

**ZÁPADOČESKÁ UNIVERZITA V PLZNI**

**FAKULTA EKONOMICKÁ**

**Bakalářská práce**

**Vývoj aplikací pro mobilní telefony**

**Development of mobile phone applications**

**Petr Špíral**

**Plzeň 2024**



## **Čestné prohlášení**

Prohlašuji, že jsem bakalářskou práci na téma

*„Vývoj aplikací pro mobilní telefony“*

vypracoval/a samostatně pod odborným dohledem vedoucí/vedoucího bakalářské práce za použití pramenů uvedených v příložené bibliografii.

Plzeň dne 21.04.2024

v. r. *Petr Špíral*

## **Zásady pro vypracování práce**

1. Popište různá vývojová prostředí a odůvodněte volbu.
2. Popište různé operační systémy a odůvodněte volbu.
3. Navrhněte a implementujte mobilní aplikaci.
4. Otestujte a zhodnoťte funkcionalitu aplikace.
5. Proved'te uživatelské testování a vyhodnoťte výsledky.

## **Studijní program**

Informační management

## **Poděkování**

Chtěl bych tímto poděkovat vedoucímu bakalářské práce, doc. RNDr. Mikuláši Gangurovi Ph.D. za odborné vedení, čas a ochotu, kterou mi při psaní této práce věnoval.

# Obsah

Úvod .....	10
<b>1 Informační systém a mobilní aplikace .....</b>	<b>11</b>
1.1 Informační systém a data .....	11
1.2 Informační systém a software a hardware .....	11
1.3 Informační systém a lidé .....	11
1.4 Infomační systém a mobilní aplikace .....	12
<b>2 Historie mobilních aplikací .....</b>	<b>13</b>
2.1 Úvod do historie mobilních aplikací .....	13
2.1.1 1992 – První smartphone od IBM .....	13
2.1.2 1997 – Fenomén had .....	13
2.1.3 2007 – Představen první iPhone .....	13
2.1.4 2008 – Spuštění App Store .....	13
2.1.5 2012 – Rozvoj operačního systému Android .....	14
2.1.6 2016 – Nástup aplikací s rozšířenou realitou .....	14
<b>3 Operační systémy mobilních aplikací .....</b>	<b>15</b>
3.1 Trh s mobilními telefony v USA, Evropě a Asii .....	16
3.2 Trh s mobilními telefony v České republice .....	17
<b>4 Proces vývoje mobilních aplikací .....</b>	<b>19</b>
4.1 Vývoj strategie .....	19
4.1.1 Identifikování cílů aplikace .....	19
4.1.2 Průzkum trhu .....	19
4.1.3 Výběr platformy .....	19
4.1.4 Monetizace .....	19
4.2 Analýza a plánování .....	20

4.2.1	Návrh funkcionalit a nefunkčních požadavků .....	20
4.2.2	Definice entit.....	21
4.3	Produktový plán.....	21
4.4	Design aplikace.....	21
4.4.1	Wireframes.....	22
4.4.2	Návrh stylů.....	22
4.4.3	Detailní Návrhy.....	22
4.4.4	Vývoj Prototypu.....	23
4.5	Programování aplikace .....	23
4.5.1	Backend .....	23
4.5.2	Frontend .....	23
4.5.3	Vývoj pro konkrétní Platformu.....	23
4.5.4	Vývoj pro více platforem zároveň (Cross-Platform) .....	24
4.5.5	Hybridní vývoj.....	24
4.6	Testování aplikace .....	24
4.6.1	Manuální vs automatické testování.....	25
4.6.2	Manuální testování.....	25
4.6.3	Automatické testování .....	25
4.7	Distribuce aplikace .....	25
4.8	Podpora aplikace.....	26
<b>5</b>	<b>Vývoj a programování aplikací pro zařízení s operačním systémem iOS.....</b>	<b>27</b>
5.1	Vývoj mobilních aplikací pro iOS .....	27
5.2	Distribuce aplikace na App Store .....	27
5.2.1	Příprava a testování.....	28
5.2.2	Vytvoření iOS distribučního certifikátu .....	28
5.2.3	Vytvoření App ID, testovací zařízení a provisioning účet.....	28

5.2.4	App Store Connect.....	28
5.2.5	Podepsání aplikace.....	28
5.2.6	Nahrání aplikace a testování pomocí TestFlight.....	28
5.2.7	Schvalovací proces .....	29
5.2.8	Oprava chyb.....	29
5.2.9	Stažení aplikace .....	29
5.2.10	Aktualizace aplikace .....	29
<b>6</b>	<b>Vývoj a programování aplikací pro zařízení s operačním systémem Android</b>	<b>30</b>
6.1	Vývoj mobilních aplikací pro android.....	30
6.2	Distribuce aplikace na Google play .....	30
<b>7</b>	<b>Cross-platform a hybridní app development .....</b>	<b>31</b>
7.1	Xamarin .....	31
7.2	React Native.....	31
7.3	Flutter.....	31
<b>8</b>	<b>Vývojová prostředí pro vývoj mobilních aplikací .....</b>	<b>33</b>
8.1	Xcode.....	33
8.2	Android studio .....	33
8.3	Visual studio .....	33
<b>9</b>	<b>Metodika .....</b>	<b>35</b>
9.1	Výběr tématu mobilní aplikace.....	35
9.2	Výběr technologie vývoje mobilní aplikace .....	35
9.3	Proces vývoje mobilní aplikace .....	35
<b>10</b>	<b>Vývoj strategie.....</b>	<b>37</b>
10.1	Strategie vývoje a definice cílů.....	37
10.2	Průzkum trhu.....	37

10.3	Výběr platformy.....	37
10.4	Monetizace aplikace .....	37
<b>11</b>	<b>Analýza a plánování.....</b>	<b>38</b>
11.1	Funkcionality aplikace Vozový park .....	38
11.2	Definice entit.....	38
<b>12</b>	<b>Produktový plán.....</b>	<b>40</b>
<b>13</b>	<b>Design aplikace.....</b>	<b>41</b>
13.1	Logo aplikace.....	41
<b>14</b>	<b>Programování aplikace.....</b>	<b>42</b>
14.1	Příprava počítače pro vývoj .....	42
14.1.1	Vytvoření projektu .....	42
14.2	Vytvoření aktivit aplikace.....	43
14.3	Základní komponenty použité při tvorbě aktivit.....	44
14.4	Zpracování vstupů.....	45
14.5	Vytvoření navigace .....	45
14.6	Realizace navigace.....	45
14.6.1	Implementace navigace.....	46
14.7	Vytvoření databáze .....	46
14.7.1	SQL příkazy .....	47
14.8	Nastavení notifikací .....	48
<b>15</b>	<b>Testování aplikace.....</b>	<b>50</b>
<b>16</b>	<b>Publikace aplikace v App Store .....</b>	<b>51</b>
16.1	Postup publikace: .....	51
16.2	První schvalovací proces .....	52
16.3	Druhý schvalovací proces .....	52
<b>17</b>	<b>Publikace aplikace na Google Play.....</b>	<b>53</b>



<b>18 Podpora Aplikace.....</b>	<b>54</b>
18.1 Vyhodnocení výsledků .....	54
<b>Závěr .....</b>	<b>56</b>
<b>Seznam použitých zdrojů .....</b>	<b>57</b>
<b>Seznam tabulek .....</b>	<b>59</b>
<b>Seznam obrázků .....</b>	<b>60</b>
<b>Seznam příloh.....</b>	<b>61</b>
<b>Přílohy</b>	
<b>Abstrakt</b>	
<b>Abstract</b>	

# Úvod

Tato bakalářská práce se zabývá tématem vývoje mobilních aplikací. Hlavním cílem práce je vytvoření funkční mobilní aplikace, která bude úspěšně distribuována na platformách jako jsou App Store a Google Play. Dílčími cíli jsou:

- výběr a vyhodnocení alternativ pro vývoj mobilních aplikací (jazyk, operační systém, vývojové prostředí)
- provedení průzkumu trhu mobilních aplikací
- návrh funkcionalit aplikace a konceptuálního schématu dat
- implementace funkční aplikace
- otestování aplikace

Na začátku práce je poskytován přehled o historii mobilních aplikací, vysvětleno, co je to informační systém a jaké jsou charakteristiky mobilní aplikace. Třetí kapitola se věnuje trhu s mobilními telefony. V další kapitole se práce věnuje procesu vývoje aplikace, který zahrnuje průzkum trhu, návrh funkcionality, samotný vývoj, testování, a nakonec distribuci na obchod s aplikacemi. Zatímco pátá kapitola je zaměřena na vývoj aplikací pro iOS, šestá kapitola se věnuje aplikacím pro operační systém Android. Tyto kapitoly zahrnují specifika související s těmito platformami, od počátečních kroků vývoje až po distribuci v příslušných obchodech s aplikacemi. Tato práce se mimo jiné zabývá i takzvaným cross-platform software developmentem, který umožňuje vytvářet aplikace pro různé operační systémy současně. Poslední část literární rešerše je zaměřena na porovnání různých možností vývoje, včetně srovnání operačních systémů, programovacích jazyků a vývojových prostředí.

Praktická část práce je zaměřena na konkrétní kroky spojené s realizací mobilní aplikace. Obsahuje výběr operačního systému, programovacího jazyka a vývojového prostředí, průzkum trhu, návrh funkcionality, samotný vývoj mobilní aplikace, její testování a nakonec distribuci na obchod s aplikacemi.

# 1 Informační systém a mobilní aplikace

Informační systémy (IS) se staly neodmyslitelnou součástí každodenního života. Jsou všudypřítomné doma, v práci dokonce když saháme po mobilním telefonu, nebo využijeme samoobslužné pokladny v obchodě. Informační systém lze definovat jako propojený soubor nejen hardwaru, softwaru, dat, ale i lidí, kteří jsou velmi důležitou součástí těchto systémů. Uživatelé, jakožto nezbytná součást informačních systémů, nejen že je používají, ale také je spravují a neustále je zdokonalují, aby lépe sloužily svému účelu a reflektovaly měnící se potřeby a technologický pokrok. (Pour, 2001)

## 1.1 Informační systém a data

Základem každého IS jsou data a informace. Abychom s daty mohli pracovat, musí být někdy uloženy. Pro ukládání dat nejčastěji slouží relační databáze. Tyto databáze využívají strukturovaný přístup ke správě dat prostřednictvím tabulek, kde jsou data organizována do řádků a sloupců. Každý řádek představuje unikátní záznam a každý sloupec určitý atribut tohoto záznamu. Díky relacím mezi tabulkami je možné efektivně propojovat různé datové sady a získávat z nich hodnotné informace. (Pour, 2001)

## 1.2 Informační systém a software a hardware

Informační software a hardware jsou klíčovými komponentami každého informačního systému. Software zahrnuje aplikace a programy, které umožňují uživatelům zpracovávat data, provádět analýzy a spravovat informace efektivně. Od databázových systémů, přes analytické nástroje, až po uživatelské rozhraní – všechny tyto prvky spadají do kategorie informačního softwaru. Na druhé straně, hardware zahrnuje fyzické komponenty, jako jsou servery, počítače, síťová zařízení a úložná zařízení, které jsou nezbytné pro uložení, zpracování a přenos dat. Společně tvoří software a hardware základ, na kterém stojí veškerá funkcionální IS, a umožňují organizacím efektivně spravovat a využívat svá data. (Pour, 2001)

## 1.3 Informační systém a lidé

Informační systémy jsou nejen o sofistikovaném softwaru a databázích, ale hlavně jsou o lidech, kteří je navrhují, spravují, používají a neustále rozvíjejí. Lidé poskytují

IS kontext, smysl a směr, protože technologie sama o sobě nemůže generovat hodnotu bez lidské interakce a zásahu. (Pour, 2001)

#### **1.4 Infomační systém a mobilní aplikace**

Mobilní aplikace představuje softwarovou komponentu určitého informačního systému, spuštěnou na mobilním hardwarovém zařízení, jako jsou chytré telefony nebo tablety, které ovládá uživatel.

## **2 Historie mobilních aplikací**

### **2.1 Úvod do historie mobilních aplikací**

Mobilní aplikace by neexistovali bez chytrých telefonů. Přesto, že chytré telefony a mobilní aplikace jsou tu s námi poměrně krátkou dobu, jejich historie je poměrně obsáhlá. Zde jsou důležité milníky, které z této historie vyčnívají.

#### **2.1.1 1992 – První smartphone od IBM**

V roce 1993 byl vydán mobilní telefon, který by mohl být považován za první smartphone na trhu a s tím i první mobilní aplikace. Telefon obsahoval aplikaci na posílání emailů, kalendář a psaní poznámek. (CHANTEL, 2023)

#### **2.1.2 1997 – Fenomén had**

Nokia 6110 představuje významný mobilní telefon, který, ačkoliv není zařazen mezi smartphony, sehrál klíčovou roli ve vývoji mobilních aplikací. Zejména hra Snake, která byla předinstalovaná na tomto modelu, představovala jednu z prvních interaktivních zábav na mobilních zařízeních. (CHANTEL, 2023)

#### **2.1.3 2007 – Představen první iPhone**

Apple revolučně změnil trh mobilních telefonů v roce 2007, když představil první iPhone. Tento telefon nejen že přinesl nový způsob interakce s mobilním zařízením díky svému dotykovému displeji, ale také položil základ pro budoucí vývoj mobilních aplikací. iPhone se stal ikonou v oblasti smartphonů a jeho operační systém iOS umožnil vývojářům vytvářet sofistikovanější a interaktivnější aplikace než kdykoliv předtím. (CHANTEL, 2023)

#### **2.1.4 2008 – Spuštění App Store**

Rok po uvedení iPhone, Apple spustil App Store, což byl první skutečně masový obchod s aplikacemi pro mobilní zařízení. Umožnil vývojářům a firmám distribuovat své aplikace přímo uživatelům, což vedlo k významnému nárůstu počtu dostupných aplikací. App Store změnil způsob, jakým lidé používají své smartphony, a stal se klíčovým hráčem v oblasti digitální ekonomiky. (The App Store turns 10, 2018)

### **2.1.5 2012 – Rozvoj operačního systému Android**

Google se s jeho operačním systémem Android stal hlavním konkurentem iOS. Díky otevřenému charakteru Androidu a různorodosti zařízení, na kterých běžel, se trh mobilních aplikací stal ještě více rozmanitým. Obchod Google Play nabídl alternativu k App Store od Apple a umožnil uživatelům přístup k širokému spektru aplikací. (Apple in 2012: Controversy and competition, 2012)

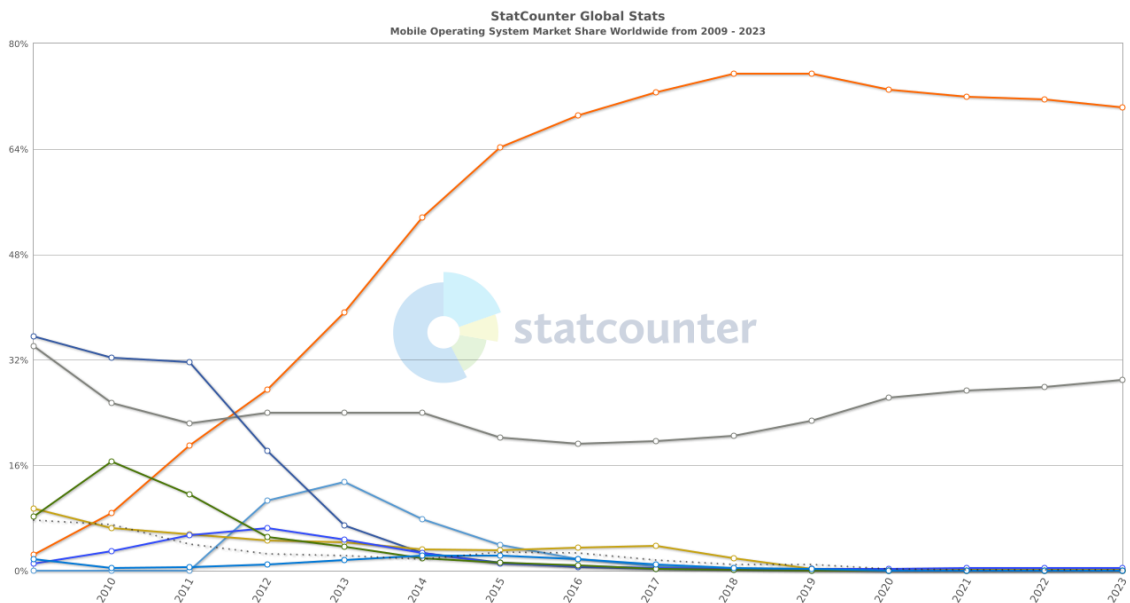
### **2.1.6 2016 – Nástup aplikací s rozšířenou realitou**

V roce 2016 došlo k průlomům v oblasti mobilních aplikací s příchodem her jako Pokémon Go, které využívaly technologii rozšířené reality (AR). Tento trend ukázal nové možnosti interakce s mobilními aplikacemi a rozšířil hranice toho, co mobilní aplikace mohou nabídnout. (Javorník, 2016)

### 3 Operační systémy mobilních aplikací

Při tvorbě mobilních aplikací je nezbytné zvážit, pro jaký operační systém bude aplikace určena, což je klíčové pro její úspěch. První smartphone, jak ho známe dnes, se objevil v roce 2009 spolu s operačním systémem iOS od společnosti Apple. Brzy poté se na trhu objevily další operační systémy, jako Android, Windows pro mobilní telefony nebo BlackBerry OS. Je podstatné poznamenat, že na rozdíl od počítačů, kde si lze vybrat a nainstalovat operační systém, nebo dokonce mít na jednom zařízení více operačních systémů, mobilní telefony jsou prodávány s předinstalovaným operačním systémem, který nelze jednoduše změnit nebo nahradit jiným. Výběr cílového operačního systému je zásadní nejen z technologického hlediska, ale i kvůli velikosti a charakteru trhu, na který je aplikace zaměřena. V následujícím grafu je možné vidět zastoupení různých operačních systémů na celosvětovém trhu s mobilními telefony v průběhu posledních téměř 14 let.

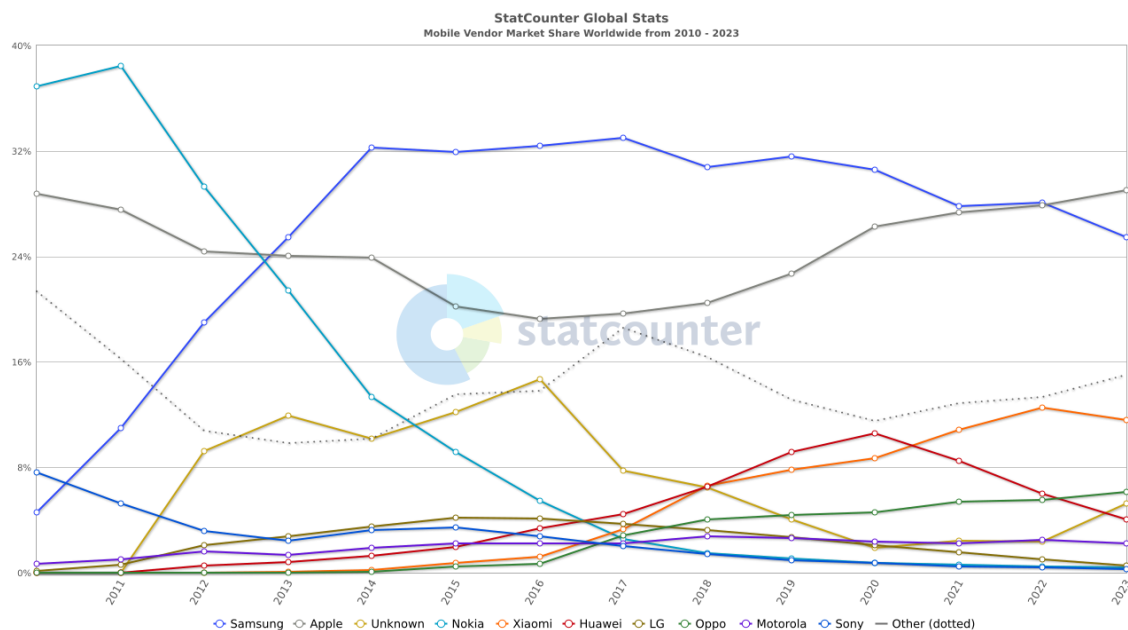
Obrázek 1: Podíl operačních systémů na trhu mobilních telefonů



Zdroj: StatCounter Global Stats, 2024

Na tomto grafu si lze povšimnout, že na počátku minulé dekády bylo na trhu vícero operačních systémů pro mobilní telefony. V současnosti se však výběr pro spotřebitele většinou omezuje na dva hlavní systémy: iOS a Android. Z grafu na obrázku č.1 lze vyčíst, že k roku 2023 využívá Android přes 70 % uživatelů, což je zejména důsledkem jeho cenové dostupnosti mobilních telefonů, který Android obsahují. Mezi nejrozšířenější výrobce telefonů s operačním systémem Android patří Samsung, Xiaomi nebo například Oppo (viz obrázek č.2). Z grafu je také patrný rostoucí trend u iOS. Na první pohled by se mohlo zdát, že iOS se pouze vrací na pozice, které měl v roce 2009, ale je důležité si uvědomit, že graf zobrazuje pouze procentuální podíl na trhu, nikoliv absolutní počet uživatelů.

Obrázek 2: Zastoupení výrobců mobilních telefonů s operačním systémem Android na trhu



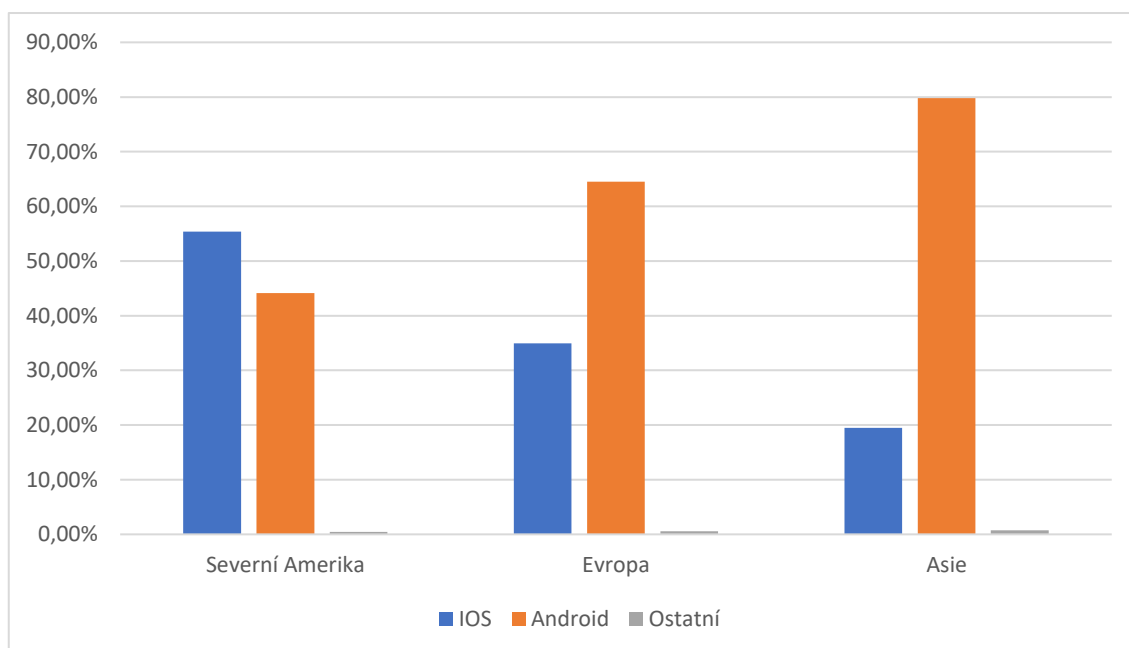
Zdroj: StatCounter Global Stats, 2024

### 3.1 Trh s mobilními telefony v USA, Evropě a Asii

Trh s mobilními telefony se výrazně liší v různých částech světa, což je důležité brát v úvahu při výběru cílového trhu pro vývoj mobilní aplikace. Zohlednění specifik lokálního trhu je klíčové. Například při tvorbě aplikace pro firmu, která působí výhradně v České republice, není nutné brát v potaz globální data. Místo toho je důležité se soustředit na preferované operační systémy v rámci České republiky. Lokalizace je pro úspěšný vývoj zásadní.



Obrázek 3: Podíl operačních systému na trhu mobilních telefonů v Severní Americe, Evropě a Asii



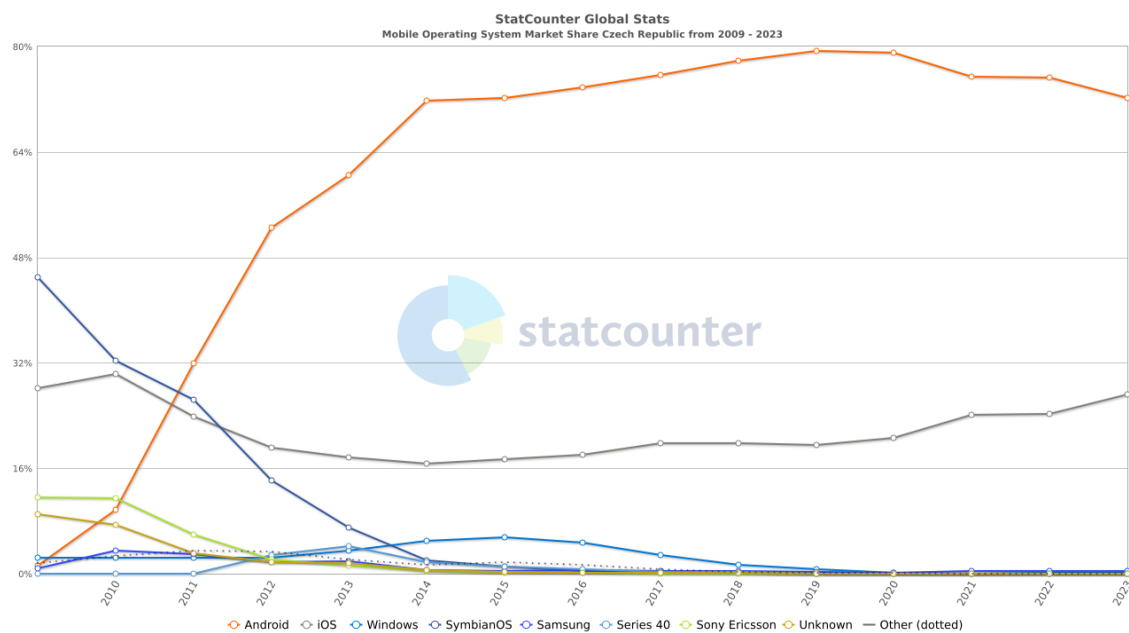
Zdroj: StatCounter Global Stats, 2024

Graf na obrázku č.3 srovnává tři hlavní trhy s mobilními telefony: Severní Amerika, Evropa a Asie. Pozorujeme, že každý z těchto trhů má rozdílné rozložení operačních systémů. V Asii je zřetelně dominantní Android, což může být důsledkem toho, že mnoho významných výrobců telefonů s tímto operačním systémem, jako jsou Samsung z Jižní Koreje nebo Xiaomi z Číny, pochází právě z tohoto regionu. Na evropském trhu Android převažuje s téměř 65 % tržním podílem, zatímco iOS má zhruba 35 %. Evropa tak v mnohém odráží globální trend. Severní Amerika se odlišuje tím, že zde má iOS větší podíl na trhu než Android, což je jedinečné ve srovnání s ostatními regiony.

### 3.2 Trh s mobilními telefony v České republice

V České republice je trh velice podobný tomu světovému a evropskému. Má zde vysoce dominantní postavení operační systém Android, který má tržní podíl více než 72 %, zatímco iOS pouze lehce přes 27 %. Na obrázku č. 4 si lze všimnout klesajícího trendu u Androidu a stoupajícího trendu u iOS.

Obrázek 4: Podíl operačních systémů na trhu mobilních telefonů



Zdroj: StatCounter Global Stats, 2024

## 4 Proces vývoje mobilních aplikací

Vývoj mobilních aplikací je komplexní a náročný proces. Proto je důležité postupovat podle určitých kroků, aby byla aplikace správně a efektivně vytvořena. Samotný vývoj aplikace začíná ještě dávno před tím, než je napsán první řádek kódu.

### 4.1 Vývoj strategie

#### 4.1.1 Identifikování cílů aplikace

Určení cílů aplikace je základem celého projektu. Následující tři klíčové otázky mohou pomoci vymezit tyto cíle: Jaký problém bude moje aplikace řešit? Kdo bude cílovou skupinou mé aplikace? Jaké výsledky očekávám, že moje aplikace splní? (Carrington, n.d.)

#### 4.1.2 Průzkum trhu

Průzkum trhu hraje klíčovou roli. Je nezbytné zjistit, zda již existuje podobná aplikace. Pokud ano, je důležité prozkoumat její úspěšnost, silné a slabé stránky. Dále je třeba posoudit, zda má smysl vyvíjet vlastní aplikaci, zda přinese nějakou přidanou hodnotu a proč by se lidé měli rozhodnout právě pro vaši aplikaci. (Carrington, n.d.)

#### 4.1.3 Výběr platformy

Dnes jsou k dispozici pouze dva hlavní operační systémy pro mobilní aplikace: Android a iOS. Výběr vhodného systému pro vývoj naší aplikace je klíčový. Běžně se vytvářejí aplikace buď pro jeden nebo pro druhý systém. Existuje také možnost vyvinout aplikaci pro oba systémy současně, což lze udělat dvěma způsoby. Jedna možnost je vytvořit dvě samostatné aplikace pro každý systém, což je však náročné. Druhou možností je využití takzvaného vývoje cross-platform vývoje aplikací, který umožňuje vytvářet aplikace pro více platform pomocí jediného kódu. (Carrington, n.d.)

#### 4.1.4 Monetizace

Posledním krokem ve vývoji strategie je výběr modelu monetizace, který by nejlépe odpovídal charakteru aplikace. Mezi hlavní modely monetizace patří:

- Reklama v aplikaci: Tento model zahrnuje zobrazování reklamních bannerů nebo videí uvnitř aplikace.

- Nákupy v aplikaci: Umožňuje uživatelům pořídit si dodatečné funkce nebo obsah přímo v aplikaci.
- Předplatné: Předplatné je v dnešní době hodně populární model zpoplatnění zejména u platforem, které streamují video obsah. V tomto modelu si uživatel předplatí službu či aplikaci na určitou dobu a po tu dobu může její služby využívat.
- Placené aplikace: V tomto modelu uživatelé platí jednorázově za stažení a používání aplikace.

Model monetizace by měl být pečlivě vybrán tak, aby odpovídal charakteru a funkcím aplikace. Například u fitness aplikace, která nabízí cvičební plány, je vhodný model předplatného. Na druhou stranu, pokud se jedná o nějaký typ vylepšeného kalendáře, může být vhodnější jednorázová platba za stažení aplikace. (Carrington, n.d.)

## 4.2 Analýza a plánování

### 4.2.1 Návrh funkcionalit a nefunkčních požadavků

Funkcionalita specifikuje soubor funkcí, které aplikace nabízí. Nefunkční požadavky popisují kvalitativní aspekty a vlastnosti aplikace, které určují, jak aplikace funguje a jak je vnímána uživateli. Jako příklad můžeme uvést populární sociální síť Instagram. (Carrington, n.d.)

Funkcionality sociální sítě Instagram:

- Vytváření a správa uživatelského profilu.
- Nastavení profilového obrázku uživatele.
- Přidání popisu k profilu.
- Publikování fotografií a videí.
- Možnost komentovat, sdílet a dávat "lajky" na příspěvky ostatních.

Nefunkční požadavky sociální sítě Instagram:

- Bezpečnost
- Spolehlivost
- Kompatibilita
- Uživatelská přívětivost (Carrington, n.d.)

#### 4.2.2 Definice entit

Při vývoji mobilní aplikace je nezbytné pečlivě navrhnout entity a definovat jejich integritní omezení. Prvním krokem je identifikovat klíčové objekty, které budou v aplikaci používány a ukládány. To může zahrnovat uživatele, produkty, objednávky nebo jakékoli jiné relevantní entity. (Carrington, 2023)

### 4.3 Produktový plán

Produktový plán, známý také jako produktová mapa nebo roadmap, je dokument, který nastiňuje sérii kroků vedoucích k dokončení aplikace. Jeho hlavním účelem je poskytnout jasný přehled o implementaci funkcí, vylepšeníh a inovacích v produktu. Existuje několik typů produktových map, přičemž každý se zaměřuje na konkrétní aspekty vývoje. Mezi běžně používané typy patří:

- Plán vydání (Release plan roadmap): Obsahuje data vydání a seznam funkcí, které budou uvedeny od počátečního vydání aplikace až po vydání jednotlivých dodatečných funkcí.
- Sprintový plán (Sprint plan roadmap): Rozděluje úkoly do specifických časových rámců, obvykle v rámci agilního vývojového procesu.
- Kanban roadmap: Slouží k organizaci úkolů do tří kategorií: backlog (seznam úkolů k budoucímu zpracování), planned (naplánované úkoly) a in progress (aktuálně zpracovávané úkoly).
- Časová osa funkcí (Feature timeline roadmap): Zaměřuje se na specifické funkce nebo vlastnosti produktu, které mají být vyvinuty a implementovány.
- Časová osa cílů (Objective timeline roadmap): Fokus na strategické cíle a obchodní výsledky, kterých má aplikace nebo projekt dosáhnout.
- Časová osa vydání (Release timeline roadmap): Podobná plánu vydání, ale zahrnuje také časové aspekty pro lepší plánování a přehlednost. (KMS SOLUTIONS, 2022)

### 4.4 Design aplikace

Dalším krokem je vytvoření návrhu uživatelského rozhraní (UI – User Interface) a návrhu uživatelského zážitku (UX – User Experience). Oba koncepty jsou pro úspěch aplikace klíčové. UI se soustředí na vizuální aspekty aplikace, včetně rozložení a grafiky. Naproti

tomu, UX se zaměřuje na celkový zážitek uživatele při používání aplikace. To zahrnuje, jak snadné a intuitivní je používání aplikace. (Carrington, n.d.)

#### **4.4.1 Wireframes**

Wireframes, v češtině by šlo volně přeložit jako náčrty rozvržení, jsou náčrtové vizualizace rozvržení stránek nebo obrazovek v rámci designu webových stránek, mobilních aplikací nebo jiných digitálních produktů. Jsou to obvykle černobílé, minimalistické kresby, které zobrazují pouze základní strukturu a umístění prvků, jako jsou text, obrázky, tlačítka a další komponenty, bez detailního grafického designu. (Carrington, n.d.)

#### **4.4.2 Návrh stylů**

Další fází je vytvoření designových stylů. Je klíčové, aby design aplikace byl konzistentní a všechny její prvky měly jednotný vzhled. V této fázi se zaměřujeme na návrh následujících komponent:

- **Výběr písma pro text v aplikaci:** Určuje typ a velikost písma používaného v různých částech aplikace.
- **Barevné schéma:** Definuje hlavní barvy aplikace, které budou použity v celém uživatelském rozhraní.
- **Tlačítka:** Navrhuje vzhled a funkce tlačítek, včetně jejich barvy, tvaru a umístění.
- **Pokyny pro rozestupy a pozicování:** Stanovujeme, jak budou jednotlivé prvky aplikace umístěny a jaký bude mezi nimi rozestup.
- **Widgety:** Design vlastních widgetů, které budou použity pro zobrazování informací nebo interakci.
- **Další důležité prvky:** Zaměření také na další klíčové prvky designu, které jsou důležité pro celkový vzhled a funkčnost aplikace. (Carrington, n.d.)

#### **4.4.3 Detailní Návrhy**

Detailní návrhy, známé jako mockupy, jsou výrazně vizuálně podrobnější než náčrty zobrazení a zahrnují konkrétní prvky, jako jsou barevné schéma, typografie, ikony, tlačítka a další grafické komponenty. Tyto návrhy demonstrují, jak budou vypadat a jak budou organizovány různé prvky aplikace. Hlavním cílem mockupů je nabídnout

jasnou vizualizaci finálního vzhledu a atmosféry produktu, aby bylo zřejmé, jak se produkt bude prezentovat uživatelům. (Carrington, n.d.)

#### **4.4.4 Vývoj Prototypu**

Posledním krokem ve vývoji designu je vytvoření prototypu. Prototypy jsou vytvářeny s cílem simulovat skutečný uživatelský zážitek a funkčnost aplikace. K vytvoření těchto prototypů lze použít nástroje jako Adobe XD, Sketch nebo Figma. (Carrington, n.d.)

### **4.5 Programování aplikace**

V této části se dostáváme k samotnému programování aplikace. Standardně se programování aplikace skládá z dvou částí, kterými jsou frontend a backend. (Carrington, n.d.)

#### **4.5.1 Backend**

Backend je pro běžného uživatele neviditelná část představující serverovou stranu, kde dochází k zpracování a ukládání dat. Hlavními funkcemi backendu jsou správa dat, zajištění autorizace a autentizace uživatelů. (Carrington, n.d.)

#### **4.5.2 Frontend**

Frontend je část aplikace, se kterou přímo interaguje uživatel. Existují tři základní způsoby vývoje frontendu aplikací: vývoj pro konkrétní platformu, multiplatformní vývoj a hybridní vývoj. (Carrington, n.d.)

#### **4.5.3 Vývoj pro konkrétní Platformu**

Při tomto způsobu se vytvářejí aplikace speciálně pro danou platformu, a to s využitím nativních programovacích jazyků, jako je Swift pro iOS nebo Kotlin a Java pro Android. Tento přístup umožňuje aplikaci využívat plně všech funkcí dané platformy. (Carrington, n.d.)

#### 4.5.4 Vývoj pro více platforem zároveň (Cross-Platform)

Způsob známý také jako cross-platform app development umožňuje vývoj mobilních aplikací pomocí jednoho kódu pro více platforem současně. Pro tento účel se používají nástroje jako React Native nebo Flutter. (Carrington, n.d.)

#### 4.5.5 Hybridní vývoj

Hybridní vývoj zahrnuje prvky nativních i webových aplikací. Aplikace jsou vytvořeny pomocí webových technologií (HTML, CSS, JavaScript). Aby bylo možné jejich spuštění na mobilních zařízeních, jsou zabaleny do nativního kontejneru. Pro hybridní vývoj se často používají nástroje Ionic nebo Xamarin. (Carrington, n.d.)

### 4.6 Testování aplikace

Aplikace byla úspěšně naprogramována, což značí, že je čas přejít k fázi testování. Tento krok je nezbytný, protože testování odhaluje, jak dobře je aplikace napsána a zda splňuje všechny požadované funkce a specifikace. Tento proces pomáhá identifikovat a opravit chyby, zlepšit uživatelský zážitek a zajišťuje, že aplikace bude fungovat hladce a efektivně. Při testování aplikace by měly být zohledněny následující klíčové aspekty:

- **Funkcionalita aplikace:** Testování zaměřené na ověření, zda aplikace splňuje všechny stanovené funkční požadavky a zda správně vykonává všechny své úkoly a procesy.
- **Výkon aplikace (Performance):** Prověření rychlosti, odezvy a stability aplikace při různých zátěžích a podmínkách používání.
- **Zabezpečení:** Kontrola aplikace z hlediska bezpečnosti, což zahrnuje ochranu před externími hrozbami, šifrování dat a prevenci proti zneužití uživatelských informací.
- **Kompatibilita s různými platformami a zařízeními:** Ověření, že aplikace správně a efektivně funguje na různých operačních systémech, zařízeních a v různých prohlížečích.
- **Rozšířené hodnocení:** Tento proces zahrnuje testování uživatelského rozhraní, použitelnosti, přístupnosti a zkušeností, aby se zajistilo, že aplikace je intuitivní, snadno použitelná a přístupná pro široké spektrum uživatelů. (Carrington, n.d.)



#### **4.6.1 Manuální vs automatické testování**

Testování aplikací lze provádět dvěma základními způsoby, a to manuálním a automatickým.

#### **4.6.2 Manuální testování**

Manuální testování je proces, při kterém testovací pracovníci přímo interagují s mobilní aplikací a hodnotí její funkcionalitu a další aspekty testování. V průběhu tohoto testování se aplikace zkoumá na různých zařízeních a s různými operačními systémy. Cílem je ověřit, zda aplikace pracuje podle předpokladů, identifikovat a reportovat všechny zjištěné chyby a posoudit celkový uživatelský dojem. Manuální testování vyžaduje více času a nemusí být schopné otestovat tolik scénářů jako automatizované testování. Vzhledem k trhu s mobilními zařízeními, je doporučené otestovat na 10 až 20 zařízení. (Manual vs. Automated Testing For Mobile Apps: Which Do You Need?, 2023)

#### **4.6.3 Automatické testování**

Automatizované testování obvykle znamená použití automatizovaného nástroje, který ovládá uživatelské rozhraní aplikace a kontroluje očekávané výsledky. Automatizované testování vyžaduje méně času než manuální. Čím větší je aplikace, tím více času ušetří automatizované testování.

(Manual vs. Automated Testing For Mobile Apps: Which Do You Need?, 2023)

Existuje mnoho aplikací určených pro testování mobilních aplikací, které lze kategorizovat podle typu podporovaného operačního systému. Například Appium je univerzální nástroj vhodný pro všechny typy mobilních aplikací (nativních, hybridních i cross-platform) a oba operační systémy (iOS i Android). Další oblíbené nástroje

pro testování zahrnují XCUITest specificky pro iOS a Selendroid pro Android. (Chowdhury, 2023)

### **4.7 Distribuce aplikace**

Po dokončení testovací fáze může následovat fáze distribuce aplikace. Aplikace se obvykle distribuují do dvou hlavních obchodů s aplikacemi. Aplikace pro Android jsou distribuovány do Google Play Store, zatímco aplikace pro iOS jsou distribuovány do App Store. Proces distribuce aplikací do těchto obchodů je komplexní a bude

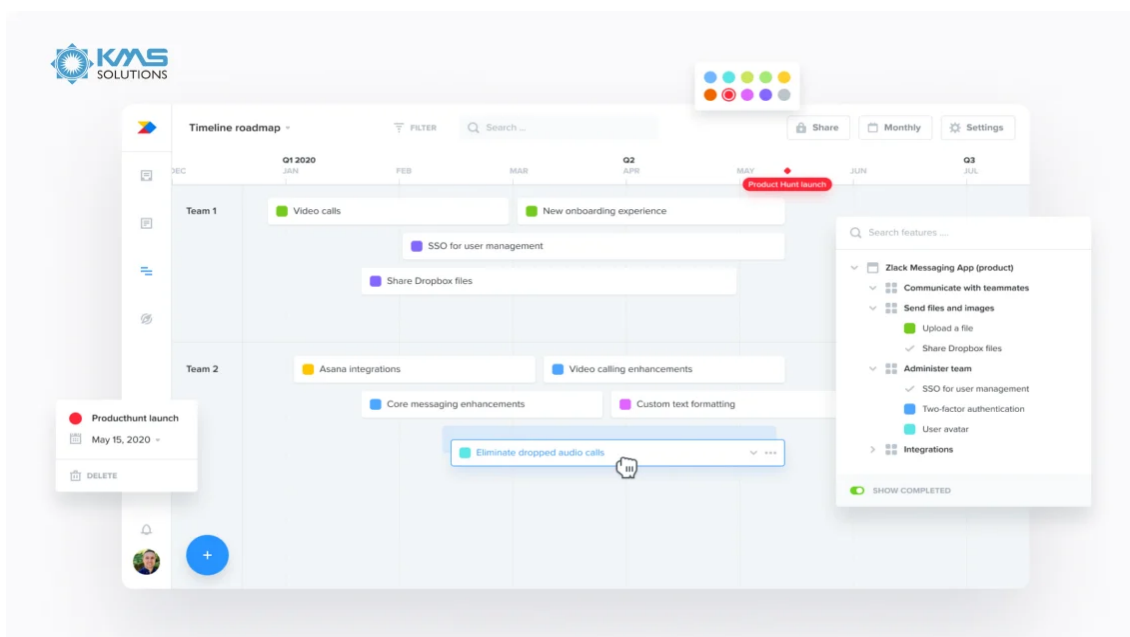
podrobněji probíráán v jednotlivých kapitolách zabývajících se vývojem aplikací pro tyto operační systémy. (Carrington, n.d.)

## 4.8 Podpora aplikace

I když by se mohlo zdát, že po distribuci aplikace je vše dokončeno, ve skutečnosti tomu tak není. Pro úspěšné fungování aplikace je nezbytné poskytovat stálou podporu. Mobilní aplikace vyžadují pravidelné aktualizace kvůli změnám v operačních systémech. Důležité je také sledovat klíčové ukazatele úspěšnosti aplikace, jako jsou:

- Počet stažení aplikace
- Aktivní uživatelé
- Průměrná doba návštěvy
- Udržení uživatelů (Retention)
- Odliv uživatelů (Churn)
- Konverze
- Hodnota zákazníka po celou dobu jeho života (Customer lifetime value)
- Hodnocení
- Recenze (Carrington, n.d.)

Obrázek 5: Časová osa funkcí



Zdroj: KMS Solutions, 2024

## **5 Vývoj a programování aplikací pro zařízení s operačním systémem iOS**

iOS představuje operační systém vyvinutý firmou Apple, který je určen výhradně pro zařízení této společnosti. Využívání iOS se primárně zaměřuje na mobilní zařízení, zejména na řadu iPhone. Jeho design a funkcionality jsou optimalizovány pro uživatelsky příjemné a intuitivní prostředí, což přispívá k celkovému uživatelskému zážitku. Jednou z klíčových vlastností iOS je vysoká úroveň bezpečnosti, která je v souladu s filozofií společnosti Apple v ochraně uživatelů. Stabilní a plynulý chod systému přispívá k celkové spolehlivosti zařízení. iOS není izolován, ale naopak je pevně zakomponován do širšího ekosystému Apple. Tato integrace umožňuje uživatelům plynule propojovat svá zařízení, což vytváří jednotné a harmonické prostředí pro práci s produkty od Apple. (Moes, What is iOS? Everything You Need to Know, n.d.)

### **5.1 Vývoj mobilních aplikací pro iOS**

Vývoj mobilních aplikací pro iOS se standardně provádí pouze na počítačích od firmy Apple. Pro tento účel se používá vývojové prostředí Xcode, které není dostupné pro počítače s operačními systémy Windows nebo Linux. (Mannotra, 2023)

Apple rovněž vyvinul programovací jazyk Swift, určený speciálně pro vývoj aplikací pro iOS. Swift je podobný jazyku Objective-C, který byl do roku 2014 standardem pro vývoj iOS aplikací. Objective-C je stále možné používat pro vývoj iOS aplikací, ale Swift je preferován díky své rychlosti, vyšší úrovni zabezpečení a jednoduššímu používání. (What is iOS app development?, n.d.)

Jediná možnost, jak vyvíjet nativní aplikaci bez vlastnictví počítače od Apple, je využití pronájmu Macu v cloudu. Tuto možnost nabízejí služby jako MacinCloud, RentYourMac nebo MacStadium.

### **5.2 Distribuce aplikace na App Store**

Vydání aplikace v App Store je komplexní proces, který vyžaduje splnění několika podmínek. První podmínkou je vlastnictví vývojářského účtu (Developer account) s ročním poplatkem 99 amerických dolarů. Dále je nutné mít nejnovější verzi vývojového prostředí Xcode. Je také potřeba vygenerovat certifikát v aplikaci iCloud Keychain.

Aplikace Transporter slouží k nahrání sestavené (build) aplikace do vývojářského účtu. Pro testování sestavení aplikace před jejím nahráním je zapotřebí aplikace TestFlight. Poslední nezbytnou věcí je Bundle ID, což je unikátní identifikátor pro aplikaci. Při nahrání aplikace na AppStore je důležité dodržovat následujících deset kroků. (Pul, 2022)

### **5.2.1 Příprava a testování**

Před odesláním aplikace se ujistěte, že je správně otestována na místních iOS zařízeních, aby se minimalizovaly potřebné změny a urychlil proces schválení. Zároveň je důležité dodržovat jak obsahové, tak technické směrnice Apple, aby byla aplikace nebo hra bez problémů schválena. (Pul, 2022)

### **5.2.2 Vytvoření iOS distribučního certifikátu**

Druhým krokem je vytvoření tzv. P.12 certifikátu, který se generuje v aplikaci iCloud Keychain a v Apple Developer účtu. (Pul, 2022)

### **5.2.3 Vytvoření App ID, testovací zařízení a provisioning účet**

Třetím krokem je vytvoření App ID, což je identifikátor určený pro aplikaci. Dále je nutné vytvořit UDID, což je identifikátor pro testovací zařízení. Po získání UDID je možné zařízení přidat do souboru testovacích zařízení. Je také důležité vytvořit takzvaný provisioning účet, který spojuje testovací zařízení s vývojářským účtem. (Pul, 2022)

### **5.2.4 App Store Connect**

Čtvrtý krok spočívá ve vytvoření aplikace v App Store Connect a přiřazení jejího správného App ID. (Pul, 2022)

### **5.2.5 Podepsání aplikace**

Pátým krokem je podepsání aplikace pomocí vytvořeného P.12 certifikátu. Následně lze získat .IPA soubor s aplikací. (Pul, 2022)

### **5.2.6 Nahrání aplikace a testování pomocí TestFlight**

V šestém kroku probíhá nahrání aplikace na Transporter a její následné testování pomocí aplikace TestFlight. (Pul, 2022)

### **5.2.7 Schvalovací proces**

Nyní nastupuje fáze schvalovacího procesu. Tým společnosti Apple se věnuje schválení nové aplikace, přičemž obvykle odpovídají do dvou pracovních dnů. (Pul, 2022)

### **5.2.8 Oprava chyb**

Tento krok je relevantní pouze v případě, že aplikace nezískala schválení. Je potřeba napravit chyby a znovu podat žádost o schválení. (Pul, 2022)

### **5.2.9 Stažení aplikace**

Po úspěšném absolvování schvalovacího procesu je aplikace dostupná v App Store ke stažení. (Pul, 2022)

### **5.2.10 Aktualizace aplikace**

Je zásadní pravidelně aktualizovat aplikaci, aby byla stále aktuální. Každá aktualizace musí projít schvalovacím procesem týmu Apple, stejně jako v kroku 7. (Pul, 2022)

## **6 Vývoj a programování aplikací pro zařízení s operačním systémem Android**

Android, operační systém vytvořený firmou Android Inc. založenou v roce 2003 v Kalifornii, byl v roce 2005 zakoupen společností Google, čímž se stal její dceřinou společností. Android, používaný různými značkami mobilních telefonů jako jsou Samsung nebo Xiaomi, je založen na linuxovém jádru. Na rozdíl od iOS je Android open source, což umožňuje vývojářům a programátorům upravovat operační systém dle svých potřeb, což je klíčovým rysem Androidu. Příklady upravených verzí Androidu zahrnují MIUI od Xiaomi a One UI od Samsungu. (Moes, What is Android? Everything You Need to Know, n.d.)

### **6.1 Vývoj mobilních aplikací pro android**

Pro vývoj mobilních aplikací pro Android se používají dva hlavní programovací jazyky: Kotlin a Java. Ačkoli Java byla dříve považována za oficiální a široce používaný jazyk, skutečným oficiálním jazykem pro vývoj Android aplikací se stal Kotlin, který byl Googlem představen v roce 2017. Android aplikace je možné vyvíjet v různých vývojových prostředích, přičemž oficiálním prostředím je Android Studio, uvedené na trh v roce 2013. Dalšími používanými vývojovými prostředími jsou Eclipse nebo Visual Studio. (How to make an Android app, n.d.)

### **6.2 Distribuce aplikace na Google play**

Pro zveřejnění aplikace na Google Play je nejprve nutné zakoupit Google Play Developer Console za cenu 25 amerických dolarů. Pokud aplikace obsahuje nákupy uvnitř aplikace, je třeba vytvořit tzv. Merchant účet a propojit jej s vývojářskou konzolí. Dále je potřeba vytvořit novou aplikaci a vyplnit informace o ní. Poté můžeme nahrát soubory aplikace. Je také důležité vyplnit hodnocení obsahu aplikace, neboť bez toho může být aplikace z Google Play odstraněna. Před publikováním aplikace je nutné určit region, kde bude aplikace dostupná, a stanovit její cenu. Aplikaci lze zveřejnit zdarma, ale poté již nelze změnit na placenou. S cenou placené aplikace je možné dále manipulovat. (How to make an Android app, n.d.)

## 7 Cross-platform a hybridní app development

### 7.1 Xamarin

Xamarin je open-source platforma pro tvorbu mobilních aplikací pro iOS, Android a Windows, využívající .NET. Aplikace Xamarin lze programovat na PC nebo Macu a kompilovat do nativních aplikací, jako je .apk pro Android nebo .ipa pro iOS. .NET je open-source vývojová platforma od společnosti Microsoft, která umožňuje vývojářům tvořit aplikace pro různé platformy a účely. Platforma .NET poskytuje rozsáhlý soubor nástrojů, knihoven a frameworků, které umožňují vývoj aplikací pro web, mobilní zařízení, desktop, cloud a dokonce i pro IoT zařízení. (Microsoft, 2022)

Nejznámější aplikace pro mobilní telefony, vytvořené pomocí Xamarinu, jsou UPS a Azure App. (.NET customers showcase, n.d.)

K vytváření Xamarin aplikací lze použít vývojové prostředí Visual Studio.

### 7.2 React Native

React Native je open-source framework pro vývoj mobilních aplikací, který byl vytvořen a je udržován společností Facebook. Tento framework umožňuje vývojářům psát aplikace pro iOS a Android použitím JavaScriptu a Reactu, což je populární knihovna pro vytváření uživatelských rozhraní vytvořená také Facebookem. (Budziński, n.d.)

Nejznámější aplikace pro mobilní telefony, vytvořené pomocí React Native, jsou Facebook, Microsoft Teams, Microsoft Office nebo Discord. (Who is using React Native?, n.d.)

K vytváření React Native aplikací lze použít vývojové prostředí Visual Studio. React Native aplikace lze také vytvářet v cloudové aplikaci Snack.

### 7.3 Flutter

Flutter je open-source framework, který vytvořila společnost Google. Flutter využívá jazyk Dart, který je objektově orientovaný. Umožňuje vytvářet aplikace pro iOS, Android, počítače nebo webové aplikace. (Amazon Web Services, n.d.)

Nejznámější aplikace pro mobilní telefony, vytvořené pomocí Flutter, jsou Google Classroom nebo mobilní aplikace BMW. (Flutter apps in production, n.d.)

K vytváření Flutter aplikací lze použít vývojové prostředí Visual Studio Code, Android Studio a IntelliJ IDEA.



## 8 Vývojová prostředí pro vývoj mobilních aplikací

Vývojová prostředí pro tvorbu mobilních aplikací, označovaná jako IDE (Integrated Development Environment), jsou softwarové aplikace, které poskytují programátorům základní nástroje pro vývoj softwaru. Mezi klíčové nástroje každého IDE patří:

- Textový editor – slouží k psaní a úpravě zdrojového kódu.
- Debugger – umožňuje testování a hledání chyb v kódu.
- Kompilátor – převádí napsaný kód do strojového kódu, který je zpracováván procesorem
- Vývojových prostředí je mnoho, zde jsou některé populární volby pro vývoj mobilních aplikací. (Khalid, n.d.)

### 8.1 Xcode

Xcode je vývojové prostředí vytvořené společností Apple, které je k dispozici pouze pro počítače od této společnosti. Umožňuje vývoj aplikací pro různá zařízení Apple, jako jsou mobilní telefony, tablety, počítače, chytré hodinky a televize. Kromě programovacích jazyků Swift a Objective-C, které jsou primárně používány pro vývoj aplikací pro Apple, Xcode podporuje také jazyky jako Python nebo Java. (Mannotra, 2023)

### 8.2 Android studio

Android Studio je oficiální integrované vývojové prostředí pro vývoj aplikací pro Android. Je dostupné zdarma pro všechny hlavní operační systémy počítačů, včetně Windows, Linuxu a macOS. Android Studio je založeno na IntelliJ IDEA, což je jiné populární vývojové prostředí. Podporuje programovací jazyky Java a Kotlin, které jsou používány pro vývoj aplikací pro Android. (TOPS Technologies, n.d.)

### 8.3 Visual studio

Visual Studio je programovací prostředí vytvořené firmou Microsoft. Napsané v C++ a C#, podporuje různé programovací jazyky například C, C++, C#, .NET, JavaScript a další. Umožňuje vývoj mobilních aplikací pro iOS a Android, desktopových aplikací pro Windows, webových stránek a webových aplikací. Zatímco individuální uživatelé

mohou využívat Visual Studio zdarma, pro firemní použití je nutná placená verze.  
(What is Microsoft Visual Studio? Where can I download it?, n.d.)

## 9 Metodika

### 9.1 Výběr tématu mobilní aplikace

Pro práci bylo vybráno téma Správa vozového parku. Navržená aplikace bude sloužit k snadnému a efektivnímu řízení vozidel uživatele. Bude umožňovat uchovávání klíčových informací, jako jsou registrační značky (SPZ) nebo identifikační čísla vozidel (VIN). Kromě toho bude aplikace upozorňovat na důležité termíny, jako je datum technické kontroly, pravidelné servisní prohlídky nebo expirace dálniční známky. Tyto informace jsou sice dostupné samostatně, ale naše aplikace je sjednotí na jedno místo, což usnadní správu vozového parku. Průzkum trhu neodhalil žádnou dostupnou aplikaci s podobnými funkcemi.

### 9.2 Výběr technologie vývoje mobilní aplikace

Je klíčové, aby aplikace byla dostupná na co nejvíce zařízeních, a proto byla zvolena technologie cross-platform vývoje. To umožní vytvořit aplikaci pro obě hlavní platformy, Android i iOS, pomocí jednoho zdrojového kódu. Vývoj pro každou platformu samostatně by vedl k nutnosti vytvořit aplikaci dvakrát nebo vybrat pouze jednu platformu, což by znamenalo ztrátu potenciálního uživatelského základu. Z možností cross-platform vývoje byla zvolena technologie React Native. Tato volba je motivována zejména použitím populárního jazyka JavaScript, který je následně kompilován do nativního kódu pro obě platformy, a také díky rozsáhlému portfoliu aplikací, které již byly úspěšně vytvořeny pomocí této technologie. Jako vývojové prostředí bylo vybráno Android Studio. Toto vývojové prostředí bylo zvoleno především díky Android emulátoru, který umožňuje testování aplikace, a také proto, že je k dispozici zdarma ke stažení.

### 9.3 Proces vývoje mobilní aplikace

Proces vývoje mobilní aplikace bude řízen podle kapitoly 4. První fáze vývoje, jako je průzkum trhu, výběr platformy a identifikace cílů, již částečně proběhla. Následující kroky zahrnují fáze analýzy a plánování, které zahrnují například návrh funkcionalit a definici entit. Poté bude vytvořen produktový plán, který bude sloužit jako základ pro další postup. Design aplikace bude následující fází, která zahrnuje první náčrty

aplikace. Celkový design bude vytvořen při samotném programování aplikace. Když jsou hotové náčrty aplikace, které udávají základní styl aplikace, je možné přejít do fáze programování. Po dokončení aplikace následuje fáze testování. Po úspěšném testování bude aplikace distribuována do obchodů s aplikacemi a následně bude monitorována.

## **10 Vývoj strategie**

### **10.1 Strategie vývoje a definice cílů**

Hlavním cílem aplikace je poskytnout řešení pro uživatele, kteří potřebují efektivně spravovat informace o svých vozidlech, zejména v případě vlastnictví více automobilů. Aplikace usnadňuje sledování klíčových údajů jako jsou registrační čísla, VIN kódy, data technických kontrol a platnost dálničních známek. Centralizace těchto informací na jednom místě a jejich snadný přístup přes mobilní telefon, včetně upozornění na důležité termíny, představuje primární cíl aplikace.

### **10.2 Průzkum trhu**

Na základě průzkumu trhu bylo zjištěno, že neexistuje aplikace s podobným zaměřením. Ačkoliv Ministerstvo dopravy České republiky od roku 2024 provozuje web, na kterém lze nalézt informace obsažené v technickém průkazu vozidla, tento zdroj pokrývá pouze omezený rozsah informací. Pro přístup k datům je nutné zadat VIN, číslo technického průkazu, nebo číslo osvědčení o registraci vozidel.

### **10.3 Výběr platformy**

V současné době dominují na trhu dva hlavní mobilní operační systémy, což vyžaduje, aby aplikace byla dostupná pro oba a aby dosáhla co nejširšího spektra uživatelů. Proto bylo rozhodnuto využít cross-platformového vývojového nástroje, umožňujícího tvorbu aplikací pro obě platformy z jedné zdrojové kódu, což je efektivnější než vytvářet samostatnou aplikaci pro každou platformu.

### **10.4 Monetizace aplikace**

Bylo rozhodnuto, že aplikace bude uživatelům nabízena zdarma. Zkušenosti ukazují, že přímá monetizace prostřednictvím jednorázových poplatků může negativně ovlivnit počet stažení, jelikož mnoho uživatelů dává přednost bezplatným aplikacím. Alternativní monetizační modely, jako je omezení počtu vozidel, která lze v aplikaci zdarma spravovat, následované poplatkem za přidání dalších, byly zamítnuty jako nevhodné z důvodu nesouladu s cílem poskytovat uživatelům plný a neomezený přístup k funkcím aplikace.

# 11 Analýza a plánování

## 11.1 Funkcionality aplikace Vozový park

Hlavní funkcionalitou aplikace je přidání vozidla, což znamená uložení informací o vozidle do aplikace. Aplikace bude ukládat značku, model, SPZ, VIN, datum technické kontroly, datum servisní kontroly a datum konce platnosti dálniční známky.

Další funkcionality aplikace:

- Zobrazení seznamu uložených vozidel
- Zobrazení informací o vozidle
- Upravení informací o vozidle
- Odstranění vozidla

Nefunkční požadavky aplikace:

- Bezpečnost
- Uživatelská přívětivost
- Kompatibilita
- Spolehlivost

## 11.2 Definice entit

Pro aplikaci definujeme tři entitní množiny. První entitní množinou je množina Auta. Tato entita zastupuje informace o jednotlivých automobilech, s následujícími atributy pro ukládání specifických dat:

- ID\_auta (číslo): Jednoznačný identifikátor pro každé auto. Slouží jako primární klíč v databázové tabulce.
- Značka (řetězec): Umožňuje uložení značky (výrobce) automobilu.
- Model (řetězec): Slouží pro specifikaci modelu automobilu.
- SPZ (řetězec): Státní poznávací značka vozidla.
- VIN (řetězec): Globálně unikátní identifikátor, který obsahuje důležité informace o vozidle, jako je výrobce, místo výroby a sériové číslo.
- Datum technické kontroly (řetězec): Datum poslední provedené technické kontroly vozidla.
- Datum servisní kontroly (řetězec): Datum poslední servisní kontroly.

- Datum konce platnosti dálniční známky (řetězec): Označuje datum, kdy skončí platnost aktuálně zakoupené dálniční známky.

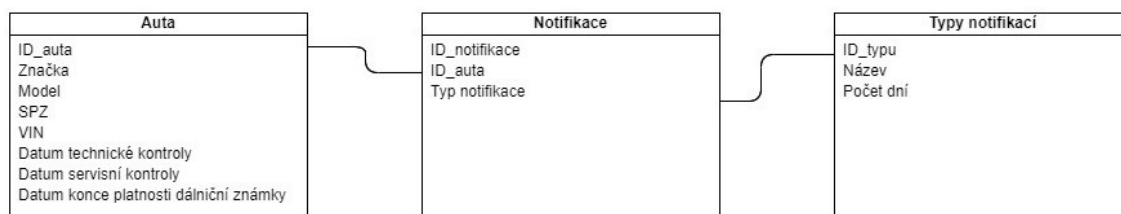
Druhá entitní množina se nazývá Notifikace a slouží k ukládání identifikátorů notifikací. Toto usnadňuje zejména zrušení starých notifikací a plánování nových. Entitní množina Notifikace obsahuje následující atributy:

- ID\_notifikace (řetězec): Unikátní identifikátor pro každou notifikaci, který je automaticky generován při vytvoření. Tento identifikátor může obsahovat jak číslce, tak znaky.
- ID\_vozidla (číslo): Cizí klíč pro propojení s tabulkou Auta.
- Typ (číslo): Cizí klíč pro propojení s tabulkou Typy notifikací.

Poslední entitní množina se jmenuje Typy notifikací a zahrnuje následující atributy:

- ID\_typu (číslo): Jednoznačný identifikátor pro každý typ notifikace.
- Název (řetězec): Popisuje druh notifikace, například servisní kontrola, technická kontrola nebo dálniční známka.
- Počet dní (číslo): Určuje, zda notifikace přichází 7 nebo 30 dní před termínem.

Obrázek 6: ER diagram



Zdroj: Vlastní zpracování, 2024

Vztahy mezi entitami jsou následující:

Mezi entitou Auto a Notifikace existuje vztah typu 1:N. Každá notifikace je přiřazena právě jednomu autu, zatímco jedno auto může mít přiděleno několik notifikací. V této konkrétní aplikaci má každé vozidlo přiřazeno šest notifikací. Mezi entitami Notifikace a Typy notifikací je také vztah typu N:1. Každá notifikace je přiřazena k jednomu konkrétnímu typu, ale jeden typ notifikace může být spojen s více notifikacemi. Každá notifikace musí mít přiřazen typ.

## 12 Produktový plán

Produktový plán pro tuto aplikaci obsahuje jednoduchý seznam kroků, které je potřeba postupně realizovat pro vývoj aplikace, aniž by bylo stanoveno konkrétní časové rozvržení. Plán vypadá následovně:

- Instalace a konfigurace vývojového prostředí
- Design aplikace
- Vytvoření stránek (aktivit) aplikace a zajištění navigace mezi nimi.
  - Tento krok zahrnuje vytvoření všech stránek aplikace, včetně všech komponent.
- Vytvoření databáze a vytvoření spojení s databází
- Implementaci obsluhy tlačítek
  - V tomto kroku bude vytvořena obsluha všech tlačítek, aby správně pracovali.
- Kompletní testování funkčnosti aplikace
  - Testování aplikace. V případě zjištěných nedostatků, přepracování aplikace a opětovné testování.
- Publikaci aplikace v obchodě s aplikacemi

K vizualizaci toho plánu byl použit MS Excel.

Tabulka 1: Produktový plán

	Dokončeno
Instalaci a konfiguraci vývojového prostředí.	ANO
Design aplikace	ANO
Vytvoření stránek (aktivit) aplikace a zajištění navigace mezi nimi.	NE
Vytvoření databáze a vytvoření spojení s databází	NE
Implementaci obsluhy tlačítek	NE
Kompletní testování funkčnosti aplikace.	NE
Publikaci aplikace v obchodě s aplikacemi.	NE

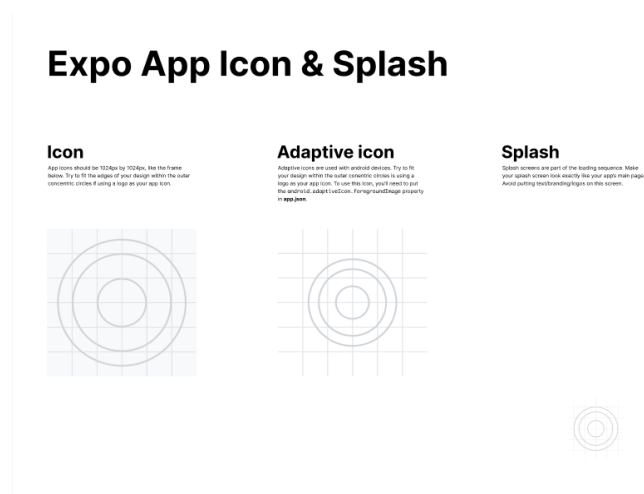
Zdroj: Vlastní zpracování, 2024



## 13 Design aplikace

Design aplikace se vzhledem k jeho nízkým požadavkům na vizuální stránku omezí pouze na návrhy provedené na papíře. Tyto návrhy slouží jako základ pro finální design, který bude vyvinut během fáze programování. Ikona aplikace a úvodní obrazovka jsou navrženy v nástroji Figma, kde je k dispozici speciální šablona určená pro tyto účely. Tato šablona byla připravena vývojáři Expa.

Obrázek 7: Šablona grafického návrhu

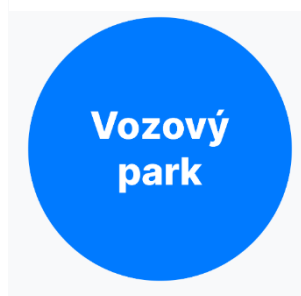


Zdroj: Figma, 2024

### 13.1 Logo aplikace

Logo aplikace bylo navrženo s důrazem na minimalismus. Skládá se pouze z modrého kruhu, který slouží jako pozadí, a v něm je umístěn text „Vozový park“. Toto logo slouží pro všechny tři druhy ikon v Figma šabloně.

Obrázek 8: Logo aplikace



Zdroj: Vlastní zpracování, 2024

## 14 Programování aplikace

Jak již bylo zmíněno aplikace bude programována pomocí Expo, což je framework, který je nadstavbou React Native. Expo umožňuje rychlejší a snadnější tvorbu React Native aplikací.

### 14.1 Příprava počítače pro vývoj

Pro zahájení práce je nejdříve nutné nainstalovat Node.js. Aplikace vytvořené pomocí Expo jsou psány v JavaScriptu a Node.js zde slouží k vytvoření serverového prostředí na počítači, které umožňuje spustit JavaScriptový kód.

Dalším krokem je instalace Windows Terminálu, který je používán pro vytváření projektů a instalaci React balíčků obsahujících různé funkce a komponenty pro použití v aplikaci.

#### 14.1.1 Vytvoření projektu

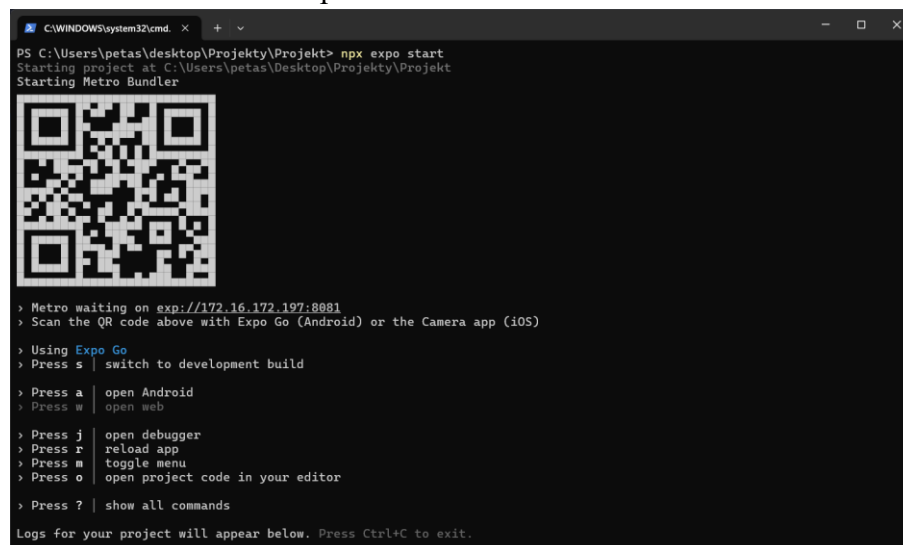
Pro vytvoření projektu je potřebné otevřít Windows terminál a přesunout se do složky, ve které má být projekt vytvořen. Projekt bude vytvořen po napsání následujícího příkazu:

```
npx create-expo-app „Název aplikace“
```

Po úspěšném vytvoření projektu, stačí přejít do složky vytvořeného projektu a napsat následující příkaz:

```
npx expo start
```

Obrázek 9: Terminál Expo cli



Zdroj: Vlastní zpracování, 2024

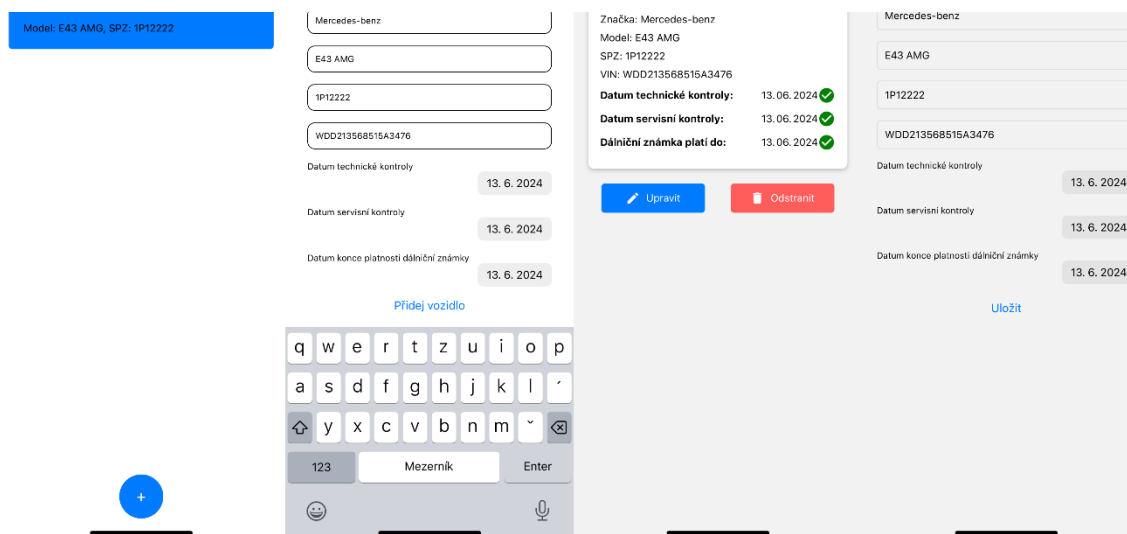
Po úspěšné instalaci a spuštění nového projektu se objeví obrazovka s náhledem základních serverových příkazů. Součástí této obrazovky je i QR kód, který je určen pro aplikaci Expo Go. Expo Go je nástroj pro testování aplikací na skutečných zařízeních a je dostupný zdarma v App Store pro uživatele iOS a v Google Play pro uživatele Androidu. Tato metoda testování je často preferována před použitím emulátoru, protože poskytuje realističtější zkušenost s aplikací. Hlavní nevýhodou je však nutnost mít fyzické zařízení k dispozici. Na druhou stranu, emulátor umožňuje testování na široké škále zařízení bez nutnosti je fyzicky vlastnit. Při vývoji aplikací pro iOS na Windows je Expo Go často jedinou možností, jak aplikaci testovat, protože emulátory iOS nejsou na této platformě snadno dostupné. Ke spuštění aplikace na reálném zařízení s využitím aplikace Expo Go je potřeba jednoduše naskenovat QR kód, který se zobrazí na obrazovce po spuštění projektu. U zařízení s iOS se k tomu využívá integrovaná aplikace fotoaparátu, zatímco u Androidu se skenování provádí přímo v aplikaci Expo Go, a následně se aplikace automaticky spustí.

Posledním krokem je instalace vývojového prostředí. Pro vývoj aplikací pro iOS je možné využít Xcode, oficiální vývojové prostředí od Apple. Na platformě Windows jsou populární různé alternativy, včetně Android Studio, které je oblíbené především díky tomu, že je dostupné zdarma. Další možností je WebStorm od JetBrains, specializovaný na JavaScript. I když použití textového editoru není povinné, výrazně usnadňuje práci s kódem. Pro tento projekt bylo zvoleno vývojové prostředí Android Studio.

## **14.2 Vytvoření aktivit aplikace**

Aplikace Vozový park se skládá ze čtyř základních obrazovek. První obrazovka zobrazuje seznam všech uložených vozidel společně s tlačítkem pro přidání nového vozidla. Druhá obrazovka je určena pro přidání vozidla, obsahuje formulář pro zadání údajů o vozidle a tlačítko pro uložení těchto informací. Třetí obrazovka slouží k zobrazení detailních informací o konkrétním vozidle a nabízí tlačítka pro úpravu nebo odstranění vozidla. Tlačítko pro úpravu nás přesměruje na čtvrtou obrazovku, která umožňuje úpravu údajů o vozidle. Tato obrazovka je ve své podstatě velmi podobná obrazovce pro přidání vozidla, ale namísto tlačítka pro přidání zde najdeme tlačítko pro uložení změn.

Obrázek 10: Aktivity aplikace



Zdroj: Vlastní zpracování, 2024

### 14.3 Základní komponenty použité při tvorbě aktivit

- View: Slouží jako kontejner pro layout a uspořádání prvků aplikace.
- Text: Umožňuje zobrazit textové informace uživatelům.
- TouchableOpacity: Interaktivní komponenta, která při dotyku snižuje svou průhlednost, čímž vizuálně simuluje stisknutí tlačítka.
- FlatList: Optimalizovaná komponenta pro zobrazování dlouhých seznamů dat. V této aplikaci slouží pro zobrazení seznamu vozidel.
- Button: Standardní tlačítko, které reaguje na uživatelské interakce.
- TextInput: Slouží pro vstup textu od uživatele. Zde je komponenta využita k získání informací o vozidlech od uživatele.
- DateTimePicker: Komponenta pro výběr data, není součástí základních komponent React Native a vyžaduje instalaci pomocí příkazu: `npx expo install @react-native-community/datetimepicker`.
- ScrollView: Umožňuje obsah zobrazit v prostředí, které je možné scrollovat, pokud je obsah delší než displej.
- MaterialIcons: Knihovna ikon, která poskytuje vizuální symboly pro aplikaci. V rámci této aplikace slouží pro indikaci platnosti datumů.

## 14.4 Zpracování vstupů

V této aplikaci se nacházejí 2 formuláře vyžadující uživatelské vstupy. Vstupy jsou zabezpečeny pomocí parametrických SQL dotazů, které uživatelské vstupy zpracovávají jako parametry a nejsou do nich přímo vkládány. Tímto způsobem lze jednoduše bránit útokům jako například SQL injection.

Jediná přímá kontrola vstupů, která je ve formulářích aplikována, se týká zadávání SPZ a VIN. SPZ musí obsahovat pět až osm znaků, jež mohou být písmena nebo čísla. VIN musí obsahovat sedmnáct znaků, také buď písmena nebo čísla.

## 14.5 Vytvoření navigace

Pro úspěšnou implementaci navigace je nejprve potřebné nainstalovat následující balíčky:

```
npm install @react-navigation/native
```

```
npm install react-native-screens react-native-safe-area-context
```

```
npm install @react-navigation/stack
```

```
npm install @react-navigation/native-stack
```

## 14.6 Realizace navigace

První aktivitou, která se spouští při spuštění projektu, je App.js. V této aktivitě, která ve skutečnosti nezobrazuje žádnou stránku aplikace, bude navigace realizována. Prvním krokem je důležité importovat react navigation a navigation/native stack a také importovat všechny aktivity, které budou obsaženy v navigaci.

Příkazy pro import navigace:

```
import { NavigationContainer } from '@react-navigation/native';
```

```
import { createNativeStackNavigator } from '@react-navigation/native-stack';
```

Příkazy pro import ostatních aktivit:

```
import HomeScreen from './Screens/HomeScreen';
```

```
import AddCarScreen from './Screens/AddCarScreen';
```

```
import CarDetailScreen from './Screens/CarDetailScreen';
```

```
import EditCarScreen from './Screens/EditCarScreen.js';
```

### 14.6.1 Implementace navigace

Kód zobrazený na obrázku 11 demonstruje implementaci navigace v souboru App.js, kde jsou všechny obrazovky přidány do navigačního zásobníku. Atribut name funguje jako identifikátor pro navigaci mezi obrazovkami. Příklad použití tohoto identifikátoru je následující:

```
navigation.navigate('CarDetail', { carId: carId })
```

Tento příkaz naviguje na obrazovku CarDetail, která slouží k zobrazení detailů o vozidle. Parametr carId se předává metodě a slouží k určení, které vozidlo má být zobrazeno.

Klíčovým prvkem kódu je také „initialRouteName“, který určuje, která obrazovka se má zobrazit jako první po spuštění aplikace.

Obrázek 11: Kód navigace

```
<NavigationContainer>
  <Stack.Navigator initialRouteName='Home'>
    <Stack.Screen name='Home' component={HomeScreen} />
    <Stack.Screen name='AddCar' component={AddCarScreen} />
    <Stack.Screen name='CarDetail' component={CarDetailScreen} />
    <Stack.Screen name='EditCar' component={EditCarScreen} />
  </Stack.Navigator>
</NavigationContainer>
```

Zdroj: Vlastní zpracování, 2024

## 14.7 Vytvoření databáze

Databáze je vytvářena ve stejném místě jako navigace a tím je soubor App.js. Nejdříve musí být nainstalován balíček expo-sqlite,

```
import * as SQLite from 'expo-sqlite';
```

V kontextu aplikací využívajících SQLite, jako je tento případ, jsou zásadní tři metody, které umožňují efektivní práci s databází. První z nich, metoda openDatabase, slouží k otevření existující databáze nebo k vytvoření nové, pokud ještě neexistuje. Tato metoda vrací objekt databáze, který je poté využíván pro další operace, jako je vytváření, čtení, aktualizace, nebo mazání dat. Druhou klíčovou metodou je transaction, která je používána k zahájení databázové transakce, což umožňuje efektivně spravovat změny dat a zajišťuje

konzistenci transakcí. Poslední metodou je `executeSql`, která umožňuje provádění konkrétních SQL příkazů.

Pro vytvoření databáze byla vytvořena metoda `innitialiseDatabase`. Tato metoda obsahuje vytvoření databáze pomocí příkazu `openDatabase` a sérii tří SQL příkazů, které vytvářejí potřebné entitní množiny a relace mezi nimi.

### 14.7.1 SQL příkazy

#### Vytvoření entitních množin

Tento SQL příkaz slouží pro vytvoření entitní množiny, která slouží pro ukládání aut a informací o nich:

```
CREATE TABLE IF NOT EXISTS cars (id INTEGER PRIMARY KEY
AUTOINCREMENT, brand TEXT, model TEXT, licensePlate TEXT, vin TEXT,
technicalInspectionDate TEXT, serviceInspectionDate TEXT,
highwayStampValidityDate TEXT);
```

Pomocí tohoto příkazu je vytvořena entitní množina, která ukládá typy notifikací:

```
CREATE TABLE IF NOT EXISTS notification_types (id_type INTEGER
PRIMARY KEY AUTOINCREMENT, name TEXT, days_count INTEGER);
```

Poslední entitní množina je vytvořena pomocí tohoto příkazu, který zároveň vytváří všechny relace:

```
CREATE TABLE IF NOT EXISTS notifications (id_notification TEXT
PRIMARY KEY, car_id INTEGER, notification_type INTEGER, FOREIGN KEY
(car_id) REFERENCES cars(id), FOREIGN KEY (notification_type)
REFERENCES notification_types(id_type));
```

#### Vkládání dat

Vložení záznamu do entitní množiny notifikace:

```
INSERT INTO notifications (id_notification, car_id, notification_type) VALUES
(?, ?, ?);
```

Vložení záznamu do entitní množiny auta:

```
INSERT INTO cars (brand, model, licensePlate, vin, technicalInspectionDate,
serviceInspectionDate, highwayStampValidityDate) VALUES (?, ?, ?, ?, ?, ?, ?);
```

### Výběr dat

Výběr všech záznamů z entitní množiny auta:

```
SELECT * FROM cars;
```

Výběr konkrétní notifikace z entitní množiny notifikací podle id auta a id notifikace:

```
SELECT id_notification FROM notifications WHERE car_id = ? AND  
notification_type = ?;
```

Výběr konkrétního vozu podle jeho id:

```
SELECT * FROM cars WHERE id = ?;
```

### Aktualizace dat

Aktualizace id notifikace:

```
UPDATE notifications SET id_notification = ? WHERE car_id = ? AND  
notification_type = ?;
```

Aktualizace informací o autě:

```
UPDATE cars SET brand = ?, model = ?, licensePlate = ?, vin = ?,  
technicalInspectionDate = ?, serviceInspectionDate = ?,  
highwayStampValidityDate = ? WHERE id = ?;
```

### Mazání dat

Smazání notifikace dle id auta:

```
DELETE FROM notifications WHERE car_id = ?;
```

Smazání auta podle id auta:

```
DELETE FROM cars WHERE id = ?;
```

## **14.8 Nastavení notifikací**

Prvním krokem k nastavení notifikací je instalace balíčku pomocí následujícího příkazu:

```
import * as Notifications from 'expo-notifications';
```



Pro naplánování notifikace jsem vytvořil metodu `schedulenotifications`. Metoda má následující vstupy – id vozidla, titulek notifikace, zprávu notifikace, datum, kolik dní předem má notifikace upozornit a id podle typu notifikace. Tato metoda vytvoří notifikace podle zadaných parametrů a uloží jí do databáze. Dále je vytvořena druhá metoda, která notifikace nejdříve zruší a pak je naplánuje stejným způsobem jako metoda pro vytvoření.

Obrázek 12: Metoda pro vytvoření notifikace

```
const scheduleNotification = async (carId, notificationType, body, date, daysBefore, type) => {
  const triggerDate = new Date(date);
  triggerDate.setDate(triggerDate.getDate() - daysBefore);
  triggerDate.setHours(8, 0, 0, 0);

  const currentDate = new Date();
  let notificationId;
  if (triggerDate > currentDate) {
    notificationId = await Notifications.scheduleNotificationAsync({
      content: {
        title: notificationType,
        body,
        sound: true,
      },
      trigger: triggerDate,
    });
  } else {
    console.log("Datum notifikace je v minulosti nebo je příliš blízko, notifikace nebude naplánována.");
    notificationId = carId;
  }

  const db = SQLite.openDatabase('MyCars.db');
  db.transaction(tx => {
    tx.executeSql(
      `INSERT INTO notifications (id_notification, car_id, notification_type) VALUES (?, ?, ?);`,
      [notificationId, carId, type],
      () => console.log("Notifikace úspěšně uložena."),
      (_, error) => console.log("Chyba při ukládání notifikace:", error)
    );
  });
};
```

Zdroj: Vlastní zpracování, 2024

## 15 Testování aplikace

Z důvodu specifik funkcionalit této aplikace nebylo považováno za nezbytné provádět automatizované testování, a tak bylo testování omezeno pouze na manuální metody. Manuální testování probíhalo na trojici typů zařízení:

Android emulátor – Tento typ zařízení je integrován do Android Studio. Při testování byl použit model:

- Google Pixel A3 s operačním systémem Android 14.0.

Fyzická zařízení s využitím aplikace Expo Go – Na těchto zařízeních byla aplikace spouštěna skrze Expo Go a zahrnovala jak Android, tak iOS platformy. Testovací zařízení zahrnovala:

- iPhone 14 Pro s iOS 17.3.1
- iPad Air (2023) s iOS 17.3.1
- iPad (2021) s iOS 17.3.1
- iPhone 12 s iOS 17.3.1
- Honor 10 s Androidem 10.0
- Samsung S22 s Androidem 14.0

Fyzická zařízení s přímo nainstalovanou aplikací – Tato kategorie zahrnuje zařízení s operačním systémem Android, na která byla aplikace nainstalována přímo. Mezi testovací zařízení patřilo:

- Honor 10 s Androidem 14.0

## 16 Publikace aplikace v App Store

Po dokončení testovací fáze je čas aplikaci zveřejnit v App Store. Prvním krokem je instalace nástroje EAS CLI, který je určen pro publikaci React a Expo aplikací do obchodů s aplikacemi. Instalace probíhá příkazem `npm install --global eas-cli`. Dále je nezbytné vlastnit Apple Developer účet, jehož cena činí 2 799 Kč.

### 16.1 Postup publikace:

- Spuštění příkazu `eas build --ios`
- Přihlášení k Apple Developer účtu
- Vytvoření identifikátoru aplikace - `com.Vozovy-park`
- V App Store Connect vytvoření nové aplikace a zadání podstatných informací:
  - Název aplikace: Vozový park
  - Popis aplikace: Aplikace pro správu vozového parku
  - Způsob monetizace: Aplikace bude zdarma
  - Vytvoření Zásad ochrany osobních údajů pomocí webu [privacypolicies.com](https://www.privacypolicies.com), kde lze jednoduše generovat zásady ochrany osobních údajů. Příklad pro tuto aplikaci je dostupný na této adrese:  
<https://www.privacypolicies.com/live/078d739e-0c4c-4028-9c6c-f7799cf5b9b8>
  - Nastavení věkového omezení: Bez omezení
  - Nahrání screenshotů z aplikace
  - Volba způsobu publikování: Aplikace Vozový park bude publikována manuálně
- Pro zahájení procesu publikování zadat v terminálu tento příkaz:
  - `eas submit -p ios --latest`

Po dokončení těchto kroků je aplikace připravena na schvalovací proces v App Store Connect. Schvalovací proces může trvat několik dní. Po jeho úspěšném dokončení je možné aplikaci publikovat v App Store.

## 16.2 První schvalovací proces

Aplikace Vozový park na první pokus neprošla schvalovacím procesem. Doba čekání na schválení při prvním pokusu byla přibližně 68 hodin a samotné schvalování trvalo přibližně 3 hodiny. Důvodem neschválení byly nevhodně zvolené screenshoty. Stačilo tedy znovu vytvořit screenshoty aplikace, nahrát je do App Store Connect a opět zahájit proces schvalování.

## 16.3 Druhý schvalovací proces

Druhý schvalovací proces byl již úspěšný a kratší. Schvalování aplikace začalo již přibližně po 13 hodinách a samotné schválení trvalo přibližně 4 hodiny. Následně mohla být aplikace publikována na App Store.

Aplikace je dostupná pod následujícím odkazem:

<https://apps.apple.com/cz/app/vozový-park/id6482855043?l=cs>

## 17 Publikace aplikace na Google Play

Proces publikace aplikace na Google Play je podobný tomu pro App Store. Nejprve je nezbytné mít nainstalovaný nástroj EAS CLI a vlastnit účet Google Play Console, jehož cena je 25 EUR. Pro vytvoření souboru .aab, který je nezbytný pro nahrání aplikace na Google Play, se použije příkaz `eas build --android`. Tento soubor je možné poté stáhnout z webové stránky Expo.

V Google Play Console je třeba založit novou aplikaci a vyplnit informace, které jsou hodně podobné jako při vytváření aplikace v App Store Connect. Pro účty, které byly nedávno vytvořeny, Google vyžaduje, aby byla aplikace testována skupinou alespoň dvaceti testerů po dobu minimálně 14 dní prostřednictvím Google Play Console. Tento požadavek brání okamžitému zveřejnění aplikace na Google Play.

Vzhledem k otevřenosti Android systému je však možné aplikaci nainstalovat přímo na jakékoli zařízení s tímto operačním systémem. Pro instalaci je potřeba .apk soubor, který lze získat konverzí z .aab souboru. K tomuto účelu lze využít konvertor, který vytvoří .apk soubor. Odkaz na tento soubor bude k dispozici následně.

[https://drive.google.com/file/d/1vktufTkMY1AYhIZrYL2MhXXENXhNvdCE/view?usp=drive\\_link](https://drive.google.com/file/d/1vktufTkMY1AYhIZrYL2MhXXENXhNvdCE/view?usp=drive_link)

## 18 Podpora Aplikace

Na první pohled by se mohlo zdát, že nahráním aplikace na App Store skončila práce vývojáře. Tomu tak určitě není. Důležité je monitorování aplikace. Je důležité monitorovat funkčnost aplikace na nových zařízeních a operačních systémech a také monitorovat hodnocení uživatelů, co se týče designu aplikace a chybějících funkcionalit, co by uživatelé ocenili.

V rámci podpory této aplikace bylo provedeno uživatelské testování pomocí dotazníku, které pomohlo celkově ohodnotit aplikaci. Uživatelské testování bylo provedeno až po distribuci aplikace do obchodu App Store. Jinak by nebylo možné zajistit uživatelské testování pro zařízení s operačním systémem iOS.

Otázky v dotazníku:

- Používaný operační systém
- Přesný typ zařízení
- Přesný operační systém
- Hodnocení vizuální stránky aplikace
- Hodnocení funkcionalit aplikace
- Narazili jste během používání aplikace na nějakou chybu?
- Pokud ano, o jakou chybu se jednalo
- Nedostatky aplikace
- Které funkcionality byste chtěli do aplikace přidat?

### 18.1 Vyhodnocení výsledků

Na dotazník odpovědělo celkem 14 uživatelů. Z toho 64,3 % používá zařízení s operačním systémem iOS (9 uživatelů). Průměrné hodnocení vizuální stránky je 8,43 bodu a je v intervalu 5 až 10 bodů. Funkcionalita aplikace byla průměrně hodnocena 8,46 bodu a hodnocení se pohybovalo v intervalu 7 až 10 bodů. Nikdo z uživatelů nenarazil na chybu ve funkčnosti aplikace. Jako nedostatky aplikace byly uvedeny například barva dlaždic na úvodní stránce či nedostatečné informace, které se týkají funkčnosti aplikace na hlavní stránce. Mezi funkcionality, které by uživatelé využili, jsou dark mode, uložení datumu záruční doby vozidla, přidání fotografií vozidla nebo

zobrazení loga značky. Toto jsou návrhy, které budou sloužit pro další vývoj této mobilní aplikace.

## Závěr

Hlavním cílem této práce bylo vyvinout mobilní aplikaci a distribuovat ji do obchodu s aplikacemi. Průzkum trhu ukázal, že neexistuje aplikace umožňující uživatelům jednoduše spravovat a uchovávat informace o jejich vozidlech.

Po zhodnocení dostupných metod vývoje byla aplikace vyvinuta pro operační systémy Android i iOS, aby byla co nejdostupnější. Pro vývoj bylo využito cross-platform řešení s použitím nástroje Expo React Native, což umožnilo vytvářet aplikaci pro obě platformy z jediného zdroje kódu.

Během vývoje byly nejprve specifikovány funkcionální a nefunkční požadavky, stejně jako definice entit. Následně byl sestaven produktový plán. Logo aplikace bylo navrženo s využitím nástroje Figma a předpřipravené šablony od vývojářů Expo. Design aplikace se vyvíjel současně s programováním. Programovací fáze zahrnovala nastavení a přípravu vývojového prostředí, instalaci Expo React Native a Android Studia. Dále byly vyvíjeny aktivity aplikace, databáze a její integrace, navigace v aplikaci a systém notifikací.

Po vývoji byla aplikace testována na různých zařízeních od emulátorů až po skutečná zařízení. iOS verze byla úspěšně schválena a nyní je dostupná v App Store. Android verzi nebylo možné zveřejnit na Google play, kvůli vysokým nárokům na testování v rámci Google Play Console, nicméně aplikace je dostupná ke stažení na odkazu, který je součástí této práce. Poslední částí této práce byla podpora aplikace. V této části bylo provedeno uživatelské testování aplikace formou dotazníku.

Hlavní cíl kvalifikační práce byl splněn pouze částečně, protože se nepodařilo nahrát aplikaci do Google Play. Všechny ostatní cíle byly splněny.



## Seznam použitých zdrojů

- .NET customers showcase.* (n.d.). Dostupné 23. ledna 2024 z Microsoft. <https://dotnet.microsoft.com/en-us/platform/customers/xamarin>
- Amazon Web Services. (n.d.). *What is Flutter?* AWS. <https://aws.amazon.com/what-is/flutter/>
- Apple in 2012: Controversy and competition.* (2012). The Washington Post. [https://www.washingtonpost.com/business/technology/apple-in-2012/2012/12/27/f4ce027c-5043-11e2-8b49-64675006147f\\_story.html](https://www.washingtonpost.com/business/technology/apple-in-2012/2012/12/27/f4ce027c-5043-11e2-8b49-64675006147f_story.html)
- Budziński, M. (n.d.). *What Is React Native? Complex Guide for 2023.* Netguru. <https://www.netguru.com/glossary/react-native>
- Carrington, M. (n.d.). *Mobile App Development Process: Ultimate Guide to Build an App.* Dostupné 23. ledna 2024 z Velvetech. <https://www.velvetech.com/blog/mobile-app-development-process/>
- Flutter apps in production.* (n.d.). Dostupné 23. ledna 2024 z Flutter. <https://flutter.dev/showcase>
- How to make an Android app.* (n.d.). IBM. <https://www.ibm.com/topics/android-app>
- CHANTEL, J. (2023). *Smartphone History: The Timeline of a Modern Marvel.* Textedly. <https://blog.textedly.com/smartphone-history-when-were-smartphones-invented>
- Chowdhury, A. R. (2023). *Best Automation Mobile Testing Tools and Frameworks.* BrowserStack. <https://www.browserstack.com/guide/mobile-application-testing-frameworks>
- Javornik, A. (2016). *The Mainstreaming of Augmented Reality: A Brief History.* Harvard Business Review. <https://hbr.org/2016/10/the-mainstreaming-of-augmented-reality-a-brief-history>
- Khalid, F. (n.d.). *What Is an IDE (Integrated Development Environment)?* DevTeam.Space. <https://www.devteam.space/blog/what-is-an-ide/>
- KMS SOLUTIONS. (2022). *6 Examples of Roadmap for Mobile App Development.* KMS SOLUTIONS. <https://blog.kms-solutions.asia/example-of-mobile-app-development-roadmap>

- Mannotra, V. (2023). *What is Xcode: Features, Installation, Uses, Advantages and Limitations*. BrowserStack. <https://www.browserstack.com/guide/what-is-xcode>
- Manual vs. Automated Testing For Mobile Apps: Which Do You Need?* (2023). SauceLabs. <https://saucelabs.com/resources/blog/mobile-testing-basics-manual-vs-automated-testing>
- Microsoft. (2022). *What is Xamarin?* Microsoft. <https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>
- Moes, T. (n.d.). *What is Android? Everything You Need to Know*. Dostupné 23. ledna 2024 z SoftwareLab. <https://softwarelab.org/blog/what-is-android/>
- Moes, T. (n.d.). *What is iOS? Everything You Need to Know*. Dostupné 23. ledna 2024 z SoftwareLab. <https://softwarelab.org/blog/what-is-ios/>
- Pour, J. (2001). *Informační systémy a elektronické podnikání*. Vysoká škola ekonomická v Praze.
- Pul, T. (2022). *A Step-by-Step Guide for iOS App Store Submission in 2023*. Headjack. <https://headjack.io/knowledge-base/submit-your-app-to-the-ios-app-store-in-2021-a-step-by-step-guide/>
- The App Store turns 10*. (2018). Apple. <https://www.apple.com/newsroom/2018/07/app-store-turns-10/>
- TOPS Technologies. (n.d.). *What is Android Studio: A Step by Step Introduction*. Dostupné 23. ledna 2024 z TOPS Technologies. <https://www.tops-int.com/blog/what-is-android-studio-a-step-by-step-introduction/>
- What is iOS app development?* (n.d.). IBM. <https://www.ibm.com/topics/ios-app-development>
- What is Microsoft Visual Studio? Where can I download it?* (n.d.). Softwarekeep. <https://softwarekeep.com/help-center/what-is-microsoft-visual-studio-where-can-i-download-it>
- Who is using React Native?* (n.d.). Dostupné 23. ledna 2024 z React Native. <https://reactnative.dev/showcase>

## **Seznam tabulek**

Tabulka 1: Produktový plán.....	40
---------------------------------	----

## Seznam obrázků

Obrázek 1: Podíl operačních systémů na trhu mobilních telefonů .....	15
Obrázek 2: Zastoupení výrobců mobilních telefonů s operačním systémem Android na trhu .....	16
Obrázek 3: Podíl operačních systémů na trhu mobilních telefonů v Severní Americe, Evropě a Asii .....	17
Obrázek 4: Podíl operačních systémů na trhu mobilních telefonů .....	18
Obrázek 5: Časová osa funkcí .....	26
Obrázek 6: ER diagram.....	39
Obrázek 7: Šablona grafického návrhu.....	41
Obrázek 8: Logo aplikace .....	41
Obrázek 9: Terminál Expo cli.....	42
Obrázek 10: Aktivity aplikace .....	44
Obrázek 11: Kód navigace.....	46
Obrázek 12: Metoda pro vytvoření notifikace .....	49

## **Seznam příloh**

**Příloha A:** Složka s celým projektem včetně zdrojových kódů

## **Abstrakt**

Špíral, P. (2024). *Vývoj aplikací pro mobilní telefony* [Bakalářská práce, Západočeská univerzita v Plzni].

**Klíčová slova:** vývoj mobilních aplikací, Expo React Native, iOS, Android

Tato práce se zabývá tvorbou mobilních aplikací. Cílem práce je vytvořit mobilní aplikaci a tu následně distribuovat do obchodů s aplikacemi. Mezi dílčí cíle patří výběr a vyhodnocení alternativ pro vývoj mobilních aplikací, provedení průzkumu trhu, návrh funkcionalit a implementace a otestování mobilní aplikace. V teoretické části se práce zabývá zejména postupem vývoje a porovnání různých alternativ vývoje mobilních aplikací. Praktická část je zaměřena na vytvoření mobilní aplikace Vozový park, která slouží pro uchování klíčových informací o vozidlech.

## **Abstract**

Špíral, P. (2024). *Development of mobile phone applications* [Bachelor Thesis, University of West Bohemia].

**Key words:** development of mobile phone applications, Expo React Native, iOS, Android

This thesis focuses on the development of mobile applications. The goal is to develop a mobile application and subsequently distribute it to application stores. The sub-goals include selecting and evaluating alternatives for mobile app development, conducting market research, designing functionalities, and implementing and testing the mobile application. The theoretical part primarily deals with the development process and the comparison of various alternatives for developing mobile applications. The practical part is dedicated to creating the "Vehicle Fleet" mobile application, which serves to store key information about vehicles.