



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Propojení SugarCRM s kalendářem Google
Student:	Karel Čech
Vedoucí:	Ing. Pavel Pohořský, MIM, FICS
Studijní program:	Informatika
Studijní obor:	Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Cílem práce je návrh a implementace integrace formou obousměrného propisu aplikací Google Calendar a SugarCRM.

1. Seznamte se se strukturou uložených dat a s rozhraními obou aplikací.
2. Stručně popište existující řešení integrace těchto dvou systémů a jejich nedostatky.
3. Navrhněte integraci formou obousměrného propisu společných údajů v obou systémech.
4. Zvolte vhodné prostředí a integraci implementujte.
5. Svě řešení důkladně otestujte a zdokumentujte.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 6. února 2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Propojení SugarCRM s Google kalendářem

Karel Čech

Vedoucí práce: Ing. Pavel Pohořský, MIM, FICS

15. května 2017

Poděkování

Chtěl bych poděkovat firmě SugarFactory, která mi umožnila tuto bakalářskou práci napsat v rámci jejího projektu. Dále bych chtěl poděkovat Vojtěchu Černému a Pavlovi Pohořskému za jejich podporu a odborné rady v průběhu implementace řešení a psaní této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. Dále prohlašuji, že jsem s Českým vysokým učení technickým v Praze uzavřel dohodu, na základě níž se ČVUT vzdalo práva na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona. Tato skutečnost nemá vliv na ust. § 47b zákona č. 111/1998 Sb., o vysokých školách, ve znění pozdějších předpisů.

V Praze dne 15. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Karel Čech. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Čech, Karel. *Propojení SugarCRM s Google kalendářem*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Tato bakalářská práce se zabývá propojením webové aplikace SugarCRM s Google kalendářem. Cílem bylo vytvořit univerzální rozšíření, které toto propojení umožní. Rozšíření bude fungovat obousměrně a okamžitě, čímž by mělo předčít většinu ostatních řešení nabízejících toto propojení, jako například Collabspot nebo GSync. Práce by měla čtenáři ukázat možnosti a úskalí při vytváření rozšíření pro SugarCRM a synchronizace dvou kalendářů.

Klíčová slova SugarCRM, Google kalendář, Google API, Webová aplikace, Synchronizace

Abstract

This bachelor thesis deals with linking the SugarCRM web application with the Google Calendar. The aim was to create a universal extension that allows this connection. The extension will work bidirectionally and instantly, surpassing most other solutions, which connects SugarCRM and Google Calendar, such as Collabspot or GSync. The work should show the reader all the possibilities and pitfalls of extending SugarCRM and synchronization of two calendars.

Keywords SugarCRM, Google Calendar, Google API, Web application, Synchronization

Obsah

Úvod	1
Cíl práce	1
Struktura práce	1
1 Analýza	3
1.1 SugarCRM	3
1.1.1 O Aplikaci	3
1.1.2 Struktura	4
1.1.2.1 Moduly	4
1.1.2.2 Frontend	5
1.1.2.3 Backend	6
1.1.2.4 Databáze	6
1.1.3 Rozšíření	6
1.1.3.1 LogicHooky	7
1.1.3.2 REST API	7
1.1.3.3 Javascript	7
1.1.4 Naplánované úlohy	7
1.1.5 Kalendář	8
1.1.5.1 Události	8
1.2 Google	8
1.2.1 Autentizace	9
1.2.2 Google API Client Library	10
1.2.3 Google kalendář	11
1.2.3.1 Calendar API	11
1.2.3.2 Push notifikace	12
1.2.3.3 Události	12
1.3 Existující řešení	13
1.3.1 Collabspot	13
1.3.2 RT GSync	13

1.3.3	iCalendar protokol	14
1.4	Požadavky	14
1.4.1	Funkční	14
1.4.2	Nefunkční	14
2	Návrh	15
2.1	Shrnutí	15
2.2	Spuštění synchronizace	15
2.3	Synchronizace událostí	16
2.3.1	Nová pole na událostech	16
2.3.2	Nastavení Push notifikací	16
2.3.3	Ze SugarCRM do Google kalendáře	16
2.3.4	Z Google kalendáře do SugarCRM	16
2.3.5	Opakované události	17
2.4	Změny vlastníků událostí	17
2.5	Změna Google kalendáře	18
2.6	Nový modul v SugarCRM	18
3	Realizace	21
3.1	Vývoj	21
3.1.1	Vývojové prostředí	21
3.1.2	Verzování	22
3.1.3	Dokumentace	22
3.1.4	Testování	22
3.2	Řešení	23
3.2.1	Třídní diagram	23
3.3	Nedostatky	23
3.4	Návrhy na vylepšení	24
	Závěr	27
	Literatura	29
	A Vysvětlení odborných pojmů	31
	B Seznam použitých zkratk	33
	C Obsah příloženého CD	35

Seznam obrázků

1.1	Ukázka z aplikace SugarCRM[16]	4
1.2	Struktura aplikace SugarCRM[18]	5
1.3	Kalendář v SugarCRM	8
1.4	Vytváření nového záznamu schůzky v SugarCRM. Záznam již obsahuje přidání pole a vazby	9
1.5	Proces získávání přístupových tokenů [4]	10
1.6	Google kalendář	11
1.7	Detail události v Google kalendáři	13
3.1	Modální okno pro výběr kalendáře	23
3.2	Třídní diagram	24

Úvod

SugarCRM je webová aplikace určená pro střední až velké společnosti, která ulehčuje uživatelům udržovat, prohlížet a dokonce i získávat informace o potenciálních i aktuálních klientech, aktivitách, obchodních případech, dokumentech a spoustě dalších věcí, o kterých potřebují mít přehled pro lepší chod své společnosti. Aplikace je plně rozšiřitelná, dá se v ní proto vytvořit či upravit úplně cokoliv.

V základu SugarCRM nabízí i možnost evidovat si své hovory, schůzky a úkoly. Právě tyto položky by bylo potřeba synchronizovat s Google kalendářem, aby uživatelé, kteří ho používají jako svůj primární kalendář, měli své schůzky přehledně k dispozici na obou platformách a mohli využít všechny jejich výhody. Datové modely jsou si velmi podobné, takže v tomto ohledu by propojení nemělo být problém. Nějaké odlišnosti však přeci jenom existují a kvůli tomu se nepůjde obejít bez kompromisů.

Cíl práce

Cílem práce je vytvořit spolehlivou, rychlou a zároveň jednoduchou synchronizaci mezi webovými aplikacemi SugarCRM a Google kalendářem. Dalším cílem je popsat základní postupy při rozšiřování SugarCRM a postupy při synchronizaci kalendářů.

Struktura práce

V první kapitole jsou rozebrány aplikace SugarCRM a Google kalendář a jejich funkcionality související s touto prací. V druhé kapitole je navrženo možné řešení tohoto problému. Ve třetí kapitole je rozebrána samotná realizace. V přílohách A a B jsou vysvětleny některé odborné pojmy a zkratky.

Analýza

1.1 SugarCRM

SugarCRM je velmi rozsáhlá aplikace a samotné její podrobné popsání by vystačilo na samostatnou práci. Proto je SugarCRM níže popsána pouze velmi stručně a trochu více jsou rozebrány pouze části týkající se této bakalářské práce.

1.1.1 O Aplikaci

SugarCRM [17] je placená webová aplikace používaná po celém světě a zároveň firma sídlící v Kalifornii, která tuto aplikaci vlastní a spravuje. Firma byla založena roku 2004. Jejím hlavním účelem je si přehledně evidovat informace o zákaznících, partnerech, zaměstnancích atd., interakce s nimi a následně z těchto dat získávat díky propracovanému reportovacímu systému užitečné informace. Díky SugarCRM mohou různé oddělení firem (například podpora, marketing, obchodní oddělení) velmi rychle a účinně sdílet informace bez zbytečné administrativy a duplikování dat. Tuto aplikaci mohou velmi efektivně používat například call centra, finanční poradenství nebo oddělení firemní podpory. SugarCRM by ale mělo být možné přizpůsobit pro potřeby kohokoliv i proto, že je open-source.

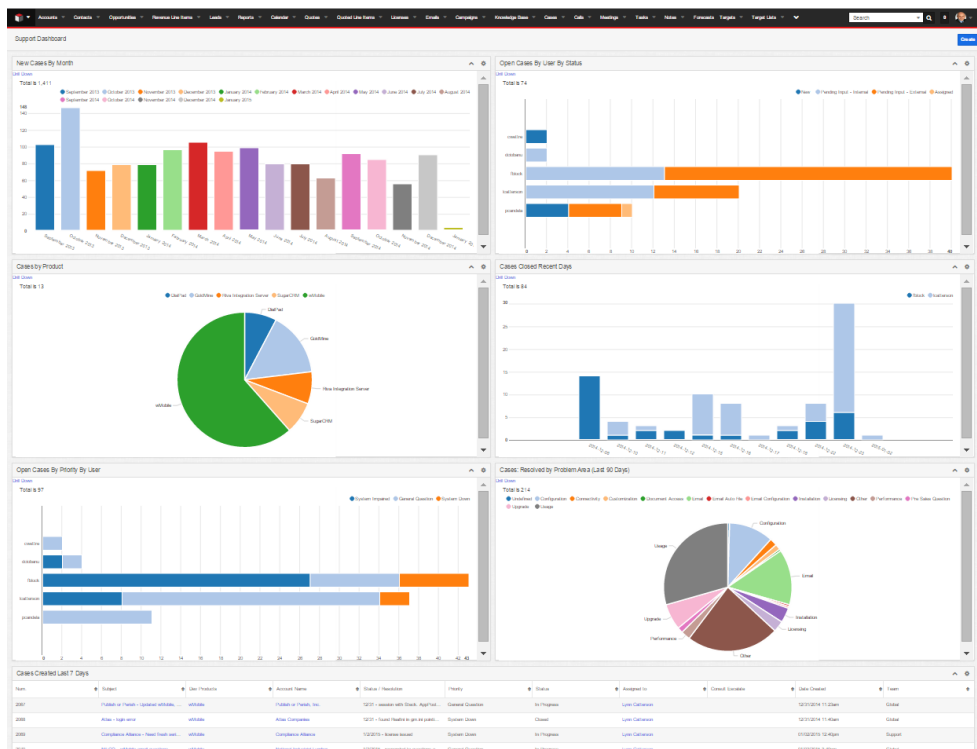
Mezi největší konkurenty a velmi podobné aplikace patří například SalesForce nebo Microsoft Dynamics, které ale na rozdíl od SugarCRM nejsou open-source. SugarCRM používají i velké světové firmy jako IBM nebo HTC.

SugarCRM se postupně vyvíjí a existuje ve velkém množství verzí. Existuje i volně dostupná verze „Community Edition“, která ale neobsahuje spoustu funkcionalit jako například Sidecar framework a SugarCRM za ní nijak neručí.

Vše níže popsané platí pro placenou verzi 7.6.0.0. Pro tuto verzi se bude týkat i implementace synchronizace s Google kalendářem. Také existují různé varianty, takzvané „příchutě“ (orig. „flavours“). Varianty určují, které funkci-

1. ANALÝZA

Obrázek 1.1: Ukázka z aplikace SugarCRM[16]



onality mají uživatelé dostupné, a liší se v ceně. Použitá varianta je Enterprise, implementace a analýza by však neměla být variantou nijak ovlivněna.

Veškeré níže popsané informace lze podrobněji najít v manuálu [19], který je přiložený k této práci.

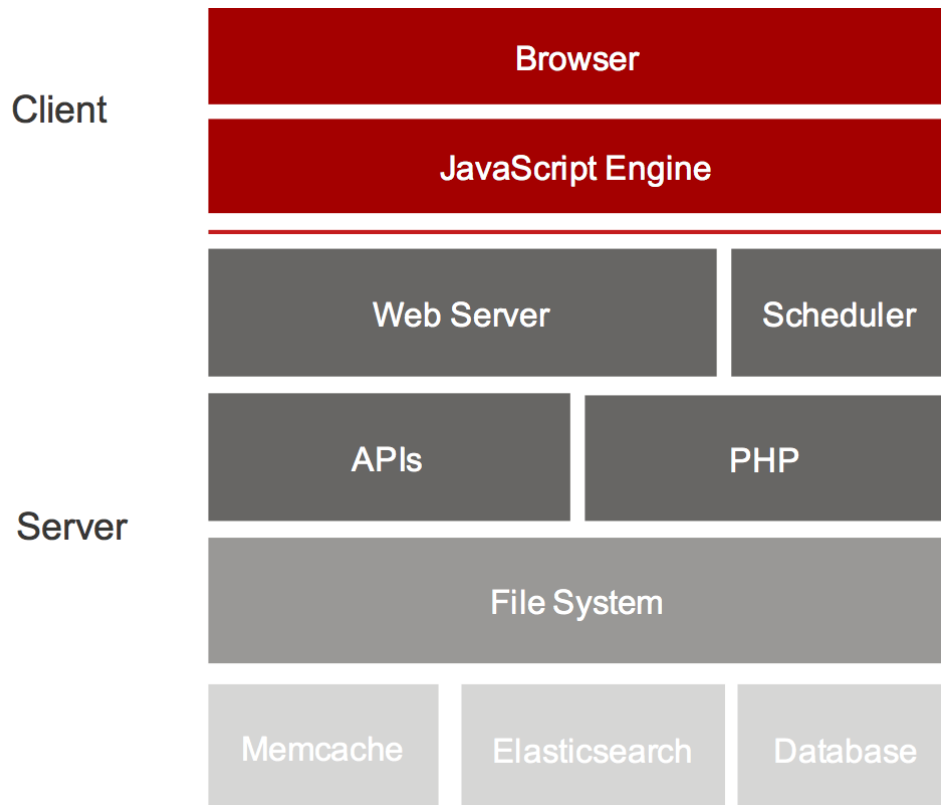
1.1.2 Struktura

SugarCRM je klasická webová aplikace, skládající se z propracovaného front-endu psaném v Javascriptu, třívrstevném backendu v jazyku PHP a databázovým serverem. Backend podporuje více typů databází (MySQL, Microsoft SQL server, DB2, Oracle). Backendová část poskytuje plnohodnotné REST API, přes které frontend získává ve většině případů data. Z důvodu postupného vývoje však některé komponenty front-endu Javascript ani API nepoužívají a jsou stále vykreslovány na serveru. Proto se nedá říct, že je SugarCRM postavené na plnohodnotné architektuře řízené přes API.

1.1.2.1 Moduly

Hlavními prvky SugarCRM jsou takzvané „moduly“, které reprezentují jednotlivé typy záznamů a záznamy samotné anebo systémové funkcionality, jako

Obrázek 1.2: Struktura aplikace SugarCRM[18]



například logování aktivit, konfigurace, domácí obrazovka a jiné. Pro jednoduchost však budeme za modul považovat pouze takový modul, který reprezentuje uživatelsky přístupné a použitelné záznamy, jako například Firmy, Schůzky, Kontakty, Smlouvy a podobně.

Modul je zjednodušeně řečeno tabulka v databázi a jeden záznam reprezentuje řádek v této tabulce spolu se všemi vazbami.

1.1.2.2 Frontend

Frontend SugarCRM je z velké části tvořen frameworkem postaveným nad knihovnamy Backbone.js, jQuery, Handlebars.js, a Bootstrap, nazývaný Sidecar založený na MVC architektuře. [14] Pohledy, rozložení, modely, pole a další jsou převážně řízeny metadatami, které jsou získávány ze serveru. Některé starší komponenty, jako třeba kalendář, jsou však stále vykreslovány na straně serveru pomocí jazyka PHP s použitím Smarty šablon.

1.1.2.3 Backend

Backend je objektově orientovaný a tvořen čistým PHP verze 5.6. Jsou zde popsány všechny prvky systému a modulů, definováno REST API, napojení na databázi a vykonávána většina bussiness logiky (kvůli bezpečnosti). Každý modul zde má svoji třídu.

1.1.2.4 Databáze

V databázi jsou vykonávány pouze obyčejné operace, proto je možné k SugarCRM použít více různých databází. Doporučený typ databáze je MySQL. V podstatě platí, že co model, to tabulka. Veškeré vztahy mezi modely jsou popsány v separátních tabulkách.

Platí, že jeden záznam v SugarCRM odpovídá jedné řádce v tabulce odpovídajícího modulu.¹

Všechny vazby jsou v separátních tabulkách, aby k nim bylo možné přidávat další atributy.

1.1.3 Rozšíření

Kód SugarCRM lze rozdělit na dvě části – na jádro a na „custom“ kód. V kořenové složce SugarCRM je složka „custom“, která napodobuje strukturu kořenové složky. Například modul „Accounts“ je definován v „modules/Accounts“ rozšíření pro tento modul by tedy bylo v „modules/custom/Accounts“. V této složce by měly být prováděny všechny kódové úpravy SugarCRM.

Teoreticky by tedy při jakémkoli rozšíření měla být použita používat jenom custom část, která je takzvaně „upgrade-safe“. To znamená, že pro přechodu na vyšší verzi SugarCRM nejsou úpravy přemazány. Prakticky však některé úpravy systém rozšiřování nepodporuje a je potřeba upravit soubory v jádru. Pokud jsou soubory upraveny, je za potřeby při případné změně verze tyto soubory znovu upravit.

Existují dva typy rozšíření. První funguje tak, že je na určené místo ve složce custom dán soubor s určeným jménem, který je načten místo původního souboru z jádra. Cesta k novému souboru většinou kopíruje cestu k souboru původnímu, jen je do cesty přidána složka custom. Tímto způsobem jsou především rozšiřovány základní třídy modulů z jádra, kde custom třída dědí z původní třídy z jádra a změni či přidá nějaké metody.

Druhý typ rozšíření nazývaný „Extension framework“ je používán především na změnu a rozšíření základních metadat. Zde nejsou upravovány třídy, ale pouze různá globální pole obsahující metadata a další definice (např. LogicHooky). Pro tento druh rozšíření je ve složce custom složka „Extensions“, která je rozdělená do dalších podsložek, z nichž SugarCRM jednotlivé soubory

¹Prakticky je většina modulů v databázi rozdělena na dvě tabulky. Jedna s původními sloupci a druhá s uživatelsky přidanými.

sloučí do jednoho souboru, který je vytvořen v jiném umístění a který je poté reálně načten při provozu aplikace.

1.1.3.1 LogicHooky

LogicHook je nazývána serverová logice, která se vykoná při nějaké akci. Typicky jsou LogicHooky využívány před uložením, kdy je potřeba na serveru ukládaný záznam ještě nějak před uložením upravit nebo verifikovat. Existuje ale i celá řada jiných LogicHooků jako například při přihlášení uživatele, při načtení záznamu z databáze, při smazání záznamu, při přijetí požadavku přes API a spousta dalších.

1.1.3.2 REST API

REST API SugarCRM lze také upravovat a rozšiřovat. Lze definovat nové cesty i s proměnnými parametry přímo v cestě a to pro 4 základní CRUD operace. Třída, která API rozšiřuje musí dědit z třídy SugarApi a ve specifikované metodě vracet pole s definicí pro cestu, kde je například určeno, jaká metoda výše zmíněné třídy má být zavolána, jestli je potřeba být přihlášen (respektive použít autentizační tokeny), prioritu apod.

SugarCRM má logiku pro vyhodnocování priority jednotlivých cest, pokud jsou cesty nejednoznačné. To znamená, že pokud je jedna cesta definována jako „/google/<id>“, kde „<id>“ je proměnná, a druhá cesta jako například „/google/all“, tak je automaticky přiřazena priorita té druhé a tím pádem je i spuštěna logika k ní přiřazená. To však znamená, že nějaký záznam nemůže mít identifikátor „all“, protože by se na něj nešlo dotázat.

1.1.3.3 Javascript

Rozšíření Javascriptu lze rozdělit na dvě části. První typ rozšíření je kód, který je připojen k „hlavnímu“ javascriptovému kódu SugarCRM a je tedy spuštěn okamžitě při spuštění aplikace. Takto jsou například dají zaregistrovány různé události, či měněny nějaké základní třídy.

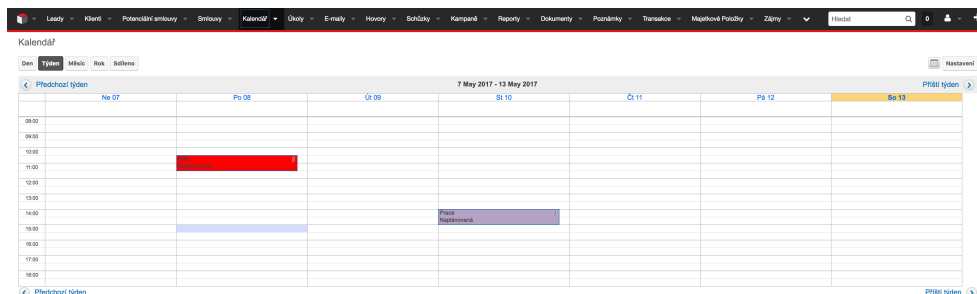
Druhý typ rozšíření je použit při úpravě či vytváření nových rozložení, pohledů anebo polí. Kód je technicky také spuštěn při načtení, ale prakticky je do systému nahrána nová třída, která poté lze používat v rámci aplikace. Tyto třídy reprezentují různé pohledy, rozložení a pole.

1.1.4 Naplánované úlohy

SugarCRM dokáže periodicky spouštět funkce v určeném časovém intervalu. Takovýmto funkcím se říká naplánované úlohy.

1. ANALÝZA

Obrázek 1.3: Kalendář v SugarCRM



1.1.5 Kalendář

Kalendář v SugarCRM slouží pro zobrazení záznamů v modulech Schůzky, Hovory (dohromady události) a Úkoly na časové ose. Lze si vybrat pohled na různá časová období i pro více jiných uživatelů najednou.

Kalendář je modul ze verze 6 v režimu zpětné kompatibility. Prakticky to znamená, že je vykreslován pomocí PHP a nepoužívá API. To způsobuje, že je velmi těžké ho rozšířit upgrade-safe metodou, jelikož staré moduly nemají příliš dobrou podporu pro takovéto rozšíření.

Jednotlivé záznamy v kalendáři jsou otevřeny po kliknutí na událost na nové stránce se všemi informacemi.

Kalendář v SugarCRM slouží spíše pro zaznamenání si informací o událostech (co se na nich řešilo, čeho se týkají, kdo se události zúčastnil), spíše než pro správu a plánování času uživatele.

1.1.5.1 Události

Události mají v základu pole pro název, polohu, pozvané uživatele a možnosti pro nastavení upozornění opakování a upozornění. Další pole či vazby na jiné moduly však lze přidávat podle libosti.

Každá událost v SugarCRM má k sobě přiřazeného uživatele z . Události mají vazbu typu M:N ke třem modulům – Uživatelé, Kontakty a Potenciální zákazníci, které jsou zobrazeny u záznamu události jako pozvaní. U vazby je také zachycen stav pozvaného – jestli se zúčastní, neodpověděl, možná nebo ne. Jinak v základu události nemají vazbu na žádný jiný modul.

Opakované schůzky jsou řešeny tak, že první schůzka je takzvaně mateřská a všechny další schůzky v opakování na ní odkazují.

1.2 Google

Google je světově známa společnost poskytující především webové služby. Tato práce se nebude věnovat popsání této společnosti a všech jejích produktů. Více

Obrázek 1.4: Vytváření nového záznamu schůzky v SugarCRM. Záznam již obsahuje přidávané pole a vazby

The screenshot displays the SugarCRM 'Meetings' form. At the top, there is a navigation bar with various modules like Cases, Work Logs, Reports, Accounts, Contacts, Leads, Opportunities, Calls, Meetings, Campaigns, Emails, Tasks, and Notes. The main form area includes:

- Subject:** A text field with the value 'Required' and a 'Status' dropdown set to 'Scheduled'. Buttons for 'Cancel' and 'Save' are visible.
- Start Date:** 13.05.2017, **End Date:** 13.05.2017, **Start Time:** 21:00, **End Time:** 21:30. A 'Repeat Type' dropdown is set to 'Select...'.
- Location:** An empty text field.
- Description:** A large text area for notes.
- Meeting Type:** A dropdown menu currently showing 'Sugar'.
- Related to:** A dropdown menu set to 'Account' with a 'Select...' button.
- Guests:** A calendar grid showing dates from 17 to 23. A 'No Reply' button is present.
- Assigned to:** A dropdown menu showing 'Karel Čech'.
- Teams:** A dropdown menu showing 'Global' with a star icon and a plus sign.
- Telesales:** A field with the text 'Select Telesales...'.
- Tags:** An empty text field.
- Date Modified:** No data, **Date Created:** No data.

informací o Googlu lze získat z knížky In the Plex: How Google Thinks, Works, and Shapes Our Lives[11].

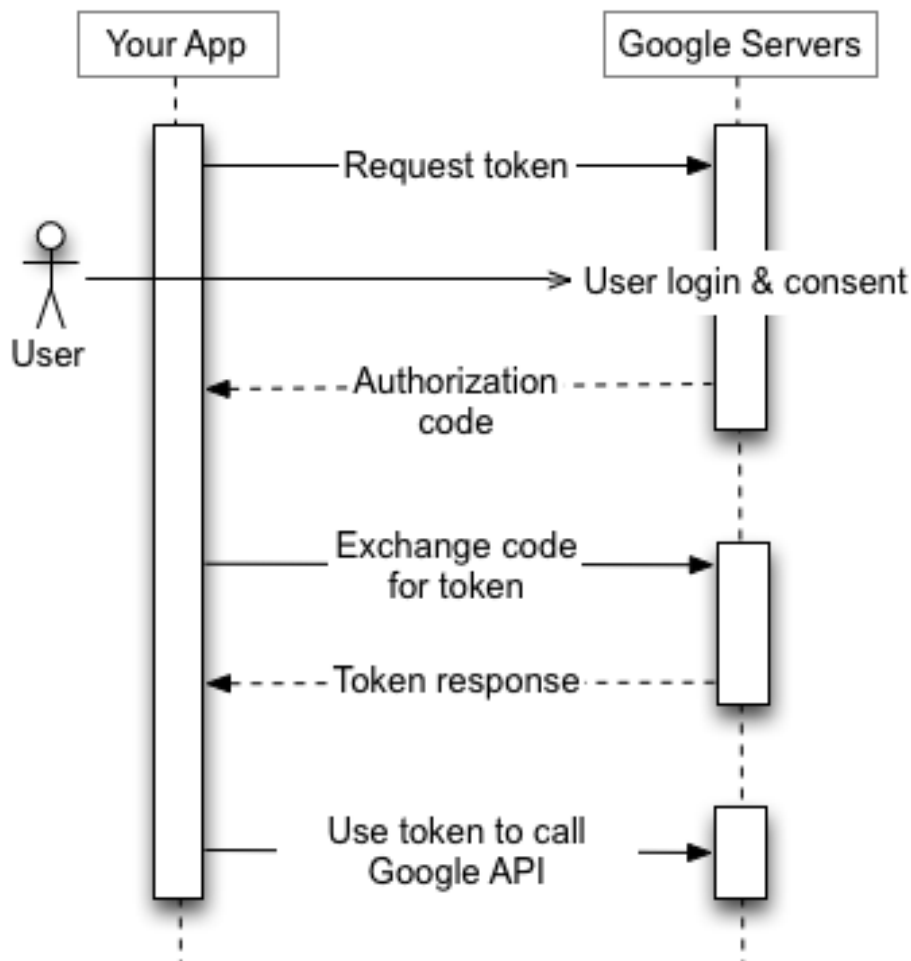
Popsány jsou pouze ty části, potřebné k praktické části této práce, což jsou z větší Google API a Google kalendář.

1.2.1 Autentizace

První věc, kterou je třeba vyřešit při přístupu ke Google API, je autentizace uživatele. Pro přístup k záznamům určitého uživatele není zapotřebí jeho přihlašovací jméno ani heslo. Pro přístup k datům je potřeba, získat uživatelské povolení používat konkrétní aplikaci Googlu s konkrétními právy (čtení, zápis, mazání). Toto povolení stačí získat pouze jednou (takzvaný offline přístup) a může být i na nespecifikovanou dobu.

Povolení uživatel uděluje na speciální stránce Googlu, kam ho aplikace požadující oprávnění musí přeměrovat. Pokud uživatel povolení přijme, je přeměrován na určenou adresu s prvním klíčem v parametru URL adresy. Tento klíč poté aplikace může použít pro získání Autentizačního a Obnovovacího tokenu, spolu s informací o tom, kdy autentizační token vyprší. (zpravidla za jednu hodinu od obdržení). Autentizační token je poté používán v hlavičkách API dotazů pro konkrétní aplikaci Googlu. Pokud platnost Autentizačního tokenu vyprší, je potřeba získat nový pomocí Obnovovacího tokenu. Platnost Obnovovacího tokenu by v Implementaci Googlu neměl nikdy vypršet. Tento způsob autentizace je nazýván OAuth 2.0. [10] Aby mohla nějaká aplikace dostat povolení od uživatele, je nejdříve potřeba se zaregistrovat na Google API konzoli, založit projekt, použít přidělené aplikační tokeny a

Obrázek 1.5: Proces získávání přístupových tokenů [4]

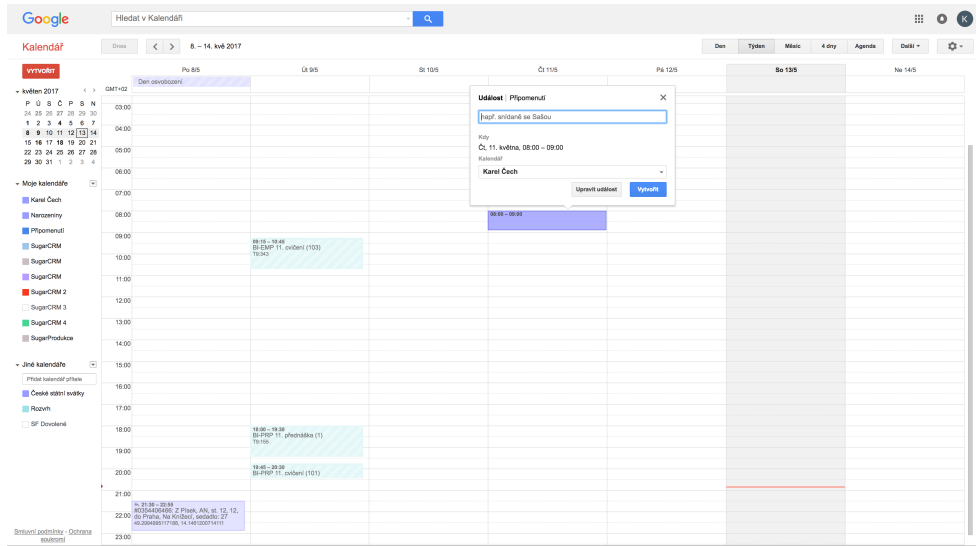


určit povolené URL, na které může být uživatel přesměrován i s prvním autorizačním klíčem ke jménu aplikace. Tento postup zvyšuje bezpečnost celého procesu, jelikož znemožňuje falešným aplikacím neoprávněně získávat práva k uživatelským datům.

1.2.2 Google API Client Library

Google poskytuje oficiální knihovnu [5] pro práci s jeho API pro většinu nejvíce používaných programovacích jazyků včetně PHP a Javascriptu. Tato knihovna může být velmi užitečná a odproštuje od „low-level“ programování API dotazů.

Obrázek 1.6: Google kalendář



1.2.3 Google kalendář

Jedna z mnoha aplikací firmy Google je i Google kalendář [9]. Uživatelé ji mohou používat pro přehledné spravování svých událostí, sdílení událostí mezi uživateli či skupinami, pro upozornění, zvaní na události a další.

Databáze a vnitřní logika je zcela oddělená od GUI a veškeré dotazy a operace jsou prováděny přes REST API, které je volně přístupné. To znamená, že kdokoliv může vytvořit svojí vlastní aplikaci, která bude plnohodnotně používat Google kalendář. Díky tomu můžeme kromě oficiální aplikace od Googlu nalézt i jiné aplikace používající Google kalendář, především na mobilní zařízení, či jako různá rozšíření do prohlížečů.

Jedna z funkcionalit je, že uživatel nemusí používat pouze jeden kalendář, ale může jich mít více najednou a poté je například prolínat na časové ose. Důvod pro použití více kalendářů může být například oddělení událostí barvami, či možnost sdílet pouze konkrétní kalendář, kdy uživatel, se kterým nějaký kalendář sdílíme vidí pouze události ze sdíleného kalendáře a z ostatních ne.

1.2.3.1 Calendar API

Calendar API je standardní REST API. Po získání přístupových tokenů je přes toto API možné plnohodnotně ovládat uživatele Google kalendář a je možné dělat úplně vše, co by uživatel mohl sám dělat přes oficiální webovou aplikaci od Googlu. Z komplikovanějších funkcionalit to může například být přesunutí události do jiného (i cizího) kalendáře, odesílání pozvánek do události jenom konkrétním osobám, nastavování práv a viditelností jednotlivých kalendářů a mnohé další.

Při získávání událostí z kalendáře lze použít takzvaný synchronizační token. Tento token API vrátí při každém natažení událostí. Pokud je token v dotazu použit, API vrátí pouze ty záznamy, které byly od vydání tokenu změněny, anebo založeny.

Pokud je záznamů velké množství, vrací API takzvaný stránkovací token a s ním pouze nějakou část záznamů. Pro získání všech záznamů je potřeba posílat dotaz se stránkovacím tokenem do té doby, než API žádný nevrátí.

Dokumentace Calendar API existuje na uživatelsky přívětivé formě i s vysvětlením postupů a příklady[6]. Existuje ale také kompaktnější dokumentace. [7]

1.2.3.2 Push notifikace

Kalendář umožňuje zaregistrovat požadavek pro takzvané Push notifikace ke konkrétnímu kalendáři. Push notifikace jsou notifikace, které jsou posílány Google kalendářem na zvolenou URL, v okamžik, kdy, je v uživatelské kalendáři něco změněno. Push notifikace už ale neposkytuje informaci o tom co bylo změněno.

Díky Push notifikacím však může server stahovat informace jen v moment, když bylo v kalendáři něco doopravdy změněno. Pokud by Push notifikace nechodily, aplikace by byla nucena se neustále periodicky dotazovat na události, aby případné změny zjistila. Push notifikace jdou pro jeden kalendář zaregistrovat maximálně po dobu jednoho týdne. Po uplynutí tohoto času je potřeba je zaregistrovat znova.

Podrobné informace o Push notifikacích lze nalézt v dokumentaci od Googlu. [8]

1.2.3.3 Události

Události v Google kalendáři také obsahují pole pro název, umístění, popis, datum a další. Lze k nim přikládat dokumenty a zvat jiné účastníky, a to pouze přes emailovou adresu, nemusí mít v Google kalendáři účet. Dále obsahuje pole pro nastavení opakovaných událostí a upozornění na událost. Také lze nastavit, kdo může událost upravovat. Většina polí a nastavení lze vidět na obrázku 1.7

Každá událost má svého organizátora, což je, ne příliš intuitivně, v podstatě kalendář, ve kterém se nachází.

Opakované schůzky jsou řešeny tím způsobem, že existuje jeden mateřský záznam, který v kalendáři není vůbec vidět. Tento záznam obsahuje informace o opakování a interpret kalendáře dokáže následně schůzky „vypočítat“ a zobrazit v kalendáři. Pokud je nějaký termín opakované události změněn, stává se z něj plnohodnotná událost a hromadné operace nad opakovanou schůzkou už ho neafektují.

S

Obrázek 1.7: Detail události v Google kalendáři

The screenshot shows the Google Calendar event creation interface. At the top, there's a search bar and navigation buttons like 'ULOŽIT', 'Zahodit změny', 'Smazat', and 'Další akce'. The event title is 'Snidane se Sašou'. The date and time are set to 11/5/2017 from 10:00 to 11:00. There are options for 'Celý den' and 'Opakovat...'. Below this, there are sections for 'Podrobnosti události' (Location, Video call, Calendar, Description, Attachment), 'Barva události' (Color selection), 'Oznámení' (Notification settings), and 'Zobrazit mě jako' (Visibility options). On the right, there's a 'Hosté' section for adding guests and their permissions.

1.3 Existující řešení

1.3.1 Collabspot

Collabspot [2] je externí aplikace ve formě doplňku do webového prohlížeče, která dokáže synchronizovat informace z emailu, adresáře a kalendáře mezi SugarCRM (ale i dalšími CRM systémy) a Googlem. Dokáže například v prohlížeči v bočním panelu ve webové aplikaci Gmail zobrazovat informace o kontaktech z příchozích e-mailů. K záznamům v SugarCRM přistupuje přes API, je tedy potřeba jí poskytnout přístupové údaje do systému. Propis změn ze SugarCRM do Google kalendáře je okamžitý, propis opačným směrem však trvá hodinu (v prémiové verzi 15 minut). Aplikace je také poměrně drahá.

1.3.2 RT GSync

RT GSync [15] je placené rozšíření do SugarCRM, které periodicky prohlíží uložené záznamy v Sugaru i v Googlu a synchronizuje je. Kromě kalendáře navíc synchronizuje i e-maily, kontakty a dokumenty. Jeho nevýhoda je, že

události jsou do google kalendáře propsány až za několik hodin a to samé na opačnou stranu.

1.3.3 iCalendar protokol

Kalendář SugarCRM lze importovat do jiných kalendářů pomocí iCalendar protoklu [3]. Nevýhoda tohoto řešení je, že funguje pouze jedním směrem (v ostatních kalendářích nelze editovat) a změny provedené v kalendáři Sugaru se v ostatních kalendářích neprojeví hned.

1.4 Požadavky

1.4.1 Funkční

- Synchronizace musí být obousměrná
- Synchronizace musí probíhat okamžitě (do pěti sekund)
- Synchronizovat se musí účastníci, stav účastníků, čas, místo, popis, vlastník a smazání schůzky.
- Uživatel si může vybrat, který Google kalendář bude se SugarCRM synchronizován.
- Synchronizují se i schůzky, na které je uživatel pozván.

1.4.2 Nefunkční

- Přihlašování musí být bezpečné
- Synchronizace musí být bezpečná
- Přihlašování musí být jednoduché a pro uživatele srozumitelné
- Uživatel by neměl pozorovat žádný výrazný rozdíl v prezentovaných informacích mezi kalendáři

Návrh

2.1 Shrnutí

Bude vyvinuto rozšíření pro aplikaci SugarCRM v jazycích PHP a Javascript, které bude pomocí Google Calendar API získávat události a další informace z Google kalendáře a provádět další operace, jako například změna kalendáře, ve které se událost nachází. Pro urychlení implementace bude použita knihovna Google Api Client Library pro jazyk PHP, která bude sloužit pro komunikaci s API Google kalendáře. Pro ukládání přihlašovacích tokenů a údajích o kalendářích pro jednotlivé uživatele bude v SugarCRM vytvořen nový modul.

2.2 Spuštění synchronizace

Na hlavní liště SugarCRM budou v záložce modulu kalendář vytvořena tlačítka pro zapnutí/vypnutí synchronizace s uživatelským Google kalendářem. Pokud se uživatel bude chtít přihlásit, použije toto tlačítko a bude přesměrován na ověřovací stránku Googlu. Po přidělení práv bude přesměrován zpátky na stránku SugarCRM, která pošle získaný klíč přes API na server se SugarCRM.² Server za pomoci klíče získá z Googlu autentizační tokeny a tyto tokeny uloží do databáze. Uživateli je poté zobrazeno okno, kde může vybrat jeden ze svých Google kalendářů nebo založit nový. Po vybrání kalendáře je tento údaj též uložen do databáze, jsou nastaveny Push notifikace pro uživatelův kalendář a propojení je tímto dokončeno.

Po spuštění synchronizace jsou uživateli staženy všechny schůzky z Google kalendáře do SugarCRM, maximálně však týden do minulosti. Ze SugarCRM do Google kalendáře nejsou nahrány žádné schůzky.

Pokud již měl uživatel dříve synchronizovaný nějaký jiný Google kalendář než vybraný, bude mu nabídnuta po spuštění synchronizace možnost přesu-

²Tento způsob je bezpečný, jelikož k použití API je oprávněn pouze přihlášený uživatel

nout události ze starého Google kalendáře do nového. Postup při přesouvání událostí je popsán níže.

2.3 Synchronizace událostí

2.3.1 Nová pole na událostech

Na událostech v SugarCRM budou vytvořena dvě nová pole. V jednom bude uloženo ID události v Google kalendáři a v druhém bude ID Google kalendáře, ve kterém by se měla nacházet.

2.3.2 Nastavení Push notifikací

Při spuštění synchronizace jsou zaregistrovány Push notifikace pro uživatelev Google kalendář. Tato registrace však po jednom týdnu vyprší a je potřeba ji znova zaregistrovat. Pro tuto potřebu bude v SugarCRM vytvořena naplánovaná úloha, která bude Push notifikace obnovovat. Dále je potřeba vytvořit API, na které bude Google kalendář Push notifikace odesílat. Přístup k tomuto API musí být bez klasické autorizace SugarCRM

Pokud API zachytí Push notifikaci a najde uživatele s Google kalendářem, který jí odeslal. Pro tohoto uživatele budou aktualizovány události z jeho Google kalendáře (viz. níže).

Pro jistotu bude vytvořena naplánovaná úloha, která bude periodicky spouštět aktualizaci pro všechny uživatele.

2.3.3 Ze SugarCRM do Google kalendáře

Událost bude ze SugarCRM propsána do Google kalendáře přiřazeného uživatele při založení nebo uložení, pokud má přiřazený uživatel aktivní synchronizaci. Pro tyto účely je vytvořen LogicHook, který za pomoci Google Api Client Library s uživatlovým přístupovým tokenem přes API při uložení vytvoří nebo změní událost v Google kalendáři. No události budou přidány (pokud tam ještě nejsou) všichni pozvaní, kteří mají v SugarCRM přiřazený email spolu se stavem jejich pozvání. Čas, název a popis události, bude zduplikován tak, jak je.

2.3.4 Z Google kalendáře do SugarCRM

Pokud přijde podnět pro stažení schůzek pro uživatele (od Push notifikace, naplánované úlohy, nebo po spuštění synchronizace) pro daného uživatele budou staženy veškeré schůzky od posledního týdne. Pokud bude u uživatele uloženy synchronizační token, bude použit. Po stažení schůzek se synchronizační token uživatele aktualizuje (pokud nový token existuje) ještě před samotným procesem párování a ukládání do databáze.

Při přijetí události je potřeba pomocí pole s ID události v Google kalendáři v databázi zjistit, jestli v SugarCRM existuje napárovaná událost. Pokud ne, je založena.

Je potřeba najít přiřazeného uživatele, jelikož mohou být staženy i schůzky, na které je aktuální uživatel pouze pozvaný. Nejdříve bude vyhledáváno podle ID přiřazeného Google kalendáře, poté podle e-mailu Google účtu ke Google kalendáři a poté podle primárního e-mailu v SugarCRM. Pokud bude podle těchto podmínek nějaký uživatel nalezen, bude nastaven jako přiřazený uživatel. Pokud žádný uživatel nalezen nebude, jako přiřazený uživatel bude nastaven Administrátor. Pokud by měl být přiřazený uživatel administrátor, nebude se vyskytovat v pozvaných.

Stejným způsobem jako výše budou heldáni pozvaní. Pokud žádný nebude nalezen, v SugarCRM nebude vidět. Pozvaní mohou být i z modulů Leady nebo Kontakty.

Ostatní relevantní atributy události se zkopírují.

Při implementaci bude potřeba dát pozor na to, že když pokud se události v Google kalendáři bude účastnit více uživatelů ze SugarCRM se synchronizovanými Google kalendáři, ta samá událost se při synchronizaci stáhne pro každého z nich.

2.3.5 Opakované události

Opakování událostí nebude přeneseno vzhledem k rozdílným řešením v obou systémech. V jednom systému budou jednotlivé termíny opakované události druhého považovány za normální událost. Proto pokud bude chtít uživatel použít nějakou funkcionalitu týkající se hromadných událostí (např. smazání), bude muset tuto operaci provést v systému, kde opakovanou událost založil.

Pokud je založena opakovaná událost v Google kalendáři, standardně je při stažení vrácena pouze jedna mateřská událost. Je proto potřeba události stahovat všechny, aby byla správně spárována ID událostí.³

2.4 Změny vlastníků událostí

Při změně organizátora události v Google kalendáři bude při synchronizaci schůzka stažena s novým organizátorem. Bude tedy potřeba při každém stažení události kontrolovat organizátora a případně změnit vlastníka schůzky v SugarCRM.

Pokud bude změněn vlastník v SugarCRM, bude schůzka přesunuta z Google kalendáře původního vlastníka do Google kalendáře nového vlastníka.

³ Teoreticky by nejspíše bylo možné stahovat z Google kalendáře pouze mateřské události, jelikož z pokusů je patrné, že ID termínu opakované události lze složit jako ID mateřské události s časovou značkou termínu, takže teoreticky by bylo možné ID vygenerovat se značkou termínů. V dokumentaci Google kalendáře se ale o ničem takovém nepíše, proto na to nebudeme spoléhat.

Pokud nový vlastník nemá synchronizaci s Google kalendářem, SugarCRM nedovolí vlastníka změnit.⁴ pro přesun mezi Google kalendáři bude potřeba, aby starý vlastník měl práva pro zapisování do Google kalendáře nového vlastníka.

2.5 Změna Google kalendáře

Uživatel si bude moct v modálním okně změnit Google kalendář, se kterým je SugarCRM synchronizováno. Změnit bude moct pouze na kalendář, který je pod jeho Google účtem. Pokud se uživatel rozhodne kalendář změnit, bude mu nabídnuta možnost přesunout jeho události ze starého Google kalendáře do nového. Pokud bude souhlasit, všechny schůzky s datem konání před týdnem a novější budou přesunuty do nově zvoleného Google kalendáře.

2.6 Nový modul v SugarCRM

Pro uchování informací o synchronizaci a jejím nastavení pro jednotlivé uživatele bude vytvořen nový skrytý modul.⁵ Pro každého uživatele bude existovat právě jeden záznam v tomto modulu, kde ID tohoto záznamu bude stejné jako ID uživatele. Budou ukládány tyto informace:

- Autentizační token
- Obnovovací token
- Doba platnosti Autentizačního tokenu
- Datum získání Autentizačního tokenu
- Informace o tom, zda si uživatel přeje mít spuštěnou synchronizaci
- E-mail uživatelovo Google účtu
- ID uživatelovo používaného kalendáře
- Synchronizační token

⁴Změna nebude povolena z toho důvodu, že by poté vznikla situace, kdy by původní vlastník mohl stále manipulovat s událostí ve svém Google kalendáři a tím manipulovat událostí někoho jiného v SugarCRM nebo (pokud by bylo zrušeno spárování události) by vznikla duplicita, jelikož by pro něj událost byla znova založena. Smazání události v Google kalendáři také není možné, jelikož na událost mohou být pozváni nějací další uživatelé.

⁵Normálně by tyto informace byly umístěny přímo na záznam uživatele. SugarCRM však při každém požadavku kontroluje hash uživatelovo záznamu. Pokud hash mezi prohlížečem a serverem neseďí, požadavek selže a budou staženy nové údaje o uživateli. Jelikož se například Synchronizační token mění při každém stažení změn z Google kalendáře, k rozhození hashe by docházelo velmi často, tím pádem by bylo velmi pravděpodobné, že budou ztraceny nové informace o záznamu, který by se zrovna uživatel pokoušel uložit, když by hash neseďel.

- ID kanálu pro Push notifikace
- ID záznamu Push notifikace
- Uživatelský předchozí Obnovovací token
- ID uživatelského předchozího kalendáře
- E-mail uživatelského předchozího Google účtu

Realizace

3.1 Vývoj

Synchronizace SugarCRM s Google kalendářem byla vyvíjena na tři etapy. V první byla zprovozněna autentifikace a propis událostí ze SugarCRM do Google kalendáře. V druhé byla vylepšena autentifikace a spustila synchronizace směrem z Googlu do SugarCRM. Ve třetí etapě byly zprovozněny Push notifikace, které velmi šetří výpočetní čas.

Implementace nemohla probíhat na lokálním počítači z důvodu absence veřejné IP adresy (respektive domény) lokálního serveru, na kterou by Google přesměroval uživatele po povolení přístupu a na kterou by posílal Push notifikace. Proto implementace probíhala na vzdáleném vývojovém serveru, na který bylo přistupováno přes SSH.

Propojení bylo vyvíjeno pro konkrétní instanci SugarCRM, kterou v době vývoje používalo přibližně 200 uživatelů.

Při vývoji byly používány postupy a standardy popsány v knížce Learning PHP, MySQL & JavaScript.[13]

3.1.1 Vývojové prostředí

Veškerý vývoj probíhal v programu PhpStorm 2016.3.2 od firmy JetBrains, který je momentálně jeden z nejlepších nástrojů pro vývoj v PHP. V programu PhpStorm lze zkopírovat a synchronizovat soubory ze vzdáleného serveru pomocí SSH do lokálního prostředí. Technicky se tedy pracuje na lokálních souborech, které mají stejnou strukturu jako na serveru, ale po uložení jsou přes SSH automaticky také nahrány na vzdálený server. Zdánlivě to tedy vypadá, jako kdyby se pracovalo přímo se soubory na vzdáleném serveru. Nevýhoda je, že pokud někdo jiný na serveru změní nějaký soubor, tak tento soubor je přepsán, pokud je poté uložen v programu PhpStorm bez předchozí synchronizace souborů. Více o programu PhpStorm se lze dozvědět v knížce PhpStorm Cookbook. [12]

Instance SugarCRM, na které byla funkcionální implementována byla již velmi upravená a zároveň s vývojem propojení bylo vyvíjeno několik dalších funkcionalit, takže bylo potřeba dávat velký pozor, abychom si s jinými vývojáři nenarušovali práci.

3.1.2 Verzování

Verzování funkcionalit pro SugarCRM je poměrně komplikované, jelikož velká část úprav je spojena i se změnami v databázi. Proto pokud by byl repozitář přímo z instance SugarCRM a větev byla změněna, aplikace by pravděpodobně přestala fungovat, jelikož by data v databázi neodpovídala aktuálním souborům. Proto vznikla (ne v rámci této práce) utilita nazývaná „Deployer“, která dokáže z instance vykopírovat relevantní soubory a zároveň vyexportovat potřebné data z databáze. Zároveň tato utilita data dokáže nahrát zpátky do instance SugarCRM.

A právě nad těmito vykopírovanými soubory a vyexportovanými soubory z databáze má smysl vést repozitář. Když je změněna větev, Deployer zkopíruje soubory z repozitáře zpátky do instance SugarCRM a s pomocí vyexportovaných databázových souborů dokáže databázi dovést do požadovaného stavu. Velká nevýhoda spočívá v tom, že pokud nejsou pomocí Deployeru vyexportovány a zaverzovány zrovna upravené soubory, tak nahrání směrem z repozitáře do instance SugarCRM tyto úpravy nenávratně smaže (pokud nebyly zazálohovány). Při práci s Deployerem je tedy potřeba si dávat velký pozor, aby se toto nestalo.

Pro verzování byl použit program GIT. V rámci jednotlivých etap bylo založeno několik vývojových větví z hlavního repozitáře instance SugarCRM, na které byly úpravy prováděny.

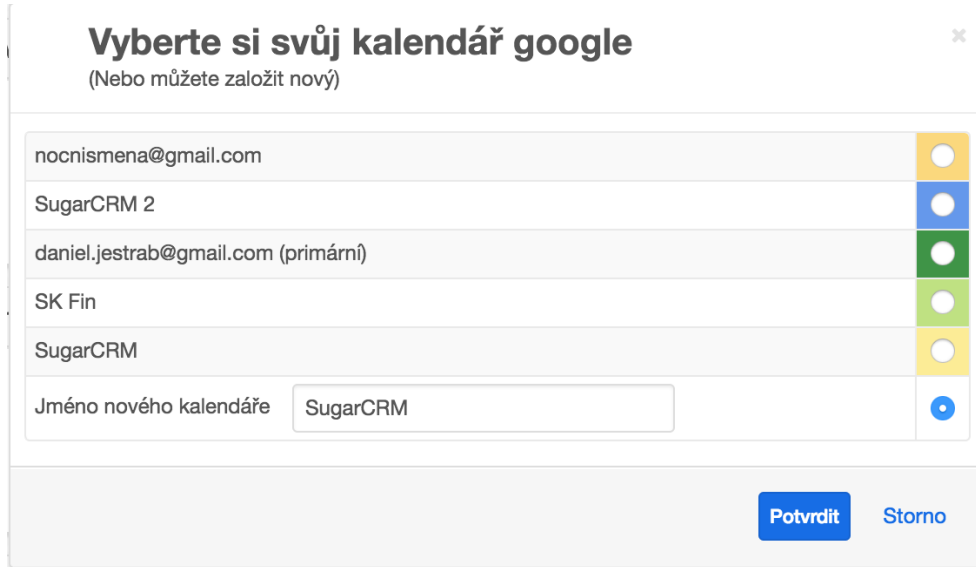
3.1.3 Dokumentace

Vznikla programátorská a uživatelská příručka, obě dostupné online po přihlášení. Programátorská příručka byla napsána v angličtině, uživatelská v češtině. Třídy a metody jsou standardně zdokumentovány v kódu.

3.1.4 Testování

V průběhu implementace byl kód podrobován kontrole dalšími členy vývojového týmu ve webové aplikaci Phabricator, kde jsou přehledně vidět rozdíly mezi jednotlivými verzemi repozitáře. Další testování mělo na starosti testovací oddělení firmy SugarFactory a není součástí této práce.

Obrázek 3.1: Modální okno pro výběr kalendáře



3.2 Řešení

Vzniklo několik tříd, které mají na starost jednotlivé činnosti okolo synchronizace. Také vzniklo nové API, které se stará o operace spojené se synchronizací mezi prohlížečem a serverem. Dále vznikl nový pohled, který se ve formě modálního okna stará o výběr nebo založení uživatelského Google kalendáře. Dále vznikl nový modul, ve kterém jsou ukládány informace o uživatelském propojení s jeho Google kalendářem.

Všechny třídy mají předponu SF, která je používána pro označení produktu společnosti SugarFactory.

3.2.1 Třídní diagram

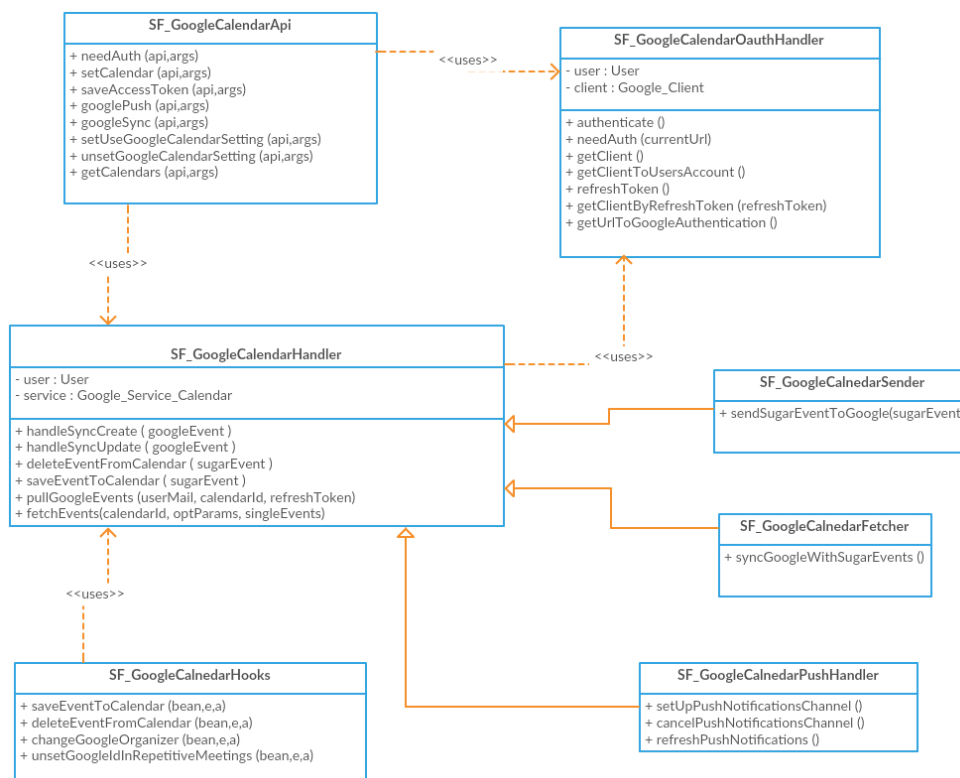
V třídním diagramu na obrázku 3.2 jsou zobrazené nové PHP třídy a vztahy mezi nimi ve standardu UML, které vznikly za účelem synchronizace s Google kalendářem. U tříd jsou pro přehlednost zobrazeny pouze důležité metody a proměnné.

3.3 Nedostatky

Pokud v SugarCRM nějaký uživatel nemá propojený Google kalendář a pozve na svou událost někoho kdo ano, tak se pozvanému schůzka nezobrazí v Google kalendáři. Tato situace nastane z toho důvodu, že není žádný účet v Google kalendáři, kde by mohla být vytvořena. Řešením by mohlo být založení účtu

3. REALIZACE

Obrázek 3.2: Třídní diagram



pro konkrétní Instanci SugarCRM v Google kalendáři, kde by tyto schůzky byly vytvářeny.

Pokud je potřeba přesunout schůzku mezi Google kalendáři, je potřeba dočasně nastavit přístupová práva. Google bohužel po změně práv odešle notifikaci o tom, že uživatel má práva upravovat Google kalendář druhého uživatele (to ale není pravda, protože jsou práva poté zase okamžitě zrušena). Odeslání této notifikace nejde z venku zabránit.

Při zpětném ohlédnutí by dávalo větší smysl třídu SF_GoogleCalendarHandler rozpadnout na menší třídy. První menší třída by se měla starat o databázové operace, další dvě jako obal pro funkce měnící objekty událostí SugarCRM a Google kalendáře a čtvrtá by obsahovala zbytek funkcí a zároveň by z ní dědily třídy, které dědí z původní třídy teď.

3.4 Návrhy na vylepšení

Když v Google kalendáři někdo někoho pozve, pozvanému se událost objeví v jeho primárním kalendáři. Pokud má však se SugarCRM synchronizovaný

jiný kalendář a chce, aby se mu schůzka, na kterou je pozván zobrazila v SugarCRM, musí manuálně vytvořit kopii události v synchronizovaném Google kalendáři. Do budoucna by tedy například bylo možné procházet i primární kalendář a stahovat z něj události, na které je uživatel pozván.

Ve zpětné vazbě uživatelů bylo velmi často slyšet, že by si přáli synchronizovat více než jeden Google kalendář. Pro tuto funkcionalitu už by nejspíše stálo za to založit plnohodnotný modul pro Google kalendáře, které by byly přiřazeny jednotlivým událostem v SugarCRM.

Závěr

Propojení SugarCRM s Google kalendářem bylo úspěšně naimplementováno a nahráno na produkční server, kde tuto funkcionalitu bez větších problémů používá přes 200 uživatelů. Všechny funkční i nefunkční požadavky z analýzy byly splněny, kromě požadavku na synchronizaci všech schůzek, na které je uživatel pozván, z důvodů popsaných výše.

Podle výkazu odpracovaných hodin bylo na implementaci stáveno přibližně 250 hodin. Návrh a implementace řešení byly poměrně přímočaré a většina základních funkcionalit byla hrubým odhadem naimplementována během 30ti hodin. Většina času byla tedy strávena hledáním a odstraňováním chyb, které byly většinou způsobeny špatným pochopením, či nestandardním fungováním SugarCRM, a dopilováváním detailů, jako jsou třeba opakované, či celodenní schůzky, odesílání notifikací nebo změny vlastníků. Rozsah implementace nelze přesně spočítat, ale dá se přibližně stanovit na něco kolem dvou tisíc řádek funkčního kódu (bez metadat).

S Google Calendar API se pracovalo velmi příjemně, vše dává smysl, je řádně zdokumentované a především se API i podle dokumentace chová. To samé se bohužel nedá říct o SugarCRM, kde vnitřní funkcionality nejsou příliš dobře zdokumentovány a většinou je potřeba vše pochopit přímo z kódu.

Celkový výstup také velmi negativně ovlivňoval fakt, že se pracovalo pod velkým časovým tlakem ze strany zákazníka, takže druhá a největší etapa (synchronizace z Google kalendáře do SugarCRM) byla nasazena do produkčního prostředí po pouhých dvou týdnech vývoje.

Řešení bylo psané s ohledem na univerzálnost, neměl by tedy být problém po drobných úpravách propojení zprovoznit i na jiné instanci SugarCRM.

Literatura

- [1] Čápka, D.: *www.itnetwork.cz* [online]. 2016 [cit. 14.05.2017]. Dostupné z: <https://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor>
- [2] Collabspot: *www.collabspot.com* [online]. Collabspot, 2017 [cit. 14.05.2017]. Dostupné z: <https://www.collabspot.com/gsuite/>
- [3] Desruisseaux, B.: Internet Calendaring and Scheduling Core Object Specification (iCalendar). RFC 5545, Oracle, September 2009. Dostupné z: <https://tools.ietf.org/html/rfc5545>
- [4] Google: *developers.google.com* [online]. Google, Inc, 2017, 13.04.2017 [cit. 13.04.2017]. Dostupné z: <https://developers.google.com/accounts/images/webflow.png>
- [5] Google: *developers.google.com* [online]. Google, Inc., 2017 [cit. 04.05.2017]. Dostupné z: <https://developers.google.com/api-client-library/>
- [6] Google: *developers.google.com* [online]. Google, Inc, 2017, 13.04.2017 [cit. 13.04.2017]. Dostupné z: <https://developers.google.com/google-apps/calendar/overview>
- [7] Google: *developers.google.com* [online]. Google, Inc, 2017, 13.04.2017 [cit. 13.04.2017]. Dostupné z: <https://developers.google.com/apis-explorer/#p/calendar/v3/>
- [8] Google: *developers.google.com* [online]. Google, Inc, 2017, 13.04.2017 [cit. 13.04.2017]. Dostupné z: <https://developers.google.com/google-apps/calendar/v3/push>
- [9] Google, I.: Google Calendar Help Center. *support.google.com* [online]. Google, Inc., 2017 [cit. 04.05.2017]. Dostupné z: <https://support.google.com/calendar>

- [10] Hardt, D.: The OAuth 2.0 Authorization Framework. RFC 6749, Internet Engineering Task Force, October 2012. Dostupné z: <https://tools.ietf.org/html/rfc6749>
- [11] Levy, S.: *In the Plex: How Google Thinks, Works, and Shapes Our Lives*. Simon & Schuster, 2011.
- [12] Mukund Chaudhary, A. K.: *PhpStorm Cookbook*. Packt publishing, December 2014.
- [13] Nixon, R.: *Learning PHP, MySQL & JavaScript, 4th Edition With jQuery, CSS & HTML5*. O'Reilly Media, čtvrté vydání, 2014.
- [14] Osmani, A.: *Developing Backbone.js Applications Building Better JavaScript Applications*. O'Reilly Media, 2013.
- [15] Rolustech: www.rolustech.com [online]. Rolustech, 2017 [cit. 14.05.2017]. Dostupné z: <https://www.rolustech.com/rtgsync/>
- [16] Solutionsmetrix: solutionsmetrix.com [online]. Solutionsmetrix, 2017 [cit. 04.05.2017]. Dostupné z: <http://solutionsmetrix.com/wp-content/uploads/2016/01/sugarcrm-77-dashboard.png>
- [17] SugarCRM: www.sugarcrm.com [online]. SugarCRM, 2017 [cit. 14.05.2017]. Dostupné z: <https://www.sugarcrm.com/>
- [18] SugarCRM: support.sugarcrm.com [online]. SugarCRM, 2017 [cit. 14.05.2017]. Dostupné z: <http://support.sugarcrm.com/files/59b8a364-04d8-34d9-432e-56ec677c53fd.png>
- [19] SugarCRM, Inc.: *Sugar Developer Guide 7.6*. První vydání, Semptember 2015.

Vysvětlení odborných pojmů

Frontend Část aplikace, se kterou přímo pracuje uživatel, například ve webovém prohlížeči

Backend Část aplikace, která vykonává logiku a uživatel s ní nepřichází do kontaktu. Většinou běží na vzdáleném serveru.

MVC architektura Softwarová architektura používaná moderními aplikacemi. Rozděluje systém na tři víceméně nezávislé komponenty - datovou část, uživatelské rozhraní a řídicí logiku. [1]

API Rozhraní aplikace, pomocí kterého s ní lze komunikovat anebo ji ovládat.

Open-source Open-source aplikace mají přístupné zdrojové kódy.

hash Výstup z hashovací funkce

Hashovací funkce Funkce, která z libovolně velkého vstupu deterministicky vytvoří krátký výstup.

Seznam použitých zkratk

- REST** Representational State Transfer
- API** Application Programming Interface
- GUI** Graphical user interface
- CRM** Customer relationships management
- XML** Extensible markup language
- MVC** Model-view-controller
- SQL** Structured Query Language
- CRUD** Create, retrieve, update, delete

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
├─ BP_Cech_Karel_2017.tex	zdrojová forma práce ve formátu L ^A T _E X
├─ csn690.bst.....	funkce upravující citace
├─ FITthesis.cls.....	šablona bakalářské práce
├─ bibfile.bib.....	soubor s literaturou
├─ assignement.pdf.....	zadání práce
├─ images.....	adresář obsahující použité obrázky
text	
├─ BP_Cech_Karel_2017.pdf.....	text práce ve formátu PDF
├─ sugar_developer_guide_7.6.pdf ...	vývojářský manuál pro SugarCRM