

# **VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA**

Katedra elektrotechniky a informatiky

Obor: Aplikovaná informatika

## **Semestrální projekt**

Téma: Analýza systému pro instant messaging

Jméno a příjmení: Ondřej Schrek

Vedoucí práce: Ing. Marek Musil

Jihlava 2017

## **Zadání a cíl projektu:**

Analýza systému pro instant messaging. Cílem práce je analyzovat možnosti realizace systému pro instant messaging a provést prvotní návrh pro implementaci vlastního systému, které by mělo zvládat vyšší datové zatížení.

*Podpis vedoucího projektu*

**Předložený semestrální projekt doporučuji k zápočtu.**

*Datum a podpis vedoucího projektu*

# Obsah

---

1	Úvod.....	5
2	Instant messaging.....	6
2.1	Existující softwarová řešení .....	6
2.1.1	Facebook Messenger.....	6
2.1.2	WhatsApp .....	7
2.1.3	Skype .....	8
2.1.4	Slack.....	8
3	Technologická řešení .....	10
3.1	Protokol XMPP .....	10
3.2	Návrh vlastní implementace.....	11
4	Služby typu mBaaS.....	14
4.1	Porovnání současných poskytovatelů .....	14
4.1.1	Firebase .....	14
4.1.2	Kinvey.....	15
4.1.3	Backendless .....	15
4.2	Shrnutí.....	16
5	Firebase .....	17
5.1	Analytics .....	17
5.2	Develop .....	17
5.2.1	Firebase Cloud Messaging.....	17
5.2.2	Authentication.....	18
5.2.3	Realtime Database .....	18
5.2.4	Storage .....	19
5.2.5	Hosting.....	20
5.2.6	Test Lab .....	20

5.2.7	Crash Reporting .....	20
5.3	Grow .....	20
5.3.1	Notifications .....	20
5.3.2	Remote Config .....	21
5.3.3	App Indexing .....	21
5.3.4	Dynamic Links .....	21
5.3.5	Invites .....	21
6	Požadavky na systém .....	22
6.1	Požadavky na klientskou aplikaci .....	22
6.2	Požadavky na serverovou aplikaci .....	22
8	Závěr .....	24
	Seznam použité literatury .....	25
	Seznam obrázků .....	26
	Seznam použitých zkratek .....	27

# 1 Úvod

Tématem tohoto semestrálního projektu je analýza a následná implementace jednoduchého systému pro rychlý přenos zpráv tzv. instant messaging. V dnešní době se jedná o velmi rozšířený segment aplikací, který má již dlouhou tradici mezi uživateli a také tento segment v posledních letech zažil významný technologický pokrok, například díky společnostem jako jsou Facebook, Google, Skype a mnoha dalším, které se aktivně podílejí na zlepšování současných řešení.

Zajímavým aspektem v realizaci vlastního systému je použití Firebase, což je cloudové řešení typu *backend-as-service*, které slouží v systému jako primární backend mobilní aplikace. Jedním z cílů tohoto projektu je popis systému Firebase, mimo jiné také formou vlastní realizace systému. Oboje současně poskytne ucelené informace o tomto způsobu řešení. Jelikož v současné době se zatím nevyskytuje moc ucelených informací, zkušeností či případně případových studií o používání systému Firebase.

Vlastní systém se skládá ze tří segmentů: mobilní klientské aplikace pro Android, nakonfigurovaného systému Firebase a serverové aplikace poskytující rozhraní celému systému a rozšiřuje také možnosti Firebase.

Aplikace může nalézt smysl v nějaké menší organizaci, která potřebuje škálovatelné a otevřené řešení pro osobní komunikaci a šíření hromadných zpráv mezi zaměstnanci. Stejně tak může dobře posloužit jako součást komplexnější aplikace, či jen zůstat jako studijní ukázka s dokumentací pro studenty.

Klientská i serverová část aplikace je vyvíjena v jazyce Java, na serverové části bude použit framework Spring boot.

## **2 Instant messaging**

Instant messaging představuje komunikaci mezi dvěma či více uživateli v reálném čase přes počítačovou síť např. internet. Komunikace může probíhat v textové formě, ale v dnešní době se můžeme setkat s komunikačními programy, které rozšiřují funkčnost o zasílání multimediálních souborů, hlasu či videa. Jedná se o flexibilnější komunikaci, než pomocí emailu, který není primárně určen pro komunikaci v reálném čase. Jednou z dalších výhod oproti emailu je možnost sledování přítomnosti uživatele.

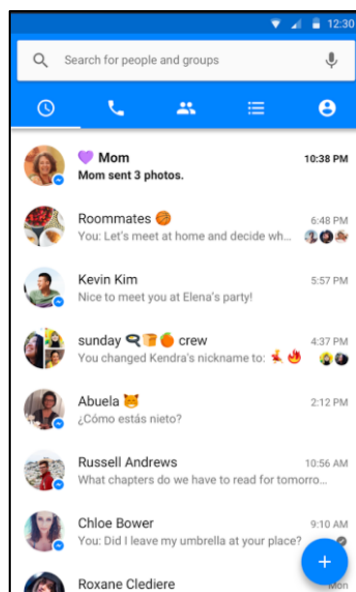
### **2.1 Existující softwarová řešení**

Na současném trhu existuje již mnoho řešení pro instant messaging. V této části práce zhodnotím systémy, které jsem si vybral a se kterými mám již osobní zkušenost a mohu je subjektivně zhodnotit. Testovat budu mobilní verze aplikací, konkrétně na platformě Android.

#### **2.1.1 Facebook Messenger**

Sociální síť Facebook provozuje aplikaci Messenger, která slouží jako komunikační prostředek mezi uživateli sociální sítě. Aplikace je poměrně masivně rozšířená. První verze aplikace byla vydána v srpnu 2011. Aplikace prochází častými updaty a změnami UI. Messenger umožňuje posílání zpráv uživateli či skupině, uskutečňování hovorů, nebo zasílání multimediálního obsahu.

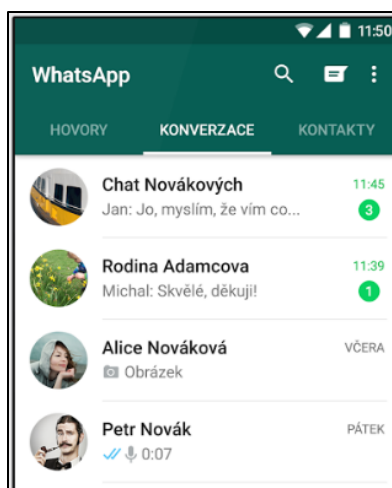
Primární účel aplikace je rozšíření funkčnosti sociální sítě. Při užívání aplikace musíme navíc počítat s podmínkami a politikou, kterou Facebook má např. v souvislosti s autorskými právy, filtrování obsahu atd. Pokud navíc přihlédneme k častým updatům aplikace, které výrazně mění user experiences, lze tvrdit, že to není vhodná komunikační aplikace pro firmu, která hledá stabilní řešení, je spíše vhodnější např. pro zájmovou skupinu či osobní použití.



Obrázek 1: Ukázka aplikace Messenger

### 2.1.2 WhatsApp

Aplikace umožňující výměnu zpráv, zakládání skupinových konverzací a přenosu multimediálních souborů. Jedná se o velmi minimalistickou aplikaci. Mezi její zvláštnosti patří identifikace pomocí telefonních čísel resp. synchronizace kontaktů aplikace s kontakty v telefonu (tel. čísla). Aplikace byla vydána v roce 2009. Aplikaci bych doporučil spíše jednotlivcům, pro firemní účely se moc nehodí, už jen z důvodu existence pouze mobilní verze aplikace.

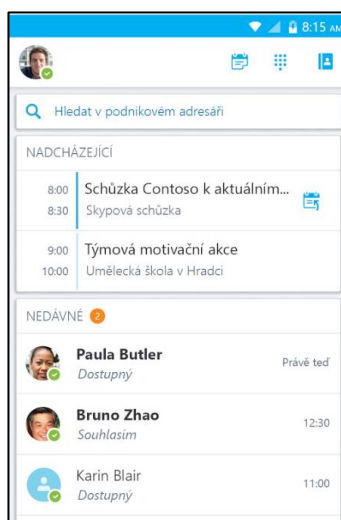


Obrázek 2: Ukázka aplikace WhatsApp

### 2.1.3 Skype

Aplikace umožňující primárně hlasové či video hovory, výměnu zpráv, pořádání skupinových konverzací a přenos multimediálních souborů. Poměrně jednoduchá aplikace s příjemným UI. Výhodou je dostupnost klienta na PC, či online ve webovém prohlížeči. Skype nabízí i customizované řešení pro firmy. První verze se objevila v roce 2003 (Android 2013).

Skype používá vlastní protokol ke komunikaci, který je privátní. Skype lze označit jako vhodné řešení pro firmy i ostatní uživatele. Skype řešení pro firmy je však zpoplatněno (*platí se za každého uživatele*).



Obrázek 3: Ukázka aplikace Skype

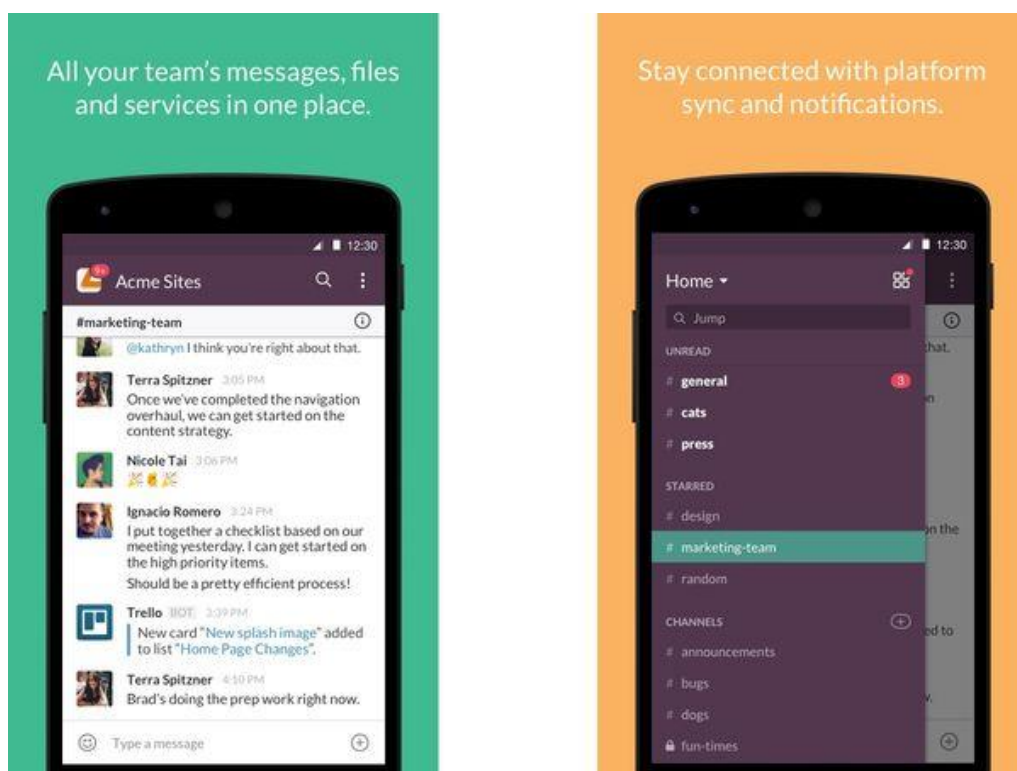
### 2.1.4 Slack

Aplikace speciálně vyvinuta pro týmovou kooperaci, umožňuje posílání zpráv a uskutečňování hovorů mezi uživateli. Slack také umožňuje pohodlně tvořit informační kanály a týmy, ke kterým se uživatelé mohou připojovat a odebírat novinky. Team/kanál má nástěnku na které mohou členové přidávat zprávy či jiný obsah.

Výhodná je i možnost integrace různých informačních zdrojů např. Github, Bitbucket, Twitter. Aplikace je velmi intuitivní a má velmi zdařilou koncepci. Další výhodou Slacku je možnost poměrně velké škálovatelnosti díky API. Aplikace Slacku je dostupná na mnoha platformách. První verze se objevila v roce 2013.



Slack se primárně hodí pro firmy, pro ostatní uživatele pravděpodobně nebude mít velké využití. Nevýhodou Slacku je poměrně vysoká cenová náročnost. V porovnání se Skype, který začíná již na 50 Kč za uživatele měsíčně se Slack pohybuje v částkách od 170 – 323 Kč za uživatele měsíčně. Slack však umožňuje používání ve free režimu s celou řadou omezení, tou nejpálčivější je pouze 10 tisíc zpráv, starší zprávy se odmazávají.



Obrázek 4: Ukázka aplikace Slack

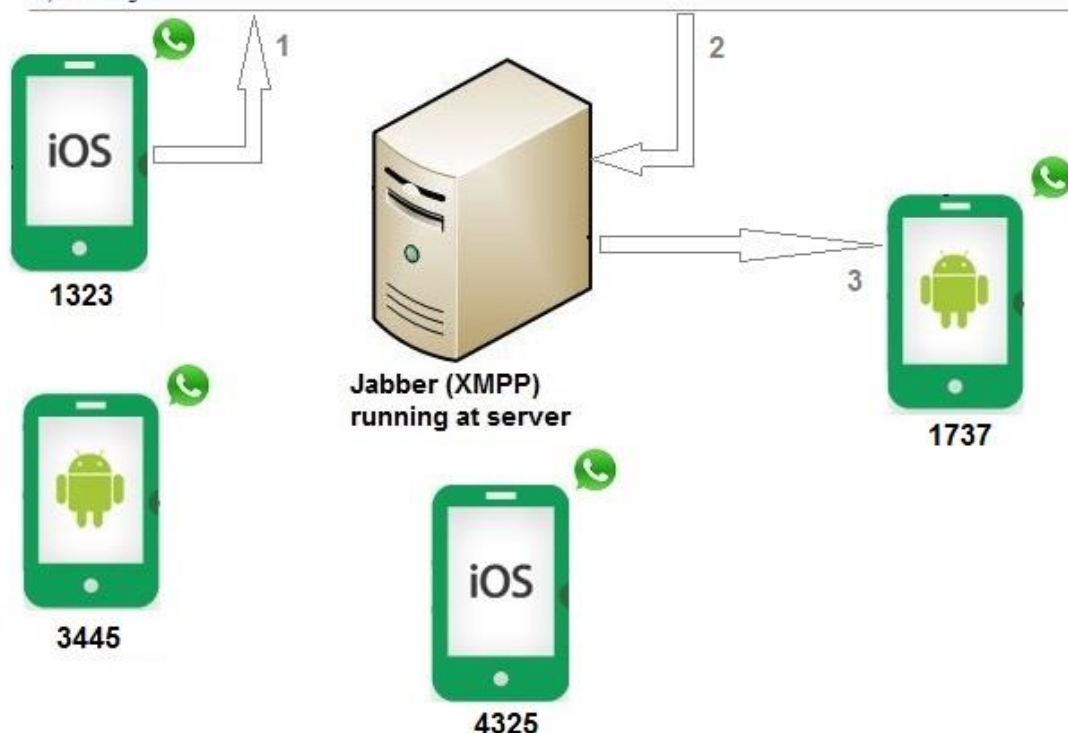
### 3 Technologická řešení

Princip fungování většiny systému poskytující IM lze popsat jako server, či skupinu serverů přes, které probíhá komunikace klientských aplikací. Klient odešle zprávu na server, který zajistí její doručení adresátovi. Většina zmiňovaných aplikací ukládá historii komunikace na server, například Whatsapp je výjimkou, která uchovává komunikaci šifrovaně na klientském zařízení. Většina výše zmíněných aplikací používá ke komunikaci protokol XMPP.

#### 3.1 Protokol XMPP

XMPP (Extensible Messaging and Presence Protocol) je open source komunikační protokol, který ke komunikaci používá formát XML. Primárně slouží k zasílání zpráv typu instant messaging, avšak již od začátku je počítáno s možným rozšířením například o VOIP služby. První zmínky o protokolu se objevují již v roce 1998, v roce 2004 byl protokol standardizován. Protokol je založen na architektuře klient-server. Klienti nekomunikují přímo, ale přes decentralizované servery (podobně jako email). Uživatel má možnost volby serveru, připojí se k tomu, ke kterému má důvěru. Ve světě XMPP neexistuje centrální server, který by spojoval uživatele, avšak díky proprietární implementaci XMPP protokolu ve výše uvedených aplikací nemusí být dogmaticky dodržovány všechny standardy protokolu. Uživatelé musí být jednoznačně identifikovatelní, většinou pomocí tzv. JID (*username@server.com*).

```
<message type="text" to="1737@s.whatsapp.net" from="1323@s.whatsapp.net" id="msg-141">
  <body>Marty, Are we in Madagascar?</body>
</message>
```



Obrázek 5: Ukázka komunikace WhatsApp pomocí XMPP

### 3.2 Návrh vlastní implementace

Během zkoumání stávajících řešení bylo třeba hledat způsoby, jakými by bylo možné vytvořit vlastní systém pro IM, který bych byl schopen vyvinout vlastními silami a přitom aby byl dostatečně robustní.

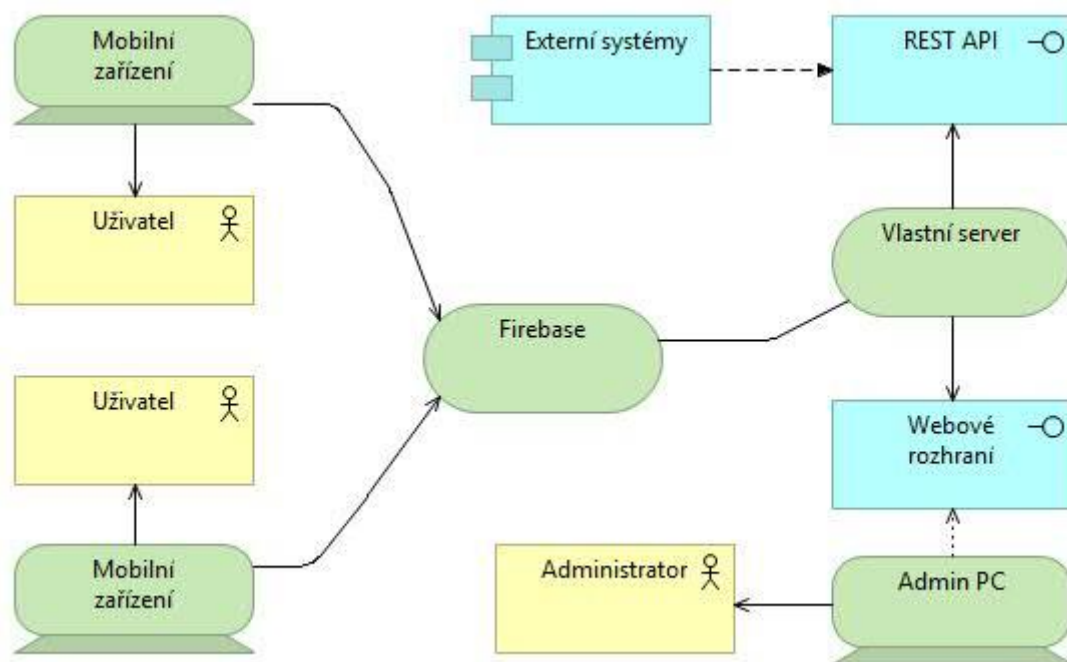
První návrh bylo použití webových služeb, což pod drobnohledem nebyl úplně ideální přístup, než jak se zpočátku jevil. Počáteční výhoda byla poměrně jasná a to jednoduchost řešení. Systém by měl server, který by byl přístupný pod veřejnou adresou a nabízel by určitou paletu potřebných služeb pro klienta. První úskalí spočívalo v samotném fakt, že webové služby nejsou primárně určené pro systémy v reálném čase, jako další problém byla samotná efektivita síťového provozu z klientských aplikací, spočívající v neustálých dotazech na stav/aktualizace. Řešení by mohlo být

například tunelování pomocí VPN ze serveru, který by sám posílal nové události na zařízení, avšak realizace by byla velmi náročná.

Druhé řešení bylo pomocí XMPP protokolu v němž funguje velká spousta reálně nasazených aplikací. Výhoda by byla v poměrně dostupné paletě klientských knihoven pro tvorbu XMPP klienta. Horší situace je však při realizaci serveru XMPP, který se ve většině případů používá jako již existující robustní open source serverové řešení, ke kterému se integrují customizované součásti. Realizace XMPP serveru na „zelené louce“ by byla časově náročná a velmi neefektivní v porovnání s použitím existujícího open source řešení. Pro naše použití je open source řešení poměrně zbytečně robustní, složitě customizovatelné a využili bychom jen malou část jeho funkčnosti.

Třetí a finální technologické řešení bylo použití systému typu mBaaS (mobile Backend as a Service). Toto řešení zažívá v posledních letech poměrně strmý vzestup, jelikož většina potřebné funkcionality na klientských zařízeních je již implementována v cloudu a vývojář je již pouze využívá ve své aplikaci. Tato strategie může poměrně výrazně ušetřit náklady a čas na vývoj serverové části aplikace. Často se může stát, že si plně vystačíme s poskytovanými službami na cloudu a naše aplikace má podporu robustního backendu, bez jediného řádku serverového kódu. Což umožňuje malému týmu, či jednotlivci tvořit poměrně rozsáhlé a komplexní mobilní a webové aplikace.

Výhoda tohoto řešení pro nás je, že nás odstíní od komunikace mezi zařízením a serverem. Aplikace bude bez problému fungovat i ve veřejné síti internetu. Mezi další výhody patří, že nebudeme muset vlastnit server s veřejnou adresou, takže aplikaci může mít nasazenou i organizace, která nemá vybudovanou IT infrastrukturu se servery.



Obrázek 6: Návrh vlastního systému

## 4 Služby typu mBaaS

První mBaaS služby se začali objevovat v roce 2011. Většina poskytovatelů v současné době nabízí poměrně stejnou paletu základních služeb. Poskytují cloudové úložiště pro web či mobilní zařízení, registraci a přihlášení včetně integrace sociálních sítí, push notifikace, analytické nástroje pro mobilní aplikaci. Výhodou je, že služby jsou primárně určeny pro použití v reálném čase.

Mezi nejvýraznější výhody použití mBaaS služeb patří úspora zdrojů v projektu, „oživení“ aplikace, sběr dat, škálovatelnost. K nevýhodám patří proprietární uzamčení tzv. vendor lock, nemožnost kompletní kontroly, možnost nedostupnosti služeb.

Mezi známé uživatele těchto služeb ve svých aplikacích patří např. eBay, Warner Bros., Udacity.

### 4.1 Porovnání současných poskytovatelů

Dnes se můžeme setkat s mnoha poskytovateli backend řešení pro mobilní aplikace. V této části bych rád představil nejznámější zástupce a porovnal jejich výhody a nevýhody. Vybral jsem si tři zástupce a to Firebase, Kinvey, Backendless.

#### 4.1.1 Firebase

Společnost byla založena roku 2011, od roku 2014 ji provozuje společnost Google. Platformy, které Firebase nativně podporuje a jsou pro ně odpovídající SDK: iOS, Android, Web (Angular, Java Script atd.), C++, Unity, Java (pouze pro server). Databáze je zde realizována jako JSON objekt, který můžeme rozšiřovat přidáváním potomků. Firebase neobsahuje geo služby.

Firebase neumožňuje tvoření serverového kódu (známé také jako Parse Cloud Code), k tomu slouží Java knihovna pomocí již, můžeme implementovat server k rozšíření funkčnosti Firebase. Firebase má možnost definovat uživatelské role pro realtime databázi a rolím přiřazovat oprávnění.

Firebase poskytuje poměrně příznivou cenovou politiku. Většina aplikací si vystačí s tarifem Spark, který je zdarma. Google zde nenabízí žádné úlevy pro studenty, open

source, s odvoláním na štědrost tarifu Spark. Další tarif je Flame a pak následuje vlastní customizovaný tarif Blaze, který je přesně na míru uživatele a platí skutečně jen za to, co používá.

#### **4.1.2 Kinvey**

Společnost byla založena roku 2010. Platformy, které Kinvey nativně podporuje a jsou pro ně odpovídající SDK: iOS, Android, Web (Angular, Node.js atd.), Java. Databáze je zde realizována jako JSON objekt, který můžeme rozšiřovat přidáváním potomků.

Pro DBMS je použita MongoDB. Kinvey také neobsahuje geo služby, avšak objekty mohou mít položku geologic property. Kinvey umožňuje tvoření serverového kódu (PCC) pomocí JavaScriptu, nebo triggerů či předdefinovaných funkcí. Kinvey má vertikálně odstupňované oprávnění např. pro databázové záznamy a určené vlastnictví např. pro celou třídu záznamů atd.

Kinvey také nabízí free tarif, který je v porovnání s Firebase poměrně střídmy a placené varianty se nedají škálovat dle potřeby.

#### **4.1.3 Backendless**

Společnost byla založena roku 2012. Platformy, které Backendless nativně podporuje a jsou pro ně odpovídající SDK: iOS, Android, Java Script, Java a jako jediná z vybraných Windows Phone (.NET). Databáze je zde realizována jako klasická relační databáze používající MySQL jako svůj DBMS. Backendless obsahuje geo služby pomocí relace geopointů a datových objektů.

Backendless umožňuje tvoření serverového kódu pomocí jazyka Java, případně se dají použít trigger, předdefinované funkce či timery. Backendless má řízení přístupů na úrovni uživatelských rolí a uplatňování vlastnictví na data a soubory.

Backendless poskytuje poměrně složitou cenovou politiku. Avšak na menší aplikace se lze vejít do Free limitu.

## 4.2 Shrnutí

Souhrně lze říct, že každý poskytovatel poskytovatel má ve své řešení silné a slabé stránky. U Firebase je značná nevýhoda, že nepodporuje desktopové aplikace, jelikož SDK pro Javu je určeno pouze na serverovou aplikaci, díky přítomnosti privátního klíče v aplikaci. Další nevýhoda spočívá v případě nutnosti malé úpravy funkčnosti tvořit serverovou aplikaci, ostatní služby mají alespoň podporu serverového kódu, kterým můžeme upravit jednoduše stávající funkčnost, s velmi malou náročností na zdroje. U ostatních problém spočíval v poměrně složité cenové politice, která je poměrně nepřehledná a částka se u složitější aplikace může poměrně prudce zvýšit.

Pro realizaci systému jsem si vybral Firebase, který i přes zmíněné nevýhody poskytuje štědré free limity. Firebase má pod správou společnost Google, tudíž se dá očekávat dobrá dostupnost a poměrně jistotě a bezpečné uložení dat. Mezi další výhody patří i to, že funkčnost se díky vlastnictví Googlem má tendenci rychle rozšiřovat, tudíž je zde oproti jiným poskytovatelům dlouhodobější perspektiva pro aplikaci.



## 5 Firebase

V této části práce budou představeny funkce systému Firebase, které se vztahují nějakým způsobem k realizované aplikaci. Funkce se dělí do čtyřech skupin: Analytics, která poskytuje data o používání aplikace, druhou je Develop, která poskytuje funkce především vývojářům. Třetí skupina Grow poskytuje funkce pro podporu virálnosti aplikace. Čtvrtá skupina Earn poskytuje prostředky pro monetizaci aplikace, například pomocí reklamy v aplikaci.

### 5.1 Analytics

Základní modul, který musí být vždy přítomný v aplikaci, byť se analytické funkce nepoužívají. Tvoří totiž jádro Firebase. Analytics umožňuje odesílat až 500 různých událostí z aplikace, každá událost může obsahovat až 25 doplňujících atributů. V základu jsou některé běžné události předdefinovány. Analytics nefunguje realtime, ale v rámci úspory baterie zařízení, odesílá data v dávkách zhruba po jedné hodině. Služba má poměrně propracované možnosti analýzy dat v konzoli aplikace.

Analytics jako takové jsou bez omezení, co se týče kapacity uchovávaných dat, avšak takzvané BigQuery pro customizované dotazování již zdarma provádět nelze. V naší aplikaci budeme sledovat zájem uživatelů o jednotlivé části a četnost používání aplikace.

### 5.2 Develop

Tato skupina funkcí bude pro naši aplikaci stěžejní, jelikož nám jde primárně o funkčnost aplikace.

#### 5.2.1 Firebase Cloud Messaging

Funkce umožňující zasílání zpráv do mobilních zařízení z webového rozhraní Firebase, nebo FCM serveru. Zprávy se dělí na notificační (limit 2kB) a datové (limit 4kB) zprávy. Zprávy lze posílat na jednotlivá zařízení, skupiny zařízení, nebo zařízení zapsaná k odběru určitého tématu. V aplikaci tato funkčnost může být použita při zasílání hromadných informací a upozornění na novou zprávu.



Obrázek 7: Schéma FCM

### 5.2.2 Authentication

Tato funkčnost umožňuje integraci přihlašování, která podporuje i přihlašování z jiných sítí. Momentálně podporuje Firebase Google, Facebook, Github, Twitter účty, případně lze poskytnout uživatelům i anonymní účet podle potřeby.

Autentizaci provádí Firebase, který poskytuje vývojáři stejný způsob přístupu k údajům, z různých sítí. Po přihlášení je k dispozici jméno, email, url profilového obrázku uživatele a také identifikátor sítě ze které se uživatel přihlásil. Každému uživateli Firebase přidělí jedinečný user-id. Po úspěšném přihlášení se údaje o uživateli uloží do databáze.

Firebase také umožňuje správu uživatelských session napříč webem a mobilními aplikacemi. V naší aplikaci bude funkce využita pro přihlašování, pomocí emailu a hesla, Facebooku a sítě Google.

### 5.2.3 Realtime Database

Tuto funkčnost bych se nebál označit jako vlajkovou loď celého systému Firebase. Jedná se o poměrně sofistikovaný systém NoSQL databáze, který umožňuje synchronizaci dat mezi zařízeními v reálném čase. Například změna či přidání záznamu

se okamžitě projeví v okně webového prohlížeče i na mobilní aplikaci, bez jakékoliv implementace navíc, ze strany vývojáře.

Databáze je také optimalizována pro práci v offline režimu, který u mobilních zařízení nastává poměrně často a nečekaně, data se v offline režimu ukládají na lokální cache a po získání připojení se automaticky se synchronizují. Databáze umožňuje řízení přístupu jednotlivým uživatelům, či skupinám uživatelů pomocí pravidel, umožňující reflektovat uživatelské parametry (např. user-id) a jiné proměnné v databázi.

Byť je databáze koncipována jako JSON objekt, můžeme ovlivňovat strukturu ukládaných dat pomocí širokého spektra podmínek. V realizované aplikaci bude databáze hrát klíčovou roli a bude se o ni opírat většina funkcí aplikace a celého systému.

Databáze v naší aplikaci bude provozována na tarifu Spark, tudíž se na ní vztahují určitá omezení. Databáze může mít maximálně 100 současně běžících připojení, 1GB velikost uložště, traffic 10GB měsíčně, databáze také neumožňuje automatickou záloh. Žádný z těchto limitů se naší aplikace zásadně nedotkne.

#### **5.2.4 Storage**

Uložiště, které slouží pro ukládní uživatelského obsahu. Oproti klasickému cloudovému úložišti má několik výhod. Umožňuje ochranu před neoprávněným přístupem npř. nepřihlášenému uživateli aplikace. Všechny přenosy, které probíhají mezi uložštěm a zařízením jsou zabezpečené.

Stejně jako u databáze je zde ošetřeno chování při výpadku připojení, tak aby po opětovném připojení pokračovalo stahování dále. Storage podobně jako databáze obsahuje v našem tarifu určitá omezení.

Velikost uložště je 5GB, denní traffic je 1GB a jsme limitováni i počtem operací 20k upload a 50k download. Žádné z těchto omezení by se nemělo dotknout aplikace. Storage může být použita v aplikaci jako uložště pro multimediální zprávy mezi uživateli

### **5.2.5 Hosting**

Hosting představuje klasický webhosting, který neobsahuje žádnou specifickou vlastnost, která by zde stála za zmínku. V aplikaci Hosting nebude použit. Hosting je limitován 1GB uložště a trafficem 10GB za měsíc.

### **5.2.6 Test Lab**

Funkce umožňující otestování naší aplikace na různých zařízeních. Testy mohou být robotické, kdy Firebase objevuje jednotlivé části aplikace a testuje je náhodným způsobem, nebo můžeme dodat vlastní testovací scénáře. Výsledky testů jsou podloženy logy, videem a snímky obrazovky. Aplikace bude využívat základní robotické testování, které by mohlo objevit chyby a snížit čas potřebný na manuální testování aplikace, které by se soustředilo pouze na testování běžného chování uživatele. V našem tarifu jsme omezení 15 testy denně.

### **5.2.7 Crash Reporting**

Umožňuje reportovat pády aplikace v reálném provozu a shromažďovat je do webové konzole aplikace. Součástí reportu je kompletní výpis stack trace, tudíž můžeme lépe lokalizovat pád a jeho příčinu. Do aplikace si také můžeme přidat naše vlastní podpůrné logy, které případně uvidíme při pádu. Tato celá funkčnost je plně zdarma. V naší aplikaci bude tato funkčnost použita.

## **5.3 Grow**

Tuto část funkčnosti představím ve stručnosti, jelikož realizovaný systém nebude využívat nic ze skupiny Grow. Avšak znalost těchto funkcí je velmi vhodná z pohledu marketingu a samotného šíření aplikace, protože dokážou zvýšit viralitu aplikace.

### **5.3.1 Notifications**

Notifications jsou nadstavbou pro výše popsany Cloud Messaging, poskytuje nástroje na zacílení notifikací na určitou skupinu, např. z Analytics. Použití Notifications je vhodné například na cílené nabídky, reklamu, nebo poskytování bonusů pro konkrétní segment uživatelů aplikace.

### **5.3.2 Remote Config**

Remote Config umožňuje nastavovat vzdáleně aplikacím určité parametry, aniž by uživatelé aplikace museli provést update na novější verzi. Může se například hodit postupné uvolňování nové funkčnosti do ostrého provozu, případně při velké chybovosti funkčnost deaktivovat aniž by uživatel musel řešit update na opravenou verzi aplikace.

### **5.3.3 App Indexing**

App Indexing umožňuje propojení aplikace a vyhledávače Google, pokud aplikace obsahuje stejný obsah (klíčová slova) jako webová stránka, je uživatel z mobilního zařízení odkázán speciálním odkazem (Android App Link) do aplikace, ve které se otevře hledaný obsah.

### **5.3.4 Dynamic Links**

Úzce souvisí s Android App Indexing, díky této funkci můžeme odkazovat např. z webu na určitou konkrétní aktivitu aplikace, výhodou je, že pokud uživatel danou aplikaci nemá, tak se automaticky přesměruje na získání aplikace, po nainstalování se aplikace otevře na požadované aktivitě.

### **5.3.5 Invites**

Umožňuje uživateli zasílání pozvání k instalaci aplikace svým přátelům, např. ze soc. sít pomocí emailu, či sms.

## 6 Požadavky na systém

V této části budou popsány hranice realizovaného systému a požadavky na něj. Definovány jsou základní požadavky, které se realizují v první iteraci vývoje. Rozšířené požadavky patří k dalším iteracím vývoje systému, případně jako námět pro studenty k samostatné realizaci.

### 6.1 Požadavky na klientskou aplikaci

Aplikace bude fungovat na mobilní platformě Android, aplikace podporuje od verze 4.4 KitKat, čímž pokryjeme až 85%<sup>1</sup> používaných zařízení v současné době. Primární použití bude na mobilních zařízeních, aplikace tudíž nebude optimalizována na větší obrazovky zařízení např. tablety.

1. Posílání zpráv mezi uživateli
2. Přihlašování
3. Možnost mít přátele
4. Posílání všesměrových a skupinových zpráv
5. Skupinové konverzace
6. Posílání souborů

Body č. 5, 6 patří k rozšíření základní funkčnosti klientské aplikace. Jejich případná absence neovlivní použití aplikace.

### 6.2 Požadavky na serverovou aplikaci

Serverová aplikace bude napsána v jazyku Java a bude využívat framework Spring boot pro tvorbu REST rozhraní, případně Thymeleaf pro webovou administraci. Nasazena bude na aplikační server.

Základní ovládání bude probíhat přes REST rozhraní. V dalším vývojovém inkrementu bude obsahovat jednoduché webové rozhraní, které umožní pohodlnější správu serverové aplikace.

---

<sup>1</sup> <https://developer.android.com/about/dashboards/index.html> k datu 5.1. 2017

1. Správa všesměrového a skupinového zasílání zpráv
2. Správa uživatelů klientské aplikace
  - a. Přidávání
  - b. Mazání
  - c. Úprava
3. Správa zdrojů / skupin
  - a. Přidávání
  - b. Odebírání
  - c. Úprava
4. Správa odběratelů
5. Správa historie (záloha)

Body č. 2, 3, 4, 5 patří k rozšíření základní funkčnosti serverové aplikace. Jeho případná absence neovlivní použití aplikace.

## **8 Závěr**

Předem definované cíle semestrálního projektu se mi povedlo naplnit. Zpracování semestrálního projektu poskytlo úvodní studii problému na, který chci navázat v bakalářské práci a provést implementaci systému.



## Seznam použité literatury

1. KADLEC, Jiří. *REST a webové služby v jazyce Java*. Brno, 2010.
2. SOMMERVILLE, Ian. *Softwarové inženýrství*. Brno: Computer Press, 2013. ISBN 978-80-251-3826-7.
3. NOVÁ, Jana. *Využití webových služeb a práce s nimi*. Plzeň, 2013.
4. Firebase guide. *Firebase* [online]. [cit. 2017-01-16]. Dostupné z: <https://firebase.google.com/docs>
5. WEISS, Petr a Marek RYCHLÝ. *ARCHITEKTURA ORIENTOVANÁ NA SLUŽBY, NÁVRH ORIENTOVANÝ NA SLUŽBY, WEBOVÉ SLUŽBY*. Brno, 2007.
6. EELES, Peter a Peter CRIPPS. *Architektura softwaru*. Brno: Computer Press, 2011. ISBN 978-80-251-3036-0.
7. Backend as a service comparsion. In: *SlideShare* [online]. [cit. 2017-01-16]. Dostupné z: <http://www.slideshare.net/serhiysnizhny/backend-as-a-service-comparison>

## Seznam obrázků

Obrázek 1: Ukázka aplikace Messenger .....	7
Obrázek 2: Ukázka aplikace WhatsApp .....	7
Obrázek 3: Ukázka aplikace Skype .....	8
Obrázek 4: Ukázka aplikace Slack .....	9
Obrázek 5: Ukázka komunikace WhatsApp pomocí XMPP .....	11
Obrázek 6: Návrh vlastního systému .....	13
Obrázek 7: Schéma FCM.....	18

## **Seznam použitých zkratk**

IM – Instant Messaging

JID – Jabber Identifier

mBaaS – Mobile Backend as a Service

SMS – Short Message Service

VOIP – Voice over Internet Protocol

VPN – Virtual Private Network

XMPP – Extensible Messaging and Presence Protocol