

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

Gmail Assistant

2011

Andrej Bihuň

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

GMAIL Personal Assistant

Bakalárska práca

Evidenčné číslo: 90c46108-c979-48e9-9f94-f129e76e77b4

Študijný program : Aplikovaná informatika

Študijný odbor: 9.2.9 Aplikovaná informatika

Školiace pracovisko: Katedra aplikovanej informatiky

Školiteľ: Mgr. Pavel Petrovič, PhD.

Bratislava, 2011

Andrej Bihuň



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Andrej Bihuň
Študijný program: aplikovaná informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.2.9. aplikovaná informatika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský

Názov: Gmail Assistant


Cieľ: Navrhnuť a implementovať webovú aplikáciu rozširujúcu funkcionality mailového systému Gmail podľa špecifikácie zadania.

Vedúci: Mgr. Pavel Petrovič, PhD.

Spôsob sprístupnenia elektronickej verzie práce:
bez obmedzenia

Dátum zadania: 09.11.2009

Dátum schválenia: 12.10.2010


doc. RNDr. Mária Markošová, PhD.
garant študijného programu



študent



vedúci

Čestné prehlásenie

Čestne prehlasujem, že túto bakalársku prácu som vypracoval samostatne pod odborným dohľadom a vedením Pavla Petroviča s použitím uvedenej literatúry.

V Bratislave:

.....

podpis autora

Poďakovanie

Týmto sa chcem poďakovať svojmu školiteľovi za jeho nápady, rady a hlavne trpezlivosť. Mojim rodičom za podporu počas štúdia.

Abstrakt

Cieľom mojej bakalárskej práce je navrhnúť a implementovať webovú aplikáciu rozširujúcu Gmail o funkcie ktoré zlepšia používanie.

Aplikácia má zobrazovať emaily chronologicky na rozdiel od originálu, kde sú radené do konverzácií. Ďalej má ponúknuť možnosť plánovaného odosielania emailov a v neposlednom rade ponúkne správu príloh.

Kľúčové slová: Gmail, email, imap, smtp

Abstract

The goal of this thesis is to design and implement a web application that adds several new features to Gmail, enriching its user experience.

While Gmail show the email messages as conversations, our application displays them in chronological order. Moreover, we also added the support for scheduling of automated e-mail sending and implemented a comprehensive system for attachment management.

Key words: Gmail, email, imap, smtp

OBSAH

1.	ÚVOD	7
2.	PREHLÁD PROBLEMATIKY	8
2.1.	Email.....	8
2.1.1.	Formát emailu	8
2.2.	Ako Email funguje	9
2.3.	Emailový klient.....	9
2.3.1.	Web-based	10
2.3.1.1.	Gmail.....	10
2.3.1.1.1.	Gmail Labs.....	10
2.3.1.1.2.	Better Gmail.....	10
2.3.2.	Desktop applications.....	11
2.3.2.1.	Microsoft Outlook	12
2.3.2.2.	Mozilla Thunderbird	12
2.4.	Mail server	12
2.5.	Protokoly.....	12
2.5.1.	SMTP	12
2.5.2.	POP	13
2.5.3.	IMAP	13
2.6.	PHP.....	13
2.7.	(X)HTML + CSS.....	14
2.8.	Javascript + AJAX.....	15
2.9.	MySQL.....	15
3.	NÁVRH.....	16
3.1.	Prihlásenie.....	16
3.1.1.	Zresetovanie hesla	16
3.2.	Prezeranie emailov - HOME.....	16
3.3.	Odosielanie - SEND	17
3.4.	Plánovanie - SEND.....	17
3.5.	Prezeranie súborov - FILES.....	17
3.6.	Nastavenia - SETTINGS.....	17
3.7.	Použité knižnice / frameworky	18
3.7.1.	Dibi	18
3.7.2.	HTML_Template_IT.....	19
3.7.3.	jQuery.....	19
3.7.4.	LiveValidation.....	19
4.	REALIZÁCIA.....	20
4.1.	Návrh databázy	20
4.2.	Spojenie s emailovým serverom.....	22
4.3.	Príjem emailov	22
4.4.	Zoznam Schránok.....	23
4.5.	Príjem príloh	24
4.6.	Odosielanie emailov.....	24
4.7.	Plánovanie.....	24
4.8.	Súbory (Prílohy)	25
5.	ZÁVER.....	26
6.	POUŽITÁ LITERATÚRA	27

Zoznam obrázkov

Obrázok 1:	Schéma fungovania emailu	9
Obrázok 2:	Prostredie Gmail-u	10
Obrázok 3a:	Emailový klient Microsoft Outlook 2007	11
Obrázok 3b:	Emailový klient Mozilla Thunderbird vo verzii 3.1.10	11

1. ÚVOD

V súčasnosti je internet natoľko rozšírený, že ho možno nájsť skoro v každej domácnosti.

Cieľom mojej bakalárskej práce je navrhnúť a implementovať webovú aplikáciu rozširujúcu web mailovú službu Gmail o funkcie ktoré zlepšia s príjemnía používanie.

Aplikácia má zobrazovať emaily chronologicky na rozdiel od originálu, kde sú radené do konverzácií. Ďalej má ponúknuť možnosť plánovaného odosielania emailov a v neposlednom rade ponúkne správu príloh.

2. PREHLAD PROBLEMATIKY

2.1.Email

Email(E-mail) [1] je správa v elektronickej podobe. Elektronická pošta je v súčasnej dobe veľmi rozšírený spôsob komunikácie. Email je starší než samotný internet. Neskôr po rozmachu internetu sa stal email jeho nepostrádateľnou súčasťou.

V začiatkoch museli emailové adresy obsahovať kompletnú sieťovú cestu, mená počítačov, od odosielateľa až po prijímateľa. Jednotlivé mená počítačov boli oddelené „bang“ znakom, t.j. „!“ . Vzor:

Adresa: stroj1!stroj2!stroj3!stroj4!jozo

Výklad: email poputuje k počítaču stroj1, od neho ďalej ku počítaču stroj2 až po počítač s názvom stroj4, a email bude doručený užívateľovi jozo.

Neskôr v roku 1972 zaviedol Ray Tomlinson znak „@“ na oddelenie mena užívateľa a mena stroja. Ale adresa sa zmenila len čiastočne, keďže počítače nepoznali informácie o polohe všetkých „počítačoch“ (serveroch) ale iba pár susediacich. Vzor:

Adresa: stroj1!stroj2!stroj3!jozo@stroj4

V dnešnej dobe máme DNS (Domain Name Server) servery ktoré obsahujú informácie o „polohách“ (adresách) „počítačov“ tak sa adresa skrátila na tvar:
jozo@stroj4

2.1.1. Formát emailu

Je definovaný v RFC2822.

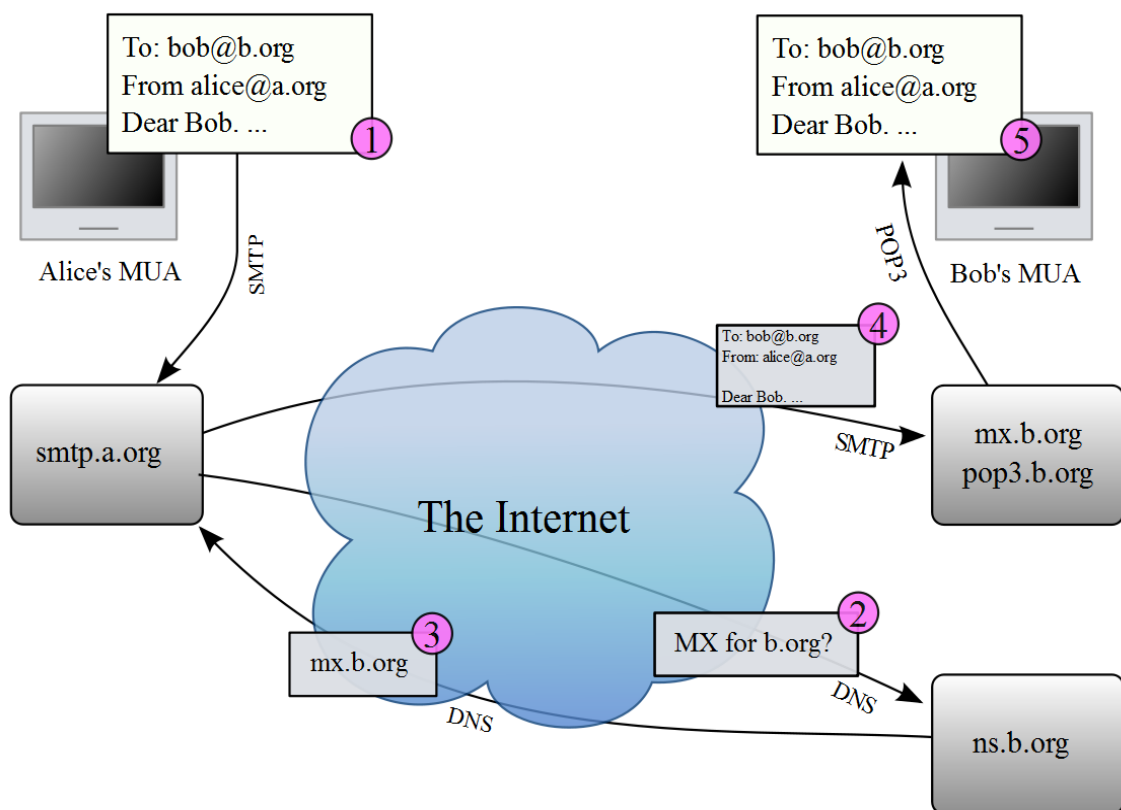
Skladajú sa z dvoch hlavných častí:

- Hlavička (header) - Obsahuje niekoľko polí. Najdôležitejšími pre užívateľa sú: Od (From), Komu (For), Predmet (Subject), Dátum (Date), alebo menej dôležitými a nepovinnými Kópia (Cc – Carbon copy), Skrytá kópia (Bcc – Blind carbon copy) a iné. Pre programátora a správny beh klienta sú medzi inými potrebné aj polia: Typ Obsahu (Content-type) a Typ De\Kódovania (Encoding).
- Telo (body) - Je samotná správa, text ktorý odosielateľ napíše.

2.2. Ako Email funguje

Povedzme že Andrej (andrej.bihun@gmail.com) chce poslať email Janke (jana.XYZ@zoznam.sk).

Andrej pomocou Emailový klient napíše email, ktorý odošle. Následne sa tento email odošle pomocou SMTP Mail server Googlu (smtp.google.com). Mailový server správu prijme, spracuje a odošle. Spracovanie prebieha tak že rozdelí emailovú adresu na 2 časti. Časť pred (miestna časť) a za (vzdialená časť) „@“. Druhú časť vyhľadá v DNS a následne pošle správu serveru zoznam.sk kde zoznam.sk prepošle email užívateľovi jana.XYZ. Používateľ si email stiahne emailovým klientom buď pomocou SMTP alebo IMAP.



Obrázok 1 [1]: Schéma fungovania emailu

2.3. Emailový klient

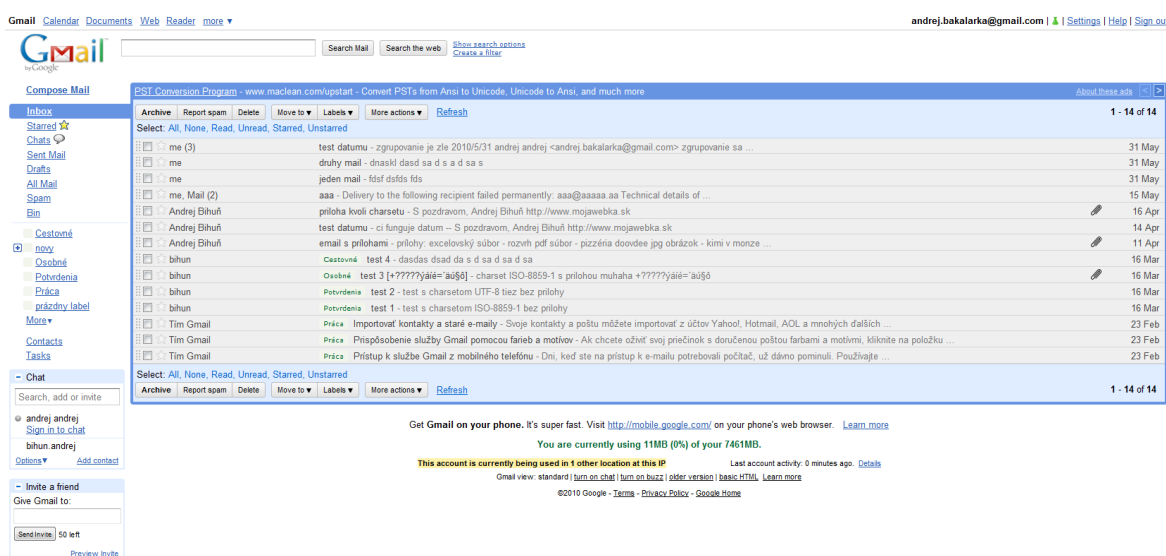
Mail User Agent (MUA) [2] je program (aplikácia) určený na správu emailov, ich odosielanie a prijímanie. Najznámejšou aplikáciou je Microsoft Outlook, lebo je integrovaný vo Windowse. Ďalšími známymi sú napríklad Mozilla Thunderbird, Lotus Notes, The Bat! a iné.

2.3.1. Web-based

Webová aplikácia určená na správu emailov inak nazývaná aj webmail. Na rozdiel od bežných desktopových aplikácií je prístupná z ktoréhokoľvek miesta kde máme prístup na internet. Príklady webmailov: Gmail(gmail.com, mail.google.com), Zoznam(mail.zoznam.sk), Yahoo(mail.yahoo.com) a iné webmailové služby.

2.3.1.1. Gmail

Je bezplatná emailová služba poskytovaná spoločnosťou Google. Je prístupná cez webové rozhranie alebo pomocou protokolov.



Obrázok 2: Prostredie Gmail-u

2.3.1.1.1. Gmail Labs

Sú pomocné, rozširujúce balíčky ktoré dovoľujú testovanie nových funkcií z dielne googlu. Jedným z balíčkov je napríklad Nested Labels (Hierarchické záložky) ktorý dovoľuje organizáciu štítkov (schránok) do stromovej hierarchickej štruktúry. Užívateľia si môžu tieto balíky kedykoľvek aktivovať a deaktivovať. Nevýhodou je uzavretosť skupiny programátorov, a nemožnosť si naprogramovať svoj vlastný balíček prípadne upraviť existujúci.

2.3.1.1.2. Better Gmail

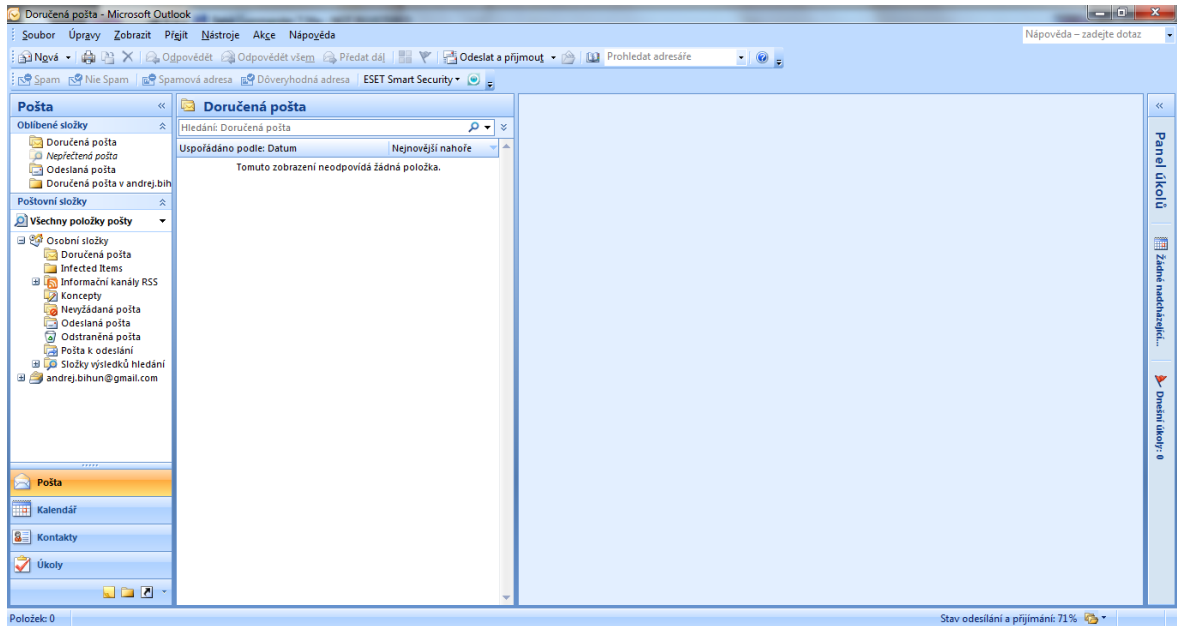
Je balíček užívateľských skriptov napísaných v javascripte pre Greasemonkey.

Greasemonkey je rozšírenie (doplnok) do Firefoxu, ktorý umožňuje vďaka užívateľským scriptom prispôbiť vzhľad a chovanie zobrazovanej webovej stránky.

Nevýhodou je naviazanosť na prehliadač Firefox.

2.3.2. Desktop applications

Desktopových (lokálnych) aplikácií je množstvo. Ja uvádzam 2 príklady, ktoré sú podľa mojich dosavadných skúseností najpoužívanejšie na systémoch Windows.



Obrázok 3a: Emailový klient Microsoft Outlook 2007



Obrázok 3b: Emailový klient Mozilla Thunderbird vo verzii 3.1.10

2.3.2.1. Microsoft Outlook

Je aplikácia na správu e-mailov, obsiahnutá v balíku Microsoft Office. Má príjemné prostredie a je používaná vo väčšine firmách na Slovensku, keďže najbežnejším operačným systémom je Windows a neoddeliteľnou súčasťou je kancelársky balík Microsoft Office.

2.3.2.2. Mozilla Thunderbird

Je bezplatná aplikácia s jednoduchým prostredím a bezproblémovou manipuláciou aj pre menej skúseného používateľa. Podporuje všetky e-mailové protokoly, aj šifrovanie.

2.4. Mail server

Mail Transfer Agent (MTA) [3] v preklade Emailový server alebo Poštový server je aplikácia ktorá sa stará o prenos emailov medzi jednotlivými klientmi. Pracuje na pozadí, zatiaľ čo užívateľ pracuje s emailovým klientom. Mail server pracuje aj ako klient (odosielanie emailov) aj ako server (prijímanie emailov).

2.5. Protokoly

Vďaka protokolom IMAP, POP a SMTP je možné prijímať a odosielať emaily. Protokoly pracujú na Aplikačnej vrstve. Všetky tri spomínané protokoly využívajú TCP (Transmission Control Protocol) spojenie, ale každý na inom porte.

2.5.1. SMTP

Simple Mail Transfer Protocol (SMTP) [4] je protokol určený na odosielať emailov. Správu doručí do adresátovej schránky, kde si ju užívateľ „vzdvihne“ pomocou IMAP alebo POP. Protokol využíva port 25.

Extended Simple Mail Transfer Protocol (ESMTP) [5] rozširuje SMTP o nové príkazy.

Napríklad umožňuje používať 8-bitové kódovanie, šifrované spojenie pomocou TLS, posielat' viac príkazov počas jedného spojenia a ďalšie.

2.5.2. POP

Post Office Protocol (POP) [6], najnovšia verzia je 3 (POP3), umožňuje sťahovanie emailov zo serveru. Pôvodne bol tento protokol navrhnutý tak, aby stiahol všetky emaily zo serveru, uložil v klientovi, a zo servera zmazal, čo však v nových verziách ide obísť a nechať správy na servery. Na zložitejšie úkony tu máme IMAP.

Nevýhodou je že sa sťahujú aj správy, ktoré boli označené serverom ako SPAM (nevyžiadaná pošta).

Ďalšou nevýhodou je schopnosť synchronizácie len s jedným klientom. Ako vidno na obrázku nižšie, ak máme 2 a viac zariadení, a v jednom klientovi spustíme sťahovanie správ, na ostatných zariadeniach už tieto správy nie je možné stiahnuť.

Na spojenie využíva port 110, v prípade šifrovaného spojenia port číslo 995.

2.5.3. IMAP

Internet Message Access Protocol (IMAP) [7] umožňuje „online“ prístup k emailom a manipuláciu priamo na serveri. V súčasnosti je najnovšia IMAP verzia 4 vydanie 1.

Má podporu viacerých funkcií než POP, medzi ktoré patrí napríklad vytváranie, premenovávanie a mazanie schránok, nastavovanie a mazanie príznakov.

Synchronizácia správ je možná aj na viac ako jednom zariadení, tzn. že môžem prijať správu napríklad na telefóne, aj na počítači, a v oboch sa mi ukáže ten istý obsah (na rozdiel od POP) a v prípade že na jednom zo zariadení prečítam správu, klient správu označí ako prečítanú (Seen), na druhom sa ukáže správa tiež ako prečítaná.

Pri nešifrovanom spojení využíva port 143, pri šifrovanom 993.

2.6.PHP

Je skriptovací jazyk slúžiaci na programovanie webových aplikácií a vytváranie dynamických webových stránok. PHP spracováva kód na strane servera, na rozdiel od javascriptu. Spolupracuje s relačnými databázami ako MySQL, Oracle, PostgreSQL, SQLite a ďalšími. Je multiplatformový, beží na väčšine najrozšírenejších operačných systémoch, ako napríklad Windows, Linux, Unix, Mac OS X.

Syntax jazyka je podobná jazykom ako Java, C, Pascal, Perl.

2.7.(X)HTML + CSS

HTML (HyperText Markup Language) je aplikáciou SGML.

XHTML (Extensible HyperText Markup Language) je aplikáciou XML (podmnožina SGML). Je podobný HTML ale prihladá sa tu viac na syntax.

Je značkový jazyk, pomocou ktorého sa vytvárajú webové stránky. Jazyk je množinou elementov (značiek, tagov) a ich atribútov. Názvy elementov sa uzatvárajú do ostrých zátvoriek „<“ a „>“. Poznáme párové a nepárové elementy.

Párové: , , <html></html>, <body></body>

Nepárové:

HTML -
, <hr>,

XHTML -
, , <script />

XHTML 1.0 a HTML 4.01 sa od seba nelíšia. Rozdiely prišli až s verziou XHTML 1.1 kde boli odstránené niektoré elementy.

Základná syntax:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Title of webpage</title>
  </head>
  <body>
    <p>Content ...</p>
  </body>
</html>
```

CSS (Cascading Style Sheets) je rozšírenie (X)HTML. Je to jednoduchý mechanizmus na vizualizáciu webových dokumentov. CSS umožňuje oddelenie štruktúry (obsahu) stránky od vzhľadu čo sprehľadní a zjednoduší kód. CSS je tiež možné oddeliť do externého súboru. V prípade externého súboru, sa CSS pripája cez element <link type="text/css" href="path to css file" /> a vpisuje sa do elementu <head>. V opačnom prípade sa CSS zapisuje do elementu <style type="text/css"></style> kdekoľvek v elemente <html></html>

Syntax jazyka má svoje pravidlá ktoré je potrebné dodržiavať.

Príklad:

```
body {
  background-color: white;
  color: black;
  padding: 10px !important;
}
```


Kde:

- Selektor, napríklad body, li, ul, div ...
- „{“ a „}“ obaľujú blok s deklaráciami
- deklarácie sú oddelované bodkočiarkou - „;“
 - o identifikátor – napríklad color, background-color a iné
 - o hodnota – white, #FFFFFF, underline a iné

2.8.Javascript + AJAX

JavaScript

Je skriptovací, objektovo orientovaný, jazyk používaný na tvorbu dynamického obsahu webových stránok, animácie, rôznych efektov a veľa iného.

Pôvodne sa rávalo s názvom Mocha, neskôr LiveScript. Syntax je podobná ako Java/C/C++.

Kód napísaný v JavaScripte sa vykonáva až po úplnom načítaní webovej stránky na strane klienta.

JavaScript je netyповý jazyk, to znamená že sa nedefinuje typ premennej, len sa prideli hodnota, ktorá môže byť raz String, neskôr Integer, následne Boolean, atď.

AJAX (Asynchronous JavaScript and XML)

- je vhodný na vytváranie rýchlych a dynamických stránok.
- umožňuje asynchrónnu zmenu obsahu stránky na pozadí, bez nutnosti znovu načítavania stránky.
- nie je sám o sebe programovacím jazykom, ale je kombináciou DOM + JavaScript, XML a http protokolom (napríklad XMLHttpRequest)

2.9.MySQL

„MySQL je slobodný a otvorený viacvláknový, viacúčivateľský SQL relačný databázový server. MySQL je podporovaný na viacerých platformách (ako Linux, Windows či Solaris a je implementovaný vo viacerých programovacích jazykoch ako PHP, C++ či Perl. Databázový systém je relačný typu DBMS (database management system). Každá

databáza je v MySQL tvorená z jednej alebo z viacerých tabuliek, ktoré majú riadky a stĺpce. V riadkoch sa rozoznávajú jednotlivé záznamy, stĺpce udávajú dátový typ jednotlivých záznamov, pracuje sa s nimi ako s poľami. Práca s MySQL databázou je vykonávaná pomocou takzvaných dotazov, ktoré vychádzajú z programovacieho jazyka SQL (Structured Query Language).“

Zdroj: wikipedia (<http://sk.wikipedia.org/wiki/MySQL>)

3. NÁVRH

Aplikácia je tvorená len z jednej časti, a to z užívateľskej. Užívateľ si totiž inštaluje aplikáciu na svoj vlastný server, a bude jediný kto k nej bude mať po nainštalovaní prístup.

Aplikácia pozostáva zo štyroch hlavných častí:

- HOME
- SEND
- FILES
- SETTINGS

3.1.Prihlásenie

Pri vstupe na stránku, je používateľ vyzvaný prihlásiť sa pomocou hesla ktoré zadal pri inštalácii aplikácie. Po úspešnom prihlásení je možné plne využívať aplikáciu.

3.1.1. Zresetovanie hesla

V prípade že užívateľ zabudne heslo do aplikácie, je možné zresetovať heslo. Na emailovú adresu zadanú pri registrácii sa odošle email s URL odkazom na zadanie nového hesla.

3.2.Prezeranie emailov - HOME

Jedným z dôvodov pre ktorý bola táto aplikácia vytvorená je práve prezeranie emailov. Google totiž vo svojej aplikácii Gmail radí emaily do rozhovorov, to jest, emaily

s rovnakým predmetom zoskupuje. Táto funkcia môže byť užitočná, ale občas užívateľ potrebuje vidieť emaily presne v takom poradí v akom boli prijaté.

Emaily sú teda radené podľa dátumu od najnovších po najstaršie. V emailoch možno aj vyhľadávať. Ďalej je možné prepínať medzi účtami, ak ich má užívateľ viac.

3.3.Odosielanie - SEND

Pôvodnou funkciou malo byť aj hromadné odosielanie emailov, ale Gmail zmenil politiku a povolil odosielanie až 500 adresátom naraz. Tak som sa rozhodol implementovať obyčajné odosielanie. Samozrejmosťou je možnosť posilať aj prílohy.

3.4.Plánovanie - SEND

Plánovanie je súčasťou odosielania súborov. Užívateľ napíše požadovaný email, aktivuje plánovanie, zadá potrebné dáta ako deň, mesiac, rok atď. a dokončí proces odoslaním formulára. Následne sa do databázy uložia odoslané dáta a v prípade že boli zadané aj prílohy, prenesie súbory na server, ktoré sa potom odošlú spolu s emailom v danom termíne.

3.5.Prezeranie súborov - FILES

Ešte som sa nestretol s aplikáciou, ktorá by poskytla užívateľovi zoznam všetkých príloh s možnosťou ich triedenia a vyhľadávania v nich. Tak som sa rozhodol realizovať to. Viac v realizácii.

3.6.Nastavenia - SETTINGS

Accounts - Účty

Tu užívateľ spravuje Gmail kontá, na ktoré sa aplikácia pripája. Je možné kontá pridávať, upravovať a mazať.

Emails - Emaily

Sekcia určená pre zmenu nastavení pre emaily. Nastavujeme tu počet zobrazovaných emailov na stránku, a počet zobrazovaných znakov obsahu emailu v zozname.

System - Systém

Sekcia určená pre zmenu systémových nastavení, ktoré určujú chovanie aplikácie. Nastavujeme adresu servera na ktorom je aplikácia nainštalovaná a imap-ovské konštanty pre spojenie so serverom (hostname, port, flags)

3.7.Použité knižnice / frameworky

3.7.1. Dibi

Dibi je knižnica na komunikáciu s databázovým serverom. Podporuje 8 rôznych databáz: MySQL, PostgreSQL, SQLite, MS SQL, Oracle, Access and generic PDO and ODBC. Má za úlohu maximálne uľahčiť prácu programátorom.

Pripojovanie:

`$connection = new DibiConnection($options);` alebo aj `dibi::connect($options);` kde `$options` je pole stringov (reťazcov) s informáciami potrebnými na pripojenie:

'driver' – typ databázy (napríklad v mojom prípade mysql)

'host' – adresa db servera, väčšinou localhost, ale môže byť aj iná

'username' – meno na prihlásenie do db

'password' – heslo na prihlásenie do db

'database' – meno databázy

Select:

```
$result = dibi::select('*')->from('table')->where('condition')  
->fetchAll() alebo ->fetch() alebo ->fetchSingle()
```

Update:

```
$result = dibi::update('table', array('key' => 'value'))->where('condition')->execute()
```

Delete:

```
$result = dibi::delete('table')->where('condition')->execute()
```

3.7.2. HTML_Template_IT

Je template-ovací systém (templating engine) z dielne PEAR (PHP Extension and Application Repository) ktorá je v novších verziách PHP automaticky implementovaná.

Je navrhnutý aby oddeľoval PHP kód od „prezentačného“ kódu. Prezentačným kódom zväčša rozumieme HTML alebo XML, ale môžu byť aj iné.

3.7.3. jQuery

Je knižnica rozširujúca Javascript ktorá zjednodušuje prácu s udalosťami, manipuláciou a animáciami v html dokumente.

zobrazovanie/skrývanie: `$('#id_bloku').show(); $('#id_bloku').hide();`
`$('#id_bloku').toggle();`

prepísanie bloku: `$('#id_bloku').html('hodnota');`

získanie hodnoty atribútu: `$('#id_bloku').attr("hodnota");`

získanie hodnoty vlastnosti štýlu: `$('#id_bloku').css('hodnota');`

3.7.4. LiveValidation

Ďalšia knižnica rozširujúca Javascript, validujúca formulár.

```
function validate(){
    var email_to = new LiveValidation( 'emailto' );
[1]    email_to.add( Validate.Presence );
    .....
    if(LiveValidation.massValidate( [ email_to, email_subject, email_text ])){
        if($('#sendaction').attr('value') == 'schedule'){
            var day = new LiveValidation( 'pday' );
[2]    day.add( Validate.Exclusion, { within: [ 'X' ] } );
            .....
            if(LiveValidation.massValidate( [ day, month, year, hour, minute ])){
                $('#allok').attr('value', 'ok');
                my_submit();
            }
            } else {
                $('#allok').attr('value', 'ok');
                my_submit();
            }
        }
    }
}
```

Použitie pri odosielaní emailu. Validácia prázdneho políčka [1] – pole nemôže ostať prázdne a určitej hodnoty [2] – pole nemôže obsahovať znak X

4. REALIZÁCIA

4.1.Návrh databázy

V databáze sa nachádzajú nasledovné tabuľky:

- Accounts
- Attachments
- Emails
- Files
- Synchronized
- System_settings
- Tasks

Niektoré tabuľky tvoria celky:

- Planner – Attachments, Emails, Tasks
- Files – Files, Synchronized

Accounts:

aid	int(2)		ID konta - PRIMARY
name	varchar(20)	utf8_general_ci	alias pre konto
email	varchar(60)	utf8_general_ci	emailová adresa
password	varchar(60)	utf8_general_ci	heslo
position	int(2)		(nevyužité)

Tasks:

id	int(8)		ID úlohy - PRIMARY
email_id	int(8)		ID konta pre úlohu
datetime	int(12)		dátum vykonania
executed	enum('y','n')	utf8_general_ci	stav spracovania

Emails:

id	int(8)		ID emailu - PRIMARY
from	int(8)		ID konta pre úlohu
to	text	utf8_general_ci	príjemcovia
cc	text	utf8_general_ci	kópia
bcc	text	utf8_general_ci	skrytá kópia
subject	varchar(255)	utf8_general_ci	predmet
message	text	utf8_general_ci	obsah správy
attachments	enum('y','n')	utf8_general_ci	prílohy

Attachments:

id	int(8)		ID prílohy - PRIMARY
email_id	int(8)		ID konta pre úlohu
filename	varchar(255)	utf8_general_ci	názov súboru
filepath	varchar(255)	utf8_general_ci	cesta k súboru
filetype	varchar(50)	utf8_general_ci	typ súboru

Files:

m_id	varchar(150)	utf8_general_ci	unikátny kľúč emailu
aid	int(3)		ID prílohy
filename	varchar(150)	utf8_general_ci	názov súboru
subtype	varchar(100)	utf8_general_ci	typ súboru

Synchronized:

m_no	int(9)		poradové číslo emailu
acc_id	int(3)		ID konta
m_id	varchar(150)	utf8_general_ci	unikátny kľúč emailu
sync_date	int(12)		dátum synchronizácie
m_date	int(12)		dátum prijatia emailu
from	varchar(100)	utf8_general_ci	odosielateľ

System settings:

name	varchar(255)	utf8_general_ci	názov
value	varchar(255)	utf8_general_ci	hodnota

name	value
emails_per_page	20
email_flags	/ssl
email_hostname	imap.gmail.com
email_port	993
email_preview_length	200
files_last_sync	-1
files_synced_at_once	20
files_synchronizing	false
login_attempts	0
login_restriction	0
main_acc	4
password	a3f160ef80b5276ef5575838b7c78e3a
password_reset	-
server_hosturl	http://localhost/bc/
synchronize_at_start	no

4.2.Spojenie s emailovým serverom

Pre spojenie so serverom používam funkciu `imap_open(mailbox, user, pass)` ktorá je už implementovaná v php. Má 3 povinné parametre skladajúce sa z:

- mailbox v tvare `{email_server_url:port}/mailbox_path` v našom prípade je to tvar `{imap.gmail.com: 993/ssl}/mailbox_path`
mailbox_path môže byť napríklad INBOX, SPAM, SENT, ...
- user – prihlasovacie meno
- pass – prihlasovacie heslo

Pre ukončenie spojenia sa používa `imap_close()`.

4.3.Príjem emailov

Príjem správ zo servera zabezpečujú viaceré funkcie. Každá z nich vracia špecifické informácie, pre lepšiu manipuláciu s danou správou, aby sme zbytočne nezhrmažďovali informácie ktoré práve nepotrebujeme. Jednou z nich je aj `imap_headerinfo(connection, index)`, ktorú som použil na získanie informácií pre zoznam emailov a ktorá vracia nasledovné: (príklad)

```
["date"]=> string(31) "Wed, 16 Mar 2005 09:53:03 -0800"
["Date"]=> string(31) "Wed, 16 Mar 2005 09:53:03 -0800"
["subject"]=> string(49) "Gmail is different. Here's what you need to know."
["Subject"]=> string(49) "Gmail is different. Here's what you need to know."
["message_id"]=> string(43) ""
["toaddress"]=> string(56) "=?ISO-8859-2?Q?Andrej_Bihu=F2?="
["to"]=> array(1) {
  [0]=> object(stdClass)#6 (3) {
    ["personal"]=> string(31) "=?ISO-8859-2?Q?Andrej_Bihu=F2?="
    ["mailbox"]=> string(12) "bihun.andrej"
    ["host"]=> string(9) "gmail.com"
  }
}
["fromaddress"]=> string(37) "Gmail Team "
["from"]=> array(1) {
  [0]=> object(stdClass)#5 (3) {
    ["personal"]=> string(10) "Gmail Team"
    ["mailbox"]=> string(13) "gmail-noreply"
    ["host"]=> string(10) "google.com"
  }
}
["reply_toaddress"]=> string(37) "Gmail Team "
["reply_to"]=> array(1) {
  [0]=> object(stdClass)#8 (3) {
    ["personal"]=> string(10) "Gmail Team"
    ["mailbox"]=> string(13) "gmail-noreply"
    ["host"]=> string(10) "google.com"
  }
}
```



```

["senderaddress"]=> string(37) "Gmail Team "
["sender"]=> array(1) {
  [0]=> object(stdClass)#9 (3) {
    ["personal"]=> string(10) "Gmail Team"
    ["mailbox"]=> string(13) "gmail-noreply"
    ["host"]=> string(10) "google.com"
  }
}
["Recent"]=> string(1) " "
["Unseen"]=> string(1) " "
["Flagged"]=> string(1) " "
["Answered"]=> string(1) " "
["Deleted"]=> string(1) " "
["Draft"]=> string(1) " "
["Msgno"]=> string(4) " 1"
["MailDate"]=> string(26) "16-Mar-2005 17:53:03 +0000"
["Size"]=> string(4) "4529"
["update"]=> int(1110995583)

```

Z toho pre mňa dôležité sú:

- Date, ktorý som pomocou strtotime() zkonvertoval do unixového tvaru
- Subject
- from – kde je potrebné zisťovať typ kódovania, a správne daný string dekódovať vďaka funkcii iconv()

Na stránke zobrazujem len určitý počet správ, podľa nastavenia, pre časovú náročnosť danej funkcie, aby v prípade väčšieho množstva správ nedošlo k prekročeniu limitu spracovávania php skriptu ktorý je bežne nastavený na 60s.

Pre správne zobrazovanie emailov chronologicky, je potrebné použiť ďalšiu funkciu imap_sort() pomocou ktorej získam pole indexov, pre môj prípad zoradené podľa dátumu príchodu. Potom mi stačí pre, napríklad, najnovší email volať správu s číslom na poslednom (alebo prvom, podľa nastavenia) indexe.

Samotný obsah správy získavam až neskôr, po kliknutí na požadovanú správu, pomocou AJAXu za pomoci imap_fetchbody(). Kde si ale musím dávať pozor na typ správy (HTML, ALTERNATIVE, ...) a kódovanie obsahu, ktoré zistím z imap_fetchstructure().

4.4.Zoznam Schránok

Pri získavaní správ zo servera sa zadáva aj takzvaná schránka (pričínok), v ktorej práve pracuje a sťahuje emaily iba odtiaľ. Na to nám slúži funkcia imap_list() ktorá vráti pole s názvami schránok, ktoré je ale treba ešte správne dekódovať. Pre odľahčenie spojenia s imap serverom, si ukladám schránky do SESSION.

4.5. Príjem príloh

Email obsahuje prílohu v prípade že `imap_fetchstructure()` vráti *type* rovný 1 a *subtype* rovný „MIXED“. Potom sa v *parts* dopátram názvu súboru, jeho kódovania a typu.

Pomocou `imap_fetchbody()` stiahnem prílohu zo serveru (samozrejme vo forme textu) ktorú následne dekódujem a vypíšem špeciálnym spôsobom tak aby prehliadač rozumel a vytvoril súbor ktorý následne stiahnem.

4.6. Odosielanie emailov

Odosielanie je realizované pomocou frameworku PHPMailer.

```
$mail = new PHPMailer();  
$mail->CharSet = "utf-8";  
$mail->IsSMTP();  
$mail->SMTPAuth = true;  
$mail->SMTPSecure = 'tls';  
$mail->Host = "smtp.gmail.com";  
$mail->Port = 587;  
$mail->Encoding = 'quoted-printable';  
$mail->IsHTML(true);  
$mail->Username a $mail->Password
```

Najprv vytvorím novú inštanciu triedy, a následne nastavím: charset na utf8, zvolím si smtp spôsob, nastavím autentifikáciu, zabezpečenie na tls, host na smtp.gmail.com, port je v prípade zabezpečeného spojenia 587, encoding, účet z ktorého sa bude email odosielať a nakoniec pridám informácie z vyplneného formulára ako adresu prijímateľa, prílohy, obsah emailu a iné.

4.7. Plánovanie

Je nadstavbou pre odosielanie emailov. Ponúka možnosť neodoslať email hneď ale uložiť ho do databázy, a poslať v určitý deň/hodinu. Pre plnú funkčnosť je potrebné mať na serveri CRON alebo iný daemon. CRON je plánovač pre unix, ktorý beží na pozadí, a spúšťa skripty podľa tabuľky ktorú má. A práve tu treba nastaviť aby sa v nejakých pravidelných intervaloch spúšťal skript ktorý sa stará o odosielanie emailov, a tým je `mycron.php` ktorý je umiestnený v root-e projektu.

Je možné si dané naplánované činnosti aj prezerať, editovať prípadne mazať. Ďalej je možné aj pridať k úlohe aj prílohy. Tie sa po skončení zo servera zmažú.

4.8.Súbory (Prílohy)

Táto časť je zameraná na správu príloh z emailovej schránky. V prvom rade ale treba najprv daný účet zosynchronizovať s vlastnou databázou, keďže získanie potrebných údajov pre výpis všetkých príloh zo schránky je veľmi časovo náročné. Získanie informácií jedného emailu činí v priemere 3 sekundy, v prípade pomalšieho pripojenia aj dlhšie, z čoho vyplýva že na 60 sekúnd, čo je bežný povolený čas spracovania php, pripadá získanie informácií ani nie 20 emailov.

Preto som sa rozhodol riešiť synchronizáciu cez AJAX. Ale aj tu som narazil na problém. Keďže je tu potreba zakaždým otvárať nové spojenie na email server, google po približne 200 až 300 pokusoch na chvíľu zablokuje prístup s hláškou, že bolo vykonaných veľmi veľa prihlásení.

S databázou synchronizujem cez 2 tabuľky – files a synchronized, kde do prvej menovanej vkladám jednotlivé súbory, a do druhej každý email ktorý zkontrolujem. Tabuľky sú spojené cez index m_id, kde každý email má tento kľúč unikátny.

Súbory synchronizuje od najstaršieho po najnovšie. Synchronizujú sa len údaje o prílohách, nie samotné súbory. Súbory sú neskôr sťahované priamo zo schránky. Synchronizované údaje sa používajú čisto na vyhľadávanie a radenie.

V súboroch je možné vyhľadávať podľa odosielateľa, dátumu, názvu a typu. Taktiež je možné zoraďovať ich podľa názvu, dátumu a typu.

5. ZÁVER

Cieľom bakalárskej práce bolo vytvoriť aplikáciu, ktorá rozšíri funkcie web mailovej služby Gmail o chronologické radenie emailov, plánované odosielanie emailov, a správu príloh emailov.

Prvý cieľ je asi najmenej podstatný, keďže Google pár mesiacov po zadaní tejto práce implementoval túto funkcionálnosť.

Druhý a tretí cieľ sa mi podarilo splniť, odosielanie funguje, len je potrebné mať na servere CRON, ktorý bude spúšťať skript v pravidelných intervaloch a kontrolovať udalosti. Pri tretej časti som mal najviac problémov. Spôsobilo to indexovanie emailov zo strany servera, a dalo mi veľa času prísť na to ako zistiť index emailu v prítomnom čase a čase keď bol synchronizovaný, v prípade že bol nejaký email zmazaný, presunutý, prípadne niečo iné.

Vidím tu určité oblasti ktoré by sa dali v budúcnosti vylepšovať a rozširovať, ako napríklad lepšiu správu súborov, použitie JS na zgrupovanie súborov, vyhľadávanie vo viacerých zložkách, nielen v jednej, prípadne možnosť zálohovať určité súbory a emaily na vlastný server.

6. POUŽITÁ LITERATÚRA

- [1] Email: (eng) <http://en.wikipedia.org/wiki/E-mail>
- [2] MUA – Emailový klient: (eng) http://en.wikipedia.org/wiki/Email_client
- [3] MTA – Emailový server: (eng) http://en.wikipedia.org/wiki/Message_transfer_agent
- [4] SMTP: (eng) http://en.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol
- [4] ESMTP: (eng) http://en.wikipedia.org/wiki/Extended_SMTp
- [6] POP: (eng) http://en.wikipedia.org/wiki/Post_Office_Protocol
- [7] IMAP: (eng) http://en.wikipedia.org/wiki/Internet_Message_Access_Protocol

Obsah CD

Aplikácia – zdrojové kódy

DB – exportovaná databáza

BC práca vo formáte doc a pdf