

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Aplikace pro sběr online dat o zpoždění vlaků osobní  
dopravy

Jaroslav Demel

Bakalářská práce

2012

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2011/2012

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jaroslav Demel**  
Osobní číslo: **I07594**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Aplikace pro sběr online dat o zpoždění vlaků osobní dopravy**  
Zadávající katedra: **Katedra informačních technologií**

### Z á s a d y p r o v y p r a c o v á n í :

Cílem bakalářské práce je navrhnout a implementovat aplikaci pro získání online dat pro vybrané železniční tratě na území České republiky s tím, že tato aplikace bude umožňovat zápis získaných dat do XML souborů. Dále bude aplikace umožňovat statistické vyhodnocení výsledků z dat, která budou v XML souborech k dispozici (minimální, maximální hodnoty pro jednotlivé vlaky za vybrané období, histogram s možností definice tříd apod.). V úvodu práce bude uveden přehled možností pro získání dat online a stručné představení technologie použité pro vývoj aplikace.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. KOSEK, Jiří. XML pro každého: podrobný průvodce. Praha: Grada publishing, 2000. ISBN 80-7169-860-1.
2. KUBANOVÁ Jana. Matematická statistika. Pardubice, 1999. ISBN 80-7194-215-4.

Vedoucí bakalářské práce:

**Ing. Michael Bažant, Ph.D.**  
Katedra softwarových technologií

Datum zadání bakalářské práce: **16. prosince 2011**

Termín odevzdání bakalářské práce: **11. května 2012**



prof. Ing. Simeon Karamazov, Dr.  
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 30. března 2012

## **Prohlášení autora**

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 17. 08. 2012

Jaroslav Demel

## **Poděkování**

Rád bych zde poděkoval vedoucímu mé bakalářské práce panu Ing. Michaelovi Bažantovi, Ph.D. za cenné rady a připomínky, za jeho čas, trpělivost a ochotu při řešení problémů vzniklých při zpracování této práce.

## **Anotace**

Tato bakalářská práce se zabývá dvěma částmi, serverovou a klientskou. Serverová část řeší návrh informačního systému, který produkuje fiktivní data o zpoždění vlaků. Je zde popsána základní charakteristika systému, chod, a struktura databáze. Klientská část se zabývá implementací a architekturou widgetu pro prohlížeč Opera. Tento widget stahuje data o zpoždění vlaků ze serveru a umožňuje statistické vyhodnocení z těchto dat.

## **Klíčová slova**

Zpoždění vlaku, informační systém, statistika zpoždění vlaků, serverová část informačního systému, klientská část informačního systému

## **Title**

Application for collecting online data on passenger train delays.

## **Annotation**

This bachelor work deals two parts, server and client. Server part addresses the design of information system, which produces fictitious data on delay of trains. It describes the basic characteristics of the system operation, and database structure. Client part deals with implementation and architecture widget for Opera browser. This widget downloads data about train delay from the server and allows the statistical evaluation of these data.

## **Keywords**

Train delays, the information system, statistics train delays, the server portion of the information system, the client portion of the information system

## Obsah

|   |           |
|---|-----------|
| <b>Seznam zkratk</b> .....                            | <b>8</b>  |
| <b>Seznam obrázků</b> .....                           | <b>8</b>  |
| <b>Seznam tabulek</b> .....                           | <b>9</b>  |
| <b>1 Úvod</b> .....                                   | <b>10</b> |
| <b>2 Přehled možností získání online dat</b> .....    | <b>11</b> |
| 2.1 Obecně.....                                       | 11        |
| 2.2 Způsoby získávání dat .....                       | 12        |
| 2.3 Zpoždění vlaků osobní dopravy, analýza webů ..... | 13        |
| 2.3.1 Web - Česká železnice.....                      | 13        |
| 2.3.2 Web - Dopravní web.....                         | 14        |
| 2.3.3 Web - Babitron .....                            | 15        |
| 2.3.4 Web - České dráhy .....                         | 16        |
| 2.3.5 Web - Poloha vlaku .....                        | 17        |
| 2.3.6 Srovnání vybraných webů .....                   | 18        |
| 2.4 Závěr.....  | 19        |
| <b>3 Použité technologie</b> .....                    | <b>20</b> |
| 3.1 HTML, XHTML, XML.....                             | 20        |
| 3.1.1 HTML.....                                       | 20        |
| 3.1.2 XHTML .....                                     | 21        |
| 3.1.3 XML .....                                       | 22        |
| 3.2 CSS .....   | 24        |
| 3.3 JAVASCRIPT .....                                  | 25        |
| 3.4 AJAX.....   | 26        |
| 3.5 JQUERY .....                                      | 27        |
| 3.6 PHP.....  | 27        |
| 3.7 MYSQL .....                                       | 28        |
| <b>4 Serverová část</b> .....                         | <b>28</b> |
| 4.1 Základní charakteristika .....                    | 28        |
| 4.2 Politika uživatelů .....                          | 30        |
| 4.3 Rich picture diagram .....                        | 30        |
| 4.4 Adresářová struktura.....                         | 32        |

|          |   |           |
|----------|---|-----------|
| 4.5      | Struktura databáze .....  | 32        |
| 4.5.1    | E-R diagram.....  | 33        |
| 4.5.2    | Entita VLAK .....   | 34        |
| 4.5.3    | Entita VESTANICI .....  | 35        |
| 4.5.4    | Entita STANICE.....   | 35        |
| 4.5.5    | Entita ZPOZDENI .....   | 35        |
| 4.5.6    | Entita UZIVATEL.....  | 36        |
| 4.6      | Rozvržení a struktura webu .....                                  | 36        |
| 4.7      | Umístění hostingu.....  | 37        |
| <b>5</b> | <b>Klientská (aplikační) část.....</b>                            | <b>38</b> |
| 5.1      | Opera widget.....   | 39        |
| 5.2      | Packaging widgetu.....  | 40        |
| 5.3      | Adresářová struktura widgetu.....                                 | 40        |
| 5.4      | Konfigurační soubor widgetu .....                                 | 41        |
| 5.5      | Widget a AJAX .....   | 42        |
| 5.6      | Widget security model.....  | 42        |
| 5.6.1    | Element security .....  | 43        |
| 5.7      | Widget File I/O.....  | 44        |
| 5.7.1    | Povolení File I/O .....   | 45        |
| 5.7.2    | Mount points.....   | 45        |
| 5.8      | Architektura widgetu .....  | 46        |
| 5.8.1    | Postup stažení dat aktuálně jedoucích vlaků .....                 | 48        |
| 5.8.2    | Postup stažení dat konkrétních vlaků .....                        | 48        |
| 5.8.3    | Struktura stažených dat ze serveru .....                          | 48        |
| 5.9      | Statistické vyhodnocení výsledků .....                            | 48        |
| 5.9.1    | Histogram .....   | 49        |
| <b>6</b> | <b>Závěr.....</b>   | <b>50</b> |
|          | <b>Literatura .....</b>   | <b>51</b> |
|          | <b>Příloha A – Ukázka webu (serverová část) .....</b>             | <b>55</b> |
|          | <b>Příloha B – Ukázka generátoru dat (serverová část) .....</b>   | <b>56</b> |
|          | <b>Příloha C – Ukázka widgetu pro Operu (klientská část).....</b> | <b>57</b> |
|          | <b>Příloha D – Parsing XML struktury.....</b>                     | <b>58</b> |
|          | <b>Příloha E – XML struktura .....</b>                            | <b>59</b> |



## Seznam zkratk

|      |                                 |
|------|---------------------------------|
| AJAX | Asynchronous JavaScript and XML |
| CSS  | Cascading Style Sheets          |
| HTML | HyperText Markup Language       |
| IS   | Informační systém               |
| UML  | Unified Modeling Language       |
| URL  | Uniform Resource Locator        |
| WWW  | World Wide Web                  |
| XML  | Extensible Markup Language      |

## Seznam obrázků

|   |    |
|---|----|
| Obrázek 1 - Data.....   | 11 |
| Obrázek 2 - Informace.....  | 11 |
| Obrázek 3 - Obecné pojmy.....   | 12 |
| Obrázek 4 - Presentace Česká železnice.....   | 13 |
| Obrázek 5 - Presentace Dopravní web.....  | 14 |
| Obrázek 6 - Presentace Babitron.....  | 15 |
| Obrázek 7 - Presentace České dráhy.....   | 16 |
| Obrázek 8 - Presentace Poloha vlaku.....  | 17 |
| Obrázek 9 - Ukázka kódu XML.....  | 24 |
| Obrázek 10 - Ukázka kódu CSS.....   | 24 |
| Obrázek 11 – Javascript.....  | 26 |
| Obrázek 12 - Ukázka soupisu aktuálně jedoucích vlaků z webu (serverové části).....  | 29 |
| Obrázek 13 - Rich picture diagram - Schéma systému serverové části.....             | 31 |
| Obrázek 14 - Adresářová struktura serverové části.....                              | 32 |
| Obrázek 15 - E-R diagram systému.....   | 33 |
| Obrázek 16 - Ukázka výběru vlaku z funkce.php.....                                  | 37 |
| Obrázek 17 - Administrace webu vlaky.uvadi.cz.....                                  | 38 |
| Obrázek 18 - Nejpoužívanější webové prohlížeče v ČR.....                            | 39 |
| Obrázek 19 - Běh widgetu na ploše.....  | 39 |
| Obrázek 20 - Adresářová struktura widgetu.....                                      | 40 |
| Obrázek 21 - Část konfiguračního souboru.....                                       | 41 |
| Obrázek 22 - Inicializace požadavku XMLHttpRequest.....                             | 42 |
| Obrázek 23 - Chybová zpráva v konzoli opery.....                                    | 42 |
| Obrázek 24 - Povolení přístupu k síti.....  | 43 |
| Obrázek 25 - Element security.....  | 44 |
| Obrázek 26 - Element feature.....   | 45 |
| Obrázek 27 - Zápis souboru v javascriptu.....                                       | 46 |
| Obrázek 28 - Architektura widgetu - Rich picture diagram.....                       | 47 |
| Obrázek 29 - Statistika vlaku SC 30 Pendolino v období 2012/07/22 - 2012/07/23..... | 48 |

Obrázek 30 - Histogram vlaku R 440 Excelsior v období 2012/07/22 - 2012/08/08..... 49

## Seznam tabulek

|                                   |    |
|-----------------------------------|----|
| Tabulka 1 - Česká železnice ..... | 14 |
| Tabulka 2 - Dopravní web .....    | 15 |
| Tabulka 3 - Babitron.....         | 16 |
| Tabulka 4 - České dráhy .....     | 17 |
| Tabulka 5 - Poloha vlaku.....     | 18 |
| Tabulka 6 - Srovnání webů.....    | 19 |
| Tabulka 7 - Entita VLAK .....     | 34 |
| Tabulka 8 - Entita VESTANICI..... | 35 |
| Tabulka 9 - Entita STANICE .....  | 35 |
| Tabulka 10 - Entita ZPOZDENI..... | 36 |
| Tabulka 11- Entita UZIVATEL ..... | 36 |

# 1 Úvod

V dnešní době neustále rozvíjejícího se internetu a také vzrůstajícím počtem uživatelů, kteří jej používají, ať už pro zábavu či práci, roste i zájem o různé webové pomůcky, rozšíření, doplňky, mini aplikace, nebo taky lidověji řečeno „vychytávky“, které mohou mnohým lidem usnadnit spoustu času a úsilí při prohlížení stránek na webu. Těchto pomůcek je dnes již nespočetně mnoho, z nichž velká část je už součástí webových prohlížečů a spousty dalších lze snadno dohledat a stáhnout pro příslušný prohlížeč. Mezi ty nejpoužívanější patří například rychlý přístup k aktuálním informacím o počasí či snadný přístup k sociálním sítím, jako je Facebook a Twitter, nebo taky různé vyhledávací moduly, které informují o dopravní situaci či dopravním spojení ve městě.

A právě díky tomuto zájmu o leccjaké vylepšení vlastního prohlížeče se zrodil nápad navrhnout a sestavit aplikaci pro Operu, která by umožňovala nejen získávat, ale také shromažďovat a zpracovávat online data o zpoždění vlaků na území České republiky s tím, že vzhledem k velké obsáhlosti spojů v ČR budou data získávána jen na vybraných železničních tratích.

V první části mé práce vás nejprve seznámím s metodami a možnostmi, jakými lze na webu získat aktuální online data o zpoždění vlaků a jejich výhody, či zadrhele.

V druhé části budou představeny nezbytné technologie použité pro vývoj aplikace.

Další kapitola se bude zabývat serverovou částí aplikace, která bude umět generovat v reálném čase zpoždění vlaků a sestavovat jejich přehled, výpisy.

A v poslední kapitole se budu zabývat klientskou částí aplikace, která bude umět data o vlacích ze serveru stahovat a provádět základní statistické vyhodnocení z těchto dat.

## 2 Přehled možností získání online dat

Nejprve by bylo vhodné si vůbec říci co to vlastně data jsou, kde všude můžeme na ně na internetu narazit a proč právě mluvit o online datech.

### 2.1 Obecně

Data je výraz pro údaje, používané pro popis nějakého jevu nebo vlastnosti pozorovaného objektu, dají se získat měřením nebo pozorováním. Jiným způsobem se dá o datech mluvit jako o jakékoliv fyzicky (materiálně) zaznamenané znalosti (vědomosti), poznatku, zkušenosti nebo výsledku pozorování procesů, projevů, činností a prvků reálného světa (reality). Často si lze data splést s informací. [1], [3]



Obrázek 1 - Data

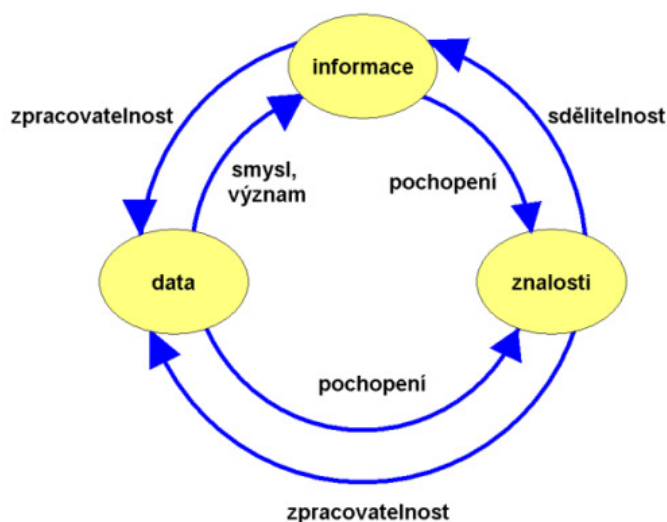
Informace je sdělení nebo zpráva, ve které se co neobjektivněji a věcně konstatují určitá fakta. Na webu se nejčastěji vyskytují v podobě různých článků, návodů, popisů, obecně lze proto říct že se jedná o jakýkoliv text členěný do odstavců, kdežto samotná data bývají většinou obsažena v tabulkách či grafech, nebo v jinak logicky členěných prvcích. [1], [2]



Obrázek 2 - Informace

Znalost je to co jednotlivec vlastní (ví) po osvojení dat a informací a po jejich začlenění do souvislostí. Jde o výsledek poznávacího procesu, předpoklad uvědomělé činnosti. [1]

Online data, jsou taková data, nebo lze je v tomto případě označit za taková, která se s postupem času mohou dynamicky měnit či přibývat. Většinou vše funguje tak, že na webové stránce jsou prezentovány v podobě tabulky, která se při prohlížení jeví staticky (nic se nemění), ale ve skutečnosti během této doby, kdy od načtení této stránky mohlo již uplynout spousta času dochází mnohdy ke změnám, které se už viditelně na stránce neprojevují. Tyto poslední změny se projeví až po opětovném znovunačtení celé stránky.



Obrázek 3 - Obecné pojmy [1]

## 2.2 Způsoby získávání dat

Jedním z nejjednodušších způsobů jak získat aktuální data je mít přístup přímo k samotným datům, která jsou na většině webových serverech uložena v podobě databází. Problémem je, že přístup k těmto databázím je velice dobře chráněný a tudíž obyčejný uživatel, který k ní nemá přístup a nezná heslo, se k ní jenom tak nedostane.

Dalším způsobem jak získat data, lze za pomoci webových služeb. Webové služby (web services) jsou vlastně systémem pro podporu součinné spolupráce počítačů v síti. V podstatě jde o komunikaci mezi 2 počítači, při níž má jeden funkci poskytovatele webové služby a druhý je klient. Poskytovatel (provider) služby poskytuje data specifikovaným způsobem na síti. Na druhé straně si klient (requestor) zjistí adresu služby (vyhledá v registru nebo má adresu přímo od poskytovatele), stáhne si popis služby a je ji následně schopen využívat. Data se přenášejí v XML formátu a díky němu jsou snadno rozšiřitelná. [40]

Třetím a méně praktickým způsobem je nechat si stránku s daty v určeném časovém intervalu opětovně generovat ze serveru a sledovat změny. Jenomže v tomto případě vždy nedostaneme přímo data, což potřebujeme, ale zakódovanou stránku v nějakém formátu, většinou v HTML, proto je nutné stránku ještě tzv. parsovat<sup>1</sup>, kdy se pomocí značkovacího jazyku propracujeme až k tabulce s údaji, kterou jsme potřebovali.

<sup>1</sup> Parsing – Český syntaktická analýza je v informatice a v lingvistice proces analýzy posloupnosti formálních prvků s cílem určit jejich gramatickou strukturu vůči předem dané (byť ne nutně explicitně vyjádřené) formální gramatice. [5]

## 2.3 Zpoždění vlaků osobní dopravy, analýza webů

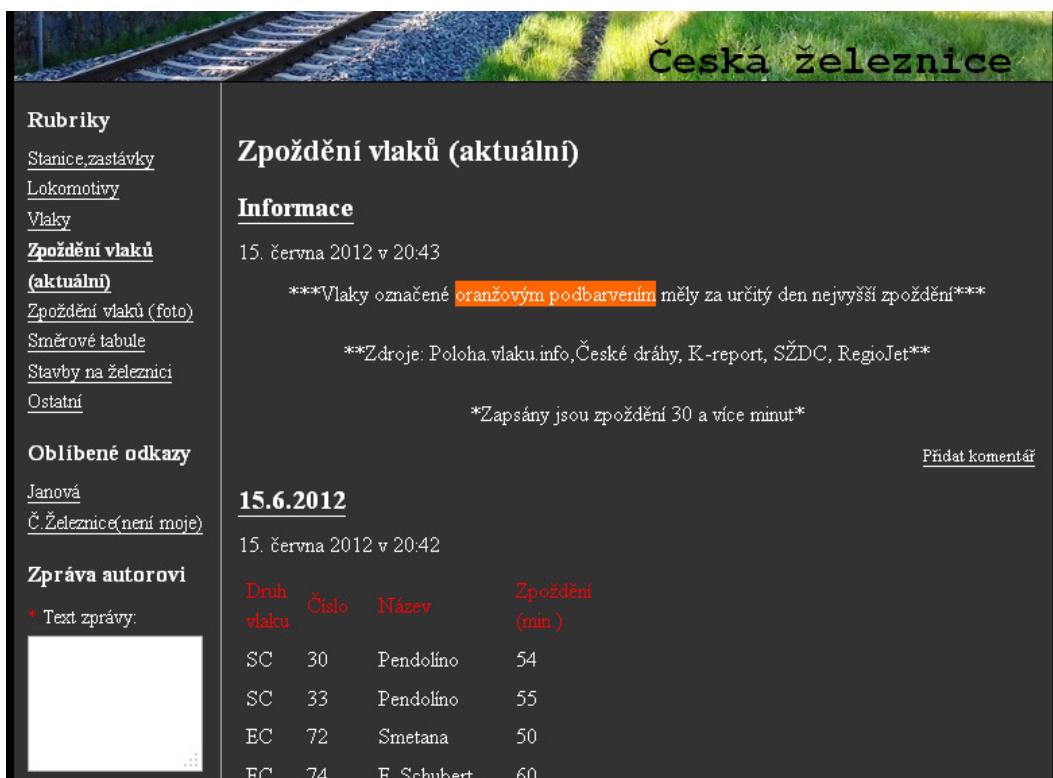
Obecně jak lze k datům přistupovat, již bylo popsáno výše, proto je třeba nyní věnovat pozornost tomu, odkud budeme data o zpoždění vlakové osobní dopravy čerpat. Při prohlídce internetu na dotaz „zpoždění vlaku“ ve vyhledávacích nástrojích, narazíme na spousty webových prezentací, které se na první pohled jeví pro čerpání dat o zpoždění jako velice vhodné pro výběr, avšak opak je pravdou. Velká většina webových stránek má data buďto neaktuální, nebo jen odkazuje na jiné stránky se skutečnými aktuálními daty, či má jiné nedostatky nevhodné pro výběr. Z tohoto důvodu je dále uveden přehled a srovnání těch nejdůležitějších stránek či webů.

### 2.3.1 Web - Česká železnice

V této kapitole jsem čerpal ze zdroje [6].

Jde o stránky věnované české železnici, které běží na webovém serveru blog.cz, což je vlastně publikační systém fungující na českém internetu. Díky tomu si může každý uživatel snadno a jednoduše vytvořit vlastní web. Z tohoto důvodu je proto jasné, že jde o amatérsky udělaný počín a proto stojí zato se pozastavit nad věrohodností publikovaných dat.

Nalezneme zde sekci aktuální zpoždění vlaků, kde jsou tabulky s daty přehledně členěny pod sebe a rozděleny podle datumu, ale při bližším prozkoumání se dozvíme, že data jsou přebírána odjinud a navíc jsou zde zveřejněny jenom ty vlaky co jsou zpožděny více jak 30 minut.



**Rubriky**

- [Stanice, zastávky](#)
- [Lokomotivy](#)
- [Vlaky](#)
- [Zpoždění vlaků \(aktuální\)](#)**
- [Zpoždění vlaků \(foto\)](#)
- [Směrové tabule](#)
- [Stavby na železnici](#)
- [Ostatní](#)

**Oblíbené odkazy**

- [Janová](#)
- [Č. Železnice \(není moje\)](#)

**Zpráva autorovi**

\* Text zprávy.

**Zpoždění vlaků (aktuální)**

**Informace**

15. června 2012 v 20:43

\*\*\*Vlaky označené **oranžovým podbarvením** měly za určitý den nejvyšší zpoždění\*\*\*

\*\*Zdroje: Poloha.vlaku.info, České dráhy, K-report, SŽDC, RegioJet\*\*

\*Zapsány jsou zpoždění 30 a více minut\*

[Přidat komentář](#)

**15.6.2012**

15. června 2012 v 20:42

| Druh vlaku | Číslo | Název       | Zpoždění (min.) |
|------------|-------|-------------|-----------------|
| SC         | 30    | Pendolino   | 54              |
| SC         | 33    | Pendolino   | 55              |
| EC         | 72    | Smetana     | 50              |
| EC         | 74    | F. Schubert | 60              |

Obrázek 4 - Prezentace Česká železnice [6]

|                     |  |
|---------------------|--|
| Adresa webu:        | ceskazeleznice.blog.cz   |
| Přímá URL k datům:  | http://ceskazeleznice.blog.cz/rubrika/zpozdzeni-vlaku-aktualni |
| Nalezeno nástrojem: | seznam.cz  |
| Datum hledání:      | 15.6.2012  |

Tabulka 1 - Česká železnice

### 2.3.2 Web - Dopravní web

V této kapitole jsem čerpal ze zdroje [7].

Dopravní web, který se zabývá železnicemi a městskou hromadnou dopravou. Stejně jako v minulém případě i tyto stránky pohání publikační internetový nástroj, který obstarává webnode.cz a umožňuje tak uživatelům bez jakýchkoliv technických znalostí vytvořit profesionálně vypadající webovou prezentaci.

Zde nemá cenu ani porovnávat obsah stránek s jinými, protože data o zpoždění vlaků vůbec neobsahuje. Je zde pouze odkaz na web babilon.

The screenshot shows the homepage of 'Dopravní web'. The main header features the title 'Dopravní web' and the subtitle 'Webovky nejen o MHD a železnici'. Below this is a navigation bar with links for 'MĚSTSKÁ HROMADNÁ DOPRAVA', 'ŽELEZNIČNÍ DOPRAVA', 'AKTUALIZACE', 'PODWEBY', and 'KONTAKT'. The current page path is 'Úvodní stránka > Železniční doprava > Aktuální zpoždění vlaků'. A search bar with the text 'HLEDAT' is visible. On the left, a sidebar menu lists: 'Tratě ČD', 'Řazení vlaků ČD', 'Aktuální zpoždění vlaků' (highlighted), 'Zpravodajství', 'Nejkrásnější nádraží České republiky', and 'Odkazy'. The main content area is titled 'Aktuální zpoždění vlaků' and lists a single item: 'Zdeněk Kamínek ml.'. Below this, a note states: 'Aktuální zpoždění vlaků (nejen z ČR) naleznete zde: <http://kam.mff.cuni.cz/~babilon/zpmapa>'.

Obrázek 5 - Prezentace Dopravní web [7]

|                     |  |
|---------------------|--|
| <b>Adresa webu:</b> | <b>dopravni.webnode.cz</b>                                     |
| Přímá URL k datům:  | http://dopravni.webnode.cz/zeleznicni/aktualni-zpozdeni-vlaku/ |
| Nalezeno nástrojem: | seznam.cz  |
| Datum hledání:      | 23.6.2012  |

Tabulka 2 - Dopravní web

### 2.3.3 Web - Babbitron

V této kapitole jsem čerpal ze zdroje [8].

Jedná se o webové stránky, které se přímo zabývají zpožděním vlaků a jejich přehledem. Web je součástí a běží na stránkách Katedry aplikované matematiky Univerzity Karlovy v Praze a mimo jiné na provozu spolupracují České dráhy, jak se lze taky na stánkách dočíst, proto není pochyb, že veškerý prezentovaný obsah je důvěryhodný.

Obsah stránky je rozdělen do více sekcí, ve kterých můžeme informace o aktuálních datech zobrazovat. Ty lze tedy zobrazit buďto na malé či velké mapě, nebo v přehledných tabulkách, kdy při poklepnání na jednotlivý vlak získáme podrobnější detaily.

## Babbitron - Zpoždění vlaků

Na provozu těchto stránek spolupracují [České dráhy a.s.](#) a Institut teoretické informatiky

**Aktuální poloha vlaků:**  
mapa [standard](#), [velká](#)  
v tabulkách  
[SC až R](#) [Sp + Os](#)

[Vysvětlivky](#)  
[Historie stránek](#)  
[Zpoždění v cizině](#)  
[Mapa tabulí SŽDC](#)

[ŽeLPage \(výluky, řazení\)](#)  
[ČD \(výluky, poloha vlaku\)](#)  
[Železniční rallye](#)

---

### Aktuální zpoždění

Aktualizace: středa 27.6., 20:39

| vlak                   |                 | stanice                           | prav./skut. čas  | zpoždění                             |
|------------------------|-----------------|-----------------------------------|------------------|--------------------------------------|
| <a href="#">SC 33</a>  | SC Pendolino    | Praha hl.n. - Ostrava hl.n.       | Ostrava hl.n.    | 20:16/20:32 (přij.) 16 min. zpoždění |
| <a href="#">EC 72</a>  | Smetana         | Wiener Neustadt Hbf - Praha hl.n. | Praha-Libeň      | 19:12/19:08 (odj.) 4 min. náskok     |
| <a href="#">EC 74</a>  | Franz Schubert  | Wiener Neustadt Hbf - Praha hl.n. | Záboří n.L.      | 20:32/20:34 (odj.) 2 min. zpoždění   |
| <a href="#">EC 76</a>  | Antonín Dvořák  | Wiener Neustadt Hbf - Praha hl.n. | Brno hl.n.       | 20:37/20:40 (přij.) 3 min. zpoždění  |
| <a href="#">Ex 126</a> | Fatra           | Žilina - Praha hl.n.              | Praha-Libeň      | 20:37/20:36 (přij.) včas             |
| <a href="#">Ex 140</a> | Beskyd          | Žilina - Olomouc hl.n.            | Suchbát n.O.     | 20:36/20:38 (odj.) 2 min. zpoždění   |
| <a href="#">Ex 142</a> | Odra            | Žilina - Praha hl.n.              | Pardubice hl.n.  | 20:38/20:38 (přij.) včas             |
| <a href="#">Ex 151</a> | Jan Perner      | Praha hl.n. - Žilina              | Č. Těšín         | 20:33/20:38 (odj.) 5 min. zpoždění   |
| <a href="#">Ex 153</a> | Hutník          | Praha hl.n. - Třinec              | Červenka         | 20:31/20:36 (odj.) 5 min. zpoždění   |
| <a href="#">EC 175</a> | Jan Jesenius    | Hamburg-Altona - Budapest         | Brno-H. Heršpice | 18:26/20:36 (odj.) 130 min. zpoždění |
| <a href="#">EC 177</a> | Johannes Brahms | Berlin - Wien Praterstern         | Brno hl.n.       | 20:22/20:37 (odj.) 15 min. zpoždění  |
| <a href="#">EC 179</a> | Alois Negrelli  | Berlin - Praha hl.n.              | Roudnice n.L.    | 20:37/20:36 (odj.) včas              |
| <a href="#">R 200</a>  | Anton Bruckner  | Linz Hbf - Praha hl.n.            | Praha-Vršovice   | 20:36/20:37 (odj.) 1 min. zpoždění   |
| <a href="#">EC 270</a> | Avala           | Beograd - Praha hl.n.             | Březová n.Svit.  | 20:20/20:31 (odj.) 11 min. zpoždění  |

Obrázek 6 - Prezentace Babbitron [8]



|                     |  |
|---------------------|--|
| Adresa webu:        | kam.mff.cuni.cz                          |
| Přímá URL k datům:  | http://kam.mff.cuni.cz/~babilon/zponline |
| Nalezeno nástrojem: | seznam.cz, google.com                    |
| Datum hledání:      | 27.6.2012                                |

Tabulka 3 - Babitron

### 2.3.4 Web - České dráhy

V této kapitole jsem čerpal ze zdroje [9].

Jde o samotný webový portál našeho národního dopravce Českých drah. Nabízí spousty užitečných funkcí a informací o dění na železniční trati, kde ČD provozují dopravu, jako jsou: vyhledávání spojení, řazení vlaků, poloha vlaku, jízdní řád, on-line jízdní řád, hledání spojení přes mobilní telefon a mnoho dalších, ale přímo nějakou speciální sekci o meškání vlaků zde nenajdeme.

Z hlediska řešení bakalářské práce je nejdůležitější pouze funkce poloha vlaku, která po zadání čísla nebo jména (názevu) vlaku, vyhledá zda-li je vlak právě na trati a po té zobrazí informace o zpoždění. Nevýhodou je v tomto případě, že společně se zadáváním vlaku musíme vyplnit ještě jeden údaj a to opsat tři kontrolní znaky z CAPTCHA<sup>2</sup>.

Úvod / ČD on-line / Poloha vlaku

### Poloha vlaku

Zadejte číslo vlaku nebo jeho jméno (je-li pojmenován). Následně obdržíte informaci, zda vlak jede podle jízdního řádu nebo velikost jeho případného zpoždění (případně hlášení o nenalezení vlaku apod.). Neznáte-li číslo vlaku ani jeho jméno, vyhledejte je v [jízdním řádu](#). V knižním jízdním řádu je číslo vlaku uvedeno v hlavičce vlaku, název vlaku najdete ve sloupci časových údajů. Tyto informace obsahují i elektronické jízdní řády.

Zadejte číslo nebo název (část) vlaku:

Opište 3 kontrolní znaky z obrázku vpravo:

[Špatně čitelný kód](#)

**EC 148 (Hukvaldy)** [Vše o vlaku](#) [Ukázat na mapě](#)

|                           |   |
|---------------------------|---|
| Směr jízdy                | <a href="#">Praha hl.n.</a>   |
| Informace ze stanice      | <a href="#">Pardubice hl.n.</a>   |
| Skutečný čas              | 28. 6. 2012 12:58 (odj.)  |
| Pravidelný čas            | 12:41 (odj.)  |
| Poloha                    | <b>17 min. zpoždění</b>   |
| Informace vystavena       | 28.6.2012 13:04:39  |
| Zobrazit historii v grafu | <input type="text" value="Třinec"/> <input type="button" value="Zobrazit"/> |

**Průvodce on-line službami ČD**

- Vyhledat spojení
- Spočítat cenu (Cenový asistent)
- Koupit jízdenku (eShop ČD)
- Řazení vlaků
- Poloha vlaku
  - Poloha vlaku pomocí SMS
  - Mapa sítě
- Seznam stanic
- eMapa ČD
- ČD-Kurýr on-line
- Omezení provozu
- ČD Low cost

**Zákazníci**

- Tel Em Kor
- Přejít na jiný

**O firmě**

- Veřejné z
- Nemovit
- Tiskové c
- Kariéra

Obrázek 7 - Prezentace České dráhy [9].

<sup>2</sup> CAPTCHA - je akronym pro „completely automated public Turing test to tell computers and humans apart“, tedy „plně automatický veřejný Turingův test k odlišení počítačů a lidí“. Test spočívá zpravidla v zobrazení obrázku s deformovaným textem, přičemž úkolem uživatele je zobrazený text opsat do příslušného vstupního políčka. [11]

|                     |   |
|---------------------|---|
| Adresa webu:        | www.cd.cz   |
| Přímá URL k datům:  | http://www.cd.cz/cd-online/poloha-vlaku/default.php |
| Nalezeno nástrojem: | seznam.cz, google.com                               |
| Datum hledání:      | 28.6.2012   |

Tabulka 4 - České dráhy

### 2.3.5 Web - Poloha vlaku

V této kapitole jsem čerpal ze zdroje [10].

O těchto webových stránkách lze říci, že jsou oproti ostatním webům jen a pouze speciálně určeny pro účely sledování vlaků. Žádný jiný „nadbytečný“ obsah či služby zde nenajdeme. Vlaky jsou sledovány na území Slovenské republiky. Ostatně to se lze taky na webu ve vysvětlivkách dočíst: „Jsou sledovány všechny vlaky jedoucí po síti ŽSR minimálně mezi dvěma stanicemi, které zaznamenávají časové údaje o pohybu vlaku do PISu. V závislosti na dni v týdnu, nebo datu v měsíci jsou ze sledování vyřazeny vlaky, na které se vztahuje omezení jízdy kvůli výluce na trati“.

Vzhledově a logickým členěním tabulek se stránky velice podobají českému webu Babbitron. Najdeme zde rozdělení např. dle sekcí on-line tabulka, on-line mapa, eurocity, expresy, rychlíky a další. Při poklepnání na detail vlaku se zobrazí soupis stanic a statistika zpoždění během posledních sedmi dnů. Snad jediným zadrhelem tohoto webu je tedy to, že není podporováno sledování všech vlaků na území české republiky, ale pouze ty co se účastní i provozu na slovenské železnici.

jazyk: [česky](#) | [slovensky](#) | [english](#) | [deutsch](#) | [polsk](#)

|                 |              |          |           |           |         |          |        |        |
|-----------------|--------------|----------|-----------|-----------|---------|----------|--------|--------|
| on-line tabulka | On-line mapa | EuroCity | InterCity | EuroNight | Expresy | Rychlíky | Spěšné | Osobní |
|-----------------|--------------|----------|-----------|-----------|---------|----------|--------|--------|

Vysvětlivky | Návštěvní kni

Vyhledat

- Celá síť - - Všechny stanice -

Poslední aktualizace: 14:45:02

| vlak                   | trasa   | info ze stanice           | skutečný čas | odchyka          |
|------------------------|---|---------------------------|--------------|------------------|
| EC 70 GUSTAV MAHLER    | Wiener Neustadt (12.04) - Praha hl.n. (17.18)       | Brno hl.n.                | 14:40        | včas             |
| EC 110 PRAHA           | Warszawa Wschodnia (9.08) - Praha hl.n. (17.50)     | Polom                     | 14:41        | včas             |
| Ex 120 KOŠTĚAN         | Košice (10.48) - Praha hl.n. (18.50)                | Púchov                    | 14:30        | 7 min. zpoždění  |
| Ex 121 KOŠTĚAN         | Praha hl.n. (9.17) - Košice (17.12)                 | Varín                     | 14:34        | 4 min. zpoždění  |
| Ex 128 HRADČANY        | Košice (8.48) - Praha hl.n. (16.50)                 | Lukavice na Moravě        | 14:42        | 3 min. zpoždění  |
| EC 136 MORAVIA         | Budapest-Keleti pu. (7.25) - Ostrava hl.n. (12.57)  | Ostrava-Svinov            | 13:02        | 14 min. zpoždění |
| Ex 144 LANDEK          | Žilina (14.27) - Praha hl.n. (19.50)                | Krásno n/Kysucou          | 14:40        | 1 min. náskok    |
| Ex 146 LEDŠ JANÁČEK    | Žilina (10.27) - Praha hl.n. (15.50)                | Uhersko                   | 14:41        | 11 min. zpoždění |
| Ex 147 LEDŠ JANÁČEK    | Praha hl.n. (12.17) - Žilina (17.33)                | Štěpánov                  | 14:36        | včas             |
| Ex 149 HUKVALDY        | Praha hl.n. (14.17) - Žilina (19.33)                | Český Brod                | 14:41        | 2 min. náskok    |
| EC 170 HUNGARIA        | Budapest-Keleti pu. (9.25) - Berlin Hbf (21.13)     | Březov nad Svitavou       | 14:37        | 17 min. zpoždění |
| EC 171 HUNGARIA        | Berlin Hbf (6.45) - Budapest-Keleti pu. (18.35)     | Vranovice                 | 14:38        | 1 min. zpoždění  |
| EC 172 VINDOBONA       | Villach Hbf (5.26) - Hamburg Altona (21.48)         | Praha-Holešovice          | 14:42        | 5 min. zpoždění  |
| EC 173 VINDOBONA       | Hamburg Altona (6.14) - Villach Hbf (22.37)         | Pardubice hlavní nádraží  | 14:41        | včas             |
| EC 175 JAN JESENÍUS    | Hamburg Altona (8.14) - Budapest-Keleti pu. (22.35) | Hněvice                   | 14:42        | včas             |
| R 202 F. A. GERSTNER   | Summerau (12.39) - Praha hl.n. (16.40)              | Veselí nad Lužnicí        | 14:42        | 3 min. zpoždění  |
| R 203 F. A. GERSTNER   | Praha hl.n. (11.16) - Summerau (15.09)              | Velešín                   | 14:41        | 7 min. zpoždění  |
| Ex 221 DETVAN          | Praha hl.n. (11.17) - Zvolen os. stanica (18.39)    | Valašské Meziříčí         | 14:40        | 2 min. zpoždění  |
| Ex 230 KYSUCA          | Žilina (12.27) - Ostrava hl.n. (14.27)              | Ostrava střed             | 14:03        | 4 min. náskok    |
| Ex 231 KYSUCA          | Ostrava hl.n. (13.37) - Žilina (15.33)              | Český Těšín (Cz.-Cieszyn) | 14:38        | 7 min. zpoždění  |
| EC 273 JAROSLAV HAŠEK  | Praha hl.n. (7.42) - Budapest-Keleti pu. (14.35)    | Štárovo                   | 13:12        | včas             |
| EC 274 SLOVAN          | Budapest-Keleti pu. (11.25) - Praha hl.n. (18.18)   | Devínská Nová Ves         | 14:38        | 17 min. zpoždění |
| EC 275 SLOVAN          | Praha hl.n. (9.42) - Budapest-Keleti pu. (16.35)    | Tinovec n/Všohm           | 14:36        | 6 min. zpoždění  |
| Ex 354 JAN HUS         | Praha hl.n. (13.07) - Nürnberg Hbf (17.56)          | Rokycany                  | 14:40        | 10 min. zpoždění |
| Ex 355 ALBERT EINSTEIN | München Hbf (9.01) - Praha hl.n. (14.51)            | Praha-Radotín             | 14:36        | včas             |
| IC 504 TATRAN          | Košice (13.33) - Bratislava hl.st. (18.32)          | Vydmík                    | 14:35        | 3 min. zpoždění  |
| IC 505 TATRAN          | Bratislava hl.st. (9.34) - Košice (14.35)           | Kostolány nad Hornádou    | 14:33        | 5 min. zpoždění  |

Obrázek 8 - Prezentace Poloha vlaku [10]

|                     |   |
|---------------------|---|
| <b>Adresa webu:</b> | <b>poloha.vlaku.info</b>                  |
| Přímá URL k datům:  | http://poloha.vlaku.info/cs/kategorie/EC/ |
| Nalezeno nástrojem: | seznam.cz, google.com                     |
| Datum hledání:      | 28.6.2012                                 |

**Tabulka 5 - Poloha vlaku**

### 2.3.6 Srovnání vybraných webů

Nyní se budeme věnovat srovnáním a shrnutím všech vybraných webů. Před srovnáváním je třeba si stanovit jistá pravidla nebo kritéria, pomocí kterých budeme weby hodnotit.

Následně všem těmto kritériím budou přiřazeny body 0-10, přičemž nejmenší číslo značí nejhůře hodnocené kritérium a naopak největší nejlépe. Web s největším počtem bodů bude následně vybrán a použit v implementaci samotné aplikace jako zdroj pro získávání dat.

Tato kritéria jsou:

- **Způsob získání dat** – data je možné získat třemi způsoby, které již byly popsány a jsou jimi: přímý přístup k databázi (PPD - 10b), pomocí webové služby (WBS - 5b) a parsování webu (PAR - 1b)
- **Aktuálnost** – je potřeba aby data byla v co nejkratším časovém okamžiku aktualizována, weby které se aktualizují jen jednou denně, nejsou moc vhodné pro výběr na implementaci
- **Přehlednost** – důležitá je taky přehlednost webu, orientace v datech a tabulkách
- **Přístupnost** – při pravidelném automatickém čerpání dat, je potřebné mít přímý přístup k datům bez překážek, kterými jsou na webu především testy na ověřování zda jde o uživatele či roboty tzv. CAPTCHA testy
- **Obsáhlost a hodnota dat** – jedním z dalších faktorů je to zda jsou data obsahem postačující k následnému zpracování a jejich váha
- **Oprávnění k stahování dat** – jedno z dosti podstatných kritérií, protože pravidelné automatické stahování dat může být pod ochranou autorských práv a při takovém počínání může vést k zablokování přístupu na web (povoleno - 10b, není jasné - 5b, zakázáno - 0b)

|                           | Česká železnice | Dopravní web | Babitron  | České dráhy | Poloha vlaku |
|---------------------------|-----------------|--------------|-----------|-------------|--------------|
| Způsob získání dat        | PAR             | -            | PAR       | PAR         | PAR          |
|                           | 1               | 1            | 1         | 1           | 1            |
| Aktuálnost                | 3               | 0            | 9         | 9           | 9            |
| Přehlednost               | 6               | 0            | 9         | 8           | 10           |
| Přístupnost               | 10              | 0            | 10        | 0           | 10           |
| Obsáhlost a hodnota dat   | 1               | 0            | 6         | 3           | 2            |
| Oprávnění k stahování dat | 5               | 5            | 0         | 5           | 5            |
| <b>Konečné hodnocení</b>  | <b>26</b>       | <b>6</b>     | <b>35</b> | <b>26</b>   | <b>37</b>    |

Tabulka 6 - Srovnání webů

Podle tohoto bodového hodnocení je tedy nejlepší variantou web Poloha vlaku následovaný weby Babitron, Česká železnice, České dráhy a Dopravní web.

Ve skutečnosti se ale jako zdroj pro sběr dat o zpoždění vlaků nehodí ani jeden z webů a to z těchto konkrétních důvodů:

- **Poloha vlaku** – neobsahuje všechny vlaky na území ČR, vlaky jsou sledovány jen na ŽSR
- **Babitron** – není povoleno stahovat data
- **Česká železnice** – neaktuálnost dat, neobsahuje všechny vlaky
- **České dráhy** – není přímý přístup, pouze přes CAPTCHA
- **Dopravní web** – vůbec neobsahuje data

## 2.4 Závěr

V této části práce měl být uveden přehled získání dat online, kdy nejprve byly vysvětleny obecné pojmy, dále objasněny možné způsoby získávání dat a po té provedena analýza webů. Celkem jich bylo vybráno a představeno pět, přičemž byly podle kritérií srovnány. Nakonec jsem usoudil, že pro získávání dat se nehodí ani jedna možnost, proto byla práce dále rozdělena na dvě části: serverovou a aplikační(klientskou). Serverová část se bude zabývat návrhem a implementací webu, ze kterého se budou čerpat online data.

Kdežto klientská část se bude zabývat vlastní aplikací pro sběr dat, jak bylo původním záměrem.

### 3 Použité technologie

Před tím než se pustíme do serverové části práce je potřeba se seznámit s technologiemi použitými pro vývoj aplikace.

#### 3.1 HTML, XHTML, XML

Nejprve bych chtěl popsat technologie, bez kterých se dnes neobejde žádná webová stránka. Jsou to značkovací jazyky HTML, XHTML a XML.

##### 3.1.1 HTML

V této kapitole jsem čerpal ze zdroje [12].

HyperText Markup Language, označovaný zkratkou HTML, je značkovací jazyk pro hypertext. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu. Jazyk je aplikací dříve vyvinutého rozsáhlého univerzálního značkovacího jazyka SGML (Standard Generalized Markup Language). Vývoj HTML byl ovlivněn vývojem webových prohlížečů, které zpětně ovlivňovaly definici jazyka.

Jazyk je charakterizován množinou značek (tzv. tagů) a jejich atributů definovaných pro danou verzi. Mezi značky se uzavírají části textu dokumentu a tím se určuje význam (sémantika) obsaženého textu. Názvy jednotlivých značek se uzavírají mezi úhlové závorky <a>. Část dokumentu tvořená otevírací značkou, nějakým obsahem a odpovídající ukončovací značkou tvoří tzv. element (prvek) dokumentu. Například <strong> je otevírací značka pro zvýraznění textu a <strong>Červená Karkulka</strong> je element obsahující zvýrazněný text. Součástí obsahu elementu mohou být další vnořené elementy. Atributy jsou doplňující informace, které upřesňují vlastnosti elementu. Značky jsou obvykle párové, přičemž koncová značka je shodná se značkou počáteční, jen má před názvem znak lomítka. Příklad pro označení odstavce:

```
<p>Text odstavce</p>
```

Dokument v jazyku HTML má předepsanou strukturu:

- Deklarace DTD<sup>3</sup> – je povinná až ve verzi 4.01, je uvedena direktivou <!DOCTYPE>.
- Kořenový element – element html (značky <html> a </html>) reprezentuje celý dokument. Kořenový element je povinný, ale otevírací a ukončovací značka

---

<sup>3</sup> DTD - Document Type Definition, česky Definice typu dokumentu je jazyk pro popis struktury XML případně SGML dokumentu. Omezuje množinu přípustných dokumentů spadajících do daného typu nebo třídy. [25]

samotná povinná není (pokud tyto značky nebudou v těle dokumentu uvedeny, prohlížeč si je sám doplní podle kontextu).

- Hlavička elementu – obsahuje metadata, která se vztahují k celému dokumentu. Definují např. název dokumentu, jazyk, kódování, klíčová slova, popis, použitý styl zobrazení. Hlavička je uzavřena mezi značky `<head>` a `</head>`. Element `head` je opět povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle kontextu.
- Tělo dokumentu – obsahuje vlastní text dokumentu. Vymezuje se značkami `<body>` a `</body>`. Element `body` je povinný, ale jeho otevírací a koncová značka povinná není, prohlížeč ji sám doplní podle kontextu.

### 3.1.2 XHTML

V této kapitole jsem čerpal ze zdroje [13].

XHTML je zkratka anglického extensible hypertext markup language – „rozšiřitelný hypertextový značkovací jazyk“. Jde o značkovací jazyk pro tvorbu hypertextových dokumentů v prostředí WWW vyvinutý W3C<sup>4</sup>. Původně se předpokládalo, že se stane nástupcem jazyka HTML, jehož vývoj byl verzí 4.01 ukončen. V roce 2007 však došlo k založení pracovní skupiny, která má za cíl vytvořit novou verzi HTML, která ponese označení HTML 5 a její XML variantu XHTML 5.

Existuje více verzí tohoto značkovacího jazyka jako jsou: XHTML 1.0, XHTML Basic 1.1, XHTML Mobile Profile, XHTML 1.1 - modulově založené XHTML, XHTML-Print, XHTML 2.0, XHTML 5.

Verze XHTML 1.0 je první specifikace, jejíž cílem bylo převedení staršího jazyka HTML tak, aby vyhovoval podmínkám tvorby XML dokumentů a přitom byla zachována zpětná kompatibilita. Existuje ve třech verzích: Strict, Transitional a Frameset:

- XHTML 1.0 Strict - používá se, pokud chcete strukturovaný dokument osvobozený od formátovacích značek souvisejících s rozvržením stránky. Předpokládá se jeho užití společně s CSS, které vám umožní dosáhnout potřebných grafických efektů. Nicméně i tato verze obsahuje formátovací elementy, například `<b>` nebo `<i>` a naopak zavrhuje některé sémantické elementy, například `<menu>`.
- XHTML 1.0 Transitional - je přechodným DTD pro webové stránky, který vám umožní používat překonané tagy. Je vhodný pro formátování stránek vytvářených pro staré prohlížeče, které nerozumí kaskádovým stylům CSS nebo chcete-li používat ve svých dokumentech některé zavržené, ale sémantické elementy, například již zmíněné `<menu>`.

---

<sup>4</sup> W3C - World Wide Web Consortium je mezinárodní konsorcium, jehož členové společně s veřejností vyvíjejí webové standardy pro World Wide Web. Cílem konsorcia je „Rozvíjet World Wide Web do jeho plného potenciálu vývojem protokolů a směrnic, které zajistí dlouhodobý růst Webu“. [26]

- XHTML 1.0 Frameset - umožňuje používat zastaralé značky jako XHTML 1.0 Transitional a přidává podporu pro rámce. V dnešní době by se mělo rámcům vyhýbat použitím CSS, AJAXu nebo serverových skriptů jako například PHP.

Rozdíly XHTML oproti HTML:

- Všechny tagy musí být ukončené a to včetně nepárových jako jsou <meta>, <link>, <br>, <hr> nebo <img>. Zápis může mít více podob. Buď použijeme klasické (a validní) <img></img> nebo zkrácené <img/> nebo mírně upravené <img />.
- Všechny tagy a jejich atributy musí být zapsány malými písmeny a to z toho důvodu, že jsou takto deklarované v odkazované DTD a X(HT)ML je case sensitive, tedy záleží na velikosti písem.
- Dokument musí začínat XML deklarací. Její použití není povinné, pokud je dokument kódován v UTF-8 nebo pokud určíme kódování vyšší protokolem.

### 3.1.3 XML

V této kapitole jsem čerpal ze zdroje [14] a [15].

Zkratka je odvozena z anglického Extensible Markup Language, česky si lze přeložit jako rozšiřitelný značkovací jazyk. Jde o obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Je zjednodušenou podobou staršího jazyka SGML. Umožňuje snadné vytváření konkrétních značkovacích jazyků (tzv. aplikací) pro různé účely a různé typy dat. Používá se pro serializaci dat, v čemž soupeří např. s JSON či YAML. Zpracování XML je podporováno řadou nástrojů a programovacích jazyků.

Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů, u kterých popisuje strukturu z hlediska věcného obsahu jednotlivých částí, nezabývá se vzhledem. Prezentace dokumentu (vzhled) může být definována pomocí kaskádových stylů. Další možností zpracování je transformace do jiného typu dokumentu, nebo do jiné aplikace XML.

Mezi základní vlastnosti XML patří:

- Standardní formát pro výměnu informací - Není vhodné zasílat dokumenty ve tvaru, který vyžaduje pro zpracování speciální software konkrétní firmy, jako je např. formát DOC, XLS nebo T602. Je používána celá řada operačních a informačních systémů a není možné předpokládat, že každý uživatel vlastní příslušný software. Vznikla tak potřeba vytvořit nějaký jednoduchý otevřený formát, který není úzce svázan s nějakou platformou nebo proprietární technologií. Tím může být právě XML, který je založen na jednoduchém textu a je zpracovatelný (v případě potřeby) libovolným textovým editorem.

- Mezinárodní podpora - XML hned od samého počátku myslel na potřeby i jiných jazyků než je angličtina. Jako znaková sada se implicitně používá ISO 10646 (také Unicode). V XML proto můžeme vytvářet dokumenty, které obsahují texty v mnoha jazycích najednou – můžeme přepínat mezi různými jazyky v jednom dokumentu. Současně je přípustné i jiné libovolné kódování (např. windows-1250, iso-8859-2), musí však být v každém dokumentu přesně určeno. Odpadají tak problémy s konverzí z jednoho kódování do druhého.
- Vysoký informační obsah - Pomocí XML značek (tagů) vyznačujeme v dokumentu význam jednotlivých částí textu. Dokumenty tak obsahují více informací, než kdyby se používalo značkování zaměřené na prezentaci (vzhled) – definice písma, odsazení a podobně. XML dokumenty jsou informačně bohatší.
- Snadná konverze do jiných formátů - Při používání XML dokumentu potřebujeme také dokument zobrazit. XML samo o sobě žádné prostředky pro definici vzhledu nenabízí. Existuje ale několik stylových jazyků, které umožňují definovat, jak se mají jednotlivé elementy zobrazit. Souboru pravidel nebo příkazů, které definují, jak se dokument převede do jiného formátu, se říká styl.
- Automatická kontrola struktury dokumentu - XML neobsahuje předdefinované značky (tagy), je třeba definovat vlastní značky, které budeme používat. Tyto značky je možné (nepovinně) definovat v souboru DTD (Document Type Definition). Potom je možné automaticky kontrolovat, zda vytvářený XML dokument odpovídá této definici. Program, který tyto kontroly provádí, se nazývá parser. Při vývoji aplikací můžeme parser použít, a ten za nás detekuje většinu chyb v datech.
- Hypertext a odkazy - XML stejně jako HTML umožňuje vytváření odkazů v rámci jednoho dokumentu i mezi dokumenty, má však více možností. Je možné vytvářet i vícesměrné odkazy, které spojují více dokumentů dohromady. Tvorba odkazů je popsána ve třech standardech – XLink, XPointer a XPath.

Příklad jednoduchého receptu zapsaného v XML by mohl vypadat tak jak je na obrázku 9.



```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- Poznamka je nutné přidat více receptů. -->
<recept jméno="chleba" čas_přípravy="5 minut" čas_vaření="3 hodiny">
  <titulek>Jednoduchý chleba</titulek>
  <přísada množství="3" jednotka="šálky">Mouka</přísada>
  <přísada množství="0,25" jednotka="unce">Kvasnice</přísada>
  <přísada množství="1,5" jednotka="šálku">Horká voda</přísada>
  <přísada množství="1" jednotka="kávová lžička">Sůl</přísada>
  <instrukce>
    <krok>Smíchejte všechny přísady a dobře prohnětte.</krok>
    <krok>Zakryjte tkaninou a nechte hodinu v teplé místnosti.</krok>
    <krok>Znovu prohnětte, umístěte na plech a pečte v troubě.</krok>
  </instrukce>
</recept>

```

Obrázek 9 - Ukázka kódu XML [14]

## 3.2 CSS

V této kapitole jsem čerpal ze zdroje [16].

Kaskádové styly (v anglickém originále Cascading Style Sheets se zkratkou CSS) je jazyk pro popis způsobu zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML.

Jazyk byl navržen standardizační organizací W3C. Byly vydány zatím dvě úrovně specifikace CSS1, CSS2 a nyní se pracuje na verzi CSS3. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují obsah a strukturu dokumentu, ale i způsob jeho zobrazení.

Definice kaskádových stylů sestává z několika pravidel. Každé pravidlo obsahuje selektor a blok deklarácí. Každý blok deklarácí pak obsahuje seznam deklarácí oddělených středníky a každá deklarace sestává z identifikátoru vlastnosti, následuje dvojtečka a hodnota vlastnosti. Nepovinně ještě může následovat označení `!important`, které zvýší sílu deklarace. Na obrázku 10 uvedu příklad pravidla.

```

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

```

Obrázek 10 - Ukázka kódu CSS [16]

Používání kaskádových stylů ve srovnání se samotným HTML v praxi přináší výhody:

- Rozsáhlejší možnosti formátování. Například pro formátování bloku textu – tj. určení vzdálenosti od jejich elementu či okraje stránky nenabízí HTML nic. CSS má vlastnosti padding a margin. V HTML by byla potřeba vytvořit složitou konstrukci vnořených tabulek.
- Jednodušší údržba webové prezentace. Pokud chceme změnit nějaký detail, jako třeba barvu nadpisu, nemusíme složitě procházet HTML kód nebo různé HTML šablony, ale můžeme změnit pouze jednu vlastnost v CSS souboru, který je přilinkován ke všem stránkám.“
- Oddělení struktury a stylu. V jednom (HTML) dokumentu budeme mít pouze sémanticky označen obsah a v druhém (CSS) dokumentu máme definice vzhledu stránek. Tím dosáhneme snadnějšího strojového zpracování samotného obsahu stránek, do kterého se nám nepletou prvky definující vzhled.
- Cachování stylů – webový prohlížeč si může soubor se styly uložit do cache paměti, čímž může být dosaženo zrychlení načtení stránky.
- CSS vlastnosti jednotlivých elementů můžeme dynamicky měnit pomocí Javascriptu.
- S pomocí CSS můžeme jednoduše formátovat i jakýkoliv jazyk XML.
- Mohou také existovat různé styly pro různá výstupní zařízení. Jednoduše tak můžeme nadefinovat různý styl pro tisk, projekci, mobil, PDA, běžný prohlížeč či dokonce pro každý prohlížeč jiný styl.
- V kombinaci s Javascriptem mohou vzniknout účinné bookmarklety, které mohou různě vylepšovat vzhled stránky. Například odstranit všechny obrázky na pozadí, změnit pozadí na bílé a písmo na černé apod.

### 3.3 JAVASCRIPT

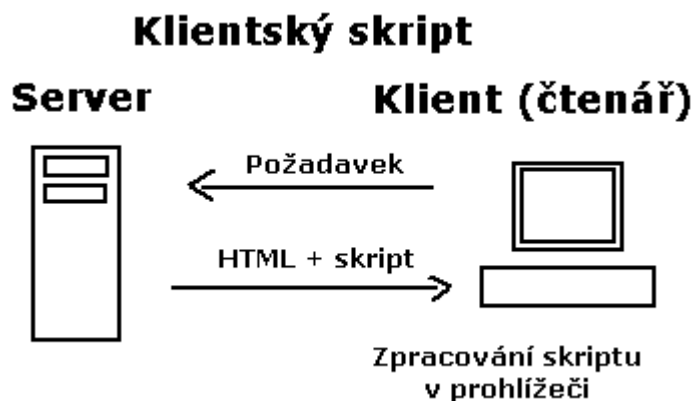
V této kapitole jsem čerpal ze zdroje [17] a [18].

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, jehož autorem je Brendan Eich z tehdejší společnosti Netscape. Používá se jako interpretovaný programovací jazyk pro WWW stránky, často vkládaný přímo do HTML kódu stránky. Jsou jím obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků.

Jeho syntaxe patří do rodiny jazyků C/C++/Java. Slovo Java je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript byl původně obchodní název implementace společnosti Netscape, kde byl vyvíjen nejprve pod názvem Mocha, později LiveScript, ohlášen byl společně se společností Sun Microsystems v prosinci 1995 jako doplněk k

jazykům HTML a Java. Pro verzi firmy Microsoft je použit název JScript. Ten je podporován platformou .NET.

Program v JavaScriptu se obvykle spouští až po stažení WWW stránky z Internetu (tzv. na straně klienta), na rozdíl od ostatních jiných interpretovaných programovacích jazyků (např. PHP a ASP), které se spouštějí na straně serveru ještě před stažením z Internetu. Z toho plynou jistá bezpečnostní omezení, JavaScript např. nemůže pracovat se soubory, aby tím neohrozil soukromí uživatele.



Obrázek 11 – Javascript [18]

### 3.4 AJAX

V této kapitole jsem čerpal ze zdroje [19].

Zkratka AJAX je odvozena z anglického Asynchronous JavaScript and XML, což je obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovunačítání. Na rozdíl od klasických webových aplikací poskytují uživatelsky příjemnější prostředí, ale vyžadují použití moderních webových prohlížečů.

Tyto aplikace jsou vyvíjeny s využitím technologií:

- HTML (nebo XHTML) a CSS pro prezentaci informací.
- DOM<sup>5</sup> a JavaScript pro zobrazování a dynamické změny prezentovaných informací.
- XMLHttpRequest pro asynchronní výměnu dat s webovým serverem (typicky je užíván formát XML, ale je možné použít libovolný jiný formát včetně HTML, prostého textu, JSON či EBML).

---

<sup>5</sup> DOM - (Document Object Model – objektový model dokumentu) je objektově orientovaná reprezentace XML nebo HTML dokumentu. DOM je API umožňující přístup či modifikaci obsahu, struktury, nebo stylu dokumentu, či jeho částí. [27]

### 3.5 JQUERY

V této kapitole jsem čerpal ze zdroje [20] a [21].

jQuery je malá javascriptová knihovna (sada funkcí), která usnadňuje interakci mezi JavaScriptem a HTML. Klade důraz na jednoduchost, čitelnost, rychlost. Je multiplatformní (funguje na více operačních systémech) a je dostupná zdarma. Byla vydána Johnem Resigem v lednu 2006 na newyorském BarCampu.

Stejně jako CSS odděluje „zobrazovací“ charakteristiky od struktury HTML, jQuery odděluje „chování“ od struktury HTML. Například místo přímé specifikace onclick události přímo v HTML kódu tlačítka by stránka řízená jQuery napřed našla vhodný element tlačítka a potom změnila jeho manipulátor události. Takovéto oddělení chování od struktury se také často nazývá jako princip nevtíravého JavaScriptu.

Mezi to co všechno jQuery nabízí patří například tyto funkce:

- Výběr DOM elementů
- Funkce pro procházení a změnu DOM
- Události
- Manipulace s CSS
- Efekty a animace
- AJAX
- Rozšiřitelnost
- Utility – např. informace o prohlížeči nebo funkce each
- Javascriptové pluginy

### 3.6 PHP

V této kapitole jsem čerpal ze zdroje [22] a [23].

PHP je zkratka odvozená původně z anglického Personal Home Page, nyní znamená Hypertext Preprocessor. Jde o skriptovací programovací jazyk. Je určený především pro programování dynamických internetových stránek a webových aplikací například ve formátu HTML, XHTML či WML. PHP lze použít i k tvorbě konzolových a desktopových aplikací.

Při použití PHP pro dynamické stránky jsou skripty prováděny na straně serveru – k uživateli je přenášen až výsledek jejich činnosti.

Interpret PHP skriptu je možné volat pomocí příkazového řádku, dotazovacích metod HTTP nebo pomocí webových služeb. Syntaxe jazyka je inspirována několika programovacími jazyky (Perl, C, Pascal a Java). PHP je nezávislý na platformě, rozdíly v různých operačních systémech se omezují na několik systémově závislých funkcí a skripty lze většinou mezi operačními systémy přenášet bez jakýchkoli úprav. Podporuje mnoho knihoven pro různé účely - např. zpracování textu, grafiky, práci se soubory, přístup k většině databázových systémů (MySQL, ODBC, Oracle, PostgreSQL, MSSQL), podporu celé řady internetových protokolů (HTTP, SMTP, SNMP, FTP, IMAP, POP3, LDAP).

Je rozhodně alespoň jedna funkce PHP, která se hodí snad do každého webu. Na webových stránkách ve zdrojové (HTML) části se obvykle opakují některé sekce, hlavička s odkazy, menu, patička. S PHP si lze snadno vytvořit šablonu pro web, do které se budou vkládat soubory s menu, patičkou atd. Můžeme tedy mít menu nakódované v jediném souboru a do dalších stránek ho pouze kopírovat. Až budeme chtít menu změnit bude to pak nesmírně jednoduché. Zde uvedu příklad vložení menu do šablony:

```
<?php include_once("menu.html") ?>
```

### 3.7 MYSQL

V této kapitole jsem čerpal ze zdroje [24].

MySQL je multiplatformní databáze, vytvořená švédskou firmou MySQL AB, nyní vlastněná společností Sun Microsystems. K dispozici je jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci.

Komunikace s touto databází probíhá, jak už název napovídá, pomocí jazyka SQL což je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. SQL je zkratka anglických slov Structured Query Language (strukturovaný dotazovací jazyk).

Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. Velmi oblíbená a často nasazovaná je kombinace Linux, MySQL, PHP a Apache jako základní software webového serveru.

## 4 Serverová část

V této kapitole se budu zabývat serverovou částí, tzn. implementací a návrhem systému, který bude zastřešovat a sloužit jako zdroj pro čerpání dat pro cílovou aplikaci.

### 4.1 Základní charakteristika

Základní charakteristika se odvíjí od toho co je očekáváno nebo požadováno aby tento systém měl splňovat.

Sytém by měl být tedy schopen zobrazovat aktuální, ale i starší data o vlacích. Vše v praxi funguje tak, že každý vlak má vlastní soupis stanic (jízdni řád) kde a kdy staví, přičemž u každé stanice má údaj o zpoždění na příjezdu či odjezdu tohoto vlaku v uvedený den. Přístup k těmto soupisům daného vlaku je umožněn buďto přes seznam všech vlaků, nebo na základě zadaných údajů vložených uživatelem aplikace.

Tato data by měla být přehledně formátována a logicky členěná do tabulek pro pozdější jednodušší manipulaci s případným parsováním. Taktéž by nebylo od věci aby systém umožňoval export přímo do XML formátu, což velice urychlí a taky zjednoduší tok dat mezi tímto systémem a cílovou aplikací pro sběr dat.

Aktuální datum a čas: 2012/08/16 12:30:38

| Vlak č. | Celý název          | Směr trasy    | Výchozí st.              | Konečná st.              | Info ze stanice  | Prav./skut. čas    | Zpoždění(min) |
|---------|---------------------|---------------|--------------------------|--------------------------|------------------|--------------------|---------------|
| 110     | EC 110 Praha        | OSTRAVA-PRAHA | 09:20 Warszawa Centralna | 17:55 Praha hl.n.        | Katowice         | 12:08 / 12:13(odj) | 5             |
| 111     | EC 111 Praha        | PRAHA-OSTRAVA | 10:17 Praha hl.n.        | 18:54 Warszawa Centralna | Zábřeh na Moravě | 12:19 / 12:28(odj) | 9             |
| 145     | EX 145 Landek       | PRAHA-OSTRAVA | 08:17 Praha hl.n.        | 13:39 Žilina             | Karviná hl.n.    | 12:13 / 12:19(odj) | 6             |
| 146     | EX 146 Leoš Janáček | OSTRAVA-PRAHA | 10:27 Žilina             | 16:01 Praha hl.n.        | Ostrava hl.n.    | 12:13 / 12:24(odj) | 11            |
| 147     | EX 147 Leoš Janáček | PRAHA-OSTRAVA | 12:17 Praha hl.n.        | 17:39 Žilina             | Praha-Libeň      | 12:24 / 12:30(odj) | 6             |
| 148     | EX 148 Hukvaldy     | OSTRAVA-PRAHA | 08:27 Žilina             | 14:04 Praha hl.n.        | Česká Třebová    | 12:02 / 12:16(odj) | 14            |
| 506     | SC 506 Pendolino    | OSTRAVA-PRAHA | 09:41 Ostrava hl.n.      | 12:46 Praha hl.n.        | Pardubice hl.n.  | 11:50 / 11:50(odj) | 0             |
| 507     | SC 507 Pendolino    | PRAHA-OSTRAVA | 11:12 Praha hl.n.        | 14:24 Ostrava hl.n.      | Pardubice hl.n.  | 12:07 / 12:13(odj) | 6             |
| 508     | SC 508 Pendolino    | OSTRAVA-PRAHA | 11:41 Ostrava hl.n.      | 14:46 Praha hl.n.        | Ostrava-Svinov   | 11:50 / 11:50(odj) | 0             |
| 511     | SC 511 Pendolino    | PRAHA-OSTRAVA | 12:19 Františkovy Lázně  | 18:27 Bohumín            | Cheb             | 12:29 / 12:29(odj) | 0             |

Export do XML

### Obrázek 12 - Ukázka soupisu aktuálně jedoucích vlaků z webu (serverové části)

System by měl také umět zobrazovat soupis právě jedoucích vlaků. To ve skutečnosti znamená, že vlaky které jsou zrovna v tomto časovém intervalu v pohybu na trati, tak by měly být přidány do soupisky. Součástí tohoto přehledu jsou taky informace odkud a kam daný vlak míří, potom informace z poslední stanice kde vlak stávil a v neposlední řadě informace o současném celkovém zpoždění.

Dále by měl systém splňovat ty podmínky, že se bude jednat o vlaky na území ČR a to na vybraných železničních tratích. Mezi tyto železniční tratě na území ČR jsem se rozhodl vybrat jen ty nejpodstatnější. Proto jsem tedy vybral vlaky jedoucí na jedné z nejméně frekventovaných železničních tratí v ČR, tzn. v úseku Praha – Ostrava a zpět. Všechny tyto vlaky a jejich jízdni řady jsou importovány a uloženy v databázi, kdy jsem jako zdroj dat pro import do databáze čerpal především z webu Českých drah, přičemž některé údaje jsem i mírně upravil pro jednoduchost. To jak je databáze navržena a implementována se budu věnovat v jedné z dalších kapitol.

Se zobrazováním soupisu právě jedoucích vlaků na trati a vůbec aktualizací dat je potřeba ještě zabudovat do systému další mechanismus. Tento mechanismus by měl být schopen sám automaticky přidávat a tím vlastně aktualizovat data o zpoždění každého vlaku. Vlak ve skutečnosti po žel. trati vůbec nejede (celý systém je fiktivní), tudíž by nemohl mít ani zpoždění ve stanici, proto je potřeba toto zpoždění generovat na základě jeho jízdniho plánu. Tuto funkci bude vykonávat generátor. Ten je navržen tak, že nejprve

z databáze vytáhne údaje jízdních řádů všech vlaků a každou minutu kontroluje shodu v tomto čase s časem příjezdu či odjezdu všech stanic daného vlaku. Pokud shoda nastane, je vygenerováno zpoždění v rozmezí 0-30 minut. Poté je podle tohoto časového rozmezí zpoždění automaticky zapsáno do databáze. Ukázka generátoru dat je v příloze B.

## 4.2 Politika uživatelů

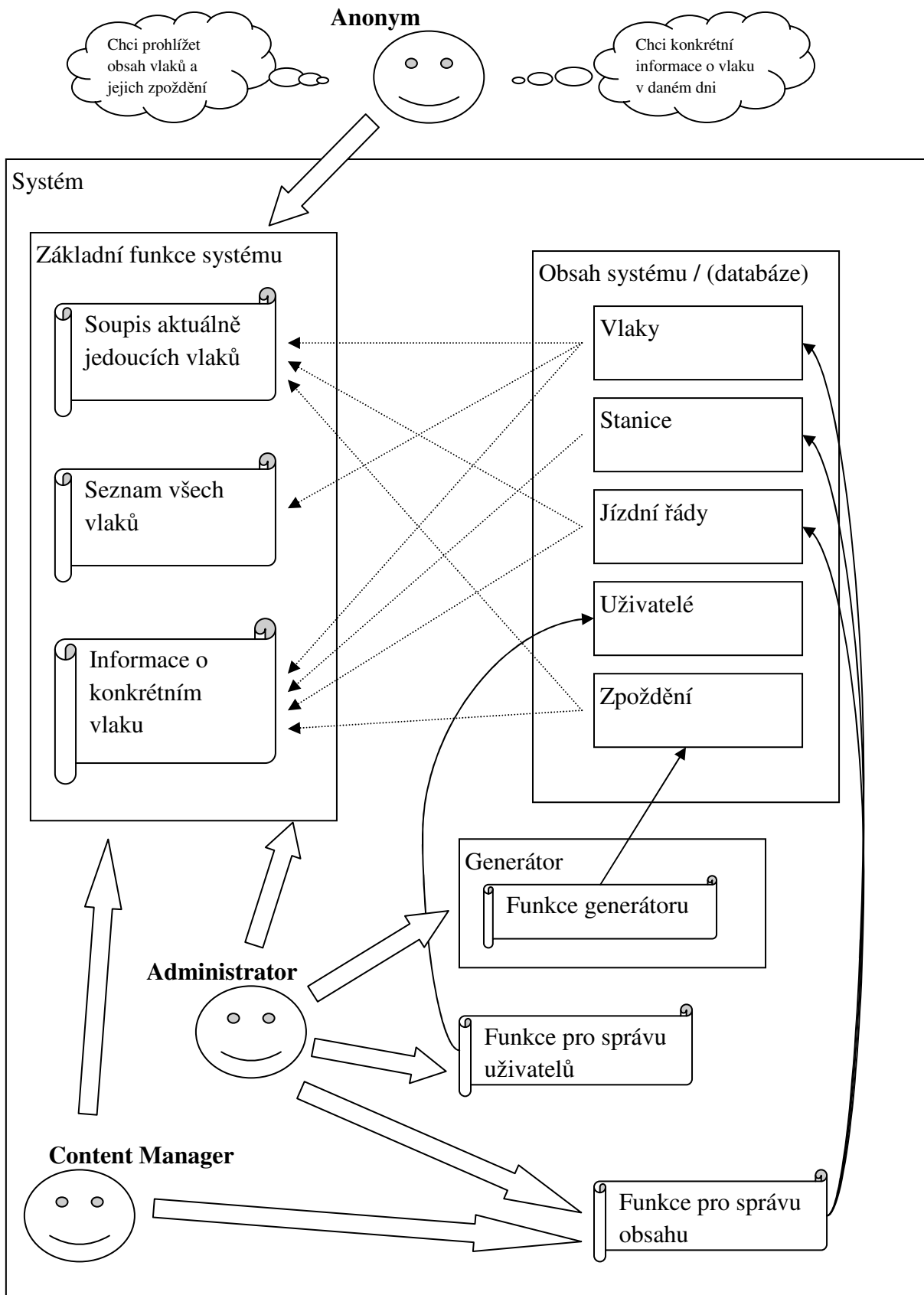
Protože systém poběží na webovém rozhraní na internetu a tudíž bude přístupný všem uživatelům, je nutné zavést i určité oprávnění na některé funkce systému. Ne všichni uživatelé budou moct provádět ty samé operace. Hlavní důvodem na sestavení politiky uživatelů je generátor dat. Ten nesmí být spouštěn více uživateli současně kvůli kolizi vkládaných dat, ale pouze jednou osobou. Dalším důvodem nasazení politiky je kvůli možnosti rozšíření systému o další vlaky.

Práva uživatelů jsem navrhl následovně:

- Administrator – Jde o uživatele, kteří spravují celý systém. Tzn. mohou přidávat další uživatele nebo je odstraňovat ze systému. Přidávat a odebírat obsah (vlaky, jízdní řády, stanice). Spouštět generátor dat zpoždění. A samozřejmě je prohlížení obsahu.
- Content Manager – Jde o uživatele, kteří mohou spravovat pouze obsah systému. Tzn. mohou přidávat a odebírat vlaky, jízdní řády, stanice a mohou prohlížet obsah systému.
- Anonym – Jde o nepřihlášené anonymní uživatele, kteří mohou pouze prohlížet obsah. Žádné jiné oprávnění nemají.

## 4.3 Rich picture diagram

Rich picture je nástroj pro vyjádření složitých nebo těžko vymezitelných problémů tím, že se v podobě obrázků nebo náčrtků nastíní popis vzniklé situace či nějakého hlubšího systému. Popis různých systémů je společný pouze v tom že obsahuje řadu obrázků nebo náčrtků, které mohou být například nějak logicky propojeny. Pouze takto lze Rich picture diagram definovat, protože jinak žádná další společně dohodnutá syntaxe neexistuje. Níže jsem tedy nastínil pomocí tohoto diagramu jak můj systém funguje. [28]



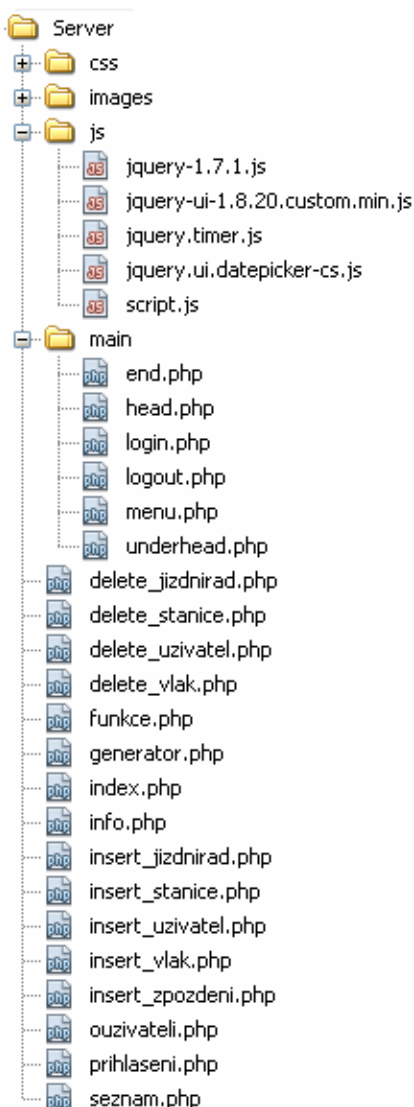
Obrázek 13 - Rich picture diagram - Schéma systému serverové části



## 4.4 Adresářová struktura

Adresářová struktura vyjadřuje způsob členění souborů ve složkách. Začíná od kořenového adresáře, od něhož se větví dále do dalších podadresářů. Vždy by měla mít logickou hierarchii pro pozdější snadný přístup a manipulaci se soubory.

V mém případě je kořenovým adresářem Server, kde je dále členěn na css, images, js a main.



- css - v tomto adresáři jsou veškeré soubory pro pozicování a design stránek.
- images - tento adresář obsahuje obrázky které jsou použity ve webové části.
- js - obsahuje skripty napsané v jazyku javascript.
- main – obsahuje soubory s php skripty, které rozčleňují webové stránky na jednotlivé úseky, z důvodu zjednodušení psaní php kódu.

Obrázek 14 - Adresářová struktura serverové části

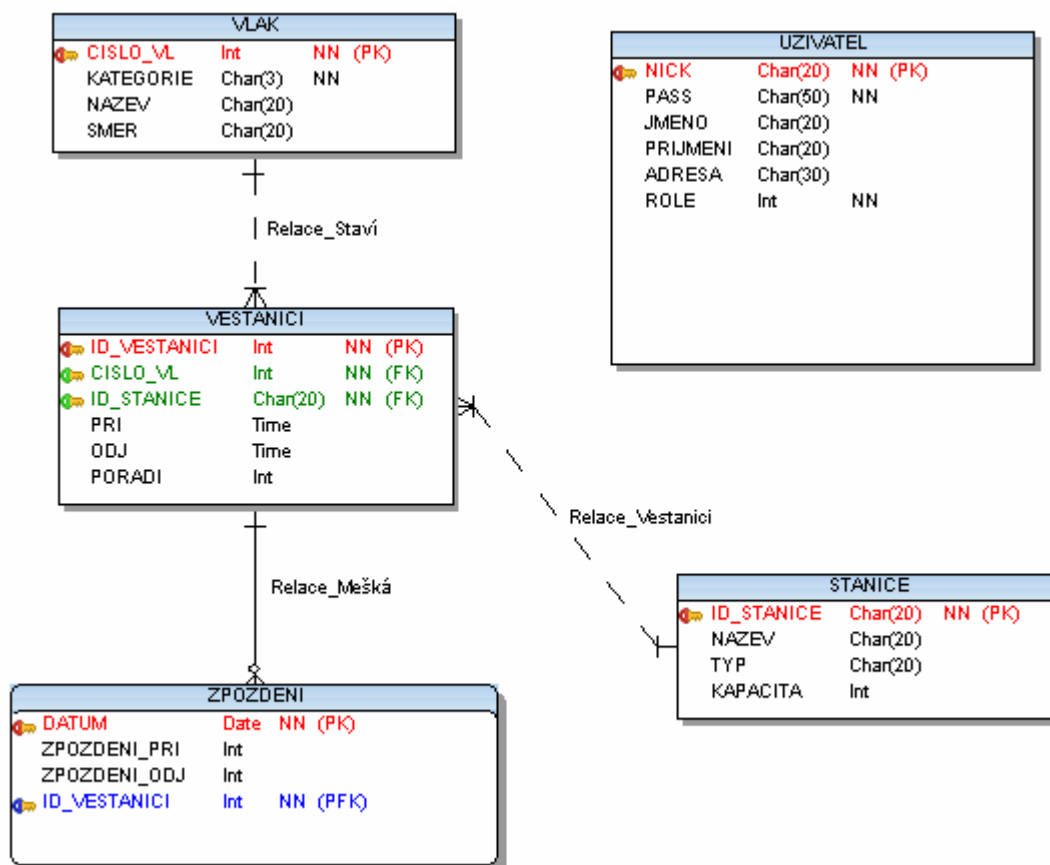
## 4.5 Struktura databáze

Databáze je jedna z nejdůležitějších částí celého systému a prakticky žádná větší webová aplikace pracující s nějakými daty se bez ní neobejde. Klíčovou etapou je návrh databáze, protože právě při špatném návrhu mohou nastat obrovské problémy, chyby nebo

omezení, a to jak na straně databáze a jejího chodu nebo taky ve webové části aplikace. V mém případě jsem pro ukládání dat použil databázi Mysql.

#### 4.5.1 E-R diagram

V prvé řadě je nezbytné určit strukturu informací, které bude databáze obsahovat. Je mnoho způsobů, jak vytvořit funkční model databáze. Jednou z možností je vytvořit E-R diagram.



Obrázek 15 - E-R diagram systému

V souvislosti s E-R diagramem se vyskytují pojmy entita, atribut a vztah.

Entita se dá definovat jako věc schopná samostatné existence a je jednoznačně identifikovatelná. Entita je abstrakcí komplexity určité oblasti. Hovoříme-li o entitě obvyklým způsobem, hovoříme určitým aspektu reálného světa, který se dá od ostatních aspektů odlišit. Entita může být fyzicky existující objekt, jako například dům nebo automobil, nebo událost jako je prodej domu nebo servis automobilu nebo může jít o pojem jako je například zákaznická transakce nebo objednávka. Entitami jsou v mém případě Vlak, Uzivatel, Vestanici, Stanice a Zpozdeni. V E-R diagramu se vyznačují jako obdélníky s popiskem. [30]

Mezi jednotlivými entitami, které jsou propojeny musí být definovány vztahy či relace. Vztah (relationship) zachycuje, jakým způsobem jsou dvě nebo více entit vztažené

mezi sebou. Vztahy se označují slovesy, spojujícími dvě nebo více podstatných jmen. Příklad: vztah vlastní je mezi společností a člověkem, vztah dohlíží je mezi zaměstnancem a oddělením a vztah hraje je mezi umělcem a písní. Zde nám vztahy říkají, že vlak může mít více soupisů ve stanici, dále že ve stanici vlak mešká a může mít více zpoždění a v neposlední řadě, že stanice jsou vázány na entitu vestanici.

Každá relace může mít ještě navíc i určitá omezení. Tato omezení se nazývají kardinalita a parcialita. [37]

Kardinalita představuje omezení v počtu instancí druhé entity, které mají vztah s jakoukoliv instancí první entity. Obecně se rozlišují tři druhy kardinality:

- 1:1 - vztah, ve kterém na obou stranách vystupuje pouze jeden objekt dané entity (např. vztah manželé mezi entitou muž a entitou žena)
- 1:N – na jedné straně je jeden objekt, který může vstoupit do vztahu s jedním nebo více objekty na straně druhé (např. obchod a zboží)
- M:N – vztah, kde na obou stranách vstupuje více objektů

Parcialita nám udává povinnost a volitelnost existence příslušné entity vztahu. Jinak řečeno nám říká, zda záznam v tabulce A musí nebo jen může mít příslušný záznam v tabulce B. [37], [38]

Atributy jsou vlastnosti entity a v diagramu nemusejí být vyznačeny. Jednotlivé atributy jsou patry z mého předchozího schématu databáze. Jsou jimi položky například Nick, Pass, Jmeno, Prijmeni atd.

#### 4.5.2 Entita VLAK

Tato tabulka slouží pro uchování dat o jednotlivých vlacích. Pokud tedy v systému potřebuji získat seznam všech vlaků z databáze, tak pro výpis používám právě tuto tabulku. Atributy jsou CISLO\_VL, KATEGORIE, NAZEV a SMER. Číslo vlaku je jediným primárním klíčem tabulky, z toho plyne že každý vlak má svoje specifické číslo, kterým je identifikován, proto se v tabulce nemohou vyskytnout dva stejné vlaky. Kategorie udává o jaký typ vlaku se jedná např. SC, EX, R a podobně. Zbylé dva atributy jsou jen pro doplnění informací o vlaku, jak byl celý vlak pojmenován a jaký má směr trasy (úseku).

| Klíč | Atribut   | Datový typ | Not null |
|------|-----------|------------|----------|
| PK   | CISLO_VL  | Int        | Ano      |
|      | KATEGORIE | Char (3)   | Ano      |
|      | NAZEV     | Char (20)  |          |
|      | SMER      | Char (20)  |          |

Tabulka 7 - Entita VLAK

### 4.5.3 Entita VESTANICI

Díky této tabulce lze jednoznačně každému vlaku sestavit jízdní řád. Nebo-li každý vlak stává v nějaké stanici podle určitého časového plánu. Tento časový plán je dán pravidelným příjezdem do stanice a pravidelným odjezdem ze stanice, což je popsáno atributy `PRI`, `ODJ`, které jsou právě proto datového typu `Time`. Atribut `PORADI` určuje pořadí stanice od začátku trasy. Primárním klíčem v této tabulce je `ID_VESTANICI` v kombinaci s cizími klíči `CISLO_VL`, `ID_STANICE`.

| Klíč | Atribut      | Datový typ | Not null |
|------|--------------|------------|----------|
| PK   | ID_VESTANICI | Int        | Ano      |
| FK   | CISLO_VL     | Int        | Ano      |
| FK   | ID_STANICE   | Char (20)  | Ano      |
|      | PRI          | Time       |          |
|      | ODJ          | Time       |          |
|      | PORADI       | Int        |          |

Tabulka 8 - Entita VESTANICI

### 4.5.4 Entita STANICE

Tato tabulka slouží pro uchování dat o stanicích na území ČR. Každá stanice je jednoznačně identifikována primárním klíčem `ID_STANICE`, proto nemůže nastat situace kdy by byly v tabulce dvě shodné stanice. Dalšími doplňujícími atributy jsou `NAZEV`, `TYP` a `KAPACITA`, které mohou být využity při rozšíření systému jako doprovodné informace o stanici.

| Klíč | Atribut    | Datový typ | Not null |
|------|------------|------------|----------|
| PK   | ID_STANICE | Char (20)  | Ano      |
|      | NAZEV      | Char (20)  |          |
|      | TYP        | Char (20)  |          |
|      | KAPACITA   | Int        |          |

Tabulka 9 - Entita STANICE

### 4.5.5 Entita ZPOZDENI

Tabulka zpoždění, je jednou z nejpodstatnějších v celém systému, která uchovává data o zpoždění na příjezdech a odjezdech vlaku, který projíždí stanicemi v daném dni, přičemž každý vlak projíždí jednotlivou stanicí k danému datu jen jednou. Každé zpoždění je jednoznačně identifikováno dvěma klíči, jedním vlastním primárním (`DATUM`) a druhým cizím primárním (`ID_VESTANICI`), který je také primárním klíčem entity `VESTANICI`. Díky této kombinaci klíčů lze zaručit, že vlak k danému datu bude ve stanici stavět jen jednou.

| Klíč | Atribut      | Datový typ | Not null |
|------|--------------|------------|----------|
| PK   | DATUM        | Date       | Ano      |
|      | ZPOZDENI_PRI | Int        |          |
|      | ZPOZDENI_ODJ | Int        |          |
| PFK  | ID_VESTANICI | Int        | Ano      |

Tabulka 10 - Entita ZPOZDENI

#### 4.5.6 Entita UZIVATEL

Tabulka UZIVATEL, jak už z názvu napovídá je určena pro uchování záznamů o uživateli a slouží také pro autentifikaci při přihlašování uživatelů do systému. Podstatnými atributy jsou NICK, který jednoznačně identifikuje uživatele, jde o jeho přezdívku, dále PASS což je heslo uživatele a ROLE, s pomocí které lze uživateli nastavit oprávnění. Toto oprávnění může nabývat celých čísel, kdy nejmenším číslem jsou nastaveny nejvyšší pravomoci. Role s číslem 1 - jde o správce, role s číslem 2 - jde o údržbáře, atd. Ostatní atributy jsou opět jen doplňující a nehrají podstatnou roli v přihlašování.

| Klíč | Atribut  | Datový typ | Not null |
|------|----------|------------|----------|
| PK   | NICK     | Char (20)  | Ano      |
|      | PASS     | Char (50)  | Ano      |
|      | JMENO    | Char (20)  |          |
|      | PRIJMENI | Char (20)  |          |
|      | ADRESA   | Char (30)  |          |
|      | ROLE     | Int        | Ano      |

Tabulka 11- Entita UZIVATEL

#### 4.6 Rozvržení a struktura webu

Celý systém běží na webovém rozhraní, kdy jsem stránky napsal za pomoci značkovacího jazyka HTML a dále pak rozšířil o PHP skripty, z důvodu potřeby dynamických stránek. Na statických stránkách by se těžko daly provozovat operace jako přihlášení uživatele, odhlášení uživatele, ověřování, výpisy vlaků, připojení k databázi a spousta dalších operací, které používám. Ukázka webu je v příloze A.

Protože většina stránek má shodnou strukturu dokumentu (liší se pouze v obsahové části) a psaní stejného kódu na všech stránkách zvlášť je zdlouhavé a taky nepraktické, rozdělil jsem stránky na jednotlivé části. Tyto části jsou rozděleny za podpory právě php a jsou uloženy v adresáři main. Jsou to:

- head.php – v této části je napsán kód hlavičky, základní část dokumentu.

- menu.php – v této části je kód hlavního menu.
- end.php –zde je konec, kus kódu, který se opakuje pro ukončení dokumentu.
- underhead.php – tato část obsahuje kód, který následuje po hlavičce.
- login.php – obsahuje skript pro přihlašování.
- logout.php – obsahuje skript pro odhlašování.

Zbylé části stránek jsou uloženy přímo v kořenovém adresáři a všechny se liší pouze v obsahové části, proto jsou každá v jiném souboru. Jsou to např. seznam.php, info.php, ouzivатели.php, atd.

Jediný soubor, který nemá se strukturou stránek nic společného a nachází se taktéž v kořenovém adresáři je funkce.php. V tomto souboru jsou veškeré php skripty pro práci s databází, ať už od přihlašování až po sql výběr dotazu.

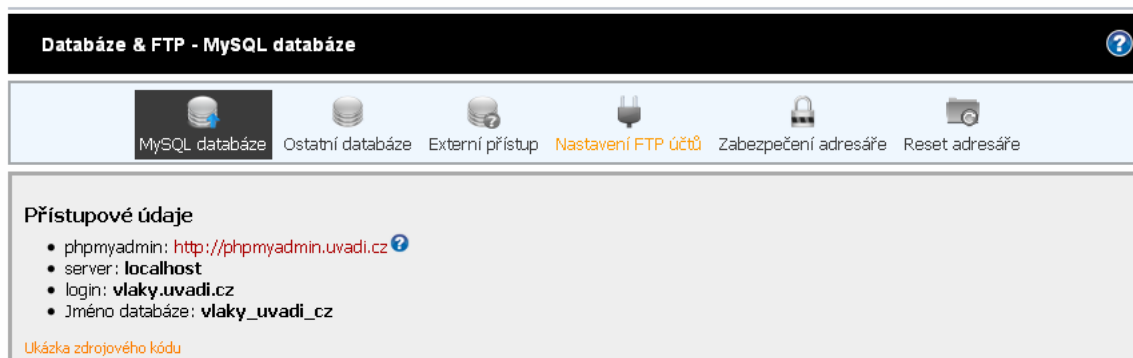
```
function seznamVlaku()
{
    dbConnect();
    $sql = "SELECT * FROM VLAK";
    $res = mysql_query($sql);
    return $res;
}
```

**Obrázek 16 - Ukázka výběru vlaku z funkce.php**

## 4.7 Umístění hostingu

Pro samotný a správný běh mého systému je potřeba web někam umístit. Na internetu existuje celá řada firem, které hosting nabízí. Některé za poplatek, jiné zadarmo. Pro tyto účely jsem vybral firmu banan.cz, která nabízí freehosting s diskovým prostorem 1Gib s podporou Mysql databáze a taky PHP, což bylo při výběru klíčové. Jedinou podmínkou pro používání je umístění reklamy na každé stránce.

Při registraci stačilo zadat pouze pár základních údajů, které byly název subdomény, výběr domény, jméno, heslo a email. Subdoméně jsem dal název vlaky a jako doménu jsem vybral uvadi.cz. Celá adresa webu, kde jsem stránky umístil je tedy vlaky.uvadi.cz .

[Databáze & FTP](#) >> MySQL databáze

**Databáze & FTP - MySQL databáze**

MySQL databáze   Ostatní databáze   Externí přístup   **Nastavení FTP účtů**   Zabezpečení adresáře   Reset adresáře

**Přístupové údaje**

- phpmyadmin: <http://phpmyadmin.uvadi.cz>
- server: **localhost**
- login: **vlaky.uvadi.cz**
- Jméno databáze: **vlaky\_uvadi\_cz**

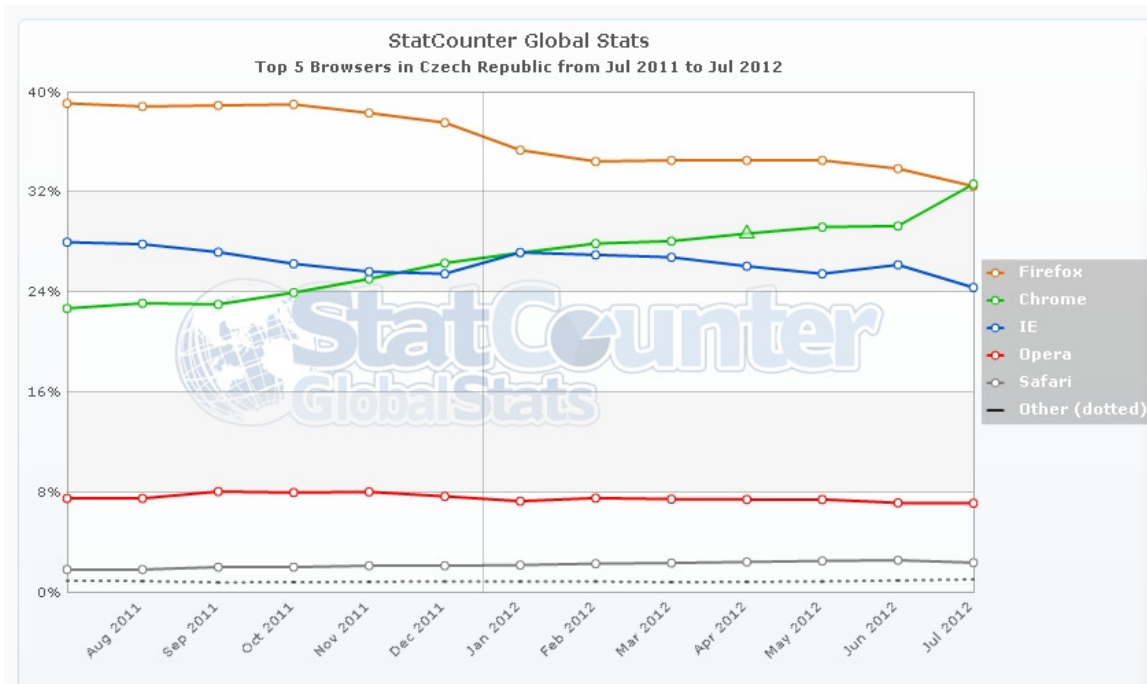
[Ukázka zdrojového kódu](#)

Obrázek 17 - Administrace webu vlaky.uvadi.cz

## 5 Klientská (aplikační) část

V této kapitole se budu zabývat samotnou klientskou částí, tzn. návrhem a implementací aplikace, která bude stahovat, shromažďovat a provádět statistiky o zpoždění vlaků z dat, která jsou dostupná na webu vlaky.uvadi.cz (v serverové části).

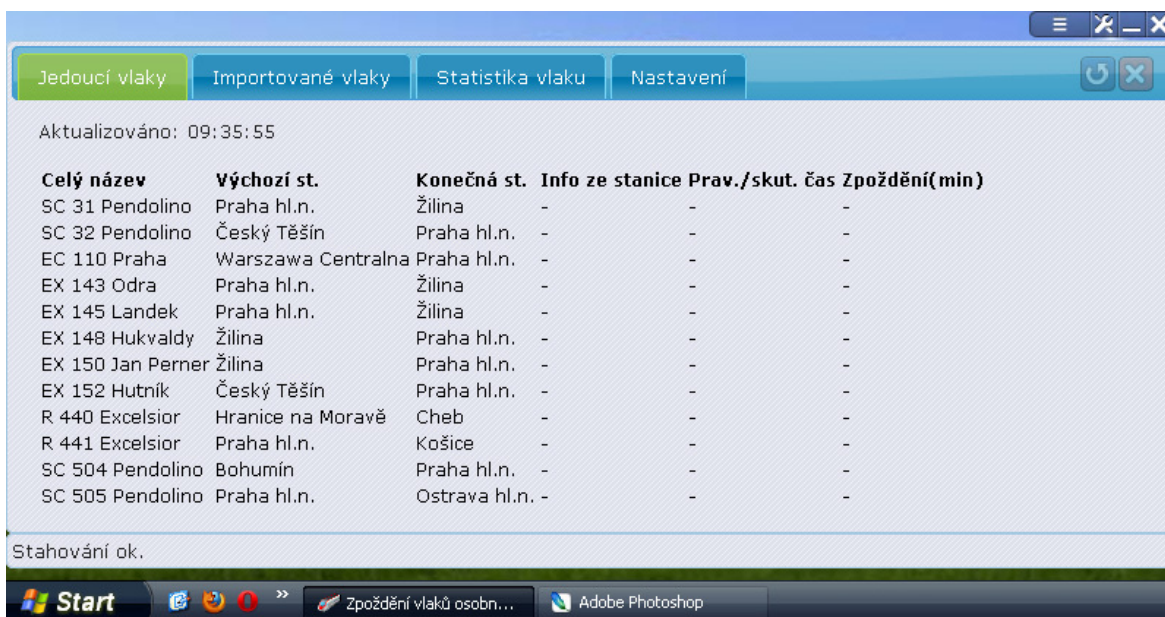
Před vlastní implementací jsem se rozmýšlel pro jaký webový prohlížeč toto rozšíření (aplikaci) navrhnu. Mezi ně jsem uvažoval jen ty nejznámější a nejpoužívanější. To jsou podle serveru StatCounter Global Stats tyto webové prohlížeče: Firefox, Chrome, Internet explorer, Opera a Safari. Důležitými faktory pro výběr byly: nezávislost běhu aplikace na prohlížeči (běh v režimu desktopové aplikace), možnost stahování dat z webu a možnost práce se souborovým systémem uvnitř aplikace. Na základě těchto faktorů jsem tedy zvolil webový prohlížeč Opera, kdy je celá aplikace napsána jako widget (pomůcka) pro tento prohlížeč. [39]



Obrázek 18 - Nejpoužívanější webové prohlížeče v ČR [39]

## 5.1 Opera widget

Widgety jsou webové aplikace běžící na ploše stejně tak jako desktopové (obr. 19). Jejich funkce jsou prováděny na straně klienta za pomoci webových technologií HTML, CSS a JAVASCRIPT, a vytvářejí se velmi podobně jako klasické webové stránky, až na to, že běží v poněkud jiném kontextu. [31]



Obrázek 19 - Běh widgetu na ploše



## 5.2 Packaging widgetu

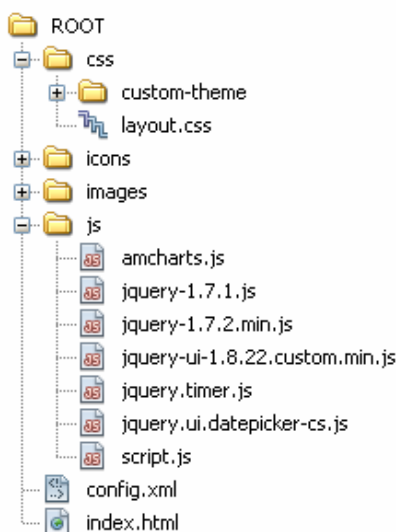
Všechny widgety se instalují do opery prostřednictvím vytvořených balíčků, což jsou soubory s příponou `.wgt`. Tento balíček vytvoříme tak, že zabalíme všechny soubory, které souvisí se strukturou widgetu do ZIP formátu a poté přejmenujeme příponu na `.wgt`. [32]

## 5.3 Adresářová struktura widgetu

V této kapitole čerpáno ze zdroje [31].

Typický Widget obsahuje následující prvky:

- konfigurační soubor - `config.xml`. Jde o soubor v kořenovém adresáři, který obsahuje informace o základním nastavení widgetu, včetně jeho velikosti, názvu, autorovi a bezpečnostních konfigurací.
- soubor index - `index.html`. Stejně jako při klasické webové stránce, tento dokument obsahuje základní kostru. Tudíž obsah widgetu může být vytvořen pomocí libovolných značek, které Opera zpracovává nativně, například HTML, SVG i XML soubory. Tento soubor je také v kořenovém adresáři.
- images soubory – Ty jsou obsaženy většinou v jedné složce `images`.
- javascript soubory – Ty jsou obsaženy ve složce `js`.
- css soubory – Soubory s kaskádovými styly. Ty jsou ve složce `css`.
- icons soubory – Ty jsou obsaženy ve složce `icons`.



Obrázek 20 - Adresářová struktura widgetu

## 5.4 Konfigurační soubor widgetu

V této kapitole čerpáno ze zdroje [31].

Tento konfigurační soubor je nezbytný pro spuštění widgetu. Musí být vždy pojmenován `config.xml`. Obsahuje informace o určitých vlastnostech aplikace, které jsou zapsány pomocí XML značkovacího jazyka. Níže jsou uvedeny některé vlastnosti, které soubor může obsahovat :

- `widgetname` – Jméno widgetu. Jde o povinný údaj.
- `description` – Popis widgetu a k čemu je určen.
- `width` – Rozměr šířky widgetu v pixelech.
- `height` – Rozměr výšky widgetu v pixelech.
- `icon` – Připojení vlastních ikon k widgetu.
- `autor` – Informace o autorovi.
- `id` - Unikátní ID widgetu. Toto číslo se skládá ze tří částí: `host`, `name`, a `revised`.

```
<?xml version='1.0' encoding='utf-8'?>
<widget>
  <widgetname>Zpozdeni vlaku osobni dopravy</widgetname>
  <description>Aplikace slouzi k ziskani online dat pro vybrane
zeleznici trate na uzemi Ceske republiky</description>
  <width>800</width>
  <height>500</height>
  <icon height="128" width="128">icons/128.png</icon>
  <icon height="48" width="48">icons/48.png</icon>
  <icon height="32" width="32">icons/32.png</icon>
  <icon height="16" width="16">icons/16.png</icon>
  <author>
    <name>Jaroslav Demel</name>
    <email>st22120@upce.cz</email>
    <link>http://upce.cz</link>
    <organization>Univerzita Pardubice</organization>
  </author>
  <id>
    <host>upce.cz</host>
    <name>Zpoždění vlaků osobní dopravy</name>
    <revised>2012-02</revised>
  </id>
</widget>
```

Obrázek 21 - Část konfiguračního souboru

## 5.5 Widget a AJAX

Skripty běžící na klasických webových stránkách se nemohou připojovat k libovolným webovým serverům na cizích doménách, ale pouze k té ze které byly načteny. Toto opatření je základním stavebním kamenem každého webu a je tomu tak z důvodu bezpečnostní politiky. [33]

Opera widgety mají liberálnější model zabezpečení, což widgetům umožňuje kontaktovat cizí servery a otevírá tím tak mnoho zajímavých možností. Díky tomu ale také otevírá zadní vrátka k možným útokům, nicméně proto existují způsoby, jak před těmito útoky lze widgety zamknout a zabránit tak zasílání informací do různých webových stránek či je přijímat. [33]

Na obrázku 22 a 23 uvedu příklad jak funguje bezpečnostní politika opery mimo widget. Následující kus kódu je v javascriptu, kde je použita metoda XMLHttpRequest pro čtení z webové stránky na adrese www.opera.com, přičemž samotný skript je načten z jiného umístění.

```
var xhr = new XMLHttpRequest ();  
xhr.open( 'GET', 'http://www.opera.com', false );  
xhr.send();
```

**Obrázek 22 - Inicializace požadavku XMLHttpRequest**

Prohlížeč spustí skript a vydává zprávu v konzoli chyb javascriptu, která je na obrázku 23.

```
JavaScript URL thread: "javascript:var xhr = new XMLHttpRequest..."  
Error:  
name: Error  
message: Security violation
```

**Obrázek 23 - Chybová zpráva v konzoli opery**

Samotná Opera tedy neumožňuje obcházet bezpečnostní politiku, tu lze obejít pouze ve widgetu.

## 5.6 Widget security model

V této kapitole je čerpáno ze zdroje [34].

Bezpečnostní model umožňuje určit, ke kterým protokolům, doménám, a portům se ve widgetu můžeme připojit. Pokud žádná omezení nebyla výslovně definována, můžeme se připojovat ke všem hostitelům a portům. Nicméně, je potřeba prvně povolit přístup k síti.

Ve výchozím nastavení widgetu není přístup k síti povolen. Pro získání přístupu k síti se vše musí nejprve nastavit v souboru config.xml. To se provádí zapsáním atributu network s hodnotami private nebo public, či s oběma hodnotami zároveň.

```
<widget network="public private">  
  ...  
</widget>
```

**Obrázek 24 - Povolení přístupu k síti [34]**

Následující IPv4 rozsahy IP adres jsou definovány jako soukromé (private) adresy :

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255
- 169.254.0.0 - 169.254.255.255

Všechny ostatní adresy jsou definovány jako adresy veřejné (public). Je-li přístup k síti povolen a nic není uvedeno v bezpečnostní sekci (security) platí následující:

- widget se může obrátit na jakéhokoliv hostitele přes protokoly http a https.
- widget nemůže kontaktovat protokoly, jiné než http nebo https, pokud není jinak uvedeno v config.xml.
- widget nemůže přistupovat k souborovému systému použitím file protokolu.
- widget nemůže kontaktovat nestandardní porty v rozmezí 0-1023, s výjimkou portu 80, není-li stanoveno jinak v config.xml.
- widget může využít java appletů, nebo plug-inů.
- widget může přistupovat k místnímu souborovému systému, ale pouze prostřednictvím přímé uživatelem zahájené akce.

### 5.6.1 Element security

Pomocí tohoto elementu nastavujeme již daná omezení pro připojování k síti. Každý <security> element může obsahovat řadu dalších <access> elementů ve kterých se blíže specifikuje co chceme omezit:

- <protocol> - Určuje protokoly které mohou být použity při kontaktování externích serverů. Všechny protokoly kromě protokolu file:// jsou povoleny.

- `<host>` – Určuje které domény mohou být kontaktovány. Jejich názvy musí být přesně zapsány. To znamená, že uvedením domény např. `www.example.com` nelze kontaktovat doménu `example.com`. IP adresy mohou být také použity jako hodnoty.
- `<port>` – Určuje porty. Hodnotou je buď přímo číslo, rozsah čísel oddělených pomlčkou (např. `1024-2048`), nebo čárkami oddělený seznam portů (např. `80, 1337`).
- `<path>` – Určuje část cesty následovanou za doménou.

Pokud některý z těchto elementů není prezentován, původní bezpečnostní model poskytuje přístup ke všem nespécifikovaným hodnotám. Příklad nastavení elementu `<security>` je na obrázku 25.

```

<security>
  <access>
    <host>example.com</host>
    <port>2048-4906</port>
  </access>
  <access>
    <protocol>https</protocol>
    <host>example.org</host>
    <port>80,1337</port>
  </access>
</security>

```

Obrázek 25 - Element security

## 5.7 Widget File I/O

V této kapitole je čerpáno ze zdroje [35].

Samotný javascript běžně neumožňuje přímý přístup k souborovému systému a práci se soubory, ale ve widgetech existuje speciální File I/O API rozhraní, díky kterému lze se souborovým systémem pracovat.

File I / O API se skládá ze tří tříd: `FileSystem`, `File` a `FileStream`.

- `FileSystem` třída je k dispozici přes `opera.io.filesystem`. Toto je virtuální souborový systém. Aby bylo možné ho skutečně používat, je nejprve nutné přidat ze svého aktuálního adresáře souborového systému přípojný bod do virtuálního souborového systému.
- `File` objekt funguje obdobně jako v jiných vývojových prostředích. Tzn., že může ukazovat na adresář, archiv nebo na klasický soubor. Dále zpřístupňuje vlastnosti, jako je `path`, `isDirectory`, `exists`, `parent` apod.

Funguje také jako pole, aby umožnil přístup k souborům a podadresářům v adresáři.

- FileStream třída se používá, když chceme číst nebo zapisovat soubory v souborovém systému. Podporuje čtení a zápis textového souboru, obrázky, binární data a Base64 text.

### 5.7.1 Povolení File I/O

Aby byl souborový systém a jeho metody k dispozici, je potřeba přidat element `feature` a atribut `file` do konfiguračního souboru, tak jak je na obrázku 26.

```
<widget file="yes">...  
<feature name="http://xmlns.opera.com/fileio">  
<param name="folderhint" value="home" />  
</feature>  
... </widget>
```

Obrázek 26 - Element feature

### 5.7.2 Mount points

Namísto přímého přístupu k systému souborů používá API rozhraní koncept přípojných bodů (mount points). Pro přístup k části disku musí být nejprve namountován přípojný bod ve virtuálním souborovém systému. Existují dva typy přípojných bodů:

- Předdefinované systémové přípojný body, které lze aktivovat metodou `io.filesystem.mountSystemDirectory()`.
- Obecné přípojný body vytvořené z libovolného místa na disku uživatele, pomocí metod `opera.io.filesystem.browseForDirectory()` a `opera.io.filesystem.browseForFile()`.

Díky předdefinovaným mount points se lze připojit ke třem adresářům:

- `application` - adresář samotné aplikace widgetu, je připojen pouze pro čtení.
- `storage` – soukromý skladovací adresář, který lze použít pro uložení dočasných souborů a informací o konfiguraci specifické pro aplikaci. Soubory uložené v tomto adresáři přetrvávají až do odinstalování aplikace.
- `shared` - soukromý sdílený adresář. Tento adresář obvykle uživatel zvolí při instalaci aplikace.

V mém případě používám systémový přípojný bod `storage`, který plně postačuje při práci se soubory. Na obrázku 27 je funkce pro zápis souboru v javascriptu uvnitř widgetu.

```
function saveSoubor(soubor,text) {  
    var storage = opera.io.filesystem.mountSystemDirectory("storage");  
    var file = storage.resolve( soubor );  
    var fstream = file.open(file, "w");  
    fstream.writeLine( text );  
    fstream.close();  
}
```

Obrázek 27 - Zápis souboru v javascriptu

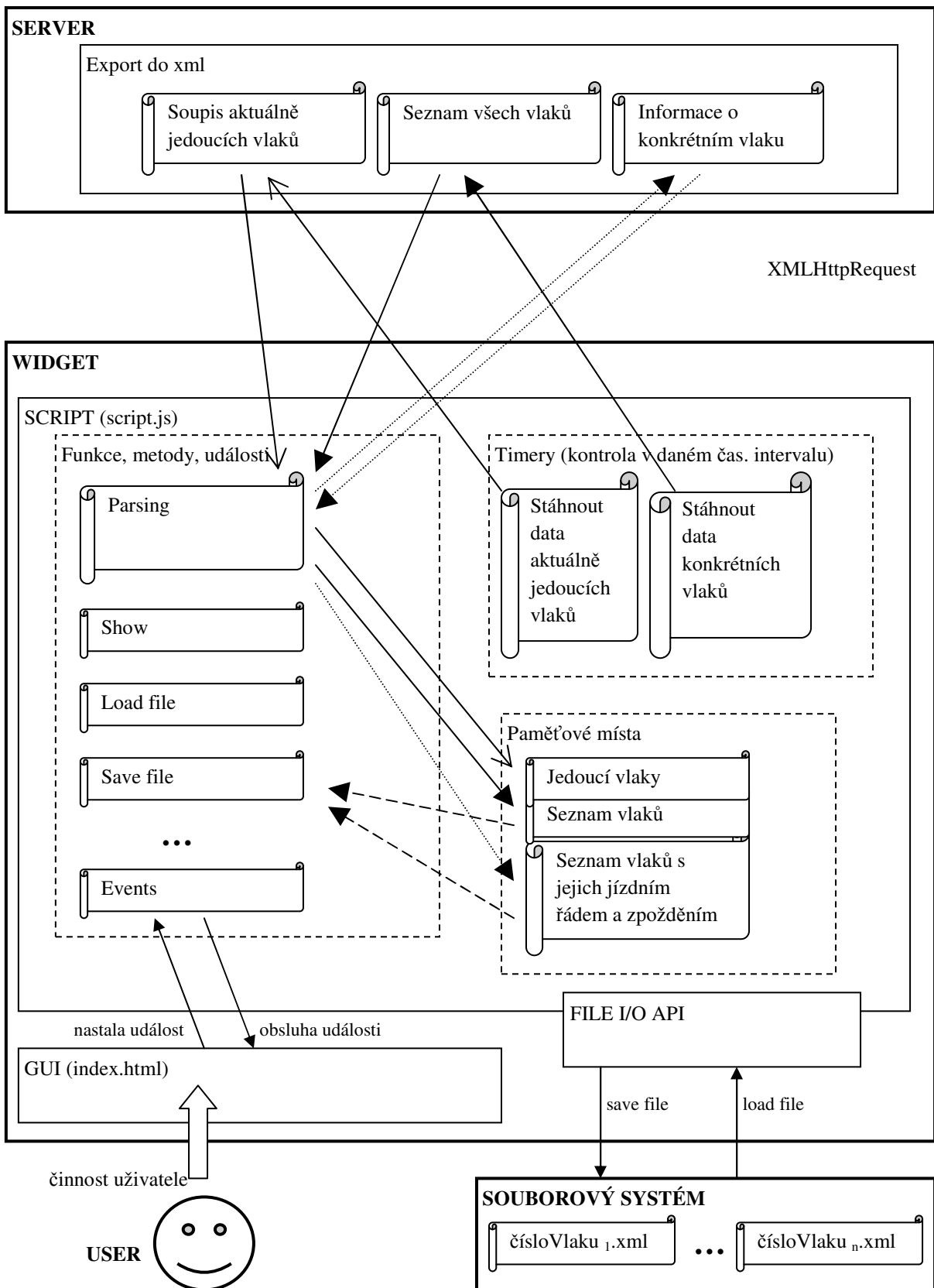
## 5.8 Architektura widgetu

Jak jsem již popsal v kapitole 5.1 a 5.3 celá aplikace (widget) je napsána v jazyce html, kde vstupní branou při spuštění widgetu je dokument index.html. V tomto dokumentu je pomocí tagů napsána základní struktura aplikace, která je rozvržena do pěti divů. Tyto divy jsem nastyloval pomocí jquery-ui-css stylu a vlastního css stylu, který je v souboru layout.css.

Obsah divů je generován a zobrazen pomocí vlastních funkcí a metod napsaných v javascriptu na základě událostí aplikace, tudíž se dá říci že srdcem widgetu je dobře napsaný script. Pomocí samotného scriptu nelze jednoduše ovládat chování a vzhled aplikace, proto je dále rozšířen o tyto moduly:

- jquery - knihovna (sada funkcí), která usnadňuje ovládání dokumentu, umí pracovat s událostmi, DOM elementy, CSS styly.
- jquery timer - modul pro periodické opakování v nastaveném čase.
- jquery ui – modul pro ovládání jquery ui kaskádových stylů, pokud byly použity v dokumentu.
- amcharts – tento modul slouží pro generování grafů.

Činnost widgetu jsem nastínil opět pomocí Rich picture diagramu, který je na obrázku 28.



Obrázek 28 - Architektura widgetu - Rich picture diagram



### 5.8.1 Postup stažení dat aktuálně jedoucích vlaků

Skript widgetu periodicky v nastaveném časovém intervalu (dle uživatele) zasílá XMLHttpRequest požadavek na server s URL<sup>6</sup> na aktuálně jedoucí vlaky pokud jsou data k dispozici jsou parsována, uložena do paměti a po té zobrazena v GUI.

### 5.8.2 Postup stažení dat konkrétních vlaků

Postup je podobný s tím rozdílem, že je nejprve stažen seznam vlaků a po té až všechny konkrétní. Skript widgetu tedy periodicky zasílá XMLHttpRequest požadavek na server s URL na seznam vlaků. Pokud jsou data k dispozici, jsou parsována a tím dostaneme seznam čísel všech vlaků, které jsou na serveru. Po té jsou znovu zaslány XMLHttpRequest požadavky již na konkrétní URL adresy vlaků (podle seznamu čísel), ty jsou také parsována a následně uložena do seznamu v paměti a také do souboru. Po uložení lze v GUI s vlaky pracovat, zobrazovat a provádět základní statistické vyhodnocení.

### 5.8.3 Struktura stažených dat ze serveru

Všechna data se stahují ze serveru již v podobě xml souboru, proto se musí tedy nejprve parsovat pomocí jquery metod. Jak vypadá xml struktura konkrétního vlaku je v příloze E a postup parsování této struktury je v příloze D.

## 5.9 Statistické vyhodnocení výsledků

U všech importovaných vlaků je prováděno pouze základní statistické vyhodnocení, bez pokročilejších statistických metod, což bylo také cílem práce. Tzn. každý vlak má minimální, průměrnou a maximální hodnotu zpoždění na příjezdu či odjezdu v dané stanici za vybrané období. Dále také četnosti v závislosti na délce zpoždění v dané stanici na příjezdu či odjezdu za vybrané období, rozdělené do intervalů tříd, které jsou: 1-5 minut, 6-10 minut, 11-20 minut a 21-30 minut. Vše je vidět níže na obrázku 29, kde je provedeno základní statistické vyhodnocení vlaku SC 30 Pendolino za období 2012/07/22 - 2012/07/23. Součástí přehledu vlaku je také histogram.

Jízdní řád:

| Název stanice   | Příjezd | Odjezd | MIN |     | AVG |     | MAX |     | 1-5 (min) |     | 6-10 (min) |     | 11-20 (min) |     | 21-30 (min) |  |
|-----------------|---------|--------|-----|-----|-----|-----|-----|-----|-----------|-----|------------|-----|-------------|-----|-------------|--|
|                 |         |        | Pří | Odj | Pří | Odj | Pří | Odj | Pří       | Odj | Pří        | Odj | Pří         | Odj |             |  |
| Ostrava hl.n.   | 15:41   |        | 0   | 0   | 0   | 0   | 0   | 0   | 0         | 0   | 0          | 0   | 0           | 0   | 0           |  |
| Ostrava-Svinov  | 15:48   | 15:50  | 0   | 0   | 2   | 0   | 3   | 0   | 1         | 0   | 0          | 0   | 0           | 0   | 0           |  |
| Olomouc hl.n.   | 16:35   | 16:37  | 0   | 0   | 0   | 1   | 0   | 2   | 0         | 1   | 0          | 0   | 0           | 0   | 0           |  |
| Pardubice hl.n. | 17:49   | 17:50  | 0   | 0   | 0   | 0   | 0   | 0   | 0         | 0   | 0          | 0   | 0           | 0   | 0           |  |
| Praha hl.n.     | 18:46   |        | 0   | 0   | 2   | 0   | 3   | 0   | 1         | 0   | 0          | 0   | 0           | 0   | 0           |  |

Zpoždění v období:

| 2012/07/22 |     | 2012/07/23 |     |
|------------|-----|------------|-----|
| Pří        | Odj | Pří        | Odj |
| 0          | 0   | 0          | 0   |
| 3          | 0   | 0          | 0   |
| 0          | 2   | 0          | 0   |
| 0          | 0   | 0          | 0   |
| 3          | 0   | 0          | 0   |

Title: Ostrava-Svinov

Obrázek 29 - Statistika vlaku SC 30 Pendolino v období 2012/07/22 - 2012/07/23

<sup>6</sup> URL - (Uniform Resource Locator) je řetězec znaků s definovanou strukturou, který slouží k přesné specifikaci umístění zdrojů informací (ve smyslu dokument nebo služba) na Internetu. [4]

### 5.9.1 Histogram

Histogram je sloupcový diagram, kde na ose x jsou znázorněny třídy náhodného výběru a na ose y odpovídající četnosti  $m_i$ . Nad každou třídou je sestrojený obdélník, jehož výška odpovídá absolutní, případně relativní četnosti této třídy. [36]

Pod náhodným výběrem o rozsahu  $n$ , utvořeným ze základního souboru, reprezentovaného náhodnou veličinou  $X$ , budeme rozumět  $n$ -tici nezávislých náhodných veličin  $(X_1, X_2, \dots, X_n)$ , majících stejné rozdělení pravděpodobností jako náhodná veličina  $X$ . [36]

V mém případě jsem v histogramu jako třídy vybral jednotlivé délky zpoždění v kterékoliv stanici vybraného vlaku. Ukázka histogramu vlaku R 440 Excelsior je na obrázku 30. U histogramu lze ještě navíc rozlišit data jen na příjezdech, odjezdech nebo na příjezdech i odjezdech současně. Tzn. vlak může mít ve stanici délku zpoždění maximálně 30 minut, proto tříd je také maximálně 30, které jsou na ose x a na ose y jsou odpovídající četnosti k dané třídě.



Obrázek 30 - Histogram vlaku R 440 Excelsior v období 2012/07/22 - 2012/08/08

## 6 Závěr

Cílem této bakalářské práce bylo navrhnout a implementovat aplikaci pro získání online dat pro vybrané železniční tratě na území České republiky s tím, že tato aplikace měla umožňovat zápis získaných dat do XML souborů a statisticky vyhodnocovat výsledky z těchto dat. Dále pak měl být uveden přehled možností pro získání dat online a představeny technologie použité pro vývoj aplikace. Všechny tyto požadavky jsem se snažil co nejlépe splnit s ohledem na přehlednost a jednoduchost.

Na úplném začátku jsem se pokoušel porovnat celkem pět různých webů, které jsem chtěl dále použít jako zdrojová data pro mou aplikaci. Byly to weby Česká železnice, Dopravní web, Babitron, České dráhy a Poloha vlaku. Ačkoliv díky porovnání byl na tom nejlépe web Poloha vlaku, nehodil se pro tento účel ani jeden web, což byl celkem zádrhel. Proto jsem se rozhodl, že web, ze kterého se budou čerpat data navrhnu a implementuji sám, z toho důvodu byla také práce rozdělena na dvě části. První serverovou, která se zabývá právě návrhem webu a druhou aplikační, která se zabývá vlastním návrhem aplikace pro sběr dat.

Před tím než jsem se pustil do serverové části práce, popsal jsem nejprve všechny technologie, které jsem v praktické části použil. Mezi nimi byly HTML, PHP, Ajax, jQuery, XML, CSS, Javascript a taky Mysql.

V serverové části jsem se první snažil charakterizovat základní postupy jak má web fungovat, jejich podmínky a co vše by měl tento systém zvládat. Nezbytnou součástí byla například politika uživatelů, kde jsem popsal jednotlivé role používané pro přihlašování. Nastínil jsem pomocí diagramů Rich picture a UML aktivity jak systém funguje, jaké má vazby a chod. Dále jsem rozebral databázovou část, kde jsem sestavil E-R model. Nechybí ani popis struktury webu a také to na který hosting jsem web umístil.

V druhé klientské části jsem popsal co je to widget opery, po té způsob balíčkování widgetu, adresářovou strukturu, to jak widget pracuje s AJAXEM, security model widgetu, způsob přístupu k souborům ve widgetu, architekturu widgetu a taky jaké základní statistické vyhodnocení jsem použil v praktické části práce.

## Literatura

- [1] Definice informace. KUČEROVÁ, Helena. *Vyšší odborná škola informačních služeb* [online]. 20. 2. 2012 [cit. 2012-03-14]. Dostupné z: <http://web.sks.cz/users/ku/ZIZ/inform1.htm>
- [2] Informace. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001, 5. 2. 2012 [cit. 2012-03-14]. Dostupné z: <http://cs.wikipedia.org/wiki/Informace>
- [3] Data. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001, 8. 1. 2012 [cit. 2012-03-15]. Dostupné z: <http://cs.wikipedia.org/wiki/Data>
- [4] Uniform Resource Locator. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001, 23. 5. 2012 [cit. 2012-03-15]. Dostupné z: <http://cs.wikipedia.org/wiki/URL>
- [5] Syntaktická analýza. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001, 25. 3. 2012 [cit. 2012-03-15]. Dostupné z: <http://cs.wikipedia.org/wiki/Parsing>
- [6] Zpoždění vlaků (aktuální). *Česká železnice* [online]. 2012 [cit. 2012-06-15]. Dostupné z: <http://ceskazeleznice.blog.cz/rubrika/zpozdeni-vlaku-aktualni>
- [7] Aktuální zpoždění vlaků. *Dopravní web* [online]. 2. 7. 2010 [cit. 2012-06-23]. Dostupné z: <http://dopravni.webnode.cz/zeleznicni/aktualni-zpozdeni-vlaku/>
- [8] Zpoždění vlaků. *Babitron* [online]. [cit. 2012-06-27]. Dostupné z: <http://kam.mff.cuni.cz/~babilon/zponline>
- [9] Poloha vlaku. *České dráhy, a.s.* [online]. [cit. 2012-06-28]. Dostupné z: <http://www.cd.cz/cd-online/poloha-vlaku/default.php>
- [10] Aktuální zpoždění vlaků na síti ŽSR. *Poloha vlaku* [online]. [cit. 2012-06-28]. Dostupné z: <http://poloha.vlaku.info/cs/kategorie/EC/>
- [11] CAPTCHA. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-15]. Dostupné z: <http://cs.wikipedia.org/wiki/Captcha>
- [12] HyperText Markup Language. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-15]. Dostupné z: <http://cs.wikipedia.org/wiki/Html>

- [13] Extensible HyperText Markup Language. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-15]. Dostupné z: <http://cs.wikipedia.org/wiki/Xhtml>
- [14] Extensible Markup Language. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-15]. Dostupné z: <http://cs.wikipedia.org/wiki/XML>
- [15] KOSEK, Jiří. XML pro každého: podrobný průvodce. Praha: Grada publishing, 2000. ISBN 80-7169-860-1.
- [16] Kaskádové styly. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-15]. Dostupné z: <http://cs.wikipedia.org/wiki/Css>
- [17] JavaScript. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-15]. Dostupné z: <http://cs.wikipedia.org/wiki/Javascript>
- [18] JavaScript. JANOVSÝ, Dušan. *Jak psát web* [online]. 03. dubna 2012 [cit. 2012-03-15]. Dostupné z: <http://www.jakpsatweb.cz/javascript/>
- [19] AJAX. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-18]. Dostupné z: <http://cs.wikipedia.org/wiki/AJAX>
- [20] JQuery. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-18]. Dostupné z: <http://cs.wikipedia.org/wiki/JQuery>
- [21] Co je to jQuery. JADRNÝ, Tomáš. *JQuery návod - vše okolo jQuery* [online]. [cit. 2012-03-18]. Dostupné z: <http://jquery-navod.cz/kategorie-ostatni-clanky/1-uvodni-clanek>
- [22] PHP. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-18]. Dostupné z: <http://cs.wikipedia.org/wiki/Php>
- [23] PHP. *Tvorba-webu.cz* [online]. [cit. 2012-03-18]. Dostupné z: <http://www.tvorba-webu.cz/php/>
- [24] MySQL. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-18]. Dostupné z: <http://cs.wikipedia.org/wiki/Mysql>
- [25] Document Type Definition. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-20]. Dostupné z: <http://cs.wikipedia.org/wiki/DTD>

- [26] World Wide Web Consortium. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-20]. Dostupné z: <http://cs.wikipedia.org/wiki/W3C>
- [27] Document Object Model. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-20]. Dostupné z: [http://cs.wikipedia.org/wiki/Document\\_Object\\_Model](http://cs.wikipedia.org/wiki/Document_Object_Model)
- [28] Rich picture. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-05-10]. Dostupné z: [http://en.wikipedia.org/wiki/Rich\\_picture](http://en.wikipedia.org/wiki/Rich_picture)
- [29] Diagram aktivít. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-05-10]. Dostupné z: [http://cs.wikipedia.org/wiki/Diagram\\_aktivit](http://cs.wikipedia.org/wiki/Diagram_aktivit)
- [30] Entity-relationship model. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-05-10]. Dostupné z: [http://cs.wikipedia.org/wiki/Entity-relationship\\_model](http://cs.wikipedia.org/wiki/Entity-relationship_model)
- [31] Creating your first Opera widget. *Opera Software* [online]. Tuesday, February 9, 2010 [cit. 2012-07-11]. Dostupné z: <http://dev.opera.com/articles/view/creating-your-first-opera-widget/>
- [32] Packing and deploying your Opera Widget. *Opera Software* [online]. Tuesday, February 9, 2010 [cit. 2012-07-11]. Dostupné z: <http://dev.opera.com/articles/view/packing-and-deploying-your-opera-widget/>
- [33] Opera Widgets and Ajax. *Opera Software* [online]. Tuesday, February 9, 2010 [cit. 2012-07-12]. Dostupné z: <http://dev.opera.com/articles/view/opera-widgets-and-ajax-connecting-to-mu/>
- [34] Opera Widgets security model. *Opera Software* [online]. Tuesday, February 9, 2010 [cit. 2012-07-12]. Dostupné z: <http://dev.opera.com/articles/view/opera-widgets-security-model/>
- [35] File I/O API. *Opera Software* [online]. Tuesday, February 9, 2010 [cit. 2012-07-13]. Dostupné z: <http://dev.opera.com/articles/view/file-i-o-api-for-widgets/>
- [36] KUBANOVÁ Jana. *Matematická statistika*. Pardubice, 1999. ISBN 80-7194-215-4.
- [37] WebML - datové modelování. ZELENKA, Petr. *Interval.cz* [online]. 2004, 08. 01. 2004 [cit. 2012-07-15]. Dostupné z: <http://interval.cz/clanky/webml-datove-modelovani/>
- [38] Kardinalita vztahu. *Studentske.cz* [online]. [cit. 2012-07-15]. Dostupné z: <http://interval.cz/clanky/webml-datove-modelovani/>

[39] *StatCounter Global Stats* [online]. 2012, 2012-06 [cit. 2012-08-01]. Dostupné z: <http://gs.statcounter.com/>

[40] Co Vám přináší webové služby?. *Symbio* [online]. 2006, 4. 7. 2006 [cit. 2012-08-01]. Dostupné z: <http://www.symbio.cz/clanky/co-vam-prinasi-webove-sluzby.html>

## Příloha A – Ukázka webu (serverová část)



### Zpoždění vlaků osobní přepravy

Tato aplikace slouží k zobrazení online dat o zpoždění vlaků na vybraných železničních tratích v ČR. Tyto data byla zde náhodně vygenerována, nezobrazuje skutečný stav vlaků na trati!

Icons by Lukas Software

Role: Administrator, Nick: admin

Aktuální zpoždění jedoucích vlaků | Seznam všech vlaků | Informace o vlaku | **Generátor dat v reál. čase** | **O uživateli**

**VLAK**   | **STANICE**   | **JÍZDNÍ RÁD**   | **UŽIVATEL**

SC 31 Pendolino

Název stanice | Příjezd | Odjezd | Přijezd (min) | Odjezd (min) | zpoždění ze dne 2012/07/22

|                  |       |       |   |   |
|------------------|-------|-------|---|---|
| Praha hl.n.      | 07:12 | 0     | 0 | 0 |
| Pardubice hl.n.  | 08:06 | 08:07 | 0 | 0 |
| Olomouc hl.n.    | 09:19 | 09:21 | 0 | 0 |
| Ostrava-Svinov   | 10:08 | 10:11 | 0 | 0 |
| Ostrava hl.n.    | 10:18 | 10:19 | 0 | 0 |
| Ostrava-Stodolní | 10:22 | 10:23 | 0 | 0 |
| Haviřov          | 10:37 | 10:38 | 0 | 0 |
| Český Těšín      | 10:55 | 10:56 | 3 | 0 |
| Třinec           | 11:03 | 11:04 | 0 | 0 |
| Žilina           | 11:57 | 0     | 0 | 0 |

[Export do XML](#)



## Příloha B – Ukázka generátoru dat (serverová část)



### Generátor dat v reál. čase

Tato část aplikace slouží ke generování dat v reálném čase o zpoždění vlaků na vybraných železničních tratích v ČR. Nejde o skutečný stav vlaků na trati!

11:03:26 2012/08/16 Poslední kontrola: 11:02:54

**Hlavní info**   **Zásobník**   **Databáze**

```
>>>vygenerováno [5], celkové nyní [5]
>>>přidáno do zásobníku s časem [11:06]

>>>11:01=[11:01] [PŘÍJEZD] stanice [817]
>>> Zpoždění vlaku číslo[150] dotud' bylo [0]
>>>vygenerováno [0], celkové nyní [0]
>>>přidáno do zásobníku s časem [10:59]

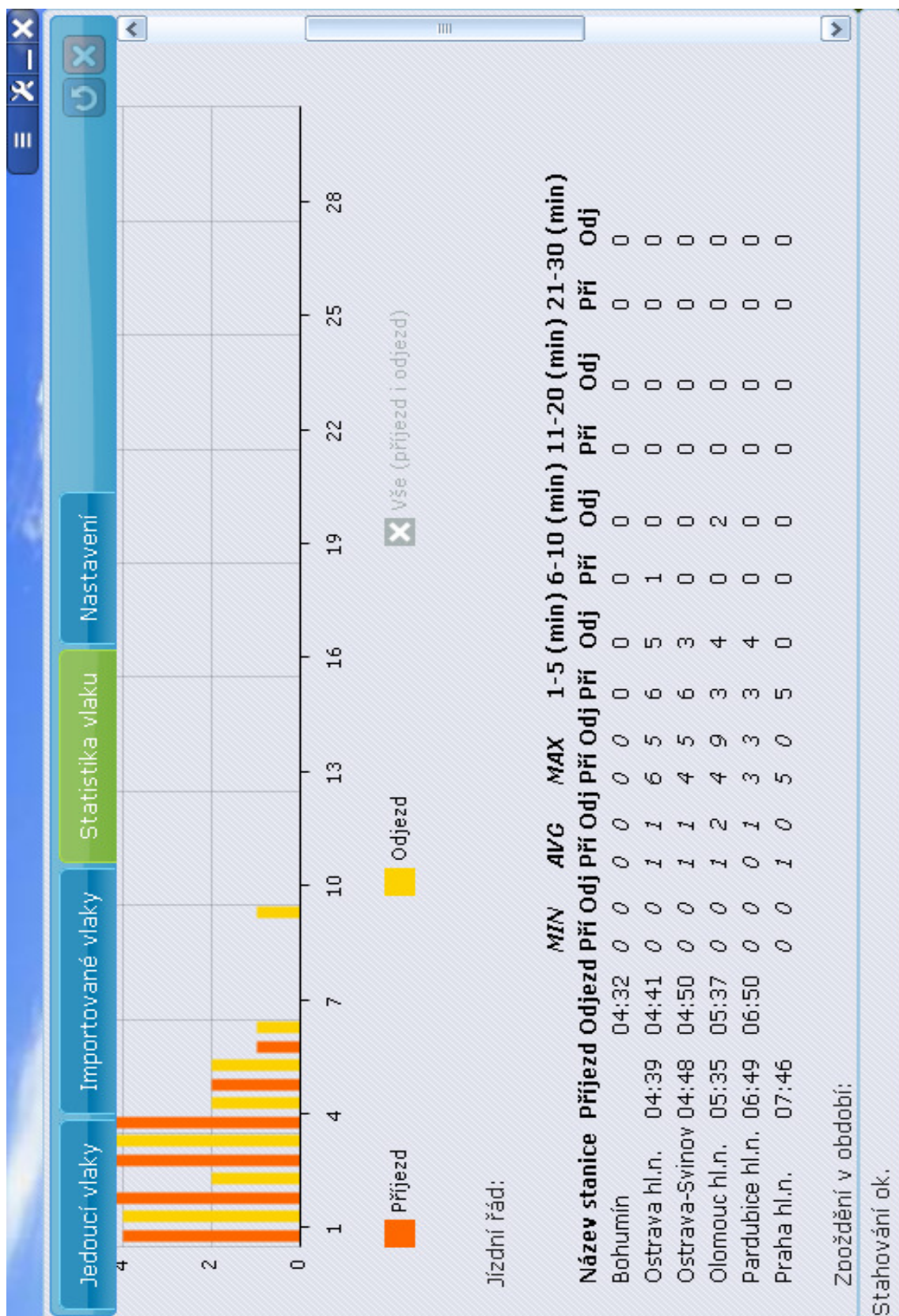
>>>10:59=[10:59] [ODJEZD] stanice [1066]
>>> Zpoždění vlaku číslo=[111] dotud' bylo [0]
>>>vygenerováno [0], celkové nyní [0]
>>>přidáno do zásobníku s časem [10:57]

>>>10:57=[10:57] [PŘÍJEZD] stanice [1066]
>>> Zpoždění vlaku číslo[111] dotud' bylo [0]
>>>vygenerováno [0], celkové nyní [0]
>>>přidáno do zásobníku s časem [10:56]

>>>10:56=[10:56] [ODJEZD] stanice [1039]
>>> Zpoždění vlaku číslo=[111] dotud' bylo [0]
```

Zastav generování   **Vyprázdnit informace**  
Vyber Datum   **Vygenerovat statistiky**

## Příloha C – Ukázka widgetu pro Operu (klientská část)



## Příloha D – Parsing XML struktury

```
$(txt).find("vlak").each(function(){  
    var cislo= $(this).attr("cislo");  
    var kategorie= $(this).find("kategorie").text();  
    var nazevVl= $(this).find("nazev:first").text();  
  
    $(this).find("jizdniRad").each(function(){  
        $(this).find("stanice").each(function(){  
            var poradi= $(this).attr("poradi");  
            var nazev_st= $(this).find("nazev").text();  
            var prijezd= $(this).find("prijezd").text();  
            var odjezd= $(this).find("odjezd").text()  
        });  
    });  
  
    $(this).find("zpozdeni").each(function(){  
        var datum= $(this).attr("datum");  
        $(this).find("stanice").each(function(){  
            var poradi= $(this).attr("poradi");  
            var nazev_st= $(this).find("nazev").text();  
            var prijezd= $(this).find("prijezd").text();  
            var odjezd= $(this).find("odjezd").text()  
        });  
    });  
});
```

## Příloha E – XML struktura

```
<?xml version="1.0" encoding="UTF-8" ?>
<vlak>
<vlak cislo="30">
  <kategorie>SC</kategorie>
  <nazev>Pendolino</nazev>

  <jizdniRad>
    <stanice poradi="1">
      <nazev>Ostrava hl.n.</nazev>
      <prijezd />
      <odjezd>15:41</odjezd>
    </stanice>
    <stanice poradi="2">
      <nazev>Ostrava-Svinov</nazev>
      <prijezd>15:48</prijezd>
      <odjezd>15:50</odjezd>
    </stanice>
    <stanice poradi="3">
      <nazev>Olomouc hl.n.</nazev>
      <prijezd>16:35</prijezd>
      <odjezd>16:37</odjezd>
    </stanice>
    <stanice poradi="4">
      <nazev>Pardubice hl.n.</nazev>
      <prijezd>17:49</prijezd>
      <odjezd>17:50</odjezd>
    </stanice>
    <stanice poradi="5">
      <nazev>Praha hl.n.</nazev>
      <prijezd>18:46</prijezd>
      <odjezd />
    </stanice>
  </jizdniRad>
</vlak>
</vlak>

<zpozdeni datum="2012/07/22">
  <stanice poradi="1">
    <nazev>Ostrava hl.n.</nazev>
    <prijezd />
    <odjezd>0</odjezd>
  </stanice>
  <stanice poradi="2">
    <nazev>Ostrava-Svinov</nazev>
    <prijezd>3</prijezd>
    <odjezd>0</odjezd>
  </stanice>
  <stanice poradi="3">
    <nazev>Olomouc hl.n.</nazev>
    <prijezd>0</prijezd>
    <odjezd>2</odjezd>
  </stanice>
  <stanice poradi="4">
    <nazev>Pardubice hl.n.</nazev>
    <prijezd>0</prijezd>
    <odjezd>0</odjezd>
  </stanice>
  <stanice poradi="5">
    <nazev>Praha hl.n.</nazev>
    <prijezd>3</prijezd>
    <odjezd />
  </stanice>
</zpozdeni>
</vlak>
</vlak>
```