

Jihočeská univerzita v Českých Budějovicích
Přírodovědecká fakulta



**Vybraná Google API – užití pro
týmovou spolupráci**

Bakalářská práce

Tomáš Kursch DiS.

Vedoucí práce: Ing. Jiří Jelínek, CSc.

České Budějovice 2014

ZADÁVACÍ PROTOKOL BAKALÁŘSKÉ PRÁCE

Student: Tomáš Kursch
(jméno, příjmení, tituly)

Obor – zaměření studia: Aplikovaná informatika
Katedra: Ústav aplikované informatiky

Školitel: Ing. Jiří Jelínek, CSc.
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Garant z PFF: dtto
(jméno, příjmení, tituly, katedra – jen v případě externího školitele)

Školitel – specialista, konzultant:
(jméno, příjmení, tituly, u externího š. název a adresa pracoviště, telefon, fax, e-mail)

Téma bakalářské práce: Vybraná Google API – užití pro týmovou spolupráci

Cíle práce:

1. Shromáždění informací, příkladů a výukových materiálů týkajících se vybraných Google API (OAuth 2.0, Maps, Calendar, Charts, Analytics, Google+, Search, ...).
2. Návrh aplikace pro týmovou spolupráci užívající uvedená API s použitím UML pro návrh SW.
3. Implementace funkční aplikace podporující spolupráci v týmu, která umožní členům týmu navzájem komunikovat, sdílet dokumenty, delegovat úkoly, organizovat čas a vizualizovat data.
4. Vypracování uživatelské příručky k jednotlivým API ve formě tutoriálu s užitím výstupů bodu 1,2 a 3

Základní doporučená literatura:

1. Google Calendar API. *Google Calendar API - Google Apps Platform; Google Developers* [online]. 20.12.2012 [cit. 2013-06-11]. Dostupné z: <https://developers.google.com/google-apps/calendar/>
2. Google Tasks API. *Google Tasks API - Google Apps Platform Google Developers* [online]. 14.11.2012 [cit. 2013-06-11]. Dostupné z: <https://developers.google.com/google-apps/tasks/>
3. Google+ API. *Google+ API - Google+ Platform Google Developers* [online]. 26.2.2013 [cit. 2013-06-11]. Dostupné z: <https://developers.google.com/+api/>
4. Google Accounts Authentication and Authorization. *Using OAuth 2.0 to Access Google APIs* [online]. 16.2.2013 [cit. 2013-06-12]. Dostupné z: <https://developers.google.com/accounts/docs/OAuth2>

Financování práce :
Vedoucí práce : podpis : *J. J. J.*
U externích vedoucích fakultní garant práce..... podpis :
Garant oboru bak. studia (nepožaduje se u zaměření „příprava na mag. studium biologie)
..... podpis :
Vedoucí katedry podpis : *Am*
Případný souhlas vedoucího ústavu AV podpis :

V Českých Budějovicích dne *29. 11. 2013*
Převzal/a dne *29. 11. 2013* podpis : *msc*

Bibliografické údaje

Kursch T., 2014: Vybraná Google API – užití pro týmovou spolupráci.

[Selected Google API – usage for team collaboration. Bc.. Thesis, in Czech.] – 70 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Anotace

Tato bakalářská práce se zabývá problematikou Google API a jejich využití pro týmovou spolupráci. Společnost Google patří mezi nejznámější společnosti světa, jež provozuje, kromě nejpoužívanějšího internetového vyhledávače, řadu dalších služeb, které dnes může využívat prakticky kdokoliv. Každá nabízená služba navíc poskytuje aplikační programové rozhraní, které umožňuje vývojářům webových aplikací integrovat a přizpůsobit tyto služby do vlastní aplikace. Cílem bakalářské práce je shromáždění dostupných informací o vybraných službách společnosti Google a jejich aplikačních rozhraních. Vybraná API budou dále integrována do vlastní webové aplikace, která bude sloužit jako nástroj usnadňující týmovou spolupráci. K jednotlivým Google API bude k práci přiložena uživatelská příručka ve formě tutoriálu, představující srozumitelný návod, jak integrovat vybraná API do vlastní webové aplikace.

Annotation

This thesis deals with the Google APIs and its use for team collaboration. Google is one of the most famous companies in the world, which runs the Internet's dominant search engine in addition to many other services, that can use today virtually anyone. Each service is offered in addition provides an application programming interface that allows web application developers to integrate and customize these services into their own applications . The goal of the thesis is the gathering of information available on selected Google services and their application interface. Selected API will be further integrated and adapted to their own web application that will serve as a tool for easing team collaboration . Individual Google APIs will have enclosed user manual in the form of tutorial, presenting a clear and simple instructions, how to integrate the selected API into own web application.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích dne 11. 12. 2014

Obsah

1	Úvod.....	1
1.1	Obsah práce.....	1
1.2	Cíle práce.....	2
2	Teoretická část.....	4
2.1	Společnost Google a její produkty a služby.....	4
2.2	Princip Google API.....	7
2.3	Možnosti přístupu k Google API ve vlastní aplikaci.....	8
2.3.1	Klientské knihovny.....	8
2.3.2	Google API Loader.....	9
2.4	Autentizace a autorizace Google API.....	10
2.4.1	OAuth 2.0 pro serverové aplikace.....	11
2.4.2	OAuth 2.0 pro klientské aplikace.....	11
2.4.3	Google+ Sign-in.....	12
2.5	Vybrané webové služby a jejich API.....	14
2.5.1	Google Tasks.....	14
2.5.2	Google Kalendář.....	15
2.5.3	Google+.....	16
2.5.4	Google Drive.....	18
2.5.5	Google Dokumenty.....	19
2.5.6	Google Kontakty.....	20
2.5.7	Google Maps.....	21
2.5.8	Google Web Search.....	22
2.5.9	Google Cloud Datastore.....	23
2.5.10	Google Analytics.....	25
2.5.11	Google Hangouts.....	26
2.5.12	Google Charts.....	27
3	Praktická část.....	29
3.1	Analýza a návrh aplikace.....	29
3.1.1	Diagram případů užití.....	30
3.1.2	Diagram tříd.....	31
3.2	Implementace a testování aplikace.....	32
3.2.1	Programovací jazyky a standardy.....	33
3.2.2	Knihovny.....	34
3.2.3	Frameworky.....	35
3.2.4	Vývojové prostředí a platformy.....	35
3.2.5	Testování.....	36
3.3	Tvorba uživatelské příručky.....	37
4	Závěr.....	38
5	Literatura.....	40
6	Přílohy.....	42

1 Úvod

Společnost Google nabízí velké množství služeb zjednodušujících organizování času, sdílení dokumentů, komunikaci, vizualizaci dat i plánování úkolů, proto jsem si zvolil práci na téma: „Vybraná Google API – užití pro týmovou spolupráci“, ve které se budu zabývat vybranými službami usnadňujícími týmovou spolupráci a jejich integraci do vlastní webové aplikace pomocí aplikačních rozhraní nabízených společností Google.

Google API je aplikační programové rozhraní sloužící k dotazování Google databáze pomáhající programátorům při vývoji vlastních aplikací. Google API se v podstatě skládá ze specializovaných webových služeb, programů a skriptů, které umožňují vývojářům webových aplikací lepší vyhledávání a zpracování informací na webu.

Těchto aplikačních programových rozhraní nabízí Google velké množství. V současnosti je jich kolem osmdesáti. API jsou nabízena zdarma, ale obsahují kvóty, při jejichž překročení dochází k omezení funkčnosti dané API až do druhého dne. Kvóty lze zjistit v Google Developers Console, kde se navíc všechny API dají aktivovat.

Většina Google API nabízí přístup k téměř všem funkcím dané služby, ale některá API jsou velmi omezená a nabízejí mnohem méně možností než daná služba.

1.1 Obsah práce

V teoretické části práce nejprve představím společnost Google, její přínos v informatice a neznámější produkty a služby. Dále se již budu zabývat samotnými Google API, kde vytvořím přehled všech dostupných Google API. Zároveň popíši, na jakém principu fungují, jaké jsou možnosti integrace daných API do vlastní aplikace a autentizačními mechanismy. V poslední části se budu zabývat vybranými webovými službami společnosti Google a jejich API. Nejdříve samotnou webovou službu a její API podrobněji charakterizují a poté rozeberu jejich využitelnost pro týmovou spolupráci.

V praktické části práce podrobněji představím, jaké nástroje jsem použil a jakým způsobem vytvořil aplikaci využívající vybraná Google API, včetně návrhu aplikace v grafickém jazyce UML. Nakonec stručně popíši obsah vlastní uživatelské příručky k jednotlivým API. Příručka bude ve formě tutoriálu.

1.2 Cíle práce

- ***Shromáždění informací, příkladů a výukových materiálů týkajících se vybraných Google API (OAuth 2.0, Maps, Calendar, Charts, Analytics, Google+, Search, ...).*** Cílem práce je shromáždit informace a materiály týkající se Google API a uspořádat je do celku ve formě bakalářské práce. Čtenáři by měla poskytnout ucelený obraz, co to vlastně jsou Google API, jaké jsou jejich výhody, případně nevýhody, a seznámit vývojáře s možnostmi jejich implementace do vlastní aplikace.
- ***Návrh aplikace pro týmovou spolupráci užívající uvedená API s použitím UML pro návrh SW.*** S využitím programovacího jazyka UML vytvořím návrh aplikace s ohledem na robustnost, spravovatelnost a použitelnost aplikace.
- ***Implementace funkční aplikace podporující spolupráci v týmu, která umožní členům týmu navzájem komunikovat, sdílet dokumenty, delegovat úkoly, organizovat čas a vizualizovat data.*** Výstupem práce bude mimo jiné aplikace, jež bude jednoduchým interaktivním nástrojem pro podporu spolupráce v týmu. Uživatel využívající aplikaci bude mít možnost vytvořit nový projekt, vytvořit k němu úkoly, které se musí vykonat a k nim přiřadit členy týmu, kteří budou dané úkoly vykonávat. Členové týmu budou moci vytvářet nové události pomocí Google Calendar API, případně zobrazit polohu konané události prostřednictvím Google Maps API. V aplikaci bude možné zobrazit polohu členů týmu na mapě. Úkoly bude možné delegovat pomocí Google Tasks API. Členové týmu budou moci navzájem komunikovat prostřednictvím Google+

API. Pro sdílení dokumentů použiji API Google Drive. Pro autentizaci uživatelů využiji Google Oauth 2.0 API.

- ***Vypracování uživatelské příručky k jednotlivým API ve formě tutoriálu s užitím výstupů bodu 1,2 a 3.*** K vybraným Google API vytvořím uživatelskou příručku, jež ve formě tutoriálu popíše, jakým způsobem lze integrovat konkrétní API do vlastní aplikace s využitím klientských knihoven. Uživatelská příručka bude vycházet z vlastní aplikace, kterou vytvořím v rámci bakalářské práce.

2 Teoretická část

2.1 Společnost Google a její produkty a služby

Společnost Google byla založena v roce 1998. Zakladatelé Larry Page a Sergej Brinn se seznámili na Stanfordově univerzitě v roce 1995 a již v roce 1996 vytvořili vyhledávač nazvaný „BlackRub“. Tento vyhledávač byl nakonec přejmenován na „Google“, název vznikl pozměněním slova „googol“, jež představuje matematické označení jedničky a sto nul. [1]

Postupem času se Google začal rozvíjet, až se z něj stala jedna z největších společností na světě. Kromě vyhledávače nabízí velké množství webových služeb a v současné době je jedničkou na trhu díky svému operačnímu systému Android pro přenosná zařízení.

V následujícím seznamu se nacházejí nejznámější produkty a služby, jež společnost Google poskytuje veřejnosti. Tabulky obsahují většinu nabízených API společností Google, z nichž tučně zvýrazněné jsou podrobněji popsány v rámci této práce.

Web

- **Vyhledávač** - Nejznámější vyhledávač s možností vyhledávat fulltextově na webu, dále pak umožňuje hledat obrázky, videa, zprávy a knihy.
- **Google Chrome** - Prohlížeč vytvořený s důrazem na rychlost, jednoduchost a bezpečnost.

Analytics API	Chrome Web Store API	DoubleClick Search API
Webmaster Tools API	PageSpeed API	Prediciton API
Safe Browsing API	Site Vertification API	URL Shortener API
Web Fonts Developer API	Web Search API	

Mobilní zařízení

- **Android** - Android je operační systém založený na linuxovém jádře, který vznikl zejména pro mobilní zařízení (chytré telefony, PDA, navigace, tablety, fotoaparáty).

Apps Marketplace API	Cloud Messaging API	Maps Android API
Maps SDK for iOS	Play Android Developer API	Play App State API

Firemní

- **AdWords** - Služba umožňující inzerovat vlastní reklamy.
- **Google Apps pro firmy** - G-mail, online úložiště, sdílené kalendáře, videokonference a další možnosti. Vytvořeno pro firmy a zaměřeno na týmovou práci. Jedná se o placenou službu.

Ad Exchange API	Admin SDK	AdSense API
Audit API	Google+ Domains API	Google Apps Reseller API

Média

- **Youtube** - Sledování, nahrávání a sdílení videí.
- **Picasa** - Vyhledávání, sdílení a upravování fotografií.

Books API	Play Game Management API	Play Game Services API
Youtube Analytics API	Youtube Data API v3	

Zeměpisné informace

- **Mapy** - Zobrazení map a plánování tras.
- **Earth** - Prozkoumání celého světa.
- **Panoramio** - Prozkoumávání a sdílení fotografií z celého světa.

Directions API	Distance Matrix API	Elevation API
Geocoding API	Maps Coordinate API	Maps Embed API
Street View API	Maps Geolocation API	Maps JavaScript API
Time Zone API	Places API	Static Maps API

Specializované služby

- **Vyhledávání blogů** - Nalezení blogů, které se zabývají danými tématy.

- **Scholar** - vyhledávání vědeckých článků
- **Upozornění** - Možnost odebírat e-mailem novinky s tématy dle vlastního výběru.

FreeBase API	Civic Information API	Cloud DNS API
Cloud monitoring API	Compute Engine API	Identity Toolkit API
Google Picker API	Mirror API	

Domácnost a kancelář

- **Gmail** - Rychlý email s vyhledáváním.
- **Dokumenty** - Vytváření a sdílení dokumentů, prezentací a tabulek online.
- **Kalendář** - Zorganizování událostí a sdílení událostí s přáteli.

Calendar API	Charts API	Drive API
Fusion Tables API	Gmail API	Tasks API
Translate API		

Komunita

- **Blogger** - Možnost vytvořit a sdílet vlastní blog.
- **Hangouts** - Skupinová online konverzace ve formě chatu či videokonference.
- **Skupiny** - Možnost vytvářet diskusní skupiny či přidávat příspěvky.

Blogger API	Contacts API	Google+ API
Google+ Hangouts API	Orkut REST API	Groups Settings API

Databáze

- **Datastore** – bezplatná databáze v cloudu
- **BigQuery** – analýza velkého množství dat

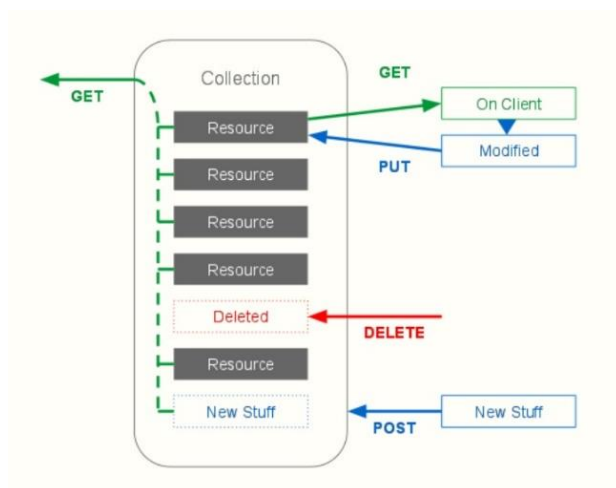
BigQuery API	Genomics API	Cloud Datastore API
Cloud SQL API	Cloud Storage API	Spectrum Database API

2.2 Princip Google API

Google API jsou založena na REST (Representational State Transfer) architektuře rozhraní, navržené pro distribuované prostředí. Rozhraní REST je použitelné pro jednotný a snadný přístup ke zdrojům (Resources). Zdrojem mohou být data, stejně jako stavy aplikace (pokud je lze popsat konkrétními daty). Všechny zdroje mají vlastní identifikátor (URI) a REST definuje čtyři základní metody pro přístup k nim. [2]

- **GET** - základní metoda pro získání zdroje, pro Google API metoda **get**
- **POST** - pro vytvoření zdroje slouží metoda POST, pro Google API metoda **insert**
- **DELETE** – zdroj lze smazat metodou DELETE, pro Google API metoda **delete**
- **PUT** - Operace změny je podobná operaci vytvoření (metoda POST), s tím rozdílem, že voláme konkrétní URI konkrétního zdroje, který chceme změnit a v těle předáme novou hodnotu, pro Google API metoda **update**

Výměna dat mezi klientem a serverem se provádí ve formátu JSON nebo XML.



Obrázek 1: REST architektura rozhraní¹

¹ Dostupné na <http://www.cs.nott.ac.uk/~rmm/teaching/g54acc/2010/material/googleapis-how-google-builds-apis.pdf>.

2.3 Možnosti přístupu k Google API ve vlastní aplikaci

Chceme-li využít různá Google API v naší aplikaci, potřebujeme k tomu klientské knihovny nebo lze využít služby Google API Loader. Klientské knihovny jsou dostupné v mnoha programovacích jazycích, za to Google API Loader je postaven pouze na jazyku JavaScript.

2.3.1 Klientské knihovny

Klientské knihovny byly vytvořeny společností Google za účelem usnadnění přístupu k požadovaným Google API ve vlastní aplikaci.

Společnost Google vyvíjí své klientské knihovny pomocí Google APIs Discovery Service. Discovery Service převede metadata o Google API do strojově čitelného formátu pro klientské knihovny a další nástroje. [3]

Mezi největší výhody Google APIs Discovery Service patří:

- Škálovatelnost
- Automatizace
- Transparentnost

Klientské knihovny jsou dostupné v následujících programovacích jazycích:

.NET, Go, Google Web Toolkit, Java, JavaScript, Node.js, Objective-C, PHP, Python a Ruby

Pro každý podporovaný jazyk klientské knihovny poskytují:

- Samotné knihovny (soubor tříd umožňující interakci s Google API)
- Jednoduché ukázky aplikací využívající dané API
- Dokumentaci ke všem třídám a metodám

Klientské knihovny pro PHP byly dříve distribuovány pouze jako součást frameworku Zend, v současné době je již možné stáhnout klientské knihovny samostatně, tudíž nejsme limitováni pouze daným frameworkem. Klientské knihovny pro PHP se stále vyvíjejí a nyní se nacházejí v Beta verzi. Výhodou klientských knihoven pro PHP je podpora autentizačního mechanismu OAuth 2.0 a dalších 48 různých Google API.

Následující ukázkou načteme Google Tasks API do vlastní aplikace s využitím klientských knihoven pro PHP. Objekt „client“ slouží k autentizaci, o které pojednává další kapitola.

```
$client = new Google_Client();
$client->setClientId('clientID');
$client->setClientSecret('clientSecret');
$client->setRedirectUri('redirectURI');
$client->addScope("https://www.googleapis.com/auth/tasks");
$service = new Google_Service_Tasks($client);
```

2.3.2 Google API Loader

Google API Loader umožňuje jednoduše importovat API do vlastní aplikace. K importování dané API stačí přidat následující skript do hlavičky webové stránky.

```
<script type="text/javascript"
src="https://www.google.com/jsapi"></script>
```

```
<script type="text/javascript">
  google.load("search", "1");
  google.load("jquery", "1.4.2");
</script>
```

Funkci `google.load` voláme s 2 argumenty. Prvním je modul, který volá specifické API, kterou si přejeme použít, a druhým je verze dané API.

Google API Loader lze využít k importování různých knihoven, například v dnešní době populární JQuery.

Google API Loader podporuje pouze několik Google API. Přehled těchto API je v následujícím seznamu, ve kterém ty kurzívou vyznačené jsou dále použity v rámci této práce. [4]

- *Google Web Search API, Google Picker API, Google Charts API*
- Google Data APIs, Google Feeds API, Orkut API, Google Earth API

2.4 Autentizace a autorizace Google API

Google API využívají protokolu OAuth 2.0 pro autentizaci a autorizaci. Google podporuje OAuth 2.0 pro aplikace serverové, nainstalované i klientské. [5]

K využití OAuth 2.0 je nutné nejprve získat takzvané „Client ID“ a k němu další potřebné autentizační údaje, které si vygenerujeme na stránkách „Google Developers Console“. Všechny potřebné údaje se vygenerují samy, kromě „Redirect URI“ (stránka, na kterou chceme přesměrovat po autorizaci) a „JavaScript Origins“ (adresa, na které může vlastní aplikace přistupovat k Google API).

Následující seznam znázorňuje ukázkou vygenerovaných autentizačních údajů. Přesný návod k získání těchto údajů se nachází v uživatelské příručce přiložené k této práci.

- **Client ID** – 123456789.apps.googleusercontent.com
- **Email Address** – 123456789@developer.gserviceaccount.com
- **Client Secret** – 24HgV_WHfs5Kl4KCLBjopIvN
- **Redirect URI** - http://localhost/oauth2callback
- **Javascript Origins** - http://localhost/
- **API Key** – AIzaNyAd_HBLTrfUK9iHdBj5xWz82NrGzKHVEza

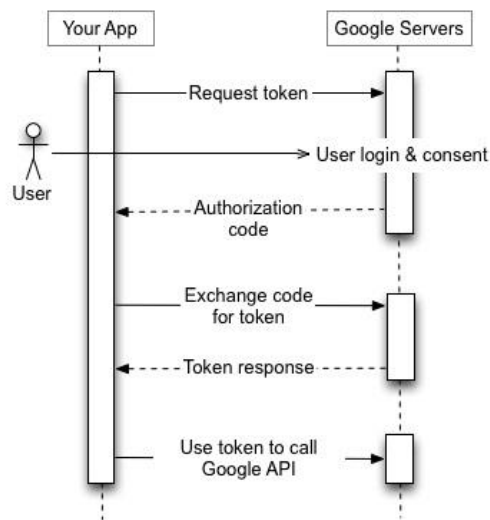
API klíč (API key) je možné použít pouze pro autentizaci, kde není nutné přistupovat k soukromým uživatelským datům (například pro využití Google Maps API ve své aplikaci není nutná autorizace pomocí OAuth 2.0 a stačí tuto API pouze autentizovat API klíčem).

Princip autorizace Google API je vcelku jednoduchý, mezi aplikací a Google serverem proběhne autorizační sekvence, na jejímž konci získá aplikace přístupový token, který využívá pro volání Google API. Autorizační sekvence se liší dle přístupu na straně klienta či serveru a je podrobněji vysvětlena v kapitole 2.5.1 a 2.5.2.

Přístupový token má omezenou životnost na šedesát minut. Pokud aplikace potřebuje přístup k Google API delší, než je životnost přístupového tokenu, lze vyžádat takzvaný „refresh token“, který umožní aplikaci automaticky vyžadovat nové přístupové tokeny.

2.4.1 OAuth 2.0 pro serverové aplikace

OAuth 2.0 podporuje webové serverové aplikace využívající jazyky a frameworky PHP, Java, Python, Ruby a ASP.NET. Autorizační sekvence začíná, když vlastní aplikace přesměruje prohlížeč na adresu Googlu. Google převezme autentizaci uživatele a jeho souhlas k přístupu. Výsledkem je autorizační kód, který aplikace může vyměnit za přístupový token.

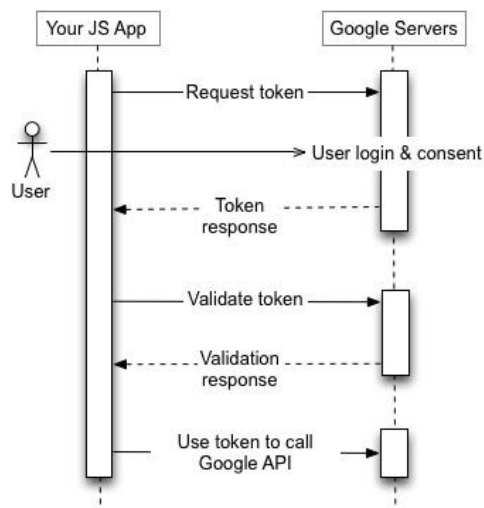


Obrázek 2: Autorizační sekvence pro serverové aplikace ²

2.4.2 OAuth 2.0 pro klientské aplikace

OAuth 2.0 též podporuje JavaScriptové aplikace běžící v prohlížeči. Princip je podobný jako u serverových aplikací s tím rozdílem, že místo autorizačního kódu Google server odesílá rovnou přístupový token, tudíž není možné využít „refresh tokenu“.

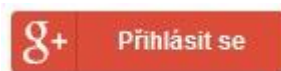
² Dostupné na <https://developers.google.com/accounts/docs/OAuth2>



Obrázek 3: Autorizační sekvence pro klientské aplikace ³

2.4.3 Google+ Sign-in

Pro autentizaci a autorizaci Google API ve své aplikaci si můžeme vytvořit vlastní odkaz či tlačítko nebo lze využít Google+ „Sign-in“ tlačítka, které přímo využívá OAuth 2.0 protokolu k vygenerování přístupového tokenu.

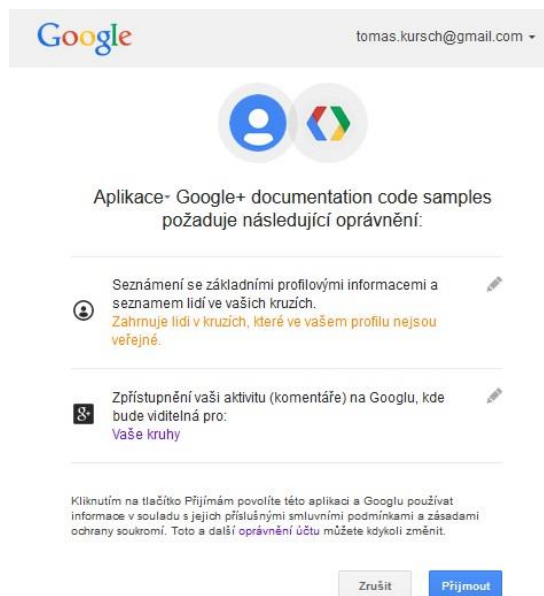


Obrázek 4: Google+ Sign-in tlačítko ⁴

Princip je vcelku jednoduchý, po kliknutí na tlačítko se uživateli zobrazí autorizační okno, kde se dozví, jakým způsobem aplikace využije jeho data. Uživatel má dále možnost potvrdit autorizaci či odmítnout přístup. Po autorizaci se navracející uživatel již nebude muset autorizovat znovu, avšak má možnost kdykoliv k aplikaci odebrat přístup.

³ Dostupné na <https://developers.google.com/accounts/docs/OAuth2>

⁴ Dostupné na <https://developers.google.com/+/web/signin/add-button>



Obrázek 5: Autorizační okno

Implementace autentizačního mechanismu pomocí „Sign-in“ tlačítka se liší dle přístupů k Google API, které mohou být celkem tři. Prvním je přístup na straně klienta využívající JavaScript a HTML (vhodné pro práci s Google API v jazyce JavaScript). Druhým způsobem je hybridní přístup (autentizace na straně klienta a práce s Google API na straně serveru). Třetím a posledním způsobem je čistě serverový přístup (autentizace i práce s Google API na straně serveru). [6]

2.5 Vybrané webové služby a jejich API

Společnost Google nabízí velké množství webových služeb, proto tato kapitola popisuje pouze vybrané služby a k nim dostupné API. Služby byly vybrány z hlediska potenciálu pro týmovou spolupráci a kromě popisu funkce dané služby a jejího aplikačního programového rozhraní se dozvíme, jakým způsobem lze danou službu či API využít pro týmovou spolupráci.

Nejlepším zdrojem informací o webových službách jsou oficiální stránky společnosti Google obsahující podrobný popis služeb včetně nápovědy. [7]

2.5.1 Google Tasks

Google Tasks jsou jednoduchým soukromým plánovačem úkolů. Google Tasks jsou integrovány jako součást Google Kalendáře i Gmailu. Lze k nim však přistupovat i z vlastní aplikace pomocí Google API.

Úkoly v Google Tasks lze vytvářet, mazat, označit jako dokončené či je znovu aktivovat. Mezi pokročilejší funkce patří vytváření víceúrovňovým seznamů a měnění pořadí úkolů. Každému úkolu lze přiřadit datum dokončení a poznámku.

Google Tasks bohužel nelze plnohodnotně sdílet, jediným dostupným řešením v současné době je odeslání seznamu úkolů pomocí Gmailu. Společnost Google slibuje tuto funkci plnohodnotného sdílení v budoucnosti vytvořit, zatím však nelze jednotlivé úkoly sdílet s ostatními.

Google Tasks API

Google Tasks API operují s dvěma typy zdrojů, konkrétně se jedná o „Task List“ a „Task“. Zdroj je samostatná entita s unikátním identifikátorem. Zdroje můžeme přidávat do kolekcí. Každý uživatel má alespoň jeden seznam úkolů pojmenovaný default. [8]

Na kolekce a zdroje můžeme zavolat pět různých metod.

- **list** – získá všechny zdroje v rámci kolekce
- **insert** – vytvoří nový zdroj a uloží jej do kolekce
- **get** – získá specifický zdroj
- **update** – aktualizuje specifický zdroj
- **delete** – smaže specifický zdroj

Následující ukázka získá všechny úkoly ze základního seznamu úkolů.

```
$taskService->tasks->listTasks('@default');
```

2.5.2 Google Kalendář

Google Kalendář je pomocníkem pro organizaci času. Google Kalendář funguje jako samostatná aplikace. Hlavní funkcí kalendáře je vytváření událostí. Událostem lze přiřadit čas, polohu, popis a připomenutí emailem či vyskakovacím oknem.

Velkou výhodou kalendáře je možnost jeho sdílení. Lze sdílet jak celý kalendář, tak i jednotlivé události. Kalendář lze učinit veřejným, to znamená, že má k němu přístup kdokoliv nebo jej sdílet s konkrétními lidmi a přidělit jim různá oprávnění.

Lze sdílet i samostatné události. Událost můžeme nastavit jako veřejnou nebo soukromou. K veřejné události má přístup kdokoliv, takže si může prohlédnout veškeré podrobnosti jednotlivých událostí. Soukromá událost bude pro ostatní skryta.

Google Calendar API

Google Calendar API umožňuje ve vlastní aplikaci vykonat většinu operací dostupných přes webové rozhraní Google kalendáře. Neautentizovaní uživatelé mohou použitím této API vyhledávat a prohlížet veřejné události. Autentizovaní uživatelé mohou také přistupovat k vlastním soukromým kalendářům a stejně tak je vytvářet, editovat, mazat a to včetně událostí v kalendářích obsažené.

Google Calendar API je založeno na pěti různých zdrojích. [9]

- **Events** – události
- **Calendars** - kalendáře
- **CalendarList** – seznam kalendářů

- **Settings** – uživatelské nastavení pro daný kalendář
- **ACL** – přístupová práva kalendáře

Kalendáře lze sdílet s ostatními uživateli pomocí ACL, kde můžeme ostatním uživatelům přiřadit role „**none**“ (žádný přístup), „**reader**“ (povolení čtení událostí), „**writer**“ (povolení čtení a zápisu), „**owner**“ (vlastník kalendáře).

Následující ukázka smaže konkrétní událost z vybraného kalendáře.

```
$calendar->events->delete('calendarID', 'eventID');
```

2.5.3 Google+

Google Plus je sociální síť podobná neznámější sociální síti facebook. V Google Plus lze přidávat nové příspěvky či je komentovat. Dále lze sdílet vlastní fotografie, odkazy, videa a události.

Sdílet nový obsah je možné veřejně, pomocí kruhů nebo pouze s konkrétními lidmi. Důležitým aspektem Google Plus jsou již zmiňované kruhy přátel. Každého člověka mající Google Plus účet lze přiřadit do vlastního kruhu. Lze vybrat z již předdefinovaných kruhů (Přátelé, Rodina, Známí, Pouze sledovaní) nebo lze vytvořit kruh nový s vlastním názvem. [10]

Stejně jako na facebooku lze i na Google Plus vytvořit vlastní profil. Můžeme si přidat vlastní fotografii, napsat pár slov o sobě, vyplnit údaje o zaměstnání, vzdělání, bydliště, kontaktní údaje, ale i odkazy na vlastní youtube kanály či aplikace.

Google Plus obsahuje několik zajímavých funkcí, kterými se odlišuje od ostatních sociálních sítí. Funkce „O čem se mluví“ zobrazuje oblíbené a zajímavé příspěvky, které jsou na Google Plus často sdíleny. Další funkcí jsou „Události“. Vytváření událostí je podobné jako u Google kalendáře, ale oproti kalendáři nabízí lepší možnosti sdílení. Funkce „Stránky“ slouží k propagaci vlastních webových stránek a je důležitá zejména pro firmy. Funkce „Místa“ vyhledává doporučená místa v okolí, jedná se zejména o restaurace, kavárny a bary.

Z hlediska týmové spolupráce jsou velmi zajímavé takzvané „komunity“. Komunitu lze vytvořit vlastní a pozvat do ní ostatní lidi, přidávat v ní nové příspěvky či

komentáře nebo danou komunitu pouze sledovat. Komunit na Google Plus existuje velké množství, mezi nejznámější patří například Android, Linux či Star Wars.

Google+ API

Google+ API jsou založeny na čtyřech různých zdrojích a každý zdroj má různé metody. [11]

- **People** – lidé
 - **get** - získá podle id informace o specifickém člověku
 - **search** – vyhledá lidi podle jména
 - **list** – získá informace o lidech v kolekci

- **Activities** – příspěvky
 - **list** – získá všechny vlastní příspěvky
 - **get** – získá konkrétní příspěvek
 - **search** – vyhledá veřejné aktivity

- **Comments** – komentáře
 - **list** – získá všechny vlastní komentáře k danému příspěvku
 - **get** – získá konkrétní komentář

- **Moments** – aktivity vážící se k uživatelům v aplikacích
 - **insert** – vloží novou aktivitu
 - **list** – seznam všech aktivit
 - **remove** - odstraní konkrétní aktivitu

Google+ API je velmi omezené oproti webovému rozhraní Google+. Z hlediska týmové spolupráce nabízí velmi málo možností, například nelze získat lidi ze specifického kruhu, ani s kruhy nijak pracovat. Příspěvky lze získat pouze veřejné a není možné vytvořit nové. Ke komunitám také nelze vůbec přistupovat.

Následující ukázka získá informace z profilu uživatele na Google+. Parametr „me“ označuje naši osobu.

```
$plusService->people->get('me');
```

2.5.4 Google Drive

Google Drive je datové úložiště v cloudu. V Google Drive lze vytvářet nové složky, dokumenty, připojovat různé aplikace či nahrávat vlastní soubory. Soubory lze přímo z Google Drive otevírat, mazat, vytvářet jejich kopie či si je stáhnout na disk.

Velkou výhodou Google Drive je možnost sdílení souborů. Soubory lze sdílet odesláním odkazu pomocí elektronické zprávy či umístěním odkazu na Google Plus, facebooku i twitteru. Dokumentům či souborům můžeme přidávat různá oprávnění. Dokument můžeme nastavit veřejný, to znamená, že vyhledávat dokument a přistupovat k němu mohou všichni uživatelé internetu. Každému uživateli, se kterým chceme dokument sdílet lze nastavit oprávnění pro prohlížení, komentování a upravování daného dokumentu.

Google Drive API

Google Drive API patří do kolekce Google Drive SDK obsahující kromě již zmíněné API například Google Picker API umožňující ve vlastní aplikaci otevírat či nahrávat soubory uložené v Google Drive.

Hlavní funkcí Google Drive API je stahování a nahrávání souborů z vlastní aplikace do Google Drive. Mezi další funkce patří export souborů do různých formátů, sledování změn v daných souborech či čtení komentářů.

V Google Drive API můžeme pracovat s dvanácti různými zdroji. [12]

- **Files** – možnost získat, vytvořit, smazat soubor
- **About** – získání informací o uživateli
- **Changes** – získání seznamu změn v Google Drive
- **Children** – umožňuje získat či vložit soubory ve specifické složce
- **Parents** – získá všechny rodičovské složky souboru
- **Permissions** – přidání či odebrání práv k souboru
- **Revisions** – získání revize souborů
- **Apps** – získá seznam aplikací uživatele v rámci Google Drive
- **Comments** – získání, mazání či vytváření komentářů v daných souborech
- **Replies** – pro práci s odpověďmi na komentáře

- **Properties** – získání, přidání, mazání vlastních informací v souborech
- **Channels** – zakázání sledování změn

Pro sdílení souborů mezi uživateli je důležitý zdroj „Permissions“, kde můžeme uživatelům nastavit různé role, lze vybrat mezi „owner“ (vlastník), „reader“ (pouze čtení) a „writer“ (čtení i zápis).

Následující ukázka získá informace o uživatelích, kteří mezi sebou sdílí konkrétní soubor.

```
$driveService->permissions->listPermissions('fileID');
```

2.5.5 Google Dokumenty

Google Dokumenty tvoří skupinu složenou z Google Docs, Google SpreadSheets, Google Slides a dalších. Google Dokumenty jsou nabízeny jako součást Google Drive. Největší výhodou Google Dokumentů je možnost online spolupráce v reálném čase. Další výhodou dokumentů je jejich kompatibilita s Microsoft Office.

Google Docs jsou podobné programu Microsoft Word, Google SpreadSheets Microsoft Excel a Google Slides Microsoft Powerpoint. Google Dokumenty vynikají nad Microsoft Office právě v možnostech online spolupráce a snadného přístupu, jelikož se dokumenty sami ukládají do cloudu v Google Drive.

Mezi Google dokumenty dále patří Google Forms, Google Drawings a Google Tables. Google Forms je služba nabízející možnost vytvoření vlastních formulářů, tyto formuláře je možné použít ve vlastní aplikaci. V Google Drawings lze pomocí různých nástrojů kreslit obrázky, diagramy či myšlenkové mapy. Google Tables je webovou službou pro správu dat, mohou být použity pro sběr, vizualizaci a sdílení dat v tabulkách.

Google Documents API

Google Documents API v současnosti již není společností Google podporována a její částečná funkcionality byla přesunuta do Google Drive API, bohužel tím přišla o významné funkce pro práci se soubory.

Google Spreadsheet API patřící do skupiny Google Documents je stále podporována PHP frameworkem Zend, pro využití této API není nutné mít celý framework, stačí pouze knihovna GData. Tato knihovna velmi usnadňuje práci s tabulkami, můžeme například získat či vytvořit nové záložky, pracovat s řádky či přímo s jednotlivými buňkami.

Následující ukázka aktualizuje obsah buňky nacházející se na prvním řádku a ve druhém sloupci. „SpreadsheetID“ označuje id dokumentu a „workSheetID“ id pracovního listu.

```
$service->updateCell(1,2,'nový  
obsah','spreadSheetID','workSheetID');
```

2.5.6 Google Kontakty

Google Kontakty jsou soukromým adresářem, umožňující uložit a organizovat kontaktní informace o lidech, se kterými komunikujeme. Každý kontakt může obsahovat základní informace jako jméno, emailové adresy, telefonní čísla, ale také rozšířené informace jako adresa bydliště, fotografie či zaměstnání.

Nevýhodou Google kontaktů je omezené sdílení s ostatními uživateli. Sdílet kontakty lze pouze v rámci Google Apps, bohužel se jedná o placenou službu, bezplatný je však import a export kontaktů do formátu CSV.

Google Contacts API

Google Contacts API umožňuje klientským aplikacím zobrazovat, vytvářet, aktualizovat či mazat vlastní kontakty. Kontakty jsou uloženy přímo v účtu uživatele, díky tomu má většina Google služeb ke kontaktům přístup.

Při práci s Google Contacts API budou PHP programátoři v nevýhodě, jelikož klientské knihovny pro PHP tuto API nepodporují, ale je možné s nimi pracovat v XML formátu.

Požadavek pro získání kontaktů může vypadat následovně:

```
https://www.google.com/m8/feeds/contacts/{userEmail}/full
```

Odpověď dostaneme ve formátu XML a může být poněkud složitější získat specifickou informaci, jelikož tímto způsobem získáme mnoho redundantních informací.

2.5.7 Google Maps

Google Mapy umožňují zobrazení satelitních a terénních snímků, map měst s ulicemi či vyhledávání firem a tras. Mapy lze jednoduše sdílet na Google+ či vložit odkaz na přesné souřadnice ve vlastní aplikaci nebo mapy přímo vložit do vlastní aplikace pomocí Google Maps API.

Google Maps API

Google Maps API jsou souborem více různých API. Přehled těchto API se nachází v následujícím seznamu. [13]

- **Google Maps Android API** – mapy pro Android mobilní aplikace
- **Google Maps SDK for iOS** – mapy pro iOS mobilní aplikace
- **Google Maps Javascript API** – mapy pro desktopové aplikace v jazyce JavaScript
- **Google Maps Embed API** – mapa ve vlastní aplikaci vložená jako odkaz
- **Google Maps Engine layers** – možnost vytvářet interaktivní kresby na mapách
- **Google Static Maps API** – mapa pouze jako obrázek
- **Google Street View Image API** – zobrazení reálného světa z pohledu ulice
- **Google Geocoding API** – převod adresy na souřadnice
- **Google Places API** – automatické nabízení adres v textovém poli
- **Google Directions API** – plánování tras
- **Google Elevation API** – obohacuje mapy o nadmořskou výšku
- **Google Time Zone API** – pro práci s časovými pásmy

Google Maps API patří mezi neznámější Google API a nevyžadují OAuth autentizaci ani autorizaci. Pro přidání této API do vlastní aplikace stačí pouze API klíč, který stačí

přidat jako parametr skriptu. Mezi nepovinné parametry patří lokalizace, verze a sensor. Lokalizace určuje, v jakém jazyce chceme mapy zobrazit. Verze udává verzi map a sensor může nabývat hodnot „true“ nebo „false“, zapnutím sensoru lze zpřesnit polohu v reálném času například pomocí GPS.

```
<script src="http://maps.googleapis.com/maps/api/js?key=APIkey&sensor=true"></script>
```

Důležitou součástí Google map jsou ovládací prvky, které můžeme libovolně přidávat či odebrat, měnit jejich polohu, vzhled či si dokonce vytvořit ovládací prvky vlastní. Každou mapu si můžeme vzhledově přizpůsobit změnou barev, jasů, kontrastu, viditelnosti různých prvků či hustoty pixelů.

Zajímavou funkcí Google Map je možnost kreslení na mapě, kde lze vytvořit různé geometrické útvary nebo symboly, ale nejužitečnější funkcí jsou takzvané „markery“, kterými zvýrazníme daný bod na mapě, každý „marker“ může mít takzvané „info window“ doplňující informace o daném bodě.

2.5.8 Google Web Search

Vyhledávač www.google.cz (oficiální globální adresa je google.com) patří mezi nejpoužívanější a nejkvalitnější vyhledávače na světě, resp. lze říci, že Google je nejlepší internetový vyhledávač vůbec. Vyhledávač každý den zpracuje více než dvě miliardy uživatelských dotazů.

Google umí více než vyhledávání webových stránek, kromě webů umí také vyhledávat obrázky, zpravodajské servery a nabídky on-line prodeje zboží. Také nabízí své jednoduché webové rozhraní hned v několika světových jazycích. Veškeré výsledky vyhledávání jde definovat také podle místa, jazyku a času zveřejnění.

Google Web Search API

Pomocí Google Web Search API si můžeme vytvořit vlastní vyhledávač ve vlastní aplikaci. Můžeme si tak vyhledávač vzhledově přizpůsobit či mu přidávat další funkcionality. Prohledávat můžeme web, videa, blogy, novinky, obrázky, knihy,

patenty i vlastní web. Dále můžeme nastavit takzvaný „centerPoint“ udávající kde se přibližně nacházíme a podle toho se upraví vyhledávané příspěvky. Důležitou funkcí je také nastavení jazykové verze vyhledávače a mít tak vyhledávač třeba v českém jazyce.

Google Web Search API je dostupná v jazyce JavaScript a ve své aplikaci lze API načíst pomocí Google Loader API. Google Web Search API je tvořena čtyřmi základními třídami. [14]

- **SearchControl** – tato třída poskytuje uživatelské rozhraní přes několik objektů, kde každý objekt je určen k vyhledávání a k vrácení výsledků specifické třídy
- **Search** – patří mezi základní třídu, ze které všechny vyhledávače dědí
- **GResult** – tato třída zapouzdřuje výsledky vyhledávání objektů
- **SearcherOptions** – slouží ke konfiguraci chování vyhledávače

Následující ukázka importuje Google Web Search API do vlastní aplikace.

```
google.load("search", "1", {"language" : "en"});
```

2.5.9 Google Cloud Datastore

Google Cloud Datastore je nerelační databáze umožňující uložení vlastních dat. Datastore je automaticky škálovatelný, vysoce spolehlivý, podporuje ACID transakce a má vysokou dostupnost čtení a zápisu. Webová služba Datastore nabízí jednoduchou a snadno přístupnou administraci pomocí HTTP s využitím JSON. Google Cloud Datastore je primárně určen pro Google App Engine, což je webový aplikační framework.

Google Cloud Datastore je založen na BigTable, což je vysoce škálovatelné, distribuované, vysoce dostupné a strukturované úložiště, avšak BigTable není sama o sobě databází (nepodporuje dotazování). BigTable využívá Google pro vyhledávač, Youtube, Earth, Analytics atd. BigTable podporuje CRUD (create, read, update, delete). [15]

Dotazování databáze není postaveno na jazyku SQL, ale lze využít jazyka GQL „Google Query Language“ což je obdoba jazyka SQL od společnosti Google. Kromě jazyka GQL lze dotazovat přímo entity, což jsou datové objekty, které vyhovují námi specifikovaným vlastnostem, které můžeme měnit pomocí filtrů.

U Google Datastore se setkáme se čtyřmi pojmy Kind, Entity, Key, Property. V porovnání s relační databází Kind představuje tabulku, Entity řádek, Key primární klíč a Property jsou atributy.

Následující ukázka (využívá jazyku GQL) najde klíče entit z tabulky „Person“, jež mají věk nastaven na „null“.

```
SELECT __key__ FROM Person WHERE age = NULL 5
```

Google Cloud Datastore API

V Google Cloud Datastore API máme jeden základní zdroj „Datasets“, pro práci s ním můžeme volat šest základních metod. [16]

- **allocateIds** – přiřadí ID nekompletním klíčům (užitečné pro odkázání na entitu předtím, než je vložena)
- **beginTransaction** – zahájí transakci
- **commit** – provede transakci, volitelně vytvoří, smaže či změní entity
- **lookup** – vyhledá entity podle klíče
- **rollback** – vrátí zpět transakci
- **runQuery** – dotazy na entity

Následující ukázka vyhledá konkrétní entitu (řádek). Podrobnější implementaci lze nalézt v uživatelské příručce přiložené k této práci.

```
$this->datastore->datasets->lookup('datasetId', $entity);
```

⁵ Získáno z https://cloud.google.com/datastore/docs/apis/gql/gql_reference

2.5.10 Google Analytics

Tato služba umožňuje sledovat návštěvnost stránek. Obsahuje obrovské množství různých statistik a grafů. V Google Analytics můžeme sledovat počet návštěv za dané období, počet unikátních uživatelů, kolik stránek bylo zobrazeno, jaká je průměrná doba trvání návštěvy či poměr nových a navracených se návštěvníků.

Význam Google Analytics je zejména z hlediska SEO optimalizace, kdy můžeme zjistit například demografické údaje návštěvníků a nabídnout vlastní web ve více jazykových verzích. Dále můžeme zjistit, z jakého prohlížeče uživatelé nejčastěji přicházejí, a tak se zaměřit na vyladění stránek pro nejpoužívanější prohlížeče. Například na vlastních stránkách mi nejvíce chodí lidé z Google Chrome, poté z Firefoxu a Internet Explorer je až na třetím místě. Dále můžeme zjistit jaký operační systém či rozlišení obrazovky lidé nejčastěji používají.

Z hlediska týmové spolupráce lze v Google Analytics využít funkci „Správa uživatelů“, kde můžeme uživatelům dát přístup k jakékoli sadě služeb či jim dát zvláštní oprávnění pro každou z těchto služeb. Lze například vytvořit následující: [17]

- **oprávnění ke správě uživatelů** - uživatelé v roli správce, jejichž jedinou funkcí v rámci účtu Analytics bude přidávat uživatele, přidat jim oprávnění pro služby a odstranit je z účtu, pokud je to nutné
- **oprávnění ke spolupráci** - uživatelé spolupracující na přehledech, kteří mohou vytvářet a sdílet osobní položky a spolupracovat na sdílených položkách, například upravit sdílený hlavní panel nebo poznámku
- **oprávnění k úpravám** - administrátoři účtů, kteří mohou vytvářet a upravovat objekty v účtu Analytics, jako jsou cíle a filtry, a kteří mohou také zobrazovat údaje v přehledu a manipulovat s nimi
- **oprávnění k zobrazení** - uživatelé pracující na přehledech (např. analytici), kteří mohou zobrazovat údaje v přehledu a manipulovat s nimi

Google Analytics API

V Google Analytics mají všichni uživatelé k dispozici tři rozhraní API: [18]

- **Rozhraní Collection API** – umožňuje personalizaci měřicího kódu, shromažďování údajů o nákupech na webových stránkách, získání přehledů pro dvě subdomény v jediném výběru dat, nastavení vlastní definice typů návštěvníků atd.
- **Rozhraní API pro správu** – pomocí rozhraní API pro správu Google Analytics lze získat přístup k účtu Google Analytics a k údajům výběru dat ve formě zdrojů Google Data API. Pomocí rozhraní API pro správu lze efektivně načíst konkrétní soubor výběrů dat uživatele. Můžete také načíst údaje o konfiguraci cíle související s konkrétním výběrem dat.
- **Rozhraní Data Export API** – slouží k vytváření aplikací pomocí údajů Google Analytics, které jsou již k dispozici v účtu Analytics. Aplikace mohou využívat rozhraní API k vyslání požadavku na údaje na úrovni přehledů ze stávajícího výběru dat Analytics a k získání personalizovaného přehledu údajů vybraného výběru dat.

Pomocí Rozhraní API pro správu můžeme spravovat uživatele a přidávat jim přístup k měření vlastních webů, tímto způsobem lze spolupracovat s ostatními uživateli. Tato funkce je výhodná především pro SEO optimalizaci větších webů.

2.5.11 Google Hangouts

Google Hangout je webová služba pomocí které lze uspořádat skupinové konverzace. V Google Hangout můžeme vybrat mezi různými styly konverzace. První je pouze textová ve formě chatu, druhá je hlasová čili telefonická a třetí je ve formě videokonference.

Po připojení do služby Google Hangout můžeme poslat pozvánku ostatním lidem, kterým poté přijde jako elektronická zpráva nebo upozornění pokud jsou právě přihlášení pod Google účtem.

Google Hangout obsahuje několik zajímavých funkcí například sdílení vlastní obrazovky či jakéhokoliv otevřeného okna a tím velmi usnadňuje týmovou spolupráci. Dále lze Google Hangout propojit s dalšími aplikacemi například s Google Drive a v reálném čase s ostatními pracovat na daném dokumentu a současně spolu komunikovat. Další užitečnou funkcí je kreslení, kdy během videokonference můžeme nakreslit cokoli na vlastní obrazovce, a ihned tak sdílet vlastní nápady.

Google Hangouts API

Google Hangout API poskytuje programové rozhraní pro Hangouts video hovory. Poskytuje rozhraní v JavaScriptu, které umožňuje vlastní aplikaci zobrazit seznam účastníků, sdílet data mezi účastníky, kontrolovat určité aspekty uživatelského rozhraní či měnit nastavení mikrofону, kamery a reproduktorů. [19]

2.5.12 Google Charts

Google Charts umožňují vizualizovat data ve své aplikaci. Nabízejí velké množství různých druhů grafů, za zmínku stojí třeba výsečový, spojnicový nebo bodový graf. Kromě těchto tradičních grafů nabízejí Google Charts i grafy poměrně netradiční, například mapu s vyznačenými státy, graf struktury organizace, graf se stromovou strukturou či různé tabulky. Velkou předností Google Charts je jejich interaktivita, téměř každý graf obsahuje různé animace, které se spouští automaticky podle události myši.

Google Charts API

Google Charts API jsou pouze v jazyce JavaScript a pro jejich využití v aplikaci není nutná autentizace. Google Charts API importujeme do aplikace pomocí „Google Loader“, což vysvětluje následující ukázka.

```
google.load('visualization', '1.0', {'packages':['corechart']});
```

Každý typ grafu obsahuje vlastní třídu, například výsečový graf vytvoříme podle třídy „PieChart“, spojnicový graf zase podle třídy „LineChart“. Tyto třídy včetně ukázek implementace lze najít v dokumentaci Google Charts API. [20]

Následující ukázka vykreslí graf v naší aplikaci.

```
var chart_div = document.getElementById('piechart');
var chart = new google.visualization.PieChart(chart_div);
chart.draw(data, options);
```

Google Charts API využívají pro importování dat primárně třídu „DataTables“, ale je možné data importovat z různých zdrojů, například z databáze, Google SpreadSheets či Google Analytics. [19]

Následující ukázka importuje data pomocí třídy „arrayToDataTable“.

```
var data = google.visualization.arrayToDataTable([
    ['úkol', 'počet hodin za den'],
    ['pracovat',      8],
    ['spát',          7]
]);
```

3 Praktická část

3.1 Analýza a návrh aplikace

Aplikace bude vytvořena za účelem demonstrace Google API a jejich využití pro týmovou spolupráci. Bude se skládat z různých modulů využívajících konkrétní API. Téměř všechny moduly se budou nacházet na jediné webové stránce.

První modul využívající hlavně Google Tasks API se bude jmenovat „úkoly“. Úkoly bude možné přidávat, mazat a označovat jako dokončené. Sdílení úkolů bude provedeno pomocí Google Calendar API.

Druhý modul využívající hlavně Google Calendar API se bude nazývat „události“. Události bude možné přidávat, mazat a zobrazovat uživateli na mapě jejich místo konání. Události budou viditelné pro ostatní spolupracovníky v Google kalendáři.

Třetí modul bude využívat Google Maps API, tento modul bude sám o sobě interaktivní mapou, na které uživatel bude moci sledovat polohu spolupracovníků v reálném čase či zobrazovat již zmíněné události.

Čtvrtý modul využívající Google Drive API bude sloužit jako diskuze, ve které budou moci spolupracovníci přidávat příspěvky spolu s komentáři. Tento modul bude provázán s Google SpreadSheets.

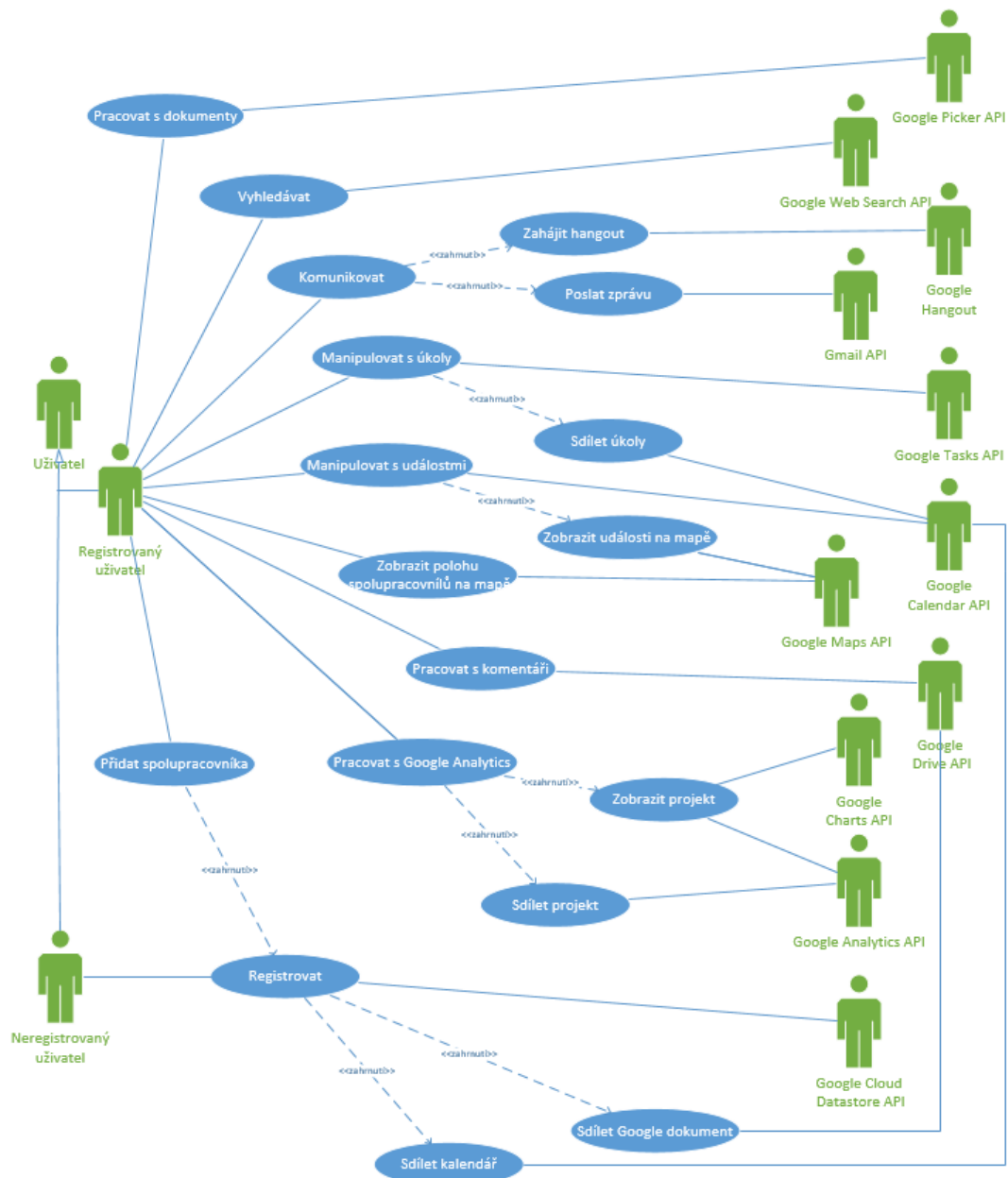
Pátý modul využívající Google Picker API se bude jmenovat „dokumenty“. Dokumenty bude možné vytvářet nové, nahrávat na Google Drive nebo je z aplikace otevírat.

Mezi další funkce aplikace bude patřit možnost zahájit Hangout (skupinový chat, videokonference), zobrazit jména spolupracovníků dle Google Contacts API, posílat novým spolupracovníkům pozvánky do aplikace a implementace vyhledávače.

Aplikace též využije Google Analytics API pro získání údajů o návštěvnosti aplikace a vizualizací dat v grafické podobě pomocí Google Charts API.

3.1.1 Diagram případů užití

Na tomto diagramu je znázorněn hlavní aktér, tj. uživatel, jež může být registrovaný či neregistrovaný. V každém případě nejprve dojde k autentizaci uživatele pomocí OAuth 2.0 protokolu. Pro vstup do aplikace se musí uživatel nejprve registrovat a až poté lze využívat všechny funkce aplikace. Uživatel může být registrován též jiným uživatelem.

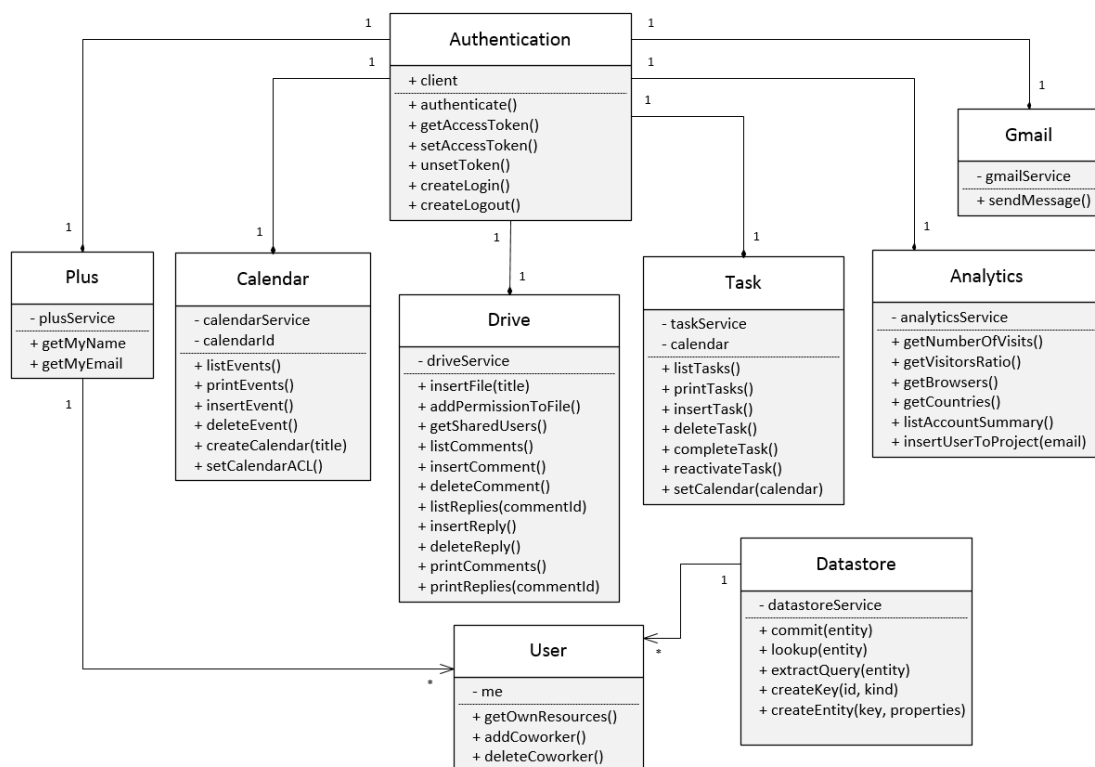


Obrázek 6: Diagram případů užití

3.1.2 Diagram tříd

Diagram tříd znázorňuje vztahy mezi třídami. Z důvodu složitosti nejsou v návrhu zobrazeny vztahy s Google klientskými knihovnami pro PHP. Popis tříd se nachází v následujícím soupisu.

- **Authentication** – třída pro autentizaci uživatele pomocí OAuth 2.0 protokolu
- **Gmail** – pro práci s Gmail API
- **Calendar** – pro práci s Google Calendar API
- **Drive** – třída pro práci s Google Drive API
- **Task** – pro práci s Google Tasks API
- **Plus** – třída pro práci s Google+ API
- **User** – pro správu spolupracovníků a získání vlastních údajů z databáze
- **Analytics** – třída pro práci s Google Analytics API
- **Datastore** – pro práci s Google Datastore API



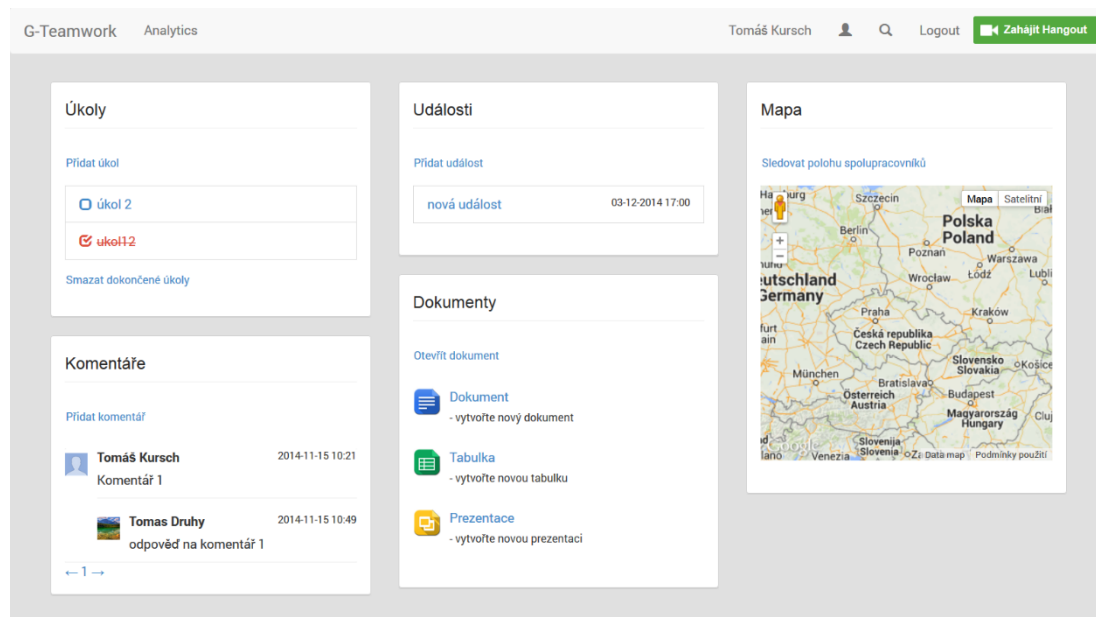
Obrázek 7: Diagram tříd

3.2 Implementace a testování aplikace

Aplikace je primárně postavena na jazyce PHP a využívá objektově orientovaný přístup. Pro práci s Google API využívá klientské knihovny pro PHP. Aplikace využívá jazyku JavaScript a rovněž, na tomto jazyce postavených, Google API.

Aplikace nefunguje na klasickém webhostingu, ale využívá platformy Google App Engine, jež byla po dlouhou dobu dostupná pouze pro jazyky Java a Python, ale nyní nově lze využívat i pro jazyk PHP. Výhodou této platformy je podpora databáze Google Cloud Datastore, jejíž API aplikace využívá.

Struktura aplikace je postavena na frameworku „Bootstrap“ využívajícím HTML5 a CSS. Následující obrázek ukazuje rozmístění prvků na hlavní stránce aplikace.



Obrázek 8: Ukázka aplikace

Modul pro práci s úkoly využívá primárně Google Tasks API, pro sdílení úkolů je využito Google Calendar API, které je stěžejní pro práci s událostmi. Mapa je vytvořena díky Google Maps API. Komentáře využívají Google Drive API a v modulu pro práci s dokumenty lze nahrávat či vybírat soubory z Google Drive pomocí Google Picker API.

V horní liště pravé části aplikace se nacházejí dvě ikony a tlačítko „Logout“, jímž se lze odhlásit. Levou ikonou se zobrazí seznam spolupracovníků vycházející z Google Drive API. Pravou ikonou se zobrazí okno pro vyhledávání, které využívá Google Web Search API. V levé části horní lišty se nachází odkaz na další stránku s názvem „Analytics“ sloužící k vizualizaci dat z Google Analytics API pomocí Google Charts API.

Komunikaci mezi spolupracovníky obstarává Gmail API a Google Hangouts. Informace o naší osobě (jméno, email) předává Google+ API.

3.2.1 Programovací jazyky a standardy

PHP

Při vývoji aplikace využívající vybrané Google API jsem nejvíce využil jazyka PHP. Klientské knihovny pro Google API jsou objektově orientované, proto jsem i při samotném vývoji aplikace použil objektově orientovaný přístup. Google API, k nimž jsem přistupoval pomocí jazyka PHP, byla většina, konkrétně Tasks, Calendar, Plus, Drive, Contacts, Cloud Datastore, Analytics a Gmail.

JavaScript

Druhým nejpoužívanějším jazykem při vývoji aplikace byl JavaScript. Tento jazyk jsem nejvíce aplikoval při práci s Google Maps API. Další API využívající JavaScript je Google Web Search, Google Picker a Google Charts.

HTML a CSS

Poslední jazyk, který jsem využíval při vývoji aplikace, byl jazyk HTML s kaskádovými styly. Tento jazyk jsem aplikoval k zobrazení výstupu z dané Google API a pro vytvoření struktury aplikace. Kaskádovými styly jsem přizpůsobil vzhled aplikace. HTML byl současně využit pro přístup k Google API, například bylo možno přidávat různá Google+ tlačítka či měnit jejich vzhled. Neznámější tlačítka jsou „+1 button“, „Hangout button“ a „Google+ Sign-in button“.

3.2.2 Knihovny

Klientské knihovny

Nejdůležitější knihovnou, již jsem používal při vývoji aplikace, je klientská knihovna pro PHP. Klientské knihovny pro PHP jsou stále ve vývoji, v současné době se nacházejí v beta verzi. Při vývoji aplikace jsem se setkal s určitými komplikacemi, tudíž jsem musel zkoušet různé verze klientských knihoven, více se o daném problému rozepíši v závěru práce.

Pro využívání klientských knihoven pro PHP je nutné mít PHP 5.2.1 nebo vyšší verze, PHP JSON extension a základní znalosti objektově orientovaného programování.

GData

Gdata je starší verzí klientských knihoven, dnes již společností Google nepodporovanou. GData pro PHP jsou nabízena jako součást frameworku Zend. S těmito knihovnamí jsem pracoval, jelikož poskytují přístup k dnes již nepodporovaným API, například Google SpreadSheets.

Pomocí těchto knihoven jsem naprogramoval možnost pracovat přímo s Google SpreadSheets ve vlastní aplikaci, to znamená číst obsah dokumentu, přidávat a mazat řádky či aktualizovat obsah buňky. Bohužel po převodu aplikace na platformu Google App Engine se mi nepodařilo tyto knihovny znovu zprovoznit, proto je v aplikaci nakonec nevyužívám.

JQuery

JQuery je knihovna postavená na jazyku JavaScript. Velkou výhodou je podpora ajaxu. Díky němu jsem mohl propojit funkce různých Google API postavených na různých jazycích, PHP a JavaScript.

3.2.3 Frameworky

Bootstrap

Bootstrap je v současnosti nejpopulárnějším frameworkem založeným na HTML, CSS a JS. Bootstrap obsahuje řadu předdefinovaných komponent a tím výrazně urychluje práci při vývoji webových aplikací. Tento framework jsem využil k vytvoření základního designu aplikace, ale i k vytvoření různých komponent jako jsou ikony, formuláře, dialogová okna, tlačítka a tabulky. Výhodou tohoto frameworku je také velká podpora různých JQuery pluginů.

3.2.4 Vývojové prostředí a platformy

Netbeans

Netbeans IDE je zdarma distribuované vývojové prostředí, vlastněné firmou Oracle, primárně je určeno pro jazyk Java, ale v současné době podporuje i jazyk PHP. Netbeans IDE je naprogramován v jazyce Java, lze jej tedy spustit na více operačních systémech, konkrétně Windows, Linux, Mac OS a Solaris.

V Netbeans IDE jsem naprogramoval téměř celou aplikaci, později, když jsem aplikaci přepisoval, aby fungovala na platformě Google App Engine, jsem přešel na nové IDE PhpStorm.

PhpStorm

PhpStorm je IDE určené primárně pro jazyk PHP, ale i další jazyky jako JS, HTML a CSS. PhpStorm vyvíjí firma JetBrains. Mezi zajímavé funkce PhpStorm patří zvýrazňování kódu, tudíž jsou jednotlivé jazyky od sebe odděleny, například SQL je jinak zvýrazněné než PHP, to samé platí i pro HTML od CSS. PhpStorm obsahuje též vlastní Debugger a za zmínku stojí i funkce, pomocí které lze z kódu vygenerovat UML diagramy.

Z Netbeans IDE na PhpStorm jsem přešel hlavně z důvodu podpory platformy Google App Engine, kdy v PhpStorm lze jedním tlačítkem pohodlně odeslat na server jakoukoli změnu v kódu.

Google App Engine

Google App Engine patří do kategorie Paas (Platform as a service). Paas představuje službu, aplikační platformu, ve které se poskytují kompletní prostředky pro vývoj a údržbu vlastních aplikací dostupných prostřednictvím internetu. Vývojář má k dispozici kompletní vývojové, ladící a aplikační prostředí, ve kterém je možné vytvářet a následně provozovat vlastní aplikace. Google App Engine nabízí služby pro aplikace napsané v jazyce Java, Python a nově i v jazyce PHP a to pomocí interpretu napsaného v Javě.

Pro zprovoznění Google App Engine pro PHP je nutné nejprve stáhnout a nainstalovat Google App Engine SDK, které je k dispozici pro operační systémy Windows, Mac a Linux. Dále je nutné nainstalovat Python, zde bych rád zdůraznil, že sice na oficiálních stránkách Google App Engine se doporučuje verze 2.7, ale s touto verzí jsem měl spoustu bugů na lokálním serveru, po nainstalování jazyku Python v nižší verzi, konkrétně 2.5, všechny bugy zmizely a aplikace se stala použitelnou i na lokálním serveru.

Výhodou Google App Engine je přímá podpora Google Cloud Datastore API, kde pomocí webového rozhraní lze provádět dotazy na databázi, procházet logy či sledovat statistiky využití platformy. Google App Engine je sice nabízen zadarmo, ale pokud je aplikace uživateli více využívána, je nutné platit poplatky, cena se liší dle náročnosti využití dané aplikace.

3.2.5 Testování

Aplikaci jsem nejdříve testoval na lokálním serveru, k tomu mi posloužil instalační balíček XAMPP, což je neplacená distribuce Apache, obsahující MySQL, PHP a Pearl. Po převodu aplikace na platformu Google App Engine jsem aplikaci testoval přímo

pro tuto platformu. Testovat aplikaci běžící na Google App Engine lze i lokálně přímo na svém počítači po nainstalování GAE SDK a Python 2.5.

K testování aplikace jsem nejvíce využíval prohlížeč Firefox, jeho výhodou je velká podpora doplňků. Nejdůležitějším doplňkem pro mě byl Firebug, hlavně pro testování Javascriptu, HTML a CSS. Firebug mi také pomohl optimalizovat dobu načítání jednotlivých API. Jelikož požadavky na ně jsou poměrně časově náročné, velmi často jsem musel hledat úspornější řešení.

3.3 Tvorba uživatelské příručky

Uživatelskou příručku jsem vytvořil jako tutoriál usnadňující vývojářům implementaci vybraných Google API ve vlastní aplikaci. Příručka vychází z vlastní aplikace, kterou jsem vytvořil v rámci této práce.

Obsah příručky jsem rozdělil do třech hlavních částí. V první části se věnuji autentizaci, ve které popisuji, jakým způsobem požadované API aktivovat a jakým způsobem vygenerovat autentizační údaje. V druhé části stručně popisuji jednotlivé funkce aplikace, které Google API aplikace využívá. Ve třetí části příručky se zabývám implementací použitých Google API, navíc do této části příručky jsem přidal různé úkoly včetně jejich řešení.

4 Závěr

Cílem této práce bylo popsat vybraná Google API a jejich využitelnost pro týmovou spolupráci, navrhnout a implementovat aplikaci využívající Google API a vytvořit uživatelskou příručku.

Z hlediska využitelnosti jednotlivých API pro týmovou spolupráci jsem došel k závěru, že nejpoužitelnější API pro spolupráci v týmu je Google Calendar API a Google Drive API. Tyto API nabízejí jednoduchou možnost sdílení zdrojů a práce s nimi probíhala velmi rychle.

Největším zklamáním pro mě byla Google+ API. Původně jsem předpokládal, že pomocí této API vytvořím kruh přátel, který bude představovat tým. Zjistil jsem, že možnost pracovat s kruhy lze pouze v „Google Apps pro firmy“, což je balíček služeb od společnosti Google využívaný hlavně pro firemní účely. Bohužel pro využívání tohoto balíčku je nutné platit poměrně vysoké měsíčné poplatky. Google+ API neumožňuje ani získání lidí z určitého kruhu, a dokonce ani zde nelze vytvářet příspěvky a komentáře.

Odpovědět na otázku, zda jsou Google API vhodná pro týmovou spolupráci není jednoduché, jelikož každá API má jinou funkcionalitu a v možnostech týmové spolupráce se od sebe velmi liší. Celkově Google API mají velký potenciál pro týmovou spolupráci, který ale zůstává nevyužit, jelikož API nejsou spolu příliš dobře provázané.

Při shromažďování informací o Google API jsem na mnoho problémů nenarazil, jelikož oficiální dokumentace je poměrně obsáhlá, ale pro vývojáře, jež nemá s API dostatek zkušeností, může být obtížné všem konceptům porozumět, proto jsem vytvořil uživatelskou příručku, ve které se nachází přesný postup, jak s Google API pracovat. Uživatelská příručka je v textovém formátu a je přiložena v příloze této bakalářské práce.

Při tvorbě uživatelské příručky jsem využil hlavně svých praktických zkušeností, jež jsem získal během shromažďování informací o Google API a implementace aplikace.

Největším přínosem příručky je ucelení informací o způsobu implementace Google API ve své aplikaci tak velmi usnadňuje práci případným vývojářům.

Při návrhu aplikace jsem došel k závěru, že použiji hlavně programovací jazyk PHP, se kterým jsem měl nejvíce zkušeností a díky jazyku UML pro návrh software jsem vytvořil diagram případů užití a třídní diagram.

Nejvíce času mi zabrala implementace modulu úkolů využívající Google Tasks API, jelikož tato API nenabízí žádnou možnost sdílení, pokusil jsem se tento problém vyřešit pomocí nástroje Google Apps Script, což je skriptovací jazyk v cloudu, založený na JavaScriptu, který přímo podporuje většinu známých Google API. Bohužel aplikace vytvořené v tomto jazyce nelze vložit do své aplikace, pouze do Google Sites. Úkoly jsem zkoušel sdílet také pomocí Google Spreadsheet API, bohužel rovněž tato API není již podporována a není k ní možné přistoupit z aktuálních klientských knihoven, naštěstí jí podporuje stará verze knihoven GData. Pomocí této knihovny se mi podařilo modul úkolů zprovoznit, ale nakonec při přechodu na Google App Engine daný modul přestal fungovat. Nakonec jsem sdílení úkolů vyřešil pomocí Google kalendáře.

Jedním z problémů, se kterým jsem se potýkal při implementaci aplikace, byla poměrně dlouhá doba požadavků na dané Google API a následných odpovědí. Během testování některé požadavky trvaly několik sekund, ale při více požadavcích se doba načtení stránky prodloužila až na několik desítek sekund. Tento problém jsem vyřešil pomocí Memcache, což je distribuovaný systém cachování paměti.

S klientskými knihovnami se mi většinou pracovalo dobře, ale narazil jsem na několik bugů a musel zkoušet různé verze, tudíž jsem strávil hodně času přepisováním kódu pro danou verzi knihoven.

Společnost Google plánuje do budoucna rozšířit funkcionalitu svých služeb a jejich API, takže se snad dočkáme i větší podpory sdílení, například Google Tasks má velký potenciál, který zatím zůstává nevyužit.

Kdybych měl zhodnotit osobní přínos této práce, tak nejvýznamnějším pro mě bylo výrazné zlepšení mých dovedností při práci s jazykem PHP a JavaScript. Dále jsem se seznámil s nejrůznějšími, a podle mého názoru, velmi zajímavými nástroji a

technologiami, které společnost Google nabízí. Za zmínku stojí určitě nástroje Google Apps Script a Google App Engine, které by mohly představovat zajímavá témata pro další bakalářské práce.

5 Literatura

- [1] *Společnost – Google* [online]. 2014 [cit. 2014-11-07]. Dostupné z: <http://www.google.cz/intl/cs/about/company/>
- [2] Google Developers. Intro to REST. In: *Youtube* [online]. Zveřejněno 7. 10. 2008 [cit. 2014-11-07]. Dostupné z <https://www.youtube.com/watch?v=YCcAE2SCQ6k>
- [3] *Google APIs Discovery Service* [online]. 2014 [cit. 2014-11-07]. Dostupné z: <https://developers.google.com/discovery/?hl=cs>
- [4] *Google Loader* [online]. 2014 [cit. 2014-11-07]. Dostupné z: <https://developers.google.com/loader/>
- [5] *Using OAuth 2.0 to Access Google APIs* [online]. 2014 [cit. 2014-11-07]. Dostupné z: <https://developers.google.com/accounts/docs/OAuth2>
- [6] *Google+ Sign-In* [online]. 2014 [cit. 2014-11-07]. Dostupné z: <https://developers.google.com/+web/signin/>
- [7] *Google support* [online]. 2014 [cit. 2014-12-03]. Dostupné z: <https://support.google.com/>
- [8] *Google Apps Tasks API Concepts* [online]. 2014 [cit. 2014-11-07]. Dostupné z: <https://developers.google.com/google-apps/tasks/concepts>
- [9] *Google Apps Calendar API Concepts and Use Cases* [online]. 2014 [cit. 2014-11-07]. Dostupné z: <https://developers.google.com/google-apps/calendar/concepts>
- [10] *Google Plus v kostce: manuál nového uživatele* [online]. 2011 [cit. 2014-11-11]. Dostupné z: <http://www.zive.cz/clanky/google-plus-v-kostce-manual-noveho-uzivatele/sc-3-a-157777/default.aspx>

- [11] *Google+ API* [online]. 2014 [cit. 2014-11-07]. Dostupné z: <https://developers.google.com/+/api/>
- [12] *Google Drive Web APIs* [online]. 2014 [cit. 2014-11-07]. Dostupné z: <https://developers.google.com/drive/v2/reference/>
- [13] *Google Maps API* [online]. 2014 [cit. 2014-11-08]. Dostupné z: <https://developers.google.com/maps/documentation/api-picker>
- [14] *Google Web Search API* [online]. 2014 [cit. 2014-11-08]. Dostupné z: <https://developers.google.com/web-search/docs/reference>
- [15] *Google's BigTable* [online]. 2006 [cit. 2014-11-08]. Dostupné z: <http://andrewhitchcock.org/?post=214>
- [16] *Google Cloud Datastore* [online]. 2014 [cit. 2014-11-08]. Dostupné z: <https://cloud.google.com/datastore/docs/apis/v1beta2/>
- [17] *Google Analytics - uživatelská oprávnění* [online]. 2014 [cit. 2014-11-08]. Dostupné z: <https://support.google.com/analytics/answer/2884495?hl=cs>
- [18] *Google Analytics* [online]. 2014 [cit. 2014-11-08]. Dostupné z: <https://developers.google.com/analytics/?hl=cs>
- [19] *Google+ Hangouts API* [online]. 2014 [cit. 2014-11-08]. Dostupné z: <https://developers.google.com/+/hangouts/api/>
- [20] *Google Charts* [online]. 2014 [cit. 2014-11-08]. Dostupné z: <https://developers.google.com/chart/interactive/docs/>

6 Přílohy

Google API - uživatelská příručka

Tato příručka byla vytvořena v rámci bakalářské práce k webové aplikaci nacházející se na internetové adrese <http://g-teamwork.appspot.com>. Zároveň ji lze použít i jako tutoriál pro vývojáře webových aplikací, kteří chtějí použít Google API ve své aplikaci. V rámci obsáhlosti aplikace příručka obsahuje pouze vybrané části kódu, úplná implementace aplikace se nachází v příloze. Tato příručka vychází výhradně z programovacích jazyků PHP a JavaScript.

Obsah příručky je rozdělen do tří částí. První část se věnuje autentizaci, kde se čtenář seznámí, jak požadované API aktivovat a jakým způsobem vygenerovat autentizační údaje. Druhá část stručně popisuje jednotlivé funkce aplikace a v závěru obsahuje krátký soupis použitých Google API. Třetí a poslední část příručky se zabývá implementací použitých Google API v rámci této aplikace, ve které navíc čtenáře čekají i různé úkoly, jejichž řešení se nachází na konci příručky.

1 Autentizace

Aplikace využívá OAuth 2.0 protokolu k autentizaci Google API. Tato kapitola popisuje přesný postup, jakým způsobem lze vybrané API aktivovat, jak vygenerovat autentizační údaje, v závěru pak samotnou implementaci pro získání OAuth 2.0 tokenu.

1.1 Aktivace Google API

Chceme-li na svých stránkách použít některou z dostupných Google API, musíme nejdříve na stránkách <https://console.developers.google.com> vytvořit nový projekt (stačí pouze zadat název projektu), nebo použít již existující projekt.

Po vytvoření projektu budeme přesměrováni na hlavní stranu, kde v levém menu pod záložkou „APIs & auth“ vybereme „APIs“, následně se nám zobrazí seznam všech dostupných API. U každé API nalezneme kvótu (počet možných požadavků v rámci

jednoho dne) a tlačítko, kterým lze požadovanou API aktivovat. Po kliknutí na tlačítko dojde k aktivaci požadované API (u některých API je nutné ještě odsouhlasit licenční podmínky) a následně se zobrazí v seznamu aktivovaných API, kterou můžeme kdykoli deaktivovat či si prohlédnout z kolika procent je v naší aplikaci využívána.

1.2 Autentizační údaje

K Google API mají přístup pouze autentizovaní uživatelé, proto si musíme, před samotným použitím API ve vlastní aplikaci, vygenerovat autentizační údaje. Tyto údaje získáme na stránkách <https://console.developers.google.com>, kde v levém menu pod záložkou „APIs & auth“ vybereme „Credentials“. Jestliže chceme pracovat s API mající přístup k privátním datům, využijeme protokolu OAuth 2.0 pro autorizovaný přístup. Pro přístup k veřejným datům není nutná autorizace a postačí pouze API klíč.

API klíč vytvoříme kliknutím na tlačítko „Create new Key“, kde se nám posléze zobrazí čtyři možnosti:

- **Server key**
- **Browser key**
- **Android key**
- **iOS key**

Přejeme-li si použít Google API v rámci jedné IP adresy, vybereme „Browser key“, pro více IP adres vybereme „Server Key“.

Rozhodneme-li se využít OAuth 2.0 protokolu, klikneme na tlačítko „Create new Client ID“, následně se nám zobrazí okno, ve kterém si vybereme typ aplikace. Máme na výběr ze tří možností:

- **Web application** – aplikace je přístupná pomocí webového prohlížeče
- **Service account** – Google API jsou volány v zastoupení vlastní aplikace namísto koncového uživatele (aplikace nevyžaduje přístup k datům uživatele), jsou použitelné pouze pro Google Cloud Datastore API a Google Prediction API
- **Installed application** – funguje nainstalována na počítači či přenosném zařízení (Android, iOS)

Jelikož chceme vytvořit webovou aplikaci, vybereme možnost „Web application“. Nyní ještě musíme vyplnit pole „Authorized JavaScript Origins“, kam zadáme adresu, na které naše aplikace poběží (například <https://www.example.com>) a nakonec vyplníme pole „Authorized redirect uri“, které udává, kam chceme po autentizaci uživatele přesměrovat (například <https://www.example.com/oauth2callback>). Po kliknutí na tlačítko „generate“ získáme autentizační údaje, které později vložíme do vlastní aplikace.

Client ID for web application	
CLIENT ID	██████████.apps.googleusercontent.com
EMAIL ADDRESS	██████████@developer.gserviceaccount.com
CLIENT SECRET	██████████
REDIRECT URIS	http://localhost http://127.0.0.1:8080/ http://localhost:8080/ More ...
JAVASCRIPT ORIGINS	http://localhost:8080/ http://127.0.0.1:8080/ http://g-teamwork.appspot.com/

[Edit settings](#) [Reset secret](#) [Download JSON](#) [Delete](#)

Obrázek 5: Autentizační údaje

1.3 Consent screen

Po vygenerování autentizačních údajů pod záložkou „APIs & auth“ vybereme „Consent screen“, což je stránka, která se ve vlastní aplikaci objeví uživateli, když se pokusí získat přístup k dané API, aby věděl, k jakým svým datům poskytuje aplikaci přístup.

Nejdříve vyplníme emailovou adresu „email address“ a název aplikace „product name“, tato pole jsou povinná. Ještě můžeme doplnit některé volitelné údaje, například logo, adresu domovské stránky, adresu podmínek užití a odkaz na Google+ stránku.

Nyní máme Google API připraveny k použití ve vlastní aplikaci.

1.4 Implementace OAuth 2.0

Nyní máme vše připravené a můžeme začít se samotnou implementací OAuth 2.0 protokolu pro autorizaci. V případě Google Maps API a Google Web Search API není nutná autorizace, tudíž je možné tuto část příručky přeskočit.

Nejdříve si otevřeme libovolný editor a vytvoříme nový php soubor, kde pomocí příkazu „require_once“ načteme knihovny do vlastní aplikace.

```
require_once 'Google/Client.php';
```

Pro autorizaci API využijeme třídu „Google_Client“ nacházející se v klientských knihovnách. Nejdříve vytvoříme objekt client a nastavíme mu potřebné vlastnosti.

```
$this->client = new Google_Client();
$this->client->setApplicationName('G-Teamwork');
$this->client->setClientId("clientID");
$this->client->setClientSecret("clientSecret");
$this->client->setDeveloperKey("APIkey");
$this->client->setRedirectUri("redirectURI");
$this->client->setScopes('https://www.googleapis.com/auth/tasks',
'https://www.googleapis.com/auth/calendar');
```

„ApplicationName“ je název naší aplikace. „ClientId“, „ClientSecret“, „DeveloperKey“ a „RedirectUri“ máme vygenerované již z předchozího kroku, zde tedy doplníme dané údaje. „Scopes“ najdeme v dokumentaci, reprezentují danou API, ke které chceme získat přístup. Každá API má svojí vlastní dokumentaci, kterou lze vyhledat přímo na stránkách společnosti Google.

Samotnou autentizaci provádí metoda „authenticate“, mimo ní použijeme metodu „getAccessToken“ a uložíme získaný token do SESSION, metodou „setAccessToken“ ho přiřadíme objektu client.

```
$this->client->authenticate($_GET['code']);
$_SESSION['access_token'] = $this->client->getAccessToken();
$this->client->setAccessToken($_SESSION['access_token']);
```

Pro vyvolání autorizačního okna ještě vytvoříme odkaz.

```
$authUrl = $this->client->createAuthUrl();  
<a href='$authUrl'>Login</a>
```

Po kliknutí na odkaz získáme přístupový token, jehož platnost je šedesát minut a můžeme začít pracovat s vybranými Google API.

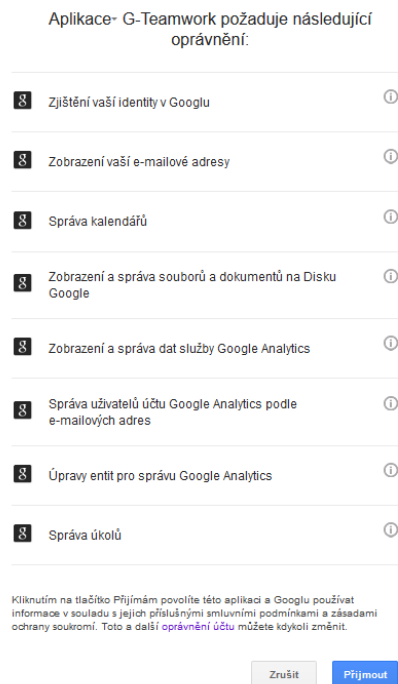
2 Popis aplikace a použitých Google API

Pro vstup do aplikace je nutné mít Google účet, bez něj se do aplikace nepřihlásíme. Aplikaci nalezneme pod tímto odkazem <http://g-teamwork.appspot.com>. Po kliknutí na zmíněný odkaz budeme přesměrováni na hlavní stránku aplikace, kde se nám nejdříve zobrazí tlačítko pro přihlášení.

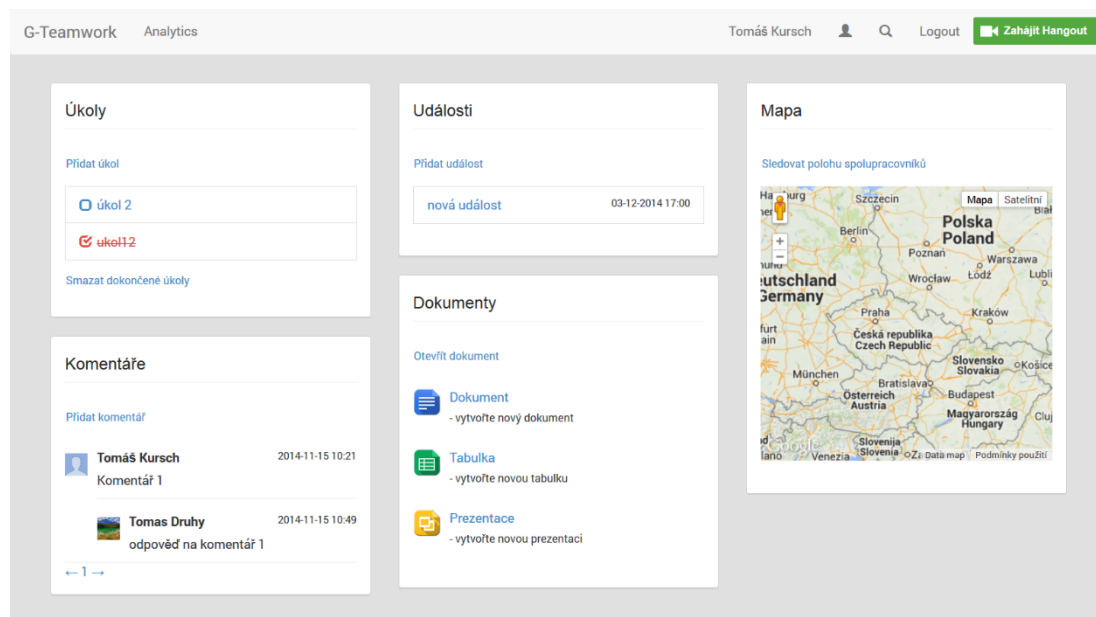
Přihlášení do aplikace je vcelku jednoduché. Klikneme na zmíněné tlačítko a zobrazí se nám takzvané „Consent screen“ okno, ve kterém povolíme aplikaci a Googlu přistupovat k našim soukromým datům. Po přijetí se již toto okno zobrazovat znovu nebude, na rozdíl od tlačítka „Sign in“, které se zobrazí vždy po uplynutí doby platnosti tokenu.



Obrázek 9: Přihlašovací tlačítko



Po úspěšném přihlášení budeme přesměrováni na hlavní stránku aplikace, zde najdeme několik hlavních modulů, jejichž implementaci dále v příručce podobroněji popíší.



Obrázek 10: Hlavní stránka aplikace

Aplikace je zaměřená na týmovou spolupráci s využitím různých Google API. Mezi hlavní funkce aplikace patří práce s úkoly, událostmi, komentáři a dokumenty. V aplikaci lze též sledovat polohu spolupracovníků a na mapě. Komunikaci mezi spolupracovníky zajišťuje služba Hangouts a Gmail. Aplikace též obsahuje vlastní vyhledávač, pomocí Google Analytics API lze získat údaje o návštěvnosti našich webových stránek a využít je pro jejich optimalizaci.

Aplikace využívá klientských knihoven pro PHP. Instalace klientských knihoven je jednoduchá, stačí stáhnout nejnovější verzi z adresy <https://github.com/google/google-api-php-client> a obsah složky zkopírovat do adresáře vlastní aplikace.

V následujícím seznamu se nachází stručný přehled jednotlivých funkcí aplikace a k nim příslušné Google API.

Projekt (nový/stávající)

- vytvoření/získání kalendáře – Google Calendar API, Google Cloud Datastore API
- vytvoření/získání Google dokumentu – Google Drive API, Google Cloud Datastore API

Úkoly

- zobrazení úkolů – Google Taks API
- přidání, dokončení, znovu aktivování a smazání úkolu – Google Tasks API
- smazání všech dokončených úkolů – Google Tasks API
- sdílení úkolů – Google Calendar API

Události

- zobrazení nadcházejících událostí – Google Calendar API
- přidání a smazání události – Google Calendar API
- zobrazení událostí na mapě – Google Maps API, Google Geocoding API
- automatická nabídka adres – Google Places API

Mapa

- interakce s mapou – Google Maps API
- zobrazení polohy spolupracovníků na mapě – Google Maps API, Google Cloud Datastore API, HTML5 Geolocation API

Komentáře

- zobrazení komentářů a odpovědí – Google Drive API
- přidání, smazání komentáře či odpovědi – Google Drive API
- odpověď na komentář – Google Drive API

Dokumenty

- výběr a otevření dokumentů – Google Picker API

Spolupracovníci

- zjištění informací o uživateli – Google+ API
- zobrazení spolupracovníků – Google Drive API
- komunikace se spolupracovníky – Google+ Hangouts API, Gmail API
- přidání spolupracovníka
 - sdílení kalendáře – Google Calendar API

- sdílení Google dokumentu – Google Drive API

Vyhledávání

- vyhledávání na webu, v novinkách, blozích, obrázcích, knihách a videích – Google Web Search API

Analytics

- zobrazení různých projektů – Google Analytics API, Google Charts API
- sdílení projektu s jedním či všemi spolupracovníky – Google Analytics API

3 Implementace Google API v aplikaci

Nejdříve si načteme vybraná API z klientských knihoven a vytvoříme proměnné, pomocí nichž budeme k jednotlivým API přistupovat. Aplikace je objektově orientovaná, proto doporučuji vytvořit pro každou API svojí vlastní třídu.

```
require_once 'Google/Service/Task.php'; // umístíme navrch třídy
$this->taskService = new Google_Service_Tasks($client); // vytvoříme instanční proměnnou

require_once 'Google/Service/Analytics.php';
$this->analytics = new Google_Service_Analytics($client);

require_once 'Google/Service/Calendar.php';
$this->calService = new Google_Service_Calendar($client);

require_once 'Google/Service/Drive.php';
$this->driveService = new Google_Service_Drive($client);

require_once 'Google/Service/Gmail.php';
$this->gmail = new Google_Service_Gmail($client);

require_once 'Google/Service/Plus.php';
$this->plus = new Google_Service_Plus($client);
```

3.1 Úkoly

V aplikaci lze pracovat s úkoly využívajícími Google Tasks API ⁶. Úkoly můžeme vypsat metodou „listTasks“.

```
$tasks = $this->taskService->tasks->listTasks('taskListID')['items'];

foreach ($tasks as $task) {

    echo $task['title'];

}
```

⁶ Dokumentace Google Tasks API - <https://developers.google.com/google-apps/tasks/>

Každý seznam úkolů má své ID („taskListID“), které můžeme zjistit vypsáním všech seznamů úkolů, jež máme vytvořeny.

```
$taskLists = $taskService ->listTasklists()['items'];  
  
foreach ($taskLists as $list) {  
    echo $list['id'];  
}
```

Nový úkol vytvoříme jako objekt „task“, kterému můžeme nastavit různé vlastnosti, například název úkolu či poznámku. Pro vložení zdroje do Google databáze využijeme metody „insert“.

```
$task = new Google_Service_Tasks_Task();  
$task->setTitle('novy ukol');  
$task->setNotes('poznámka k ukolu');  
$this->taskService->tasks->insert('taskListID', $task);
```

Již existujícím úkolům lze měnit status na „completed“ (dokončen), nebo „needsAction“ (aktivní).

```
$task = new Google_Service_Tasks_Task();  
$task->setStatus('completed');  
$this->taskService->tasks->update('taskListID', $taskID, $task);
```

Google Tasks API nepodporuje sdílení úkolů, proto využijeme Google Calendar API, kde změny v úkolech budou prezentovány jako nové události. Nové události přidáme pomocí metody „quickAddEvent“.

```
$this->calendar->quickAddEvent(array('finishTask', $_GET['task']));
```

Úkol č. 1: Smažte libovolný úkol pomocí Google Tasks API

3.2 Události

Dále lze v aplikaci pracovat s událostmi, k čemuž potřebujeme kalendář, který vytvoříme pomocí Google Calendar API ⁷ následujícím způsobem.

```
$calendar = new Google_Service_Calendar_Calendar();  
$calendar->setSummary('nazev kalendare');  
$this->calService->calendars->insert($calendar);
```

Události získáme metodou „listEvents“, kde pomocí volitelných parametrů můžeme specifikovat, jakým způsobem chceme události mít seřazené či z jakého časového rozmezí mají být. Datum musí být ve formátu RFC3339.

```
$opt = array('timeMin' => date(DateTime::RFC3339), 'orderBy' => 'startTime');  
$events = $this->calService->events->listEvents('calendarID', $opt)['items'];  
  
foreach ($events as $event) {  
    echo $event['summary'];  
}
```

Novou událost vytvoříme metodou „insert“, ale ještě předtím specifikujeme datum konání události a místo jejího konání. K vybrání data konání využívá aplikace JQuery plugin „dateTimePicker“ a ke specifikaci místa konání Google Places API ⁸.

```
$event = new Google_Service_Calendar_Event();  
$event->setSummary('nazev_udalosti');  
  
$start = new Google_Service_Calendar_EventDateTime();  
$start->setDateTime(date(DATE_RFC3339, strtotime($_GET['date'])));  
$event->setStart($start);  
$event->setLocation($_GET['location']);  
  
$this->calService->events->insert('calendarID', $event);
```

⁷ Dokumentace Google Calendar API - <https://developers.google.com/google-apps/calendar/>

⁸ Dokumentace Google Places API - <https://developers.google.com/places/>

Pomocí Google Places API se v aplikaci zobrazí nabídka adres, ze kterých můžeme vybírat. Google Places API je postavené na jazyku JavaScript.

```
function places() {  
    return new google.maps.places.Autocomplete(document.getElementById('places'));  
}
```

Událost smažeme metodou „delete“.

```
$this->calService->events->delete('calendarID', 'eventID');
```

Úkol č. 2: Přesuňte událost z vašeho kalendáře do libovolného jiného kalendáře.

3.3 Mapa

V aplikaci lze zobrazit místo konání událostí na mapě či sledovat polohu spolupracovníků v reálném čase. Mapu získáme pomocí Google Maps API ⁹.

K získání přístupu k této API přidáme do hlavičky HTML kódu jednoduchý skript, ve kterém do parametru „key“ umístíme náš API klíč. V různých tutoriálech můžeme nalézt původně povinný parametr sensor (zjištění naší polohy pomocí GPS), tento parametr byl nedávno zrušen a v případě použití bude ignorován.

```
<script src="http://maps.googleapis.com/maps/api/js?key=APIkey"></script>
```

Nyní si vytvoříme objekt „map“, představující samotnou mapu a nastavíme mu další vlastnosti jako je „center“ (na jakých souřadnicích bude mapa vycentrována) a „zoom“ (přiblížení).

```
<script>  
function initialize() {  
    var mapProp = {center:new google.maps.LatLng(51.508742,-0.120850), zoom:5,};  
    var map=new google.maps.Map(document.getElementById("googleMap"),mapProp);  
}
```

⁹ Dokumentace Google Maps API - <https://developers.google.com/maps/documentation/javascript>

```
google.maps.event.addDomListener(window, 'load', initialize);
</script>
```

Posledním krokem je vytvoření bloku na HTML stránce, u kterého musíme specifikovat šířku a výšku.

```
<div id="googleMap" style="width:500px;height:380px;"></div>
```

K zobrazení událostí na mapě využijeme Google Geocoding API ¹⁰, která je určena k převodu adresy na souřadnice.

```
function codeAddress(location) {
  initialize();

  var geocoder = new google.maps.Geocoder();
  var request = {
    address: location
  };
  geocoder.geocode(request, function (results, status) {
    if (status == google.maps.GeocoderStatus.OK) {

      var infowindow = new google.maps.InfoWindow({
        content: '<div>'+ location + '</div>'
      });
      var marker = new google.maps.Marker({
        map: map,
        position: results[0].geometry.location
      });
      map.setZoom(10);
      map.panTo(marker.getPosition());
      infowindow.open(map, marker);
    } else {
      window.console.log('geocoding nebyl úspěšný: ' + status);
    }
  });
}
```

¹⁰ Dokumentace Google Geocoding API - <https://developers.google.com/maps/documentation/geocoding/>

Ke sledování polohy využívá aplikace HTML5 Geolocation API. Tato API určí polohu z IP adresy i GPS. Funkce je v aplikaci dále propojena s PHP a Google Datastore API, kompletní implementaci lze nalézt v příloze.

```
function realtimePosition() {  
  
    if (navigator.geolocation) {  
  
        navigator.geolocation.watchPosition(function(position) {  
  
            var lat = position.coords.latitude;  
            var long = position.coords.longitude;  
            var pos = new google.maps.LatLng(lat,long);  
  
        }, function() {}, {enableHighAccuracy: true});  
    }  
}
```

Úkol č. 3: Označte místo na mapě pomocí markeru na daných souřadnicích (šířka - 52.4, délka - 14.5).

3.4 Komentáře

Aplikace obsahuje systém komentářů a odpovědí, k tomuto účelu využívá Google Drive API ¹¹. K využití komentářů ve vlastní aplikaci nejdříve potřebujeme vytvořit Google dokument, do něhož se budou komentáře ukládat.

```
$data = 'Tymovy_dokumnet';  
$mimetype = 'application/vnd.google-apps.spreadsheet';  
  
$file = new Google_DriveFile();  
$file->setTitle($data);  
$file->setMimeType($mimetype);  
  
$array = array('data' => file_get_contents($data), 'mimeType' => $mimetype);
```

¹¹ Dokumentace Google Drive API - <https://developers.google.com/drive/v2/reference/>

```
$this->drive->files->insert($file, $array);
```

Komentáře získáme metodou „listComments“ a odpovědi metodou „listReplies“

```
$comments = $this->driveService->comments->listComments('fileID');  
foreach ($comments as $comment) {  
    echo $comment['htmlContent'];  
  
    $replies = $this->driveService->replies->listReplies('fileID', $comment['id']);  
    foreach ($replies as $reply) {  
        echo $reply['htmlContent'];  
    }  
}
```

Komentáře lze též vkládat nové.

```
$comment = new Google_Service_Drive_Comment();  
$comment->setContent('Toto je nový komentář');  
$this->driveService->comments->insert('fileID', $comment);
```

Komentář smažeme metodou „delete“.

```
$this->driveService->comments->delete('fileID', 'commentID');
```

Úkol č. 4: Přidejte odpověď k libovolnému komentáři.

3.5 Dokumenty

V aplikaci lze pracovat s dokumenty, zde využijeme Google Picker API ¹². Pomocí této API se v aplikaci zobrazí dialogové okno, které umožňuje vybrat a otevřít dokumenty uložené v Google Drive a dále procházet jednotlivé složky či do Google Drive nahrát libovolný soubor.

¹² Dokumentace Google Picker API - <https://developers.google.com/picker/>

Z důvodu složitosti implementace následující ukázka obsahuje pouze část kódu napsanou v jazyce JavaScript, úplnou implementaci lze najít v příloze.

```
function createPicker() {  
  var picker = new google.picker.PickerBuilder().  
    addView(google.picker.ViewId.DOCS).  
    addView(google.picker.ViewId.FOLDERS).  
    addView(new google.picker.DocsUploadView()).  
    setOAuthToken(oauthToken).  
    setDeveloperKey('API-klíč').  
    setLocale('cs').  
    setCallback(pickerCallback).  
    build();  
  picker.setVisible(true);  
}
```

```
function pickerCallback(data) {  
  if (data[google.picker.Response.ACTION] === google.picker.Action.PICKED) {  
    var doc = data[google.picker.Response.DOCUMENTS][0];  
    var url = doc[google.picker.Document.URL];  
    window.open(url, '_blank');  
  }  
}
```

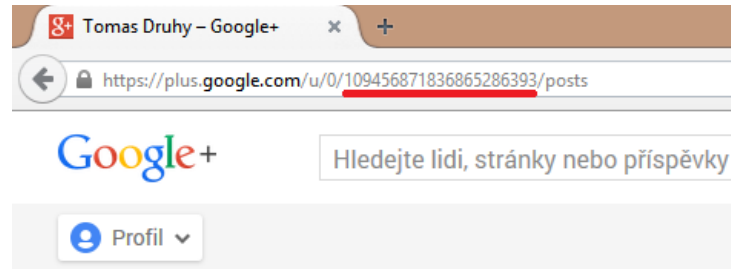
3.6 Spolupracovníci

Aplikace je zaměřená na týmovou spolupráci a využívá k tomu především Google+ API, Google Calendar API, Google Drive API a Gmail API.

Informace o uživateli získáme následujícím způsobem pomocí Google+ API.

```
$result = $this->plus->people->get('ID_uživatele');  
echo $result['displayName']; // vypíše jméno uživatele
```

ID uživatele získáme na Google+ profilu uživatele.



Obrázek 11: ID uživatele na Google+ (podtrženo červeně)

Úkol č. 4: Získejte pomocí Google+ API vlastní emailovou adresu

Spolupracovníky můžeme přidat k Google kalendáři pomocí přístupových práv ACL. K tomu využijeme tříd „AclRule“, ve kterých nastavíme roli uživatele a „AclRuleScope“, kde specifikujeme daného uživatele.

```
$rule = new Google_Service_Calendar_AclRule();  
$scope = new Google_Service_Calendar_AclRuleScope();  
$scope->setType("user");  
$scope->setValue("pepanovak@gmail.com");  
$rule->setScope($scope);  
$rule->setRole("writer");  
  
$this->calService->acl->insert('calendarID', $rule);
```

K Google Dokumentu přidáme spolupracovníky pomocí třídy „Permission“, zde nastavíme roli (owner, reader, writer), typ (user, group, domain, anyone) a emailovou adresu spolupracovníka.

```
$permission = new Google_Service_Drive_Permission();  
$permission->setRole('writer');  
$permission->setType('user');  
$permission->setValue('pepanovak@gmail.com');  
  
$this->driveService->permissions->insert('fileID', $permission);
```


V aplikaci se nachází seznam spolupracovníků, který vypíšeme pomocí Google Drive API metodou „listPermissions“.

```
$this->driveService->permissions->listPermissions('fileID')['items'];
```

Každému spolupracovníkovi lze také poslat zprávu pomocí Gmail API ¹³. Zprávu lze odeslat pouze v „raw“ formátu, který se zakóduje. Stačí vyplnit odesílatele „From“ („me“ značí nás), „To“ (příjemce), „Subject“ (předmět) a nakonec obsah zprávy.

```
$message = new Google_Service_Gmail_Message();  
$raw = "From: me\r\n To:adresa_prijemce\r\n Subject: předmět\r\n\r\n obsah zprávy";  
$message->setRaw(base64_encode($raw));  
$this->gmail->users_messages->send('me', $message);
```

3.7 Vyhledávání

Aplikace obsahuje vlastní Google vyhledávač využívající Google Web Search API ¹⁴. Vyhledávat můžeme na webu, v novinkách, blozích, obrázcích, knihách a videích. Google Web Search API též podporuje vyhledávání na vlastním webu, ale tato funkce není v aplikaci implementována, jelikož aplikace využívá především soukromá data.

```
google.load("search", "1", {"language": "en"});  
  
function initialize() {  
    var searchControl = new google.search.SearchControl();  
    searchControl.addSearcher(new google.search.WebSearch());  
    searchControl.addSearcher(new google.search.BlogSearch());  
    searchControl.addSearcher(new google.search.ImageSearch());  
    searchControl.addSearcher(new google.search.BookSearch());  
    searchControl.addSearcher(new google.search.VideoSearch());  
  
    var drawOptions = new google.search.DrawOptions();  
    drawOptions.setDrawMode(google.search.SearchControl.DRAW_MODE_TABBED);  
    searchControl.draw(document.getElementById("searchcontrol"), drawOptions);  
}
```

¹³ Dokumentace Gmail API - <https://developers.google.com/gmail/api/>

¹⁴ Dokumentace Google Web Search API - <https://developers.google.com/web-search/>

Úkol č. 5: Přidejte vyhledávací funkci vyhledávání na webu www.seznam.cz.

3.8 Hangout

V aplikaci lze využít ke komunikaci se spolupracovníky služby Google Hangout, která poskytuje jak videohovory, tak i klasický chat. Ke spuštění Google Hangout z vlastní aplikace implementujeme tlačítko „Zahájit hangout“ dostupné v rámci Google+ API ¹⁵.

```
<script src="https://apis.google.com/js/platform.js" async defer></script>  
<div class="g-hangout" data-render="createhangout"></div>
```

3.9 Analytics

Aplikace kromě hlavní stránky obsahuje ještě jednu stránku s názvem „Analytics“. Účelem této stránky je vizualizace specifických dat služby Google Analytics. Konkrétně se jedná o počet návštěvníků v jednotlivých dnech za posledních třicet dnů, poměr nových a vracejících se návštěvníků, rozdělení uživatelů podle různých prohlížečů a zobrazení návštěvníků podle států. Díky těmto údajům můžeme lépe optimalizovat náš web pro prohlížeče, přidávat různé jazykové verze webu a sledovat nárůst či pokles návštěvníků.

Google Analytics API ¹⁶ je velmi obsáhlé a skládá se z vícero API. Tato aplikace využívá pouze Core reporting API ¹⁷ (pro práci s přehledy) a Management API ¹⁸ (pro správu uživatelů).

Nezákladnější metodou pro práci s Google Analytics Management API je metoda „listManagementAccountSummaries()“, pomocí které zjistíme důležité informace vztahující se k našemu Google Analytics účtu. Nejdůležitějším parametrem je „ProjectID“, který využijeme v dalších ukázkách.

¹⁵ Dokumentace Google+ Hangouts - <https://developers.google.com/+/hangouts/button>

¹⁶ Dokumentace Google Analytics API - <https://developers.google.com/analytics/>

¹⁷ Dokumentace GA Core Reporting API - <https://developers.google.com/analytics/devguides/reporting/core/v3/>

¹⁸ Dokumentace GA Management API - <https://developers.google.com/analytics/devguides/config/mgmt/v3/>

```
$this->analytics->management_accountSummaries->listManagementAccountSummaries();
```

Pomocí Core Reporting API získáme počet návštěvníků v každého dnu za posledních 30 dnů. Důležitými parametry jsou „metrics“ a „dimensions“, které najdeme na <https://developers.google.com/analytics/devguides/reporting/core/dimsmets>.

```
$from = date('Y-m-d', time()-30*24*60*60); // 30 dnů
$to = date('Y-m-d', time()-24*60*60); // dnes

$metrics = 'ga:visits'; // metrika pro získání počtu návštěvníků
$dimensions = 'ga:date'; // datum pro danou metriku

$this->analytics->data_ga->get('ga:ProjectID', $from, $to, $metrics, $dimensions);
```

Obdobným způsobem jako v předešlé ukázce lze získat mnoho dalších přehledů, stačí pouze pozměnit dané metriky.

Kromě práce s přehledy je možné také v aplikaci spravovat uživatele pomocí Google Analytics Management API. V následující ukázce přidáme uživatele, který může číst a analyzovat náš vlastní projekt. „WebPropertyID“ a „AccountID“ získáme pomocí metody „listManagementAccountSummaries()“.

```
$userRef = new Google_Service_Analytics_UserRef();
$userRef->setEmail('email_spolupracovnika');

$permissions = new Google_Service_Analytics_EntityUserLinkPermissions();
$permissions->setLocal(array('COLLABORATE', 'READ_AND_ANALYZE'));
$link = new Google_Service_Analytics_EntityUserLink();
$link->setPermissions($permissions);
$link->setUserRef($userRef);

$this->analytics->management_webpropertyUserLinks->insert('webPropertyID', 'accountID', $link);
```

V aplikaci lze také přepínat mezi různými projekty, pokud jich máme více můžeme je následně sdílet buď s jedním libovolným uživatelem nebo se všemi spolupracovníky.

Visualizovat data z Google Analytics lze pomocí Google Charts API ¹⁹. Následující ukázka vykreslí spojnicový graf zobrazující počet návštěv za každý den po posledních 30 dnech.

```
google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);

function drawChart() {
  var data = google.visualization.arrayToDataTable([
    ['datum', 'počet návštěv'],
    <?php
      foreach ($analytics->getVisits()['rows'] as $row) {
        echo "[".$row[3].", ".$row[2].", ".$row[4].", "];";
      }
    ?>
  ]);
  var options = {
    title: 'Návštevnost',
    vAxis: {minValue: 0}
  };
  var chart_div = document.getElementById('chart_div');
  var chart = new google.visualization.AreaChart(chart_div);
  chart.draw(data, options);
}
```

Úkol č. 6: Vypište počet nových a vracejících se návštěvníků pomocí Google Analytics API.

3.10 Databáze

Aplikace využívá databázi společnosti Google, ke které přistupuje pomocí Google Cloud Datastore API ²⁰. Práce s Datastore API je trochu složitější než u ostatních API. Nejdříve musíme na stránce <https://console.developers.google.com> vygenerovat nové autentizační údaje.

¹⁹ Dokumentace Google Charts API - <https://developers.google.com/chart/>

²⁰ Dokumentace Google Cloud Datastore API - <https://cloud.google.com/datastore/docs/apis/v1beta2/>

Vybereme v levém menu „APIs & auth“ záložku „Credentials“, kde klikneme na tlačítko „Create a new Client ID“. Zde máme na výběr mezi třemi možnostmi: „Web application“, „Service account“ a „Installed application“. Zvolíme možnost „Service Account“ a dojde k vygenerování „Client ID“ a souboru obsahujícího privátní klíč. Privátní klíč si uložíme do root adresáře naší aplikace.

```
$google_api_config = [  
  'application-id' => 'nazev aplikace',  
  'service-account-name' => 'service_account_name',  
  'private-key' => file_get_contents('cesta_k_privatnimu_klici'),  
  'dataset-id' => 'datasetID'  
];
```

Google Cloud Datastore API vyžaduje tři třídy z klientských knihoven.

```
require_once 'Google/Client.php';  
require_once 'Google/Auth/AssertionCredentials.php';  
require_once 'Google/Service/Datastore.php';
```

Autentizaci provedeme pomocí třídy „AssertionCredetials“.

```
$client = new Google_Client();  
$client->setApplicationName($this->config['appId']);  
$client->setAssertionCredentials(new Google_Auth_AssertionCredentials($this->config['service-account-name'], 'https://www.googleapis.com/auth/datastore', $this->config['privateKey']));
```

Nyní si již vytvoříme instanci „datastore“, pomocí níž budeme provádět požadavky na danou API:

```
$this->datastore = new Google_Service_Datastore($client);
```

Metodou „lookup“ získáme požadované entity a metodou „commit“ provedeme databázové operace „insert“, „update“ a „delete“. Z důvodu komplexnosti těchto operací lze i implementaci těchto metod najít v příloze bakalářské práce.

```
$this->datastore->databases->lookup($this->config['datasetId'], $entity, []);  
$this->datastore->databases->commit($this->config['datasetId'], $entity, []);
```

4 Závěr

V této příručce jsem popsal většinu Google API, jimiž jsem se zabýval v rámci bakalářské práce a následně je využil při vývoji aplikace. Během tohoto vývoje jsem se však setkal s vícero API, jež jsem z určitých důvodů nepoužil, ale jejich implementaci jsem zachoval, lze je nalézt v adresáři aplikace „old“, která se nachází v příloze bakalářské práce.

Z aplikace byly odstraněny:

- **Google Contacts API** (API pro správu Google kontaktů) - implementace této API je poněkud složitější, jelikož pro ní není třída v klientských knihovnách.
- **Google SpreadSheets API** (API pro správu a editaci obsahu Google SpreadSheets dokumentu) – tato API již není společností Google podporována, ale lze s ní pracovat pomocí knihovny „Gdata“, jež je nabízena v rámci frameworku Zend
- **Google Hangouts API** - (API k přizpůsobení Hangouts aplikace) – nahrazeno tlačítkem „Hangouts“ dostupného v rámci Google+ API

Na závěr bych chtěl podotknout, že ačkoli se čtenář v rámci této příručky může naučit mnoho nového o Google API, mohou mu práci s API zneprůjemnit různé buggy, jelikož jsou klientské knihovny pro PHP stále v beta verzi. Tato příručka dále předpokládá čtenářovu znalost jazyka PHP a objektově orientovaného programování. Pokud se čtenář zatím neseťkal s žádným aplikačním programovým rozhraním, může pro něj být tato příručka velmi přínosnou.

Řešení úkolu č. 1:

```
$this->taskService->tasks->delete('taskListID', 'taskID');
```

Řešení úkolu č. 2:

```
$this->calService->event->move('primary', 'eventID', 'ID_cílového_kalendáře');
```

Řešení úkolu č. 3:

```
var myLatLng = new google.maps.LatLng(52.4,14.5);
var marker = new google.maps.Marker({
    position: myLatLng,
    map: map,
    title:"Popisek"
});
```

Řešení úkolu č. 4:

```
$reply = new Google_Service_Drive_CommentReply();
$reply->setContent('nová odpověď');
$this->driveService->replies->insert('fileID', 'commentId', $reply);
```

Řešení úkolu č. 5:

```
$plus = $this->plus->people->get('me');
echo $plus['emails'][0]['value'];
```

Řešení úkolu č. 6:

```
var siteSearch = new google.search.WebSearch();
siteSearch.setUserDefinedLabel("www.seznam.cz");
siteSearch.setUserDefinedClassSuffix("siteSearch");
siteSearch.setSiteRestriction("www.seznam.cz");
searchControl.addSearcher(siteSearch);
```

Řešení úkolu č. 7:

```
$from = date('Y-m-d', time()-30*24*60*60); // 30 dnů
$to = date('Y-m-d', time()-24*60*60); // dnes
$metrics = 'ga:sessions';
$dimensions = 'ga:userType';
$result = $this->analytics->data_ga-
>get('projectID', $from, $to, $metrics, $dimensions);
print_r($result);
```