

Vývoj mobilní aplikace pro adopci koček

Marcela Havelková

Bakalářská práce
2022



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Marcela Havelková**
Osobní číslo: **A18042**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Softwarové inženýrství**
Forma studia: **Prezenční**
Téma práce: **Vývoj mobilní aplikace pro adopci koček**
Téma práce anglicky: **Development of Mobile Application for Cat Adoptions**

Zásady pro vypracování

1. Vypracujte literární rešerši na téma vývoje mobilních aplikací, včetně technologií, které hodláte při vývoji použít.
2. Navrhněte mobilní aplikaci pro adopci koček na základě požadavků vybrané kočičí kavárny.
3. Naprogramujte mobilní aplikaci pro adopci koček dle návrhu.
4. Ověřte bezpečnost řešení a prověřte možnosti jeho publikování.
5. Vyhodnotte provedené řešení.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. VERSLUIS, Gerald a Steven THEWISSEN. *Xamarin.Forms Solutions*. New York: APress, 2018, 281 s. ISBN 9781484241349.
2. HERMES, Dan. *Xamarin mobile application development: cross-platform C# and Xamarin.Forms fundamentals*. [New York]: Apress, [2015], 432 s. ISBN 9781484202159.
3. MCWHERTER, Jeff, Scott GOWELL, David SMITH, Lauren COLTON, Ameila MARSCHALL-MILLER a Adam RYDER. *Professional mobile application development*. Indianapolis: John Wiley & Sons, [2012], xxviii, 403 s. ISBN 9781118203903.
4. Dokumentace pro Xamarin. *Docs.microsoft.com* [online]. Microsoft, 2020 [cit. 2020-11-27]. Dostupné z: <https://docs.microsoft.com/cs-cz/xamarin/>
5. Create and set up your app. *Play Console Help* [online]. Google, 2020 [cit. 2020-11-27]. Dostupné z: <https://support.google.com/googleplay/android-developer/answer/9859152?hl=en#>

Vedoucí bakalářské práce: **Ing. Tomáš Vogeltanz, Ph.D.**
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **3. prosince 2021**

Termín odevzdání bakalářské práce: **23. května 2022**



doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 24. ledna 2022

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 19. 5. 2022

Marcela Havelková, v.r.
podpis studenta

ABSTRAKT

Cílem bakalářské práce je navrhnout a implementovat mobilní aplikaci, která umožní uživatelům fyzicky či virtuálně adoptovat kočku z konkrétní kočičí adopční kavárny. Pomůže tak mnohým kočkám získat nového majitele, který jim poskytne domov. Aplikace je koncipována pro operační systém Android s možností rozšíření o iOS verzi do budoucna. Je vytvářena v jazyce C# a xaml s využitím frameworku Xamarin. Z důvodu bezpečného přístupu k databázi, bylo vytvořeno také API. V práci je popsán i samotný vývoj aplikace.

Klíčová slova: mobilní aplikace, multiplatformní aplikace, mobilní platformy, platforma Xamarin, Android, Google Play

ABSTRACT

The aim of the bachelor thesis is to design and implement a mobile application that allows users to physically or virtually adopt a cat from a specific cat adoption café. It will help many cats to find a new owner who will provide them a home. The application is designed for the Android operating system with the possibility of extension to iOS devices in the future. It is developed in C# and xaml using the Xamarin framework. For the sake of secure access to the database, an API service was also created. For secure access to the database, an API as a service was also created. The development of the application itself is also described in the thesis.

Keywords: mobile apps, multiplatform apps, mobile platforms, Xamarin platform, Android, Google Play

Chtěla bych poděkovat svému vedoucímu práce Ing. Tomáši Vogeltanzovi, Ph.D. za pomoc, trpělivost a pevné nervy během vedení mé bakalářské práce.

Velké díky také patří adopční kočičí kavárně v Kroměříži, bez které by tato aplikace nevnikla.

Dále bych chtěla poděkovat celé své rodině a kamarádům za velkou psychickou podporu a trpělivost po celou dobu studia.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 MOBILNÍ APLIKACE	12
1.1 WEBOVÁ APLIKACE.....	12
1.2 NATIVNÍ APLIKACE.....	13
1.3 HYBRIDNÍ APLIKACE	13
2 ANDROID	15
2.1 ARCHITEKTURA.....	15
2.1.1 Linuxové jádro	16
2.1.2 Nativní knihovna	17
2.1.3 Běhové prostředí systému Android.....	18
2.1.4 Aplikační rámce	18
2.1.5 Aplikace	19
2.2 HISTORIE.....	20
2.3 VERZE SYSTÉMU ANDROID.....	20
2.3.1 Android 10 a 11.....	20
2.3.2 Android 12	21
2.4 LOGO ANDROID A JEHO HISTORIE.....	21
3 IOS	24
3.1 ARCHITEKTURA IOS.....	24
3.1.1 Cocoa Touch	25
3.1.2 Media Layer	26
3.1.3 Core Services	26
3.1.4 Core OS.....	27
3.2 NÁSTROJE IOS	27
3.2.1 Xcode	27
3.2.2 AppCode	27
3.2.3 iOS Simulator.....	28
3.2.4 Test Flight	28
3.2.5 Transporter	28
3.3 HISTORIE.....	28
4 OBCHODY S MOBILNÍMI APLIKACEMI	30
4.1 GOOGLE PLAY.....	30
4.1.1 Podmínky publikování	30
4.1.2 Postup publikování.....	30
4.2 APP STORE	31
4.2.1 Podmínky publikování	31
4.2.2 Postup publikování.....	31

5	VÝVOJ MOBILNÍCH APLIKACÍ NA PLATFORMĚ .NET	33
5.1	XAMARIN.....	33
5.1.1	Xamarin.Native	33
5.1.2	Xamarin.Forms.....	34
5.2	.NET MAUI.....	34
II	PRAKTICKÁ ČÁST	36
6	NÁVRH APLIKACE	37
6.1	SPECIFIKACE POŽADAVKŮ	37
6.2	REGISTRACE A PŘIHLAŠOVÁNÍ.....	38
6.3	SPRÁVA UŽIVATELSKÉHO ÚČTU	38
6.4	ADOPCE.....	38
6.5	PŘÍPADY UŽITÍ.....	39
6.6	WIREFRAMES	41
6.6.1	Úvodní obrazovka	41
6.6.2	Stránka kavárny.....	44
6.6.3	Stránka uživatele	48
6.7	NÁVRH DATABÁZE	50
7	VÝVOJ APLIKACE	53
7.1	REST API.....	53
7.1.1	Použité frameworky	54
7.1.2	Endpointy	54
7.1.3	Controllery	54
7.2	MOBILNÍ APLIKACE V XAMARIN.FORMS	55
7.2.1	Registrace	56
7.2.2	Přihlášení.....	59
7.2.3	Nepřihlášený uživatel.....	60
7.2.4	Přihlášený uživatel	63
7.2.5	Stránka uživatele	67
7.2.6	Připojení na API.....	70
7.3	BUDOUCÍ VÝVOJ APLIKACE	70
8	OVĚŘENÍ BEZPEČNOSTI ŘEŠENÍ.....	72
8.1	REGISTRACE.....	72
8.2	PŘIHLÁŠENÍ.....	73
8.3	TOKENY	73
8.4	API	74
8.5	PŘÍSTUP K DATŮM	74
8.6	SQL INJECTION	74
	ZÁVĚR	75

SEZNAM POUŽITÉ LITERATURY.....	77
SEZNAM OBRÁZKŮ	88
SEZNAM PŘÍLOH.....	90

ÚVOD

Tématem bakalářské práce je vývoj multiplatformní mobilní aplikace pro adopci koček.

Během posledních pár let se potýkáme s přemnoženými koťaty. Nejedná se pouze o koťata, ale i o kočky, které jsou nechtěné, nalezené či zubožené. V lepších případech končí v útulcích, v horších případech už mrtvé. Existují kočičí kavárny a organizace, které se věnují tomuto problému a snaží se ho co nejvíce regulovat.

Tato práce popisuje tvorbu aplikace, která slouží k uskutečnění adopce a případné pomoci kočičím kavárnám, které se o adopci zajímají. Je vyvíjena společně ve spolupráci s adopční kočičí kavárnou v Kroměříži.

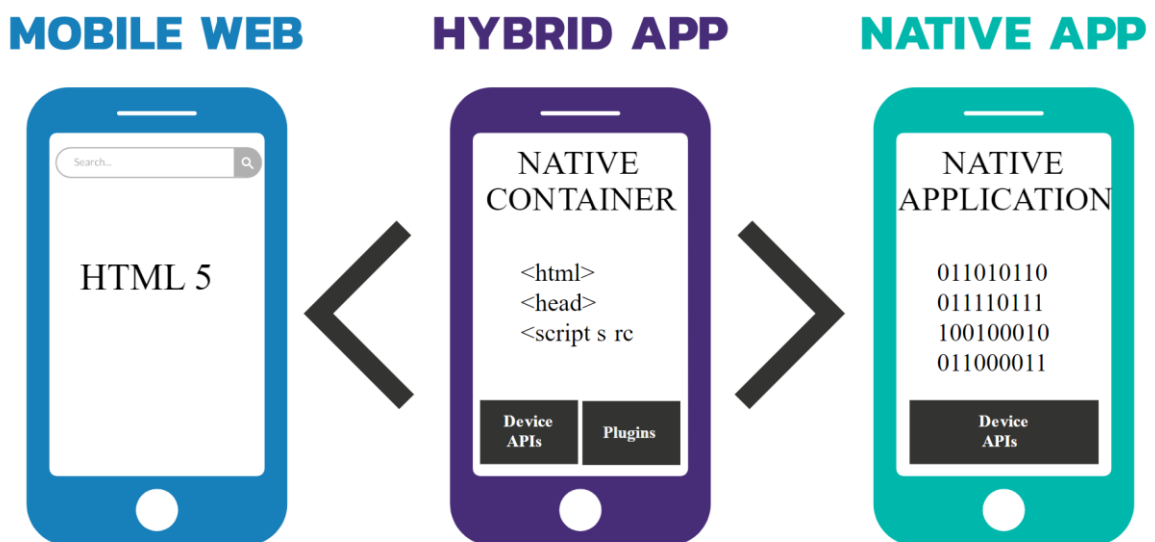
Aplikace je určena pro lidi, kteří vlastní mobilní telefon s operačním systémem Android a mají zájem podílet se na dobré věci a pomoci s řešením této situace. Vzhledem k využití multiplatformního frameworku lze v budoucnu publikovat i na platformě iOS.

První polovina práce je teoretická, věnuje se rešerši současného mobilního světa. Popisuje tři základní druhy mobilních aplikací, mobilní operační systém Android a iOS, dále popisuje obchody s mobilními aplikacemi a platformu Xamarin, která slouží pro vytváření aplikací především pro Android a iOS. Druhá polovina se naopak zabývá praktickou částí. Jedná se o návrh aplikace, jeho řešení a následnou implementaci včetně všech použitých postupů a ukázek zdrojových kódů nebo ukázek již navržené aplikace. Dále se věnuje bezpečnosti řešení v rámci vytvořené aplikace.

I. TEORETICKÁ ČÁST

1 MOBILNÍ APLIKACE

Mobilní aplikace je softwarová aplikace, která je určena ke spuštění na mobilním zařízení. Jedná se o chytré telefony, tablety apod. Poskytuje uživatelům podobné služby, jaké používají počítače. V současné době je vytvořeno mnoho mobilních aplikací, kam patří i oblíbené mobilní hry. Obrázek 1 znázorňuje 3 základní druhy aplikací a rozdíl mezi nimi.[1][2][3]



Obrázek 1 - Rozdíly mobilních aplikací[4]

1.1 Webová aplikace

Webová aplikace je aplikace, kterou není nutné instalovat do zařízení uživatele a je zprostředkována internetovým prohlížečem. Lze ji spustit z kteréhokoliv zařízení, které je připojené k internetu, pomocí webového prohlížeče. Je nutné ji vyvíjet s ohledem na specifika mobilních zařízení.[5][6]

Aplikace je spuštěna na straně serveru. Osobní data uživatele jsou uložena a zálohována na serveru a jsou přístupné z kteréhokoliv zařízení. V dnešní době jsou webové aplikace velmi oblíbené, a to svou přístupností a multiplatformností. Nejsou vyvíjeny pro určitý operační systém, to znamená, že lze k webové aplikaci přistupovat odkudkoliv. Nevýhodou webových aplikací je, že se vyžaduje připojení k internetu.[5][7][8]

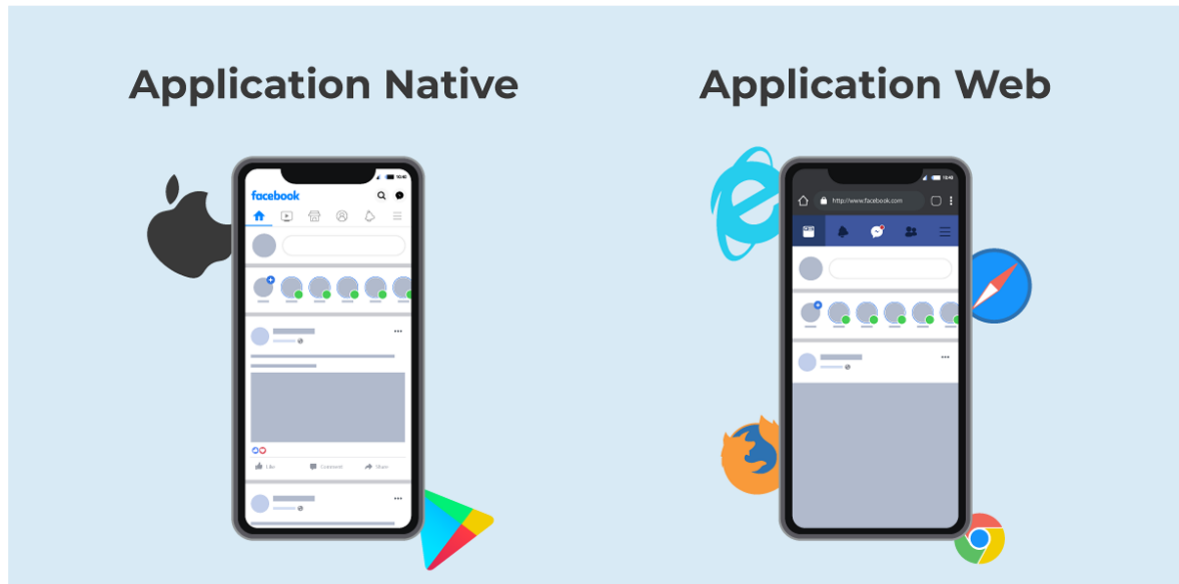
Webová aplikace může v mnoha případech vypadat jako klasická webová stránka. Rozdíl mezi webovou aplikací a webovou stránkou je to, že webová aplikace je složitější a obsahuje různé funkce a oproti webové stránce je dynamická. U webové stránky se většinou jedná o statické stránky, které slouží k čtení textu nebo prohlížení videí.[5][8][9]

1.2 Nativní aplikace

Nativní aplikace je aplikace, která je vyvinuta pro jednu specifickou platformu a plně využívá všechny funkce zařízení, např. fotoaparát a GPS. Dvě hlavní platformy mobilních operačních systémů jsou iOS společnosti Apple a Android společnosti Google. Pro každou platformu se aplikace musí vyvíjet zvlášť ačkoliv mají stejnou funkcionalitu. To znamená, pokud je aplikace vytvořena pro Android, lze ji spustit a stáhnout pouze pro zařízení s operačním systémem Android.[10][11]

Je nutné ji instalovat do zařízení. Instalují se prostřednictvím obchodu s aplikacemi – Google Play pro Android a App Store pro Apple.[11]

Je vytvořena v konkrétním programovacím jazyce. Nativní aplikace pro Android jsou napsané převážně v Javě, ale v posledních letech se dost rozšířil Kotlin. Výhodou Kotlinu je jeho 100% interoperabilita s Javou. Lze použít libovolný kód či knihovna z Javy. Jedná se o efektivní spolupráci a schopnost systémů si vzájemně rozumět. U iOS jsou nativní aplikace napsané v Objective-C a ve Swiftu. Stejně jako Kotlin, se Swift za posledních pár let značně rozvinul. Obrázek 2 ukazuje rozdíl mezi nativní a webovou aplikací.[12][13]



Obrázek 2 - Rozdíl mezi nativní a webovou aplikací[14]

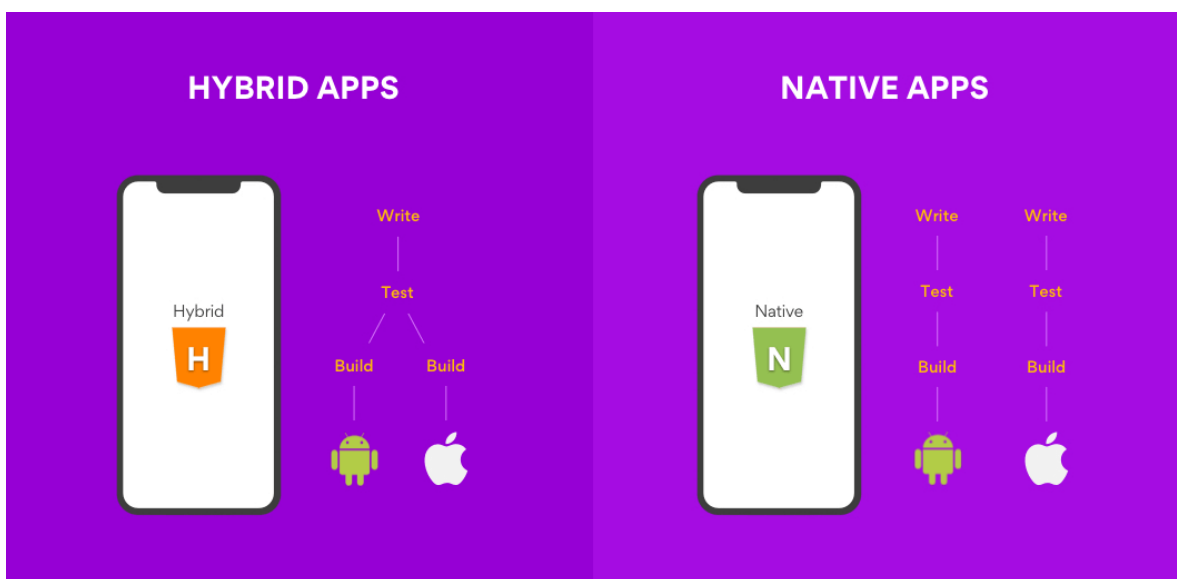
1.3 Hybridní aplikace

Hybridní aplikace je aplikace, která kombinuje prvky nativních a webových aplikací. Jádro aplikace je napsáno pomocí webových technologií (HTML, CSS a JavaScript), které jsou následně nasazeny v nativním kontejneru. Ten umožňuje aplikaci využívat určité funkce

nativní platformy a hardware zařízení, např. fotoaparát, GPS, kalendář. U klasické webové aplikace nemá k těmto funkcím přístup.[15]

Hybridní aplikace běží ve WebView. Je to vlastní zabudovaný prohlížeč v aplikaci. Právě ten při zapnutí aplikace zobrazuje webový obsah díky použitým webovým technologiím.[15]

Aplikaci je nutné instalovat do zařízení, a to prostřednictvím obchodů s aplikacemi. Vzniká z jednoho zdrojového kódu, tzn. je napsaná v jednom jazyce jak pro iOS, tak pro Android. Díky tomu ji lze spustit i na jiné platformě. Obrázek 3 znázorňuje rozdíl mezi hybridní a nativní aplikací.[16]



Obrázek 3 - Rozdíl mezi hybridní a nativní aplikací[17]

2 ANDROID

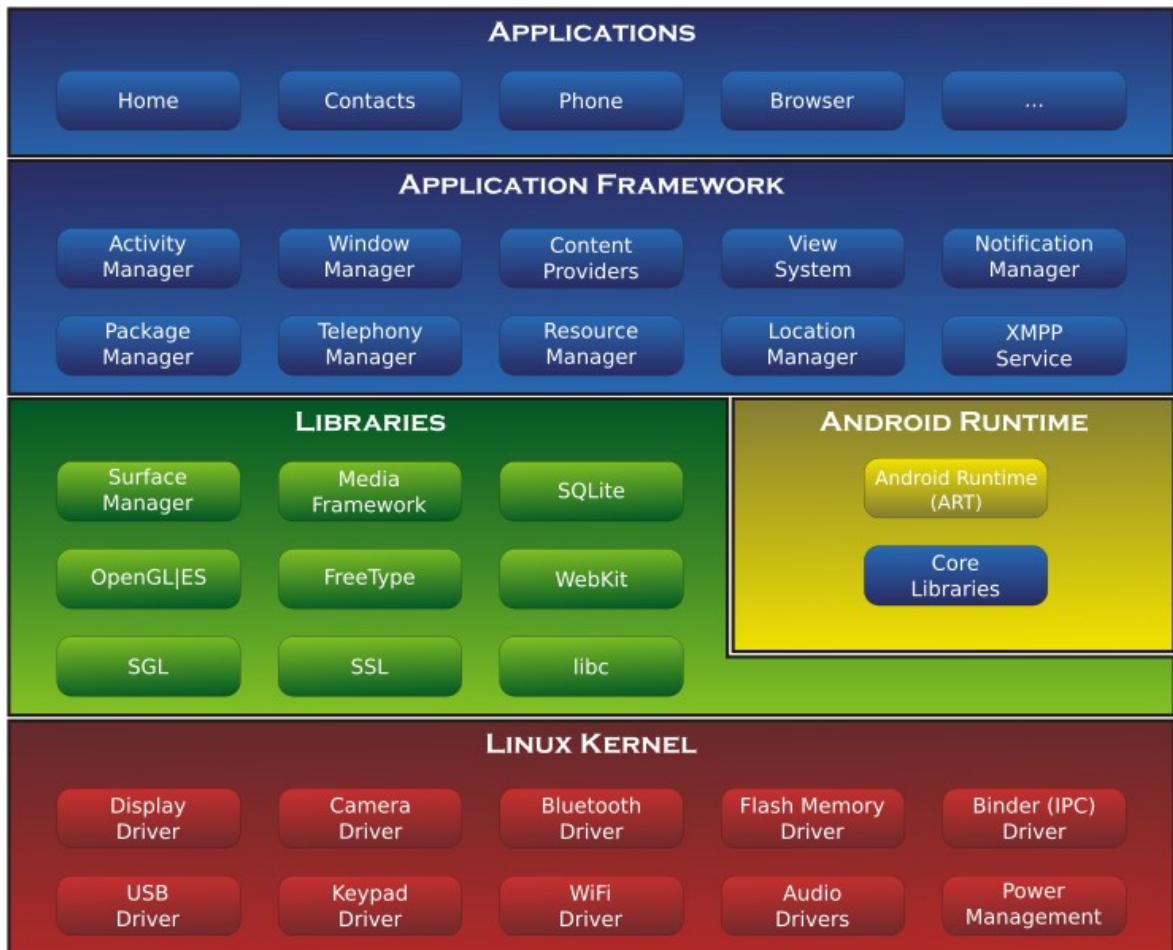
Android je mobilní operační systém postavený na linuxovém jádře. Poprvé byl vyvinut společností Android, Inc. a v roce 2005 jej koupila společnost Google, která ho vyvíjí dodnes.[18][19][20]

Systém Android vyvíjí i OHA (Open Handset Alliance). Je to koalice, která byla vytvořena s cílem vyvíjet otevřené standardy mobilních zařízení. V jejím čele je společnost Google, která se podílela na jejím vzniku. Má 84 členských společností, patří tam zejména NVIDIA Corporation, Samsung Electronics, Acer Inc., T-Mobile a Google Inc.. Členové koalice jsou především mobilní operátoři, výrobci mobilních telefonů a počítačů, softwarové společnosti a komercializační společnosti. Snaží se zejména urychlit inovaci v oblasti mobilních zařízení, nabídnout spotřebitelům co nejlepší mobilní zážitek za nižší celkové náklady a vytvořit prostředí přívětivé pro vývojáře.[21][22][23]

Systém Android je open-source. Znamená to, že zdrojový kód je volně přístupný komukoliv a lze ho využít i ke komerčnímu využití. Tím se liší od Apple iOS, který je closed-source.[24]

2.1 Architektura

Architektura Android, viz Obrázek 4, se skládá z pěti úrovní, kterými jsou aplikace, aplikační rámce, běhové prostředí systému Android, nativní knihovny a linuxové jádro.[25]



Obrázek 4 - Architektura Android[26]

2.1.1 Linuxové jádro

Linuxové jádro je nejspodnější vrstvou architektury a tvoří její naprostý základ. Je zodpovědné za propojení mezi hardwarovými zařízeními a vyššími vrstvami operačního systému. Obsahuje všechny základní hardwarové ovladače jako např. ovladače kamery, klávesnice, displeje atd.[27][28]

Základní funkce linuxového jádra se skládají z rozhraní systémového volání, řízení procesu, správy paměti, správy zařízení, virtuálního souborového systému, síťového zásobníku, multitaskingu a ovladačů zařízení. Plní i bezpečnostní funkci, kde jádro se stará o zabezpečení mezi systémem a aplikací.[27][28][29]

Nativní knihovny spolu s běhovým prostředím systému Android existují zhruba ve stejném prostoru.[25]

2.1.2 Nativní knihovna

Nativní knihovna je vrstva, která se skládá ze sady různých základních knihoven psaných v C/C++ nebo založených na Javě a běhových prostředím systému Android. Většina knihoven má veřejně přístupný zdrojový kód. Jedny z významnějších nativních knihoven jsou např. Surface Manager, SQLite, WebKit, Open GL/ES, dále potom Media Framework, FreeType, SSL, SGL a libc.[25][27][28]

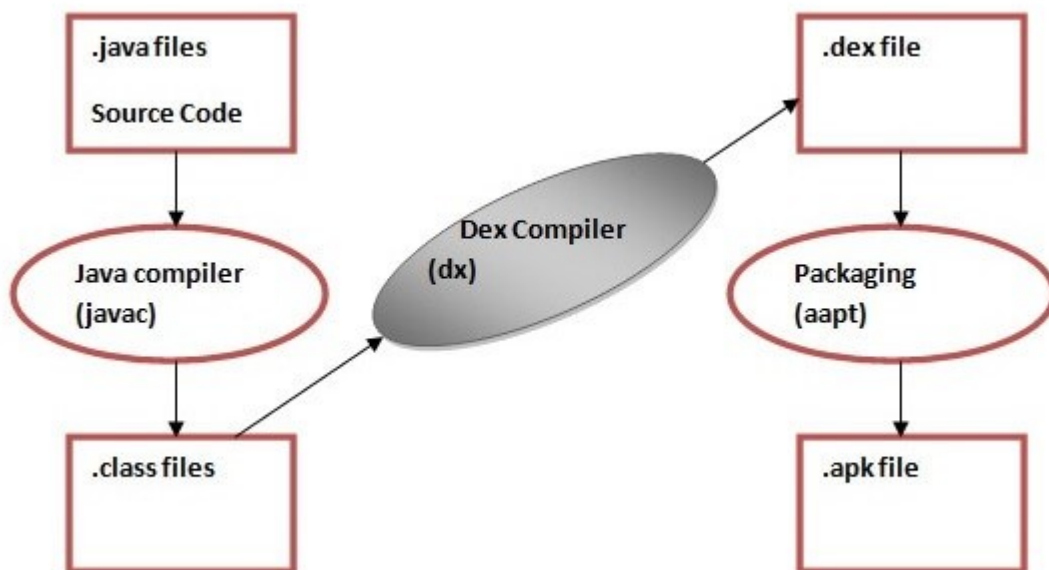
- Surface Manager spravuje přístup k zobrazovacímu subsystému. Skládá 2D a 3D grafické vrstvy z různých aplikací.[28]
- SQLite je relační databáze sloužící k zachování dat v zařízení Android. Je uložena ve vnitřní paměti systému Android. Nevyžaduje ke svému spuštění server ani žádnou instalaci. Je potřeba minimální podpory operačního systému, díky tomu je SQLite samostatná a použitelná v jakémkoliv prostředí.[25][30]
- WebKit je modul, který zajišťuje všechny funkce pro rychlé a efektivní vykreslování a zobrazování webového obsahu na Androidu. Je výchozím prohlížečem v systému Android a iOS.[25]
- Open GL je multiplatformní grafické rozhraní API pro vykreslování pokročilých 2D a 3D objektů v zařízení Android. Podporuje hardwarovou akceleraci. Open GL/ES je varianta specifikace Open GL, která je určena pro vestavěná zařízení.[25][31]
- Media Framework slouží pro přehrávání a nahrávání audio a video formátů. Umožňuje snadno integrovat zvuk, video a obrázky do aplikací.[28]
- FreeType je volně dostupná softwarová knihovna pro vykreslování bitmapových a vektorových písem.[32]
- Secure Sockets Layer, zkratka SSL, je bezpečnostní technologie pro zabezpečení internetového připojení mezi webovým serverem a webovým prohlížečem, a to pomocí šifrování. Jedná se právě o protokol pro vytváření těchto šifrovaných spojení. Přenášena data jsou nečitelná, díky tomu jsou citlivá data, jako např. osobní údaje, chráněny před zneužitím útočníky.[33][34]
- Skia graphics library, zkratka SGL, je nízko úroňová open-source grafická knihovna, která se stará o vykreslování 2D objektů.[35][36]
- Libc je standardní knihovna v programovacím jazyce C.[35]

2.1.3 Běhové prostředí systému Android

Běhové prostředí systému Android je třetí částí architektury a je jedním ze základních stavebních prvků operačního systému Android. Obsahuje dvě primární komponenty, a to základní Java knihovny a virtuální stroj Dalvik, dále jako DVM.[25][28]

DVM je velmi podobný virtuálnímu stroji Java, dále jako JVM. Vzhledem k tomu, že JVM není kompatibilní s mobilními zařízeními, vznikl z tohoto důvodu DVM. Narozdíl od JVM, je DVM speciálně navržený a optimalizovaný pro mobilní zařízení Android.[37][38]

Zajišťuje lepší výkon, správu paměti a výdrž baterie pro mobilní zařízení s nízkou spotřebou. Byl vyvinut tak, aby na zařízení mohlo běžet více virtuálních počítačů. Je založený na registrech a převádí soubory Java třídy do formátu .dex pomocí kompilátoru Dex. Ten je součástí DVM a slouží pouze k převodu vygenerovaného souboru .class na spustitelný soubor .dex, který se dále převádí na soubor .apk. Obrázek 5 zobrazuje průběh převodu.[37][38][39]



Obrázek 5 - DVM – Kompilátor Dex[40]

2.1.4 Aplikační rámce

Aplikační rámce tvoří vrstvu, která poskytuje mnoho tříd a rozhraní pro vývoj aplikací pro Android. Poskytuje také služby vyšší úrovně aplikacím ve formě tříd Java. Patří sem zejména tyto klíčové služby – správce aktivit, správce oken, poskytovatelé obsahu, správce zdrojů, správce oznámení, systém zobrazení, správce balíčků, správce telefonování, správce polohy.[27][28][37][41]

- Správce aktivit (Activity Manager) je třída, která poskytuje informace o procesech a službách. Řada metod v této třídě se používá k testování a ladění. Řídí zásobník aktivit, které se nacházejí v různých stavech – spuštění procesu, běžící proces, pozastavený proces, ukončený proces a zrušený proces.[41][42]
- Správce oken (Window Manager) je systémový software, který řídí vzhled a umístění oken v grafickém uživatelském rozhraní (GUI).[43]
- Poskytovatelé obsahu (Content Providers) umožňují jedné aplikaci sdílet data s jinými aplikacemi.[27]
- Správce zdrojů (Resource Manager) poskytuje přístup k nekódovým vloženým zdrojům, jako je definice rozvržení, řetězce uživatelského rozhraní, bitmapy atd.[27][44]
- Správce oznámení (Notification Manager) informuje uživatele o událostech spuštěných na pozadí. Jestliže dojde k události, upozorní uživatele, že se na pozadí něco stalo, a to zobrazením oznámení. Oznámení má různé formy – vizuální, zvukové či vibrační signály.[45]
- Systém zobrazení (View System) je rozšiřitelná sada zobrazení k vytvoření uživatelského rozhraní aplikace.[46]
- Správce balíčků (Package Manager) poskytuje informace o nainstalovaných aplikacích v zařízení. Jedná se zejména o informace týkající se instalace, odinstalace, různých oprávnění aplikace atd.[37]
- Správce telefonování (Telephony Manager) poskytuje informace o telefonních službách, např. ID uživatele, typ telefonní sítě, stav telefonu atd.[37]
- Správce polohy (Location Manager) poskytuje přístup k aktuální poloze pomocí GPS, které má většina Android zařízení.[37]

2.1.5 Aplikace

Aplikace jsou nejvyšší vrstvou architektury Android. Jsou vyvíjeny tak, aby se instalovaly pouze na tuto vrstvu. Jedná se o předinstalované aplikace pro Android a aplikace nainstalované třetími stranami. Nejčastěji jsou distribuované z obchodu Google Play. Tyto aplikace slouží ke každodennímu použití, patří sem např. kontakty, zprávy, e-mail, galerie, fotoaparát, hudba, hry apod.[25][27]

2.2 Historie

Společnost Android Incorporation byla založena v říjnu 2003 v Kalifornii, a to rovnou čtyřmi zakladateli – Andy Rubin, Rich Miner, Nick Sears a Chris White. Původně bylo cílem firmy vyvinout a zlepšit operační systémy pro digitální fotoaparáty, ale protože trh s digitálními fotoaparáty byl velmi nízký, změnil se později na operační systém pro chytré telefony. Název Android vznikl díky přezdívce Andyho Rubina, který měl vášeň k robotům.[47][48]

V roce 2005 společnost Android Inc. koupila americká společnost Google Inc. a tím se stala vlastněnou dceřinou společností. Částka, za kterou byla společnost prodána, nebyla nikdy nikde zveřejněna. Andy Rubin a další tři zakladatelé nadále pokračovali ve vývoji OS pod novými vlastníky. Ve společnosti Google se tým vedený Rubinem rozhodl, že jako základ pro OS Android použijí operační systém založený na Linuxu.[47][49]

2.3 Verze systému Android

Zajímavostí je, že názvy verzí Androidu jsou sladkosti a jejich názvy začínají prvním písmenem abecedy a pokračují v její posloupnosti.[48]

Android 1.0, neoficiální název Apple Pie, je úplně prvním systémem, na kterém Android začínal. V listopadu 2007 byla spuštěna beta verze pro vývojáře a v září 2008 byl uveden první mobilní telefon s tímto systémem, a to T-Mobile G1, nazývaný také HTC Dream. Zahrnoval omezené funkce, jako je Android Market, webový prohlížeč HTML, Gmail, Mapy Google, YouTube a další. Úroveň API je 1.[47][49][50]

Dále existovaly verze jako Android 1.1 (neoficiální název Banana Bread), Android 1.5 Cupcake, Android 1.6 Donut, Android 2.0-2.1 Eclair, Android 2.2 Froyo, Android 2.3 Gingerbread, Android 3.0 Honeycomb, Android 4.0 Ice Cream Sandwich, Android 4.1-4.3 Jelly Bean, Android 4.4 KitKat, Android 5.0 Lollipop, Android 6.0 Marshmallow, Android 7.0 Nougat, Android 8.0 Oreo, Android 9.0 Pie. Poté následovaly verze již pouze očíslované – Android 10, Android 11 a nejnovější Android 12. S každou verzí se zvyšovala úroveň API. [47]

2.3.1 Android 10 a 11

Android 10 a Android 11 do mobilního světa přinesly spoustu nových a vylepšených funkcí, které usnadnily uživateli snadnější a pohodlnější ovládání svého zařízení. Obě verze se dost

zaměřovaly na co nejlepší ochranu osobních údajů – pomocí ovládacích prvků se uživatel může rozhodnout, kterým aplikacím udělí či neudělí přístup k osobním datům. Hodně pokročily funkce týkající se chatovacích služeb. Nyní lze díky multitaskingu chatovat a zároveň dělat jinou činnost najednou. Při chatování se nám nabízí „rychlá odpověď“, aniž bychom museli do konverzace vstupovat. Mobilní telefony mají vestavěné nahrávání obrazovky a zvuku, tzn. že není potřeba stahovat žádnou aplikaci třetích stran.[51][52]

2.3.2 Android 12

Nejnovější verze Android 12 byla vydána v říjnu 2021. K největší změně došlo po grafické stránce a jedná se o největší facelift Androidu. Aby byl Android přístupný většině uživatelů, je navržen i pro lidi, kteří mají špatný zrak. Poskytuje jim různé funkce jako nastavování kontrastu barev, zvětšování písma, ikoněk apod.[47][53]

Uživatelské rozhraní je kompletně přepracováno. Stojí za tím především systém Material You společnosti Google a nabízí uživatelům mnohá vylepšení co se týče grafických prvků. Jsou změněny tvary, světla, animace, pohyby, a dokonce i barvy systému, které lze individuálně měnit podle sebe. Tvary tlačítek a oznámení jsou více zakulacené a animace jsou plynulejší. Navigační lišta byla také obměněna. Původní malé kulaté ikonky jsou nyní nahrazeny většími dlaždicemi s podrobnějšími informacemi. Objevují se zde i nové funkce, např. funkce k ovládní chytré domácnosti.[47][53]

Novou velmi zajímavou funkcí je automatická extrakce motivů z tapet. Systém si dokáže vytáhnout klíčové barvy z vybrané tapety a přizpůsobit zbytek celého rozhraní. Změny se projevují především u ikoněk, oznámení, lišty, widgetů, a dokonce i u vybraných aplikací.[53]

Android 12 přináší velké změny i v oblasti bezpečnosti a soukromí. Nové funkce poskytují absolutní kontrolu nad daty a chrání osobní údaje. Nyní lze sledovat, kdo využívá mikrofon nebo kameru v zařízení, a to pomocí indikátoru v navigační liště.[53]

Nyní je možné si vybrat, kterým aplikacím chceme poskytnout přístup k přesné poloze a kterým stačí pouze přibližná poloha. Například u navigace budeme chtít nastavit přesnou polohu, avšak u počasí nám postačí vědět přibližná poloha zařízení.[53]

2.4 Logo Android a jeho historie

Logo Android vzniklo v roce 2008 a vytvořila jej Irina Blok spolu se svými kolegy. Vzhledem k tomu, že logo mělo být spojené s jeho názvem, jedinou podmínkou majitelů

bylo, aby vypadalo jako robot. Inspirací k vytvoření loga se pro ni staly piktogramy univerzálního muže a ženy, které často bývají na dveřích toalet. Jsou složeny z kruhů, trojúhelníků a čtverců.[47][54]

První verze robota byla vytvořena ve dvou barvách - v černobílé a světle zelené. Světle zelený odstín byl zvolen z důvodu, že může kontrastovat téměř se všemi barvami, zejména s tmavými.[54]

V roce 2014 prošlo logo redesignem. Obrázek robota byl zachován, jen se vylepšily obrysy. Tím se stalo logo štíhlejší a úhlednější. Zelená barva byla zachována, jen byl použit tmavší odstín, který byl výraznější.[55]

K další změně došlo v roce 2019, kdy logu byla zachována pouze horní část hlavy a bylo doplněno o nápis „android“ v černé barvě. Obrázek 6 znázorňuje celou historii loga Android. [55]

Irina Blok se svými kolegy a se společností Google se rozhodli, že logo bude open-source, stejně jako samotný software. Lidé si tak mohou denně vytvářet a upravovat své vlastní verze Android loga.[47]



Obrázek 6 - Historie loga Android[54]

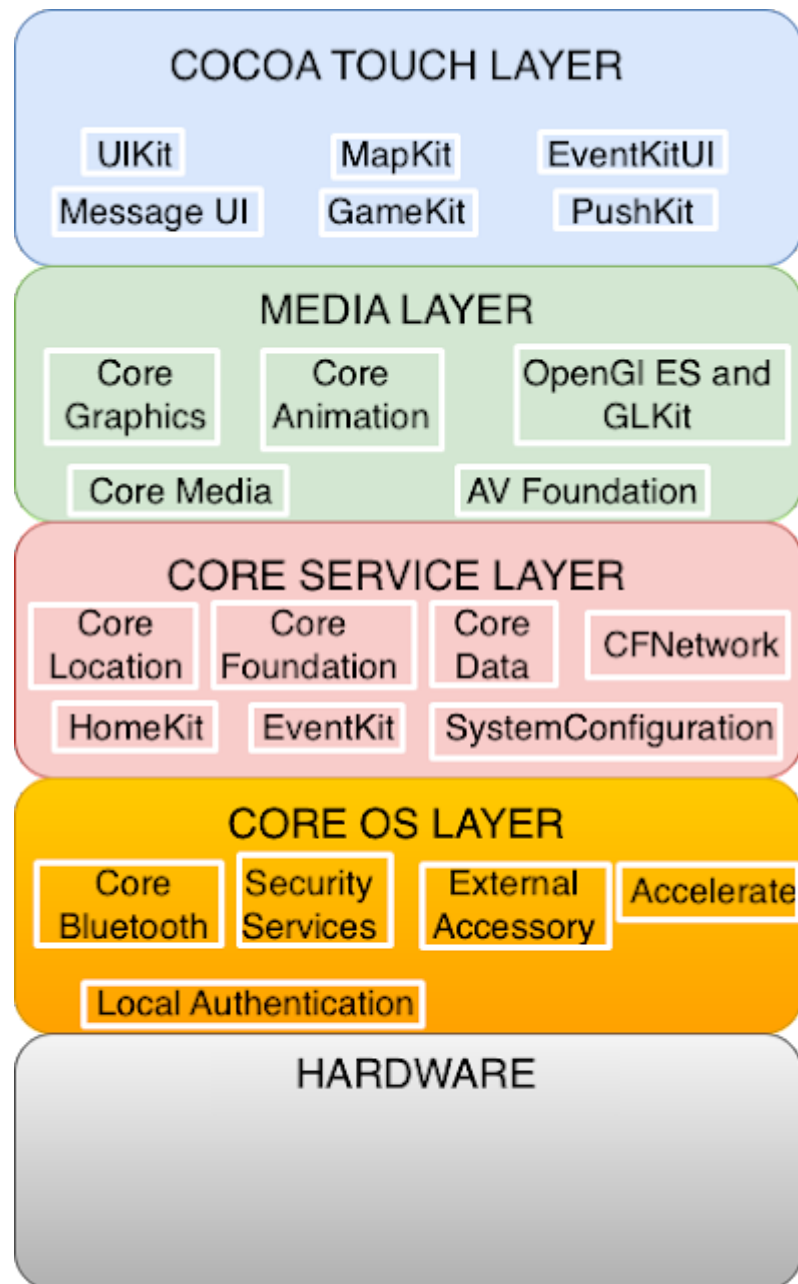
3 IOS

iOS je operační systém založený společností Apple a je určen pouze pro mobilní zařízení, které tato firma vyrábí, jedná se především o iPhone, iPad, iPod Touch apod. Je to druhá nejrozšířenější platforma používána v mobilních zařízeních, hned po operačním systému Android.[56][57]

System iOS je closed-source a tím Applu zajišťuje naprostou kontrolu nad softwarem i hardwarem. Lze instalovat pouze aplikace z App Storu, které Apple průběžně kontroluje. System je proto více bezpečný a je menší riziko napadení jej útočníkem. Některé komponenty systému jsou open-source.[58][59][60]

3.1 Architektura iOS

Architektura iOS, viz Obrázek 7, se skládá ze 4 odlišných vrstev – Cocoa Touch, Media Layer, Core services, Core OS. Každá vrstva poskytuje programové rámce pro vývoj aplikací, které běží nad operačním rámcem.[61]



Obrázek 7 - Architektura iOS[62]

3.1.1 Cocoa Touch

Cocoa Touch je vrstva, která se nachází na vrcholu architektury systému iOS. Obsahuje rámce, které používají především vývojáři aplikací pro iPhone. Důležitá je pro svůj způsob komunikace se zařízeními iOS. Stanovuje chování aplikace – řídí, jak aplikace vypadají a jak reagují při používání. Popisuje základní architekturu aplikace a zahrnuje řadu důležitých funkcí jako jsou dotykové a pohybové události, možnost multitaskingu a další.[63][64]

Poskytuje především následující rámce:

- EventKit zobrazuje standardní systémová rozhraní systému pomocí řadičů zobrazení pro zobrazování a změn událostí spojené s kalendářem.[65]
- GameKit poskytuje uživatelům podporu pro Game Center, která umožňuje online sdílení dat související s hrou.[65]
- MapKit poskytuje mapu nebo satelitní snímky, které lze zahrnout do uživatelského rozhraní aplikace.[66]
- iAd dovoluje vývojářům třetích stran vkládat bannerové reklamy do aplikací.[67]

3.1.2 Media Layer

Media Layer, v překladu mediální vrstva, je druhá vrstva v architektuře, která umožňuje zvukové, vizuální a další multimediální funkce.[65]

Poskytuje především tyto rámce:

- U grafiky zejména UIKit, Core Graphics Framework, Core Animation, Core Images atd. UIKit poskytuje podporu pro navrhování obrázků a animování obsahu zobrazení. Core Graphics poskytuje podporu v rámci 2D vektorového vykreslování a vykreslování obrázků a je nativním kreslicím jádrem (enginem) pro iOS aplikace. Core Animation pomáhá optimalizovat animaci aplikací. Core Images umožňuje používat filtry ke zpracování videí a nepohyblivých obrázků.[67][68]
- U audia zvláště Media Player Framework a OpenAL. Media Player Framework slouží k přehrávání médií, např. skladeb, a umožňuje uživateli používat knihovnu iTunes. OpenAL je multiplatformní 3D zvukové rozhraní API pro poskytování zvuku.[67][69][70]
- U videa kupříkladu AVKit, AV Foundation a Core Media. AVKit a AV Foundation poskytují uživatelská rozhraní pro přehrávání videí a pokročilé možnosti týkající se přehrávání a nahrávání videa. Core Media představuje audiovizuální prostředky, a to prostřednictvím základních datových typů.[67][71]

3.1.3 Core Services

Core Services, v překladu základní služby, je třetí vrstvou architektury iOS, která poskytuje základní služby aplikacím.[61]

Poskytuje především tyto rámce:

- Adress Book Framework umožňuje uživateli přístup k databázi kontaktů a lze s ní manipulovat – přidávání a odebírání kontaktů.[64]
- CloudKit Framework slouží pro přesun dat mezi aplikací a iCloudem.[64]
- Core Data Framework slouží pro správu aplikací založených na Model View Controller.[64]
- Core Foundation Framework poskytuje základní funkce správy dat a služeb pro iOS aplikace.[64]

3.1.4 Core OS

Core OS je nejnižší vrstvou architektury. Jedná se o všechny technologie iOS, které jsou založeny na funkcích nízké úrovně. Mezi tyto technologie patří např. Core Bluetooth Framework, Accelerate Framework, External Accessory Framework a další.[65]

3.2 Nástroje iOS

Pomocí nástrojů iOS si lze usnadnit proces celého vývoje aplikace a pomáhají vytvořit aplikaci podle specifických požadavků. Existuje mnoho nástrojů. Jsou různé specifické oblasti nástrojů – editory, nástroje pro design, pro ladění, pro automatizaci apod.[72][73]

3.2.1 Xcode

Xcode je integrované vývojové prostředí od společnosti Apple a slouží pro tvorbu aplikací pro iOS zařízení. Obsahuje kompletní sadu vývojářských nástrojů. Xcode je zdarma ke stažení a lze z něj publikovat aplikace přímo na App Store. Jedinou podmínkou je mít založený účet pro vývojáře. Podporuje několik jazyků, především však Objective-C, Swift, a Javu.[72][73][74]

3.2.2 AppCode

AppCode je také integrované prostředí, stejně jako Xcode, s rozdílem, že je postaven na platformě IntelliJ IDEA od společnosti JetBrains. Testování aplikace je mnohem jednodušší oproti Xcode, a to díky integraci rámců. AppCode lze stáhnout jako 30denní trial verze, ve vzdělávacích institucích je k dispozici zdarma. Podporuje webové technologie jako Javascript, HTML a CSS.[72][75]

3.2.3 iOS Simulator

iOS Simulator poskytuje spustit aplikaci v simulovaném zařízení nebo ve svém vlastním zařízení pro její otestování. Pomáhá prototypovat a ladit aplikace. Cílem je vytvořit fungující aplikaci bez chyb, a to dříve, než je publikována na App Store.[72][76]

3.2.4 Test Flight

Test Flight je nástroj pro testování beta verzí aplikací od ostatních uživatelů. Ti poté poskytují zpětnou vazbu vývojáři přímo z testované aplikace – pomocí snímku obrazovky. Testeři mohou být interní a externí. Interní testeři mají oproti externím vyšší pravomoce. Plní roli majitele účtu, roli správce, správce aplikace a vývojáře. Maximální počet interních testerů je omezen na 100 členů týmu a volí si je sám vývojář. Externí testeři jsou pozváni pomocí e-mailových adres nebo sdílením veřejného odkazu, který slouží jako pozvánka k testování aplikace. U externích testerů je potřeba znát jejich konkrétní e-mail, aby bylo možné zaslat odkaz k testování anebo pomocí otevřeného odkazu, který lze sdílet na sociálních sítích či jiných platformách pro zasílání zpráv. Tento nástroj je ke stažení zdarma na App Store a slouží pouze k testování aplikace pro iOS, iPadOS, macOS, tvOS, watchOS a iMessage.[77]

3.2.5 Transporter

Transporter je nástroj příkazového řádku, který slouží k doručení obsahu přímo do společnosti Apple. Vývojář může jednoduše odesílat aplikace, hudbu, filmy, televizní pořady nebo knihy k distribuci na App Store, Apple Music, Apple Books nebo iTunes Store. Před odesláním obsahu žádá aplikace ověření – jsou tři možnosti. Buď ověří účet iTunes Connect nebo App Store Connect anebo pomocí vlastního účtu pro kódování.[72]

3.3 Historie

Společnost Apple Computer, Inc., nyní již jen Apple Inc., založili 1. dubna 1976 dva bývalí spolužáci střední školy – Steve Jobs a Steve Wozniak. Jejich cílem bylo vynaleznout malý počítač, který bude uživatelsky přívětivý. Jejich první model Apple I byla prostá základní deska s procesorem, pamětí RAM a se základními textovými a grafickými čipy. Apple 1 byl k prodeji, ale postrádal klávesnici, monitor a pouzdro, které bylo přidáno následně v roce 1977.[78][79]

Tím vznikl Apple II (Obrázek 8), kde včetně přidání plastového pouzdra nabízel i první barevný displej. Apple II způsobil revoluci v počítačovém odvětví a oslovil tak i běžného člověka. Jednalo se o první mikropočítač značky Apple.[79]



Obrázek 8 - Apple II[80]

4 OBCHODY S MOBILNÍMI APLIKACEMI

Mezi tyto obchody řadíme Google Play a App Store. Každý z obchodů podporuje pouze svou vlastní mobilní platformu. Pro Google Play je to Android a pro App Store je to iOS.

Lze v nich najít ke stažení aplikace, hry, filmy, a dokonce i knihy. Některé tyto produkty jsou placené, ale většina je zdarma.

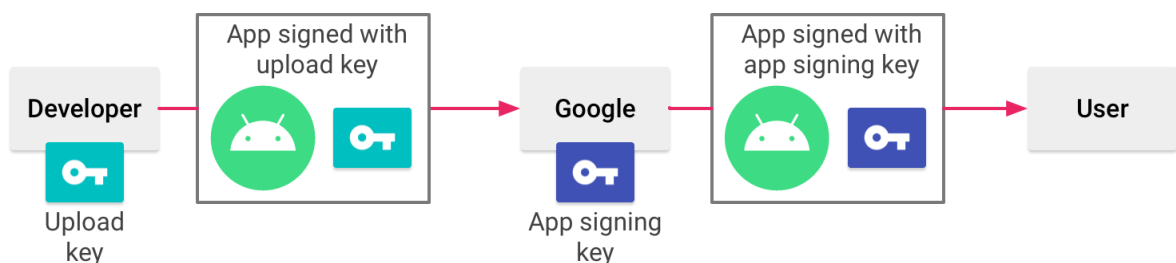
4.1 Google Play

Google Play je obchod s mobilními aplikacemi určený pouze pro Android zařízení. Lze v něm stáhnout aplikace, různé hry, koupit nebo půjčit ke stažení film či knihu. Dokonce se tam dá koupit i hudba. Aplikace jsou většinou ke stažení zdarma, ale občas se najdou i placené.

4.1.1 Podmínky publikování

K publikování první aplikace musí být založený vývojářský účet. Za založení účtu se platí jednorázový poplatek ve výši 25 USD (přibližně 550 korun).[81]

Aplikace Play používá k podepisování dva klíče – podpisový klíč a klíč pro nahrání. APK soubory musí být digitálně podepsány pomocí podpisového klíče. Pro podepsání je nutné si pro aplikaci vygenerovat klíč, kterým je nutné podepisovat i budoucí verze aplikace. Při vydávání pomocí balíčků Android App Bundle je třeba balíček aplikace před nahráním do Play Console podepsat klíčem pro nahrávání. Celý postup podepisování je zobrazen viz Obrázek 9. Aplikace musí být testovaná pro vydanou verzi. Maximální velikost APK souboru je 150 MB. Aplikace vytvořené před srpnem 2021 měli maximální velikost 100MB.[81][82][83][84][85]



Obrázek 9 - Podepisování aplikace pomocí Play App Signing[83]

4.1.2 Postup publikování

Po vytvoření vývojářského účtu Google Play lze vytvářet a nastavovat aplikace pomocí služby Play Console. V této službě je možné vytvořit aplikaci přes záložky Všechny aplikace

→ Vytvořit aplikaci. Zde se zadávají parametry aplikace. Některé parametry jdou v budoucnu změnit. Zadává se název aplikace, pod kterým se bude aplikace dát vyhledat na Google Play, dále se vybírá výchozí jazyk aplikace. Zvolí se, zda se jedná o aplikaci či hru a zda bude bezplatná nebo placená. Zadává se e-mail pro případnou komunikaci s uživateli Google Play ohledně aplikace. Potvrzuje se deklarace programových zásad pro vývojáře a vývozní zákony Spojených států amerických a přijímají se smluvní podmínky služby Google Play. Dále se nahraje vytvořený a podepsaný soubor APK. Před nahráním souboru je nutné vytvořit vydání aplikace. Vybírá se typ vydání, do kterého se nahraje první verze aplikace. Lze si vybrat mezi interním, uzavřeným a otevřeným testem, který umožňuje testovat aplikaci pouze vybraným uživatelům nebo vybrat produkční vydání aplikace. U tohoto typu vydání je verze aplikace přístupná všem. Jakmile jsou všechny kroky splněny, aplikace se vytvoří po kliknutí na tlačítko Vytvořit aplikaci.[84][86][87]

4.2 App Store

App Store je obchod s mobilními aplikacemi určený pouze pro iOS zařízení. Lze v něm stáhnout aplikace či hry. Aplikace jsou placené i neplacené. Oproti Google Play je App Store poměrně drahý, co se týče cen za aplikace.

4.2.1 Podmínky publikování

K publikování první aplikace musí být založený vývojářský účet – Apple Developer. Při založení účtu se platí ročně 99 USD (přibližně 2180 korun).[81]

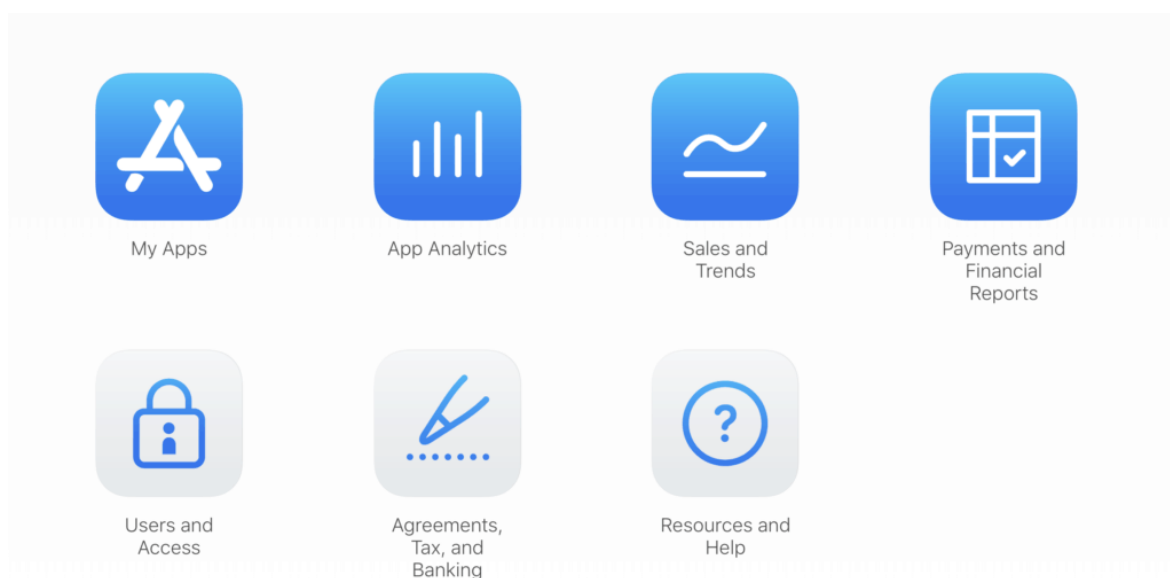
Aplikace musí být řádně otestována před publikací na App Store. Metadata a všechny informace o aplikaci jsou přesné a úplné. Aplikace se řídí dokumentací – směrnicemi. Každá aplikace musí projít procesem certifikace. Před publikováním do App Store se musí aplikace odeslat do Applu k zhodnocení. Odesílají se snímky obrazovky, popis aplikace, ikony a další potřebné informace.[88][89][90]

4.2.2 Postup publikování

Po vytvoření vývojářského účtu v Apple Developer lze přejít na hlavní platformu s názvem App Store Connect, kde se aplikace skutečně nahrávají.[91]

App Store Connect je webová stránka, ke které je přístup povolen pouze po registraci vývojářského účtu. Jedná se o portál pro správu aplikací a věcí spojených s App Store.[92]

Po přihlášení do App Store Connect se zobrazí hlavní menu. Pro vytvoření nové aplikace je nutné kliknout na Moje aplikace → tlačítko +. Poté je uživatel povinen vyplnit veškeré podrobnosti o aplikaci. Jedná se o název aplikace, její popis, kategorie, minimální věk uživatele a zásady ochrany osobních údajů spolu s adresou URL podpory. Vybírá se platforma, výchozí jazyk, ID balíčku a SKU. Dále se rozhoduje, zda aplikace bude či nebude zpoplatněná. Aplikace je archivována a nahrána pomocí Xcode. Jakmile aplikace projde certifikačním procesem, je na výběr ze tří možností, jak má být vydána. Lze být vydána manuálně, to znamená, že vývojář aplikaci může vydat kdykoliv on uzná za vhodné. Další způsob je vydat aplikaci v přesně stanovený datum, který si vývojář sám nastavil. Podmínka u těchto dvou možností je pouze ta, že aplikace musí projít certifikací. Poslední možností je okamžité vydání aplikace. Jakmile projde certifikací, aplikace se automaticky nahraje na App Store aniž by vývojář musel cokoli dělat. Obrázek 10 zobrazuje řídicí panel App Store Connect.[91][92][93][94]



Obrázek 10 - Řídící panel App Store Connect[92]

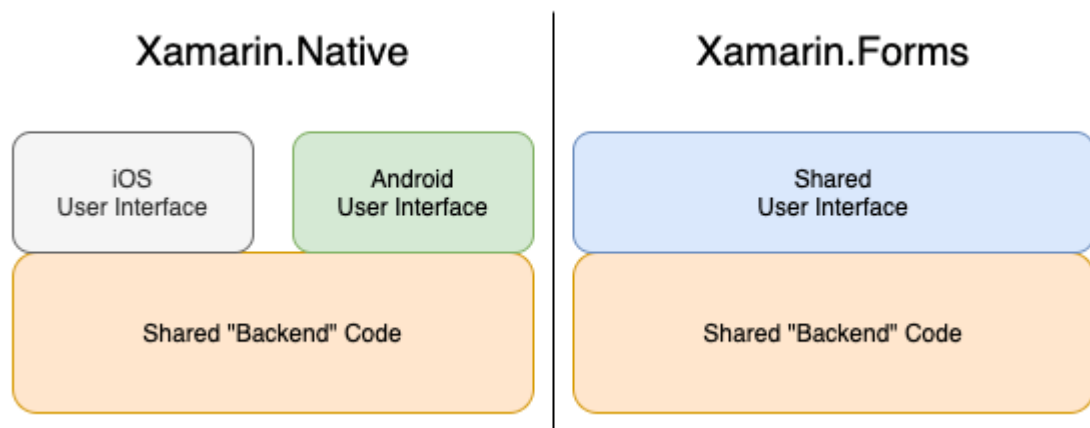
5 VÝVOJ MOBILNÍCH APLIKACÍ NA PLATFORMĚ .NET

.NET je open-source vývojářská platforma vytvořená společností Microsoft. Skládá se z nástrojů a knihoven pro vytváření mnoha různých typů aplikací. Aplikace lze psát v programovacím jazyce C#, F# nebo Visual Basic.[95]

5.1 XAMARIN

Xamarin je vývojová platforma, která umožňuje programovat nativní a multiplatformní Android a iOS aplikace v programovacím jazyce C#. Je postaven na platformě .NET. Umožňuje vývojářům sdílet až 90 procent kódu napříč platformami.[96][97]

Lze vytvořit dva typy projektů Xamarin – Xamarin.Native a Xamarin.Forms. Oba typy projektů mají jednu kódovou základnu. Rozdíl mezi těmito typy je v uživatelském rozhraní, které se u Xamarin.Native píše pro každou platformu zvlášť – iOS používá Storyboards, Android používá AXML a UWP používá XAML. U Xamarin.Forms je možné napsat uživatelské rozhraní deklarativně v kódu nebo pomocí XAML. Obrázek 11 znázorňuje rozdíl mezi Xamarin.Native a Xamarin.Forms.[97][98][99]



Obrázek 11 - Rozdíl mezi Xamarin.Native a Xamarin.Forms[100]

5.1.1 Xamarin.Native

Xamarin.Native, také často označováno jako Xamarin.Android a Xamarin.iOS, je sada nativních nástrojů pro vývoj aplikací v Xamarinu. Platforma obsahuje individuální nástroje pro vývoj pro Android, iOS. Pomáhá vytvářet aplikaci uživatelské rozhraní, které je specifické pro danou platformu.[101][102]

5.1.2 Xamarin.Forms

Xamarin.Forms je framework Xamarinu, která představuje sadu multiplatformních nástrojů pro uživatelské rozhraní. Rozšiřuje vývojářskou platformu .NET o nástroje a knihovny pro vznik mobilních aplikací v programovacím jazyce C# a XAML.[97][103]

Xamarin.Forms obsahuje třídy, které nejsou závislé na platformě. Lze vyvíjet základní nativní uživatelské rozhraní pro všechny tři platformy bez potřeby znát uživatelské rozhraní iOS či Android.[103][104]

Je postaven na třech knihovnách – Xamarin.Android pro Android, Xamarin.iOS pro iOS a UWP pro Universal Windows. Všechny knihovny mají přístup k standardní knihovně .NET.[103]

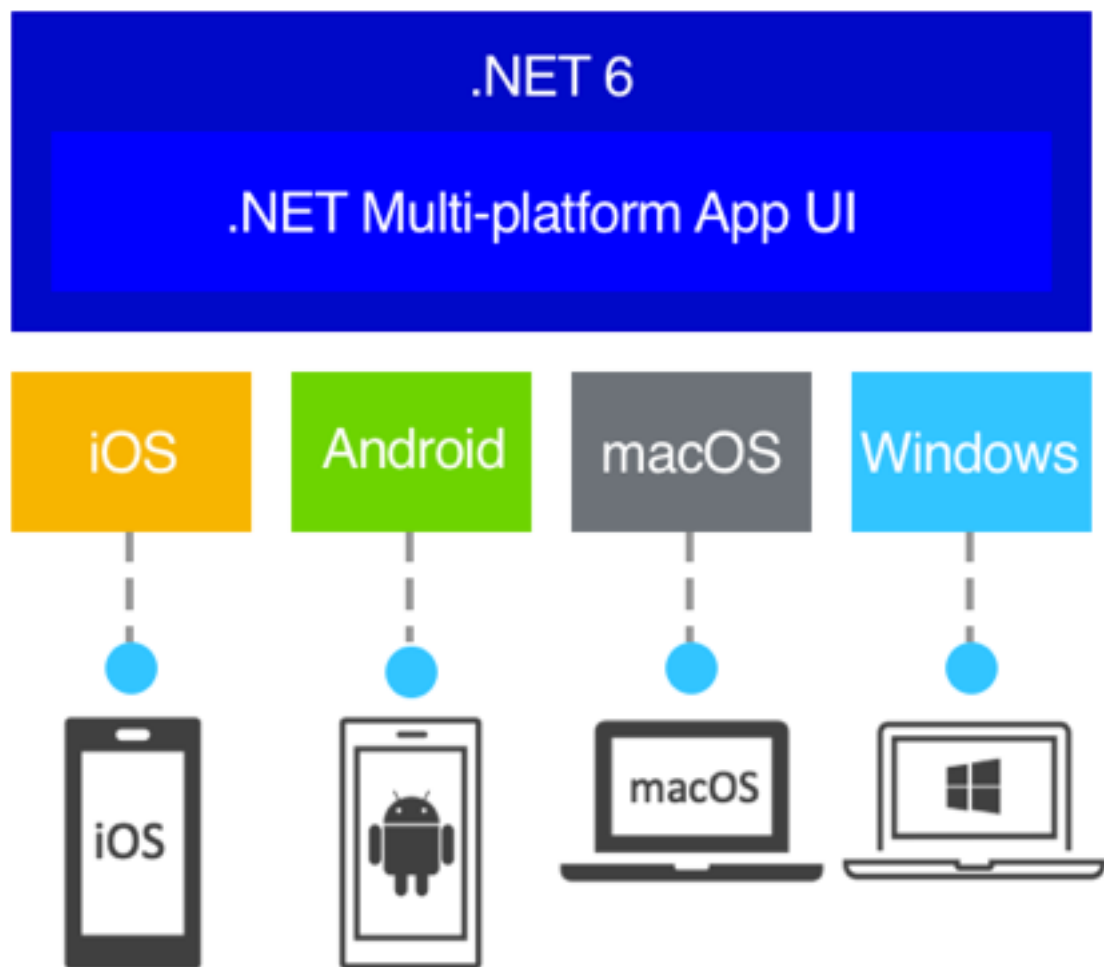
XAML je zkratka pro Extensible Application Markup Language. Je to jednoduchý a deklarativní jazyk založený na XML od Microsoftu. Slouží především k návrhu uživatelského rozhraní, ale lze ho použít i pro deklaraci workflow ve Workflow Foundation.[105]

5.2 .NET MAUI

.NET MAUI je zkratka pro Multi-platform App UI. Je to multiplatformní framework pro tvorbu nativních a desktopových aplikací v programovacím jazyce C# a XAML. Je evolucí Xamarin.Forms, a to zejména díky možnostem vytvářet i nativní desktopové aplikace pro Windows a macOS. Aplikace lze vyvíjet společně pro Android, iOS, Windows a macOS z jediné sdílené kódové základny, to znamená pouze z jednoho kódu, viz Obrázek 12. Díky tomu je vývoj efektivnější a jednodušší. Uživatelské rozhraní je vytvořeno pouze jednou a vypadá stejně na všech platformách nebo je možné ho vytvořit samostatně pro jednu či více podporovaných platforem pomocí rozhraní API konkrétní nativní platformy. K tvorbě aplikací pro iOS a macOS je potřeba Mac.[106][107][108]

.NET MAUI podporuje opětovné načtení XAML za provozu. To znamená, že zdrojový kód je již možné spravovat, i když je aplikace spuštěná. Při ukládání souborů XAML lze pozorovat změny ve spuštěné aplikaci bez rekompilace.[107][109]

Tento framework byl plánován s .NET 6 v listopadu 2021. Nakonec se plán vydání posunuje na 2. čtvrtletí roku 2022 pro .NET MAUI GA. Momentálně je ve verzi Preview.[110]



Obrázek 12 - .NET MAUI[110]

II. PRAKTICKÁ ČÁST

6 NÁVRH APLIKACE

Rozhodla jsem se vytvořit mobilní aplikaci pro adopční kočičí kavárnu v Kroměříži. Je vytvořena tak, aby bylo možné ji v budoucnu rozšířit i pro další obdobné kavárny. Aby byla jednodušší implementace a dala se aplikace vyvíjet dále i do budoucna, vytvořila jsem si API, které zahrnuje nejzákladnější funkce pro chod aplikace.

Aplikace bude uživatelům poskytovat základní informace o kavárně, její sídlo, kontakt a také seznam koček, které lze adoptovat. Adopce se budě dělit na seznam koček, které lze fyzicky adoptovat a seznam koček, které budou pouze k virtuální adopci. Bylo využito seznamu karet – TabbedPage, kde každá karta je zaměřena na určitou oblast.

Rozdíl mezi fyzickou a virtuální je ten, že u klasické adopce bude možné kočku adoptovat a poté již bude probíhat komunikace s kavárnou – domluva, navštívení kavárny, seznámení s vybranou kočkou a jestliže bude vše v pořádku, uživatel si kočku fyzicky převezme a odjíždí s ní domů.

U virtuální adopce se jedná o adopci, kde se samozřejmě uživatel může s kavárnou domluvit a kočku navštívit a seznámit se s ní, ale již si ji nemůže vzít domů. Jedná se o to, že se např. měsíčně přispěje na konkrétní kočku určitá částka, která pokrývá výdaje jako je jídlo, léky, hračky pro kočky, pelíšky, škrabadla atd.

Oba seznamy budou ukazovat pouze základní údaje o kočce jako je jméno, pohlaví a rok narození.

Při kliknutí na určitou kočku se otevře detail kočky, kde se bude nacházet více informací, včetně obrázku kočky a tlačítka „Adoptovat“. Toto tlačítko bude aktivní pouze pro přihlášené uživatele. Pro nepřihlášené uživatele bude toto tlačítko neaktivní – podle požadavku kavárny, a to za účelem, aby nepřihlášený uživatel věděl, že když se přihlásí, bude mu tlačítko k dispozici.

6.1 Specifikace požadavků

První spojení s kavárnou proběhlo pomocí emailu, kde mi byla poskytnuta spolupráce v rámci mé aplikace. Poskytli mi informace o kočkách, jejich fotky a vysvětlili mi, jak vlastně tyto kočičí adopční kavárny fungují. Podle toho jsem navrhla aplikaci, přičemž každý návrh jsem konzultovala s nimi.

Bude možnost se do aplikace registrovat, přihlásit nebo pokračovat bez přihlášení. Uživatel si bude schopen vybrat kavárnu a zobrazit si v ní seznamy koček, a to koček určených ke klasické či virtuální adopci. Bude mu umožněno nahlédnout do detailu konkrétní kočky. Aplikace bude nabízet i možnost zobrazení již adoptovaných koček uživatelem.

Požadavky jsou více popsány v kapitole Wireframes.

6.2 Registrace a přihlašování

Aplikace bude umožňovat registraci a následné přihlášení. Uživatel není povinen být přihlášen, aby se dostal do základních funkcionalit aplikace jako je zobrazení seznamu kaváren, informace o kavárně, seznamy koček a detail kočky, ale přichází tím o nejdůležitější část aplikace, a to samotnou adopci o které se hlavně v této bakalářské práci jedná. Tlačítko adoptovat se nepřihlášeným uživatelům zobrazí, ale bude neaktivní.

Registrace je jednoduchá a vyžaduje po uživateli pouze email a 2x zadání hesla. Více polí k vyplnění osobních údajů by mohlo uživatele odradit se zaregistrovat. Při úspěšné registraci by se uživatel automaticky přihlásil a byly by mu dostupné všechny funkcionality aplikace.

Pokud bude již uživatel registrován, může se přihlásit svým emailem a heslem. Po přihlášení se uživateli zobrazí okno se seznamem kaváren, včetně tlačítka ikony profilu, která bude v pravém horním rohu. Toto tlačítko se bude zobrazovat pouze přihlášeným uživatelům. Při kliknutí na ikonu profilu se uživatel dostane do správy svého uživatelského účtu.

6.3 Správa uživatelského účtu

Správa uživatelského účtu se dělí na 3 oddíly. Budou se využívat seznamy karet neboli TabbedPage, kde každá karta se věnuje určité věci. První karta jsou pouze informace o uživateli a nachází se zde i tlačítko, které bude sloužit k jeho odhlášení. Následné dvě karty se budou věnovat adopci, kterou uživatel provedl. Jedna karta bude seznam adoptovaných koček a druhá karta bude seznam adoptovaných virtuálních koček. Podle toho, kterou kočku si uživatel adoptoval, se kočka přidá do konkrétního seznamu. Uživatel si opět bude moci rozkliknout detail každé kočky a podívat se na stav vyřízení jeho adoptce.

6.4 Adopce

Adopce je umožněna pouze přihlášenému uživateli. Tlačítko „Adoptovat“ se nachází na každém detailu kočky, a to na konci.

Při klasické adopci, po kliknutí na tlačítko „Adoptovat“ se uživateli změní text na „Čeká na potvrzení“. Uživateli se daná kočka přidá do seznamu adopce v jeho uživatelském účtu, kde uvidí i stav jeho adopce.

U virtuální adopce, po kliknutí na tlačítko „Adoptovat“, uživateli vyskočí okno s vygenerovaným QR kódem. Po zavření okna s QR kódem lze kód opět otevřít, a to nově přidaným tlačítkem „Zobrazit QR“. Tlačítku „Adoptovat“ se změní text na „Čeká na zaplacení“. Uživateli se virtuální kočka přidá do seznamu virtuálních adopcí v jeho uživatelském účtu. V detailu každé kočky, pokud uživatel ještě neprovedl platbu, se bude zobrazovat tlačítko k zobrazení QR kódu. Zde také uživatel bude moci sledovat stav vyřízení jeho adopce.

6.5 Případy užití

Jediným aktérem v systému bude uživatel. V diagramu, viz Obrázek 13, jsou uvedeny všechny případy užití, které systém uživateli umožňuje. Jedná se především o registraci, přihlášení, zobrazení seznamů, zobrazení jednotlivých karet, možnost adopce a následné odhlášení.

K vytvoření diagramu případů užití jsem využila online nástroj Visual Paradigm.[111]

Visual Paradigm Online Free Edition



Visual Paradigm Online Free Edition

Obrázek 13 - Ukázka případu užití (návrh)

6.6 Wireframes

K vytvoření wireframů byl použit grafický editor Figma.[112] Byly vybrány 3 charakteristické barvy aplikace, a to především růžová, šedá a bílá.

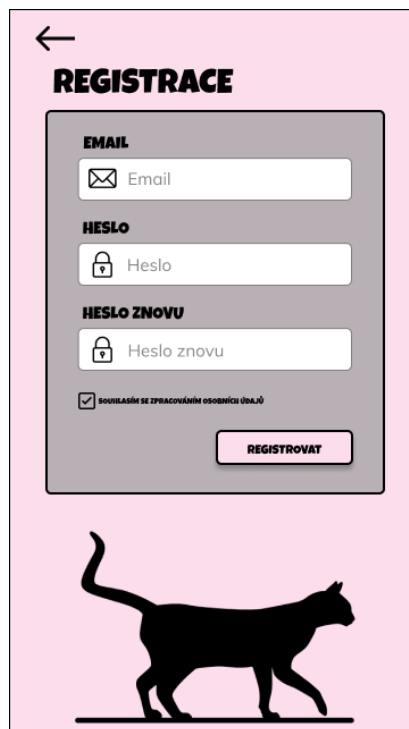
6.6.1 Úvodní obrazovka

Úvodní obrazovka, která se uživateli zobrazí po spuštění aplikace, představuje základní menu. Zde se nachází 3 tlačítka, a to pro přihlášení, registraci a pro možné pokračování bez přihlášení, viz Obrázek 14.



Obrázek 14 - Úvodní obrazovka aplikace (návrh)

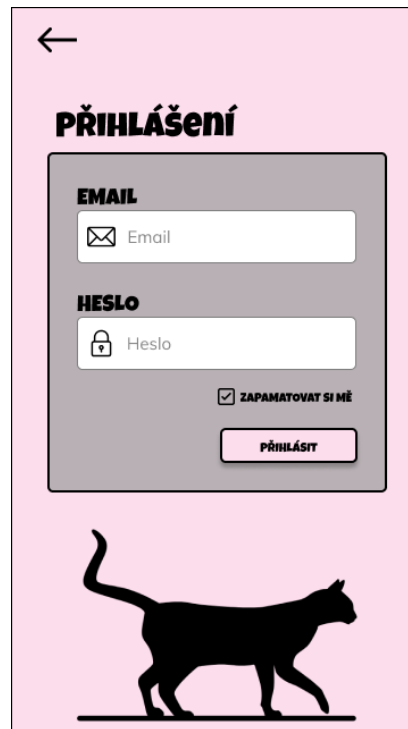
U registrace, viz Obrázek 15, se po uživateli vyžaduje pouze email a zadání 2x hesla pro kontrolu. Jednoduchá registrace je z důvodu, aby uživatel nebyl odrazen, že zadává spoustu osobních údajů hned na začátku. Zde bude i checkbox pro zpracování osobních údajů, který je uživatel povinen zaškrtnout. Pokud se tak nestane, registrace se neprovede. Dokument ke zpracování osobních údajů bude vyřešen až při nasazení aplikace, nyní se jedná pouze o vytvoření funkční části aplikace.



The image shows a registration form design on a light pink background. At the top left, there is a back arrow icon. Below it, the word "REGISTRACE" is written in bold black letters. The form itself is a grey rounded rectangle containing several fields: "EMAIL" with an envelope icon and a text input field labeled "Email"; "HESLO" with a lock icon and a text input field labeled "Heslo"; "HESLO ZNOVU" with a lock icon and a text input field labeled "Heslo znovu". Below these fields is a checkbox labeled "SOULASÍM SE ZPRACOVÁNÍM OSOBNÍMI ÚDAJI" which is checked. At the bottom right of the form is a pink button labeled "REGISTROVAT". Below the form is a black silhouette of a cat walking to the right.

Obrázek 15 - Registrace (návrh)

Hned po registraci je uživatel schopen se přihlásit do aplikace a využívat aplikaci v plném rozsahu, viz Obrázek 16. Zadává se tu email a heslo, které bylo vyžadováno u registrace. Bude tu i checkbox pro zapamatování uživatele, aby se nemusel pokaždé, co aplikaci zapne, přihlašovat znovu.



Obrázek 16 - Přihlášení (návrh)

Po přihlášení, aplikace přesměruje uživatele na stránku se seznamem kaváren, viz Obrázek 17. Na tento seznam se dostane i nepřihlášený uživatel, který využije tlačítka „Pokračovat bez přihlášení“. Uživatel je již na úvodní stránce obeznámen s tím, že mu některé funkce nemusí být přístupné, např. adopce. Na rozdíl od přihlášeného uživatele, se nepřihlášenému neukazuje vpravo nahoře ikona profilu. Přihlášenému se naopak nebude vlevo nahoře zobrazovat šipka pro přesměrování na úvodní obrazovku aplikace. Uživateli, který zaškrtnul souhlas se zapamatováním uživatele, se tato stránka stane úvodní obrazovkou. To znamená, že pokaždé, co uživatel zapne aplikaci, zobrazí se mu rovnou seznam kaváren.



Obrázek 17 - Seznam adopčních kaváren (návrh)

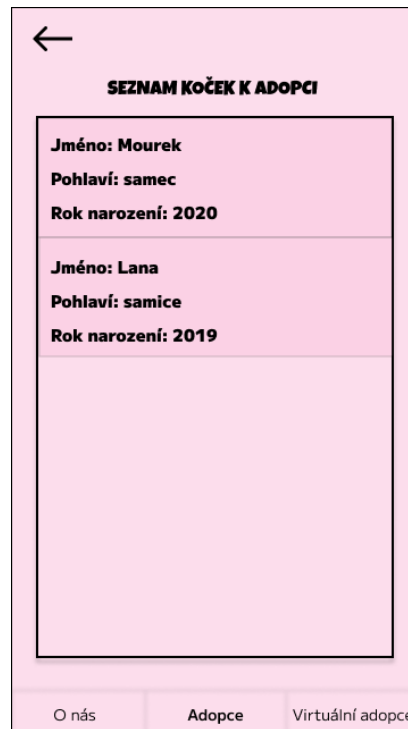
6.6.2 Stránka kavárny

Při výběru konkrétní kavárny se zobrazí seznam karet – TabbedPage. Každá karta je zaměřena na určitou část – O nás, adopce a virtuální adopce. V kartě „O nás“ budou informace o kavárně jako představení kavárny, kontakt a adresa, viz Obrázek 18.



Obrázek 18 - Stránka kavárny – O nás (návrh)

Další dvě karty budou představovat seznamy koček a virtuálních koček k adopci. U každé kočky v seznamu budou pouze základní informace – jméno, pohlaví, rok narození. Pro více informací je nutné na konkrétní kočku kliknout pro zobrazení jejího detailu. Obrázek 19 představuje ukázkou karty „Adopce“. Karta „Virtuální adopce“ bude vypadat dost podobně.



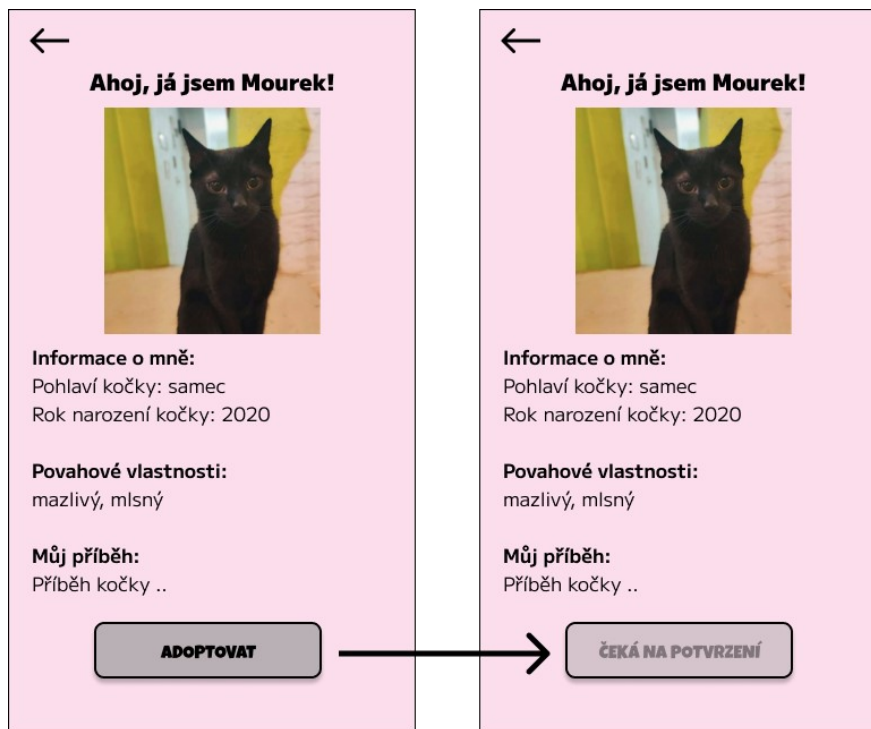
Obrázek 19 - Stránka kavárny – Adopce (návrh)

Při kliknutí na konkrétní kočku se zobrazí její detail. Zde jsou další informace o kočce, její fotka, povahové vlastnosti a příběh. Nachází se zde i tlačítko „Adoptovat“, které je aktivní pouze přihlášenému uživateli, viz Obrázek 20.



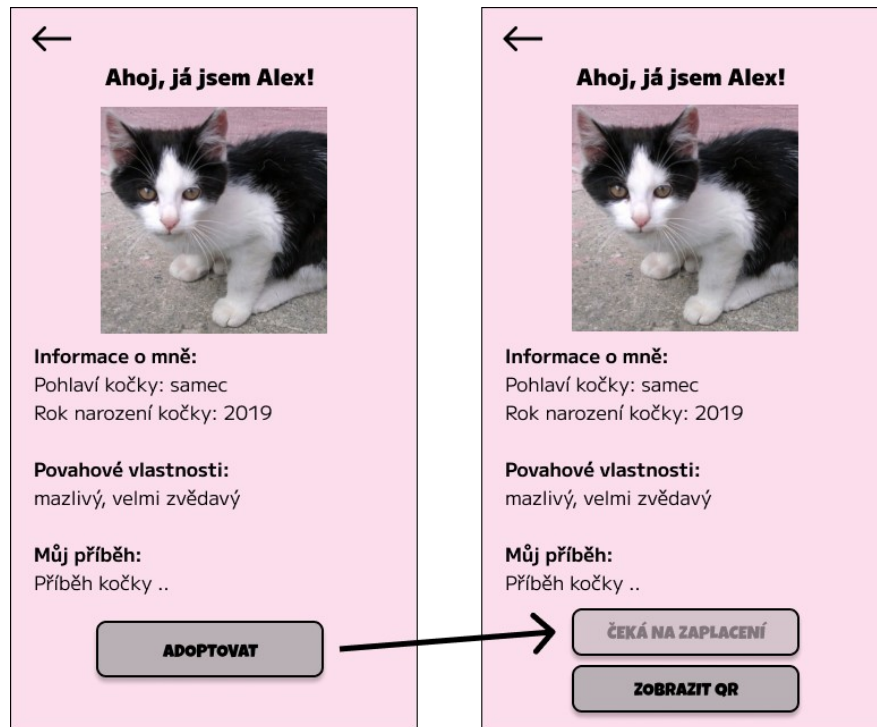
Obrázek 20 - Detail kočky – Rozdíl mezi přihlášeným a nepřihlášeným uživatelem (návrh)

Detail kočky je stejný jak pro klasickou, tak i pro virtuální kočku. Liší se pouze tím, co se stane po kliknutí na tlačítko „Adoptovat“. Pokud se jedná o kočku, kterou lze adoptovat fyzicky - tlačítko se zneaktivní a změní se text na „Čeká na schválení“, viz Obrázek 21.



Obrázek 21 - Detail kočky – Klasická adopce (návrh)

Naopak u virtuální kočky je to více komplexní. Po kliknutí na tlačítko se zobrazí QR kód pro zaplacení. Tlačítko se zneaktivní a změní se text na „Čeká na zaplacení“. Navíc se přidá tlačítko pro opětovné zobrazení QR kódu, viz Obrázek 22.



Obrázek 22 - Detail kočky – Virtuální adopce (návrh)

6.6.3 Stránka uživatele

Uživatelský profil využívá také seznam karet – TabbedPage. V tomto případě se bude jednat o karty – Uživatelský profil, moje adopce a moje virtuální adopce. Uživatelský profil bude obsahovat pouze email, kterým se uživatel přihlašuje a tlačítko pro odhlášení, viz Obrázek 23. Po odhlášení je uživatel přesměrován na úvodní obrazovku aplikace.



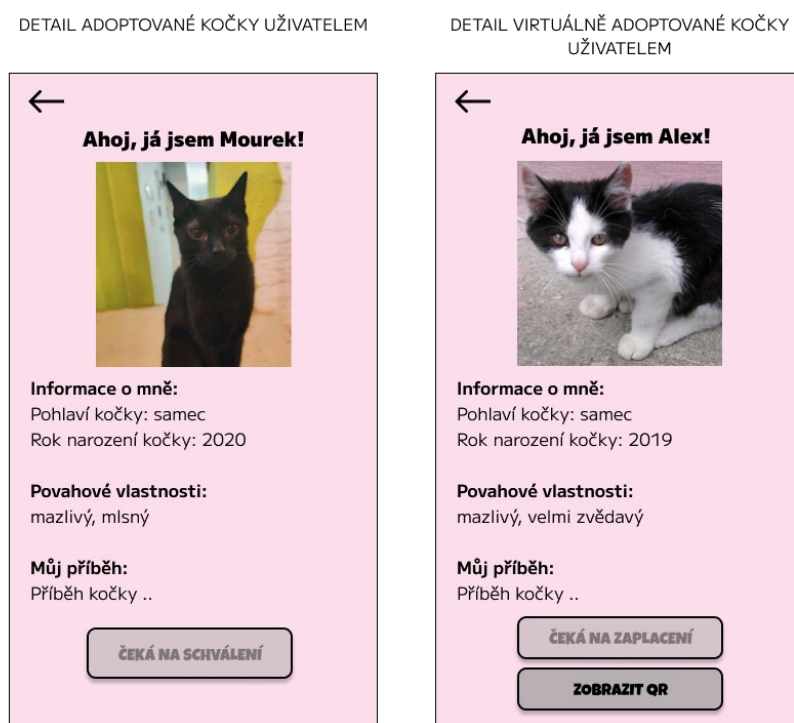
Obrázek 23 - Stránka uživatele – Uživatelský profil (návrh)

Na dalších dvou kartách se nachází seznamy klasických a virtuálních koček, které si uživatel adoptoval. Obrázek 24 představuje ukázkou seznamu virtuálních koček uživatele.



Obrázek 24 - Stránka uživatele – Moje virtuální adopce (návrh)

Po kliknutí na konkrétní kočku se uživatel dostane do detailu kočky. Zde může sledovat stav jeho vyřízení adopce. Detail kočky je dost podobný jako po provedení adopce kočky u konkrétní kavárny. U virtuální kočky zůstává stejný jak po provedení adopce, to znamená, že tlačítko nebude aktivní, zobrazuje se text „Čeká na zaplacení“ a je tu tlačítko pro zobrazení QR kódu. Na rozdíl od virtuální kočky, se u klasické kočky detail mění – tlačítko zůstává nadále neaktivní, mění se pouze textace z „Čeká na potvrzení“ na „Čeká na schválení“, viz Obrázek 25 .



Obrázek 25 - Stránka uživatele – Detail adoptované a virtuálně adoptované kočky (návrh)

6.7 Návrh databáze

Databáze se bude skládat z 11 tabulek, které jsou vytvořené pomocí migrace, jedná se o 3 tabulky navržené mnou, o 7 tabulek, které generuje framework Identity a jedné tabulky sloužící pro sledování verze migrace.

První 3 navržené tabulky jsou tabulky kavarny, kocky a kocky_uzivatele.

Tabulka kavarny obsahuje sloupce – Kavarna_id, Nazev, Informace, Email, Telefon, AdresaMesto, AdresaUlice, BIC a IBAN. Adresa bude rozdělena z důvodu, aby bylo možné použít buď pouhou ulici kavárny či pouze její město, ve kterém se nachází. Sloupce BIC a IBAN se využijí při generování QR kódu u virtuální adopce.

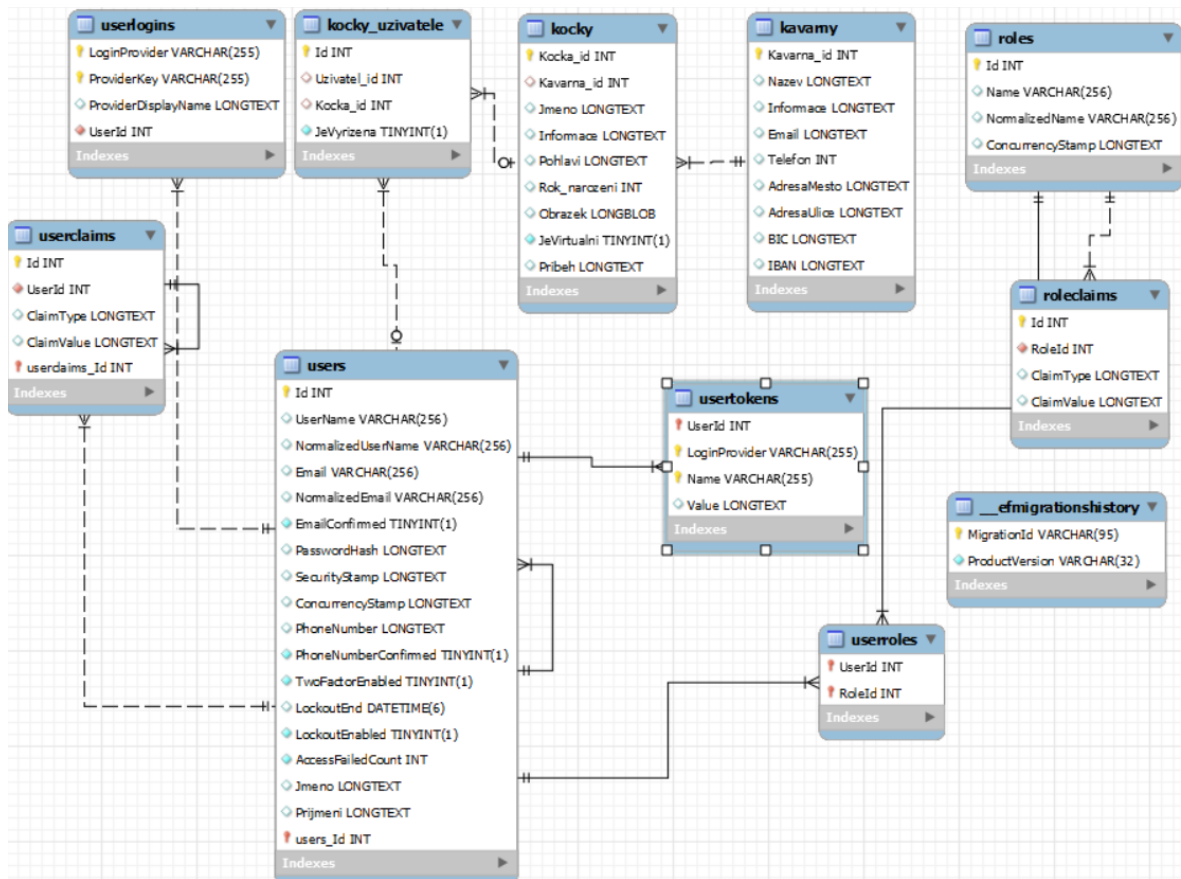
Tabulka kocky se skládá ze sloupců Kocka_id, Kavarna_id, Jmeno, Informace, Pohlavi, Rok_narozeni, Obrazek, JeVirtualni a Pribeh. Kavarna_id je cizí klíč, který propojuje tabulku kocky s tabulkou kavarny. Je to z důvodu, aby každá kočka měla přidělenou kavárnu, do které patří. Sloupec JeVirtualni nabírá pouze hodnot true nebo false, využívá se datového typu boolean.

Poslední tabulkou je tabulka kocky_uzivatele. Tato tabulka bude sloužit pro ukládání adoptovaných koček konkrétního uživatele. Bude obsahovat sloupce Id, Uzivatel_id, Kocka_id a JeVyrizena. Uzivatel_id a Kocka_id jsou opět cizí klíče, které vytváří vztah mezi tabulkami. Sloupec JeVyrizena bude představovat datový typ boolean – hodnoty true nebo false.

Tabulky vytvořené pomocí frameworku Identity jsou tabulky roleclaims, roles, userclaims, userlogins, userroles, users a usertokens. Z těchto tabulek se bude využívat pouze jedna, a to tabulka users. Ostatní tabulky prozatím nebudou využity, ale jsou nutné pro chod Identity.

Tabulka users obsahuje sloupce Id, UserName, NormalizedUserName, Email, NormalizedEmail, EmailConfirmed, PasswordHash, SecurityStamp, ConcurrencyStamp, PhoneNumber, PhoneNumberConfirmed, TwoFactorEnabled, LockoutEnd, LockoutEnabled, AccessFailedCount, Jmeno, Prijmeni. Přímou budou využívány 3 sloupce, a to UserName, Email a PasswordHash. Sloupce Jmeno a Prijmeni se budou využívat až v budoucnu. Pro zachování bezpečnosti před zneužitím útočníka je heslo uloženo ve formě hashe.

K vytvoření databáze bude používán systém MySql, který slouží pro tvorbu relačních databázových modelů. Obrázek 26 představuje schéma databáze.



Obrázek 26 - Schéma databáze

7 VÝVOJ APLIKACE

Cílem aplikace pro adopci koček, byla možnost adoptovat si kočku z adopční kočičí kavárny pomocí svého chytrého mobilního zařízení s operačním systémem Android (do budoucna rozšíření na iOS).

Pro zrealizování tohoto cíle, byla vytvořena databáze v MySQL, která bude obsahovat požadované údaje o kočkách, kavárnách, uživateli a propojeních mezi nimi. Samotnou databázi jsme si již popsali v návrhu aplikace. Data z databáze zpracovává web API, se kterým následně komunikuje aplikace z mobilního zařízení. Tyto dvě části jsou blíže popsány v následujících kapitolách.

7.1 REST API

REST API zprostředkovává přístup do databáze naší aplikace. Nespornou výhodou použití API je, že v budoucnu můžeme nahradit mobilní aplikaci jinou aplikací. Také bude možné vytvořit další mobilní či jiný druh aplikace, která toto API může využívat, a to z důvodu přístupu ke stejné databázi koček. Další výhodou je lepší kontrola nad přístupem aplikace do databáze, a to z důvodu že klienti budou přistupovat k datům výhradně skrze API službu. Jelikož je použit pro vývoj mobilní aplikace programovací jazyk C#, byl tento jazyk zvolen i pro API službu.

API přináší možnost implementace různých funkcionalit, jako je například generování QR kódu k platbě. Tato funkcionalita by mohla být implementována v mobilní aplikaci, ale implementací v API se rozšiřuje o možnost využití frameworků, které pro mobilní aplikace nemusí být dostupné. Zároveň pokud QR platbu vygeneruje API, tak je případně možné využít tuto platbu i v jiných aplikacích, které by se připojily na API. To všechno se provede bez potřeby implementovat kód znovu. To je docílené tím, že z mobilní aplikace putuje http dotaz na konkrétní koncové body v API rozhraní, ze kterých následně dostává odpovědi.

Do databáze přistupuje pouze API rozhraní, a ne samotná aplikace. Tímto způsobem je možné kontrolovat autorizaci uživatele a povolit volání pouze na chtěné činnosti. To v praxi znamená, že nepřihlášený uživatel dostane přístup k údajům ohledně koček a kaváren a přihlášený dostane navíc přístup k údajům o sobě samém a svých adoptovaných kočkách. Zároveň je mu umožněna adopce koček, a to za pomoci insertu záznamu do databáze.

7.1.1 Použité frameworky

Pro zrychlení vývoje vlastní API služby byly použity tyto frameworky:

- **Entity Framework** je využíván ke snadnějšímu přístupu k MySQL databázi. Tento framework obsahuje metody pro obsluhu databáze. To zahrnuje například základní operace nad databází: vložit, zobrazit, upravit a vymazat záznam z tabulky. V tomto případě bylo využito přístupu „Code First“, to znamená, že byly nejprve vytvořeny datové modely v samostatné knihovně C# projektu a následně pomocí migrace Entity Frameworku byla vytvořena potřebná databáze.
- **QRCoder** slouží k tvorbě QR kódů, které jsou používány v aplikaci pro snadné zaplacení virtuální adopce. QRCoder generuje obrázek ve formátu PNG, který vyobrazuje požadovaný QR kód určený k platbě. Tento QR kód obsahuje platební údaje v podobě textového řetězce, které jsou formátovány ve standardu ČBA.

7.1.2 Endpointy

Endpointy neboli koncové body, se nachází v API. Jsou na místech, většinou v controllerech, na které se lze dotazovat z vnějšku aplikace.

Samotné dotazování může obsahovat požadavek na získání dat nebo na provedení nějaké akce. Krásným příkladem je přihlášení do aplikace. API služba obsahuje několik takových endpointů. V projektu jsme členili endpointy do oblastí, podle toho, jakou činnost s daty vykonávají. Těmito oblastmi se rozumí controllery.

7.1.3 Controllery

V API jsou vytvořené 4 základní controllery – AccountController, CatCafeController, CatController a CatUserController. Každý z controllerů obsahuje endpointy směřující ke konkrétní oblasti.

AccountController obsahuje endpointy sloužící k obsluze práce s uživatelským účtem. Zde se řeší především registrace, přihlášení a následné odhlášení uživatele.

CatCafeController obsahuje endpointy sloužící k obsluze práce s kavárnami v databázi. V tomto případě především k navrácení údajů jedné kavárny nebo seznamu všech kaváren v databázi.

CatController obsahuje endpointy sloužící k obsluze práce s kočkami v databázi. Stejně jako v předchozím případě, jen s rozdílem objektu. Základními endpointy zde jsou endpointy pro

vrácení jedné kočky nebo seznamu všech koček. Dále obsahuje endpoint pro vrácení seznamu koček podle ID kavárny, ve které se nachází. Poslední endpoint slouží k přidání nové kočky do databáze.

CatUserController obsahuje endpointy sloužící k obsluze práce s kočkami a jejich propojení s uživatelem. Najdeme zde vrácení záznamu jedné adopce a také všech adopcí k jednomu uživateli, dle jeho ID. V tomto controlleru se nachází i endpoint pro vytvoření propojení mezi uživatelem a kočkou, jedná se o samotnou adopci. Další endpoint umožňuje autorizovanému uživateli se doptat na své údaje – jméno, e-mail apod. Poslední endpoint v controlleru je zaměřen na vrácení platebních údajů k virtuální adopci.

7.2 Mobilní aplikace v Xamarin.Forms

Hlavní část bakalářské práce je mobilní aplikace napsaná v jazyce C# za pomoci Xamarin.Forms frameworku. Tento framework je využitý k překladu kódu napsaného v jazyce C# a XAML do nativního jazyka dané platformy – Android a iOS. Aplikace používá dva přístupy k datům o kočkách. Jeden je za pomoci autorizovaného spojení ověřeného registrací a následným přihlášením uživatele a druhý bez nutnosti přihlášení uživatele. V tomto případě jsou omezeny některé funkcionality aplikace, a to zejména možnost adopce. Aplikace se skládá z modelu dat, viewmodelů a views. Model dat je zvolen jako samostatná DLL knihovna z důvodu využití jak v mobilní aplikaci, tak v API službě. Model dat obsahuje popis objektů používaných v aplikaci. To jsou například, objekt kočky, kavárny, uživatele atd. Viewmodely popisují logiku, která se děje na pozadí views a odkazují na další třídy v programu, které se starají například o připojení k API a práci s přihlašovací tokenem. Views zobrazují frontend aplikace se kterým může interagovat uživatel.

Aplikace byla testována v emulátoru Pixel 2 Pie 9.0 - API 28 (Android 9.0 - API 28).

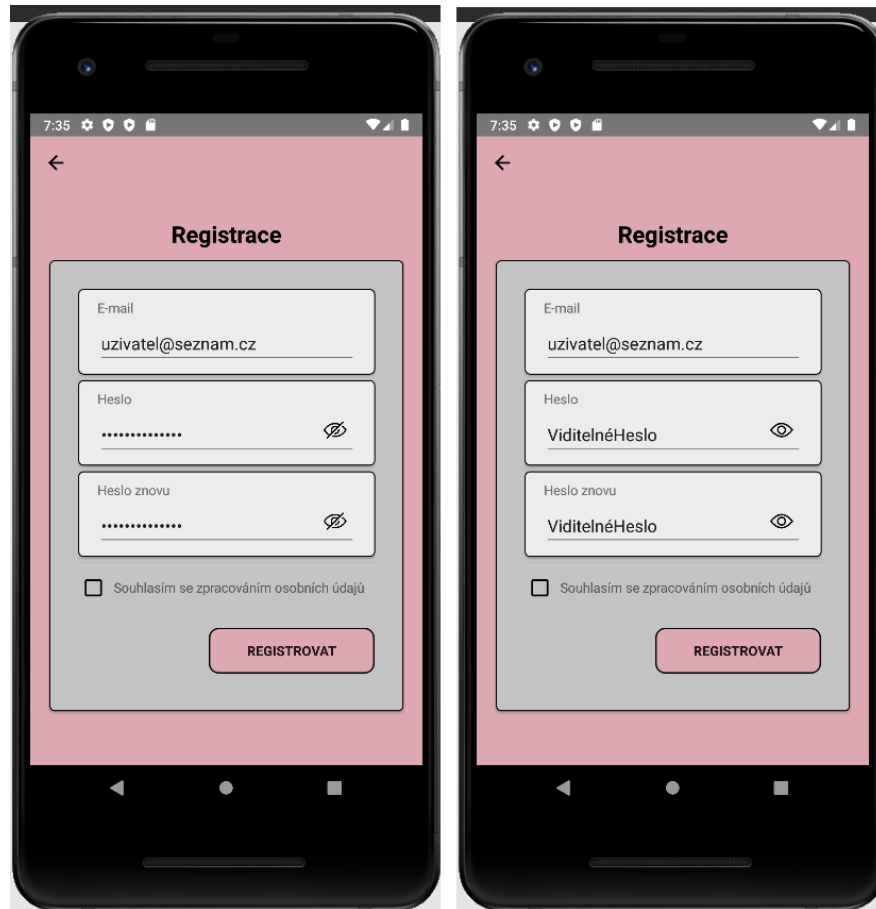
Úvodní stránka aplikace, viz Obrázek 27, obsahuje 3 tlačítka – pro přihlášení, registraci anebo pro možnost pokračovat bez přihlášení. Po kliknutí na jakékoliv tlačítko, aplikace přesměruje uživatele na konkrétní stránku.



Obrázek 27 - Úvodní strana aplikace (aplikace)

7.2.1 Registrace

Po kliknutí na tlačítko „Registrovat“ je uživatel přesměrován na stránku s 3 inputy, jedním checkboxem a tlačítkem pro registraci. Zde uživatel zadává email a heslo 2x po sobě, z důvodu ověření shody obou hesel. Uživatel má automaticky hesla skrytá, ale pomocí tlačítek pro zobrazení/skrytí hesla je schopen si heslo zobrazit či skrýt, viz Obrázek 28.



Obrázek 28 - Registrace – možnost zobrazení/skrytí hesla (aplikace)

Aby byla možná registrace, je uživatel povinen zaškrtnout checkbox pro souhlas s osobními údaji. Pokud se tak nestane, je na to uživatel upozorněn. Při kliknutí na tlačítko „Registrovat“ je pár podmínek, které musí být dodrženy k validní registraci. Jedná se o povinnost mít zaškrtnutý checkbox, mít obě hesla identická, povinnost zadat email, který ještě není registrován a zadání hesla o minimální délce 6 znaků. Toto vše se kontroluje, než je registrace povolena. Na Obrázek 29 jsou vidět podmínky a následné upozornění uživatele pomocí dialogového okna, které se ukáže jen v momentě, kdy uživatel něco nedodržel.

```
private async void Button_Clicked(object sender, EventArgs e)
{
    if(ViewModel.CheckboxSouhlas == true)
    {
        if(ViewModel.Password == ViewModel.ConfirmPassword)
        {
            HttpResponseMessage response = await CallAPI.UseAPI("/Account/Register", ViewModel);

            if (response == null)
            {
                return; // v budoucnu řešit zobrazení chyby
            }

            if (response.IsSuccessStatusCode)
            {
                Debug.WriteLine(@"Registration was successfull");

                LoginViewModelX LViewModel = new LoginViewModelX();

                LViewModel.Username = ViewModel.Email;
                LViewModel.Password = ViewModel.Password;
                LViewModel.CheckboxZapamatovat = false;
                LViewModel.IsPassword = true;

                await LoginViewModelX.CallAPIAndLogin(LViewModel, this);
            }

            else
            {
                string responseJsonString = await response.Content.ReadAsStringAsync();

                if (responseJsonString.Contains("DuplicateUserName:"))
                {
                    await DisplayAlert("Chyba", "Email je již registrován!", "OK");
                }
                else if(responseJsonString.Contains("The Password must be at least 6 characters long.))")
                {
                    await DisplayAlert("Chyba", "Heslo musí obsahovat nejméně 6 znaků.", "OK");
                }
                else
                {
                    Debug.WriteLine(response.StatusCode + ": " + response.ReasonPhrase);
                    await DisplayAlert("Ups", "Něco se pokazilo, zkuste to znovu.", "OK");
                }
            }
        }

        else
        {
            Debug.WriteLine("Different passwords");
            await DisplayAlert("Chyba", "Hesla se musí shodovat!", "OK");
        }
    }

    else
    {
        Debug.WriteLine("Unchecked checkbox");
        await DisplayAlert("Chyba", "Není zaškrtnut souhlas s osobními údaji", "OK");
    }
}
```

Obrázek 29 - Podmínky nutné k registraci (kód)

Pro jakoukoliv další chybu, která se může vyskytnout, je vytvořena univerzální hláška „Něco se pokazilo, zkuste to znovu“. Do budoucna se tyto podmínky rozrostou a budou zachycovat více možných chyb, které mohou při registraci nastat.

Po úspěšné registraci se uživatel okamžitě přihlásí a je přeměrován na stránku s kočíčky kavárnami.

7.2.2 Přihlášení

Po kliknutí na tlačítko „Přihlásit se“ je uživatel přesměrován na stránku s 2 inputy, jedním checkboxem a tlačítkem pro přihlášení, viz Obrázek 30. Zde má uživatel také možnost zobrazení hesla při psaní.



Obrázek 30 - Přihlášení (aplikace)

Aby bylo možné se přihlásit, musí být již uživatel v aplikaci registrován. Pro přihlášení se zadává validní email a heslo. Jestliže je jeden z inputů zadán nevalidně, objeví se uživateli dialogové okno s upozorněním, že bylo zadáno špatně heslo nebo email.

Checkbox, který se zde nachází, slouží k tomu, aby uživatel mohl zůstat přihlášen. Po úspěšném přihlášení je uživatel přesměrován na stránku s kočičími kavárnami a je mu vygenerován token, který umožňuje právě využívat veškeré funkcionality aplikace, jako např. adopci či nahlédnout do uživatelova profilu. Tento token má omezený čas a po jeho vypršení se uživatel automaticky odhlásí. Toto nastane pouze v případě, pokud checkbox není zaškrtnutý. Jestliže uživatel využije možnosti zůstat přihlášen, přihlašovací údaje se

uloží do SecureStorage a při každém zapnutí aplikace je uživatel automaticky přihlášen a objeví se na stránce s kočíčky kavárnami.

Na Obrázek 31 je ukázka metody, která se vykoná po kliknutí na tlačítko „Přihlásit se“, které se nachází na stránce Přihlášení.

```
public async static Task CallAPIAndLogin(LoginViewModelX ViewModel, Page page)
{
    HttpResponseMessage response = await CallAPI.UseAPI("/Account/Login", ViewModel);

    if (response == null)
    {
        return; // v budoucnu řešit zobrazení chyby
    }

    var responseJsonString = await response.Content.ReadAsStringAsync();

    if (response.IsSuccessStatusCode)
    {
        UserToken.Token = JsonConvert.DeserializeObject<Token>(responseJsonString);
        UserToken.NastavTimer();
        Debug.WriteLine(UserToken.Token.userID);
        Debug.WriteLine(@"Login was successfull");

        if (ViewModel.CheckboxZapamatovat == true)
        {
            await SecureStorage.SetAsync(LoginViewModelX.SecureKeyEmail, ViewModel.Username);
            await SecureStorage.SetAsync(LoginViewModelX.SecureKeyPassword, ViewModel.Password);
        }

        else
        {
            SecureStorage.Remove(LoginViewModelX.SecureKeyEmail);
            SecureStorage.Remove(LoginViewModelX.SecureKeyPassword);
        }

        await SecureStorage.SetAsync(LoginViewModelX.SecureKeyToken, responseJsonString);

        if (page != null)
        {
            await page.Navigation.PushAsync(new BezPrihlaseniPage());
        }

    }

    else
    {
        if (page != null)
        {
            //await page.DisplayAlert("Error", responseJsonString, "ok");

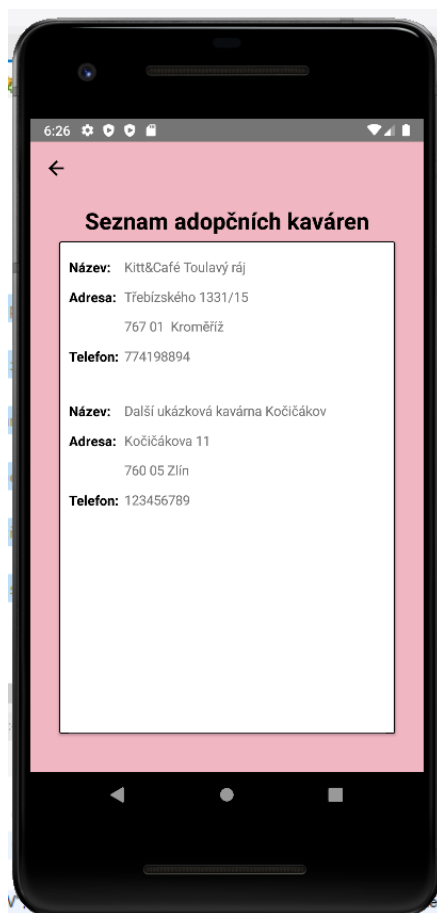
            await page.DisplayAlert("Chyba", "Špatně zadaný email nebo heslo.", "OK");
        }

        Debug.WriteLine(response.StatusCode + ": " + response.ReasonPhrase);
    }
}
```

Obrázek 31 - Metoda sloužící k přihlášení (kód)

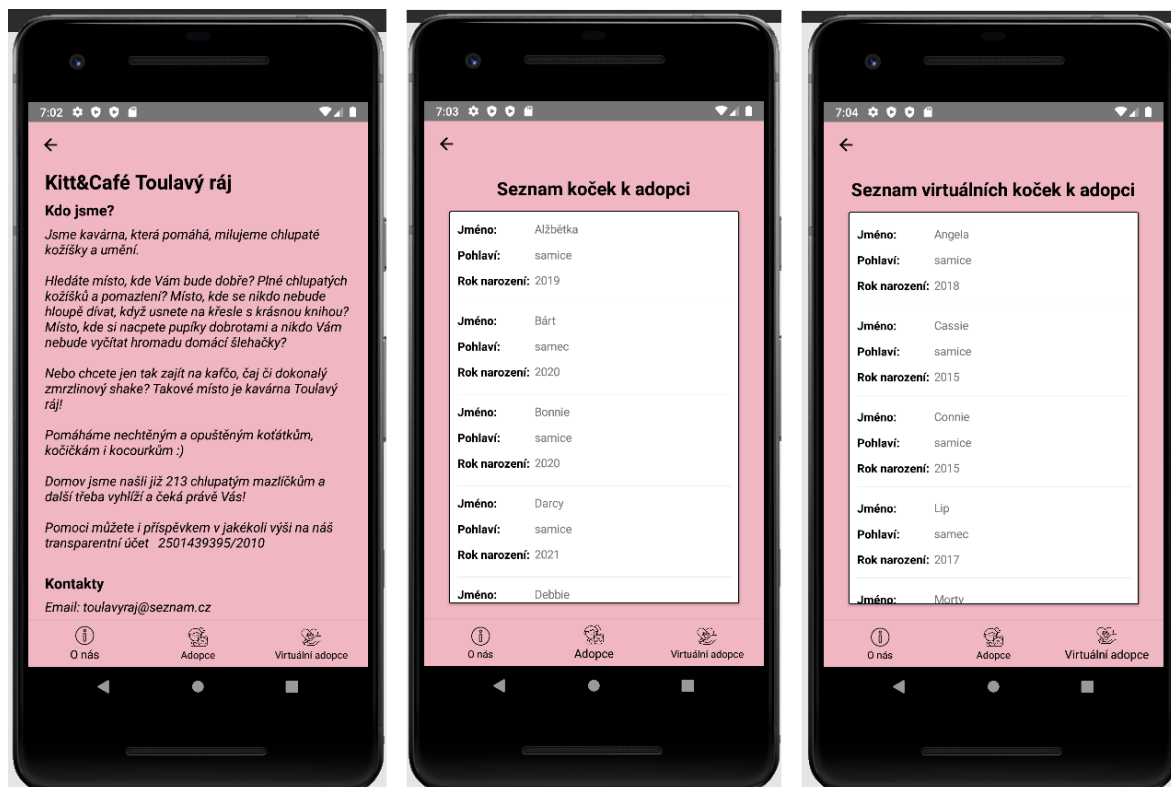
7.2.3 Nepřihlášený uživatel

Uživatel má možnost využívat aplikaci bez předchozího přihlášení, ale jsou mu některé funkce omezeny. Po kliknutí na tlačítko „Pokračovat bez přihlášení“ je uživatel přesměrován na stránku s kočíčky kavárnami, viz Obrázek 32.



Obrázek 32 - Seznam adopčních kaváren – nepřihlášený uživatel (aplikace)

Zde si uživatel může konkrétní kavárnu vybrat a kliknout na ní. Zobrazí se mu seznam karet (TabbedPage), kde každá karta představuje určitou oblast – informace o kavárně, seznam koček k adopci a seznam virtuálních koček k adopci, viz Obrázek 33.



Obrázek 33 - Stránka kavárny (aplikace)

Aby bylo možné posouvání obsahu v kartě „O nás“, je využito prvku ScrollView. K posouvání obsahu v seznamu se využívá ovládacího prvku ListView, který je speciálně navržený pro seznamy se stejnými položkami.

U každého seznamu lze na konkrétní kočku kliknout a zobrazit si její detail. Detail kočky je nová stránka, kterou si může zobrazit jak přihlášený, tak nepřihlášený uživatel. Obsahuje fotografii, informace, povahové vlastnosti kočky, její příběh a tlačítko „Adoptovat“.

Detail adopce u nepřihlášeného uživatele se liší pouze tím, že uživateli není umožněno kliknout na tlačítko „Adoptovat“. Toto tlačítko je uživateli viditelné, ale neaktivní. Je to z důvodu, aby uživatel věděl, že po registraci a následném přihlášení mu tato funkce bude umožněna. Zde je také možnost posouvání obsahu, tlačítko „Adoptovat“ je vždy na konci, viz Obrázek 34.

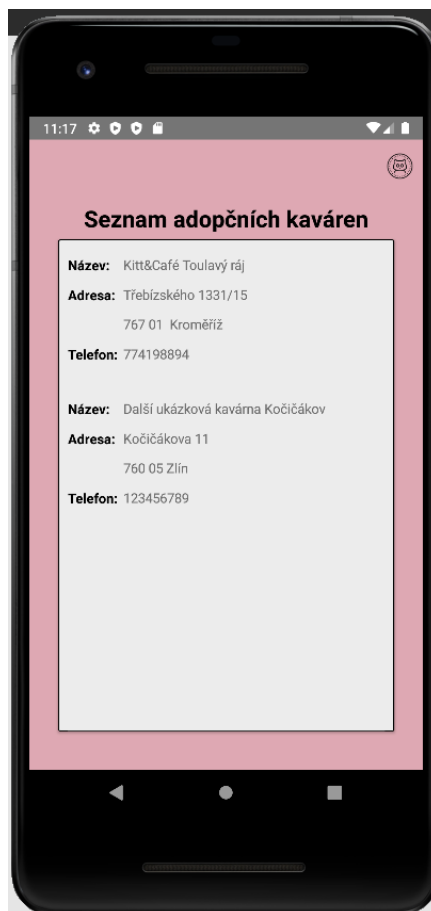


Obrázek 34 - Detail kočky – nepřihlášený uživatel (aplikace)

7.2.4 Přihlášený uživatel

Jestliže je uživatel přihlášen, může využívat všechny funkce, které aplikace nabízí. Jedná se o možnost adopce, zobrazení uživatelského profilu a zobrazení seznamu koček, které uživatel adoptoval či chce adoptovat.

Přihlášenému uživateli se úvodní stránkou nastavuje stránka s kočičími kavárnami, viz Obrázek 35. Zde má uživatel také možnost se dostat do svého uživatelského profilu. Nepřihlášenému uživateli je ikona profilu skryta.



Obrázek 35 - Seznam adopčních kaváren – přihlášený uživatel (aplikace)

Při výběru konkrétní kavárny a kliknutí na ní se uživatel přesměruje na stránku kavárny, která je rozdělená do 3 karet (TabbedPage). Stránka kavárny je stejná jak u nepřihlášeného uživatele.

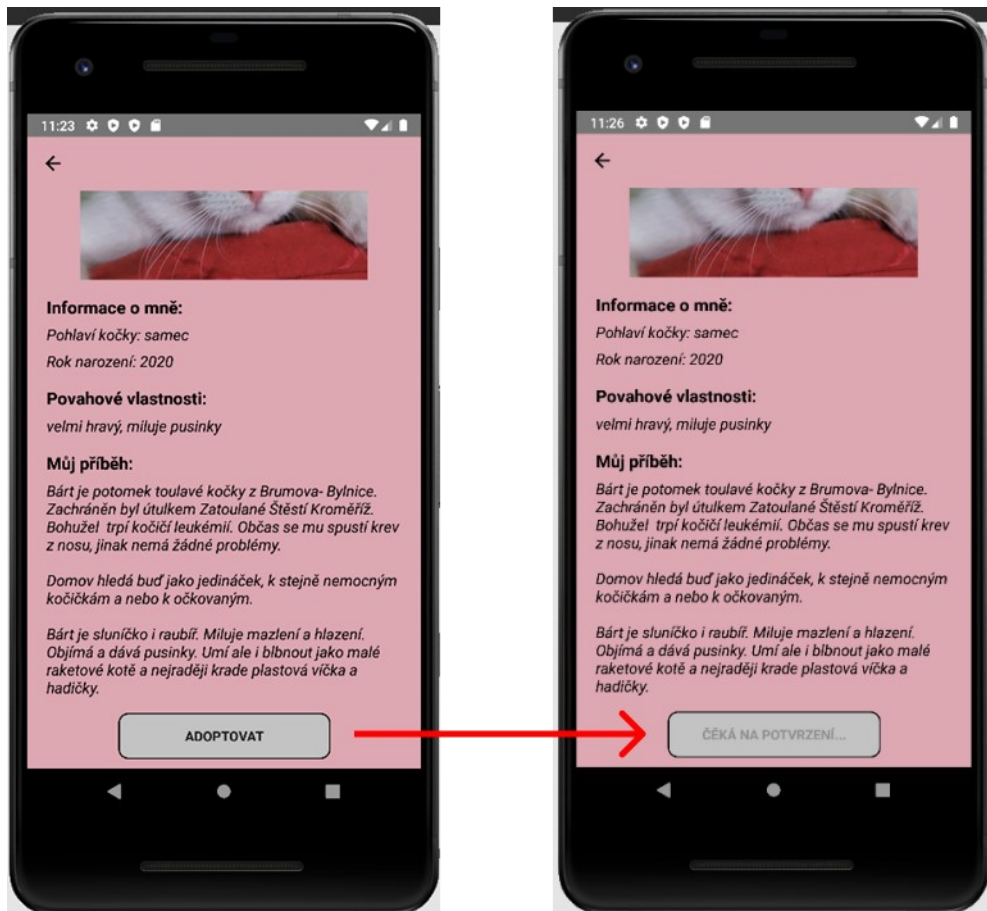
Při kliknutí na kočku z konkrétního seznamu se zobrazí její detail. Zde již je umožněna přihlášenému uživateli adopce, viz Obrázek 36.



Obrázek 36 - Detail kočky – přihlášený uživatel (aplikace)

Detail kočky je stejný jak pro klasickou, tak pro virtuální kočku do doby, než se klikne na tlačítko „Adoptovat“.

Po kliknutí na tlačítko se tlačítko zneaktivní a text se změní podle toho, o jakou kočku se jedná. Pokud se jedná o kočku, která je k fyzické adopci, text se změní na „Čeká na potvrzení“ a dále se nic neděje, viz Obrázek 37.



Obrázek 37 - Detail kočky – Klasická adopce (aplikace)

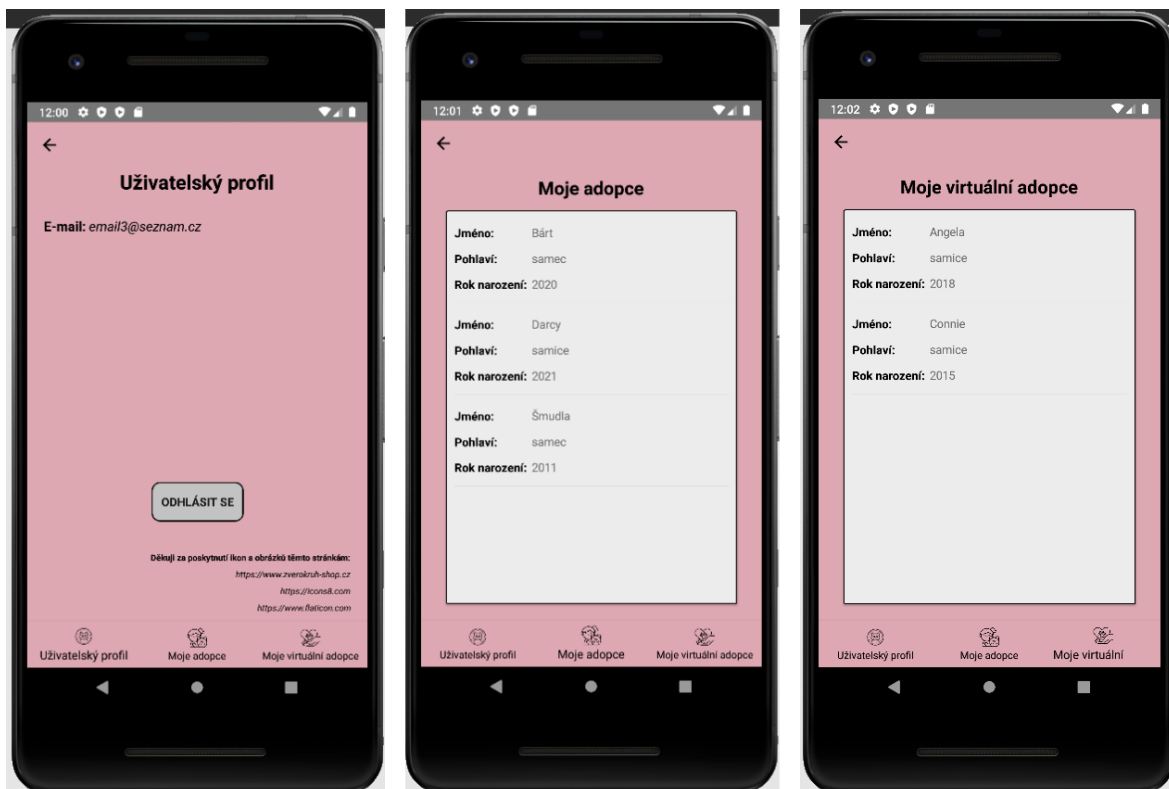
Ale jestliže se jedná o kočku k virtuální adopci, po kliknutí na tlačítko „Adoptovat“ se otevře dialogové okno s vygenerovaným QR kódem. U detailu virtuálně adoptované kočky se text změnil na „Čeká na zaplacení“ a přibude zde nové tlačítko, které slouží pro opětovné zobrazení QR kódu. Celý tento proces zobrazuje Obrázek 38.



Obrázek 38 - Detail kočky – Virtuální adopce (aplikace)

7.2.5 Stránka uživatele

Zobrazit si uživatelský profil může pouze přihlášený uživatel. Na jeho úvodní stránce, tedy s kočičími kavárnami, je ikona profilu, na kterou je nutné kliknout. Po kliknutí se uživateli zobrazí seznam karet (TabbedPage). Jedná se o 3 karty, kde se jedna věnuje uživatelskému profilu a další dvě obsahují seznamy uskutečněných adopcí uživatele, viz Obrázek 39.



Obrázek 39 - Stránka uživatele (aplikace)

Na kartě „Uživatelský profil“ najdeme nyní pouze uživatelský e-mail a tlačítko pro odhlášení. Také se zde nachází mimořádné poděkování stránkám, ze kterých jsem použila ikony či obrázky v aplikaci. V budoucnu zde bude možnost přidání jména a příjmení a také umožněna změna e-mailu.

Další dvě karty jsou obdobné jako u stránky kavárny se seznamy koček. Zde se ale uživateli ukládají kočky, které chce adoptovat nebo již adoptoval. Při výběru konkrétní kočky může uživatel sledovat v detailu stav vyřízení jeho adoptace.

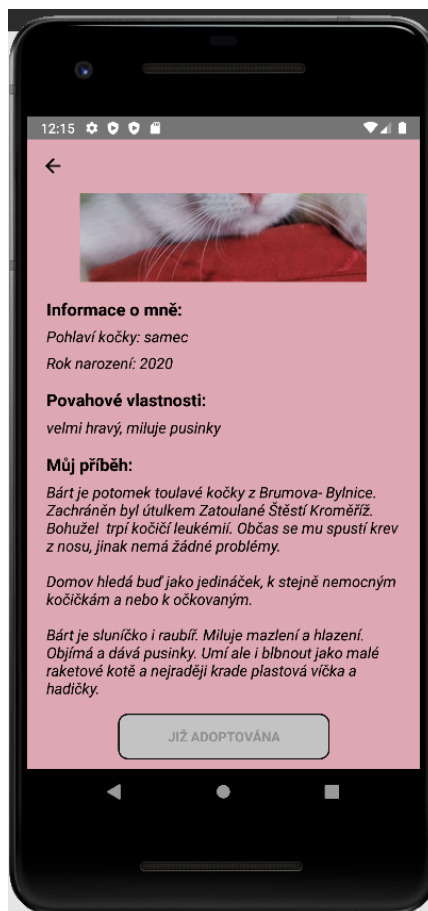
U virtuální kočky se detail, oproti detailu hned po provedení adoptace, nemění a zůstává stejný, to znamená neaktivní tlačítko s textem „Čeká na zaplacení“ a přidané tlačítko „Zobrazit QR“.

U klasické adoptace se detail kočky mění, a to pouze textace u neaktivního tlačítka. Nyní místo „Čeká na potvrzení“ se zobrazuje „Čeká na schválení“, viz Obrázek 40.



Obrázek 40 - Stránka uživatele – Detail adoptované kočky (aplikace)

Jakmile je virtuální či klasická adopce vyřizena, uživateli se změní u neaktivních tlačítek na „Již adoptovaná“, viz Obrázek 41. Nyní to lze nasimulovat pouze přes databázi, a to pomocí změny sloupce *JeVyrizena* z *false* na *true*, který se nachází v tabulce *kocky_uzivatele*. Zde je názorná ukázka, jak to vypadá v aplikaci.



Obrázek 41 - Detail kočky – Ukázka vyřizené adopce (aplikace)

7.2.6 Připojení na API

Připojení na API je realizováno pomocí „HttpClienta“ ze System.Net.Http jmenného prostoru knihovny .NET. Pomocí tohoto klienta voláme jednotlivé endpointy API služby. Využíváme volání typu GET pro získání dat a volání typu POST k odeslání dat. Aplikace využívá autorizovaná ale i neautorizovaná volání API. Neautorizovaná volání se v aplikaci využívají k získání koček a kaváren. Zatímco autorizovaná volání se využívají k získání adoptovaných koček, vyvolání adopce a získání přehledu informací o uživateli.

7.3 Budoucí vývoj aplikace

Aplikace by se mohla v budoucnu rozrůst o další adopční kavárny. Díky vytvořené API službě stačí údaje o další kavárně vložit do databáze.

Seznam kaváren by byl obměněn za mapu české republiky, která by byla rozdělena na kraje. Uživatel by měl možnost vybrat kraj, ve kterém hledá adopční kočičí kavárnu. Po kliknutí na konkrétní kraj by měl uživatel možnost zobrazit kočičí adopční kavárny právě ve vybraném kraji.

Uživatelský profil by obsahoval více údajů o uživateli, např. jméno, příjmení, telefon apod. Uživatel by tyto údaje měl možnost vyplnit ve svém uživatelském profilu. Při fyzické adopci bude potřeba dodat kavárně co nejvíce dat o uživateli. Aby uživatel nemusel při každé adopci zadávat všechny údaje znovu a znovu, uloží si je do svého uživatelského profilu. Tyto údaje bude možné měnit. Bude možné měnit i e-mailovou adresu, která je nyní pouze zobrazená.

Po kliknutí na tlačítko „Adoptovat“ bude kavárna obeznámena pomocí e-mailu, že má někdo zájem adoptovat kočku. Jestliže se bude jednat o fyzickou adopci, kavárna se s uživatelem spojí a komunikace probíhá již mimo aplikaci. Pokud adopce bude schválena kavárnou, uživatel v profilu uvidí, že má kočku adoptovanou. Tato kočka bude následně ze seznamu konkrétní kavárny odebrána.

U virtuální adopce je možné, aby kočku mělo adoptováno více uživatelů. Do budoucna je v plánu zde přidat i bankovní bránu, nyní se používá pouze QR kód či manuální platba.

Do stránky kavárny se v budoucnu přidá další karta s názvem „Naši kočičáci“, kde budou vypsaní uživatelé, kteří mají virtuálně adoptovanou kočku. Jestliže uživatelé souhlasili se zveřejněním údajů, bude se zobrazovat jejich jméno a příjmení, jestliže tyto údaje nevyplňovali v uživatelském profilu, zobrazuje se pouze jejich email. Dále se bude zobrazovat jméno adoptované kočky a přispěná částka.

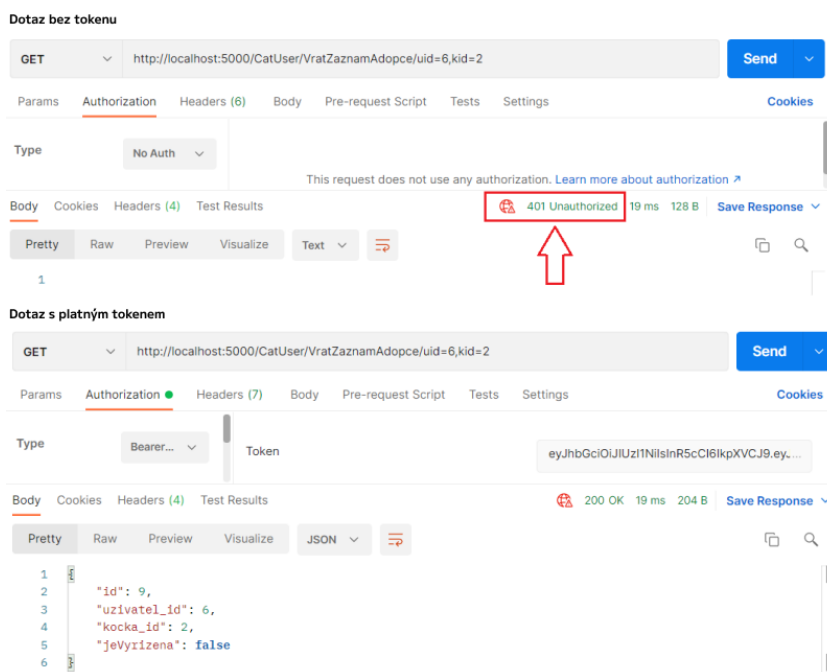
Velkou oblastí by bylo přidání administrátorské části z důvodu, aby kavárna měla možnost upravovat seznamy koček, jejich detail, popřípadě popis samotné kavárny.

8 OVĚŘENÍ BEZPEČNOSTI ŘEŠENÍ

Aplikace obsahuje bezpečnostní prvky pro zajištění ochrany před neautorizovanou adopcí. Uživatel nemá možnost bez přihlášení adoptovat jakoukoliv kočku. Je nutná registrace a následné přihlášení pro autorizovanou adopci. Pomocí API je vytvářen uživateli token, který se generuje každému uživateli po přihlášení.

API slouží jako zprostředkovatel přístupu do databáze. Rozděluje mobilní aplikaci a databázi tak, že samotná mobilní aplikace nemá přístup ke všem tabulkám v databázi. Na rozdíl od přímého přístupu mobilní aplikace do databáze, je API bezpečnější a lépe kontrolovatelné.

Pomocí Postmana byl ověřen přístup k autorizovanému endpointu. Jestliže se uživatel dotazuje bez tokenu, přístup je mu odepřen ale pokud se dotazuje s platným tokenem tak získá z endpointu validní odpověď, viz Obrázek 42.



Obrázek 42 - Rozdíl mezi autorizovaným a neautorizovaným voláním

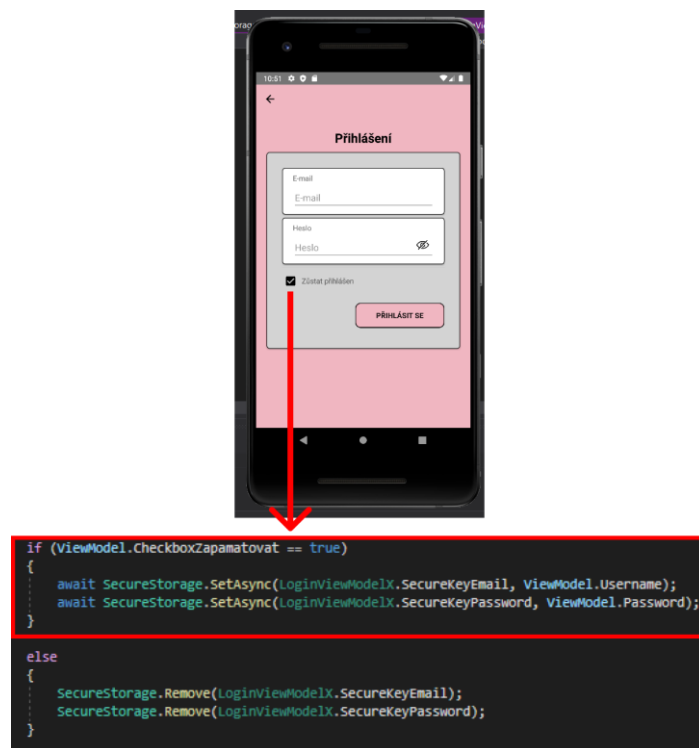
8.1 Registrace

Pomocí registrace se uživatel může zaregistrovat. Je vyžadováno vložení e-mailové adresy a zadání hesla 2x po sobě, a to kvůli shodě obou hesel. Je požadován souhlas se zpracováním osobních údajů, který je uživatel povinen zaškrtnout. Dokument ke zpracování osobních údajů bude vyřešen až při nasazení aplikace, nyní se jedná pouze o vytvoření funkční části aplikace. Mimo jiné je vyžadovaný korektní formát emailu. Heslo musí splňovat nastavené minimální požadavky, např. minimální počet znaků. Pokud není splněno některé z kritérií,

registraci nelze provést. Toto zabezpečení je vytvořeno pomocí frameworku Identity v API, na které se mobilní aplikace pouze dotazuje a reaguje na odpověď, kterou API obdrží.

8.2 Přihlášení

K úspěšnému přihlášení je požadovaná předchozí registrace uživatele, i zde se využívá frameworku Identity na straně API. Po úspěšném přihlášení se uživateli zobrazí zabezpečená část aplikace, jako je uživatelský profil a možnost adopce. Při přihlášení má uživatel v aplikaci možnost zůstat přihlášen. K tomuto účelu bylo využito třídy SecureStorage z jmenného prostoru Xamarin.Essentials. Třída SecureStorage slouží k bezpečnému uložení jednoduchých dat v podobě klíče versus jeho hodnoty. Na Obrázek 43 je ukázka kódu.



Obrázek 43 - Uložení přihlašovacích údajů do SecureStorage (aplikace + kód)

8.3 Tokeny

Tokeny v aplikaci jsou využívány k ověření, zda je uživatel v aplikaci přihlášený. Platný token je aplikaci udělen po přihlášení. Token uděluje web API mobilní aplikaci. Udělený token obsahuje identifikační textový řetězec, čas expirace a id přihlášeného uživatele.

Identifikační textový řetězec je identifikátor sloužící k identifikaci aktuální relace.

Čas expirace, je čas, po kterém vyprší platnost tokenu a dojde k jeho zneplatnění.

Id přihlášeného uživatele slouží k následné identifikaci uživatele v dotazech vůči API.

8.4 API

API uděluje aplikaci platný token po korektním přihlášení. Tento token si API zapamatuje a následně vůči němu porovnává token z aplikace. API poskytuje endpointy, které vyžadují autorizaci a které naopak autorizaci nevyžadují. Jestliže autorizaci vyžadují, tak součástí dotazu volajícího API endpointu musí být platný token.

8.5 Přístup k datům

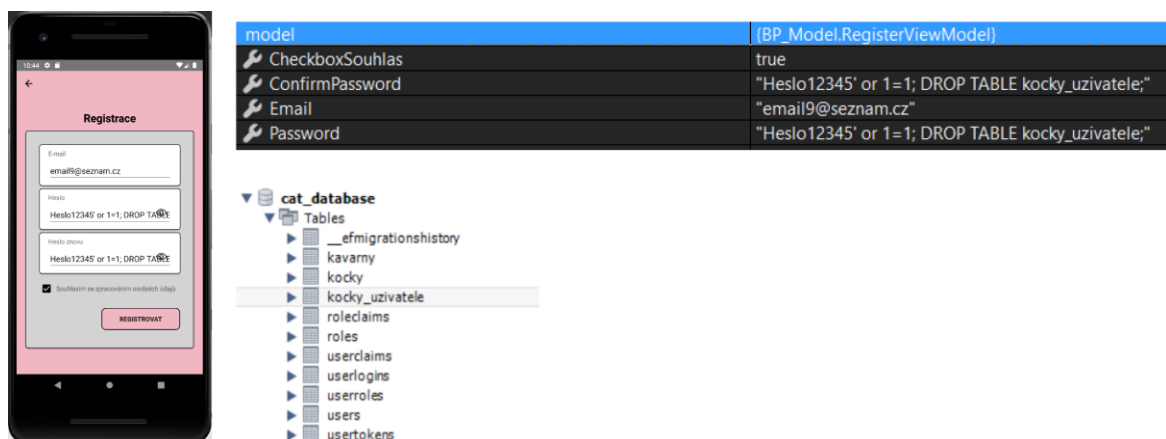
Přístup k datům daného uživatele je řešený skrze endpointy, které vyžadují autorizaci dotazů. Autorizovaný dotaz je dotaz s platným tokenem.

8.6 SQL Injection

Byl vyzkoušen databázový útok SQL injection, který se nezdařil. Byly odzkoušeny všechny vstupy, které aplikace má, a to email a heslo. U obou vstupů prošlo vše v pořádku a nedošlo k narušení databáze.

U registrace byly odzkoušeny 2 vstupy u hesla. Jeden ve tvaru - ' or 1=1; DROP TABLE kocky_uzivatele; a druhý ve tvaru - ' or 1=1; DROP TABLE kocky_uzivatele; --. Registrace se podařila, ale tabulka kocky_uzivatele nebyla smazána, viz Obrázek 44.

U emailu se se zadával nejdříve email a pak "' or 1=1; DROP TABLE kocky_uzivatele". Ani tady se SQL útok nepodařil. Do emailu je umožněno zadat pouze písmena, číslice a znak tečky a zavináče, to v tomto případě neodpovídá a framework Identity odpoví chybovou hláškou o nekorektním formátu emailové adresy.



Obrázek 44 - Ukázka útoku SQL Injection

ZÁVĚR

Předmětem bakalářské práce byla realizace aplikace za účelem zprostředkování fyzické a virtuální adopce koček z adopční kavárny. Byla navázána spolupráce s Kroměřížskou kočičí kavárnou, která specifikovala svoje požadavky na mobilní aplikaci, které by měli být splněné pro spokojenost kavárny a zároveň jejich návštěvníků. Tím bylo docíleno získání praktického pohledu zákazníka na cílovou doménu. Z následující konzultace vyšly požadavky na mobilní aplikaci. Aplikace byla vyvíjena pro operační systém Android. Nicméně jedním z požadavků byla také možnost jednoduché rozšiřitelnosti mobilní aplikace na systém iOS, což framework Xamarin umožňuje.

Teoretická část práce se zabývá přezkoumáváním možností použití existujících technologií v této oblasti. První část se zabývá druhy mobilních aplikací. V druhé části popisuje možnosti systému Android. V další části jsou popsány možnosti konkurenční platformy iOS od Applu. Po této kapitole následuje kapitola s popisem obchodů s mobilními aplikacemi. Poslední kapitola se věnuje popsání konkrétních možností vývoje pomocí frameworku Xamarin.Forms pro programovací jazyk C#.

Praktická část začíná analýzou požadavků klienta, tedy kočičí kavárny z Kroměříže. Jejich realizace byla dále popsána požadavky užití. Následně bylo navrženo uživatelské rozhraní pomocí grafického modelu rozložení ovládacích prvků.

Na začátku programování mobilní aplikace bylo zjištěno, že některé postupy realizace nebudou možné a budou muset být nahrazeny jinými. Největší zjištěný problém byl, že samotná mobilní aplikace nebude moci být přímo napojena na databázi. To bylo z důvodu, že principiálně není z hlediska bezpečnosti vhodné přistupovat do databáze napřímo z mobilní aplikace. Proto vznikl menší projekt, kterým bylo vytvoření RESTového API. Tímto byl vytvořen prostředník mezi mobilní aplikací a databází. Díky API byla získána lepší bezpečnost přístupu do databáze, kdy je možné spravovat přístup mobilní aplikace na úrovni API a ne přímo v databázi, kde by to mohlo představovat bezpečnostní riziko. Prakticky samotná mobilní aplikace neví o existenci databáze, ví pouze o existenci API a snižuje se tedy riziko nežádoucího přístupu k datům uloženým v databázi.

Protože bylo nutné získat totožnost uživatele žádajícího o adopci, byla vytvořena registrace do aplikace. Pro samotnou adopci je nutná registrace v aplikaci a následné přihlášení. V uživatelském profilu je možné sledovat adoptované kočky a stav jejich adopce.

V budoucnu je v plánu rozšířit tuto aplikaci z platformy Android na konkurenční platformu iOS a pokrýt tak širší okruh uživatelů. Zároveň by bylo vhodné zavést širší možnosti plateb za virtuální adopce. V současné době se používá pouze možnost platby QR nebo manuální platba zadáním platebních pokynů do internetového nebo mobilního bankovníctví. Do budoucna se plánuje zavedení platební brány, například od společnosti ComGate. Další velkou oblastí by bylo vytvoření administrátorské části aplikace pro správu ze strany kavárny.

SEZNAM POUŽITÉ LITERATURY

- [1] Mobilní aplikace. *APPKEE* [online]. c 2015-2022 [cit. 2022-05-02]. Dostupné z: <https://www.appkee.cz/stranka/mobilni-aplikace>
- [2] MROCZKOWSKA, Agnieszka. What Is a Mobile App? | App Development Basics for Businesses. *DROIDS ON ROIDS* [online]. 01.02.2021 [cit. 2022-05-03]. Dostupné z: <https://www.thedroidsonroids.com/blog/what-is-a-mobile-app-app-development-basics-for-businesses>
- [3] Mobile Application (Mobile App). *Techopedia* [online]. 07.08.2020 [cit. 2022-05-03]. Dostupné z: <https://www.techopedia.com/definition/2953/mobile-application-mobile-app>
- [4] Mobile App Testing Challenges Rise with Higher User Expectations. In: *QualityLogic* [online]. c 2022 [cit. 2022-05-03]. Dostupné z: <https://www.qualitylogic.com/2018/06/11/mobile-app-testing-challenges-rise-with-higher-user-expectations/>
- [5] KOŘOUSKOVÁ, Barbora. Web, Webová stránka a webová aplikace, v čem je rozdíl?. *Rascasone* [online]. 22.06.2021 [cit. 2022-05-03]. Dostupné z: <https://www.rascasone.com/cs/blog/web-webova-aplikace-rozdil>
- [6] Webová aplikace (Web Application). *Management Mania* [online]. c 2011-2016 [cit. 2022-05-03]. Dostupné z: <https://managementmania.com/cs/webova-aplikace-web-application>
- [7] MOC, Dalibor. *Vývoj webových aplikací* [online]. Praha, 2019 [cit. 2022-05-03]. Dostupné z: https://is.ambis.cz/th/dfcat/Diplomova_prace_Dalibor_MOC.pdf. Diplomová práce. Vysoká škola regionálního rozvoje a Bankovní institut – AMBIS Katedra financí a ekonomie. Vedoucí práce Ing. Vladimír Beneš, Ph.D.
- [8] Objevte svět webových aplikací. *Vývoj webových aplikací* [online]. [b.r.] [cit. 2022-05-03]. Dostupné z: https://vyvojwebovychaplikaci.cz/#web_app
- [9] MCWHERTER, Jeff, Scott GOWELL, David SMITH, Lauren COLTON, Ameila MARSCHALL-MILLER a Adam RYDER. Professional mobile application development. Indianapolis: John Wiley & Sons, [2012], xxviii, 403 s. ISBN 9781118203903.

- [10] Native Apps, Web Apps or Hybrid Apps? What's the Difference?. *MobiLoud* [online]. c 2022 [cit. 2022-05-03]. Dostupné z: <https://www.mobiloud.com/blog/native-web-or-hybrid-apps#5>
- [11] Hybridní, nativní nebo webové aplikace?. *iQuest* [online]. c 2020 [cit. 2022-05-03]. Dostupné z: <http://www.hybridniaplikace.cz/srovnani.html>
- [12] GILLIS, Alexander S. Native app. *TechTarget* [online]. c 2006 - 2022 [cit. 2022-05-03]. Dostupné z: <https://searchsoftwarequality.techtarget.com/definition/native-application-native-app>
- [13] MIŠTOVÁ, Jaroslava. Nativní aplikace. *Ackee* [online]. 23.10.2018 [cit. 2022-05-03]. Dostupné z: <https://www.ackee.cz/blog/glossary/nativni-aplikace>
- [14] What is the difference between a native mobile app vs. web app?. In: *Insivia* [online]. [b.r.] [cit. 2022-05-03]. Dostupné z: <https://www.insivia.com/what-is-the-difference-between-a-native-mobile-app-vs-web-app/>
- [15] Hybrid application (hybrid app). *TechTarget* [online]. c 2006-2022 [cit. 2022-05-03]. Dostupné z: <https://searchsoftwarequality.techtarget.com/definition/hybrid-application-hybrid-app>
- [16] GRIFFITH, Chriss. What is Hybrid App Development?. *Ionic* [online]. c 2022 [cit. 2022-05-03]. Dostupné z: <https://ionic.io/resources/articles/what-is-hybrid-app-development>
- [17] Native App vs Hybrid App vs Web App. In: *Ami Web Solutions* [online]. [b.r.] [cit. 2022-05-03]. Dostupné z: <https://www.amiwebsolutions.com/mobile-app-development/native-hybrid-web-apps/>
- [18] What is Android. *Javatpoint* [online]. c 2011-2021 [cit. 2022-05-03]. Dostupné z: <https://www.javatpoint.com/android-what-where-and-why>
- [19] Co je to ten Android?. *Android aplikace* [online]. [b.r.] [cit. 2022-05-03]. Dostupné z: <https://androidaplikace.cz/index.php/co-je-operacni-system-android/>

- [20] CHEN, James. Android Operating System. *Investopedia* [online]. 03.02.2021 [cit. 2022-05-04]. Dostupné z: <https://www.investopedia.com/terms/a/android-operating-system.asp>
- [21] Open Handset Alliance (OHA). *Techopedia* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://www.techopedia.com/definition/16273/open-handset-alliance-oha>
- [22] What would it take to build a better mobile phone?. *Open Handset Alliance* [online]. [cca. 2009] [cit. 2022-05-04]. Dostupné z: <https://www.openhandsetalliance.com/>
- [23] Mobile Operators. *Open Handset Alliance* [online]. [cca. 2009] [cit. 2022-05-04]. Dostupné z: https://www.openhandsetalliance.com/oha_members.html
- [24] BROWN, C. Scott. What is Android? Here's everything you need to know. *Android Authority* [online]. 19.02.2022 [cit. 2022-05-04]. Dostupné z: <https://www.androidauthority.com/what-is-android-328076/>
- [25] KRAJCI, Iggy a Darren CUMMINGS. *Android on x86: An Introduction to Optimizing for Intel Architecture* [online]. V Berkeley, Kalifornie: Apress, c 2013 [cit. 2022-05-04]. ISBN 978-1-4302-6131-5. Dostupné z: <https://link.springer.com/content/pdf/10.1007/978-1-4302-6131-5.pdf>
- [26] File:Android-System-Architecture.svg. In: *Wikimedia Commons* [online]. 26.02.2022 [cit. 2022-05-04]. Dostupné z: <https://commons.wikimedia.org/wiki/File:Android-System-Architecture.svg>
- [27] Android Architecture – 5 Components of Android Architecture. *Data Flair* [online]. [cca. 2019] [cit. 2022-05-04]. Dostupné z: <https://data-flair.training/blogs/android-architecture/>
- [28] <https://www.geeksforgeeks.org/android-architecture/>. *GeeksforGeeks* [online]. 26.02.2021 [cit. 2022-05-04]. Dostupné z: <https://www.geeksforgeeks.org/android-architecture/>
- [29] Linux Kernel Tutorial for Beginners. *Linuxhint* [online]. [cca. 2017] [cit. 2022-05-04]. Dostupné z: <https://linuxhint.com/linux-kernel-tutorial-beginners/>
- [30] What Is SQLite. *SQLite Tutorial* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://www.sqlitetutorial.net/what-is-sqlite/>

- [31] OpenGL ES. *Developers Android* [online]. [b.r.] [cit. 2022-05-04]. Dostupné z: <https://developer.android.com/guide/topics/graphics/opengl>
- [32] FreeType is a freely available software library to render fonts. *FreeType* [online]. 2022 [cit. 2022-05-04]. Dostupné z: freetype.org
- [33] What is SSL?. *SSL* [online]. 28.09.2021 [cit. 2022-05-04]. Dostupné z: <https://www.ssl.com/faqs/faq-what-is-ssl/>
- [34] What is SSL, TLS and HTTPS?. *Digicert* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>
- [35] GOLDING, Paul. *Connected Services: A Guide to the Internet Technologies Shaping the Future of Mobile Services and Operators*. V Hoboken, New Jersey: Wiley, 2011. ISBN 9780470974551.
- [36] Welcome to Skia: The 2D Graphics Library. *Skia* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://skia.org/>
- [37] Android Architecture. *SPatidar blog* [online]. 30.11.2016 [cit. 2022-05-04]. Dostupné z: <https://spatidarblogger.wordpress.com/2016/11/30/android-architecture/>
- [38] Dalvik Virtual Machine | DVM. *Javatpoint* [online]. c 2011-2021 [cit. 2022-05-04]. Dostupné z: <https://www.javatpoint.com/dalvik-virtual-machine>
- [39] Android Dalvik Virtual Machine. *TechVidvan* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://techvidvan.com/tutorials/android-dalvik-virtual-machine/>
- [40] Dalvik Virtual Machine | DVM. In: *Fastread* [online]. c 2012 [cit. 2022-05-04]. Dostupné z: https://www.fastread.in/tutor?Course_id=630&CourseName=Android&CourseH=Dalvik%20Virtual%20Machine%20%7C%20DVM
- [41] THOMPSON, Barbara. Android Architecture: Application Layers, Framework, Component. *Guru99* [online]. 19.03.2022 [cit. 2022-05-04]. Dostupné z: <https://www.guru99.com/android-architecture.html>
- [42] ActivityManager. *Developers Android* [online]. [b.r.] [cit. 2022-05-04]. Dostupné z: <https://developer.android.com/reference/android/app/ActivityManager>

- [43] Window Manager. *Techopedia* [online]. 25.04.2013 [cit. 2022-05-04]. Dostupné z: <https://www.techopedia.com/definition/10043/window-manager>
- [44] App resources overview. *Developers Android* [online]. [b.r.] [cit. 2022-05-04]. Dostupné z: <https://developer.android.com/guide/topics/resources/providing-resources>
- [45] NotificationManager. *Developers Android* [online]. [b.r.] [cit. 2022-05-04]. Dostupné z: <https://developer.android.com/reference/android/app/NotificationManager>
- [46] Platform Architecture. *Developers Android* [online]. [b.r.] [cit. 2022-05-04]. Dostupné z: <https://developer.android.com/guide/platform>
- [47] CALLAHAM, John. The history of Android: The evolution of the biggest mobile OS in the world. *Android Authority* [online]. 25.12.2021 [cit. 2022-05-04]. Dostupné z: <https://www.androidauthority.com/history-android-os-name-789433/>
- [48] History of Android. *Javatpoint* [online]. c 2011-2021 [cit. 2022-05-04]. Dostupné z: <https://www.javatpoint.com/android-history-and-versions>
- [49] Google's Android OS: Past, Present, and Future. *Phone arena* [online]. 17.08.2011 [cit. 2022-05-04]. Dostupné z: https://www.phonearena.com/news/Googles-Android-OS-Past-Present-and-Future_id21273
- [50] Difference between Android 1.0 and Android 1.1. *GeeksforGeeks* [online]. 02.07.2020 [cit. 2022-05-04]. Dostupné z: <https://www.geeksforgeeks.org/difference-between-android-1-0-and-android-1-1/>
- [51] Smart Reply. *Android* [online]. [b.r.] [cit. 2022-05-04]. Dostupné z: <https://www.android.com/android-10/#q-smart-reply>
- [52] Control your phone with your voice. *Android* [online]. [b.r.] [cit. 2022-05-04]. Dostupné z: <https://www.android.com/android-11/#a11-accessibility>
- [53] Android 12. *Android* [online]. [b.r.] [cit. 2022-05-04]. Dostupné z: <https://www.android.com/android-12/>
- [54] Android Logo. *Logos-world* [online]. 17.02.2022 [cit. 2022-05-04]. Dostupné z: <https://logos-world.net/android-logo/>

- [55] Android logo. *1000Logos* [online]. 30.03.2022 [cit. 2022-05-04]. Dostupné z: <https://1000logos.net/android-logo/>
- [56] KENTON, Will. Apple iOS. *Investopedia* [online]. 25.10.2021 [cit. 2022-05-04]. Dostupné z: <https://www.investopedia.com/terms/a/apple-ios.asp>
- [57] IOS. *Techopedia* [online]. 28.08.2012 [cit. 2022-05-04]. Dostupné z: <https://www.techopedia.com/definition/25206/ios>
- [58] IOS. *Mobilizujeme* [online]. c 2007-2022 [cit. 2022-05-04]. Dostupné z: <https://mobilizujeme.cz/stitky/ios>
- [59] “Open” vs. “Closed” Software Ecosystems: A Primer. *Lease Pilot* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://leasepilot.co/blog/open-vs-closed-software-ecosystems-a-primer/>
- [60] HOFFMAN, Chris. Android Is “Open” and iOS Is “Closed” — But What Does That Mean to You?. *How-To Geek* [online]. 20.06.2017 [cit. 2022-05-04]. Dostupné z: <https://www.howtogeek.com/217593/android-is-open-and-ios-is-closed-but-what-does-that-mean-to-you/>
- [61] Architecture of iOS. *SSLA* [online]. c 2013 [cit. 2022-05-04]. Dostupné z: <https://www.ssla.co.uk/architecture-of-ios/>
- [62] IOS Layered Architecture. In: *IOS Programming Tricks* [online]. 27.09.2017 [cit. 2022-05-04]. Dostupné z: <https://codeingwithios.blogspot.com/2017/09/ios-layered-architecture.html>
- [63] IOS Architecture Components: Layers & Components. *Study* [online]. c 2003-2022 [cit. 2022-05-04]. Dostupné z: <https://study.com/academy/lesson/ios-architecture-components-layers-components.html>
- [64] Architecture of IOS Operating System. *GeeksforGeeks* [online]. 15.03.2021 [cit. 2022-05-04]. Dostupné z: <https://www.geeksforgeeks.org/architecture-of-ios-operating-system/>
- [65] Apple iOS Architecture. *Tutorialspoint* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://www.tutorialspoint.com/apple-ios-architecture>
- [66] MapKit. *Developer Apple* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://developer.apple.com/documentation/mapkit/>

- [67] What is the architecture of iOS. *IntelliPaat* [online]. c 2011-2022 [cit. 2022-05-04]. Dostupné z: <https://intellipaas.com/blog/tutorial/ios-tutorial/ios-architecture/>
- [68] Core Image. *Developer Apple* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://developer.apple.com/documentation/coreimage>
- [69] Media Player. *Developer Apple* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://developer.apple.com/documentation/mediaplayer>
- [70] What is OpenAL?. *OpenAL* [online]. [b.r.] [cit. 2022-05-04]. Dostupné z: <https://www.openal.org/>
- [71] Core Media. *Developer Apple* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://developer.apple.com/documentation/coremedia>
- [72] Top 20 Mobile App Development Tools for Building Apps. *Spaceo technologies* [online]. 14.06.2021 [cit. 2022-05-04]. Dostupné z: <https://www.spaceotechnologies.com/best-ios-development-tools/>
- [73] 25 Must-have tools for Apple platforms developers + Infographic. *Codemagic* [online]. 31.07.2021 [cit. 2022-05-04]. Dostupné z: <https://blog.codemagic.io/25-must-have-tools-as-an-apple-platforms-developer/>
- [74] Xcode vs Swift. *KnowLedgeHut* [online]. 11.11.2019 [cit. 2022-05-04]. Dostupné z: <https://www.knowledgehut.com/blog/programming/swift-vs-xcode>
- [75] BLEWITT, Alex. JetBrains releases AppCode, an IDE for Objective-C. *InfoQ* [online]. 27.10.2011 [cit. 2022-05-04]. Dostupné z: <https://www.infoq.com/news/2011/10/jetbrains-appcode/>
- [76] Running Your App in the Simulator or on a Device. *Developer Apple* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://developer.apple.com/documentation/xcode/running-your-app-in-the-simulator-or-on-a-device>
- [77] Beta Testing Made Simple with TestFlight. *Developer Apple* [online]. c 2022 [cit. 2022-05-04]. Dostupné z: <https://developer.apple.com/testflight/>
- [78] RICHARDSON, Angelique. The Founding of Apple Computers, Inc. *Library - Library of Congress* [online]. 2008 [cit. 2022-05-04]. Dostupné z:

<https://guides.loc.gov/this-month-in-business-history/april/apple-computers-founded>

- [79] LEVY, Steven. Apple Inc. American company. *Britannica* [online]. 04.04.2022 [cit. 2022-05-04]. Dostupné z: <https://www.britannica.com/topic/Apple-Inc>
- [80] The History Of Apple. In: *Feedough* [online]. 07.09.2021 [cit. 2022-05-04]. Dostupné z: <https://www.feedough.com/the-history-of-apple/>
- [81] Jak publikovat aplikaci na Google Play a App Store?. *ThinkEasy* [online]. 11.06. 2020 [cit. 2022-05-04]. Dostupné z: <https://thinkeasy.cz/jak-publikovat-aplikaci-na-google-play-a-app-store/>
- [82] Generate an upload key and keystore. *Developers Android* [online]. 31.03.2022 [cit. 2022-05-04]. Dostupné z: <https://developer.android.com/studio/publish/app-signing#generate-key>
- [83] Sign your app. *Developers Android* [online]. 31.03.2022 [cit. 2022-05-05]. Dostupné z: <https://developer.android.com/studio/publish/app-signing>
- [84] Vytvoření a nastavení aplikace. *Nápověda Play Console* [online]. c 2022 [cit. 2022-05-05]. Dostupné z: <https://support.google.com/googleplay/android-developer/answer/9859152?hl=cs#zippy=%2Cmaxim%C3%A1ln%C3%AD-velikost>
- [85] Publish your app. *Developers Android* [online]. 02.08.2021 [cit. 2022-05-05]. Dostupné z: <https://developer.android.com/studio/publish>
- [86] Create and set up your app. *Play Console Help* [online]. Google, 2020 [cit. 2020-11-27]. Dostupné z: <https://support.google.com/googleplay/android-developer/answer/9859152?hl=en#>
- [87] ORAGUI, David. How to Publish an App on Google Play: A Step-by-Step Guide. *The Manifest* [online]. 15.06.2018 [cit. 2022-05-05]. Dostupné z: <https://themanifest.com/app-development/blog/publish-app-google-play>
- [88] Before You Submit. *Developer Apple* [online]. 30.03.2022 [cit. 2022-05-05]. Dostupné z: <https://developer.apple.com/app-store/review/guidelines/#before-you-submit>

- [89] CHING, Chris. Prepare your app for submission. *Code with Chris* [online]. 31.10.2019 [cit. 2022-05-05]. Dostupné z: <https://codewithchris.com/submit-your-app-to-the-app-store/#prepare-your-app>
- [90] Publikování aplikací Xamarin. iOS do App Store. *Microsoft Build* [online]. 03.04.2022 [cit. 2022-05-05]. Dostupné z: <https://docs.microsoft.com/cs-cz/xamarin/ios/deploy-test/app-distribution/app-store-distribution/publishing-to-the-app-store?tabs=windows>
- [91] App Store Connect. *Developer Apple* [online]. c 2022 [cit. 2022-05-05]. Dostupné z: <https://developer.apple.com/support/app-store-connect/>
- [92] CHING, Chris. How To Submit Your App To the App Store. *Code with Chris* [online]. 31.10.2019 [cit. 2022-05-05]. Dostupné z: <https://codewithchris.com/submit-your-app-to-the-app-store/>
- [93] CAPPELAS, Charles. Publishing an iOS App to App Store. *Medium* [online]. 18.01.2021 [cit. 2022-05-05]. Dostupné z: <https://medium.com/swlh/publishing-an-ios-app-to-app-store-d4f88c3686de>
- [94] How to Submit Your App to the App Store in 2022. *Instabug* [online]. 05.01.2022 [cit. 2022-05-05]. Dostupné z: <https://instabug.com/blog/how-to-submit-app-to-app-store/>
- [95] What is .NET?. *Microsoft* [online]. c 2022 [cit. 2022-05-05]. Dostupné z: <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>
- [96] Dokumentace pro Xamarin. *Docs.microsoft.com* [online]. Microsoft, 2020 [cit. 2020-11-27]. Dostupné z: <https://docs.microsoft.com/cs-cz/xamarin/>
- [97] VERSLUIS, Gerald a Steven THEWISSEN. *Xamarin.Forms Solutions*. New York: APress, 2018, 281 s. ISBN 9781484241349.
- [98] CARTER, Luce. Building Your First Mobile App with C# .NET and Xamarin. *Twilio blog* [online]. 06.06.2019 [cit. 2022-05-05]. Dostupné z: <https://www.twilio.com/blog/build-first-mobile-app-csharp-dotnet-xamarin>
- [99] The Good and The Bad of Xamarin Mobile Development. *Altexsoft* [online]. 13.11.2020 [cit. 2022-05-05]. Dostupné z: <https://www.altexsoft.com/blog/mobile/pros-and-cons-of-xamarin-vs-native/>

- [100] FERRISSEY, Nigel. When to Use Xamarin.Forms vs Xamarin Native?. In: *Progress Telerik* [online]. 05.10.2020 [cit. 2022-05-05]. Dostupné z: <https://www.telerik.com/blogs/when-to-use-xamarin-forms-vs-xamarin-native>
- [101] How We Are Choosing Between Xamarin.Forms vs Xamarin Native. *Jelvix* [online]. c 2022 [cit. 2022-05-05]. Dostupné z: <https://jelvix.com/blog/xamarin-forms-vs-xamarin-native>
- [102] Xamarin Native vs. Xamarin.Forms: How to Choose. *DZone* [online]. 29.04.2020 [cit. 2022-05-05]. Dostupné z: <https://dzone.com/articles/xamarin-native-vs-xamarinforms-how-to-choose>
- [103] BACO, Abdelmajid. Understanding Xamarin Forms. *Medium* [online]. 22.07.2019 [cit. 2022-05-05]. Dostupné z: <https://abdelmajid-baco.medium.com/understanding-xamarin-forms-ffc6f1b1f6e6>
- [104] Xamarin.Forms. *Microsoft* [online]. c 2022 [cit. 2022-05-05]. Dostupné z: <https://dotnet.microsoft.com/en-us/apps/xamarin/xamarin-forms>
- [105] HERMES, Dan. *Xamarin mobile application development: cross-platform C# and Xamarin.Forms fundamentals*. [New York]: Apress, [2015], 432 s. ISBN 9781484202159.
- [106] What is .NET MAUI?. *Microsoft* [online]. 21.04.2022 [cit. 2022-05-05]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/maui/what-is-maui>
- [107] Co je .NET MAUI?. *Microsoft* [online]. 03.05.2022 [cit. 2022-05-05]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/maui/what-is-maui>
- [108] .NET Multi-platform App UI (.NET MAUI). *GitHub* [online]. c 2022 [cit. 2022-05-05]. Dostupné z: <https://github.com/dotnet/maui>
- [109] What Is .NET MAUI And Should You Use It?. *Xam* [online]. c 2022 [cit. 2022-05-05]. Dostupné z: <https://xam.com.au/what-is-net-maui-and-should-you-use-it/>
- [110] RAMEL, David. .NET MAUI 'Slips the Schedule,' Won't Ship with .NET 6 in November. *Visual Studio Magazine* [online]. 15.09.2021 [cit. 2022-05-05]. Dostupné z: <https://visualstudiomagazine.com/articles/2021/09/15/net-maui-delay.aspx>

- [111] *Visual Paradigm online* [online]. c 2022 [cit. 2022-05-16]. Dostupné z: <https://online.visual-paradigm.com/>
- [112] *Figma* [online]. 2021 [cit. 2022-05-16]. Dostupné z: <https://www.figma.com/>

SEZNAM OBRÁZKŮ

Obrázek 1 - Rozdíly mobilních aplikací[4]	12
Obrázek 2 - Rozdíl mezi nativní a webovou aplikací[14]	13
Obrázek 3 - Rozdíl mezi hybridní a nativní aplikací[17]	14
Obrázek 4 - Architektura Android[26]	16
Obrázek 5 - DVM – Kompilátor Dex[40]	18
Obrázek 6 - Historie loga Android[54]	23
Obrázek 7 - Architektura iOS[62]	25
Obrázek 8 - Apple II[80]	29
Obrázek 9 - Podepisování aplikace pomocí Play App Signing[83].....	30
Obrázek 10 - Řídící panel App Store Connect[92].....	32
Obrázek 11 - Rozdíl mezi Xamarin.Native a Xamarin.Forms[100].....	33
Obrázek 12 - .NET MAUI[110]	35
Obrázek 13 - Ukázka případu užití (návrh)	40
Obrázek 14 - Úvodní obrazovka aplikace (návrh).....	41
Obrázek 15 - Registrace (návrh).....	42
Obrázek 16 - Přihlášení (návrh).....	43
Obrázek 17 - Seznam adopčních kaváren (návrh)	44
Obrázek 18 - Stránka kavárny – O nás (návrh)	45
Obrázek 19 - Stránka kavárny – Adopce (návrh)	46
Obrázek 20 - Detail kočky – Rozdíl mezi přihlášeným a nepřihlášeným uživatelem (návrh)	46
Obrázek 21 - Detail kočky – Klasická adopce (návrh).....	47
Obrázek 22 - Detail kočky – Virtuální adopce (návrh)	48
Obrázek 23 - Stránka uživatele – Uživatelský profil (návrh)	49
Obrázek 24 - Stránka uživatele – Moje virtuální adopce (návrh).....	49
Obrázek 25 - Stránka uživatele – Detail adoptované a virtuálně adoptované kočky (návrh)	50
Obrázek 26 - Schéma databáze	52
Obrázek 27 - Úvodní strana aplikace (aplikace).....	56
Obrázek 28 - Registrace – možnost zobrazení/skrytí hesla (aplikace)	57
Obrázek 29 - Podmínky nutné k registraci (kód).....	58
Obrázek 30 - Přihlášení (aplikace).....	59
Obrázek 31 - Metoda sloužící k přihlášení (kód)	60
Obrázek 32 - Seznam adopčních kaváren – nepřihlášený uživatel (aplikace).....	61

Obrázek 33 - Stránka kavárny (aplikace)	62
Obrázek 34 - Detail kočky – nepřihlášený uživatel (aplikace).....	63
Obrázek 35 - Seznam adopčních kaváren – přihlášený uživatel (aplikace)	64
Obrázek 36 - Detail kočky – přihlášený uživatel (aplikace).....	65
Obrázek 37 - Detail kočky – Klasická adopce (aplikace).....	66
Obrázek 38 - Detail kočky – Virtuální adopce (aplikace)	67
Obrázek 39 - Stránka uživatele (aplikace)	68
Obrázek 40 - Stránka uživatele – Detail adoptované kočky (aplikace).....	69
Obrázek 41 - Detail kočky – Ukázka vyřizené adopce (aplikace).....	70
Obrázek 42 - Rozdíl mezi autorizovaným a neautorizovaným voláním	72
Obrázek 43 - Uložení přihlašovacích údajů do SecureStorage (aplikace + kód)	73
Obrázek 44 - Ukázka útoku SQL Injection	74

SEZNAM PŘÍLOH

Příloha P I: Obsah CD

PŘÍLOHA P I: OBSAH CD

Obsah CD:

AdoptionApp.zip obsahující 2 složky, 1 textový soubor a 1 SQL skript:

- složka BP_AdoptionApp - obsahuje jednotlivé soubory s kódem aplikace pro adopci
- složka CatAPI - obsahuje jednotlivé soubory s kódem API aplikace
- textový soubor JakVytvoritDatabazi-migrace.txt - obsahuje návod pro tvorbu databáze pomocí migrace
- SQL skript cat_database.sql - skript pro vytvoření již naplněné databáze