

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ON-LINE DASHBOARD PRO STARTUPY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RÓBERT PECSÉRKE

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ON-LINE DASHBOARD PRO STARTUPY

ON-LINE DASHBOARD FOR STARTUPS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

RÓBERT PECSÉRKE

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÖKE, Ph.D.

BRNO 2013

Abstrakt

Cílem této bakalářské práce je analyzovat, navrhnout a vytvořit webovou aplikaci, která umožní přihlášeným uživatelům vytvářet si emailové účty s náhodným názvem účtu - Dropbox. Na tyto pak uživatelé mohou přesměrovat emaily, které generují další služby, které využívají, případně posílat vlastní emaily. Aplikace vyhledá specifická data v těchto emailích a umožní vizualizovat jejich změnu v průběhu času. V práci je zároveň ověřený zájem uživatelů o takový typ služby.

Abstract

The purpose of this bachelor's thesis is to analyze, design and develop a web application that allows registered users to create an email account with a random name - dropbox. Users can either redirect emails generated by other services or send their emails to the dropboxes. The application searches for specific data in these emails and to visualize their change over time. This thesis also verifies the interest of users of this type of service.

Klíčová slova

Webová služba, elektronická pošta, dropbox, Minimum Viable Product

Keywords

Web service, email, dropbox, Minimum Viable Product

Citace

Róbert Pecsérke: On-line dashboard pro startupy, bakalářská práce, Brno, FIT VUT v Brně, 2013

On-line dashboard pro startupy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szökeho, Ph.D.

.....

Róbert Pecsérke

12. května 2013

Poděkování

Ďakujem môjmu školiťovi, Ing. Igorovi Szökemu, Ph.D. za cenné rady, odborné usmer-
nenia, podporu a najmä za ľudský prístup a pomoc, ktorú mi poskytol pri vypracovaní
bakalárskej práce

© Róbert Pecsérke, 2013.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informa-
čních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění
autorem je nezákonné, s výjimkou zákonem definovaných případů.*

Obsah

1	Úvod	3
2	Startupy	4
2.1	Minimum Viable Product	4
2.2	Tvorba obchodného modelu	5
3	Analýza a návrh	7
3.1	Fáza hľadania zákazníkov	7
3.2	Popis platformy	7
3.3	Dropbox	7
4	Realizácia	9
4.1	Nasadenie a správa aplikácie	9
4.2	Poštový server	10
4.2.1	Poštový užívateľský agent	10
4.2.2	Poštový prenosový agent	10
4.2.3	Agent doručenia správy	10
4.2.4	E-mailové protokoly	11
4.2.5	Použité technológie	11
4.2.6	Konfigurácia	11
4.2.7	Filtrovanie obsahu a spam	12
4.3	Webový server	12
4.3.1	HTTP	12
4.3.2	URI	12
4.3.3	MVC	12
4.3.4	Predný radič	13
4.3.5	Architektúra webovej aplikácie	13
4.4	Databázový server	15
4.4.1	Tabuľky existujúce pri nasadení	15
4.4.2	Tabuľky s dátami dropboxov	16
4.4.3	Bezpečnosť	17
4.5	Parovací skript	17
4.5.1	App.py	17
4.5.2	Parser.py	18
4.5.3	Mail.py	18
4.5.4	SimpleParser.py	18
4.5.5	AddThisParser.py	19
4.5.6	FacebookParser.py	19

5	Vývoj riadený užívateľmi	20
6	Záver	24

Kapitola 1

Úvod

Malé a stredne veľké firmy reprezentujú 99% všetkých spoločností v EÚ. Sú najväčší sektor ekonomie EÚ zamestnávajúci približne 75 miliónov ľudí. Sú zodpovedné za vznik jedného z každých dvoch nových pracovných miest a vyprodukujú spolu viac ako polovicu únijskeho HDP. Súčasný trendy ukazujú, že v najbližšej budúcnosti môžeme očakávať nárast vzniku startupov. Ich zakladatelia musia riešiť množstvo problémov, okrem iného sú zaplavení množstvom informácií, ktoré s prevádzkou startupu súvisia.

Táto práca si kladie za cieľ poskytnúť zakladateľom nástroj na správu a vizualizáciu týchto informácií vo forme grafov. Aby sa maximálne šetril čas užívateľov, informácie sú predávané ľahko automatizovateľným spôsobom - zasielaním emailu. Ďalším cieľom je rýchle nasadenie aplikácie do reálnej prevádzky a rýchly vývoj na základe požiadaviek užívateľov.

Mojou motiváciou pri zápise tejto práce bolo to, že považujem oblasť webových aplikácií za veľmi perspektívnu, a časom by som rád pracoval v tejto oblasti, no doposiaľ som nemal žiadne skúsenosti s tvorbou webu. Táto práca bola skvelou príležitosťou nazbierať potrebné skúsenosti a vytvoriť skutočný produkt, ktorý budú používať užívatelia s reálnymi potrebami.

V kapitole 2 je popísaná stručná história startupov ako začínajúcich firiem a postupná zmena prístupu ich zakladateľov od kopírovania veľkých firiem až po súčasný návrh Minimum Viable Product-u a vývoj riadený užívateľmi.

V kapitole 3 sú vytvorené hypotézy o zákazníkoch a o produkte. Ďalej je tu analyzovaný problém webovej aplikácie schopnej prijímať dáta prostredníctvom emailov. Aplikácia je rozdelená na webovú časť, poštovú časť a parsovaciu časť.

Kapitola 4 hovorí o riešení jednotlivých podproblémov z predchádzajúcej kapitoly systémom zhora nadol a spôsobom akým sú realizované.

V kapitole 5 sú popísané reakcie užívateľov a úpravy aplikácie podľa ich správania a želaní.

V závere sú zhrnuté dosiahnuté výsledky a načrtnutý možný ďalší vývoj v budúcnosti.

Kapitola 2

Startupy

Táto kapitola čerpá z [2]. V roku 1602 vydala Nizozemská východoindická spoločnosť, všeobecne považovaná za prvú modernú spoločnosť, prvé akcie. Nasledujúcich 300 rokov spoločnosti vznikali a rástli bez formálne školených vedúcich pracovníkov. Rastúca komplexita moderných spoločností začiatkom dvadsiateho storočia vytvorila rastúci dopyt po vedúcich pracovníkoch, ktorí boli školení na správu veľkých spoločností. V roku 1908 Harvard ponúkol po prvýkrát v histórii možnosť získať titul Master of business administration (MBA), aby tento dopyt uspokojil. Osnovy MBA štandardizovali a kodifikovali základné prvky, ktoré vedenie modernej spoločnosti muselo poznať: správa nákladov, produktový manažment, financie, strategický manažment, strojnictvo, správa ľudských zdrojov, operačný manažment.

V polovici dvadsiateho storočia prichádza moderná forma startupového priemyslu. Tento priemysel, spojenie začínajúcich podnikateľov a rizikového kapitálu, od tej doby prudko rastie. Po vyše polstoročí zostáva založenie startupu výzvou. Zakladatelia mali neustále problémy s aplikovaním obchodných nástrojov veľkých firiem na tie začínajúce. Tradičné MBA osnovy pre zabehnuté veľké spoločnosti ako sú IBM, GM alebo Boeing nefungujú pre startupy.

Dnes vieme, že startupy nie sú menšie verzie veľkých spoločností. Veľké spoločnosti majú obchodný model, v ktorom sú zákazníci, ich problémy a požadované vlastnosti produktu známe. Narozdiel od nich, startupy tieto informácie zatiaľ iba hľadajú. Toto hľadanie ale vyžaduje odlišný prístup, schopnosti a nástroje v snahe minimalizovať riziká a maximalizovať šancu na úspech.

Začiatkom dvadsiateho storočia začali drobní podnikatelia hľadať a vytvárať svoje vlastné obchodné nástroje. Výsledkom je rýchlo rastúca oblasť, ktorá zahŕňa agilný vývoj, business model design, rôzne nástroje podporujúce kreativitu a inovácie, Lean Startup a iné.

Startupy sú založené na hypotézach. Prichádzajú s projektom, ktorý je iba víziou ich zakladateľa a potrebuje si nájsť svojich zákazníkov. Cieľom hľadania zákazníkov je premeniť zakladateľovu počiatočnú hypotézu o trhu a zákazníkoch na fakty.

2.1 Minimum Viable Product

Minimum Viable Product (MVP) - minimálny uskutočniteľný produkt je produkt s minimálnym množstvom funkcií, ktorý ale zároveň rieši nejaký problém alebo potrebu zákazníka.

Spoločnosť nevie, kto sú jej počiatoční zákazníci a aké majú potreby. Jednou možnosťou

je vytvoriť produkt, ktorý bude mať hneď od začiatku veľké množstvo funkcií. Ukazuje sa však, že toto riešenie je mrhanie časom a peniazmi, pretože užívatelia často nepoužívajú, nechcú a ani nepotrebujú funkčnosť, ktorá vznikne bez ich vstupu.

Produktívnejšie riešenie je vyvinúť jadro produktu s jeho kľúčovými funkciami (inkrementálne a iteratívne s využitím agilných metodológií) a toto je MVP.

MVP uľahčuje testovanie, či skutočne rozumieme zákazníkovi problému a či ho nami ponúknuté riešenie podnieti používať produkt, prípadne si ho kúpiť.

Prvá verzia produktu nie je navrhnutá pre bežného zákazníka. Startupy si nemôžu dovoliť vytvárať produkt so všetkými funkciami, ktoré bežný zákazník požaduje hneď od začiatku jeho nasadenia. Namiesto toho sa musia zameriavať na malú skupinu počiatočných zákazníkov, ktorí poskytnú nevyhnutnú spätnú väzbu, ktorá umožňuje pridávanie novej funkcionality do produktu.

Posun v myslení smerom k inkrementálnemu a iteratívne vývoju je mimoriadne dôležitý. Technici majú tendencie vytvárať produkt, ktorý je technickou výzvou. Žiaľ takéto produkty sú často zbytočne veľké a zložité, s funkcionalitou po ktorej zákazník netuži. MVP im pomáha sústrediť výlučne na tie najdôležitejšie časti.

2.2 Tvorba obchodného modelu

Každý zakladateľ startupu má svoju víziu a sadu hypotéz o obchodnom modeli - o trhu, zákazníkoch a komunikačných kanáloch, o stanovenej cene. Tieto hypotézy je nevyhnutné premeniť na fakty. Tradičný model uvedenia nového produktu ponúka užívateľskú odozvu až s príchodom beta verzie.

Pre startupy vhodnejším prístupom je Customer Development model. Tento model delí všetky aktivity spojené s užívateľmi na samostatné procesy. Prvé dva kroky zahŕňajú hľadanie obchodného modelu, rozvíjajú a testujú a tretí a štvrtý krok tento obchodný model vykonávajú. Jednotlivé kroky sú:

1. Hľadanie zákazníkov - proces premeny zakladateľovej vízie na sériu hypotéz o jednotlivých častiach obchodného modelu a vytvorenie sady experimentov na testovanie týchto hypotéz.
2. Validácia zákazníkov - dokazuje, či použitý obchodný model dokáže vybudovať dostatočne veľkú základňu užívateľov potrebnú pre vznik fungujúcej spoločnosti.
3. Vytváranie zákazníkov - stavia na počiatočných finančných úspechoch spoločnosti. Produkt si už našiel svoj trh. V tomto bode startup rýchlo rastie v sprievode masívnych investícií do marketingu.
4. Budovanie spoločnosti - startup našiel svoj škálovateľný obchodný model. Od tohto okamihu to už nie je organizácia zameraná na hľadanie známa ako startup, ale stáva sa spoločnosťou.

Na ujasnenie obchodného modelu sa pri hľadaní zákazníkov používa business model canvas od Alexandra Osterwaldera. BMC obsahuje deväť priečinkov zobrazujúcich detaily o firemnom produkte, zákazníkoch, komunikačných kanáloch, vytváraní dopytu, modely výnosov, partnerov, aktivity a štruktúru nákladov. V tejto fáze by mali byť známe hypotézy:

- Veľkosť trhu

- Hodnotová propozícia, prvá časť - produkt, jeho výhody, MVP
- Zákaznícke segmenty - kto je zákazník, a aké jeho problémy produkt rieši
- Kanály - akým spôsobom bude produkt distribuovaný a predávaný
- Vzťahy so zákazníkmi - akým spôsobom bude vytváraná ponuka
- Hodnotová propozícia, druhá časť - hypotéza o type trhu a konkurenčná diferenciácia
- Kľúčové zdroje - dodávatelia, komodity a iné základné elementy podnikania
- Kľúčoví partneri - iné spoločnosti nutné k úspechu podnikania
- Zdroje príjmov - zdroje zisku a ich veľkosť

Kapitola 3

Analýza a návrh

Cieľom práce je vytvoriť webovú aplikáciu, ktorá umožní prihláseným užívateľom vytvárať si emailové účty s náhodným názvom účtu - dropboxy. Na tieto potom môžu užívatelia presmerovať emaily, ktoré generujú iné služby, ktoré využívajú, prípadne poslať vlastné emaily. Aplikácia potom vyhľadá špecifické dáta v týchto emailoch a uloží si ich do databázy. Prihláseným užívateľom potom umožní vizualizovať tieto dáta.

3.1 Fáza hľadania zákazníkov

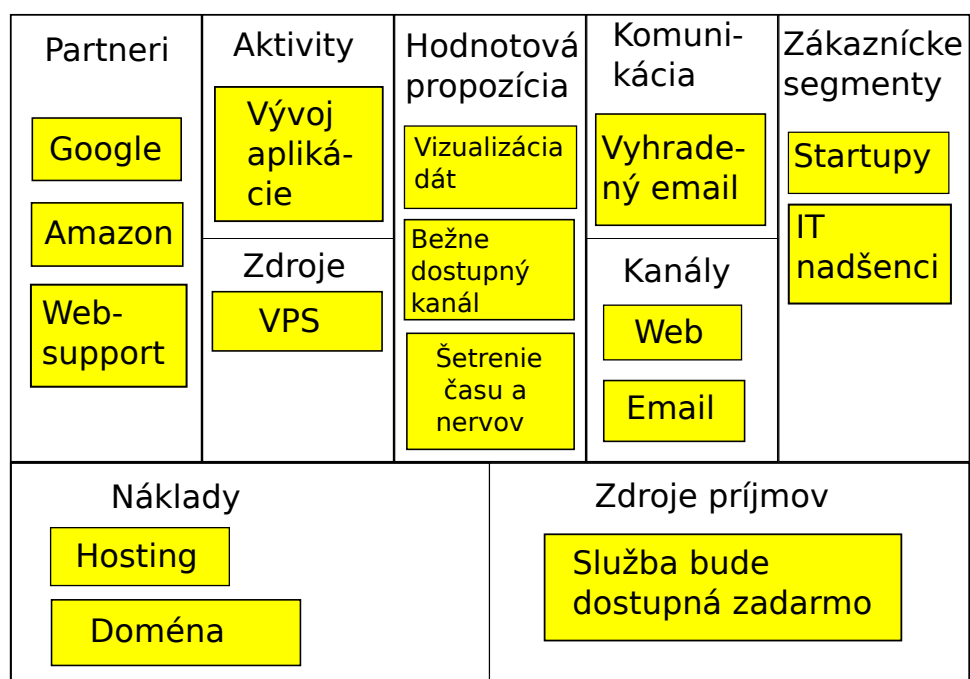
Názov aplikácie bude Dashboard for startups. Kľúčovými partnermi budú spoločnosti Google, Amazon a Websupport. Aplikácia je cieľená na startupy, freelancerov a IT nadšencov. Predpokladám, že bežní užívatelia by funkčnosť aplikácie nevyužili, pretože jej použitie vyžaduje určitý stupeň schopností a znalostí z oboru informačných technológií. Aplikácia má zjednodušiť život cieľovej skupine užívateľov, ktorí využívajú iné služby, ktoré generujú pravidelné správy vo forme emailov. Z týchto emailov prečíta dáta a dátum, ku ktorému patria a vykreslí ich zmenu v čase do grafu. Užívatelia sa môžu venovať iným, pre nich dôležitejším, činnostiam. Pre potreby komunikácie so zákazníkmi bude zriadená špeciálna emailová adresa info@dbfs.eu, kam môžu poslať svoje nápady a pripomienky. Business model canvas je zobrazený na obrázku 3.1.

3.2 Popis platformy

Na vykresľovanie grafov budú použité Google Chart Tools a na vyhodnocovanie užívateľskej aktivity nástroj Google Analytics. U firmy Amazon je prenajatý virtuálny server s operačným systémom Ubuntu Server. U firmy Websupport sú objednané domény pre webovú aplikáciu a poštový server - dbfs.eu.

3.3 Dropbox

Dropbox je základná jednotka celej aplikácie. Dropbox je poštová schránka s náhodným identifikátorom. Z prijatých emailov načíta údaje a ukladá si ich do databázy vo forme premenných, ktoré môžu byť neskôr vykreslené vo forme grafu. Každý dropbox má jednu x-ovú premennú a ľubovoľný počet y-ových premenných.

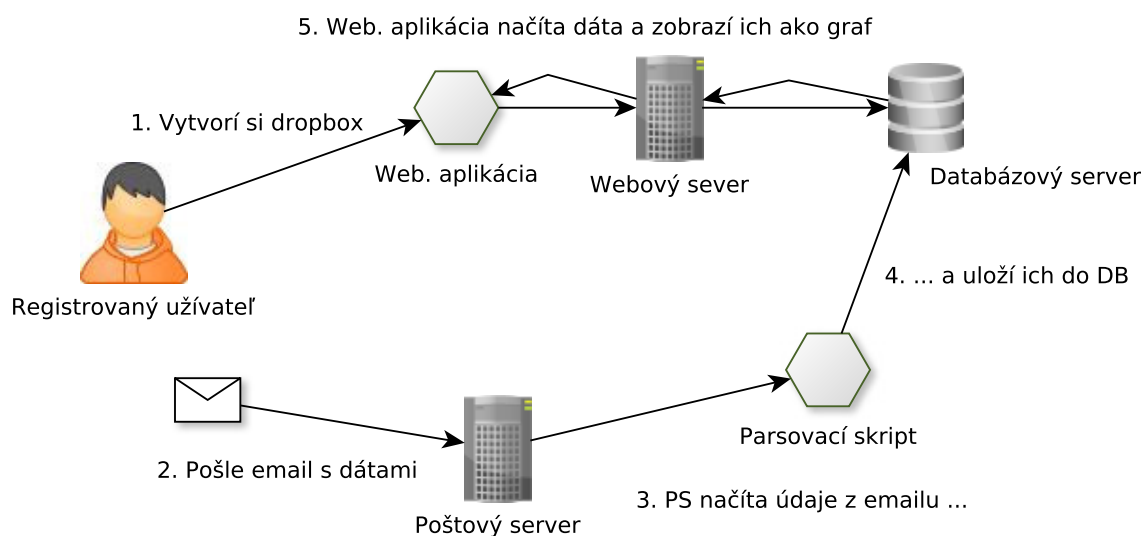


Obrázek 3.1: Business Model Canvas

Kapitola 4

Realizácia

Aplikácia je tvorená niekoľkými samostatnými časťami: poštový server, databázový server, webový server a parsovacie skripty. Webová aplikácia umožňuje prvotnú komunikáciu s užívateľmi. Užívateľ sa v nej najprv zaregistruje a vytvorí si nový dropbox. Okrem toho v nej môže vykonávať aj ďalšie úkony ako mazať dropboxy, meniť parsery, manipulovať s dátami a iné. Potom pošle email na novo vytvorený dropbox. Za nasledujúcu časť je zodpovedný poštový server. Email prijíma a uloží si ho do svojho úložiska. Aplikáčne skripty potom načítajú tieto emaily z úložiska poštového servera a pokúsia sa z nich načítať dáta. V prípade, že sa im to podarí, uložia tieto dáta do databázy. Ak potom užívateľ vo webovej aplikácii zvolí možnosť zobrazenia grafu, dáta z databázy sa načítajú a vykreslia v podobe grafu.

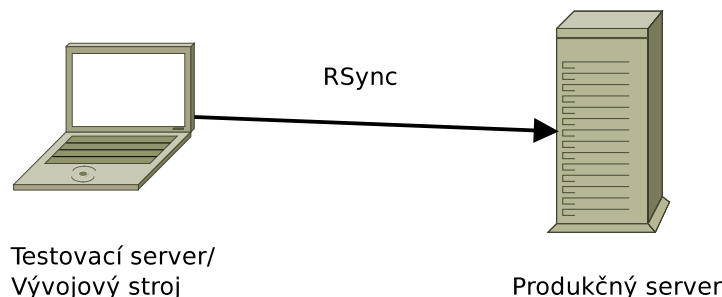


Obrázek 4.1: Architektúra aplikácie

4.1 Nasadenie a správa aplikácie

Pre potreby aplikácie je zriadený hosting na Amazon web services. Aplikácie funguje na virtuálnom serveri s operačným systémom Ubuntu Server. Tento server slúži ako produkčný server. Postupne boli nainštalované a nakonfigurované všetky potrebné súčasti. Po nasadení

sa objavil drobný problém - ostatné emailové servery odmietali s aplikačným komunikovať. Tento problém zmizol po nastavení správnej reverznej DNS adresy. Okrem produkčného servera som mal lokálny testovací server na svojom notebooku. Synchronizáciu medzi oboma servermi zabezpečoval nástroj Rsync. Produkčná schéma je na obrázku 4.2.



Obrázek 4.2: Produkčná schéma

4.2 Poštový server

Internetový email je zostavený z niekoľko štandardov a protokolov, z ktorých každý definuje určitú časť procesu prenosu správy od odosielateľa k prijímateľovi.

4.2.1 Poštový užívateľský agent

Poštový užívateľský agent (Mail User Agent - MUA) je softvér, s ktorým prichádza do kontaktu užívateľ. Slúži na písanie a čítanie správ elektronickej pošty. Typickým príkladom MUA sú mutt, elm, Pine, Mozilla Thunderbird ... Poštový užívateľský agent správy neodosiela sám, ale predáva ich na odoslanie poštovému prenosovému agentovi.^[4]

4.2.2 Poštový prenosový agent

Poštový prenosový agent (Mail Transfer Agent - MTA) má na starosti presun správy z jedného systému na druhý. MTA stanovuje či danú správu prijme alebo nie. Akonáhle správu prijme ju môže odoslať nejakému užívateľovi na vlastnom systéme, prípadne ju môže preposlať inému poštovému prenosovému agentovi. Ak MTA nemôže správu odoslať, odošle ju späť pôvodnému odosielateľovi alebo upozorní správcu systému. Správa nakoniec dorazí na cieľové MTA. Aj je správa určená užívateľovi na tomto systéme, daný MTA ju predá agentovi doručenia správy.^[4]

4.2.3 Agent doručenia správy

Agent doručenia správy (Message Delivery Agent - MDA) zaisťuje finálne doručenie správy. MDA si správu uloží do Úložiska správ (Message Store). Správy v Úložisti správ sú uložené ako súbory alebo vo špeciálnej databáze. Správy zostávajú v Úložisti až do doby, kým si ju nevyzdvihne agent MUA.^[4]

4.2.4 E-mailové protokoly

SMTP

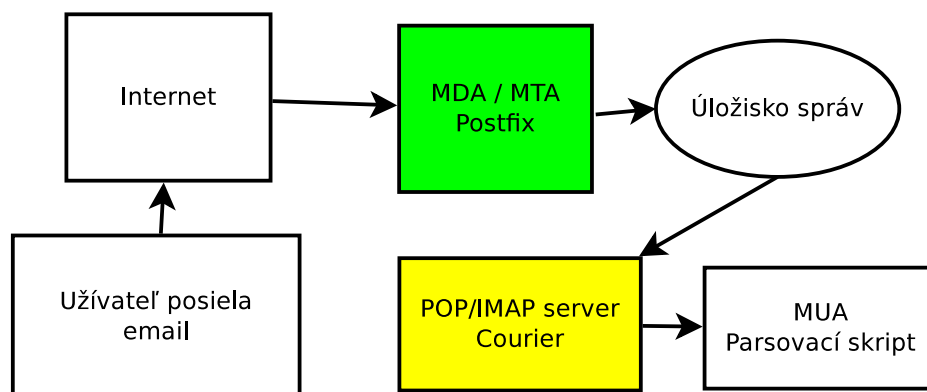
SMTP (Simple mail transfer protocol) slúži k odosielaniu správ a k ich predávaní medzi agentami MTA. Používajú ho tiež agenti MUA, ak požadujú dodanie správy.

POP/IMAP

Pomocou týchto protokolov si užívatelia môžu stiahnuť správy zo serveru. Protokol POP slúži na prevzatie správ zo serveru a na lokálnu prácu s nimi. Protokol IMAP umožňuje správu emailov priamo na serveri.

4.2.5 Použité technológie

Ako MTA som použil Postfix a POP/IMAP server Courier. POP/IMAP server nie je teoreticky nutný, parsovací skript by mohol načítať emaily aj priamo zo súborového systému, no s ohľadom na škálovateľnosť aplikácie je tiež zahrnutý. V prípade náhleho rastu používateľov služby, takto bude možné jednoducho premiestniť poštový server na samostatný fyzický stroj. Pre prístup k emailom sa používa protokol IMAP. Vďaka použitiu protokolu IMAP je možné v rámci jedného účtu vytvoriť adresárovú štruktúru. Parsovacie skripty potom pri parsovaní načítajú správy zo základného priečinku účtu a po spracovaní ich archivujú do priečinku "Parsed". Parsovací skript potom vystupuje v roli MUA, načítava si jednotlivé emaily a ďalej ich spracováva. Táto schéma je zobrazená na obrázku 4.3.



Obrázek 4.3: Tok emailov v rámci emailovej časti aplikácie

4.2.6 Konfigurácia

Postfix vyžaduje minimálne správu užívateľov, aliasov a domén. Počiatočné verzie aplikácie využívali špeciálnu databázu, ktorá obsahovala samostatné tabuľky pre tieto tri atribúty. Keďže dáta v tabuľkách aliasov boli redundantné (aliasy vždy mapovali užívateľa na samého seba a doména bola v každom riadku dbfs.eu), prešiel som na hybridnú konfiguráciu. Na užívateľské účty a domény zostali samostatné tabuľky, tabuľku pre aliasy som zrušil a ich existenciu simuloval špeciálnou konfiguráciou servera. Podobne som upravil aj konfiguráciu servera Courier.

4.2.7 Filtrovanie obsahu a spam

Pôvodný návrh počítal s tým, že emaily sa po skončení parsovania, či už úspešnom alebo neúspešnom, zmažú. Nevyžiadaná pošta by tým pádom nespôsobovala žiadne problémy, pretože by bola po neúspešnom parsovaní jednoducho zmazaná. Tento predpoklad sa však ukázal ako nepraktický, pri nasadení aplikácie do reálnej prevádzky bolo nutné začať archivovať emaily a používať ich pri odstraňovaní užívateľmi objavených chýb parserov. Proti nevyžiadanej pošte je teda systém chránený iným spôsobom - názvy emailových schránok sú náhodné reťazce.

4.3 Webový server

Na webovom serveri beží webová časť aplikácie. Pre tieto účely je využitý server Apache s nainštalovaným modulom PHP. Pôvodne bolo webové rozhranie napísané v jazyku PHP bez použitia frameworku. Toto riešenie sa po istom čase ukázalo ako nevyhovujúce, tak nutné aplikáciu prepísať s využitím PHP frameworku Symfony2.

Prostredníctvom webovej aplikácie sa užívatelia registrujú a prihlasujú. MVP ďalej umožňoval vytvoriť dropbox s jednoduchým typom parsera a jedným typom grafu. Ďalším vývojom pribúdali ďalšie typy parserov a grafov a ďalšie operácie, ktoré môžu užívatelia vykonávať. Môžu dropbox premenovať, vymazať, prípadne mazať len určité údaje, alebo si z neho exportovať údaje. Ďalej majú možnosť meniť nastavenia dropboxu, ale aj celého účtu. Keďže emaily parsuje iná aplikácia, užívatelia majú možnosť vynútiť si parsovanie. Na obrázku 4.4 je zobrazený aktuálny use case diagram webovej časti aplikácie.

4.3.1 HTTP

Http (Hypertext transfer protocol) je textový protokol, umožňujúci komunikáciu medzi dvoma počítačmi. Je základom webovej komunikácie. Všetka webová komunikácia prebieha formou žiadosť - odpoveď. Klient (webový prehliadač) vytvára žiadosť a zasiela ju serveru. Žiadosť obsahuje okrem iného dve podstatné informácie - metódu HTTP a URI požadovaného zdroja. Len čo server dostane žiadosť, podľa URI vie aký zdroj klient požaduje a podľa metódy akú akciu si užívateľ praje vykonať a podľa toho vytvorí odpoveď. Odpoveď môže obsahovať požadovaný zdroj, no vždy obsahuje status kód, ktorým server klienta informuje, či bola akcia úspešná, či a aká chyba nastala, či je nutné presmerovanie.

4.3.2 URI

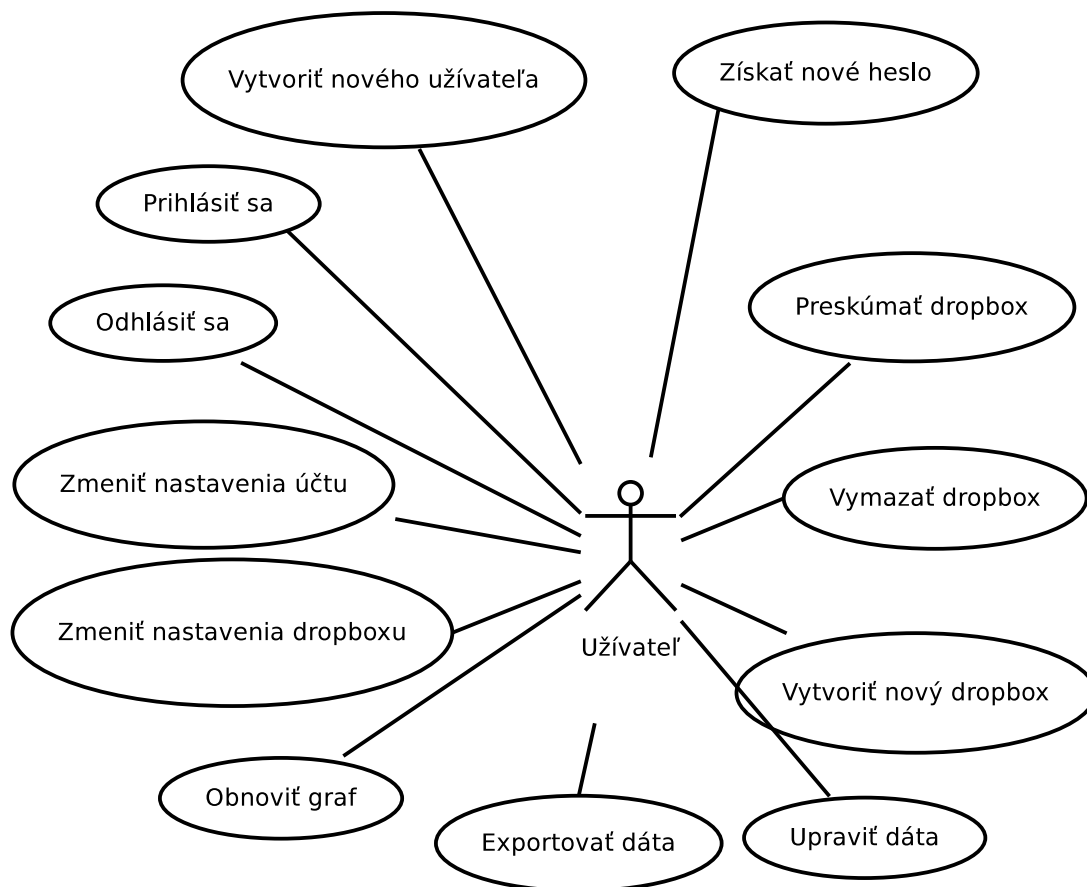
URI je unikátna adresa, ktorá identifikuje zdroj, ktorý klient požaduje.

4.3.3 MVC

MVC (Model – view – controller) je návrhový vzor softvérovej architektúry, ktorý delí aplikáciu na tri komponenty: radiče, modely a pohľady.

Radič

Radič (Controller) spracováva užívateľský vstup a vytvára reakciu. Využíva pritom modely a pohľady.



Obrázek 4.4: Use case diagram

Model

Model (model) obsahuje samotnú funkcionálnu a dáta.

Pohľad

Pohľad (view) zobrazuje dáta z modelu užívateľovi.

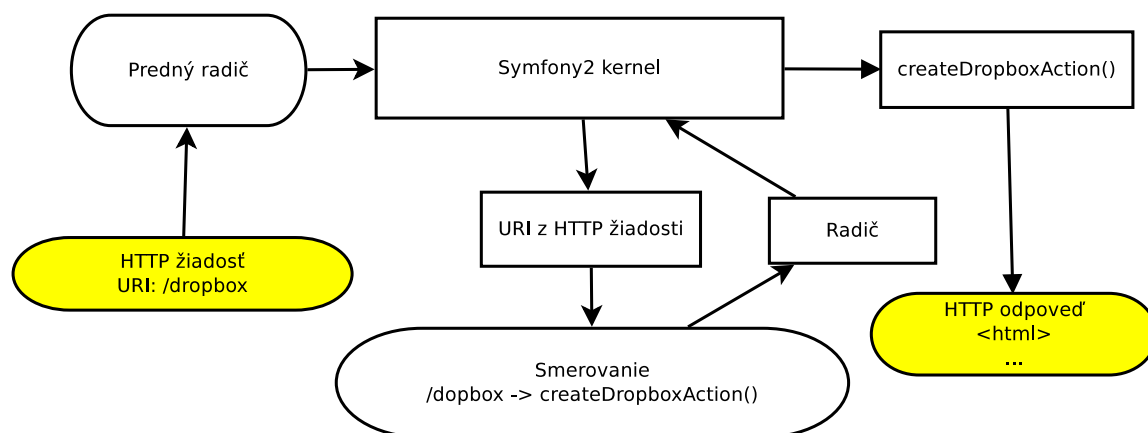
4.3.4 Predný radič

Predný radič (front controller) je návrhový vzor, ktorý poskytuje centralizovaný vstupný bod pre spracovanie žiadostí. V konkrétnom prostredí webových aplikácií sa v ňom spracováva každá HTTP žiadosť. Namiesto toho aby jednotlivé URI spúšťali rôzne súbory, sa práve spúšťa vždy predný radič, ktorý má na starosti smerovanie rôznych URI na príslušné radiče. Vďaka centralizácii je výmena vývojového prostredia za produkčné veľmi jednoduchá.

4.3.5 Architektúra webovej aplikácie

Webová aplikácia využíva spomínané návrhové vzory v rámci frameworku Symfony2. Predný radič prijme http žiadosť, následne ju odovzdá na spracovanie Symfony kernelu. Mapovanie URI na radič je uložené v smerovacom súbore. Kernel prečíta URI zo žiadosti, vyhľadá

príslušný smerovací súbor a na jeho základe ju nasmeruje na príslušný radič. Radič potom vytvorí (s prípadným využitím modelu) http odpoveď a odošle ju naspäť klientovi. Odpovede sú prezentované ako pohľady. Tento princíp je znázornený na obrázku 4.5.



Obrázek 4.5: Zjednodušený model aplikácie. Proces, ktorým sa z http žiadosti vytvorí http odpoveď. Prichádzajúce http žiadosti predáva predný radič Symfony2 kernelu. Kernel z nich prečíta URI a nasmeruje ich na príslušný radič.

Pre účely aplikácie je vytvorený balík **PecsiDBFSBundle**. Súčasťou tohto balíku sú všetky radiče, modely, konfiguračné súbory, pohľady. Existujú v ňom dve triedy radičov - DBManagmentController a UserManagmentController.

DBManagmentController

Trieda DBManagmentController obsahuje radiče a modely súvisiace so správou dropboxov. Obsahuje radiče pre tvorbu dropboxov, ich mazanie, obnovu dát, zmenu nastavení, export dát, zobrazenie grafu z dát dropboxov a nápovedy k vybraným dropboxom.

UserManagmentController

Trieda UserManagmentController obsahuje radiče a modely používané na úrovni užívateľov. Obsahuje radiče pre výpis všetkých dropboxov daného užívateľa, registráciu, prihlásenie, odhlásenie, zmenu nastavení účtu a reset hesla.

Entity

Triedy AccountSettingsEntity, LoginEntity, RegisterEntity, SettingsEntity a UserNameEntity reprezentujú rôzny typy html formulárov použitých v aplikácii. Umožňujú pristupovať k dátam z formulárov ako k atribútom objektov týchto tried.

Pohľady

Na tvorbu pohľadov je použitý šablónovací jazyk twig¹. Aplikácia používa štyri typy šablón:

¹<http://twig.sensiolabs.org/>

- **Charts** - šablóny na tvorbu grafov. Základná trieda je `base_chart.html.twig`. Definuje vzhľad stránky s grafom. Ostatné triedy v tejto zložke `base_chart` rozširujú a používajú špecifický graf na vykreslenie dát.
- **Export** - šablóny generujúce exportované dáta. Momentálne sa tu nachádza jediná trieda generujúca dáta vo formáte CSV.
- **Help** - šablóny generujúce nápovedy. Momentálne sa tu nachádzajú triedy generujúce nápovedu pre Simple parser a Dropbox.
- **Users** - šablóny, ktoré vytvárajú html stránky zodpovedajúce http odpovediam radičov z triedy `UserManagementController`.

4.4 Databázový server

Ako databázový server je použitý MySQL. Pôvodný návrh počítal s tromi samostatnými databázami: databáza poštového servera, databáza autentifikačných údajov užívateľov a databáza dát dropboxov. V priebehu vývoja ale začalo byť nutné používať abstraktnú databázovú vrstvu Doctrine a táto schéma sa ukázala ako nevhodná. Spomínané tri databázy sa zredukovali do jednej nazvanej **appdb**. Databáza **appdb** obsahuje dva typy tabuliek - tabuľky, ktoré existujú od začiatku existencie aplikácie a tabuľky, ktoré vznikajú dynamicky za behu aplikácie. Tabuľky existujúce od začiatku behu obsahujú údaje o užívateľoch a metadáta dropboxov. Do dynamicky vznikajúcich tabuliek sa ukladajú naparsované dáta z jednotlivých dropboxov.

4.4.1 Tabuľky existujúce pri nasadení

Tieto tabuľky existujú v databáze od samotného začiatku behu aplikácie. Schéma tejto časti databázy je zobrazená na obrázku 4.6.

Tabuľka **authorized_users** obsahuje údaje o prihlásených užívateľoch. Uchováva email užívateľa, hash hesla, dátum a čas registrácie užívateľa, dátum a čas posledného prístupu k účtu a informáciu, či je užívateľ povolený.

Tabuľky **domains** a **mailboxes** sú nevyhnutné pre chod poštového servera. Tabuľka **domains** obsahuje domény spracovávané poštovým serverom. Každá doména má svoj názov, typ a informáciu či je povolená. Tabuľka **mailboxes** obsahuje všetky poštové schránky poštového servera. Každá poštová schránka má svoj náhodný identifikátor, vlastníka, dátum a čas vytvorenia, dátum a čas poslednej úpravy, atribút `pattern`, ktorý je rezervovaný na neskoršie použitie, typ parsera, ktorý spracováva dáta v danej poštovej schránke, typ grafu, akým sa majú spracované dáta zobraziť a alias dropboxu, ktorého je daná schránka súčasťou.

Dáta každého dropboxu sa vykresľujú do grafu. Tieto dáta sú tvorené jednou x-ovou a ľubovoľným počtom y-ových premenných. Aby bolo možné názvy týchto premenných meniť, existuje špeciálna vrstva, ktorá mapuje názvy premenných na ich hodnoty. Toto mapovanie je uložené v tabuľke `templates`.

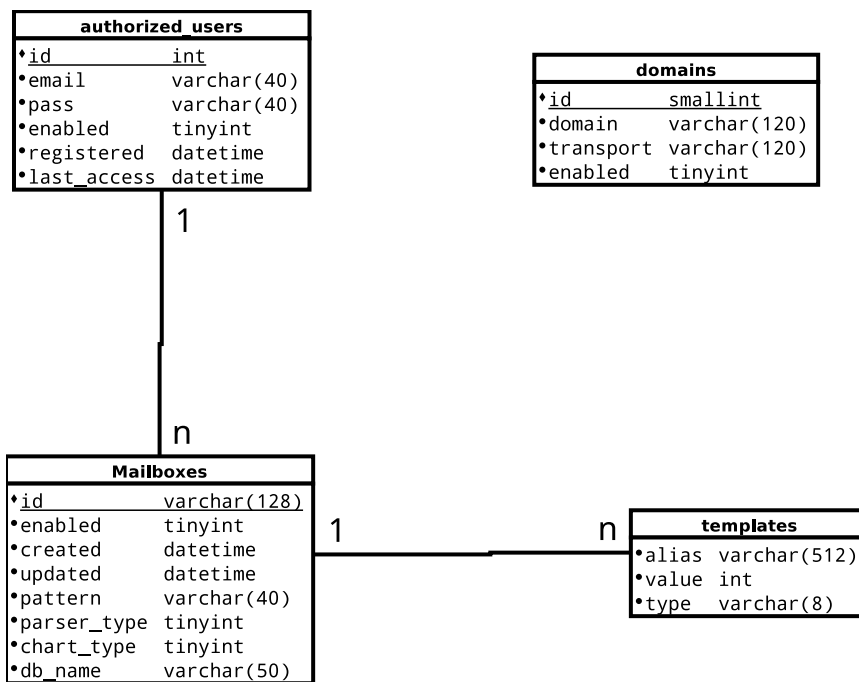
Tabuľka **templates** obsahuje šablóny, podľa ktorých sa mapujú názvy premenných dropboxu na stĺpce tabuľky, v ktorých sú uložené ich dáta. Jednu šablónu tvorí viac riadkov v tejto tabuľke, pre každú premennú dropboxu jeden. Každý riadok obsahuje identifikátor dropboxu, ktorého sa týka, typ šablóny, hodnotu stĺpca (`value`) z ktorého sa dá odvodiť

názov stĺpca dropboxovej tabuľky, kam sa budú dáta ukladať a alias stĺpca (názov, ktorý sa zobrazuje v grafe). Typ šablóny hovorí, či ide o x-ovú alebo y-ovú premennú.

Y-ové premenné sú uložené v tabuľke dát dropboxu, v stĺpci nazvanom "ya", kde a je hodnota stĺpca (value). Napríklad y-nová premenná foo, bude mať v tabuľke šablón mapovanie foo na 3 a v príslušnej tabuľke hodnôt bude stĺpec s názvom y3, v ktorom budú uložené všetky hodnoty premennej foo.

X-ové premenné fungujú na inom princípe. Každý dropbox má práve jednu x-ovú premennú. V tabuľke šablón sa dá vďaka tomu nájsť podľa typu šablóny a hodnota stĺpca sa ignoruje. Názov stĺpca v tabuľke hodnôt dropboxu je vždy x. Dôvod tohto riešenia je ten, že hodnota stĺpca musí byť jedinečné číslo (je to jednoznačný identifikátor stĺpca) a jedinečnosť je riešená ako integritné pravidlo v databáze. V dobe vzniku tabuľky dát, nie je jasné aká by mala byť hodnota stĺpca v tabuľke šablón, aby bola jedinečná a zároveň ľahko dohľadateľná.

Pri úprave tabuľky dát dropboxov (pridávaní novej premennej) sa pridáva nový riadok aj do tabuľky šablón. Pri vykresľovaní grafu sa potom najprv načíta šablóna dropboxu a na základe nej potom aj príslušné dáta. Premenovanie premenných funguje jednoduchým prepísaním aliasu na nový názov.



Obrázek 4.6: Schéma databázy

4.4.2 Tabuľky s dátami dropboxov

Užívatelia majú možnosť vytvárať si dropboxy, ktoré môžu uchovávať rozličné dáta. Preto musí pre každý dropbox existovať samostatná tabuľka, do ktorej sa ukladajú dáta získané z emailov. Tieto tabuľky vznikajú pri užívateľovom príkaze na vytvorenie nového dropboxu. Názov každej tabuľky je tvorený prefixom "db." a identifikátorom daného dropboxu. Tabuľka má implicitne jediný atribút - primárny kľúč nazvaný x (reprezentuje premennú zobrazenú na osi x) typu dátum. Ďalšie stĺpce sa pridávajú dynamicky počas parsovania

emailov. Každý z týchto stĺpcov má názov tvorený prefixom "y" a číslom, ktoré je jedinečné v rámci celej aplikácie. Táto schéma umožňuje meniť zobrazované názvy premenných, ktoré sa zobrazujú v grafe. Schéma prázdnej tabuľky je zobrazená na obrázku 4.7 a po pridaní troch premenných na obrázku 4.8.

db_570de8c31f219dd7f7e59e60b1732287316ae003	
♦x	<u>date</u>

Obrázek 4.7: Príklad tabuľky na ukladanie dát dropboxov

db_570de8c31f219dd7f7e59e60b1732287316ae003	
♦x	<u>date</u>
°y1	double
°y2	double
°y3	double

Obrázek 4.8: Počas parsovania sa mení schéma tabuľky

4.4.3 Bezpečnosť

V databázach sa používajú traja oddelení užívatelia (okrem roota) - webový server, poštový server a parsovací skript. Každý z nich funguje s minimálnym množstvom práv potrebných k jeho funkčnosti.

4.5 Parsovací skript

Parser je program (bežne súčasť kompilátora), ktorý prijíma textový vstup v určitom formáte, rozdeľuje ho na časti, kontroluje či vyhovuje zadanej sade pravidiel. Parsery v tejto aplikácii vykonávajú lexikálnu, syntaktickú a sémantickú analýzu textu prijatých emailov, načítávajú z nich zadané údaje a ukladajú ich do databázy.

Podľa [1] má lexikálny analyzátor na starosti vytváranie tokenov - skupiny znakov, tvoriace syntaktickú entitu. Každý token má svoj typ a hodnotu. Výstupom lexikálneho analyzátoru je sada tokenov. Tie preberá syntaktický analyzátor. Syntaktický analyzátor potom overuje či táto sada vyhovuje syntaktickým pravidlám jazyka a sémantická analýza kontroluje význam daných konštrukcií.

Parsovacie skripty sú napísané v jazyku Python verzie 2.7. Prechádzajú všetky emailové schránky uložené v databáze. Výsledky ukladajú do príslušných tabuliek. Podľa pôvodného návrhu mal byť skript spúšťaný periodicky každých 10 minút pomocou nástroja CRON. Pretože čakanie bolo neúnosné, znížil sa najprv interval periodických volaní na 5 minút a v súčasnosti je skript upravený tak, aby nebolo nutné žiadne čakanie.

Celý skript sa skladá zo šiestich modulov: App.py, Parser.py, SimpleParser.py, Add-ThisParser.py, FacebookParser.py, Mail.py.

4.5.1 App.py

Modul App.py je jediný spustiteľný modul. Prehľadáva všetky dropboxy a spúšťa nad nimi príslušný parser.

4.5.2 Parser.py

Modul Parser.py obsahuje metódy, ktoré všeobecný parser potrebuje k práci s dropboxom. Umožňuje pridávať nové premenné do dropboxu, premenovať ich, načítať šablónu dropboxu, načítať a nastaviť x-ovú premennú dropboxu. Dokáže previesť identifikátor užívateľa na jeho emailovú adresu (v prípade chyby v parsovaní sa na ňu odosiela chybová správa). Definuje tiež triedy reprezentujúce syntaktické a sémantické chyby.

4.5.3 Mail.py

Modul Mail.py vytvára emaily, ktoré sa v prípade chyby odosielajú užívateľovi.

4.5.4 SimpleParser.py

SimpleParser definuje jednoduchý jazyk, pomocou ktorého môžu užívatelia posilať dáta na dropbox. SimpleParser bol súčasťou MVP. Každý email je tvorený ľubovoľným počtom "časových záznamov". Časový záznam pozostáva z jednej x-ovej premennej (dátum) a ľubovoľného počtu y-nových premenných (reálne čísla) (napr. datum = 2.2.2008 foo = 3). Časový záznam musí vždy začínať definíciou x-ovej premennej a priradením jej hodnoty. V prípade, že v dropboxe ešte x-ová premenná nie je definovaná, je možné jej zvoliť ľubovoľný názov. Pri ďalšom používaní je potrebné používať tento názov (napr. ďalší záznam by mohol byť datum = 3.2.2008 foo = 2, ale nie date = 3.2.2008 alebo x = 3.2.2008). Y-ové premenné je možné do časových záznamov ľubovoľne pridávať a odoberať (napr. ak ma od 4.2. zaujíma aj premenná bar datum = 4.2.2008 foo = 2.3 bar = 15). Ak časový záznam ešte neexistuje, vytvorí sa k príslušnému dátumu dané premenné a priradí sa im hodnota. Vynechané premenné sa pridávajú ako nedefinované. Ak už pre dátum existuje časový záznam, tak sa aktualizuje. Hodnoty vynechaných premenných zostávajú nedotknuté. Týmto spôsobom je možné spätne priradiť hodnoty premenným, ktoré vznikli až neskôr (napr. datum = 2.2. bar = 10). V rámci časových záznamov je možné premenné aj premenovať (napr. bar > value premenuje premennú bar na value). Parser najprv rozdelí text na sadu tokenov, potom skontroluje či vyhovujú uvedeným pravidlám a v prípade úspechu uloží dáta do databázy. EBNF tohto jazyka je zobrazená na obrázku 4.9.

```
pismo = "A" | "B" | "C" | "D" | "E" | "F" | "G"
        | "H" | "I" | "J" | "K" | "L" | "M" | "N"
        | "O" | "P" | "Q" | "R" | "S" | "T" | "U"
        | "V" | "W" | "X" | "Y" | "Z" | "a" | "b" | "c"
        | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l"
        | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" |
        | "v" | "w" | "x" | "y" | "z" ;
cislica = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" ;
identifikator = pismo , { pismo | cislica | "_" } ;
realne_cislo = [ "-" | "+" ] , ( { cislo } , [ "." ] , { cislo } ) | ( "." , { cislo } ) ,
               [ "E" | "e" , [ "-" | "+" ] , cislo , { cislo } ] ;
datum = cislica , cislica , "." , cislica , cislica , "." , cislica , cislica , cislica , cislica ;
x_prem = identifikator , [ whitespace ] , "=" , [ whitespace ] , datum ;
y_prem = identifikator , [ whitespace ] , "=" , [ whitespace ] , realne_cislo ;
zmena_mena = identifikator , [ whitespace ] , ">" , [ whitespace ] , identifikator ;
priказ = x_prem | y_prem | zmena_mena ;
email = { priказ } ;
```

Obrázek 4.9: EBNF jazyka SimpleParsera

4.5.5 AddThisParser.py

AddThisParser vznikol až po nasadení aplikácie. Na rozdiel od SimpleParsera neparsuje jednoduchý text (plain text), ale prehľadáva emaily formátu html, ktoré generuje služba AddThisAnalytics ², vyberá z nich údaje a ukladá do databázy. Podobne ako FacebookParser najprv rozloží pomocou knižnice BeautifulSoup html stránku na strom. Filozofiou oboch týchto parserov je, že AddThisAnalytics a Facebook síce môžu generovať správy v odlišných jazykoch, ale stále rovnako formátované. Preto v html strome vyhľadajú uzly s danými CSS vlastnosťami, overia ich počet a typ a v prípade úspechu uložia do databázy.

Keďže preposielanie emailov pomocou niektorých emailových klientov deformovalo html štruktúru emailov, tento parser skúša parsovať aj podľa regulárnych výrazov a parsovanie uspeje, ak sa podarí aspoň jeden typ parsovania.

4.5.6 FacebookParser.py

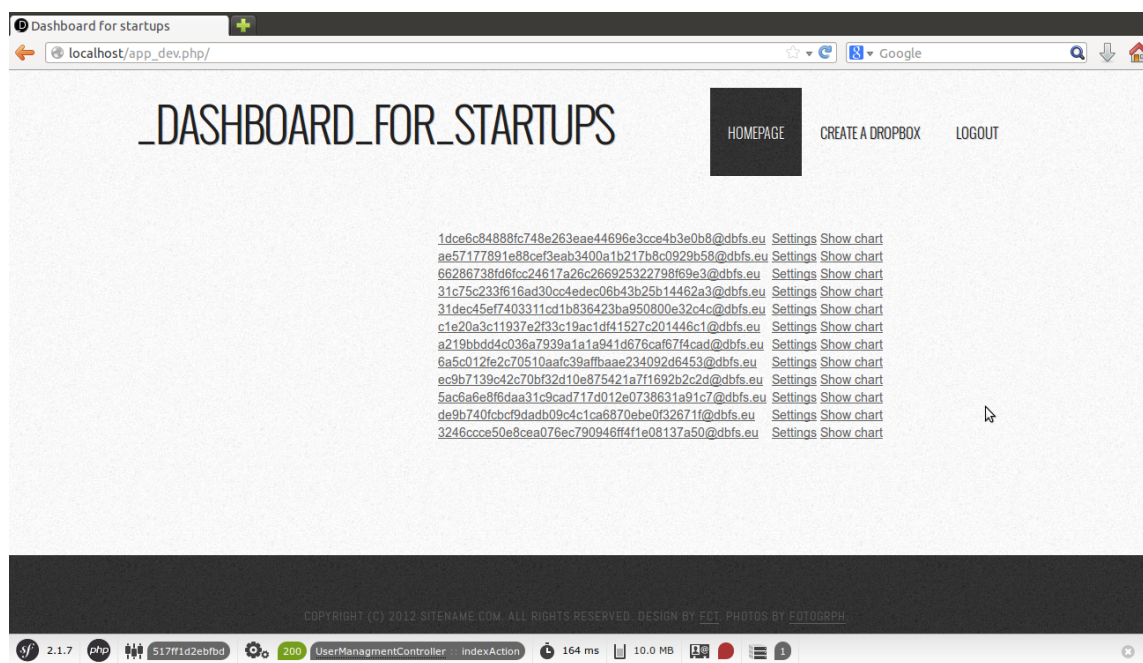
FacebookParser spracováva emaily s html formátom, ktoré generuje Facebook o aktivitách na stránkach. Tieto emaily neobsahujú dátum, preto tento parser namiesto toho používa dátum prijatia emailu z hlavičky emailu. V prvej fáze overuje, či poslaný email má html formát a či vyhovuje formátu emailov, ktorý zasiela Facebook. Email od Facebooku obsahujú informácie o "Páči sa mi", "Hovorí sa o tom" a o dosahu. Parser vyhľadá príslušné uzly html stromu (podľa vlastností CSS), overí ich počet a typ. Vďaka tomuto princípu dokáže parser spracovať aj emaily, ktoré sú v odlišných jazykoch. Musia ale používať rovnaké formátovanie.

²www.addthis.com/analytics

Kapitola 5

Vývoj riadený užívateľmi

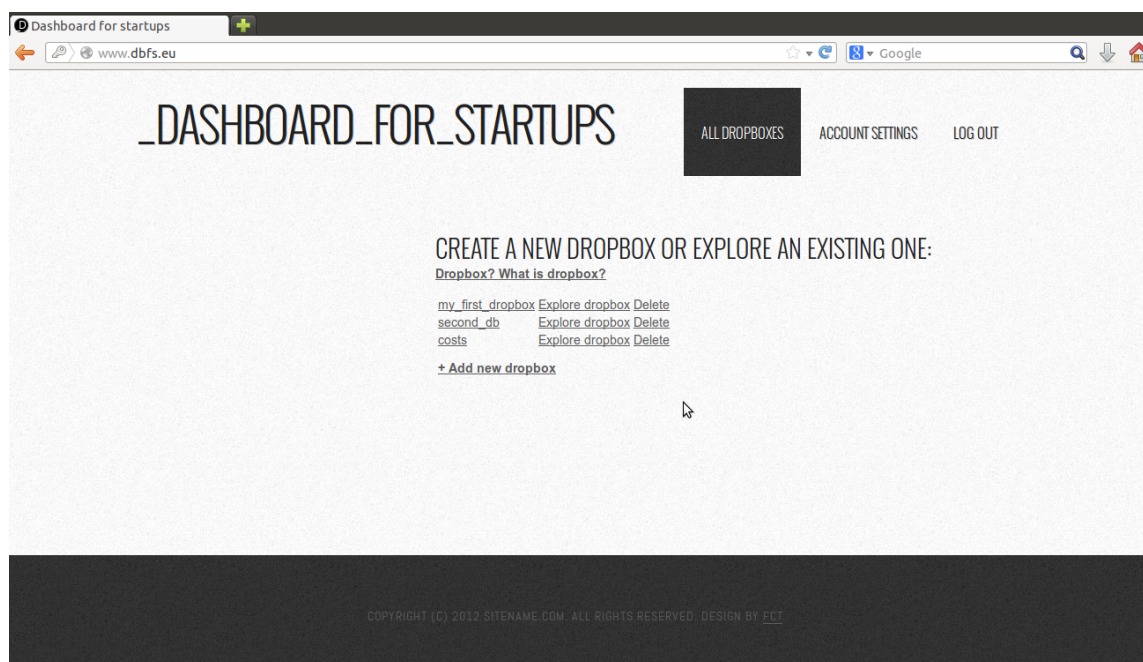
Minimum Viable Product bol hotový a nasadený 12.3.2013. Pokúsil som sa rozšíriť aplikáciu medzi ľuďmi. Keďže cieľová skupina aplikácie je veľmi špecifická, sú to ľudia, ktorí majú istý stupeň zručnosti a skúsenosti s prácou s počítačom, rozhodol som sa aplikáciu propagovať medzi známimi z IT. Zároveň som ich požiadal, aby aplikáciu preposlali ďalším ľuďom, ktorí by ju mohli využiť.



Obrázek 5.1: Minimum Viable Product. Užívatelia mali problém nájsť tlačidlo na vytvorenie nového dropboxu v hornej lište.

Podľa výsledkov Google Analytics navštevovalo úvodnú stránku pomerne veľa ľudí, ale na registráciu pokračoval iba zlomok z nich. Môj predpoklad bol, že stránka je príliš neprehľadná a nedostatočne vysvetľuje princíp aplikácie a teda dôvod prečo by mali používatelia aplikáciu používať. Preto som sa snažil zatraktívniť úvodnú stránku a vytvoriť banner vysvetľujúci ako služba funguje.

Problém sa opakoval a užívatelia sa síce registrovali, ale nevytvárali dropboxy. Mal som už k dispozícii reakcie priamo od užívateľov. Užívatelia sa sťažovali, že im nie je jasné



Obrázek 5.2: Tá istá stránka po úpravách. Keďže k dropboxu pribudli ďalšie možnosti ako ním manipulovať. Zmenilo sa "show chart" na "explore dropbox". Pridaná je aj nápoveda a možnosť zmeniť názov dropboxu.

čo by aplikácia mala robiť. Postupne som sa teda pridával nápovedy. Pridanie nápoved nepomohlo, užívatelia sa podľa Google Analytics dostávali stále iba po registráciu.

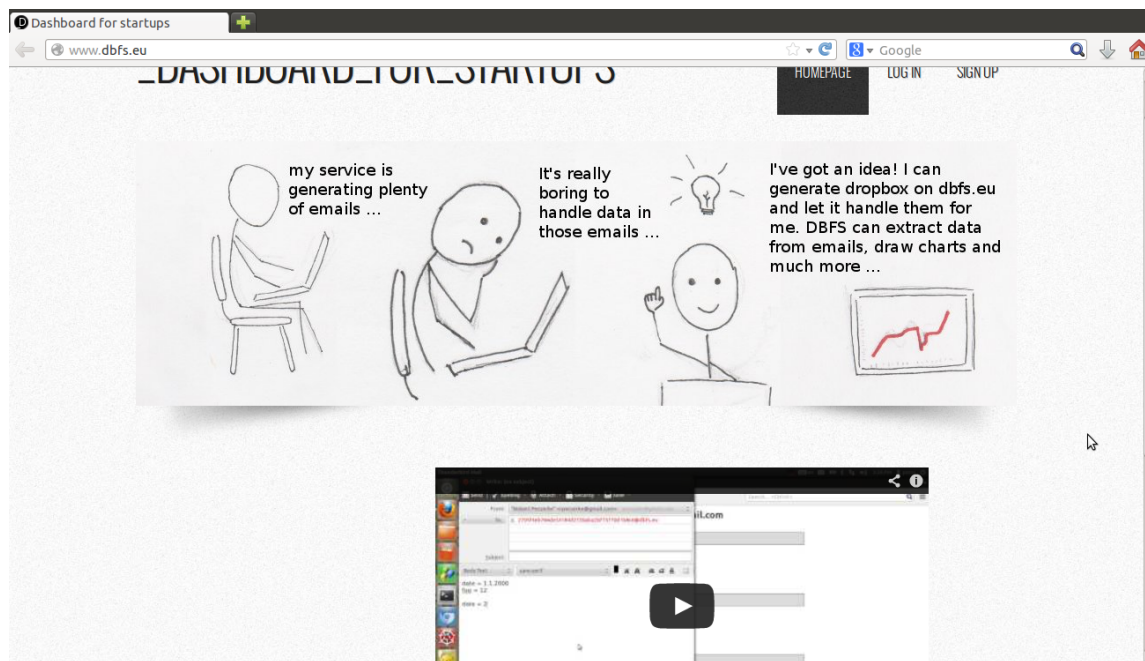
Vznikol predpoklad, že užívatelia nevedia nájsť tlačidlo na vytvorenie nového dropboxu. Pôvodné tlačidlo na vytváranie nového dropboxu bolo umiestnené v hornej lište ako je vidieť na obrázku 5.1. Zmenil som rozloženie tlačidiel na príslušnej stránke, tak aby tlačidlo na vytvorenie nového dropboxu bolo pri zozname už existujúcich dropboxov a približne v strede obrazovky v prípade, že užívateľ ešte nemá vytvorený žiadny dropbox.

Toto riešenie pomohlo a užívatelia si začali vytvárať dropboxy, ale neposielali na ne žiadne dáta. V tejto dobe aplikácia poskytovala iba SimpleParser. Hoci som sa snažil, aby bol použitý jazyk čo najjednoduchší, moja hypotéza bola, že posielanie dát je stále príliš komplikované. Vytvoril som preto vizuálnu nápovedu, kde je na obrázkoch znázornené použitie SimpleParsera a pridal k nemu tlačidlo, ktoré vytvorí ukážkový email v predvolenom emailovom klientovi.

Okrem toho som pridal dva nové typy parserov FacebookParser a AddThisParser, ktoré nevyžadovali učenie sa jazyka. Stačilo na vytvorené dropboxy presmerovať emaily, ktoré vygenerovali príslušné služby.

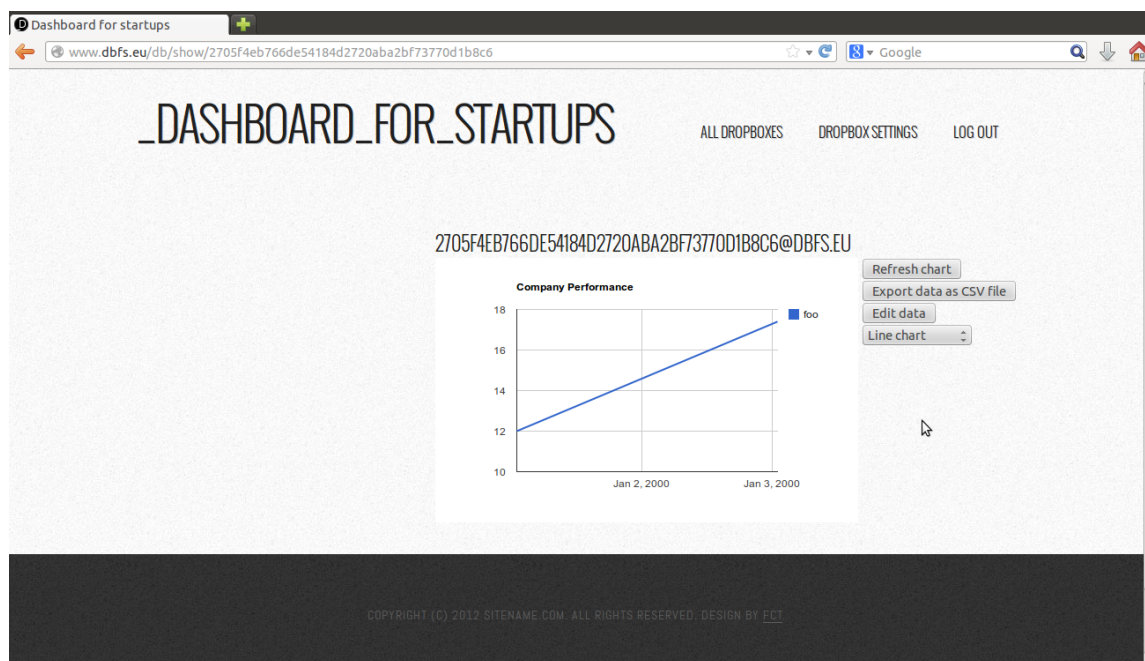
Pri AddThisParseri vznikol problém v tom, že užívatelia posielali emaily cez klienta, ktorý deformoval pôvodnú html štruktúru emailov. Riešením tejto situácie bolo pridanie možnosti, aby sa AddThisParser v prípade, že sa nepodarí parsovanie prechodom cez html strom pokúsil získať príslušné informácie prostredníctvom regulárnych výrazov.

Vďaka týmto krokom sa už objavili užívatelia, ktorí začali aplikáciu normálne používať. Títo žiadali pridať možnosť premenovať dropboxy, pretože keď vytvorili viac dropboxov, strácali prehľad, aké dáta uchováva. Pridanie tejto možnosti, ale niektorých užívateľov zmätlo a začali odosielať emaily na nimi vytvorený alias a nie na skutočný dropbox. Preto



Obrázek 5.3: Inovovaná úvodná stránka s popisným bannerom a videom

som pridal vysvetľujúci nápis pri vytváraní nového mena dropboxu. Pre niektorých užívateľov, bolo ale používanie aplikácie stále príliš náročné. Po žiadosti jedného z užívateľov som pridal aj video znázorňujúce použitie služby.



Obrázek 5.4: Z pôvodného iba zobrazenia grafu sa stal pravý dashboard s priamou možnosťou meniť typy grafov, exportovať údaje a zmazať ich.

Pri kontrole logov parsovacích skriptov som prišiel na to, že veľké množstvo parsovaných

emailov končí chybou. Všetky chyby sa týkali dropboxov, ktoré používali SimpleParser. SimpleParser počítal s emailmi s jednoduchým textom (plaintext). Tento fakt je napísaný aj pri jeho popise. Pri kontrole logov som ale zistil, že veľa parsování zlyhalo práve na tom, že ľudia posielali html emaily namiesto plaintextových.

Podľa RFC1521 [3] emaily používajúce Content-Type: multipart/alternative obsahujú z dôvodu zachovania kompatibility aj oblasť Content-Type: text/plain, v ktorej sa nachádza plaintextový obsah emailu. Preto som upravil SimpleParser, aby vedel spracovať aj tieto typy emailov, pokiaľ ich formát dodržiava tento štandard. Tento krok výrazne zvýšil úspešnosť parsovania.

Na žiadosť užívateľov som pridal možnosť meniť typ grafu priamo z obrazovky, na ktorej je graf zobrazený. Pôvodne bola táto možnosť v nastaveniach dropboxu. V nastaveniach sa zmenila možnosť nastaviť graf na nastaviť predvolený graf. Okrem toho pribudli aj tlačidlá umožňujúce exportovať dáta a mazať ich.

Podľa výsledkov z Google Analytics bolo vidieť, že existuje stále skupina užívateľov, ktorí si vytvorili dropbox a v tomto mieste sa zasekli. Predpokladal som, že sú to užívatelia, ktorí nečítali nápovedy, preto sa užívateľom zobrazí nápoveda na SimpleParser aj vtedy, ak si prezerajú dropbox využívajúci tento typ parsera, na ktorý neposlali žiadne dáta.

Kapitola 6

Záver

Cieľom práce bolo vytvoriť webovú službu, ktorá bude ukladať a vykresľovať grafy z dát, ktoré mu užívatelia zašlú emailom. Vytvoril som Minimum Viable Product a nechal ho otestovať užívateľom. Z pôvodného vykresľovania grafov sa na základe požiadavkov užívateľov stala aplikácia, ktorá spravuje dáta z rôznych služieb, ktoré generujú emaily a umožňuje ich nielen vykresľovať, ale aj inak nimi manipulovať (mazať, exportovať, ...).

Aplikácia v podstate naplnila očakávania. Našla si svojich užívateľov, ktorých síce zatiaľ nie je mnoho, ale to je možno dané krátkym časom, ktorý existuje. Užívatelia aplikáciu hodnotili pozitívne, hoci spočiatku bola pre nich príliš komplikovaná. Ukázalo sa, že parsery FacebookParser a AddThisParser sú populárnejšie ako parser s vlastným jazykom, zrejme kvôli jednoduchosti použitia.

Medzi užívateľmi je o takéto typy parserov záujem, dokonca som dostal požiadavku na pridanie parsera na správy generované systémom Nagios.

Okrem toho na aplikácii je ešte stále veľa vecí, ktoré by sa dali zlepšovať. Do budúcnosti by som chcel pridať možnosť meniť formát dátumu, aby bola použiteľnejšia aj pre ľudí z iných krajín. Okrem toho by som chcel vytvoriť všeobecný parser na html emaily, ktorý by fungoval tak, že užívatelia by najprv poslali email, aplikácia by v ňom našla všetky potenciálne zaujímavé informácie (čísla, dátumy) a nechala by užívateľa si vybrať, ktoré z nich ho zaujímajú. Ak by potom posielal emaily s rovnakým formátom, tieto dáta by sa ukladali do databázy dropboxu.

Ďalšie rozšírenie tejto práce by mohlo spočívať v žiadostiach užívateľov, ktoré som už nestihol implementovať:

- Možnosť načítať excelovské súbory
- Možnosť meniť dáta v dropboxovských tabuľkách aj z webového rozhrania

Literatura

- [1] Aho, A.: *The theory of parsing, translation, and compiling*. Englewood Cliffs, N.J: Prentice-Hall, 1972, ISBN 0-13-914556-7.
- [2] Blank, S.: *The startup owner's manual. the step-by-step guide for building a great company*. Pescadero, Calif: K&S Ranch, Inc, 2012, ISBN 978-0-9849993-0-9.
- [3] Borenstein, N.; Freed, N.: RFC 1521: MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies. 1993.
URL <http://tools.ietf.org/rfc/rfc1521.txt>
- [4] Dent, K.: *Postfix : the definitive guide*. Sebastopol, CA: O'Reilly, 2004, ISBN 978-0-596-00212-1.