posílá přes metodu GET vyplněné informace na Fler. Na obrázku 5.17 je zobrazen kód v React Native, který zobrazuje uživateli textová pole pro zadání emailu a hesla a tlačítko pro přihlášení. Pokud se přihlášení nezdaří, je také nad textovými poli zobrazen červený text o chybě.

```
<View>
  <Text style={styles.red}>{errorText}</Text>
  <TextInput
    label="Email"
    value={username}
    onChangeText={setUsername}
  />
  <TextInput
    label="Heslo"
    value={password}
    onChangeText={setPassword}
    secureTextEntry
  />
  <Button mode="contained" onPress={() => handleSubmit()}>
    Přihlásit
  </Button>
</View>
```

Obrázek 5.17: Zobrazení formuláře pro přihlášení (vlastní zpracování)

Funkce `handleSubmit()` (obrázek 5.18) volá asynchronně funkci `loginUser()` a posílá jí údaje zadané uživatelem. Po získání odpovědi následně podmínkou zjišťuje, zda jsou vrácená data v pořádku, nebo se objevila chyba. V případě chyby ukládá do proměnné `errorText` chybovou hlášku, která se následně zobrazí uživateli. Pokud

jsou data v pořádku, uloží je do proměnné `token` a data jsou následně zpracovávána jinými funkcemi.

Funkce `loginUser()` (obrázek 5.19) řeší posílání požadavku metodou POST na Fler API a čeká na odpověď. Data vrací zpět funkci, která ji volala.

```
const handleSubmit = async () => {
  const token = await loginUser( credentials: {
    username: username,
    pwd: password,
  });
  if(token.error == undefined){
    // uloz si hodnoty
    setToken(token);
  }else{
    // zobraz error
    setErrorText( value: "Nesprávné hodnoty");
  }
};
```

Obrázek 5.18: Funkce `handleSubmit()` (vlastní zpracování)

```
async function loginUser(credentials: any){
  return fetch( input: 'https://www.fler.cz/api/rest/user/auth', init: {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json'
    },
    body: JSON.stringify(credentials)
  }).then(data => data.json())
}
```

Obrázek 5.19: Funkce `loginUser()` (vlastní zpracování)

Získání dat

Získání dalších dat z Fleru je podmíněno autorizací. Speciální autorizační hash se musí sestavit z klíčů `secretKey` a `sessionId`, které jsou získány v datech při přihlášení. Fler API umožňuje volat mnoho metod pro získání různých dat. V ukázce na obrázku 5.20 je zobrazena funkce `getProducts()`, která požaduje po Fleru prvních deset produktů, které prodejce nabízí, a jejich název, popis, počet kusů v nabídce a kategorii. Je vytvořen požadavek metodou GET a v hlavičce je posílán hash. Ten je

vytvářen funkcí `calcAuth`, která jej skládá z názvu metody, url, aktuálního času a dvou klíčů. Využívá k tomu hashovací funkce z knihovny `CryptoJS`.

```
async function getProducts(secretKey: any, sessionId: any){
  return fetch( input: `https://www.fler.cz/api/rest/seller/products/list?limit=10&
  fields=title,description,price,stock,stock_unit,category`, init: {
    method: 'GET',
    headers: {
      'X-FLER-AUTHORIZATION': calcAuth(
        http_method: 'GET',
        http_path: '/api/rest/seller/products/list',
        secretKey,
        sessionId),
      'Content-Type': 'application/json'
    }
  }).then(data => data.json())
}

function calcAuth(http_method: any, http_path: any, secretKey: any, sessionId: any){
  let time = Math.floor( X: new Date().getTime() / 1000);
  let request_string = http_method + "\n" + time + "\n" + http_path;
  return "API1_SESS " + sessionId + ' ' + time + ' '
    + window.btoa(CryptoJS.HmacSHA1(request_string, secretKey));
}
```

Obrázek 5.20: Funkce pro získání seznamu produktů (vlastní zpracování)

Koláčový graf

Grafy byly vykresleny pomocí knihovny `React Native Chart Kit`. Na obrázku 5.21 je zobrazen kód pro vykreslení koláčového grafu, který zobrazuje prodeje zboží podle kategorií s testovacími daty. Výsledný graf byl zobrazen na obrázku 5.10, který ukazuje jednu stránku v aplikaci.

Do grafu je potřeba data nahrát a nastavit jim barvu pozadí, barvu písma a velikost písma. Následně se nastavuje výška a šířka grafu, která se získává podle šířky displeje funkcí `Dimensions`, jelikož každý mobil má jinou velikost. Dále se nastavuje tzv. `accessor`, do kterého se zadává položka z dat, která obsahuje číselnou hodnotu. V našem případě je to proměnná `money`, která určuje, kolik Kč bylo u dané kategorie vyděláno. Výsledný graf hodnoty přepočítá na procenta. Nakonec se nastavují další grafické parametry, jako barva pozadí celého grafu nebo počet desetinných míst při zobrazení.

```

<PieChart
  data={[
    {name: "Šperky", money: 490, color: "#acee60", legendFontColor, legendFontSize},
    {name: "Dekorace", money: 250, color: "#548fbd", legendFontColor, legendFontSize},
    {name: "Oblečení", money: 189, color: "#8d58c3", legendFontColor, legendFontSize},
    {name: "Kabelky", money: 100, color: "#f76576", legendFontColor, legendFontSize},
    {name: "Svíčky", money: 30, color: "#ffcd68", legendFontColor, legendFontSize}
  ]}
  width={Dimensions.get( dim: "window").width * 0.95}
  height={Dimensions.get( dim: "window").width * 0.4}
  accessor={"money"}
  backgroundColor={"transparent"}
  chartConfig={{
    backgroundColor: theme.colors.background,
    decimalPlaces: 0,
  }}
  paddingLeft={"0"}
/>

```

Obrázek 5.21: Nastavení koláčového grafu (vlastní zpracování)

5.4. Možné rozšíření aplikace

Cílem této mobilní aplikace bylo vhodné zobrazování statistik z prodeje rukodělných výrobků na Fleru. Když ale prodejci prodávají i na jiných místech, bylo by dobré zvažovat i tuto situaci a zobrazovat statistiky i odtud. Dle analýzy prodejci nejvíce využívají platformy Instagram a Facebook. Bylo by dobré do aplikace přidat možnost zobrazování statistik i z těchto platforem.

Jelikož se většinou jedná o prodeje uskutečněné přes zprávy, není možné je automatizovat. Prodejce by proto musel mít možnost zapisovat si takové údaje do aplikace sám. Jaké zboží se prodalo, kdy a na které platformě. V aplikaci by se musel vytvořit formulář, který by prodejce vyplnil. Následně by se tato data musela ukládat buď na jiný server nebo přímo do mobilu uživatele.

Hlavní obrazovky v aplikaci by se mohly následně rozšířit o jednu další, která by se zaměřovala na srovnání všech prodejních platforem.

6. Zhodnocení

Byl vytvořen prototyp mobilní aplikace, která bude zobrazovat přehledy z prodeje rukodělných výrobků na Fleru.

Uživatel

Uživatel bude mít díky aplikaci lepší přehled o svém obchodu a bude snadněji vytvářet obchodní strategii. Z přehledů se dozví, jaké zboží je oblíbené a často prodávané, bude moci například upravit jeho cenu, více se věnovat jeho tvorbě nebo se snažit vytvářet podobně úspěšné produkty.

Aplikace bude fungovat na obou nejpoužívanějších platformách Android a iOS, proto by měla mít většina majitelů chytrých telefonů možnost aplikaci využívat. Aplikace je navržena pro zobrazení přehledů a statistik, které jinde prodejci nenajdou. Není ale stěžejní pro prodej, proto by zpoplatnění stažení mohlo většinu uživatelů odradit. Monetizace aplikace se proto bude řešit až v rámci drobných reklam v aplikaci. Jestli budou aplikaci prodejci využívat, závisí na zvolené marketingové strategii a zda se jim aplikace bude líbit. Z analýzy však vyplývá, že by se aplikace prodejcům hodila.

Vývoj

Byla provedena analýza současné situace a definována cílová skupina uživatelů. Také byly charakterizovány požadavky na aplikaci a na jejich základě navržen drátěný model. Společně s definovanými styly aplikace byl model převeden do funkčního prototypu v React Native a testován na webovém prohlížeči.

Dokončení aplikace by podle časové analýzy v kapitole 4 mělo ještě zabrat nejméně 43 dní. Byly již splněny činnosti A až H, tudíž bude potřeba naplánovat milníky práce, vytvořit aplikaci, testovat, opravovat a nasadit ji do produkce. Testování aplikace bude důležitá fáze. Bylo by dobré aplikaci testovat přímo s vybranými prodejci na Fleru a pozorovat je, jak aplikaci využívají.

Výhodou aplikace je skutečnost, že se programuje v jednom frameworku pro obě platformy. Vývoj je rychlejší a také jednodušší. Nicméně nasazení aplikace do produkce se bude muset řešit pro každou platformu zvlášť.

7. Závěr

Tato diplomová práce se zaměřovala na návrh a vytvoření prototypu mobilní aplikace. V teoretické části se práce věnovala operačním systémům Android a iOS, což jsou nejpoužívanější platformy, frameworku React Native a postupům vývoje mobilní aplikace. V další části byla zpracována analýza současné situace, kde prodejci nejvíce prodávají své rukodělné výrobky a zda jim přehledy z prodeje chybí. Z průzkumu byla definována cílová skupina uživatelů a požadavky na aplikaci. Následně byla zpracována analýza rizik vývoje aplikace a časový plán.

Poté byl popsán vlastní návrh řešení. Byl představen drátěný model aplikace, který byl společně s navržením stylů a designu převeden do funkčního prototypu. Prototyp aplikace byl vyvíjen ve frameworku React Native, který umožňuje multiplatformní vývoj. Díky tomu se nemusí vytvářet dvě samostatné aplikace pro každý operační systém zvlášť. Základ aplikace byl doveden do stavu, kdy jsou zobrazena testovací data v již dokončené grafické podobě.

Aplikace je zaměřena na zobrazení přehledů z prodeje rukodělných výrobků na Fleru. Bude sloužit pro prodejce, kterým zobrazí data v podobě grafů a vhodných přehledů. Díky nim bude prodejce moci vyhodnocovat situaci svého obchodu a následně vytvářet nebo upravovat svou obchodní strategii.

Literatura

- [1] Adobe. *Putting Personas to Work in UX Design: What They Are and Why They're Important* [online]. Poslední změna 2019-12-17 [cit. 2021-04-26]. Dostupné z: <https://xd.adobe.com/ideas/process/user-research/putting-personas-to-work-in-ux-design/>
- [2] Android Authority. *The history of Android* [online]. Poslední změna 2020-11-12 [cit. 2021-03-09]. Dostupné z: <https://www.androidauthority.com/history-android-os-name-789433>
- [3] Android Developers. [online]. [cit. 2021-03-14]. Dostupné z: <https://developer.android.com/>
- [4] AndroidPub. *Objective-C or Swift: Which Technology to Learn for iOS App Development?* [online]. Poslední změna 2018-03-25 [cit. 2021-03-16]. Dostupné z: <https://medium.com/android-news/objective-c-or-swift-which-technology-to-learn-for-ios-app-development-3c681d1a05ac>
- [5] CORDOVA, M.: *Creating a UI Style Guide for Web and Mobile Apps in 2020* [online]. Poslední změna 2020-07-15 [cit. 2021-04-26]. Dostupné z: <https://www.krasamo.com/creating-ui-style-guide-for-web-and-mobile-apps/>
- [6] CSS-Tricks. *A Complete Guide to Flexbox* [online]. [cit. 2021-03-16]. Dostupné z: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- [7] DANNHOFEROVÁ, J.: *Velká kniha barev: Kompletní průvodce pro grafiky, fotografie a designéry*. Computer Press, 2015. 352 p. ISBN 978-80-251-3785-7.
- [8] Etsy. [online]. [cit. 2021-04-26]. Dostupné z: <https://www.etsy.com/>
- [9] Fler. [online]. [cit. 2021-04-26]. Dostupné z: <https://www.fler.cz/>
- [10] GENADINIK, A.: *MobileApp Marketing and Monetization*. 5th ed. CreateSpace Independent PublishingPlatform, 2014. 149 p. ISBN 978-1502383822.

- [11] Global Stats. *Mobile Operating System Market Share Worldwide* [online].
Poslední změna 2021-02-28 [cit. 2021-03-09]. Dostupné z:
<https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [12] HENDL, J.: *Kvalitativní výzkum: základní metody a aplikace*. Portál, 2005,
ISBN 80-7367-040-2.
- [13] IntelliPaat. *What is the architecture of iOS* [online]. [cit. 2021-03-18].
Dostupné z: <https://intellipaate.com/blog/tutorial/ios-tutorial/ios-architecture/>
- [14] Invonto. *Mobile App Development Process: A Step-by-Step Guide* [online]. [cit.
2021-04-04]. Dostupné z:
<https://www.invonto.com/insights/mobile-app-development-process/>
- [15] ITnetwork. *Úvod do TypeScriptu* [online]. [cit. 2021-04-26]. Dostupné z: <https://www.itnetwork.cz/javascript/typescript/uvod-do-typescriptu>
- [16] JACOBS, J. and B. RUDIS: *Data, Driven Security: Analysis, Visualization and Dashboards*. 1st ed. Indianapolis: John Wiley & Sons, 2014. 352 p. ISBN
978-1-118-79372-5.
- [17] JetBrains. [online]. [cit. 2021-03-14]. Dostupné z:
<https://www.jetbrains.com/>
- [18] KOZEL, R., L. MYNÁŘOVÁ a H. SVOBODOVÁ: *Moderní metody a techniky marketingového výzkumu*. 1. vyd. Praha: Grada, 2011. 304 s. ISBN
978-80-247-3527-6.
- [19] LACKO, L. a M. HERODEK: *Vývoj aplikací pro iOS*. 1. vyd. Brno:
Computer Press, 2018. 480 s. ISBN 978-80-251-4942-3.
- [20] Learn. *Reading Ios Xcode Overview* [online]. [cit. 2021-03-16]. Dostupné z:
<https://learn.co/lessons/reading-ios-xcode-overview>
- [21] LearnAppMaking. *Understanding Model-View-Controller (MVC) on iOS*
[online]. [cit. 2021-03-18]. Dostupné z:
<https://learnappmaking.com/model-view-controller-mvc-swift/>

- [22] LEE, W. a M.: *Beginning Android Application Development*. 1st ed. Indianapolis: Wiley Pub., 2011. 448 p. ISBN 978-1-118-08780-0.
- [23] LogRocket. *Why use Redux? A tutorial with examples* [online]. Poslední změna 2020-10-12 [cit. 2021-04-26]. Dostupné z: <https://blog.logrocket.com/why-use-redux-reasons-with-clear-examples-d21bffd5835/>
- [24] Marketing Artfully. *How To Analyze The New Etsy Stats* [online]. Poslední změna 2019-10-10 [cit. 2021-04-27]. Dostupné z: <https://marketingartfully.com/how-to-analyze-the-new-etsy-stats/>
- [25] Material Design. [online]. [cit. 2021-04-26]. Dostupné z: <https://material.io/>
- [26] MDN Web Docs. [online]. [cit. 2021-04-27]. Dostupné z: <https://developer.mozilla.org/>
- [27] Na volné noze. *Handmade byznys* [online]. Poslední změna 2015-05-28 [cit. 2021-04-26]. Dostupné z: <https://navolnenoze.cz/blog/handmade/>
- [28] Net Solutions. *Where Do Cross-Platform App Frameworks Stand in 2021?* [online]. Poslední změna 2020-12-08 [cit. 2021-03-16]. Dostupné z: <https://www.netsolutions.com/insights/cross-platform-app-frameworks-in-2019/>
- [29] PILGRIM, M.: *Ponořme se do HTML5*. 1. vyd. Praha: CZ.NIC, 2014. 278 s. ISBN 978-80-905802-6-8.
- [30] Rascasone. *Javascript pro začátečníky: Co to je a jak funguje* [online]. Poslední změna 2021-04-13 [cit. 2021-04-26]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-javascript-pro-zacatecniky>
- [31] Rascasone. *Tipy na strategie monetizace a propagace mobilních aplikací* [online]. Poslední změna 2021-04-30 [cit. 2021-05-03]. Dostupné z: <https://www.rascasone.com/cs/blog/monetizace-propagace-mobilnich-aplikaci>

- [32] React Native. [online]. [cit. 2021-04-04]. Dostupné z:
<https://reactnative.dev/>
- [33] React Native Paper. [online]. [cit. 2021-04-26]. Dostupné z:
<https://reactnativepaper.com/>
- [34] Statista. *Number of active Etsy sellers from 2012 to 2020* [online]. [cit. 2021-04-28]. Dostupné z:
<https://www.statista.com/statistics/409374/etsy-active-sellers/>
- [35] The Verge. *There are now 2.5 billion active Android devices* [online]. Poslední změna 2019-05-07 [cit. 2021-03-09]. Dostupné z:
<https://www.theverge.com/2019/5/7/18528297/google-io-2019-android-devices-play-store-total-number-statistic-keynote>
- [36] TUFTE, E. R.: *The visual display of quantitative information*. Graphics press Cheshire, CT, 2001, ISBN 978-1-930824-13-3.

8. Seznam použitých zkratek a symbolů

API	Application Programming Interface
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
MVC	Model View Controller
UI	User Interface
UX	User Experience
XML	Extensible Markup Language

9. Přílohy



Obrázek 9.1: Druhá vytvořená persona (vlastní zpracování)