

Esej do predmetu Tvorba mobilných aplikácií: Ionic2 framework a vývoj hybridných mobilných aplikácií

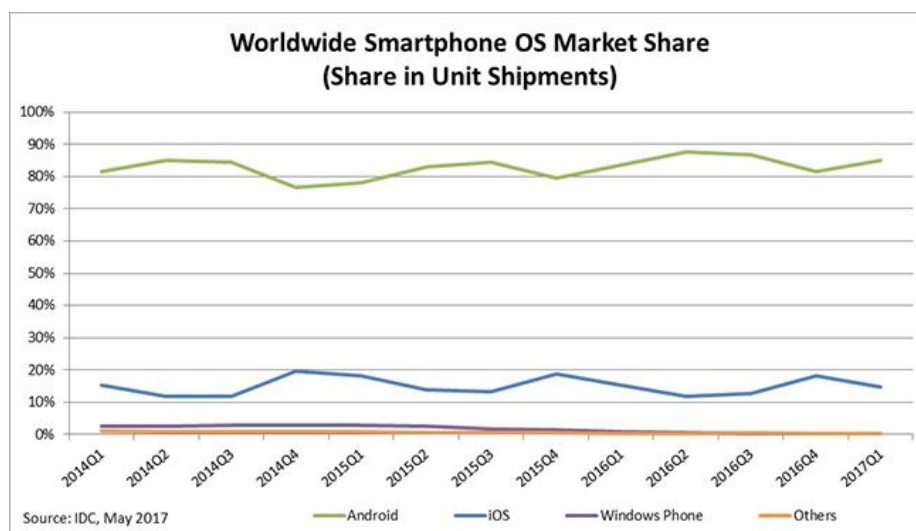
Autor: Filip Gulán (*xgulan00*)

1 Úvod

Témou tejto práce je vývoj hybridných mobilných aplikácií vo všeobecnosti a vývoj hybridných mobilných aplikácií pomocou veľmi populárneho a známeho mobilného frameworku *Ionic2*, ktorý sa v týchto dňoch dočkal už verzie 3.9.2.

2 Hybridné aplikácie

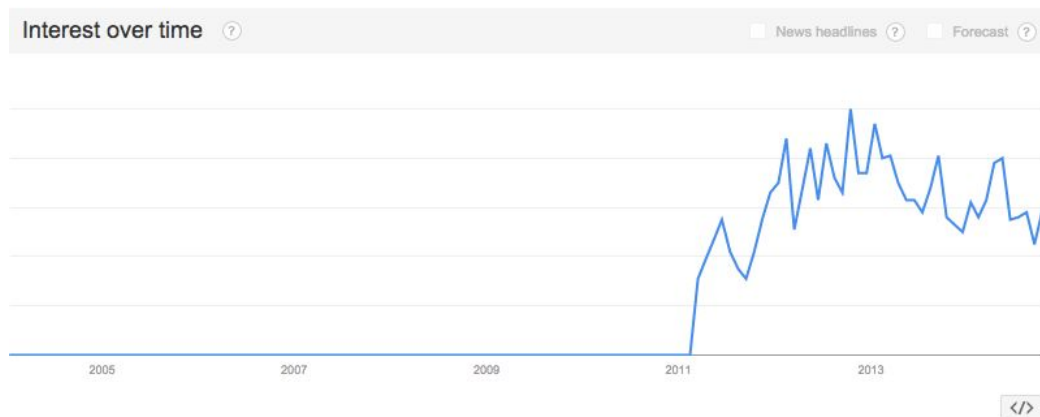
Podľa *International Data Corporation* v tejto dobe existujú 2 veľké mobilné platformy, pre ktoré je zaujímavé vyvíjať nové aplikácie za účelom masového rozšírenia. Prvou platformou je *iOS*, pochádzajúci z dielne kalifornského *Apple*. Druhou platformou je zase *Android* od softwarového giganta *Google*. V minulosti v počte zariadení jasne viedol *Apple*, avšak v posledných rokoch sa k moci dostal naopak *Google* so svojim *Androidom*, o čom svedčia aj čísla podielu na trhu jednotlivých operačných systémov pochádzajúce z *International Data Corporation*. *Android* tu má za posledný rok podiel v priemere cca okolo 85 percent, kdežto *iOS* v priemere iba niečo okolo 15 percent. Z ostatných platforiem je tu ešte zastúpený *Windows Phone* s podielom 0,1 percenta, čo je ale zanedbateľné číslo, a o ostatných platformách, ktoré všetky spolu tvoria tiež niečo okolo 0,1 percenta podielu, tu už ani nemôže byť reč.



Obrázok 1: Podiel zariadení s daným operačným systémom na trhu. Zdroj [4]

V minulosti, ak vývojár chcel vytvoriť aplikáciu pre viacero platforiem, tak musel aplikáciu vytvoriť zvlášť pre každú platformu od začiatku. To znamenalo, že musel použiť programovací jazyk (*Java/Kotlin* pre *Android* *Objective-c/Swift* pre *iOS*) určený pre danú

platformu, prípadne vývojárske prostredie danej platformy. Takto vznikla natívna aplikácia, určená iba pre danú platformu, ktorá je ešte stále najrozšírenejším druhom aplikácií a stále je oproti hybridným prístupom trochu napred, najmä vďaka rýchlosti. V tento moment to už avšak nie je jediná možnosť ako programovať mobilné aplikácie. V súčasnosti sa stále viac a viac dostávajú do popredia takzvané hybridné prístupy k tvorbe aplikácií.



Obrázok 2: Záujem o vývoj hybridných aplikácií v čase. Zdroj [7]

Hybridné aplikácie sú v podstate webové stránky, ktoré sú vstavané do mobilnej natívnej aplikácie cez to, čo nazývame *WebView*. Sú z pravidla vyvíjané *HTML*, *CSS* a *Javascriptom*. Vďaka tomu je spúšťaný ten istý kód naprieč rôznymi platformami a vývoj samotnej aplikácie je pomerne jednoduchý a intuitívny. Vďaka nástrojom ako *PhoneGap* a *Cordova* je dokonca možné používať niektoré natívne funkcie zariadenia, ako napríklad GPS, alebo kameru.

Hybridné aplikácie majú oproti natívnym určité výhody, medzi ktoré bezpochyby patrí už spomínaná jednoduchosť implementácie a hlavne aj to, že aplikáciu stačí napísať raz a potom pobeží takmer na každej bežnej platforme. Ďalšou výhodou, ktorá nadväzuje na to, že aplikácia je napísaná iba raz, namiesto x krát, je to, že aplikácia je takto podstatne jednoduchšia na udržiavanie a popríklad ďalší budúci vývoj. Zároveň cena takejto aplikácie je oveľa nižšia. Prečo ale potom nevyvíjať iba hybridné aplikácie? Ako všetko, tak aj vývoj hybridných aplikácií má okrem výhod aj početné nevýhody. Medzi nevýhody môže patriť napríklad to, že hybridné aplikácie sú v porovnaní s natívnymi o niečo pomalšie, hlavne keď sa zameriame na animácie a iné interaktívne prvky. Avšak aj tu platí, že ako technológie stále viac a viac napredujú, tak sa aj tento rozdiel skôr znižuje a dovoľm si tvrdiť, že bežný užívateľ výrazný rozdiel medzi natívnou aplikáciou a aplikáciou hybridnou už v podstate ani nepostrehne. Ďalšou nevýhodou, ktorá začína byť čoraz viac minulosťou je, a to hlavne vďaka príchodom hybridných frameworkov ako *Ionic*, alebo *Xamarin*, to, že hybridné aplikácie ponúkajú horšie *UX* a menej estetické *GUI*. Čo avšak stále platí, je to, že aj keď hybridné aplikácie ponúkajú vďaka *Cordova* pluginom prácu s natívnymi funkciami zariadenia, stále sa ale môže stať, že takýto plugin na takúto funkciu jednoducho nebude existovať. Ďalej platí to, že natívne aplikácie sú viac bezpečnejšie na používanie, keďže pri natívnych aplikáciách prebieha vždy proces schvaľovania zo strany obchodu, ktorý danú

aplikáciu ponúka (*Google play/App store*). Po prečítaní výhod a nevýhod sa môže naskytnúť otázka: “*Akým smerom sa teda uberať?*”. Všeobecná odpoveď, ako je už pomaly zvykom, neexistuje. Veľmi záleží od charakteru aplikácie, jej biznis modelu a cielenia na užívateľov.

3 *Ionic2*

Ionic2 je jeden z najpopulárnejších *open-source* frameworkov určených pre vývoj hybridných mobilných aplikácií. Originálny *Ionic* framework bol vydaný v roku 2013 a bol postavený na technológii *AngularJS* od *Google* a taktiež na už spomínanej *Apache Cordova*. Vydaním *Angular2*, ktorý nie je iba ďalšou verziou *AngularJS* ale v podstate úplne nový framework fungujúci mierne na odlišných princípoch ako jeho predošlí príbuzní, bol vydaný taktiež *Ionic2*. Nový *Ionic2* síce technológiu *Cordova* ponechal, avšak *AngularJS* vymenil za novší *Angular2*. *Ionic2*, tak ako *Angular2*, je v podstate *MVC* framework, ktorý si zakladá na tvorbe znovupoužiteľných komponentov. Framework využíva programovací jazyk *Typescript*. *Typescript* je nadstavba nad jazykom *Javascript*, ktorý vytvoril *Microsoft*. Obohacuje samotný *Javascript* o statické typovanie a atribúty typické pre objektovo orientované jazyky, a tak programátora chráni pred vytváraním niektorých bežných chýb. Samotný kód *Typescriptu* sa potom transpiluje do čistého *Javascriptu*. Keďže je *Ionic2* hybridný framework, tak prirodzene obsahuje všetky plusy a mínusy, ktoré boli uvedené už v minulej kapitole. *Ionic2* aj napriek tomu, že je primárne určený pre tvorbu hybridných aplikácií, tak je možné s ním vytvoriť aj progresívne webové aplikácie. Súčasťou frameworku je okrem iného aj *Ionic Command line interface*, čo zjednodušuje niektoré veci, ako napríklad inicializáciu nového projektu, generovanie komponentov, pridávanie natívnych pluginov a mnoho iných.

Ionic2 bol ako tak predstavený. V čom ale vyniká a v čom naopak zaostáva za svojimi konkurentami, menovite *React Native* od *Facebooku*, *Xamarin* od *Microsoftu*, alebo *JqueryMobile*? *Xamarin* oproti *Ionic2* ponúka možnosť písať aplikácie aj natívne (zamedzí to ale, aby bola aplikácia multiplatformná) aj hybridne. Okrem toho, aplikácie vytvorené pomocou *Xamarinu* sa spúšťajú oveľa rýchlejšie, ako aplikácie vytvorené v *Ionic2*. Na druhú stranu, veľkosť výslednej *Xamarin* aplikácie je omnoho väčšia, ako veľkosť *Ionic2* aplikácie. Tento fakt môže predstavovať pre niektorých užívateľov, ktorý majú málo miesta vo svojom zariadení, reálny problém keď príde na jednoduché aplikácie, ktoré napriek tomu zaberajú obrovské miesto v pamäti. Čo sa týka podpory platformy, na ktoré je možné exportovať výslednú aplikáciu, tak *Xamarin* aj *Ionic2* sú na tom dosť podobne. Výsledné aplikácie zo *Xamarinu* sú podporované na *iOS 6.1+* a *Android 4.0+* a z *Ionic2* na *iOS 7+* a *Android 4.1+*. Čo sa týka *React Native* tak *Ionic2* má oveľa lepšiu, prehľadnejšiu a celistvejšiu dokumentáciu. Avšak znova, ako pri *Xamarine*, tak aj *React Native* vedie v rýchlosti výslednej aplikácie oproti *Ionic2*. *React Native* používa *JSX*, čo je preprocesor, ktorý pridáva do *Javascriptu* podporu syntaxe *XML*. Aplikácie *React Native* sa kompilujú do natívnej aplikácie a je tak možné pristupovať k GPS, kamere a podobne priamo a nie sú potrebné *Cordova* pluginovy. To má avšak jeden nedostatok. Tým sa vytráca sa pravá multiplatformnosť, pretože aplikácia sa musí písať, síce v tom istom programovacím jazyku,

ale separátne pre rôzne platformy znovu. Čo sa týka posledného zmieneného konkurenta *Jquery Mobile*, tak *Ionic2* je oproti nemu oveľa komplexnejší framework a *Jquery* je v podstate iba akási menšia knižnica. *Jquery mobile* neponúka ani žiadnu štruktúru, ako napríklad *MVC* vzor, ako je to v prípade *Ionic2*.

4 Záver

V tejto práci boli rozobrané hybridné aplikácie vo všeobecnosti a ich výhody a nevýhody oproti klasickým natívnym. Okrem toho, bol predstavený framework *Ionic2*, ktorý je jeden z najtypickejších hybridných frameworkov a bol následne porovnaný s konkurentami vo svojej kategórii. V obidvoch prípadoch by sa dal vyvodiť záver, že neexistuje ideálna technológia a každá technológia má svoje pre a proti a taktiež, že výber vhodnej technológie závisí na viacerých faktoroch, či už je to biznis model, rozpočet, alebo niečo úplne iné.

Referencie

- [1] *TypeScript: JavaScript that scales* [online]. [cit. 2017-12-03]. Dostupné z: <https://www.typescriptlang.org/>
- [2] *Angular* [online]. [cit. 2017-12-03]. Dostupné z: <https://angular.io/>
- [3] *Ionic: Build Amazing Native Apps and Progressive Web Apps with Ionic Framework and Angular* [online]. [cit. 2017-12-03]. Dostupné z: <http://ionicframework.com/>
- [4] *IDC:: Smartphone OS Market Share* [online]. [cit. 2017-12-03]. Dostupné z: <https://www.idc.com/promo/smartphone-market-share/os>
- [5] RICO, Fabianne. *CREATING TOP APPS: THE PROS AND CONS OF NATIVE VS HYBRID MOBILE APP DEVELOPMENT* [online]. 2016 [cit. 2017-12-03]. Dostupné z: <https://impekable.com/native-vs-hybrid-mobile-app-development/>
- [6] ALEXSEYENKO, Andrew. *Native vs Hybrid apps: What to choose in 2017?* [online]. 2017 [cit. 2017-12-03]. Dostupné z: <https://blog.techmagic.co/native-vs-hybrid-apps/>
- [7] GLEICHGER, Tal. *Hybrid UI framework shootout: Ionic vs. Famo.us vs. F7 vs. OnsenUI* [online]. 2016 [cit. 2017-12-03]. Dostupné z: <https://www.airpair.com/ionic-framework/posts/hybrid-apps-ionic-famous-f7-onsen>
- [8] SOLANKI, Jignesh. *Xamarin vs Ionic for Application Development* [online]. 2017 [cit. 2017-12-03]. Dostupné z: <https://www.simform.com/xamarin-vs-ionic-application-development/>
- [9] AGGARWAL, Ankush. *Ionic vs React Native* [online]. 2017 [cit. 2017-12-03]. Dostupné z: <https://medium.com/@ankushaggarwal/ionic-vs-react-native-3eb62f8943f8>