

Vývoj mobilných aplikácií*

Lujza Šufliarska

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
xsufliarska@stuba.sk

30. september 2021

Abstrakt

V súčasnej dobe má skoro každý človek smartfón, v ktorom má množstvo aplikácií. Tieto aplikácie používa dennodenne a sú dôležitou súčasťou nie len týchto zariadení, ale aj každodenného života. Vieme, že smartfóny majú rôzne operačné systémy a medzi ne patrí aj iOS, pre smartfóny od spoločnosti Apple, a android, pre zariadenia od iných výrobcov. V mojom článku sa zameriam na vývoj mobilných aplikácií pre zariadenia so systémom iOS a android. Pozrieme sa na to, ako sa aplikácie vyvíjajú, etapy tvorby aplikácií, vývojové prostredia, front-end a back-end vývoj, UX a UI, dizajn aplikácií, wireframy a ďalšie. ...

1 Úvod

Témou tohto článku je Vývoj mobilných aplikácií. V niekoľkých kapitolách vám priblížim proces vývoja mobilných aplikácií a pozrieme sa aj na rozdiely pri vývoji aplikácií medzi systémami Android a iOS. Dozviete sa, čo je potrebné urobiť pred začiatkom vývoja, počas, ale aj potom, čo urobíte finálny produkt.

2 Mobilná aplikácia

Mobilná aplikácia je softwarová aplikácia vytvorená napríklad pre smartphony a tablety, ktorú je potrebné nainštalovať. Zariadenia sú už vopred vybavené niektorými (základnými) aplikáciami, ale oveľa viac aplikácií (vrátane tých predinštalovaných) je k dispozícii prostredníctvom aplikácie určenej pre ich šírenie – pre systém Android je to aplikácia Obchod Play a pre systém iOS App Store.

Mobilné aplikácie sú určené napr. na produktivitu, navigáciu, komunikáciu, zábavu, šport, fitness a mnoho ďalšieho. V súčasnej dobe máme aplikácie skoro na všetko. Jedným z najpopulárnejších druhov aplikácií sú sociálne siete. Týchto aplikácií je nesmierne veľa a sú tiež najpopulárnejšou oblasťou vývoja.

Načo sú nám mobilné aplikácie, keď existujú webové stránky s rovnakým obsahom? Mobilné aplikácie nám na rozdiel od web stránok ponúkajú väčšiu

*Semestrálny projekt v predmete Metódy inžinierskej práce, ak. rok 2021/22, vedenie: Vladimír Mlynarovič

interaktivitu, konkrétnejšie informácie v jednoduchej a intuitívnej podobe a okrem toho, aplikácia je menšia, prehľadnejšia, pohodlnejšia, ľahšia na prácu, ... ako webová stránka otvorená na mobilnom zariadení.

[1, 11, 16]

3 Klasifikácia aplikácií

Aplikácie vieme rozdeliť do množstva kategórii. Jednou z nich je napríklad rozdelenie podľa vývoja, ktoré je pre nás dôležité. Podľa vývoja vieme aplikácie rozdeliť na [29]:

- Webové aplikácie

Webové aplikácie sú prístupné cez internet. Tieto aplikácie nepotrebujeme stiahnuť, otvárame ich v prehliadači (Safari, Chrome, Firefox...). Fungujú však len vtedy, keď sme pripojení na internet, a preto sú najmenej praktické.

- Hybridné aplikácie

Hybridné aplikácie sú kombináciou webovej a natívnej aplikácie. Je možné ich programovať naraz pre obidve platformy. Pri zložitejších funkciách je nutné doprogramovať pre jednotlivé platformy jednotlivé súčasti samostatne. Tento druh aplikácií môže, ale nemusí vyžadovať internetové pripojenie.

- Natívne aplikácie

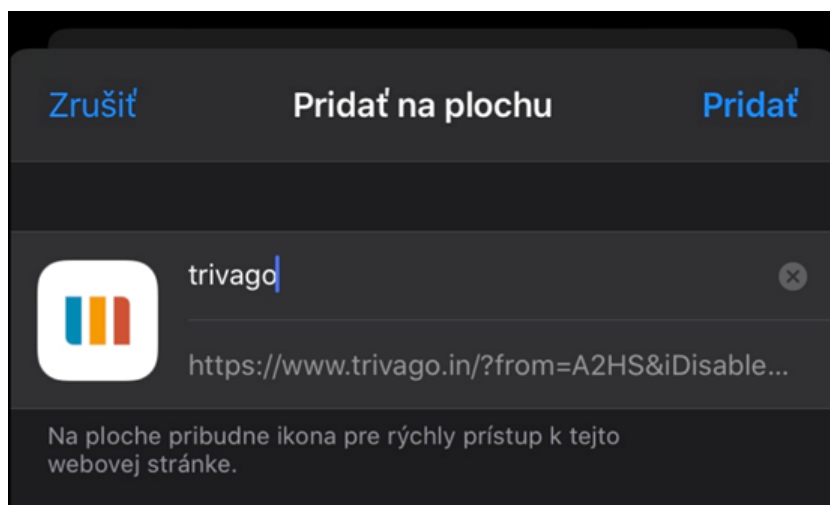
Natívne aplikácie sú aplikácie napísané v programovacích jazykoch určených pre iOS alebo pre Android a bežia priamo na operačnom systéme zariadenia. Väčšina mobilných aplikácií je natívnych. Pretože je určená primárne pre jednu platformu, tak sa tento druh aplikácie napr. rýchlejšie inštaluje, ľahšie obsluhuje, nie je závislá od internetového pripojenia, ľahko spolupracuje s funkciami zariadenia... Tento vývoj si však vyžaduje vysoké náklady.

- Cross-platformové aplikácie

Cross-platformové aplikácie sú vytvorené pre viacero platforiem súčasne. Kód je pre každú platformu skompilovaný do natívnej aplikácie. Tento vývoj môže byť efektívny a menej náročný na zdroje.

- Progresívne aplikácie

Tento typ aplikácií sa stáva čoraz populárnejším. Aplikácia sa nainštaluje do mobilného zariadenia prostredníctvom prehliadača a nie pomocou obchodu s aplikáciami.



Obr. 1: Progresívna aplikácia [16]

Ďalším veľmi známym rozdelením aplikácií je rozdelenie podľa ich účelu [29]:

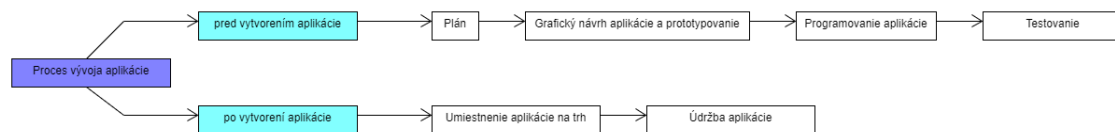
- Zábava - hry
- Pomocníci –počasie, hľadanie informácií...
- Sociálne siete
- I-komercia – nákup cez aplikácie
- ...

4 Proces vývoja aplikácie

Vývoj mobilných aplikácií je zložitý proces vytvárania aplikácií určených pre mobilné zariadenia. Toto odvetvie je stabilné a výnosné a zaisťuje množstvo pracovných pozícií [8,27].

Ako každý vývoj, tak aj vývoj mobilných aplikácií pozostáva z niekoľkých krokov, ktoré tiež závisia od nejakých činiteľov. Nikde nie je presne napísaný postup pre vývoj aplikácií. Ale takou základnou kostrou práce môžu byť tieto body, respektíve fázy [8,27]:

1. Plán
2. Grafický návrh aplikácie a prototypovanie
3. Programovanie aplikácie
4. Testovanie
5. Umiestnenie aplikácie na trh
6. Údržba aplikácie



Obr. 2: Proces vývoja aplikácie
 Zdroj: <http://www.umletino.com/>
 Autor: Lujza Šufliarska

4.1 Plán

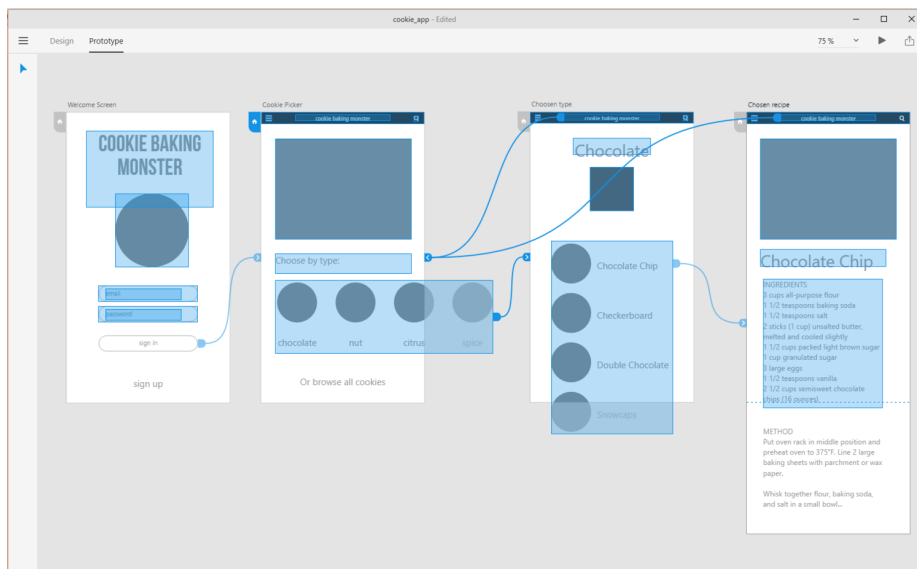
Ešte predtým ako začneme vyvíjať samotnú aplikáciu, treba poznať niekoľko odpovedí na otázky charakterizujúce predstavovanú aplikáciu. Základom k úspešnému projektu je poznať cieľ. Je dobré sa zamyslieť nad niekoľkými otázkami, ako sú napríklad [6, 23]:

- Čo je zmyslom našej aplikácie?
- Je takáto aplikácia potrebná?
- Aké problémy má aplikácia riešiť?
- Pre akú cieľovú skupinu má byť aplikácia určená?
- Aký máme budget?

4.2 Grafický návrh aplikácie

Teraz prichádza na rad dizajn, kreslenie, storyboarding a wireframing. Najprv si vytvoríme základnú predstavu aplikácie – UI (user interface – užívateľské rozhranie) a UX (user experience) dizajn (čo je UI a UX je popísané v nasledujúcich podkapitolách).

Na vytvorenie mockupov a screenshotov môžete použiť napr. Adobe XD, Adobe Photoshop, Adobe Illustrator alebo aj Sketch. Storyboard alebo wireframing je podstatnou súčasťou vývoja. Wireframe pretavuje analytickú fázu do grafickej podoby. V priemere sa nakreslí 120 – 150 obrazoviek. Všetky tieto obrazovky a stavy prvkov pomáhajú vo vizualizácii všetkých scenárov.



Obr. 3: Wireframing

Zdroj: <http://www.theresachoi.co/blog/tag/wireframes>

Po navrhnutí wireframov a dizajnu môžeme začať oživovať aplikáciu. Vytvoríme klikateľný prototyp. Opisujeme všetky prechody medzi obrazovkami a stavmi obrazovky, všetky scenáre, animácie... Na to môžeme použiť nástroje Marvel, Flint, InVision alebo Adobe XD, pomocou ktorých vytvoríme prototyp a uvidíme prvé reakcie a funkcie aplikácie.

[6, 29]

UI

User interface zahŕňa jednotlivé stránky, podstránky, tlačidlá, vizuálne prvky, ktoré pomáhajú pri komunikácii s aplikáciou alebo zariadením. Ide o navrhnutie vizuálne pútavého softvéru a ľahko navigovateľného softvéru pre všetky typy zariadení. Všetky prvky musia byť navrhnuté tak, aby ich úlohy a funkcie boli na prvý pohľad jasné. Tiež musia byť umiestnené na správnych miestach. [24]

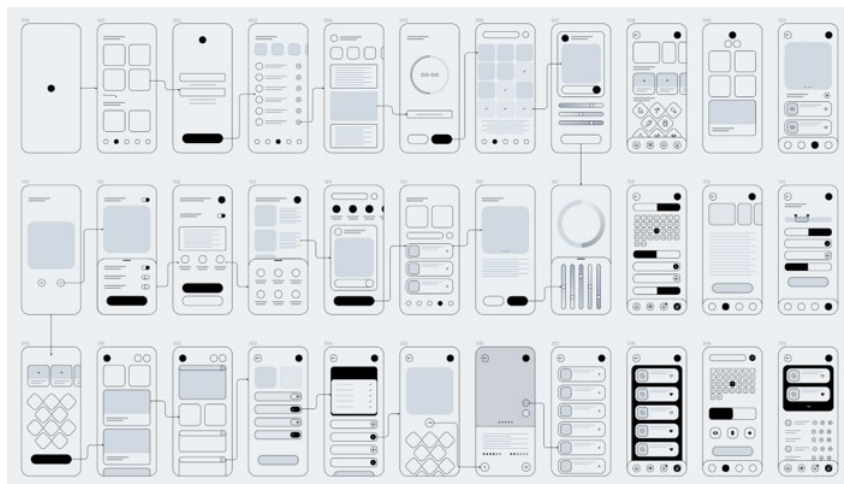
UX

User experience je zážitok, skúsenosť užívateľa po používaní aplikácie alebo webu. Chceme urobiť prepínania medzi jednotlivými podstránkami, stránkami, funkciami urobiť čo najjednoduchšie a intuitívne, aby užívateľ použil náš produkt opakovane a odporučil ho svojim známym. UX zahŕňa aj faktory ako sú dizajn, dostupnosť, marketing, použiteľnosť, výkon systému, užitočnosť... [18, 24]

Wireframe

Wireframe („drôtený model“) pomáha pri návrhu dizajnu, obsahu a funkcií. Väčšinou je to prvá vec, ktorá sa ukazuje klientovi. Prezentuje rozmiestnenie a formu projektu. Definuje textový aj grafický obsah, rozmiestnenie funkčných prvkov, navigáciu, nadpisy, kľúčový text a tlačidlá. Nie je to ale grafický návrh,

neobsahuje obrázky, nie sú dôležité farby... Pomocou neho sa vytvorí vzor pre grafikov a vývojárov a použije sa aj na diskusiu s klientom. Robiť zmeny a úpravy je oveľa jednoduchšie a rýchlejšie vo wireframe, ako v hotovom grafickom návrhu alebo naprogramovanej verzii. Wireframe je možné robiť na papier alebo aj v softvéri. [3,28]



Obr. 4: Wireframing

Zdroj: <https://www.figma.store/download/wireframes-mobile-free-ui-kit-design-for-figma/>

4.3 Programovanie aplikácie

Teraz si treba vybrať operačný systém, na ktorom bude aplikácia bežať. Môžeme si vybrať len jeden alebo môžeme vyvinúť cross-platformovú aplikáciu, ktorá najlepšou voľbou. Takto zaistíme, že aplikácia bude fungovať na každom zariadení. [3]

Väčšinou sa to dá rozdeliť na 3 časti:

- Back-end vývoj
- APIs – aplikačné programové rozhranie
- Front-end vývoj

Back-end vývoj

Back-end je vývoj potrebný zo strany servera. Používa sa na uloženie, zaistenie a spracovanie dát. Je to to, čo sa deje, keď užívateľ interaguje s aplikáciou. Dejú sa tu registrácie, prihlásenia, posielanie správ, ukladanie údajov na cloud a mnoho ďalšieho. Back-endový vývoj sa zameriava na ukladanie informácií do vzdialenej databázy, spracovanie vstupu užívateľských dát, skriptovanie na pridanie logiky do interaktivity a vytváranie architektúry, ktorá umožňuje rýchle a jednoduché triedenie informácií. Niektoré aplikácie ako kalkulačka, fotoaparát, kompas nepotrebujú back-endový vývoj, pretože na to, aby fungovali nepotrebujú internetové pripojenie ani ukladanie a obnovovanie údajov a dát zo

vzdialeného servera. Ale aplikácie ako Amazon, Netflix, Uber nemôžu fungovať bez back-endu. [18, 19]

APIs – aplikačné programové rozhranie

Application programming interface – API je rozhranie, ktoré definuje spôsoby, ktorými môže aplikácia požadovať služby od knižníc alebo operačných systémov. Aplikácie môžu medzi sebou komunikovať, spoločne pracovať a odosielať si údaje bez toho, aby sa muselo pracovať osobitne s každým programom. Teda umožňuje dvom aplikáciám nájsť sa a vzájomne spolupracovať. Vďaka nemu je rozšírenie alebo napojenie na ďalší systém jednoduché. API pomáha užívateľom rýchlo integrovať a pripojiť sa k vzdialenému cloudovému dátovému serveru. [25, 26]

Front-end vývoj

Front-end je to čo vidí používateľ. Stará sa o zobrazenie konkrétnych dát, ktoré sú aplikáciou spracované.

[19]

4.4 Testovanie

Pri testovaní si všímame či sú texty na správnych miestach, či sú funkčné všetky odkazy, tlačidlá a funkcie... Dobrými nástrojmi na testovanie sú napríklad Solidifx alebo Framer. Netreba si myslieť, že keď niečo fungovalo v predchádzajúcich verziách, že to funguje aj vo finálnej verzii. Preto sa treba vrátiť k návrhom a prejsť si všetky funkcie krok za krokom.

Funkčnosť aplikácie je nesmierne dôležitá, preto testujeme na skoro všetkých dostupných zariadeniach. Je dobré testovať aj na modeloch, ktoré dokážu odhaliť množstvo chýb. Väčšinou sú to teda problémové zariadenia, ako napríklad Samsung s5 mini.

[3, 7, 15]

4.5 Umiestnenie aplikácie na trh

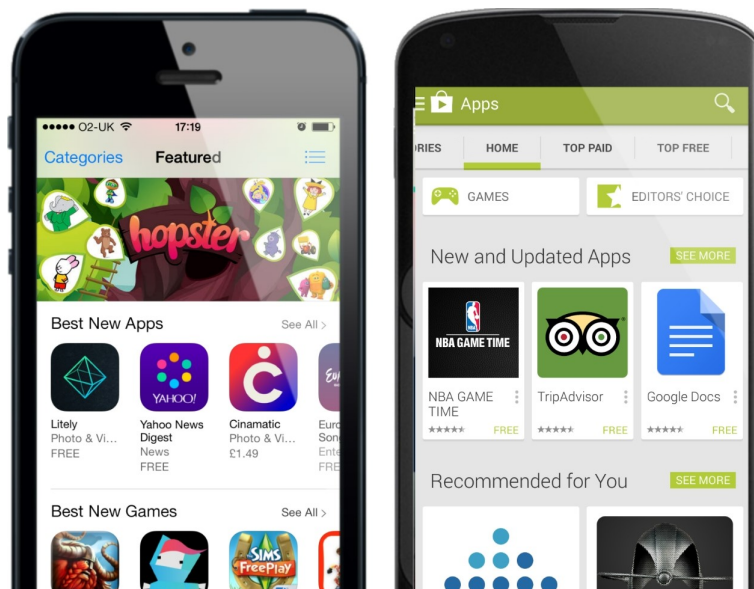
Pred publikovaním aplikácie si treba založiť účet na Google Play alebo App Store. Vytvorenie účtu môže trvať aj niekoľko dní, a preto na to netreba zabudnúť.

Pri publikácii je dôležité myslieť aj na pravidlá obchodov App Store a Google Play, ktoré sú rozdielne. Napríklad. Google hneď nekontroluje aplikáciu, ale môže to niekedy urobiť. Na druhej strane, Apple ju kontroluje hneď a môže to trvať aj 2 týždne. Existuje aplikácia PreApps, ktorá vám poskytne spätnú väzbu ešte predtým, ako svoju aplikáciu spustíte do obehu.

[7]

Ak by ste sa chceli dozvedieť viac k tomu, ako umiestniť aplikáciu na trh, odporúčam tieto zdroje (v angličtine):

- pre Android - <https://www.youtube.com/watch?v=5GHT4QtotE4> alebo <https://appinventiv.com/blog/how-to-submit-app-to-google-play-store/>
- pre iOS - <https://www.youtube.com/watch?v=YPLs3xrDcm0> alebo <https://codewithchris.com/submit-your-app-to-the-app-store/>



Obr. 5: Obchody aplikácií (App Store naľavo, Google Play napravo)

Zdroj: <https://www.thinkapple.sk/>

prijmy-app-store-su-2x-vacsie-nez-v-google-play/

4.6 Údržba aplikácie

Máte hotovú aplikáciu, ktorú používa niekoľko užívateľov. Ale tu práca nekončí. Aplikáciu treba ďalej udržiavať, optimalizovať, vylepšovať, aby bola stále aktuálna a používaná. Keby sme sa o aplikáciu nestarali, raz by sa zrútila, nefungovala a nakoniec by bola odstránená z obchodu. [17]

5 Propagácia aplikácie – pred spustením a po

Propagácia aplikácie sa dá rozdeliť na propagáciu pred spustením a po. Pred spustením aplikácie sa treba zamerať na vytvorenie záujmu o aplikáciu. Treba poskytnúť všetky dôležité informácie originálnym spracovaním. Dobrým nápadom je spolupracovať s osobami, ktoré majú široký dosah prostredníctvom sociálnych sietí. Môžu si vyskúšať aplikáciu ešte predtým, ako je sprístupnená verejnosti a spropagujú ju cez svoje sociálne siete najlepšie pozitívnou recenziou. Aplikáciu môžete propagovať aj prostredníctvom médií alebo platených článkov. Netreba zabudnúť na zaujímavý popis, informatívne obrázky v obchodoch s aplikáciami a prípadne aj propagačné video.

Po spustení aplikácie sa zamerajte na sociálne siete. Sociálne siete sú najpoužívanejším druhom aplikácií, a preto je najlepším miestom na propagáciu. Pripravte pútavé vizuály, videá a pripojte aj link na stiahnutie aplikácie. Ľudia si skôr stiahnu aplikáciu odporúčenú známymi alebo priateľmi. Toto je dobré využiť a poskytnúť odmeny za každého pozvaného užívateľa. Udržujte kontakt. Odpovedajte na otázky a dotazy v obchodoch s aplikáciami. Nezapadnite aj na negatívne komentáre, ktoré vám tiež poskytujú spätnú väzbu. Ako vidíte, je množstvo možností, ako propagovať aplikáciu pred aj po spustení. Stačí si len

vybrať, ísť s dobou, aktualizovať propagáciu a teda venovať sa tomu.
[19]



Obr. 6: Propagácia aplikácie Fitnus

Zdroj: <https://www.appfutura.com/blog/add-push-notification-for-android-and-ios-mobile-apps/>

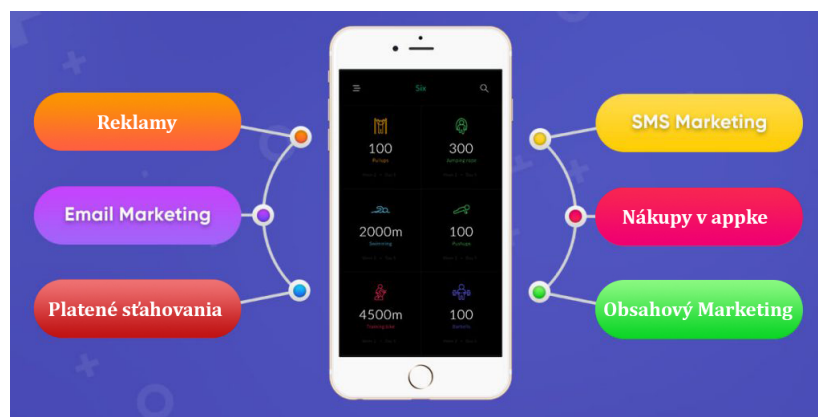
6 Platformy na vývoj aplikácií

Platformy na vývoj aplikácií sa skladajú z rôznych nástrojov a komponentov, ktoré umožňujú vývojárom písať, testovať a vydávať aplikácie. Medzi platformy na vývoj aplikácií patria napríklad [19,27]:

- **Kotlin** – Tento programovací jazyk sa stal oficiálnym jazykom pre Android. Je možné s ním pracovať napr. vo vývojárskom prostredí Android Studio.
- **Swift** – Swift je oficiálnym programovacím jazykom pre iOS aplikácie. Bol vyvinutí spoločnosťou Apple, a aby uľahčil prácu programátorom. Často je používaný XCode ako vývojárske prostredie, ktorý je tiež produktom Apple.
- **Flutter** – Umožňuje multiplatformový vývoj. Výslednú aplikáciu je možné zkompilovať do natívneho kódu pre Android aj iOS. Je to mladá technológia, vyvinutá spoločnosťou Google, ktorá používa programovací jazyk Dart 2.
- **React Native** – Hybridný framework od Facebooku, ktorý umožňuje písanie aplikácií pre Android aj iOS.
- **Apache Cordova** – Jedná sa o ďalší hybridný framework, ktorý je od spoločnosti Apache.

7 Monetizácia aplikácie

Na to, aby ste mali zisk z aplikácie, nemusí byť aplikácia hneď spoplatnená. Najrozšírenejším spôsobom zárobku sú reklamy. V aplikácii môžete poskytnúť miesta pre reklamu. Tá môže byť spoplatnená napr. na základe počtu kliknutí na ňu. Alebo môžete poskytnúť užívateľom spoplatnenú verziu bez reklám. Ďalším spôsobom je FREEMIUM model. Aplikácia je voľne dostupná, ale má obmedzené funkcie, ktoré majú používateľa tak zaujať, aby si zaplatil plnú verziu PREMIUM so všetkými funkciami. V aplikácii môžete vytvoriť špeciálne predmety, herné vychytávky, doplnkové funkcie... , ktoré si budú môcť užívatelia kúpiť za reálne peniaze. Tento spôsob tvorí až 50% obratu celého trhu. Takže aj spôsobov, ktorými môžete mať zisk z aplikácie je množstvo. O mnoho viac ako bolo spomenutých. [19]



Obr. 7: Monetizácia aplikácie

Zdroj:

<https://www.mindinventory.com/blog/app-monetization-strategies/>

Preklad: Lujza Šufliarska

8 Cena aplikácie

Ak sme si dali aplikáciu vyrobiť firme, tak nás určite zaujíma aj cena. Cena aplikácie je ovplyvňovaná viacerými faktormi. Medzi ne patrí napríklad dostupnosť aplikácie. To znamená, že na akých zariadeniach bude aplikácia dostupná. Na Android, iOS alebo oboch? Tak isto aj na akých verziách operačného systému bude dostupná. Tu treba dať pozor hlavne na iOS.

Ďalším faktorom je napr. zložitosť a rozsah funkcií. Ideálne je najprv vytvoriť aplikáciu so základnými funkciami, ku ktorým sa potom pridávajú zložitejšie funkcie, ktoré môžu vývoj predĺžiť a zdražiť.

Dizajn pri mobilných aplikáciách nie je len vzhľad aplikácie, ale aj užívateľské rozhranie (user interface – UI), ale aj funkčnosť a navigácia aplikácie, tzv. user experience – UX.

Veľkosť vývojárskeho tímu tiež ovplyvňuje cenu. Počet členov závisí od zložitosti aplikácie, od požadovaných funkcií. Väčšinou základ tvorí projektový

manažér, UX dizajnér, minimálne 2 vývojári a tester. Vo väčšom tíme už nájdeme viacej vývojárov, ktorí sú špecializovaní na určité oblasti – Android, iOS, backend...

Na čom by sa nemalo šetriť je testovanie výslednej verzie aplikácie. Pozorné testovanie môže predísť ďalším nákladom a urýchliť projekt. Ak máme projekt rozdelený na viacej fáz, je dobré po každej fáze všetko otestovať. Nemôžeme zabudnúť na dlhodobé výdavky na údržbu, úpravu aplikácie, správu dát, vytváranie updateov, poskytovanie podpory užívateľom, výdavky na propagáciu aplikácie...

[19]

9 iOS vs. Android vývoj

Počet užívateľov smartphonov rastie každým dňom a tak isto narastá aj počet vývojárov aplikácií. V súčasnosti najpoužívanějšími mobilnými operačnými systémami sú iOS, od firmy Apple a Android, od spoločnosti Google. Vývoj aplikácií pre tieto systémy je odlišný a nejaké rozdiely si povieme v tejto kapitole.

| | iOS | Android |
|---|--------------------|------------------------------|
| Programovací jazyk | Swift, objective-C | Java, Kotlin |
| IDE (vývojové prostredie) | Xcode | Android Studio |
| Cena registrácie do obchodu | \$99 ročne | \$25 jednorázovo |
| Schválenie aplikácie v obchode ¹ | V priemere 2 dni | Krátka doba (niekoľko hodín) |

Tabuľka 1: Tabuľka porovnania rozdielov medzi systémami iOS a Android vo vývoji aplikácií

Programovací jazyk

Na vývoj aplikácií pre systém Android sa používa hlavne programovací jazyk Java, vďaka ktorému je vývoj pre tento systém jednoduchší, pretože je používaný väčšinou vývojárov. Ďalšie programovacie jazyky, ktoré sa môžu použiť na vývoj Android aplikácie sú napríklad Kotlin, Flutter a React Native.

Na aplikácie pre iOS sa v súčasnej dobe používa hlavne Swift. Jeho výhodou je, že je stručný a jednoduchší ako objective-C, ktorý sa tiež používa na vývoj. Objective-C je exkluzívnejšie, čo môže byť prekážkou pre vývojárov menej kvalifikovaných v iných programovacích jazykoch. Okrem týchto jazykov je tiež možné použiť napríklad Flutter alebo React Native.

[8, 12, 18, 21]

IDE (vývojové prostredie)

Pre vývoj android aplikácií je v súčasnosti najpoužívanjšie Android Studio od Google. Umožňuje cross-platformový vývoj, ladenie aplikácie, úpravu kódu, testovanie, testovanie aplikácie v emulátore (nie je potrebné mať android zariadenie, ale treba rátať s tým, že emulátor nemusí vždy fungovať správne)... Android Studio je možné používať zadarmo na systémoch Windows, macOS aj Linux.

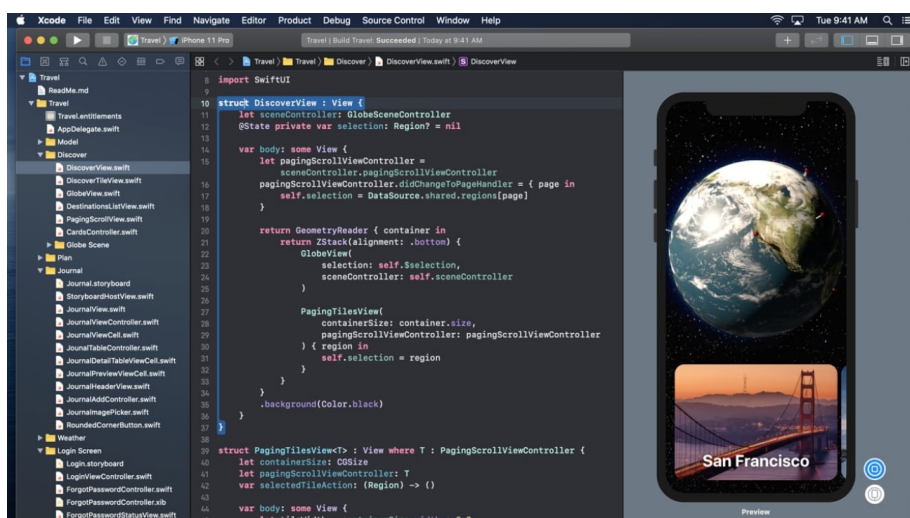
iOS vývojári používajú XCode od Apple pre vývoj iOS aplikácií. Je to rýchly a spoľahlivý softvér. Umožňuje návrh, vývoj, zápis kódu, riešenie problémov,

¹tiež spomenuté v kapitole Umiestnenie aplikácie na trh

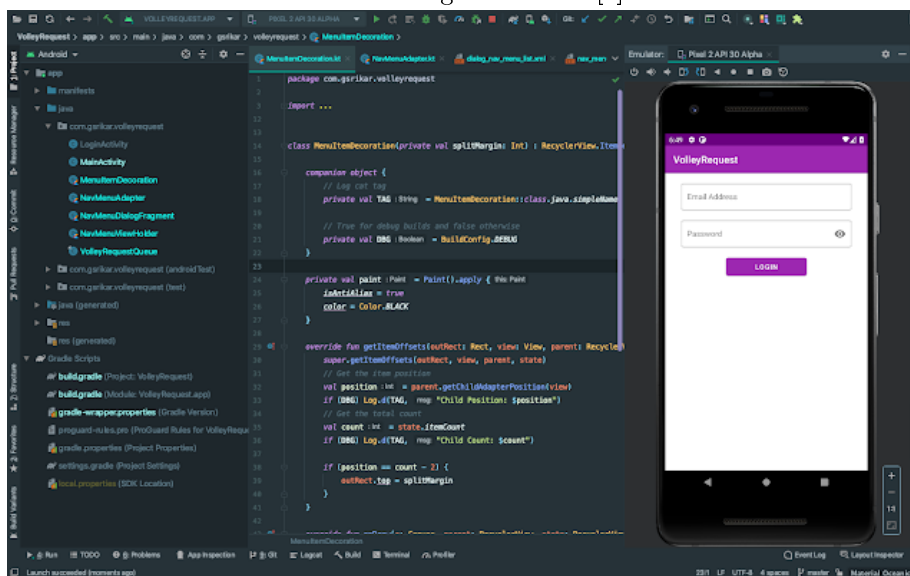
testovanie, dokáže zistiť chyby a errorry v syntaxi a logike, dokáže opraviť kód a dokonca umožňuje pridanie aplikácie priamo do obchodu. Xcode je bezplatný program, ale je možné ho používať len na notebookoch a počítačoch Apple. Sú aj nejaké iné možnosti ako vyvíjať Apple aplikácie, ktoré sú spomenuté v tomto článku - <https://codewithchris.com/xcode-for-windows/>.

[5, 14, 16]

Samozrejme neexistuje len Android Studio a Xcode. Existuje množstvo programov, v ktorých je možné vyvíjať aplikácie pre Android alebo iOS. Android Studio a Xcode sú len najpoužívanejšími aplikáciami.



Obr. 8: Program XCode [2]



Obr. 9: Program Android Studio [13]

Platba za registráciu do obchodu

Pre pridanie aplikácie do obchodu Google Play je potrebné uhradiť jednorázový registračný poplatok \$25. Tento jednorázový poplatok je za registráciu do Google Play Developer Console. [20]

Apple App Store si účtuje až \$99 ročne za to, aby mohla byť aplikácia pridaná do obchodu. Tých \$99 je za registráciu (a používanie) do programu Apple Developer Program. Tento program vám poskytne prístup k množstvu výhod, ako napríklad pridávanie aplikácií do App Storeu, pridávanie Safari rozšírení, testovacie nástroje, analýza aplikácií... [4]

[9, 16]

Schválenie aplikácie

Schválenie aplikácie v Apple App Store trvalo 3-4 týždne. V súčasnosti je to v priemere 2 dni. Trvá to tak dlho, pretože aplikáciu schvaľuje tím developerov, ktorý prechádza celou aplikáciou, aby odhalil všetky chyby a upozornil na ne autora.

Proces schvaľovania android aplikácie trvá len niekoľko hodín. Aplikáciu kontroluje tím odborníkov pomocou automatizovaných nástrojov. Tím hľadá v aplikácii porušenia, prítomnosť malwareu, spywareu, porušenie autorských práv... Po kontrole prvým tímom, aplikáciu skontroluje ešte druhý tím, ktorý aplikáciu zhodnotí. Keďže nemá Android Market takú prísnu kontrolu, na Google Play sa nachádza veľké množstvo nekvalitných aplikácií.

[10, 22]

Zhrnutie

Aj Android aj Apple majú svoje plusy aj mínusy. Je ale odporúčané vyvíjať aplikácie aj pre iOS, aj pre Android. Okrem toho, aplikácia dostupná len pre jeden operačný systém pôsobí neprofesionálne a zbytočne prichádza o množstvo potenciálnych zákazníkov.

10 Záver

Po prečítaní tohto článku sme sa dozvedeli, že vyvinúť aplikáciu nie je až také jednoduché. Je to proces, ktorý trvá nejakú dobu. Hlavne, ak chceme vyvinúť poriadnu, funkčnú a spoľahlivú aplikáciu.

Je dôležité urobiť si plán a držať sa ho. Tak si môžeme byť istý, že na nič nezabudneme. Povedali a opísali sme si pár dôležitých krokov pri vývoji, takže predstaviť si a začať vyvíjať aplikáciu by malo byť o niečo jednoduchšie. Vývoj sa dá rozdeliť na 2 časti: na vývoj pred finálnou aplikáciou a na vývoj po finálnej aplikácii. Obidve tieto časti sme si popísali. Dozvedeli sme sa aj niekoľkých možnostiach zisku z aplikácie. Nezabudli sme spomenúť, od čoho závisí cena vývoja aplikácie v prípade, že by sme ju nechceli vyvíjať sami, ale dali by sme si ju vyvinúť napríklad nejakej firme. Ďalej sme si povedali pár rozdielov vo vývoji pre systém Android a iOS. Ako ste mohli vidieť, rozdiely spočívajú hlavne v programovacom jazyku, vývojovom prostredí a ďalších aspektoch, ktoré sa týkajú hlavne fázy, kedy už máme finálnu aplikáciu.

11 Reakcia na témy

Grafické vyjadrenie informácií v informatike V 4. prednáške nám bolo poukázané na klady vizualizácie informácií pomocou grafiky. S týmto súhlasím, pretože informácie spracované graficky sú viac zaujímavejšie ako čistý text, je to prehľadnejšie, vedia tomu porozumieť aj ľudia, ktorí nehovoria daným jazykom, ľahšie a skôr si to zapamätáme... Dá sa to aplikovať aj na moju tému. Napríklad pri wireframingu, kde kreslíme všetky obrazovky aplikácie, vidíme skoro presne, ako chceme, aby aplikácia vyzerala, potom z toho vieme aj tú aplikáciu ľahšie naprogramovať. Snažila som sa aj do článku vložiť čo najviac grafického materiálu, ale nie všade sa to dalo alebo skôr hodilo.

Sedem statočných (inžinierov) Prednáška číslo 5 bola o 7 statočných inžinieroch. Bola to zaujímavá prednáška, pretože sme sa dozvedeli o informatikoch, ktorých vynálezy používame denne.

Jedným zo spomínaných informatikov bol pán Bill English, ktorý pomohol vytvoriť prvý prototyp počítačovej myši. Neviem si predstaviť presúvať sa medzi kartami webového prehliadača alebo v súbore na písanie textu bez myšky. Nie je to nemožné, je to len náročnejšie alebo menej efektívne, takže určite mi uľahčila prácu pri písaní tohto článku.

Ďalším pánom bol napríklad Larry Tesler, ktorý mi zefektívnil písanie textu z wordu do LaTeXu. On totižto vymyslel funkciu copy-paste. Podľa mňa je to asi jedna z najpoužívanejších funkcií, ktorá šetrí čas.

Načo budem inžinierom? Pozvaná prednáška s pánom profesorom Ivanom Kotuliakom bola tiež veľmi zaujímavá. Mali sme možnosť stretnúť sa a diskutovať, síce len online, s dekanom našej fakulty, ktorý nám povedal o výhodách vyštudovania inžinierskeho, respektíve bakalárskeho stupňa štúdia. Jednou z motivácií štúdia, na ktorej sme sa zhodli skoro všetci, boli peniaze. Okrem toho, že nástupné platy sa odvíjajú od vyštudovaného stupňa, tak dlhšie štúdium (v zmysle, štúdia viacerých stupňov) nás dokáže lepšie pripraviť na prácu a dá nám užitočné skúsenosti a zručnosti.

Jednou takou zručnosťou, ktorú som využila pri písaní tohto článku, je hľadanie zdrojov a materiálov. Táto zručnosť sa mi určite zíde aj v budúcnosti, nie len pri písaní nejakého textu.

Vedecké publikovanie v informatike. Plagiátorstvo a ako sa mu vyhnúť Prednáška 9 bola venovaná plagiátorstvu. Je to vec, ktorá sa dá robiť aj nevedome, ale sú aj ľudia, ktorí sú leniví vynaložiť námahu a urobiť niečo sami a radšej niečo skopírujú a prehlásia to za svoje. V tomto článku som si dala námahu, aby som všetko z čoho som čerpala všade uviedla. Použila som hlavne spomínané plagiátorstvo parafrázovaním, pričom som na konci vždy uviedla zdroj. Zdroj som tiež uviedla pri použití plagiátorstva pri preklade. Táto prednáška bola prínosná, pretože boli uvedené aj spôsoby, ako sa plagiátorstvu vyhnúť. Tieto znalosti vieme použiť aj neskôr, napríklad pri písaní bakalárskej práce.

Literatúra

- [1] AWS Amazon. What is mobile application development? <https://aws.amazon.com/mobile/mobile-application-development/>.
- [2] Appleinsider. Xcode may be coming to the iphone and ipad very soon. <https://appleinsider.com/articles/20/04/20/xcode-may-be-coming-to-the-iphone-and-ipad-very-soon>.
- [3] buildfire. 8 steps to understanding the mobile app development lifecycle. <https://buildfire.com/understanding-mobile-app-development-lifecycle/>.
- [4] Chris Ching. How to submit your app to the app store. <https://codewithchris.com/submit-your-app-to-the-app-store/>.
- [5] DDI Development. 10 major differences between android and ios app development. <https://ddi-dev.com/blog/programming/10-differences-between-android-and-ios-app-development/>.
- [6] EMM. Proces vývoja mobilnej aplikácie – Časť 1. <https://www.emm.sk/blog/our-customers-have-traditionally-valued-special-attention/>.
- [7] EMM. Proces vývoja mobilnej aplikácie – Časť 2. <https://www.emm.sk/blog/proces-vyvoja-mobilnej-aplikacie-cast-2/>.
- [8] eYewated. Android os vs. apple ios - čo je pre vývojárov lepšie? <https://sk.eyewated.com/android-os-vs-apple-ios-co-je-pre-vyvojarov-lepsie/>.
- [9] eYewated. Android os vs. apple ios - čo je pre vývojárov lepšie? <https://sk.eyewated.com/android-os-vs-apple-ios-co-je-pre-vyvojarov-lepsie/>.
- [10] eYewated. Android os vs. apple ios - čo je pre vývojárov lepšie? <https://sk.eyewated.com/android-os-vs-apple-ios-co-je-pre-vyvojarov-lepsie/>.
- [11] eYewated. Čo je mobilná aplikácia? <https://sk.eyewated.com/co-je-mobilna-aplikacia/>.
- [12] Apple Glitz. 5 jazykov pre vývoj aplikácií pre ios, aby sa z vás stal úspešný vývojár. <https://appleglitz.com/slovakia/5-jazykov-pre-vyvoj-aplikacii-pre-ios-aby-sa-z-vas-stal-uspesny-vyvojar/>.
- [13] gStrikar. Run the emulator directly in android studio. <https://www.gsrikar.com/2020/09/run-emulator-directly-in-android-studio.html>.
- [14] EGO Creative Innovations Limited. 5 major differences between ios and android app development. <https://www.ego-cms.com/post/5-major-differences-between-ios-and-android-app-development>.

- [15] Elite Blog Monday Lovers. Ako postupovať pri vývoji mobilnej aplikácie? podme krok za krokom. <https://www.eliteml.sk/blog/ako-postupovat-pri-vyvoji-mobilnej-aplikacie-podme-krok-za-krokom/>.
- [16] Elite Blog Monday Lovers. Mobilné aplikácie v skratke. <https://www.eliteml.sk/blog/mobilne-aplikacie-v-skratke/>.
- [17] The Manifest. What you should know about long-term app maintenance. <https://themanifest.com/app-development/blog/long-term-app-maintenance>.
- [18] Openxcell. Mobile app development guide. <https://www.openxcell.com/mobile-app-development/>.
- [19] Pixelfield. Vývoj mobilních aplikací - kompletní průvodce. <https://pixelfield.cz/vyvoj-aplikaci/#C1>.
- [20] Redbytes Software. Difference between ios and android app development. <https://www.redbytes.in/android-vs-ios/>.
- [21] W2S Solutions. Swift vs objective-c: Which is ideal for ios app development in 2021. <https://www.w2ssolutions.com/blog/swift-vs-objective-c-2021/>.
- [22] DZone Spotlight. The approval process: Android vs. iphone. <https://dzone.com/articles/the-approval-process-android-vs-iphone>.
- [23] SYMETECH. Jak probíhá vývoj mobilních aplikací krok po kroku? <https://synetech.cz/cs/blog/vyvoj-mobilnich-aplikaci-krok-po-kroku>.
- [24] touch4IT. Čo je to ux a ui a prečo bez toho dizajn nie je dizajnom? <https://touch4it.sk/blog/ux-a-ui-dizajn>.
- [25] Vivantina. Vysvetlenie skratky api. <https://www.vivantina.com/marketingovy-slovník/api/>.
- [26] GIS Wiki. Application programming interface. http://wiki.gis.com/wiki/index.php/Application_programming_interface.
- [27] Wikipédia. Vývoj mobilních aplikací. https://cs.wikipedia.org/wiki/V%C3%BDvoj_mobiln%C3%ADch_aplikac%C3%AD.
- [28] Wikipédia. Wireframe. <https://sk.wikipedia.org/wiki/Wireframe>.
- [29] winpcguide. Postup vývoja mobilnej aplikácie príkladom. fázy vývoja mobilných aplikácií. <https://winpcguide.ru/sk/the-security-of-your-computer/procedura-razrabotki-mobilnogo-prilozheniya-na-primere-etapy-razrabotki/>.