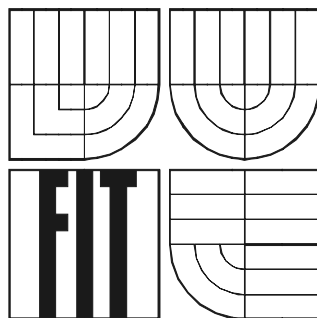


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



# **Elektronický obchod pro prodej letenek**

Ročníková práce

# Elektronický obchod pro prodej letenek

© Martin Brettschneider, 2006.

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

## Prohlášení

Prohlašuji, že jsem tuto ročníkovou práci vypracoval samostatně pod vedením Ing. Martina Vítka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Martin Brettschneider  
19.4.06

## **Abstrakt**

Cílem této práce bylo seznámení se s jazyky a prostředky pro tvorbu informačních systémů, dále seznámení s teorií grafů a s vybranými vhodnými grafovými algoritmy. Nakonec také implementovat elektronický obchod pro prodej letenek, který by vyhledával spoje nejen přímé, ale i spoje s přestupy, pokud by to bylo cenově či časově výhodnější.

## **Klíčová slova**

Informační systém, HTML, PHP, MySQL, WAP, WML, grafové algoritmy

## **Poděkování**

Děkuji Ing. Martinu Vítkovi za poskytnutí odborné pomoci při řešení této ročníkové práce.

## **Abstract**

The target of this project was to study the languages and resources to making the information system, then the theory of graphs algorithms and with suitable graphs algorithms. Next quest was to implement electronic shop for sale the air tickets, which will be able to found channels not only direct, but channels with crossing passages, if it is preferable to channels without crossing passages.

## **Keywords**

HTML, PHP, MySQL, WAP, WML, graph algorithms

# Obsah

Obsah.....	5
Úvod.....	7
1 Tvorba informačního systému .....	8
1.1 Úvodem.....	8
1.2 Návrh systému.....	8
1.2.1 Požadavky a jejich specifikace.....	8
1.2.2 Modely systému .....	8
1.3 HTML.....	9
1.3.1 Úvod.....	9
1.3.2 Tagy.....	9
1.4 PHP.....	10
1.5 SQL, MySQL .....	10
1.6 WAP, WML .....	11
2 Grafy .....	12
2.1 Úvodem.....	12
2.2 Definice grafu.....	12
2.2.1 Neformální definice grafu.....	12
2.2.2 Orientovaný graf.....	13
2.2.3 Kreslení grafů.....	13
2.2.4 Sled, cesta .....	13
2.3 Nejkratší cesty .....	14
2.3.1 Základní fakta.....	14
2.3.2 Dijkstrův algoritmus .....	14
3 Návrh a implementace systému.....	16
3.1 Zadání .....	16
3.2 Neformální specifikace .....	16
3.3 ER Diagram.....	16
3.4 Use Case diagram .....	18
3.5 Implementace .....	19
3.5.1 Úvod .....	19
3.5.2 Zabezpečení .....	19
3.5.3 Vyhledávání .....	19
3.5.4 Wapové rozhraní .....	20
4 Závěr .....	21

Literatura.....	22
Obrazové přílohy.....	23

# Úvod

Internet je fenoménem současnosti. Asi nejpoužívanější službou Internetu je WWW (World Wide Web). Slouží mimo jiné k prohlížení tzv. www stránek, využívajících grafiku, zvuky a často i animace a video. Na těchto www stránkách se dá nalézt množství informací. Ale co to vlastně je informace? Budu citovat z České terminologické databáze knihovnictví a informační vědy.

V nejobecnějším slova smyslu se informací chápe údaj o reálném prostředí, o jeho stavu a procesech v něm probíhajících. Informace snižuje nebo odstraňuje neurčitost systému (např. příjemce informace); množství informace je dáno rozdílem mezi stavem neurčitosti systému (entropie), kterou měl systém před přijetím informace a stavem neurčitosti, která se přijetím informace odstranila. V tomto smyslu může být informace považována jak za vlastnost organizované hmoty vyjadřující její hloubkovou strukturu (varietu), tak za produkt poznání fixovaný ve znakové podobě v informačních nosičích. V informační vědě a knihovnictví se informací rozumí především sdělení, komunikovatelný poznatek, který má význam pro příjemce nebo údaj usnadňující volbu mezi alternativními rozhodovacími možnostmi. Významné pro informační vědu je také pojetí informace jako psychofyzilogického jevu a procesu, tedy jako součásti lidského vědomí (např. N. Wiener definuje informaci jako "obsah toho, co se vymění s vnějším světem, když se mu přizpůsobujeme a působíme na něj svým přizpůsobováním"). V exaktní vědě se např. za informaci považuje sdělení, které vyhovuje přísným kritériím logiky či příslušné vědy. V ekonomické vědě se informací rozumí sdělení, jehož výsledkem může být zisk nebo užitek. V oblasti výpočetní techniky se za informaci považuje kvantitativní vyjádření obsahu zprávy. Za jednotku informace se ve výpočetní technice považuje rozhodnutí mezi dvěma alternativami (0, 1) a vyjadřuje se jednotkou nazvanou bit. (JONÁK)

Pod pojmem informační systém lze chápat zdroj informací. Elektronický obchod pro prodej letenek, který je předmětem mé práce je příkladem takového informačního systému, protože uživatel například získává informace o letech.

V první kapitole mé ročníkové práce se budu zabývat problémy návrhu a implementace informačního systému. Tedy od modelování systému, po jazyky a prostředky pro tvorbu webového informačního systému, jako je HTML, PHP, MySQL či WML.

V druhé kapitole se zaměřím na teorii grafů, tj. vysvětlím základní pojmy nutné pro pochopení a zvládnutí grafových algoritmů a získané dovednosti předvedu na ukázce Dijkstrova algoritmu, který slouží pro vyhledávání nejkratších cest.

V předposlední kapitole popíši několik základních věcí, se kterými jsem se musel potýkat při implementaci elektronického obchodu pro prodej letenek.

Poslední kapitola je věnována konečnému zhodnocení práce.

# 1 Tvorba informačního systému

## 1.1 Úvodem

Informačním systémem rozumíme počítačový firemní systém pro sběr a zpracovávání informací a dat. Informacemi míníme sdělení, které odstraňuje nejistotu nebo nevědomost, daty míníme jakékoli zaznamenané poznatky či fakta. Informaci lze také chápat jako data s nějakým přidaným významem (data + význam).

## 1.2 Návrh systému

### 1.2.1 Požadavky a jejich specifikace

Správný návrh je důležitou částí při tvorbě informačního systému. Základem při návrhu systému je definice požadavků, tj. zadání v přirozeném jazyce, příp. diagramy, udávající požadované služby systému a jeho omezení. Je zpravidla vytvořen na základě informace od zákazníka.

Z definice požadavků se dále vytvoří strukturovaný dokument specifikace požadavků. Tento detailně popisuje služby a omezení systému. Může mimo jiné sloužit pro uzavření kontraktu se zákazníkem.

Dokument požadavků na programové dílo je oficiální dokument o tom, co se od vývojářů požaduje. Je kombinací definice a specifikace požadavků. Dokument by měl být dobře strukturovaný s minimem vzájemných odkazů. Hlavní částí dokumentu jsou modely systému, ukazující vztahy prvků systému a vztahy systému a okolí, definice funkčních požadavků, popisující poskytované služby a definice nefunkčních požadavků, což jsou omezení kladená na produkt a proces vývoje (výkonnost, odezva, standardy, ...). Dalšími částmi dokumentu mohou být rozvoj systému, validační kritéria (třídy testů pro ověření implementace) a další.

### 1.2.2 Modely systému

Cílem modelování je vizualizace (jaký systém je nebo jaký má být), specifikace struktury a chování systému, vytvoření „šablony“ pro konstrukci systému a také dokumentace provedených rozhodnutí. Pro modelování se používají objektově orientované techniky modelování. Typů modelů je celá řada. Mezi hlavní patří model objektové struktury, který zpravidla zahrnuje třídy domény problému a třídy pro implementaci, atributy a operace tříd, statické vztahy mezi třídami a objekty.



Důvodem pro použití dalších modelů je zjišťování a pochopení požadavků na systém, včetně dynamického chování, dále dokumentace v podobě srozumitelné zákazníkovi a dostatečně podrobná informace pro návrháře a programátora.

Mezi další modely systému patří modely chování systému, což jsou modely požadované funkčnosti, použití systému a toků aktivit. Dále jsou to modely interakce objektů, modely stavů objektů a přechodů mezi nimi a modely rozčlenění systému na podsystémy (pro složitější systémy).

Mezi nejpoužívanější metody objektově orientovaného modelování patří jazyk UML. Unified Modeling Language je grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů.

## 1.3 HTML

### 1.3.1 Úvod

HyperText Markup Language (zkráceně HTML) je značkovací jazyk pro tvorbu dokumentů, který definuje vzhled textu (velikost nadpisů, použité fonty písma, okraje, ...). Jazyk HTML byl vyvinut, a neustále se vyvíjí, za účelem publikování dokumentů na World Wide Webu.

Již z názvu je patrné, že původně byl tento jazyk určen pro zobrazování textové dokumentace. Ovšem během krátké doby byla tato funkce nedostačující a jazyk musel být rozšířen např. o různé multimediální prvky, jako jsou grafika, animace, či hudba. Tímto krokem se několikanásobně znásobila jeho hodnota coby média pro přenos informace. V dnešní době již nikoho alespoň minimálně znalého Internetu nepřekvapí, že si může prohlížet fotky z družic, či se podívat na videoklip své oblíbené hudební skupiny. Zvládnutí jazyka HTML je nutné i pro užívání složitějších skriptovacích jazyků, jako je např. PHP.

HTML je interpretovaným jazykem, což znamená, že zdrojový kód je přímo předložen prohlížeči, není tedy překládán do strojového kódu. To znamená, že žádné vnější elementy, jako jsou třeba obrázky či animace, se nenachází uvnitř HTML dokumentu, ale je na ně pouze odkazováno. Jazyk je to zároveň velmi pružný, pokud totiž prohlížeč narazí na chybu, či na příkaz který nezná, prostě ho ignoruje a přesune se na příkaz další.

Jazyk HTML je vlastně základním pilířem při tvorbě webového informačního systému, protože informace, která uživatel hledá, se mu musí zobrazit v přehledné formě, tak aby pro něj byla dobře srozumitelná.

### 1.3.2 Tagy

Jak jsem již zmínil, elementem HTML dokumentu může být téměř cokoli, od prostého textu, po obrázky, či video. Základními elementy HTML dokumentů jsou tzv. tagy. Ty reprezentují prvky,

ze kterých se stránka skládá. Uvnitř těchto elementů mohou být další elementy, ve kterých mohou být další elementy atd.. Tagy by se dali rozdělit na dvě skupiny, mezi párové a nepárové. Párové tagy jsou dvojicí tagů, jako např. `<html> </html>`, kde první je tag otevírací a druhý je uzavírací. Oproti tomu nepárové tagy jsou jen jeden tag, jako například nový řádek `<br>`.

Poslední verzí HTML je 4.01, ale není tomu přesně tak, protože nové verze se vyvíjí pod názvem XHTML (The Extensible HyperText Markup Language). XHTML v dosavadních verzích 1 a 2 používá prakticky stejné tagy a atributy jako HTML.

## 1.4 PHP

Hypertext Preprocessor (zkráceně PHP) je rozšířený univerzální skriptovací jazyk, který je obzvláště vhodný pro vývoj webových aplikací a lze jej zapouzdřit do HTML. Neběží v prohlížeči návštěvníka, ale na serveru. Tímto se podobá Perlu, ale PHP byl vytvořen specificky tak, aby šlo vkládat skript do HTML stránek jednoduchým způsobem. Toto je obrovská výhoda jazyka PHP, protože můžete v podstatě libovolně přecházet s HTML kódem do skriptu jazyka PHP a naopak. PHP kód je uzavřen mezi zvláštními uvozujícími a koncovými tagy (`<?php, ?>`), které nám umožňují vstoupit a opustit "PHP mód".

Jak jsem se již zmínil, PHP je vykonáván na straně serveru, to má mimo jiné tu výhodu, že klient obdrží pouze výstup ze skriptu, bez možnosti zjistit, jaký může být použit zdrojový kód.

S PHP nejsme omezeni pouze na výstup HTML. Schopnosti PHP zahrnují výstup obrázků, souborů PDF a dokonce Flash animací (použitím libswf a Ming) generovaných za běhu. Výstupem může být rovněž libovolný text jako třeba XHTML nebo jakýkoli jiný XML soubor. PHP umí tyto soubory nejen vypisovat, ale také automaticky generovat a ukládat je do souborového systému, čímž vznikne cache dynamického obsahu na straně serveru.

Jedna z nejsilnějších a nejvýznamnějších vlastností PHP je jeho podpora pro širokou škálu databází. Vytvoření webové stránky spolupracující s databází je neuvěřitelně jednoduché. Podporovaných databází je velké množství, pro tento projekt je zvláště důležitá podpora databáze MySQL.

## 1.5 SQL, MySQL

SQL (Simple Query Language nebo Structured Query Language) je standardní dotazovací jazyk pro práci s databázemi. To znamená, že jej můžeme využít na různých platformách a s různými databázemi. Na internetu může být takovou databází MySQL server. SQL se skládá z mnoha příkazů, které se velmi podobají klasickému jazyku - angličtině.

Na webu se velmi často používají databáze, i zde se ke komunikaci používá SQL. SQL příkazy se vkládají nejdříve do PHP, které pak komunikuje s databází, třeba MySQL.

Je důležité rozlišit SQL a MySQL. SQL je jazyk, oproti tomu MySQL je program, který server využívá k ovládní databáze. SQL je tedy jazyk, kterým „působíme“ na databázi MySQL. Pod pojmem databáze rozumíme jeden nebo více datových souborů, kterým se říká tabulky. Příkazům jazyka SQL se říká dotazy (query). Dotazy dokážeme například získat data z databáze, provést úpravu dat, vytvořit novou tabulku, či novou databázi a jiné.

## 1.6 WAP, WML

Digitální bezdrátová zařízení jako jsou mobilní telefony se v průběhu let stala velmi populárními. Technicky řečeno, mobilní telefony již nejsou pouze telefony - staly se komunikačními zařízeními schopnými spouštět aplikace a komunikovat s dalšími zařízeními a aplikacemi přes bezdrátové sítě.

Wireless Application Protocol jako standard specifikuje dva zásadní elementy bezdrátové komunikace a to end-to-end aplikační protokol a aplikační prostředí založené na prohlížeči. Aplikační protokol je vrstva komunikačního protokolu, který je obsažen v každém WAP zařízení. Na straně sítě obsahuje serverové komponenty, implementující další konec protokolu, který je schopen komunikace s jakýmkoliv WAP zařízením. Často serverové komponenty vstupují do role brány routující požadavky WAP zařízení na aplikační server. Brána může být fyzicky posazena buď do telefonní sítě nebo počítačové sítě, vytvářejíce tím most mezi oběma sítěmi.

WAPová aplikace se skládá z aplikačního serveru a klientské aplikace, kterou brána stahuje z aplikačního serveru do WAP zařízení pro použití. Standardní aplikační prostředí je potřeba pro běh stejné klientské aplikace na jiném WAP zařízení. WAP poskytuje standard skládající se z prohlížeče a skriptu. Prohlížeč je velice podobný webovému prohlížeči a dokáže zpracovávat obsah popsáný ve Wireless Markup Language. Prohlížeč také obsahuje vestavěný skript pro běh aplikací ve WAP zařízení. Tyto aplikace jsou psány ve skriptovém jazyce zvaném WMLScript. Jako dodatek k programovacímu jazyku jako takovému, skript také implementuje množinu knihoven, které umožňují aplikaci přístup k různým službám pro WAP zařízení. WML a WMLScript jsou vytvořeny pro bezdrátové použití, narrowband sítě a jsou binárně kódovány pro optimální efektivitu přenosu.

## 2 Grafy

### 2.1 Úvodem

Pojem grafu je přirozeným prostředkem k vyjádření párových vztahů, tj. vztahů mezi dvojicemi. Obecnější vztahy (už např. mezi trojicemi) lze nakreslit podstatně obtížněji. Snad proto se těmto obecnějším vztahům mnohdy vyhýbáme a raději volíme jiný, komplikovanější popis téže situace, založený na vztazích párových.

Proto se nelze divit tomu, že pojem grafů (a způsob uvažování, který se označuje jako grafový) je poměrně oblíbený a rozšířený.

Některé aplikace grafů jsou zcela zjevné: daný problém „ze života“ se přeformuluje na grafovou úlohu, udělá se příslušný graf, který modeluje konkrétní situaci, grafová úloha se nějakým známým postupem vyřeší a získané řešení se přeloží z řeči grafů zpátky „do života“.

Umění aplikovat grafy zahrnuje dvě dílčí znalosti a dovednosti. Především je třeba umět správně sestavit (vymyslet) graf, kterým přesně modeluje danou situaci (tj. jsou v něm zachyceny vztahy, o které nám jde).

Potom je také třeba umět řešit alespoň ty nejdůležitější grafové úlohy, vědět, které jsou rychle a snadno řešitelné a které jsou naopak těžké. Také není špatné vědět, jak nějakou neznámou úlohu převést na úlohu, kterou řešit dovedeme.

Pro pochopení algoritmů pro hledání nejkratší cesty si zavedeme některé významné pojmy.

### 2.2 Definice grafu

#### 2.2.1 Neformální definice grafu

Graf se skládá z tzv. vrcholů a tzv. hran. Hrana vždy spojuje dva vrcholy a je buď orientovaná nebo neorientovaná. U hran orientovaných rozlišujeme vrchol počáteční a vrchol koncový a říkáme, že hrana vede z počátečního do koncového vrcholu. Za neorientované hrany bereme symetrické spojení dvou vrcholů.

Pokud hrana spojuje nějaký vrchol se sebou samým, nazýváme ji smyčkou.

Dále se budeme zabývat pouze grafy orientovanými, protože právě ty jsou důležité pro řešení úloh o nejkratších cestách.

## 2.2.2 Orientovaný graf

Orientovaný graf je trojice  $G = (V, E, \varepsilon)$ , kde  $V$  je neprázdná konečná množina, jejíž prvky nazveme vrcholy,  $E$  je konečná množina, jejíž prvky nazveme orientovanými hranami, a  $\varepsilon$  je zobrazení  $\varepsilon: E \rightarrow V^2$ , které nazveme vztahem incidence. Toto zobrazení přiřazuje všem hranám  $e \in E$  uspořádanou dvojici vrcholů  $(x, y)$ . První z nich,  $x$ , se nazývá počátečním vrcholem hrany a značíme jej  $Pv(e)$ . Druhý,  $y$ , nazýváme koncovým vrcholem hrany a značíme jej  $Kv(e)$ .

O hraně  $e$  říkáme,  $e$  vede z vrcholu  $x$  do vrcholu  $y$  nebo také, že spojuje vrcholy  $x$  a  $y$ . O vrcholech  $x, y$  pak říkáme, že jsou incidentní s hranou  $e$  a naopak, že hrana  $e$  je incidentní s vrcholy  $x, y$ . Oba vrcholy také můžeme souhrnně nazvat krajními vrcholy hrany  $e$ .

Jestliže  $Pv(e) = Kv(e)$ , pak hranu  $e$  nazýváme (orientovanou) smyčkou. Vrchol, který není incidentní s žádnou hranou, nazveme izolovaným vrcholem.

Je také možné, aby několik hran mělo stejné počáteční i koncové vrcholy, tj. aby pro hrany  $e_1, e_2$ , kde  $e_1 \neq e_2$ , platilo  $Pv(e_1) = Pv(e_2)$  a  $Kv(e_1) = Kv(e_2)$ . O takových hranách mluvíme jako o rovnoběžných nebo násobných.

Množina hran grafu může být oproti množině vrcholů prázdná.

## 2.2.3 Kreslení grafů

Grafy lze samozřejmě znázorňovat kreslením. Vrcholy obvykle kreslíme jako body (kroužky), hrany jako čáry (úsečky, či oblouky), které popisují příslušné dvojice vrcholů. U orientovaných hran značíme orientaci šipkou od počátečního do koncového vrcholu.

Protože lze graf nakreslit v podstatě nekonečně mnoha způsoby, snažíme se ho nakreslit tak, aby se co nejméně hran křížilo.

## 2.2.4 Sled, cesta

Posloupnost vrcholů a hran  $v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k$  nazýváme orientovaným sledem, jestliže pro každou hranu  $e_i$  z této posloupnosti platí  $Pv(e_i) = v_{i-1}$  a  $Kv(e_i) = v_i$ .

Orientovaný sled, v němž se neopakuje žádný vrchol, nazýváme orientovanou cestou. Z toho vyplývá, že na cestě se neopakují ani hrany.

## 2.3 Nejkratší cesty

Úlohy o nejkratších cestách patří asi k nejčastěji aplikovaným úlohám teorie grafů.

### 2.3.1 Základní fakta

Mějme orientovaný graf  $G$ , jehož každá hrana  $e \in E(G)$  je ohodnocena reálným číslem  $a(e)$ , které nazýváme délkou hrany. Délkou cesty je součet délek jednotlivých hran tvořících cestu. Jsou-li  $x, y$  dva vrcholy grafu, pak vzdálenost  $u(x, y)$  z vrcholu  $x$  do vrcholu  $y$  definujeme jako délku nejkratší cesty z  $x$  do  $y$ . Pokud cesta neexistuje, definujeme vzdálenost  $u(x, y) = \infty$ .

Délku nejkratší dosud nalezené cesty z počátečního vrcholu  $r$  do vrcholu  $x$  definujeme jako  $U(x)$ , kde  $U(x) = \infty$ , pokud jsme zatím žádnou cestu z  $r$  do  $x$  nenašli.

Dále budeme potřebovat zjistit nejen vzdálenosti, tj. délky nejkratších cest, ale také kudy nejkratší cesty vedou. To zajistíme tak, že pokaždé, když dojde ke snížení hodnoty  $U(v)$  pro nějaký vrchol  $v$ , zaznamenáme si hranu, která snížení způsobila. K zapamatování použijeme hodnotu  $ODKUD(v)$ . Na počátku výpočtu nechť není hodnota  $ODKUD$  definována pro žádný vrchol  $v$ .

### 2.3.2 Dijkstrův algoritmus

Tento algoritmus jsem použil při implementaci vyhledávání spojů, proto ho podrobně vysvětlím.

Vstupem Dijkstrova algoritmu je graf  $G$ , výchozí vrchol  $r$  a ohodnocení hran  $a : E(G) \rightarrow \mathbb{R}$  takové, že pro všechny hrany platí  $a(e) \geq 0$ .

Výstupem jsou pro všechny vrcholy  $v$  hodnoty  $U(v)$  a  $ODKUD(v)$ .

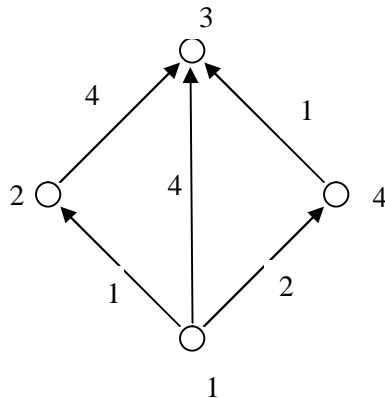
Pomocnou proměnnou bude množina vrcholů  $D$  taková, že pro všechny  $v \in D$  bude hodnota  $U(v)$  rovna vzdálenosti  $u(r, v)$ .

Algoritmus:

1. Inicializace:  $U(r) = 0$ , pro všechna  $v \neq r$  položíme  $U(v) = \infty$  a dále  $D = \emptyset$ .
2. Výběr vrcholu: Jestliže pro všechny vrcholy  $v \in (V \setminus D)$  platí  $U(v) = \infty$ , výpočet končí. Jinak z množiny  $V \setminus D$  vybereme vrchol  $x$ , který má nejnížší hodnotu  $U(x)$ , zařadíme jej do množiny  $D$  a pokračujeme krokem 3.
3. Zpracování hran vycházejících z vrcholu  $x$ : Pro každou hranu  $e$ , která vychází z vrcholu  $x$ , provedeme krok 4 a po zpracování všech hran pokračujeme dle kroku 2.
4. Položíme  $y = Kv(e)$ . Jestliže platí  $U(x) + a(e) < U(y)$ , pak provedeme  $U(y) = U(x) + a(e)$  a  $ODKUD(y) = e$ .

Pokud nás zajímá nejkratší cesta pouze do jednoho vrcholu, lze algoritmus ukončit, jakmile cílový vrchol přidáme do množiny  $D$ .

Nyní uvedu krátký příklad použití Dijkstrova algoritmu. Graf a ohodnocení jeho hran je vyjádřeno na následujícím obrázku.



Průběh výpočtu bude tento:

	$U(1) = 0$	$D = \emptyset$
$x = 1$		$D = \{1\}$
	$U(2) = 1$	
	$U(3) = 4$	
	$U(4) = 2$	
$x = 2$		$D = \{1, 2\}$
	$U(3)$ beze změny	
$x = 4$		$D = \{1, 2, 4\}$
	$U(3) = 3$	
$x = 3$		$D = \{1, 2, 3, 4\}$

A výsledek výpočtu lze zahrnout do následující tabulky:

vrchol $x$	1	2	3	4
vzdálenost $U(x)$	0	1	3	2
ODKUD( $x$ )	-	(1,2)	(4,3)	(1,4)

Z tabulky můžeme například vyčíst, že nejkratší cesta do bodu 3 je přes bod 4 a velikost této cesty je rovna hodnotě 3.

## 3 Návrh a implementace systému

### 3.1 Zadání

Mým úkolem bylo implementovat systém, který bude umět vyhledávat spoje nejen přímé, ale rovněž navrhnout zákazníkovi spoj s přestupy, je-li to cenově, časově nebo jinak výhodnější nebo v případě neexistence přímého spoje. Systém také musí kromě standardního webového rozhraní rovněž nabízet rozhraní wapové.

### 3.2 Neformální specifikace

Nefunkční požadavky kladené na systém jsou následující. Systém pracuje prakticky v jakémkoli operačním systému, podmínkou je pouze funkční připojení k internetu a internetový prohlížeč. Tím pádem nejsou kladeny žádné zvláštní nároky na hardware počítače.

Funkčními požadavky jsou tyto. Kompletní správa letadel, letišť, linek, tzn. přidávání, editování a mazání z databáze. Dále správa uživatelských účtů (např. změna práv uživatele), správa letů (tzn. generování letů na určitou dobu dopředu, příp. mazání vygenerovaných letů). Uživatelé mohou vyhledávat v letech a to spoje nejen přímé, ale i spoje s přestupy (pokud to bude cenově, časově, či jinak výhodnější nebo v případě, že přímý spoj neexistuje). Tyto vyhledané lety si pak mohou zakoupit ve formě letenky. Zakoupené letenky si uživatel může prohlížet nejen přes html, ale také na svém mobilním telefonu přes wap.

### 3.3 ER Diagram

Kompletní ER diagram je uveden v obrazové příloze (viz. Obrazová příloha: Obrázek 1), zde uvedu pouze některé zajímavé tabulky.

Jako první uvedu tabulku Uživatel (viz. Obrázek 3.1). Zde si můžeme všimnout atributu (sloupce) Práva, která slouží k rozdělení práv mezi uživatele. Neregistrovaný uživatel (tedy uživatel, který ještě nedokončil registraci použitím odkazu v registračním e-mailu) má tuto hodnotu nastavenou na 0. Registrovaný uživatel má tuto hodnotu rovnu 1 a administrátor má v právech hodnotu 2. Všechny ostatní atributy jsou podle názvu celkem jasné, snad jen atribut Registrace je datum zaregistrování uživatele.





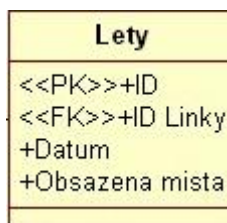
**Obrázek 3. 1**

Jako další uvedu tabulku Linky (viz Obrázek 3.2). Tato tabulka je jednou z nejdůležitějších v mém informačním systému. Linka je určena jednoznačným identifikátorem ID. Jako další atribut obsahuje identifikátor letiště, ze kterého linka odlétá, potom identifikátor letiště do kterého linka létá. Atribut ID letadla je identifikátor letadla, které k této lince přísluší. Atribut Dny je výčtovým typem, který určuje, které dny tato linka létá, jeho hodnoty nabývají hodnoty od 0 .. 6 a odpovídají dnům od neděle do soboty (například (1,3,5) znamená, že linka létá v pondělí, ve středu a v pátek). Atribut Čas je časový údaj o tom, kdy Linka startuje (např. 13:30), tento čas je uveden v středním čase (tj. GMT 0). Další dva atributy jsou již jasné z jejich názvu (Doba letu, Cena).



**Obrázek 3. 1**

Poslední tabulku, kterou uvedu bude tabulka Lety (viz. Obrázek 3.3). Lety jsou opět určeny jednoznačným identifikátorem. Atribut ID Linky určuje, ke které lince let náleží. Oproti linkám se liší tím, že lety jsou již určeny přesně na nějaké Datum (tzn., že není uvedeno, že linka létá v neděli, ale že poletí 7.5.2006).



Obrázek 3. 2

## 3.4 Use Case diagram

Kompletní Use Case Diagram je uveden v obrazové příloze (viz. Obrazová příloha: Obrázek 2). Zde popíši jednotlivé role uživatelů slovně.

Základním uživatelem systému je neregistrovaný uživatel. Tento má velmi omezená práva. Může pouze vyhledávat spoje. Pokud by si tento uživatel chtěl objednat letenku, či prohlížet zakoupené letenky, musí se buď zaregistrovat nebo se do systému přihlásit (pokud již je registrován). Registrace probíhá tak, že uživatel vyplní požadované údaje a poté mu na zadaný e-mail přijde aktivační zpráva.

Dalším aktérem v systému je registrovaný uživatel. Ten již může nejen vyhledávat letenky, ale může si letenku i objednat, popřípadě objednanou letenku zrušit. Dále si samozřejmě může objednané letenky prohlížet a to nejen přes standardní webové rozhraní, ale také přes svůj mobilní telefon pomocí rozhraní WAP.

Posledním aktérem v systému je administrátor. V systému má stejná práva jako registrovaný uživatel, ale navíc se stará o správu systému. Může spravovat uživatelské účty (měnit práva jednotlivých uživatelů, mazání uživatelů z databáze), dále spravuje letadla a letiště (ty může přidávat nové, editovat a mazat existující z databáze). Jeho další rolí v systému je správa linek a letů. Lety může generovat do systému a posléze může vygenerované lety ze systému odstranit, pokud nějaký automaticky vygenerovaný den, let nepoletí (např. v případě státního svátku apod.).

## 3.5 Implementace

### 3.5.1 Úvod

Standardní webové rozhraní je implementováno v jazycích HTML, PHP a MySQL. Alternativní wapové rozhraní je implementováno v jazyce WML, PHP, MySQL.

Kombinace tří jazyků (respektive dvou, protože pro dotazy jazyka MySQL se využívá funkcí jazyka PHP) v jednom zdrojovém kódu je někdy značně nepřehledná. Proto jsem se snažil projekt rozdělit na co nejvíce modulů, aby se co nejvíce tento zdrojový kód zpřehlednil.

Uživatelská i administrátorská část má svoji hlavní stránku (index.php), z níž se volají příslušné funkce z jiných souborů (modulů). Která funkce se má zavolat se pozná podle proměnné \$\_GET["main"], která určuje příslušnou sekci. Tedy například <http://letenka.wz.cz/index.php?main=1> znamená, že se má zavolat funkce pro zobrazení formuláře pro hledání letů.

### 3.5.2 Zabezpečení

Zabezpečení jednotlivých částí systému, tedy kontrola uživatelských práv je pomocí proměnné \$\_SESSION. Při přihlášení uživatel musí zadat svoje uživatelské jméno a heslo. Pokud přihlášení proběhne v pořádku, nastaví se tři hodnoty do sessions, je to jednoznačná identifikace uživatele (id), přezdívka uživatele (uživatelské jméno, nick) a práva tohoto uživatele. Proměnné v sessions zanikají, pokud se uživatel odhlásí, či ukončí internetový prohlížeč, ve kterém se na stránky přihlásil.

Kontrola práv probíhá tak, že pokud chce uživatel například vstoupit do administrátorské sekce zkontroluje se \$\_SESSION["prava"] a pokud jsou dostatečně vysoká (tedy administrátorská) vstup se uživateli povolí. Obdobným způsobem funguje zabezpečení u objednávání, prohlížení či rušení letenek.

### 3.5.3 Vyhledávání

Vyhledává se na základě formuláře (viz. Obrazová příloha: Obrázek 3), ve kterém uživatel vyplní město odkud a kam a který den by chtěl letět. Jako nepovinné je datum zpáteční cesty a způsob řazení výsledku (standardně je podle ceny).

Pokud je formulář správně vyplněn, zahájí se vyhledávání podle Dijkstrova algoritmu (viz. Kapitola 2.3.2). Pomocí tohoto algoritmu se najde spoj, který nejvíce vyhovuje vstupnímu formuláři. Ovšem algoritmus najde pouze jeden spoj, proto se musí volat tak dlouho, dokud již nevrací žádné další spoje. Příklad výsledku hledání viz. Obrazová příloha: Obrázek 4.

Při vyhledávání se také musí dbát na časové pásma. V systému jsou všechny časové údaje uvedené v GMT 0, při zobrazování výsledku hledání se pak přičítá resp. odečítá časové pásmo místa, ze kterého uživatel právě vyhledává.

### 3.5.4 Wapové rozhraní

Jak jsem již uvedl, Wapové rozhraní je implementováno v jazyce PHP a WML. Přes wapové rozhraní si uživatel může prohlížet zakoupené letenky, tj. slouží k tomu, že když zákazník už čeká na letišti a nemá přístup k počítači, může si i tak přes mobilní telefon zkontrolovat čas odletu (viz. Obrázek 3.4).



Obrázek 3. 4

## 4 Závěr

Již dříve jsem implementoval informační systém, ale toto byla moje první zkušenost při tvorbě webového informačního systému. Při řešení projektu jsem se důkladně naučil používat jazyky a prostředky nutné pro tvorbu webových informačních systémů jako je HTML, PHP, MySQL či WML. Dále jsem prohloubil své znalosti ohledně návrhu a modelování informačního systému. Svoje znalosti jsem také obohatil o teorii grafů a práci s grafovými algoritmy.

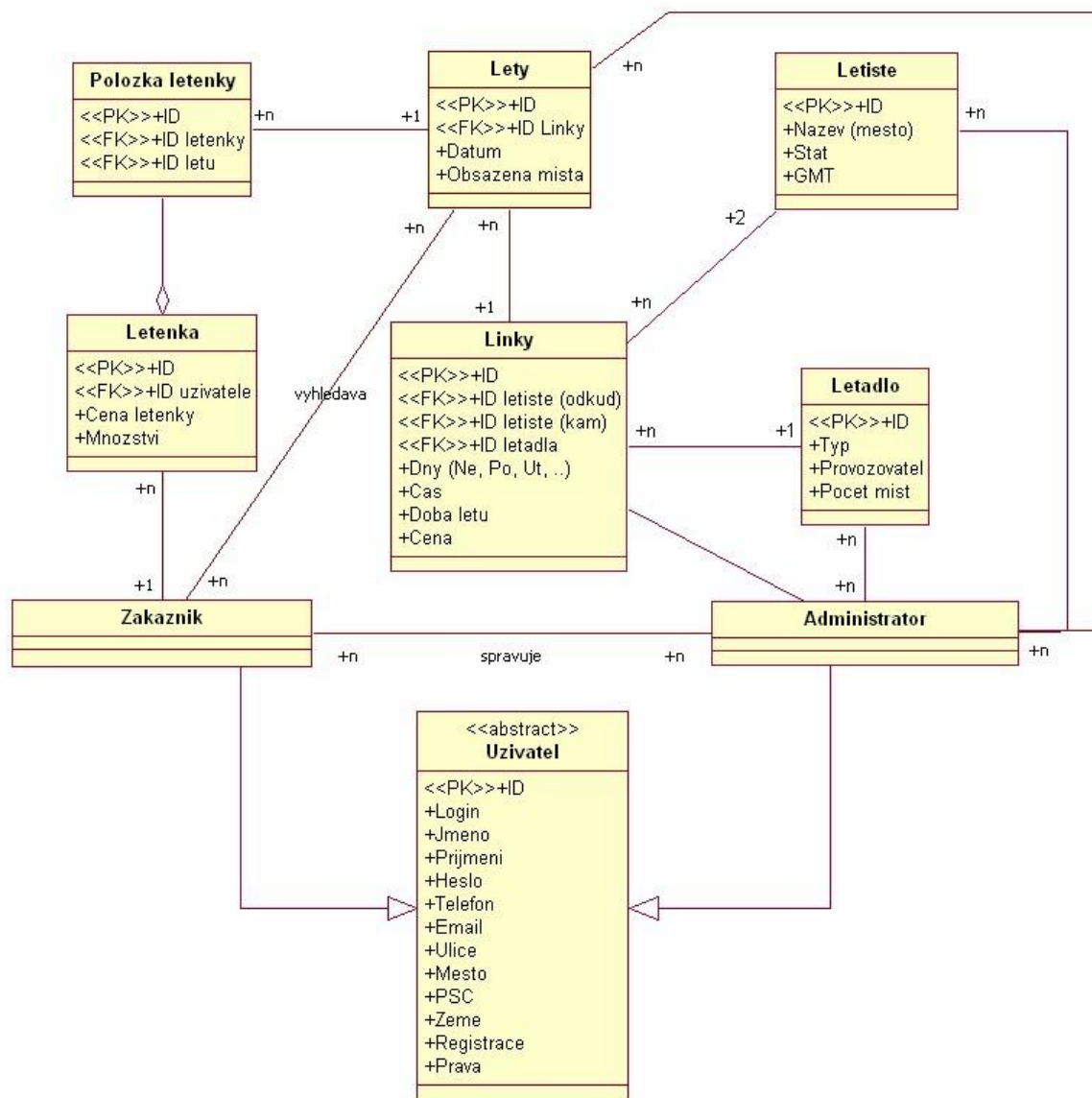
Kdybych měl srovnat svůj elektronický obchod pro prodej letenek s již existujícími systémy, musel bych říct, že by asi na trhu neuspěl. Využít by ho mohla snad jen nějaká malá letecká společnost. V reálném světě totiž většina leteckých společností tvoří jakési korporace a když vyhledáváte u nějaké společnosti lety, které tato společnost nenabízí, nabídne Vám lety od jiné společnosti, se kterou je v této korporaci. Databáze těchto společností pak musí být jednotné, resp. musí mít jednotnou strukturu, tak aby mohlo být vyhledávání úspěšné i na stránkách jiné společnosti. Ovšem u malé společnosti, která by například provozovala jenom vnitrostátní lety, by mohl můj systém nalézt uplatnění.

V tomto projektu bych rád pokračoval v rámci diplomové práce. Systém by totiž potřeboval vylepšit jak v administrátorské části, tak v části uživatelské. Například implementaci vyhledávacího algoritmu bych chtěl celkově přetvořit a zřejmě úplně změnit jazyk použitý k vyhledávání. Než jazyk PHP by byl zřejmě vhodnější JavaScript, který přece jenom nabízí větší programovací možnosti.

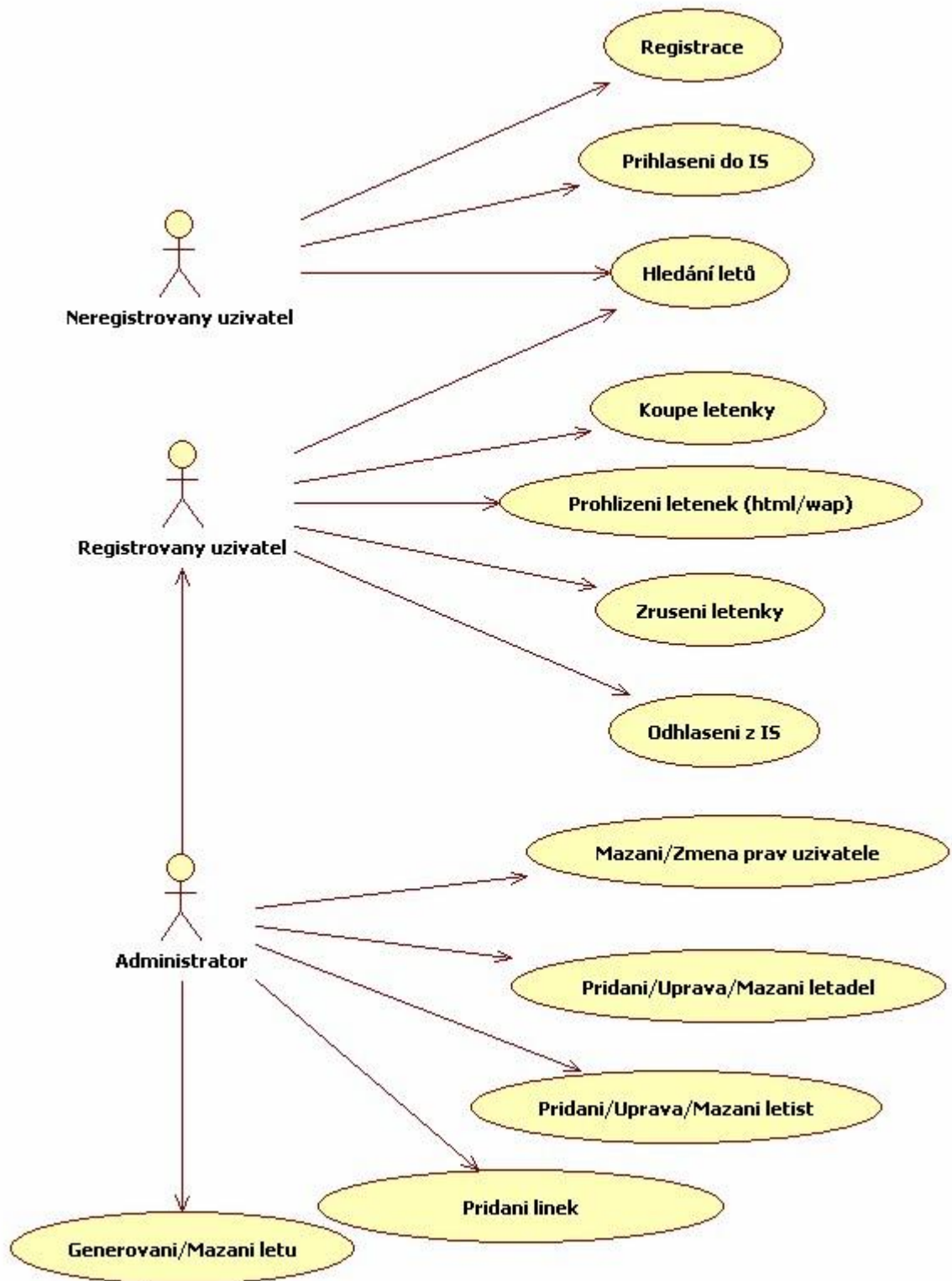
# Literatura

- [1] Gutmans, A., Bakken, S., Rethans, D. *Mistrovství v PHP*. CP Books, 1, 2005.
- [2] Demel, J. *Grafy a jejich aplikace*. Academia, 1, 2002.
- [3] PHP: Hypertext Preprocessor, <http://www.php.net>
- [4] MySQL AB, <http://www.mysql.com>
- [5] Wikipedia, the free encyclopedia, <http://en.wikipedia.org>

# Obrazové přílohy




Obrázek 1 - ER Diagram



Obrázek 2 - Use Case Diagram





[Úvod](#) | [Hledání](#) | [Košík](#) | [Administrace](#)
marten | [Odhlásit](#)

## Hledání:

---

Místo odletu:   
 Cíl cesty:   
 Datum cesty:     
 Zpáteční letenka:   
 Datum cesty zpět:     
 Radit podle:  Ceny  Délky letu

Obrázek 3 - Formulář pro vyhledávání



[Úvod](#) | [Hledání](#) | [Košík](#) | [Administrace](#)
marten | [Odhlásit](#)

## Hledání:

---

**Nalezené spoje (pro den 2006-04-21):**

Odkud	Kam	Odlet	Prilet	Cena
Brno (CZ)	Praha (CZ)	09:30	10:30	500
Praha (CZ)	Londyn (GB)	11:00	13:00	1000
<b>Celková cena</b>				<b>1500</b>
Množství: <input type="text" value="1"/>				<input type="button" value="Koupit"/>
Odkud	Kam	Odlet	Prilet	Cena
Brno (CZ)	Londyn (GB)	10:00	12:00	2000
<b>Celková cena</b>				<b>2000</b>
Množství: <input type="text" value="1"/>				<input type="button" value="Koupit"/>

Obrázek 4 - Výsledek hledání