|  |  |
| --- | --- |
|  | |
| **Návrh multiplatformní mobilní aplikace pro rychlou komunikaci vyučujících se studenty** | |
|  | |
| Martin Gabriel | |
|  | |
|  |  |
| 2019 | Popis: fai_logo_cz |
|  |  |
|  | |

// SCAN PRVNÍ STRANY ZADÁNÍ ZDE

// SCAN DRUHÉ STRANY ZADÁNÍ ZDE

**Prohlašuji, že**

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

**Prohlašuji,**

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

Ve Zlíně, dne …………………….

podpis diplomanta

ABSTRAKT

Cílem práce je návrh vhodných technologií a postupů pro aplikaci pro rychlou a efektivní komunikaci vyučujících se studenty. Aplikace bude umožňovat zasílání rychlých a cílených zpráv studentům o změnách výuky a dále bude aplikace umožňovat studentům rychlé hodnocení výuky. V teoretické části budou popsány vybrané technologie, doporučené postupy, návrhové vzory a techniky pro tvorbu mobilních aplikací. V praktické části budou popsané klíčové části navrženého řešení.

Klíčová slova:

Xamarin, iOS, Android, Azure, .NET Core, C#, Push notifikace

ABSTRACT

Abstrakt ve světovém jazyce

// DOPLNIT ABSTRAKT V AJ

Keywords:Poděkování, motto a čestné prohlášení, že odevzdaná verze bakalářské práce a verze elektronická, nahraná do IS/STAG jsou totožné ve znění:

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

OBSAH

[Úvod 8](#_Toc6759648)

1. [TEORETICKÁ ČÁST 9](#_Toc6759649)

[1 Mobilní Platformy 10](#_Toc6759650)

[1.1 Android 10](#_Toc6759651)

[1.1.1 Architektura systému Android 10](#_Toc6759652)

[1.2 Operační systém iOS 11](#_Toc6759653)

[1.2.1 Architektura systému iOS 12](#_Toc6759654)

[1.3 Další mobilní operační systémy 13](#_Toc6759655)

[2 Multiplatformní vývoj 14](#_Toc6759656)

[2.1 Nativní aplikace 14](#_Toc6759657)

[2.2 Webový a hybridní vývoj 14](#_Toc6759658)

[2.3 Herní vývoj 15](#_Toc6759659)

[2.4 Multiplatformní vývoj 15](#_Toc6759660)

[3 Xamarin 17](#_Toc6759661)

[3.1 Mono 17](#_Toc6759662)

[3.2 Xamarin.Forms 18](#_Toc6759663)

[3.2.1 Výkon aplikací v Xamarin.Forms 18](#_Toc6759664)

[3.3 Xamarin.Android 19](#_Toc6759665)

[3.4 Xamarin.iOS 20](#_Toc6759666)

[4 Azure a .NET core 21](#_Toc6759667)

[4.1 Azure functions 21](#_Toc6759668)

[4.2 Azure Notification Hubs 22](#_Toc6759669)

[4.3 .NET Core 22](#_Toc6759670)

[5 Notifikace 23](#_Toc6759671)

[5.1 Lokální notifikace 23](#_Toc6759672)

[5.2 Push notifikace 23](#_Toc6759673)

[5.2.1 Push notifikace v operačním systému Android 23](#_Toc6759674)

[5.2.2 Push notifikace v operačním systému iOS 24](#_Toc6759675)

1. [Praktická část 26](#_Toc6759676)

[6 Rozbor architektury projektu 27](#_Toc6759677)

[6.1 Požadavky projektu 27](#_Toc6759678)

[6.1.1 Multiplatformní mobilní aplikace 27](#_Toc6759679)

[6.1.2 Cloudový back-end 28](#_Toc6759680)

[6.2 Návrh architektury 28](#_Toc6759681)

[6.3 Popis navrhnuté serverové služby 28](#_Toc6759683)

[6.3.1 Serverová služba 28](#_Toc6759684)

[6.3.2 Hosting serveru 29](#_Toc6759685)

[6.3.3 Komunikace se STAG API 29](#_Toc6759686)

[6.3.4 Notification Hub 30](#_Toc6759687)

[7 Návrh serverové části projektu 32](#_Toc6759688)

[7.1 .NET Core služba 32](#_Toc6759689)

[7.2 REST API StudentsNotifier AppService 32](#_Toc6759690)

[8 Návrh uživatelského prostředí 34](#_Toc6759691)

[9 Notifikace 35](#_Toc6759692)

[9.1 Notification Hub 35](#_Toc6759693)

[9.2 Xamarin 35](#_Toc6759694)

[10 Příklad využití aplikace 36](#_Toc6759695)

[10.1 Podnadpis 36](#_Toc6759696)

[Závěr 37](#_Toc6759697)

[Seznam použité literatury 38](#_Toc6759698)

[Seznam použitých symbolů a zkratek 41](#_Toc6759699)

[Seznam obrázků 42](#_Toc6759700)

[Seznam tabulek 43](#_Toc6759701)

[Seznam Příloh 44](#_Toc6759702)

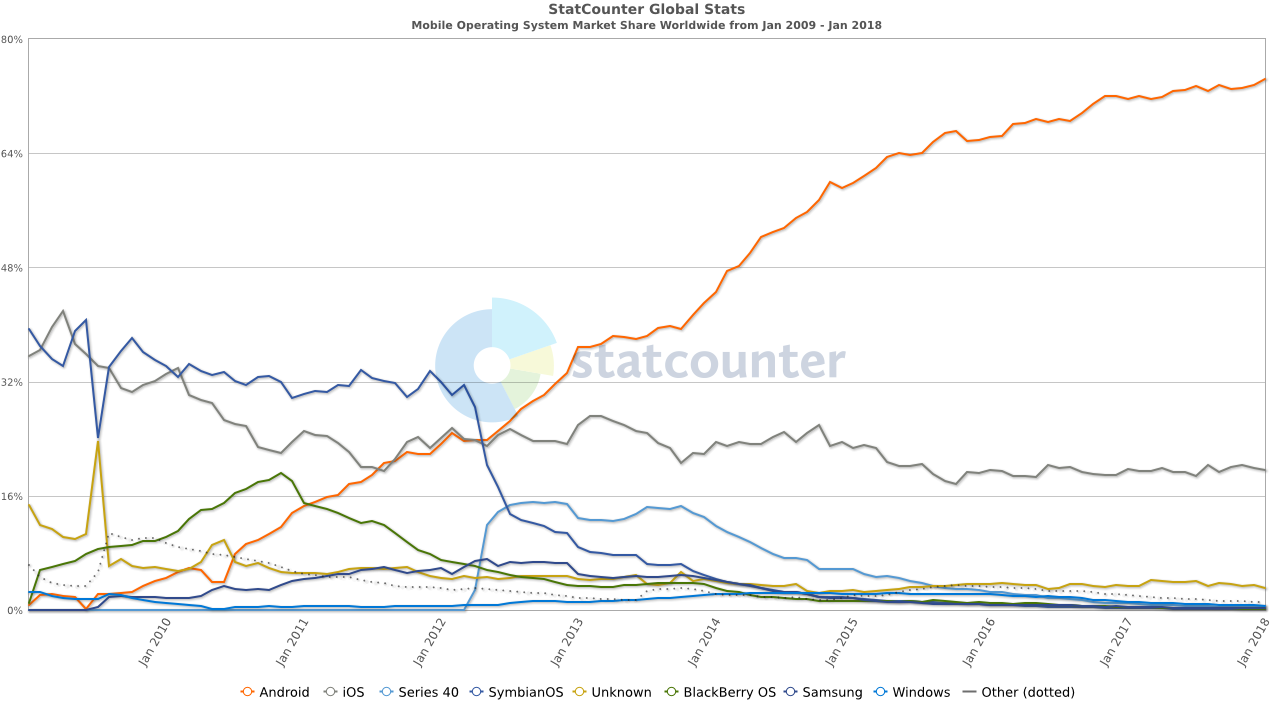
Úvod

// DOPLNIT ÚVOD

|  |  |
| --- | --- |
|  | TEORETICKÁ ČÁST |

# Mobilní Platformy

Pojmem mobilní platformy rozumíme operační systémy a ekosystémy, na kterých jsou postaveny současné mobilní telefony, tablety a jiné chytré zařízení. V rámci práce se budeme zabývat především mobilními telefony a tablety. Mezi nejvýznamnější systémy patří operační systémy Android od společnosti Google a iOS od společnosti Apple.



Graf 1 Podíl mobilních OS na trhu [1]

// DOPLNIT AKTUÁLNÍ DATA A GRAF

Z grafu vidíme postupnou dominanci operačních systémů Android a iOS. Ostatní operační systémy mají zanedbatelný podíl na trhu mobilních zařízení.

## Android

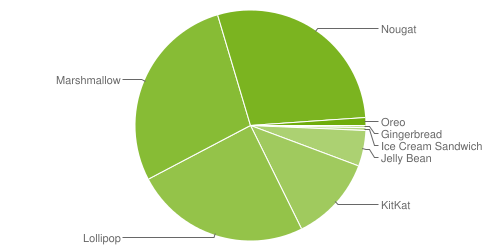
Android je mobilní operační systém založený na jádře Linuxu, který vyvíjí uskupení Open Handset Alliance (pod hlavičkou společnosti Google) a je distribuován jako opensource. Systém Android má největší zastoupení na mobilním trhu.

### Architektura systému Android

Architektura systému je rozdělena na 5 vrstev:

1. Linux Kernel - nejnižší vrstvou je jádro operačního systému, které se stará o práci mezi software a hardware.
2. Knihovny – jsou napsány v jazyce C/C++, využívají různé komponenty systému, slouží vývojářům prostřednictvím Android Application Framework k práci s práci s audio/video soubory, webovým prohlížečem, relační databází, vykreslování 3D grafiky a další
3. Android Runtime – obsahuje virtuální stroj Dalvik, základní knihovny jazyka Java,
4. Application network – poskytuje přístup ke službám, které mohou být použity v aplikacích, tyto služby mohou zpřístupňovat data v jiných aplikacích (například kontakty, aktuální polohu, atd.), přístup k hardware, notification manager, aktivity manager a další
5. Aplikace – základní aplikace s uživatelským rozhraním (například mail klient, webový prohlížeč, mapy a podobně)

Operační systém Android je nejrozšířenějším mobilním operačním systémem. Nasazuje jej do svých zařízení široká škála výrobců, z čehož pramení jeho nevýhoda – velké množství nasazených verzí s nejistou možností aktualizací systému (například bezpečnostích záplat), protože aktualizace závisí pouze na výrobci.



Obrázek 1 Podíl verzí systému Android na trhu v Q1 2018 [6]

// DOPLNIT AKTUÁLNÍ DATA A GRAF (PROCENTA)

## Operační systém iOS

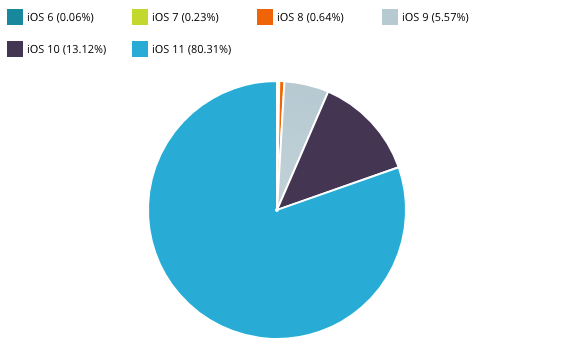
iOS je mobilní operační systém založený na macOS (je jeho odlehčenou verzí). Jedná se o systém UNIXového typu. Původně byl iOS vytvořen pouze pro mobilní telefony iPhone (pod názvem iPhoneOS), ale později se začal používat na dalších zařízeních jako hudební přehrávače iPod, nebo tablety iPad.

### Architektura systému iOS

Architektura systému je rozdělena na 4 základní vrstvy:

1. Core OS – stará se o nízkoúrovňový přístup k hardware, dále se stará například o bluetooth konektivitu, služby zabezpečení, autentifikaci a další
2. Core Services Layer – stará se o vysokoúrovňové služby, například lokační služby, relační databáze, správa úloh v aplikaci, podpora XML dokumentů a další
3. Media Layer – umožňuje vytváření grafických aplikací, přehrávání animací, videí a zvuků
4. Cocoa Touch – obsahuje nejdůležitější frameworky pro vývoj aplikací, poskytuje infrastrukturu pro implementaci grafického uživatelského rozhraní, interakce s uživatelem a poskytuje vysokoúrovňové systémové služby, například multitasking, push notifikace, rozpoznávání gest a další

Operační systém iOS je distribuován pouze pro zařízení společnosti Apple. Svou koncepcí se jedná o uzavřený ekosystém, což má výhodu v optimalizaci a bezpečnosti, ale nevýhodu v nulové upravitelnosti. Oproti Androidu se nové verze iOS aktualizují daleko rychleji na větší poměr zařízení na trhu.



Obrázek 2 Podíl verzí systému iOS na trhu Q1 2018 [7]

// DOPLNIT AKTUÁLNÍ DATA A GRAF

## Další mobilní operační systémy

Další operační systémy mají na trhu minimální zastoupení. Mezi nadějnou konkurenci donedávna patřil operační systém Windows Phone od společnosti Microsoft (dříve znám jako Windows Mobile, později jen jako Windows), který ale v současné době není vyvíjen a v roce 2019 končí oficiální podpora tohoto systému. Mezi další neúspěšné projekty paří například FirefoxOS [8], nebo Symbian. Existuje několik aktivně vyvíjených alternativ jako například Sailfish OS.

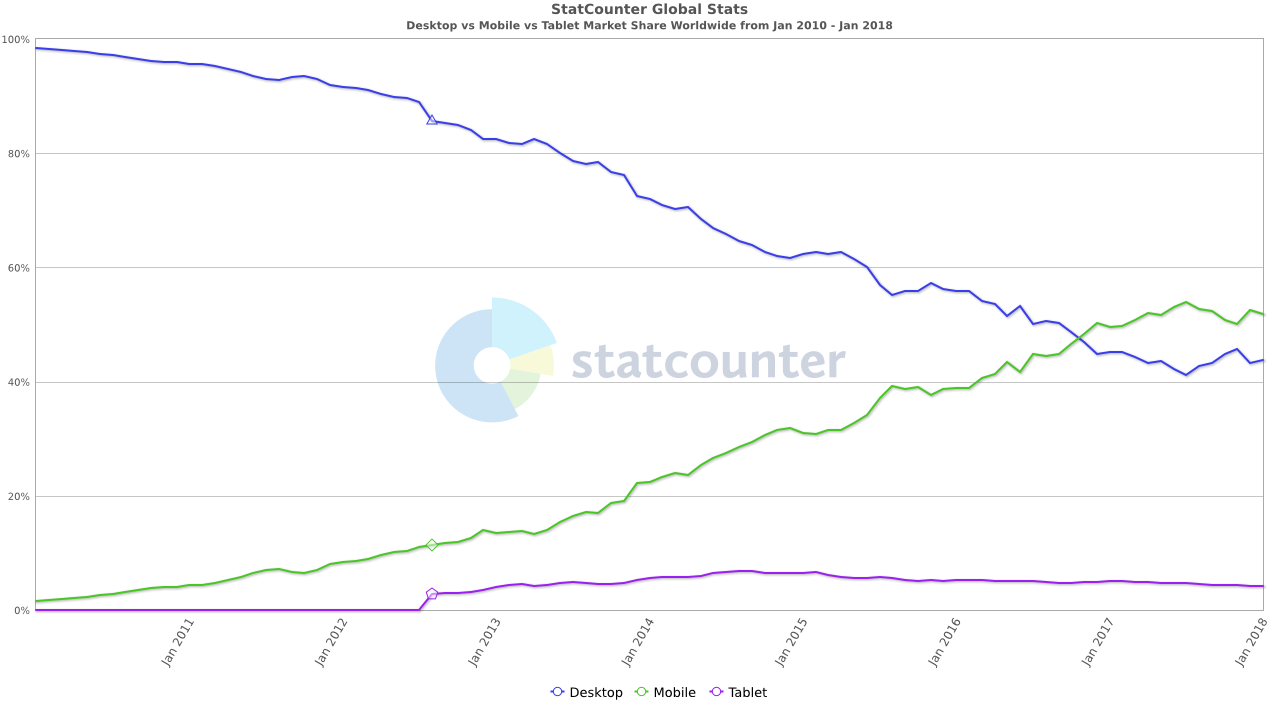
# Multiplatformní vývoj

Multiplatformní vývoj chápeme jako vývoj software, jehož výsledek lze spustit na více platformách, nebo více operačních systémech. Existuje několik druhů multiplatformního vývoje a dělíme je podle použití.

## Nativní aplikace

Nativní vývoj znamená vytvoření aplikace pro každou platformu zvlášť. Každá platforma má vlastní přístup k uživatelskému přístupu, vlastní systémové API, odlišný jazyk a vývojové prostředí. Tento přístup se používá hlavně na velkých projektech, kde je potřeba dosáhnout maximálního výkonu a není problémem časová náročnost spojená s vyššími finančními náklady na vývoj.

## Webový a hybridní vývoj



Graf 2 Poměr PC proti mobilním zařízením na webu [1]

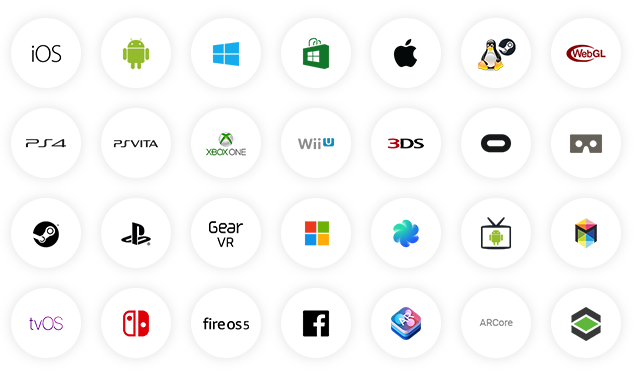
// DOPLNIT DATA A GRAF

Webové stránky přizpůsobené pro mobilní zařízení mohou sloužit jako mobilní aplikace. Při vývoji webů je potřeba dbát na vzestupný trend prohlížení webů na mobilních zařízeních. Pro tento účel vznikl pojem responzivní design, který klade důraz na to, aby se stránka přizpůsobila velikosti displeje, na kterém je zobrazena. K tomuto typu vývoje slouží přístup Mobile First, tedy návrh stránky nejprve pro mobilní zařízení, ze kterého později vyplyne i desktopová verze [19]. Nevýhodou tohoto přístupu je neustálá potřeba připojení k internetu.

Další možností jsou hybridní HTML5 aplikace, které vypadají jako nativní aplikace, ale prakticky jde o webový prohlížeč, který zobrazuje uživatelské rozhraní pomocí webových technologií a má přístup k nativnímu API k ovládání fotoaparátu, GPS, apod. Zástupcem tohoto přístupu je Framework PhoneGap distribuovaný jako open source, který vychází z Apache Cordova frameworku [21]. // progressivní webové aplikace – connect.cz

## Herní vývoj

Pro vývoj her se používá herní engine, tedy software umožňující jednoduchý a v mnohých případech i multiplatformní vývoj her. Mezi nejznámější multiplatformní herní engine patří například Unity3D. Nabízí vývoj her pro více než 25 platforem včetně mobilních telefonů, osobních počítačů a konzolí. U mobilních telefonů dosahuje nativního vysokého výkonu díky využití grafických API Metal (iOS) a Vulkan (Android) [20].



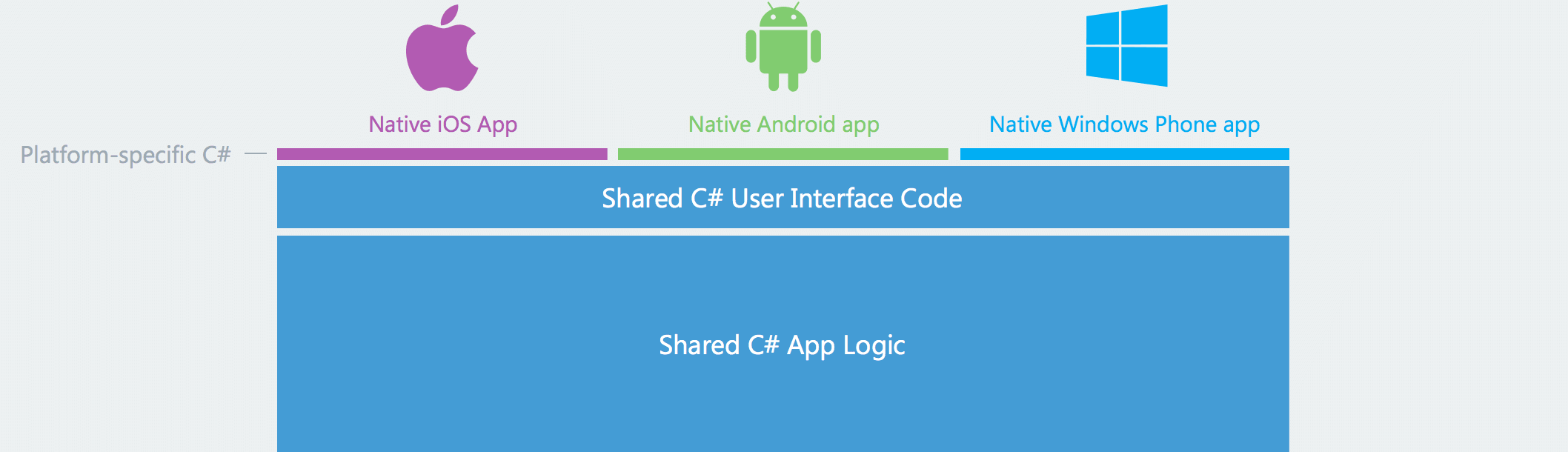
Obrázek 3 Výčet podporovaných platforem Unity3D [20] // možná jen výčet v textu

## Multiplatformní vývoj

Multiplatformní vývoj aplikací se nejvíce vyplatí při vývoji business aplikací, nebo tam, kde je potřeba sdílet co nejvíce kódu a šetřit tak čas vývoje. Mezi nejznámější zástupce multiplatformního vývoje patří Xamarin, kterému nově konkuruje open source Framework Flutter od společnosti Google [22]. Výsledné aplikace jsou vyvíjeny rychle, mají nativní výkon a flexibilní uživatelské rozhraní. //QT, Xamarin, Flutter zvlášť,

# Xamarin

Xamarin je společnost kterou založil Miguel de Icaza v roce 2011, stojí za vznikem platforem Mono, Mono for Android, MonoTouch, což jsou multiplafromní implementace Common Language Infrastructure (CLI) a Common Language Specifications (.NET). V roce 2016 byla firma Xamarin zakoupena společností Microsoft a Xamarin byl plně integrován do vývojového prostředí Visual Studio [13].



Obrázek 4 Princip sdílení kódu v Xamarin platformě [12]

Platforma Xamarin je určena pro vývoj aplikací, kde je potřeba sdílet co nejvíce společného kódu. Umožňuje vytvářet nativní aplikace se společnou aplikační logikou (Xamarin.iOS a Xamarin.Android), nebo aplikace se sdílenou aplikační logickou a uživatelským rozhraním (Xamarin.Forms) přičemž klade důraz na zachování výkonu proti čistě nativním aplikacím.

## Mono

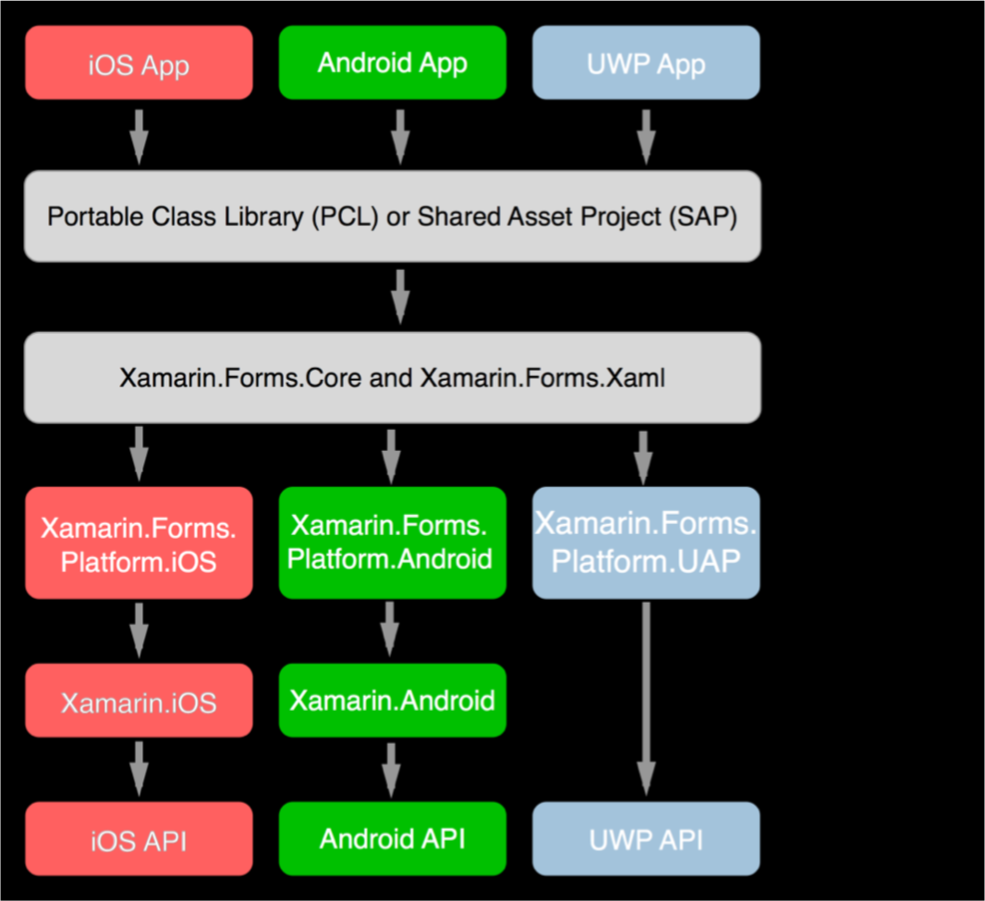
Mono je softwarová platforma vytvořená pro vývoj multiplatformních aplikací prostřednictvím .NET standart. Je vyvíjena je open source a založená na ECMA standartu pro C# a Common Language Runtime. Smyslem této plaftormy je umožnění spouštění .NET aplikací na jiných operačních systémech než Windows. [14] V době psaní této práce podporuje Mono vše z .NET verze 4.7 vyjma WPF, WWF, omezeně WCF a ASP.NET async stack [15].

Mono je v současné době podporováno na těchto platformách [15]:

* Linux
* Mac OSX, iOS, tvOS, watchOS
* Sun Solaris
* BSD – OpenBSD, FreeBSD, NetBSD
* Microsoft Windows, XboxOne
* Sony Playstation 4

## Xamarin.Forms

Xamarin.Forms umožňuje vytváření multiplatformního User Interface (dále UI), který je kompilován do nativního kódu jednotlivé platformy [17]. Podle oficiálního webu společnosti Xamarin je při vývoji v Xamarin.Forms 96% kódu sdíleného mezi platformami [16].



Obrázek 5 Popis vytvoření multiplatformní Xamarin.Forms aplikace [17]

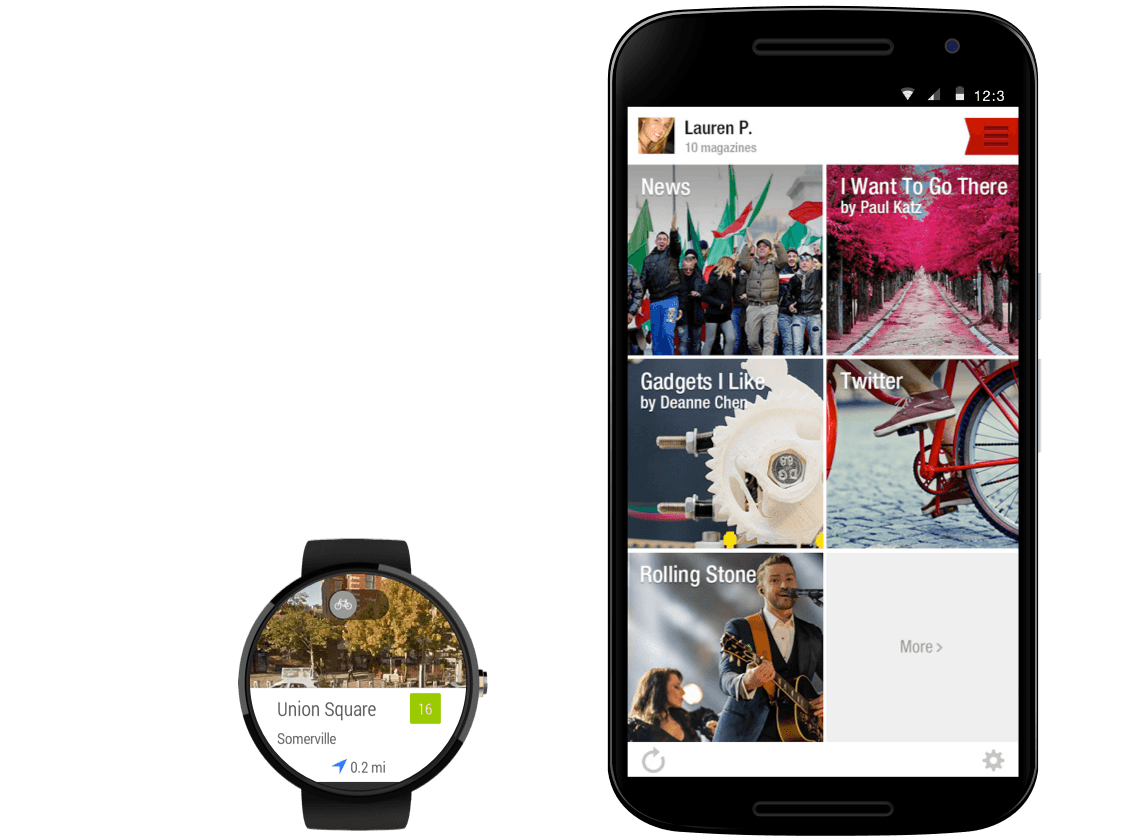
### Výkon aplikací v Xamarin.Forms

Mezi multiplatformním UI frameworkem Xamarin.Forms, nativním Xamarin.iOS nebo Xamarin.Android a čistě nativní aplikací, jsou výkonnostní rozdíly. Podle testů webu AlexSoft jsou nejvíce výkonné nativní aplikace v iOS, nejvíce vyvážené jsou pak aplikace napsané v Xamarin.iOS a Xamarin.Android, avšak uživatel nejspíše vůbec nepozná rozdíl mezi aplikací napsanou v Xamarin.Forms, které se jeví jako nejpomalejší [17].

## Xamarin.Android

Xamarin.Android je framework, který umožňuje vývoj nativních Android apliakcí s pomocí Android SDK v jazyce C# nebo F# za použití knihoven .NET a vývojového prostředí Visual Studio. Ke svému chodu potřebuje Windows 7 a vyšší, Visual Studio 2015 nebo 2017 a Xamarin for Visual Studio. Instalace obstará stažení dalších potřebných částí jako Xamarin Workbooks, Xamarin Profiler, Xamarin Remoted Simulator, Android NDK, Android SDK, Java SE Develpment Kit, Google Android Emulator a další [11].

Kompilace v Xamarin.Android využívá principu Just In Time. C# kód je zkompilován do assemblies a ty jsou součástí výsledného APK souboru. Spuštění aplikace potom probíhá pomocí Mono virtuálního stroje, který vykonává instrukce z assemblies stejně jako by virutální stroj Dalvik zpracovával nativní aplikaci napsanou v jazyku Java [11].



Obrázek 6 Xamarin.Android aplikace pro Android a Wear OS [15]

## Xamarin.iOS

Xamarin.iOS je framework, který umožňuje vývojářům vytvářet nativní iOS aplikace pomocí SDK iOS, přičemž lze psát aplikaci v jazyce C# za použití knihoven .NET Base Class Library (BCL) a vývojového prostředí Visual Studio. Mezi systémové požadavky patří iOS SDK, poslední verze Xcode a macOS Seirra 10.12 nebo vyšší, takže je k vývoji nativních iOS aplikací v Xamarinu potřeba zařízení od společnosti Apple [10].

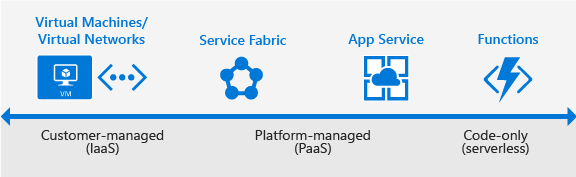
Kompilace v Xamarin.iOS využívá principu Ahead of Time Compilation, kde je C# kód převeden do nativního ARM kódu a výsledkem je balík .APP s aplikací. V Xamarin.iOS není dostupná dynamická generace kódu, takže za chodu programu musí být všechen kód již zkompilován [10].



Obrázek 7 Xamarin.iOS aplikace pro iPhone a WatchOS [12]

# Azure a .NET core

Azure je sada cloudových služeb, která slouží k vytváření, hostování a škálování webových aplikací na cloudových serverech společnosti Microsoft, k virtualizaci, nasazení databází a dalších. Umožňuje nasazení, testování, monitoring a správu služeb. Cloudová služba funguje na run-time operačním systému Windows Azure, který zprostředkovává řadu služeb, které není potřeba instalovat a jejich konfigurace probíhá z prostředí webového portálu [23].



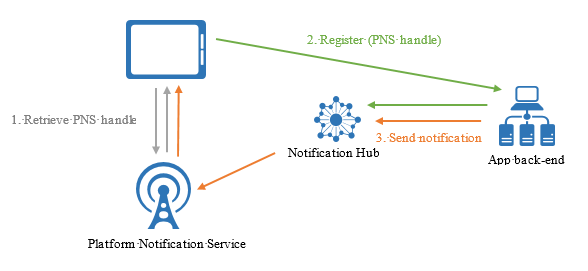
Obrázek 8 Popis Azure

## Azure functions

Jedná se o tzv. serverless výpočetní službu, která umožňuje vytváření jednoduchých API funkcí bez potřeby celé serverové aplikace a infrastruktury. Je možno psát kód v jazycích C#, F#, Node.js, Java nebo PHP [26].

## Azure Notification Hubs

Notification Hubs je modul, který umožňuje odesílání push notifikací na zařízení se systémy iOS, Android, Windows, nebo Kindle. Využívá k tomu služby Apple Push notification Servuce, Google Cloud Messaging, Windows Push Notification Service, Microsoft Push Notification Service a další. Dá se připojit k webovým službám založených na platformách .NET, PHP, Java nebo Node.js.



Obrázek 9 Způsob zpracování push notifikací pomocí Notification Hub [24]

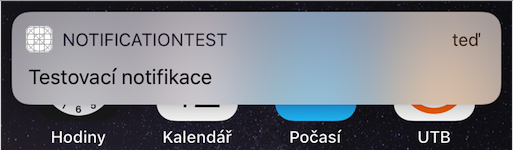
Výhodou této služby je jednoduchá implementace v .NET a správa v Azure portálu. Alternativou je služba Firebase od společnosti Google [25].

## .NET Core

Jedná se o open source framework pro systémy Windows, maxOS a Linux který podporuje multiplatformní vývoj webových aplikací ASP.NET Core, command-line aplikací, knihoven a UWP aplikací. .NET Core neobsahuje Windows Forms a WPF, které renderují GUI pro desktopové aplikace na OS Windows. Typicky se hodí na vývoj multiplatformních microservices.

# Notifikace

Notifikace jsou jednou z důležitých částí (nejen mobilních) operačních systémů. Slouží k upozornění uživatele na aktivitu aplikace. Může jít například o upozornění na příchozí zprávu, zmeškaný hovor, nebo o připomínku v kalendáři, která dává prostřednictvím notifikací uživateli vědět, že se něco stalo [2].



Obrázek 10 Příklad zobrazení notifikace v horní části obrazovky telefonu

Notifikace lze z pravidla spravovat v notifikačním centru, což je součást operačního systému, která sdružuje chronologicky seřazené notifikace ze všech aplikací a programů v zařízení [5].

Notifikace dělíme na dvě základní kategorie. Lokální notifikace a Push notifikace.

## Lokální notifikace

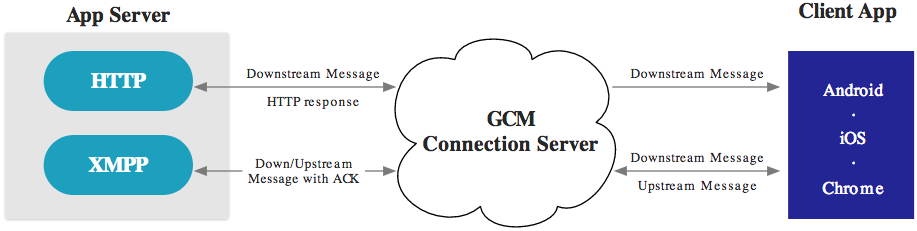
U lokálních notifikací aplikace předává detaily o notifikaci operačnímu systému uvnitř daného zařízení, který následně zpracovává zobrazení notifikace. Děje se tak i v momentě kdy aplikace, nebo program, nejsou vůbec spuštěny [2]. Může jít například o událost v kalendáři, která v určitý čas upozorní uživatele prostřednictvím notifikace.

## Push notifikace

U push notifikací operační systém přijímá a zpracovává notifikace na základně vnějšího podnětu ze strany serverové notifikační služby. Serverová aplikace komunikuje se zařízením přes serverové služby, které si každá platforma řeší zvlášť.

### Push notifikace v operačním systému Android

V mobilním operačním systému Android se zasílání a příjem push notifikací řeší pomocí služby Google Cloud Messaging (dále GCM). Serverová aplikace vyhodnotí požadavek na zaslání push notifikace na konkrétní zařízení, pomocí HTTP a XMPP protokolu předá požadavek GCM serveru. Ten následně odešle notifikace ke konkrétnímu zařízení [4].

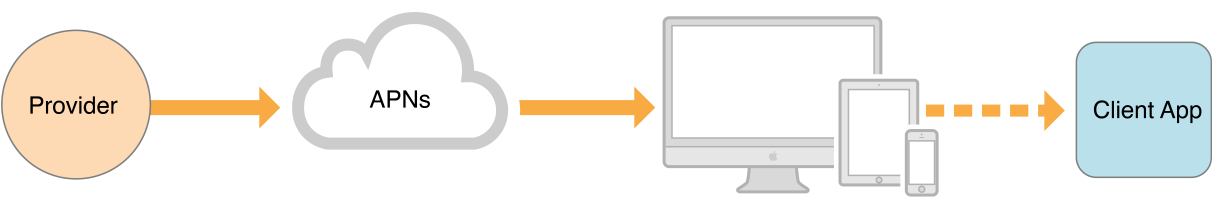


Obrázek 11 Architektura Google Cloud Messaging [4]

Přes GCM je možná i komunikace od zařízení (tedy aplikace) k aplikačnímu serveru – například pro odesílání zprávy v chatu. Zařízení musí být registrováno pro příjem notifikací. Při registraci dostane unikátní identifikátor nazvaný registrační token. Pomocí tohoto tokenu pak lze zasílat notifikace na konkrétní zařízení [4].

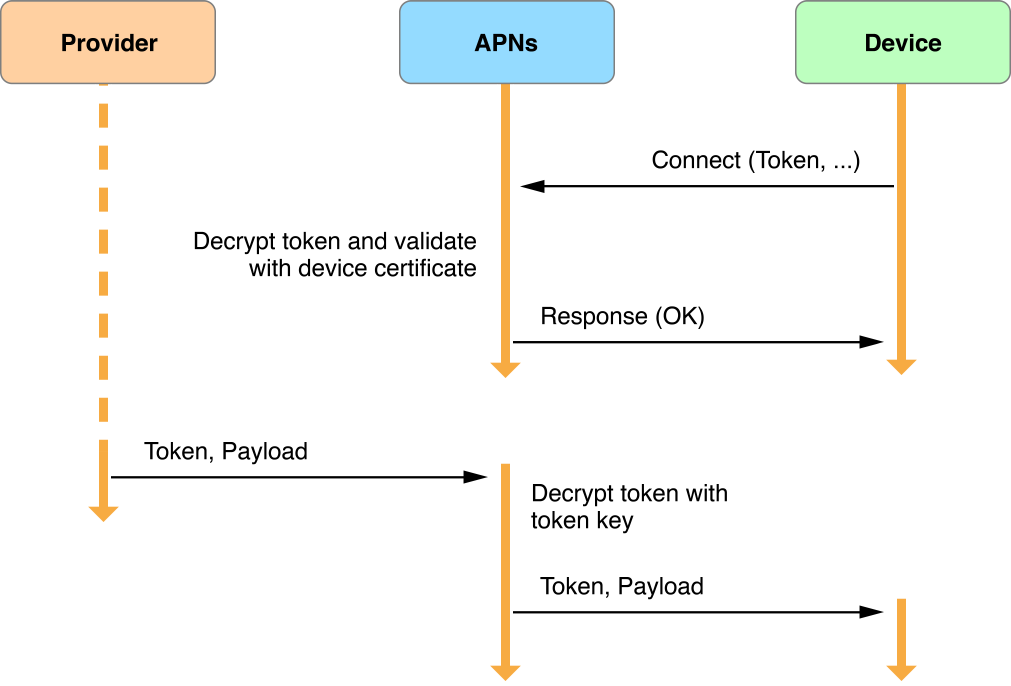
Od roku 2017 společnost Google doporučuje přechod na nový systém multiplatformního zpracování notifikací s názvem Firebase [9].

### Push notifikace v operačním systému iOS



Obrázek 12 Příjem push notifikací od odesílatele do aplikace [3]

Zpracování push notifikací v operačních systémech společnosti Apple probíhá přes službu Apple Push Notification Service (dále APN), což je serverová služba, která zprostředkovává zabezpečené a efektivní doručení notifikace do systémů iOS, tvOS a macOS [3]. Při prvním spuštění aplikace, která podporuje push notifikace, se systém přihlásí na APN k odběru notifikací, které přijímá ve formátu JSON s maximální velikostí 4KB [8]. K příjmu push notifikací aplikace potřebuje Apple Push Notification Authentification Key certifikát, který vygeneruje autor aplikace pomocí svého vývojářského účtu. Aplikace odebírající push notifikace v konkrétním zařízení je identifikována pomocí Device Tokenu, který získá po ověřeném připojení k APN. Jakmile mobilní aplikace získá Device Token od APN, může jej předat serverové aplikaci jako unikátní identifikátor pro zasílání push notifikací na dané zařízení [3].



Obrázek 13 Zaslání push notifikace od serverové aplikace do mobilní aplikace [3]

|  |  |
| --- | --- |
|  | Praktická část |

# Rozbor architektury projektu

Architektura projektu je základním kamenem celé práce. Nevhodně zvolená architektura by mohla vést k řadě problémů jako je například zdlouhavá implementace, špatná udržitelnost kódu, nebo problémy chodu jednotlivých částí projektu. Návrh architektury vychází z požadavků.

## Požadavky projektu

Požadavkem projektu je multiplatformní mobilní aplikace pro rychlou komunikaci se studenty. Jako rychlou komunikaci rozumíme odesílání a příjem zpráv v reálném čase. K tomu slouží v mobilním světě push notifikace, které jsou popsány v teoretické části. Aplikace musí umožňovat dva následující scénáře:

1. Odeslání rychlé zprávy studentům
2. Odeslání požadavku na hodnocení rozvrhové akce

Pro odeslání rychlé zprávy studentům učitel v aplikaci vybere rozvrhovou akci, u které vybere zaslání zprávy, následně přidá svoji zprávu, případně vybere seznam studentů, kterým má být zpráva doručena a zároveň mají tuto rozvrhovou akci v rozvrhu, a odešle zprávu. Studentům následně dorazí zpráva formou push-notifikace, takže doručení je téměř okamžité.

Pro odeslání požadavku na hodnocení rozvrhové akce učitel vybere rozvrhovou akci, u které vybere požadavek na hodnocení, následně vybere heslo k hodnocení a případně seznam studentů, kterým má být požadavek na hodnocení doručen a zároveň mají tuto rozvrhovou akci v rozvrhu. Studentům následně dorazí práva formou push-notifikace, v jejímž detailu naleznou formulář pro odeslání hodnocení. Učitel poté uvidí hodnocení v detailu rozvrhové akce.

Z požadavků plyne, že aplikace potřebuje back-end s API rozhraním pro posílání a příjem dat. Některá data, jako například rozvrhy, je potřeba brát ze STAG API. Serverový back-end nakonec potřebuje DB k odkládání dat.

### Multiplatformní mobilní aplikace

Pro vývoj multiplatformních mobilních aplikací existuje několik frameworků, které jsou popsány v teoretické části. Jako nejlepší volba se jeví framework Xamarin, ve kterém je možno využít maximum sdíleného kódu jako například celé GUI. Push notifikace je pak potřeba implementovat pro každou platformu zvlášť.

### Cloudový back-end

Implementovat serverovou části projektu by bylo možné v libovolném jazyce jako například PHP, C++ nebo python, ale je potřeba brát ohled na implementaci push-notifikací, komunikaci s DB, nebo taky hosting serveru. Těmto požadavkům nejlépe vyhovuje .NET Core serverová služba, kterou lze implementovat v jazyce C#. Databázi MSSQL lze jednoduše navázat ke službě. O push notifikace se postará framework Notification Hub, který lze použít taktéž v jazyce C#. Pro hosting celé této služby se nabízí cloudová platforma Azure.

## Návrh architektury

// Doplnit něco o návrhu.

a

## 

Obrázek 14 Návrh architektury projektu

## Popis navrhnuté serverové služby

### Serverová služba

Serverová služba se stará o:

* Komunikaci s klientem (Xamarin mobilní aplikace)
* Komunikace se STAG API (prvotní ověřování uživatele, stahování rozvrhů)
* Ukládání dat do vlastní DB (ukládání rozvrhů svázaných s identifikátorem pro push notifikace)
* Odesílání push notifikací pomocí Notification Hub

Pro vytvoření serverové části projektu byl zvolen Framework .NET Core, který je určen pro serverovou činnost. Zvolen byl z důvodu snadného nasazení ve službě Azure [28] a vývoji v jazyce C# a podpoře multiplatformního nástroje pro odesílání push notifikací – Notification Hub. Alternativou by mohla být serverová aplikace běžící na serveru s OS Linux napsaná v jazyce PHP, nebo python a pro odesílání push notifikací by mohla sloužit služba Firebase [9].

### Hosting serveru

Pro hosting serverové služby byl zvolen cloud Azure od společnosti Microsoft. Nasazení služby probíhá z vývojového prostředí Visual Studio.

Důvody výběru hostingu:

* jednoduché nasazení serverové aplikace na cloud
* provázanost cloudu s vývojovým prostředím Visual Studio, které umožňuje testování a nasazení služby přímo z GUI
* rychlá konfigurace výkonu serveru
* snadné nasazení Notification Hub z prostředí Azure
* financování Pay-As-You-Go

Nevýhody hostingu:

* majitel nemá chod serveru pod fyzickou kontrolou, aplikace běží v cloudu
* závislost na cenové politice Azure

### Komunikace se STAG API

Komunikace mezi serverovou službou a STAG probíhá pomocí REST API [27]. Data jsou zpracována ve formátu JSON. Probíhá zde ověřování studentů, učitelů, stahování jejich rozvrhů a stahování informací o rozvrhových akcích.

Seznam použitých služeb:

**getRozvrhByStudent**

Adresa: /ws/services/rest2/rozvrhy/getRozvrhByStudent

Vrátí rozvrh studenta jako rozvrhové akce podle jeho osobního čísla.

**getRozvrhByUcitel**

Adresa: /ws/services/rest2/rozvrhy/getRozvrhByUcitel

Vrátí rozvrh učitele jako rozvrhové akce podle jeho ID učitele.

**getRozvrhovaAkceInfo**

Adresa: /ws/services/rest2/rozvrhy/getRozvrhovaAkceInfo

Vrátí rožšířené info o rozvrhové akci podle ID rozvrhové akce.

**getUciteleRoakce**

Adresa: /ws/services/rest2/rozvrhy/getRozvrhByStudent

Vrátí rozvrh studenta jako rozvrhové akce podle jeho osobního čísla.

### Notification Hub

O zasílání push notifikací se stará služba Notification Hub, která běží na cloudové službě Azure. Serverová služba používá knihovny Notification Hub pro odesílání push notifikací nezávisle na cílové platformě (iOS, Android, Windows, Amazon).

Výhodou Notification Hub:

* sdružení zasílání push notifikací na různé cílové platformy pod jednu službu
* jednoduchá a rychlá implementace
* možnost zasílání push notifikací jak konkrétním zařízením, tak broadcast
* statistiky odesílání/přijímání push notifikací v prostředí portálu Azure

Nevýhody:

* závislost na cloudové službě Azure a na jazyku C#
* nepraktické pro zasílání push notifikací na jednu cílovou platformu

Jako alternativa pro zasílání push notifikací by mohla být použita služba Firebase od společnosti Google, případně implementace pro jednotlivé platformy zvlášť, která by ale byla zdlouhavá.



# Návrh serverové části projektu

Serverová část projektu se stará o ověřování uživatelů, stahování rozvrhů, rozesílání zpráv pomocí push notifikací a vyhodnocování hodnocení výuky. Mobilní aplikaci poskytuje rozhraní pomocí REST API.

## .NET Core služba

.NET Core služba slouží jako API rozhraní pro mobilní aplikaci a zprostředkovává rozesílání push notifikací uživatelům.

// Nějaký další odstavec o .NET Core službě.

## REST API StudentsNotifier AppService

Seznam služeb, které poskytuje .NET Core služba StudentsNotifier:

1. **Message** – posílání a čtení zpráv

[GET] /api/Message/{id}

* + Vrátí zprávu podle ID.

[POST] /api/Message

* + Vloží zprávu do kolekce zpráv a odešle zprávu formou push notifikace příslušným studentům podle jejich ID.

[GET] /api/Message

* + Vrátí seznam všech zpráv. Použito pouze pro správu a ladění.

[DELETE] /api/Message/{id}

* + Smaže zprávu podle ID. Použito pouze pro správu a ladění.

1. **User** – vytváření a čtení uživatelů

[GET] /api/User/{id}

* + Vrátí uživatele podle jeho ID.

[POST] /api/User

* + Vloží nového uživatele včetně tokenu pro odběr push notifikace.

[GET] /api/User

* + Vrátí všechny uživatele. Použito pouze pro správu a ladění.

[DELETE] /api/User/{id}

* + Smaže zprávu podle ID. Použito pouze pro správu a ladění.

[GET] /api/User/RozvrhoveAkce/{id}

* + Vrátí rozvrhové akce uživatele podle ID. Na rozvrhové akce se dotáže ze STAG API podle osobního čísla uživatele, které je nastaveno v instanci uživatele.

1. **LectionRation** – hodnocení výuky

[GET] /api/ LectionRating

* + Vrátí všechny hodnocení výuky.

[POST] /api/ LectionRating

* + Vrátí všechny hodnocení výuky.

[GET] /api/ LectionRating/{id}

* + Vrátí všechny hodnocení výuky.

[DELETE] /api/ LectionRating/{id}

* + Vrátí všechny hodnocení výuky.

[GET] /api/ LectionRating

* + Vrátí všechny hodnocení výuky.

[GET] /api/ LectionRating

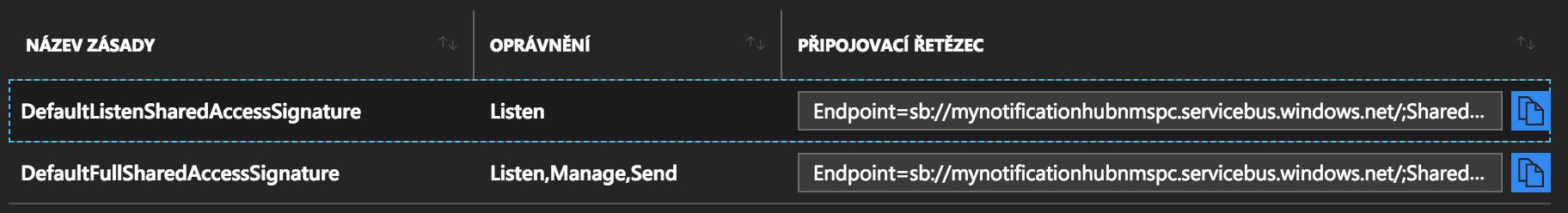
* + Vrátí všechny hodnocení výuky.

# Návrh uživatelského prostředí

# Notifikace

## Notification Hub

Pro aktivaci služby pro iOS je potřeba vložit vygenerovaný certifikát, u androidu API klíč pro odběr push notifikací. Dále nastavit zásady přístupu.



Obrázek 15 Připojovací řetězec Notification Hub pro odběr push notifikací

Připojovací řetězec se následně používá v mobilní aplikaci při prvním spuštění, kde probíhá konfigurace odběru push notifikací.

## Xamarin

// Popis implementace push notifikací v Xamarinu.

# Příklad využití aplikace

text

## Podnadpis

text

Závěr

Vytvořená multiplatformní aplikace splňuje všechny body zadání.

Seznam použité literatury

[1] *Mobile Operating System Market Share Worldwide* [online]. In: . [cit. 2018-03-11]. Dostupné z: <http://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-200901-201803>

[2] *Local and Remote Notifications Overview* [online]. [cit. 2018-03-11]. Dostupné z: <https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/index.html#//apple_ref/doc/uid/TP40008194-CH3-SW1>

[3] *APNs Overview* [online]. [cit. 2018-03-11]. Dostupné z: <https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/APNSOverview.html#//apple_ref/doc/uid/TP40008194-CH8-SW1>

[4] *Google Cloud Messaging: Overview* [online]. [cit. 2018-03-12]. Dostupné z: <https://developers.google.com/cloud-messaging/gcm>

[5] *Use notifications on your iPhone, iPad, and iPod touch* [online]. [cit. 2018-03-12]. Dostupné z: <https://support.apple.com/en-us/HT201925>

[6] *Android platform versions* [online]. [cit. 2018-03-13]. Dostupné z: <https://developer.android.com/about/dashboards/index.html>

[7] IOS Distribution and iOS Market Share. In: *Apteligent* [online]. [cit. 2018-03-13]. Dostupné z: <https://data.apteligent.com/ios/>

[8] *Creating the Remote Notification Payload* [online]. [cit. 2018-03-14]. Dostupné z: <https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/CreatingtheNotificationPayload.html>

[9] *GCM and FCM Frequently Asked Questions* [online]. [cit. 2018-03-14]. Dostupné z: <https://developers.google.com/cloud-messaging/faq>

[10] *Getting Started with iOS* [online]. [cit. 2018-03-14]. Dostupné z: <https://docs.microsoft.com/en-us/xamarin/ios/get-started/index>

[11] *Xamarin.Android* [online]. [cit. 2018-03-14]. Dostupné z: <https://docs.microsoft.com/en-us/xamarin/android/>

[12] Deliver native Android, iOS, and Windows apps, using existing skills, teams, and code. *Xamarin* [online]. [cit. 2018-03-14]. Dostupné z: <https://www.xamarin.com/platform>

[13] *Microsoft to acquire Xamarin and empower more developers to build apps on any device* [online]. [cit. 2018-03-15]. Dostupné z: <https://blogs.microsoft.com/blog/2016/02/24/microsoft-to-acquire-xamarin-and-empower-more-developers-to-build-apps-on-any-device/>

[14] *Mono* [online]. [cit. 2018-03-15]. Dostupné z: <http://www.mono-project.com/>

[15] Compatibility. *Mono* [online]. [cit. 2018-03-15]. Dostupné z: <https://www.mono-project.com/docs/about-mono/compatibility/>

[16] Xamarin.Forms. *Xamarin* [online]. [cit. 2018-03-15]. Dostupné z: <https://www.xamarin.com/forms>

[17] PETZOLD, Charles. *Creating Mobile Apps with Xamarin.Forms* [online]. Redmond, Washington 98052-6399: Microsoft Press, 2016 [cit. 2018-03-15]. ISBN 978-1-5093-0297-0. Dostupné z: <https://developer.xamarin.com/guides/xamarin-forms/creating-mobile-apps-xamarin-forms/>

[18] Performance Comparison: Xamarin.Forms, Xamarin.iOS, Xamarin.Android vs Android and iOS Native Applications. *AlexSoft* [online]. 2017 [cit. 2018-03-15]. Dostupné z: <https://www.altexsoft.com/blog/engineering/performance-comparison-xamarin-forms-xamarin-ios-xamarin-android-vs-android-and-ios-native-applications/>

[19] Responsive Web Design. *A LIST APART* [online]. [cit. 2018-03-15]. Dostupné z: http://alistapart.com/article/responsive-web-design

[20] Unity for Mobile. *Unity3D* [online]. [cit. 2018-03-15]. Dostupné z: <https://unity3d.com/solutions/mobile?utm_source=unity3d-multiplatform&utm_medium=cta>

[21] *PhoneGap* [online]. [cit. 2018-03-15]. Dostupné z: <https://phonegap.com/>

[22] *Flutter* [online]. [cit. 2018-03-15]. Dostupné z: <https://flutter.io/>

[23] Co je Azure?. *Microsoft Azure* [online]. [cit. 2018-03-16]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-is-azure/>

[24] Notification Hubs. *Microsoft Azure* [online]. [cit. 2018-03-16]. Dostupné z: <https://azure.microsoft.com/cs-cz/services/notification-hubs/>

[25] Firebase Cloud Messaging. *Firebase* [online]. [cit. 2018-03-16]. Dostupné z: <https://firebase.google.com/docs/cloud-messaging/>

[26] *An introduction to Azure Functions* [online]. [cit. 2018-03-15]. Dostupné z: <https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview>

[27] Webové služby nad IS/STAG. *Portál IS/STAG* [online]. [cit. 2018-05-05]. Dostupné z: https://stag-ws.utb.cz/ws/web/

[28] *Portál Azure* [online]. [cit. 2018-05-05]. Dostupné z: https://portal.azure.com/

[26]

[26]

Seznam použitých symbolů a zkratek

|  |  |  |
| --- | --- | --- |
| C# |  | Programovací jazyk |
| .NET |  | Framework pro vývoj software |
| iOS |  | Mobilní operační systém od společnosti Apple. |
| macOS |  | Desktopový operační systém od společnosti Apple. |
| GCM |  | Google Cloud |
| API |  | Aplikační interface |
| SQL |  | Structured Query Language |
| LINQ |  | Language Integrated Query |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Seznam obrázků

**Nenašla se položka seznamu obrázků.**

Seznam tabulek

**Nenašla se položka seznamu obrázků.**

Seznam Příloh

P I CD-ROM

Příloha P I: Název přílohy