

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

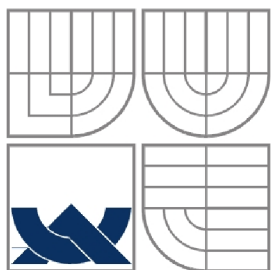
INFORMAČNÝ SYSTÉM NA REZERVÁCIU
CESTOVNÝCH LÍSTKOV

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

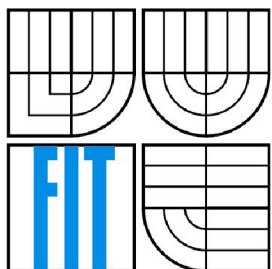
AUTOR PRÁCE
AUTHOR

MICHAL ELMAN

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÝ SYSTÉM NA REZERVÁCIU CESTOVNÝCH LÍSTKOV

INFORMATION SYSTEM FOR TICKETS RESERVATION

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MICHAL ELMAN

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ROMAN LUKÁŠ, Ph.D.

BRNO 2007

Zadanie bakalárskej práce

Řeřitel: **Elman Michal**

Obor: Informační technologie

Téma: **Informační systém pro rezervaci jízdenek**

Kategorie: Web

Pokyny:

1. Seznamte se s jazyky a prostředky pro tvorbu webových informačních systémů (XHTML, CSS, PHP, Javascript, MySQL).
2. Seznamte se s požadavky kladenými na IS pro rezervaci jízdenek. Rozsah systému konzultujte s vedoucím BP. Požadavky podrobně analyzujte.
3. Proveďte návrh systému. Při analýze požadavků a návrhu využijte vhodných modelovacích technik. Systém musí obsahovat i speciální funkčnost založenou na nějaké metodě z umělé inteligence. Výběr této metody konzultujte s vedoucím BP.
4. Daný systém implementujte.
5. Zhodnořte dosažené výsledky, porovnejte váš systém s existujícími systémy, navrhňte další možné rozšíření do budoucna.

Literatura:

- Kosek, J.: PHP - Tvorba interaktivních internetových aplikací

Vedoucí: Lukáš Roman, Ing., Ph.D., UIFS FIT VUT

Licenční zmluva

Licenční zmluva je uložena v archíve Fakulty informačních technologií Vysokého učení technického v Bmě.

Abstrakt

Cieľom bakalárskej práce je implementácia informačného systému na predaj a rezervácie cestovných lístkov pravidelnej autobusovej dopravy. Najskôr sú podrobne analyzované požiadavky, následne navrhnuté a implementované riešenie. Využívajú sa štandardné technológie na tvorbu webových aplikácií ako PHP, MySQL, HTML, CSS, JavaScript.

Kľúčové slová

webová aplikácia, rezervačný systém, cestovné lístky, PHP, MySQL

Abstract

The aim of this bachelor's thesis is implementation of information system for bus lines tickets sale and reservation. At the beginning the requirements are analyzed in detail, then the solution is designed and implemented. Standard technologies for the web application development, like PHP, MySQL, HTML, CSS, JavaScript, are used.

Keywords

web application, reservation system, tickets, PHP, MySQL

Citácia

Michal Elman: *Informačný systém na rezerváciu cestovných lístkov*, bakalárska práca, Brno, FIT VUT v Brně, 2007

Informačný systém na rezerváciu cestovných lístkov

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením Ing. Romana Lukáša, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Michal Elman
31.7.2007

© Michal Elman, 2007.

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.

Obsah

Úvod.....	3
1 Webové aplikácie.....	4
1.1 Databázová vrstva.....	4
1.1.1 MySQL.....	5
1.2 Aplikačná vrstva.....	5
1.2.1 PHP.....	5
1.3 Klientská vrstva.....	5
1.3.1 (X)HTML.....	6
1.3.2 CSS.....	6
1.3.3 JavaScript.....	6
2 Požiadavky na systém.....	7
3 Analýza a návrh.....	8
3.1 Konceptuálny model.....	8
3.1.1 Cestovné poriadky.....	9
3.1.2 Tarify.....	10
3.1.3 Autobusy a jazdy.....	10
3.1.4 Cestovné listky a rezervácie.....	11
3.1.5 Pomocné číselníky.....	11
3.1.6 Integritné obmedzenia.....	11
3.2 Prípady použitia.....	13
4 Implementácia.....	14
4.1 Databázový server.....	14
4.2 Aplikácia.....	14
4.2.1 Stránky a akcie.....	14
4.2.2 Autentifikácia a autorizácia.....	15
4.2.3 Prístup k databáze.....	15
4.3 Administračné rozhranie.....	16
4.3.1 Balíček tried metadát.....	16
4.3.1.1 Triedy stĺpcov tabuliek.....	17
4.3.1.2 Triedy kľučov.....	18
4.3.1.3 Triedy databázových trigrov.....	18

4.3.2	Balíček databázových akcií.....	19
4.3.3	Balíček formulárov.....	20
4.3.3.1	Triedy formulárových prvkov.....	20
4.3.3.2	Triedy formulárových tlačítiek.....	20
4.3.4	Balíček administračných tabuliek a formulárov.....	21
4.3.4.1	Triedy administračných tabuliek.....	21
4.3.4.2	Triedy administračných formulárov.....	21
4.4	Užívateľské rozhranie.....	22
4.5	Webové prehliadače.....	22
	Záver.....	23

Úvod

Za posledné desaťročia sa zmenil Internet zo siete statických stránok na sieť mnohých jednoduchších či zložitejších webových aplikácií. Elektronické nakupovanie sa stalo bežnou súčasťou života. Typickým predajným produktom sa stali letenky, mnohí si už nevedia ani predstaviť kúpiť letenku inak ako cez Internet. Zarážajúce však je, že predaj cestovných lístkov na autobusové linky je stále v plienkach. Najväčší „kamenný“ predpredajca cestovných lístkov začal ponúkať podobnú službu iba pred nedávnom, aj to iba na spoje vybraných dopravcov. Jeho veľkou nevýhodou je nemožnosť predaja cenovo výhodných cestovných lístkov na viac jazd. Rezervácie cez Internet umožňujú viacerí dopravcovia, avšak najväčším obmedzením týchto systémov je viazanie sa iba na jedného dopravcu, často krát umožňujú iba rezervácie vopred zakúpených lístkov a ponúkajú cestovné lístky iba v štandardných tarifách. Cieľom bakalárskej práce je práve takýto systém na predaj a rezerváciu cestovných lístkov, bez uvedených nedostatkov, navrhnuť a implementovať.

Práca je rozdelená do štyroch kapitol. Prvá sa venuje webovým aplikáciám, ich štruktúre a popisuje technológie využité pri implementácii. Druhá kapitola neformálne zhŕňa požiadavky na systém, ktoré sú v tretej časti podrobne analyzované a je navrhnuté riešenie. Posledná časť popisuje implementačné detaily riešenia.

1 Webové aplikácie

Webové aplikácie sú aplikácie, ku ktorým sa prístupuje cez webové rozhranie zo sietí ako Internet a intranet. Sú populárne vďaka svojej dostupnosti, nepotrebujú špeciálneho klienta, vystačia si s prehliadačom webových stránok, ktorý je súčasťou každej distribúcie operačného systému. Zvyčajne sú štruktúrované do troch vrstiev. Najbližšie k užívateľovi je prehliadač webových stránok, druhá vrstva je aplikačná, zabezpečuje logiku aplikácie, dáta sa ukladajú pomocou databáz.

1.1 Databázová vrstva

Databázové riadiace systémy usporiadávajú a štruktúrujú dáta tak, aby k nim užívatelia a aplikačné programy mohli získať prístup a ďalej s nimi pracovať. Štruktúru dát a techniky ich sprístupňovania, ktoré poskytujú databázové riadiace systémy, označujeme ako *dátové modely*. Postupne sa vyvinulo viacero takýchto modelov.

Pred zavedením databázových riadiacich systémov sa všetky dáta ukladali do súborov. Systém riadenia súborov bol poskytovaný ako súčasť operačných systémov. Nemal v podstate žiadny dátový model, ku všetkým súborom sa pristupovalo rovnako. Informácie o vlastnom obsahu súboru, aké dáta obsahuje a ako sú dáta usporiadané, boli začlenené v aplikačných programoch. Keď došlo k zmene dátovej štruktúry, bolo nutné modifikovať každý program, ktorý mal prístup k súboru. S postupným zväčšovaním počtu súborov a programov sa to stávalo obtiažnejšie.

Koncom šesťdesiatych rokov viedli problémy so správou súborovo založených systémov k vývoju databázových riadiacich systémov. Cieľom bolo vybrať z jednotlivých programov definíciu obsahu a štruktúry súborov a uložiť ju do databázy spolu s dátami. Vznikli tak hierarchický a sieťový dátový model.

V súčasnosti najpoužívanejším dátovým modelom je *relačný dátový model*, navrhnutý Dr. Coddom v sedemdesiatych rokoch. Dodatočne definoval 12 pravidiel, ktoré musí každá databáza spĺňať, aby mohla byť považovaná za skutočne relačnú. Základnou štruktúrou je *tabuľka*, ktorej riadky sa nazývajú *n-tice*, stĺpce *atribúty*. Každý atribút môže nadobúdať hodnoty z nejakej množiny prípustných hodnôt, nazývanej *doména*. Domény môžu obsahovať iba *atomické/skalárne* hodnoty. To znamená, že hodnota je ďalej nedeliteľná. Na identifikáciu jednotlivých záznamov slúžia *primárne kľuče*, na ktoré je možné sa odkazovať z iných záznamov pomocou *cudzích kľúčov*. Aby sa predišlo problémom s opakujúcimi sa informáciami (redundanciou), nemožnosťou reprezenovať určité informácie a složitým integritným obmedzeniam, mal by návrh tabuliek spĺňať určitú *normálnu*

formu. Normálne formy definujú, aké vlastnosti musí mať tabuľka s ohľadom na závislosti medzi jej atribútmi.

Najznámejšími databázovými systémami sú Oracle, IBM DB2, Microsoft SQL Server, Sybase, MySQL, PostgreSQL.

1.1.1 MySQL

MySQL je populárny relačný databázový riadiaci systém s otvoreným zdrojovým kódom, pod licenciou GNU GPL. Na prístup k databáze používa jazyk SQL, definovaný štandardmi ANSI/ISO. Pôvodne bol vyvinutý na rýchlu prácu s veľkými tabuľkami. Klientské knižnice sú dostupné pre mnohé programovacie jazyky. Podporuje transakčné spracovanie.

1.2 Aplikačná vrstva

Aplikačná vrstva je zvyčajne tvorená webovým serverom podporujúcim nejaký skriptovací jazyk, vďaka ktorému je možné dynamicky generovať webové stránky. V tejto vrstve je uložená logika samotnej aplikácie. Využíva dáta uložené databázovým riadiacim systémom a generuje stránky zobraziteľné vo webovom prehliadači. V súčasnosti sa najčastejšie používajú PHP, Microsoft ASP.NET, Java Server Pages, Python.

1.2.1 PHP

PHP je široko používaný skriptovací jazyk, špeciálne vhodný na tvorbu webových aplikácií. Syntax vychádza z jazykov C, Java a Perl. Hlavným cieľom jazyka je umožniť vývojárom rýchlo vyvíjať dynamicky generované stránky. Má vstavanú širokú podporu sieťových služieb a protokolov, umožňuje prístupovať k mnohým databázovým serverom, dokáže bežať pod rôznymi webovými servermi. Vo verzii 5 bol predstavený nový objektovo-orientovaný model.

1.3 Klientská vrstva

Klientom potrebným na prístup k aplikácii je prehliadač webových stránok. Štandardne býva v každej distribúcii operačného systému. Medzi najpoužívanjšie patrí Microsoft Internet Explorer, Mozilla Firefox, Safari, Opera. Špeciálnymi typmi sú textové prehliadače ako Lynx a Links.

1.3.1 (X)HTML

HTML je značkovací jazyk určený na vytváranie hypertextových dokumentov. Poskytuje prostriedky na štrukturovanie textu, ako napr. nadpisy, odstavce, zoznamy, tabuľky, umožňuje vytváranie formulárov a vkladanie obrázkov. HTML zapísaný vo formáte XML sa nazýva XHTML. Je štandardizovaný konzorcium W3C.

1.3.2 CSS

Zpočiatku sa definoval štýl zobrazenia priamo v HTML dokumente. Neskôr bolo snaha oddeliť definíciu obsahu od spôsobu zobrazenia. To viedlo ku vzniku kaskádových štýlov CSS. Umožňuje jednotlivým elementom definovať farbu, font, rozloženie a ďalšie aspekty. Jeden dokument môže mať definovaných viacero štýlov pre rôzne médiá, napr. iný pre obrazovku ako pre tlač. Je taktiež štandardizovaný konzorcium W3C.

1.3.3 JavaScript

JavaScript je interpretovaný skriptovací jazyk so syntaxou podobnou jazykom C/C++. Je objektovo orientovaný, založený na prototypoch a so slabou typovou kontrolou. Implementovaný vo väčšine súčasných webových prehliadačov. Webovým stránkam pridáva interaktívnosť, možnosť okamžite reagovať na akcie užívateľa. Pochádza od firmy Netscape, Microsoft ho implementoval ako JScript, štandardizovaný je organizáciou ECMA ako ECMAScript.

2 Požiadavky na systém

Informačný systém má zabezpečovať predaj a rezervácie cestovných lístkov na pravidelnú autobusovú dopravu viacerých dopravcov. Každý dopravca bude mať prístup k systému a bude sám zadávať všetky potrebné údaje a riadiť samotný proces predaja a rezervácií.

Administrátor bude zodpovedať za obsah zoznamu dopravcov, zastávok, mien, štátov a sviatkov.

Cestovné poriadky liniek budú vychádzať z formátu JDF, v ktorom sú dopravcovia povinní zasielať dáta do Celoštátneho informačného systému o cestovných poriadkoch verejnej linkovej dopravy (CIS JŘ). Presný formát je uvedený v [1].

Ku každej linke bude môcť predajca ponúkať viacero taríf. Okrem taríf na jednu jazdu by mal systém umožňovať viesť aj špeciálne tarify, ktoré autobusoví dopravcovia bežne používajú. Jedná sa o spätočné tarify, týždenné ale aj rôzne last minute a first moment ponuky. Cestovné bude určené ako cena z nástupnej do výstupnej zastávky. S kilometrickou tarifou netreba uvažovať, nakoľko sa na diaľkových a medzinárodných linkách, pre ktoré je systém určený, takmer vôbec nevyužíva.

Pri rezervácií si bude môcť cestujúci vybrať konkrétne sedadlo v autobuse. Z toho vyplýva pre dopravcov nutnosť evidovania typov autobusov spolu s plánom rozmiestnenia jednotlivých sedadiel a ich očíslovaním.

Priradovanie autobusov na jazdy bude prebiehať zväčša automaticky podľa vopred určených pravidiel. Zároveň však bude mať dopravca plnú kontrolu nad jazdami s možnosťou pridania alebo zrušenia autobusu. Na jeden spoj bude možné nasadiť viacero autobusov, ktoré budú rozlíšené číslami.

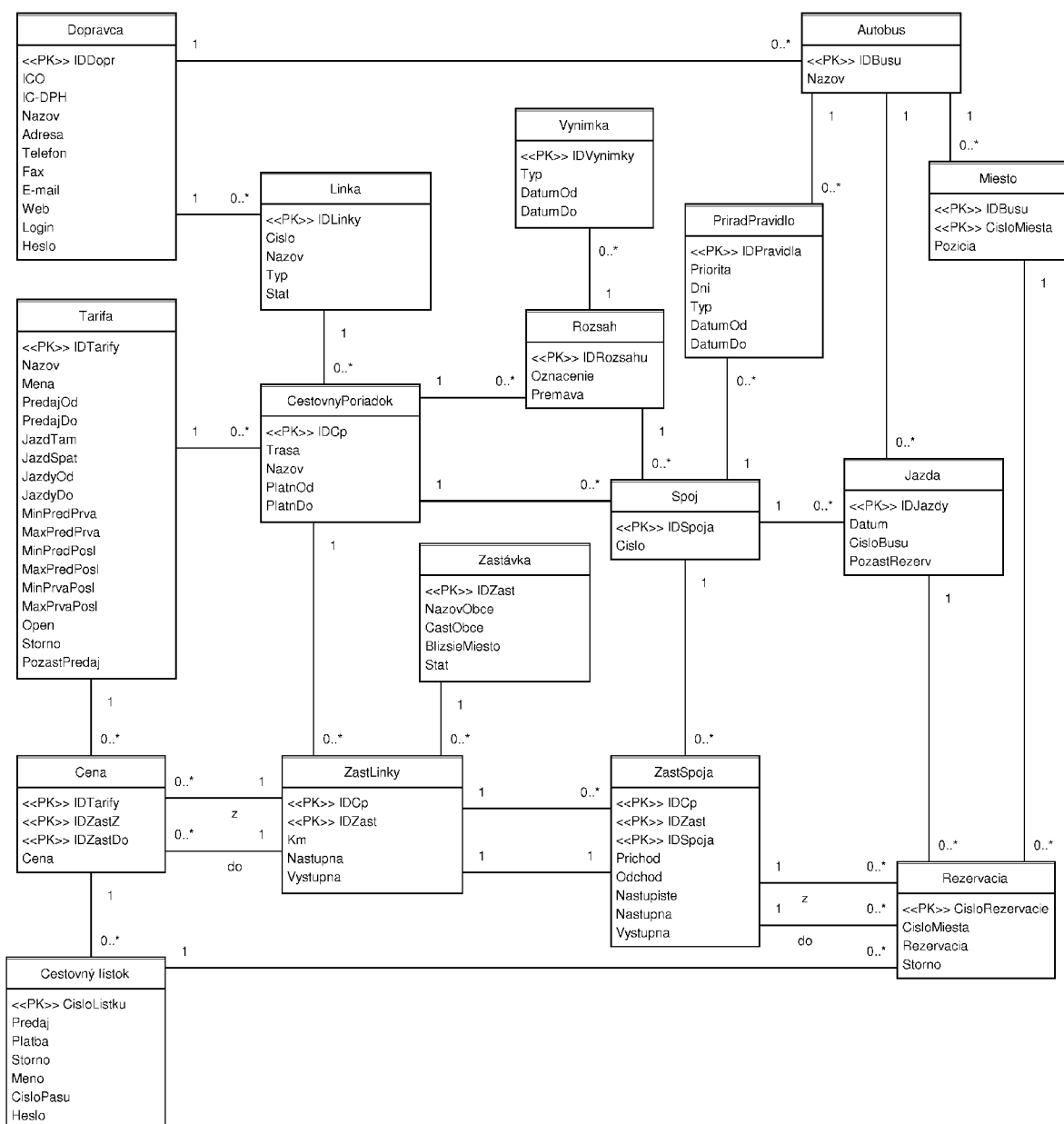
Cestovné lístky budú viazané na tarifu a nástupnú a výstupnú zastávku danej linky. Vzhľadom na existenciu taríf na viac jazd, bude môcť byť k jednému lístku viacero rezervácií. Rezervácia sa bude vzťahovať ku konkrétnej jazde, čiže určitému autobusu v určitý dátum na spoj danej linky a miestu v danom autobuse.

Informačný systém bude implementovaný ako webová aplikácia, prístupná cez Internet z ľubovoľného webového prehliadača. Platba za cestovné lístky bude taktiež prebiehať cez Internet.

3 Analýza a návrh

3.1 Konceptuálny model

Konceptuálny model informačného systému môžeme rozdeliť do piatich základných častí. Prvou časťou sú dáta týkajúce sa cestovných poriadkov. Druhá časť tvoria tarify, tretiu autobusy a jazdy. Ďalšia časť reprezentuje cestovné lístky a jednotlivé rezervácie. V poslednej časti ostávajú pomocné číselníky. Výsledný dátový model je vo forme ER diagramu na obrázku 3.1.



Obrázok 3.1: ER Diagram

3.1.1 Cestovné poriadky

Ako už bolo spomenuté v časti týkajúcej sa požiadaviek na systém, model cestovných poriadkov vychádza z formátu JDF [1].

Zastávka obsahuje atribúty *NazovObce*, *CastObce*, *BlizsieMiesto*. Celý názov zastávky vznikne spojením názvu, časti obce a bližšieho miesta, oddelených čiarkami. Je viazaná na štát.

Entita Dopravca reprezentuje dopravcu, obsahuje atribúty bežné pre organizácie a firmy ako *ICO*, *IC-DPH*, *Nazov*, *Adresu*, *Telefon*, *Fax*, *E-mail*, *Web*. K tomu je ešte potrebné pridať atribúty *Login* a *Heslo*. *Login* predstavuje prihlasovacie meno do systému, ktoré musí byť jedinečné, nečíselné, pretože čísla lístkov sa používajú ako prihlasovacie mená a rôzne od prihlasovacieho mena administrátora.

Pôvodná entita Linka, vychádzajúca z JDF, bola rozdelená na dve. Prvou je Linka, druhou CestovnyPoriadok. Účelom je zoskupiť cestovné poriadky jednej a tej istej linky. Je to nutné aj kvôli uznávaniu lístkov na viac jazd podľa taríf zo starších cestovných poriadkov na jazdy podľa novších. Linka obsahuje atribúty *Cislo*, *Nazov* a *Typ*. Odkazuje sa na dopravcu, ktorý danú linku prevádzkuje. Číslo musí byť šesťmiestne a v rámci liniek jedného dopravcu unikátne. Typ linky je buď vnútroštátna alebo medzinárodná. Pri vnútroštátnych linkách je potrebné naviazať linku na štát.

CestovnyPoriadok má atribúty *Trasa*, *Nazov*, *PlatnostOd*, *PlatnostDo*. Cestovný poriadok je viazaný na konkrétnu linku. Na rozdiel od JDF sa viaceré trasy na jednej linke nerozlišujú na úrovni zastávok spoja podľa kilometráže, ale už cestovný poriadok má presne stanovenú trasu. Jednotlivé trasy sa rozlišujú podľa atribútu *Trasa*. *PlatnostOd* a *PlatnostDo* údajujú začiatok a koniec platnosti cestového poriadku. Platnosti cestovných poriadkov na rovnako označenej trase linky sa nesmú prekryvať.

Ku každému cestovnému poriadku sú naviazané ZastLinky, ktoré presne určujú po akej trase linka premáva. Odkazuje sa na zastávku. Atribútmi sú *Km*, *Nastupna* a *Vystupna*. Pomocou kilometrov *Km* sa udáva poradie na trase. Prvá zastávka musí mať hodnotu nula. Pri vnútroštátnych linkách musí zastávka ležať na území štátu linky.

Entita Spoj sa viaže na cestovný poriadok. Jediným atribútom je povinný atribút *Cislo*. Pomocou ktorého sa vyjadruje aj smer premávky spoja, nepárne čísla znamenajú smer tam, čiže stúpajúcu postupnosť kilometrov zastáviok linky, párne čísla značia opačný smer.

Obsluhu zastávky linky spojom vyjadruje ZastSpoja. Odkazuje sa na spoj a zastávku linky. Pri každej sa určuje *Prichod*, *Odchod*, *Nastupiste* a či je zastávka *Nastupna* a *Vystupna*. Ak bola definovaná na úrovni linky ako nie nástupna, musí tak byť definovaná aj na úrovni spoja. To isté platí

o výstupnej. Ku každej nástupnej zastávke musí byť uvedený čas odchodu. Časy príchodov a odchodov musia v rámci spoja tvoriť stúpajúcu postupnosť podľa poradia zastávok na trase.

Entita *Rozsah* vyjadruje časový rozsah premávky. Každý spoj sa musí odkazovať na jeden rozsah. Atribútmi sú *Oznacenie* a *Premava*, ktorý určuje dni v ktoré spoj premáva. Možnými hodnotami sú: pracovné dni, štátne sviatky, pondelok, utorok, streda, štvrtok, piatok, sobota, nedeľa.

Výnimku z rozsahu je možné vyjadriť entitou *Vynimka*. Pri ktorej sa určuje *Typ*, *DatumOd* a *DatumDo*. Typy výnimiek sú rovnaké ako typy časových kódov podľa JDF [1]. Taktiež je potrebné dodržať všetky obmedzenia ktoré sa k jednotlivým typom vzťahujú.

3.1.2 Tarify

Ku každému cestovnému poriadku má dopravca možnosť viesť viacero taríf. *Tarifa* sa odkazuje na menu, v ktorej je vedená. Medzi atribúty patrí *Názov*, *PredajOd*, *PredajDo*, ktorý ak nie je uvedený, tak je predaj možný do konca platnosti cestovného poriadku. Počet jazd je obmedzený atribútmi *JazdTam* a *JazdSpat*. Časové obmedzenie tarify je možné určením presného rozsahu, kedy je možné vykonať jazdy pomocou *JazdyOd* a *JazdyDo*, alebo určením minimálnej a maximálnej doby od predaja po prvú jazdu, od predaja po poslednú jazdu alebo od prvej po poslednú jazdu atribútmi *MinPredPrva*, *MaxPredPrva*, *MinPredPosl*, *MaxPredPosl*, *MinPrvaPosl*, *MaxPrvaPosl*. Vyplňajú sa len tie obmedzenia, ktoré sa majú použiť. Atribút *Open* určuje, či je možné rezervácie uskutočniť aj neskôr ako pri predaji. *Storno* udáva možnosť stornovať rezerváciu v hodinách pred odchodom z nástupnej zastávky. Pomocou *PozastPredaj* je možné pozastaviť predaj cestovných lístkov danej tarify.

Cestovné určuje entita *Cena*. Je viazaná na tarifu a nástupnú a výstupnú zastávku linky podľa toho istého cestovného poriadku. *Cena* je udávaná v mene podľa tarify.

3.1.3 Autobusy a jazdy

Typ autobusu reprezentuje entita *Autobus*, ktorej jediným atribútom je *Názov*. Na každý autobus je naviazaných viacero miest entity *Miesto*, pri ktorých sa určuje *Cislo* a *Pozicia* na pláne rozmiestnenia sedadiel autobusu.

Jazda autobusu sa viaže na spoj a typ autobusu. Atribútmi sú *Datum*, *CisloBusu*, *PozastRezerv*. *CisloBusu* slúži na rozlíšenie viacerých autobusov na jednom spoji v jeden deň. Pomocou *PozastRezerv* je možné pozastaviť rezervácie cestovných lístkov na danú jazdu autobusu.

Automatické priradovanie autobusov na jazdy prebieha podľa pravidiel *PriradPravidlo*. Ak spoj v daný deň premáva vyberie sa prvé pravidlo zoradených podľa atribútu *Priorita*. *Dni* určuje,

v ktoré dni pravidlo platí. Možnými hodnotami sú: pracovné dni, štatne sviatky, pondelok, utorok, streda, štvrtok, piatok, sobota, nedeľa. Ďalšie obmedzenie platnosti pravidla je možné dosiahnuť pomocou *Typ* a *DatumOd*, *DatumDo*. *Typ* môže byť: „od ... do ...“, „párne týždne od ... do ...“, „nepárne týždne od ... do ...“.

3.1.4 Cestovné lístky a rezervácie

Cestovný lístok *Listok* sa viaže na cenu, ktorá jednoznačne určuje tarifu a nástupnú a výstupnú zastávku. *CisloListku* značí poradové číslo lístku, *Predaj* dátum a čas predaja, *Platba* dátum a čas zaplataenia a *Storno* dátum a čas stornovania lístku, ak bol stornovaný. K cestovnému lístku sa udáva ešte meno cestujúceho atribútom *Meno*, číslo cestovného dokladu *CisloPasu* a heslo *Heslo*, pomocou ktorého je možné prihlásenie do systému.

Rezervacia je naviazaná na lístok, jazdu a nástupnú a výstupnú zastávku spoja. Atribútmi sú poradové číslo rezervácie *CisloRezervacie*, číslo miesta v autobuse *CisloMiesta*, dátum a čas rezervovania *Rezervacia* a dátum a čas stornovania *Storno*, ak bola rezervácia stornovaná.

3.1.5 Pomocné číselníky

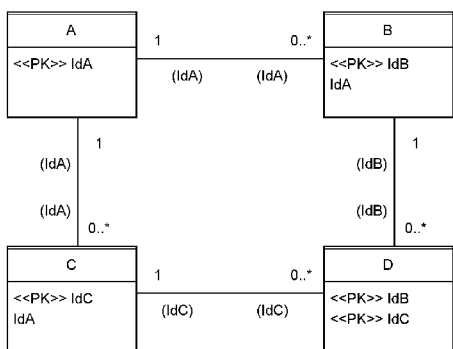
Vyššie zmienené entity sa viažu na štáty a meny. Tie sú reprezentované entitou *Stat*, s atribútmi *Skratka* a *Nazov* a entitou *Mena* s rovnakými atribútmi. Pre správnu funkčnosť automatického priradovania autobusov na jazdy je nutné v systéme evidovať štátne sviatky pre každý štát. Služi na to entita *Sviatok*, naviazaná na štát s atribútmi *Datum* a *Nazov*.

3.1.6 Integritné obmedzenia

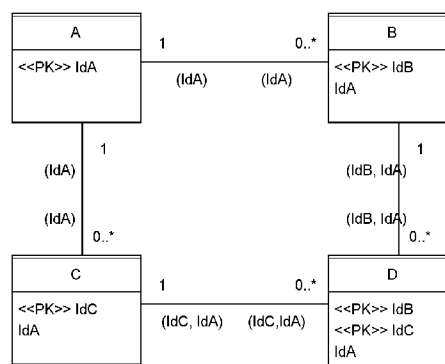
V konceptuálnom modeli boli uvedené mnohé integritné obmedzenia, ktoré je nutné počas práce s databázou, kvôli zachovaniu konzistencie dát, kontrolovať. Prvým a najpoužívanejším obmedzením je obmedzenie dátovým typom. Umožňuje priradiť stĺpcu iba hodnotu takého typu aká bola určená pri definícii schémy databázy. Pri povinných atribútoch sa využíva vlastnosť stĺpca NOT NULL, ktorá zakazuje vložiť prázdnu hodnotu NULL. Identifikátory ID sú definované ako primárne kľúče PRIMARY KEY, čo zabezpečuje ich jedinečnosť a nutnosť vyplnenia. Vďaka vlastnosti AUTO INCREMENT sa vytvára vždy nové ID pri vložení nového riadku. Nad ostatnými stĺpcami, ktoré musia byť jedinečné je možné definovať unikátny kľúč UNIQUE. Nakoľko databázový server MySQL nespracováva klauzulu CHECK v definíciách tabuliek, ostatné intrarelačné obmedzenia je nutné zabezpečiť pomocou databázových triggerov.

Ďalším dôležitým integritným obmedzením je obmedzenie cudzím kľúčom. Je to interrelačné obmedzenie medzi dvomi tabuľkami, v odkazujúcej sa tabuľke je uložený cudzí kľúč s hodnotou rovnakou ako primárny kľúč odkazovanej tabuľky. Cudzí kľúč tak musí vždy obsahovať platný primárny kľúč z odkazovanej tabuľky, prip. hodnotu NULL, ak nebol definovaný ako NOT NULL. Pri vymazaní riadku v odkazovanej tabuľke sa buď vymažú aj odkazujúce sa riadky, alebo sa vymazanie nepovolí. Podobne to funguje aj pri aktualizácii.

Zložitejšia je kontrola ak ide o vzťah medzi viacerými tabuľkami. Uvediem príklad, majme tabuľky A, B, C, D. Tabuľky B a C sa odkazujú na tabuľku A. Tabuľka D sa odkazuje na B a C. Je potrebné zabezpečiť aby sa riadok z tabuľky D mohol odkazovať iba na riadky z tabuľky B a C také, že oba sa odkazujú na ten istý riadok z tabuľky A. Jednou z možností je vynútiť si toto obmedzenie pomocou trigrov. Omnoho jednoduchšie je však odkazovať sa z tabuľky D do tabuliek B a C nie na primárne kľúče (obrázok 3.2), ale na skupinu stĺpcov zloženú z ich primárnych kľúčov a cudzieho kľúča do tabuľky A (obrázok 3.3). V ďalšom texte sa táto skupina stĺpcov označuje ako referenčný kľúč. Poruší sa tým síce pravidlo odkazovania sa na primárne kľúče a takto navrhnutá schéma nebude spĺňať tretiu normálnu formu, avšak získajú sa tým viaceré výhody. Prvou je vyhnutie sa spúšťaniu špeciálnych trigrov pri práci s tabuľkami. Druhou je získanie priamej referencie z tabuľky D na tabuľku A. Čo síce spôsobí nie nevyhnutnú redundanciu, zato ale umožní písanie jednoduchších skriptov a trigrov zabezpečujúcich ostatné integritné obmedzenia, nakoľko nie je potrebné v nich používať viacnásobne vnorené výberové dotazy ani viacnásobne spojenia.



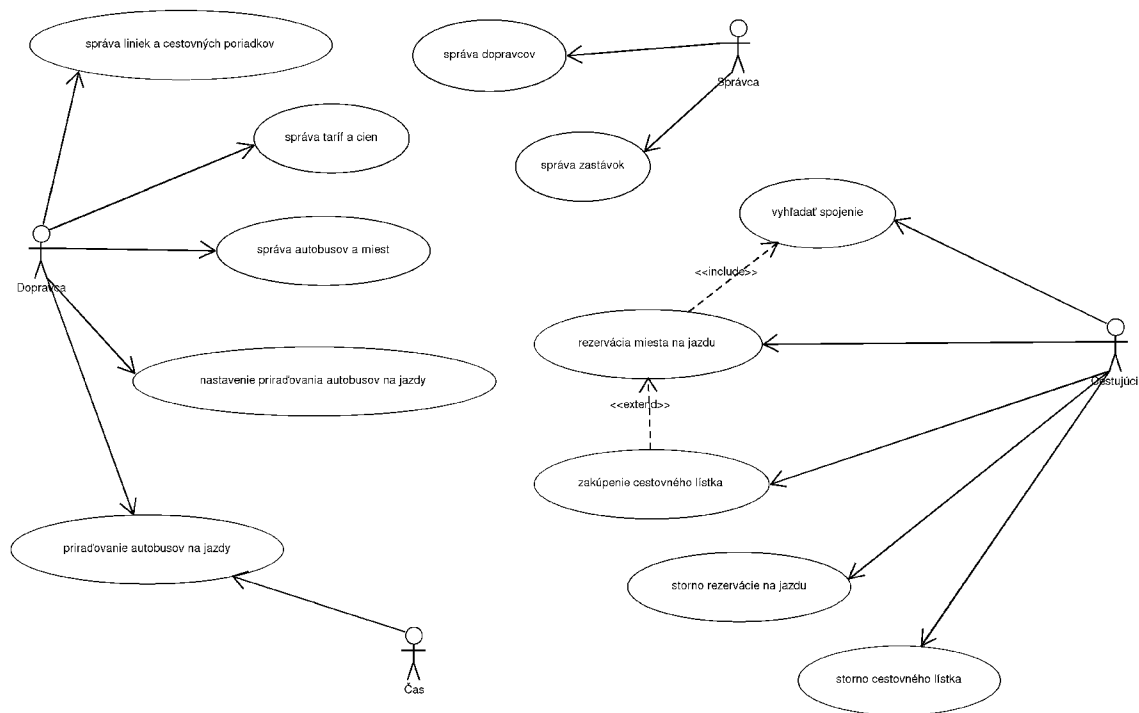
Obrázok 3.2: Odkazovanie na primárne kľúče



Obrázok 3.3: Odkazovanie na referenčné kľúče

3.2 Prípady použitia

Diagram prípadov použitia zachytáva požiadavky na funkčnosť systému. Vymedzuje jednotlivých aktérov a možnosti ich interakcie so systémom. Zjednodušený model je uvedený na obrázku 3.3.



Obrázok 3.4: Use Case Diagram

4 Implementácia

4.1 Databázový server

Ako databázový server slúži *MySQL* vo verzii *5.0*. Využíva sa typ úložiska *InnoDB*, ktorý plne podporuje transakcie, čiže atómicnosť, konzistenciu, nezávislosť a trvalosť transakcií. Taktiež umožňuje využívať obmedzenie cudzím kľúčom, čo v *MyISAM* nie je možné. Databázový server beží v *striktnom režime*, ktorý okrem iného nepovolí vkladanie nulových častí alebo nulových celých dátumov, dôsledne sa kontroluje platnosť dátumu, nielen intervaly možných hodnôt jednotlivých zložiek. Vloženie chybnéj hodnoty do výčtového typu alebo množiny, ako aj vloženie dlhšieho reťazca, spôsobí chybu namiesto varovania. Každý príkaz sa spúšťa ako transakcia, čiže buď sa vykoná všetko alebo nič. Striktný režim približuje aj syntax viac k štandardu SQL, napríklad na ohraničenie reťazcových literálov slúžia apostrofy a nie úvodzovky.

4.2 Aplikácia

Informačný systém je implementovaný ako webová aplikácia v skriptovacom jazyku *PHP 5.2*. Generované stránky sú vo formáte *XHTML*, podľa štandardu *W3C*. Formát a štýl zobrazenia sa definuje pomocou kaskádových štýlov *CSS*. Dynamickosť stránok na strane klienta je dorobená pomocou skriptovacieho jazyka *JavaScript*.

Rozhranie aplikácie môžeme rozdeliť na užívateľské, prístupne neprihláseným užívateľom a cestujúcim a administráčne prístupne administrátorovi a dopravcom. Na vytváranie týchto rozhraní sa využívajú objektové vlastnosti jazyka *PHP 5*, pričom administráčne rozhranie je výlučne objektové.

4.2.1 Stránky a akcie

Všetky požiadavky obsluhuje jeden skript *index.php*. Podľa parametra *page* zadaného metódou *GET* načíta definíciu požadovanej stránky zo súboru. Každá stránka je odvodená od abstraktnej triedy *Page*, ktorej základnými vlastnosťami sú meno *name*, zoznamom vyžadovaných parametrov *params* a vyžadovaná užívateľská rola *role*. Ďalej obsahuje metódy na nastavenie týchto vlastností a metódu *display()*, ktorá zabezpečuje jej zobrazenie. Pred samotným vykreslením stránky sa skontroluje existencia vyžadovaných parametrov, ktoré sa predávajú metódou *GET* a

autorizovanosť užívateľa. Následné sa zobrazí hlavička, menu a samotná stránka. Menu vykresluje statická metóda *display()* triedy *Menu*, ktorá slúži ako kontajner funkcií pre menu.

O spracovanie formulárov, manipuláciu s databázou, prihlásenie, odhlásenie a podobnú funkčnosť sa starajú objekty odvodené od abstraktnej triedy *Action*. Zoznam podporovaných akcií *actions* je vlastnosťou každej stránky. Pred spustením akcie sa kontroluje, rovnako ako pri zobrazení stránky, zadanie parametrov *params* akcie a autorizovanosť užívateľa na jej prevedenie. Samotné vykonanie sa spustí volaním metódy *act()*, po ktorej vykonaní sa prehliadač automaticky presmeruje na inú stránku zaslaním hlavičky *Location* protokolu HTTP. Takto sa stránka s volaním akcie nedostane do histórie navštívených stránok, čo umožní užívateľovi bezpečné používanie tlačidiel „Back“ a „Forward“ bez toho, aby došlo k opakovanému volaniu akcie. Premesmerovanie môže byť buď pevné alebo rôzne v závislosti od úspešnosti vykonania akcie. Využíva sa to v administratívnych formulároch, kde pri úspechu dôjde k presmerovaniu na administratívnu tabuľku z ktorej bol formulár otvorený, v prípade neúspechu sa zobrazí opäť formulár s chybovou hláškou. Túto hlášku ukladá akcia do premennej relácie statickou metódou *setAlert()* triedy *Alert*. Pri každom zobrazení stránky sa táto premenná relácie kontroluje, príp. sa zobrazí chybová hláška, volaním statickej metódy *showAlert()* triedy *Alert*, slúžiacej ako kontajner na funkcie chybových hlášok.

4.2.2 Autentifikácia a autorizácia

V systéme sa rozlišujú užívateľské roly administrátor, dopravca, cestujúci a neprihlásený. Prihlasovanie užívateľov prebieha pomocou užívateľského mena a hesla. Cestujúci majú ako prihlasovacie meno číslo cestovného lístku, dopravcovia ľubovoľný nečíselný reťazec a prihlasovacie meno administrátora je uvedené v konfiguračnom súbore. Heslá sú uložené ako SHA1 hash hodnoty v príslušných tabuľkách, resp. u administrátora v konfiguračnom súbore. Na autentifikačné a autorizačné účely slúžia statické metódy triedy *Auth*. Požadovaná užívateľská rola je uvedená u každej stránky. Taktiež sa kontroluje, či sa požadované dáta vzťahujú k danému užívateľovi. Správu užívateľských relácií sa zabezpečuje pomocou cookies. Po dlhšej nečinnosti dochádza k automatickému odhláseniu.

4.2.3 Prístup k databáze

Na prístup k databáze sa využíva trieda *DBConnection*, vytvorená podľa návrhového vzoru *Singleton*, ktorý zabezpečuje jedinečnosť inštancie tejto triedy. Zapúzdruje inštanačnú premennú na prístup k databáze a metódy na zistenie výskytu chyby a chybového hlásenia z posledného príkazu. K databáze sa pristupuje pomocou objektu *PDO* (PHP Data Objects). Je to rozšírenie, ktoré umožňuje

zadávať príkazy a získavať dáta z rôznych databázových systémov jednotným spôsobom. Spolu s používaním konštrukcií podľa štandardného SQL to zabezpečuje jednoduchšiu prenositeľnosť aplikácie. Pri vkladaní textových reťazcov do dotazov sa využíva metóda *PDO quote()*, ktorá vloží reťazec predaný ako parameter do úvodzoviek a nahradí špeciálne znaky únikovými sekvenciami náležite podľa používaného databázového servera. Predchádza sa tým chybám typu SQL injection. Metóda na zistenie chybového hlásenia vychádza z chybového kódu vráteného serverom, prípadne zisťuje ďalšie podrobnosti o chybe, aby užívateľ dostal čo najpresnejšiu chybovú hlášku.

4.3 Administračné rozhranie

Administračné rozhranie slúži administrátorovi na správu dopravcov, zastávok a napĺňanie pomocných číselníkov ako sú meny, štáty a sviatky. Pre dopravcu slúži na správu cestovných poriadkov a taríf, priradovanie autobusov na jazdy a prehľady o predaných lístkoch a rezerváciách na jednotlivé jazdy.

Nakoľko sa vždy jedná o bežnú prácu s tabuľkami, čiže prehliadanie, pridávanie, aktualizovanie a odstraňovanie riadkov, bolo administračné rozhranie implementované pomocou dynamického generátora prehľadových tabuliek a editačných formulárov. Vzhľadom aj na počet tabuliek (20) je to jediný spôsob ako dosiahnuť udržiavateľnosť a ľahkú rozšíriteľnosť systému. Stránky s administračnými tabuľkami a formulármi sa generujú za behu.

K činnosti tohto generátora je potrebné mať k dispozícii okrem štruktúry databázy taktiež doplňujúce údaje ako hlavičky stĺpcov tabuliek, menovky a podrobný popis prvkov formulárov. Tieto údaje sú uložené spolu s metadátami v objektoch tried z balíčku metadát.

4.3.1 Balíček tried metadát

Na reprezentáciu štruktúry databázy slúžia triedy s prefixom DB. Všetky triedy sú odvodené z abstraktnej triedy *DBObject*, ktorá je tvorená atribútom mena *name* a abstraktnou metódou *createDef()*, ktorá vracia definíciu objektu v jazyku SQL.

Trieda *DBSchema* je vytvorená podľa návrhového vzoru *Singleton*. Obsahuje zoznamy referencií na tabuľky, uložené procedúry a funkcie. Zavolaním metódy *getTable(name)* sa načíta definícia tabuľky *name* zo súbora a vráti sa objekt triedy *DBTable*. Celá štruktúra databázy sa načítava iba pri volaní metódy *createDef()*, ktorá vráti definíciu celej schémy databázy, v ostatných prípadoch sa načítajú iba definície tabuliek, s ktorými sa pracuje.

DBTable reprezentuje databázovú tabuľku. Zapúzdruje metódy na nastavenie a získanie referencií a zoznamy referencií na stĺpce, primárny kľúč, referenčný kľúč, unikátne kľúče, indexy, cudzie kľúče a trigre. Taktiež prepisuje metódu *createDef()*, vracajúcu príkaz na vytvorenie tabuľky v jazyku SQL.

4.3.1.1 Triedy stĺpcov tabuliek

Abstraktná trieda *DBCColumn* je rodičovskou triedou tried stĺpcov. Obsahuje vlastnosti ako menovka a podrobnejší popis poľa formulára, záhlavie stĺpca administratívnej tabuľky, príp. záhlavie ak je vkladán ako stĺpec z cudzej tabuľky, zakázanie prázdnej hodnoty NULL, implicitná hodnota, operátor filtra v administratívnej tabuľke. Medzi metódy patrí prevedenie na stĺpec administratívnej tabuľky, abstraktná metóda na nastavenie jeho vlastností, implementovaná v synovských triedach, abstraktná metóda na prevedenie na formulárové pole, implementovaná v synovských triedach, a metóda na nastavenie jeho vlastností. Špeciálnymi metódami sú metódy na formátovanie hodnôt získaných z databázy. Formátovanie sa môže vykonávať buď na strane SQL servera, kde sa vloží do príkazu SELECT namiesto názvu stĺpca formátovacia funkcia, využíva sa to pri dátumoch, alebo sa hodnota formátuje priamo v PHP skripte, využívané pri číslach. Obdobným spôsobom je nutné preformátovať hodnoty aj pri vkladaní z formulárov do databázy. Triedy odvodené od *DBCColumn* :

- *DBInt*, *DBSmallInt*, *DBTinyInt* – celočíselné stĺpce, s rôznym rozsahom hodnôt, pričom tento rozsah je možné ešte viac obmedziť, môžu byť so znamienkom alebo bez znamienka
- *DBId*, *DBForeignId* – stĺpec s identifikátorom, odvodené od triedy *DBInt*, bez znamienka, *DBId* neumožňuje hodnotu NULL a pri vložení nového riadku sa inkrementuje
- *DBBit* – slúži na uchovanie logických hodnôt – TRUE a FALSE – v MySQL implementácia typom BOOL, neumožňuje hodnotu NULL
- *DBChar*, *DBVarChar* – textové reťazce s pevnou alebo premenlivou, avšak určenou maximálnou dĺžkou, hodnota prázdneho reťazca bola stotožnená s hodnotou NULL, nakoľko má rovnakú vypovedaciu hodnotu
- *DBText* – textový reťazec s maximálnou dĺžkou 2^{16} , prázdny reťazec taktiež zodpovedá hodnote NULL
- *DBPassword* – 40-miestny *DBChar*, slúži na uloženie hesiel ako SHA1 hash, je možné stanoviť minimálnu a maximálnu dĺžku hesla
- *DBDate*, *DBTime*, *DBDateTime* – typy na uloženie, dátumu, času a dátumu s časom, čas sa ukladá iba ako dvojica hodiny a minúty, pričom hodiny môžu prekročiť hodnotu 24

- *DBDecimal* – desatinné čísla s presne určeným počtom celých a desatinných miest, môže byť so znamienkom SIGNED alebo bez UNSIGNED
- *DBEnum* – výčtový typ, umožňuje nadefinovať hodnoty z ktorých stĺpec môže obsahovať práve jednu, príp. žiadnu ak nie je definovaný ako NOT NULL, ku každej hodnote je priradený aj textový popis
- *DBSet* – typ množina, stĺpec môže obsahovať ľubovoľnú podmnožinu z vopred definovaných hodnôt, prázdna množina je zhodná s hodnotou NULL, čiže ak je stĺpec definovaný ako NOT NULL, musí obsahovať aspoň jednu hodnotu, ku každej hodnote je priradený textový popis a skratka, ktorá sa používa pri zobrazení v administračnej tabuľke

4.3.1.2 Triedy kľučov

- *DBPrimaryKey* – primárny kľúč, slúži na jednoznačnú identifikáciu každého riadku v tabuľke, ku každej tabuľke existuje práve jeden, daný zoznamom stĺpcov, ktorými je tvorený
- *DBReferKey* – referenčný kľúč, ako bolo spomenuté v XXX, je to kandidátny kľúč na ktorý sa odkazujú cudzie kľúče v ostatných tabuľkách, ku každej tabuľke je práve jeden, daný zoznamom stĺpcov, ktorými je tvorený, implementovaný ako *DBIndex*, pretože nie je potrebná kontrola unikátnosti, nakoľko sa jedná o nadmnožinu primárneho kľúča
- *DBUniqueKey* – unikátny kľúč, zabezpečuje jedinečnosť danej kombinácie hodnôt stĺpcov v tabuľke ako u primárneho kľúča, daný zoznamom stĺpcov, ktorými je tvorený
- *DBIndex* – index, slúži na urýchlenie vyhľadania dát v tabuľke, daný zoznamom stĺpcov, ktorými je tvorený
- *DBForeignKey* – cudzí kľúč slúži na odkazovanie sa na riadky druhej tabuľky, je daný druhou tabuľkou, jej aliasom využívanom pri spájaní tabuliek a výbere stĺpcov z druhých tabuliek v administračných tabuľkách, zoznamom odkazujúcich sa a odkazovaných stĺpcov a akciou vykonanou pri zmazaní a aktualizovaní riadku v odkazovanej tabuľke, možnými akciami sú previesť príslušnú udalosť aj na odkazujúcej sa tabuľke alebo zakázať vykonanie takejto udalosti

4.3.1.3 Triedy databázových trigrov

Databázove trigre sú reprezentované triedou *DBTrigger*. Obsahuje základné vlastnosti trigrov a to tabuľku, ku ktorej trigger patrí, čas a udalosť spustenia a samotné telo. Čas môže byť pred alebo po danej udalosti, udalosťou môže byť vloženie, aktualizácia alebo vymazanie riadku.

- *DBTriggerSet* – slúži na výpočet odvodených stĺpcov v tabuľke, keďže MySQL nemá na to prostriedky, nastaví sa stĺpec a výraz, podľa ktorého sa má vypočítať, trigger sa spúšťa pri vkladani a aktualizácii riadkov.

Nasledovné trigre slúžia na kontrolu integritných obmedzení. V prípade, že dôjde k porušeniu konzistencie, zavolá sa procedúra, ktorá ukončí príkaz s chybovou hláškou. Nakoľko MySQL takúto vstavanú funkciu nemá, je možné dosiahnuť podobnú funkčnosť nasledovne. Buď sa v trigri nastaví hodnota primárneho kľúča alebo jeho zložky na NULL, druhou možnosťou je zavolať neexistujúcu uloženú procedúru. Keďže prvá možnosť sa nedá využiť pri mazaní riadkov v tabuľke, používa sa druhá. Pri chybe sa najskôr zavolá vytvorená uložená procedúra RAISE_ERROR(), ktorej parametrom je chybová hláška. Tá sa uloží do premennej relácie, z ktorej sa pri zisťovaní chýb načíta. Následne sa volá neexistujúca procedúra TERMINATE(), ktorá spôsobí ukončenie príkazu s chybou.

- *DBTriggerCheck* – náhrada za chýbajúcu konštrukciu CHECK v MySQL, avšak je možné ju využiť aj na zložitejšie kontroly stĺpcov, podmienka v tomto trigri musí vždy platiť, ak neplatí, tak sa zavolá RAISE_ERROR() s nastavenou chybovou hláškou, kontroluje sa pri vkladani a aktualizácii riadkov, namiesto prefixov NEW a OLD pri stĺpcoch sa používa prefix THIS, ktorý sa podľa udalosti správne nastaví na NEW alebo OLD, tento spôsob sa využíva aj v ostatných typoch trigrov
- *DBTriggerDisableInsert* – ak platí podmienka v tomto trigri nepovolí sa vloženie nového riadku do tabuľky, využíva sa aj v administračných formulároch, kde sa kontroluje platnosť podmienky ešte pred zobrazením formulára
- *DBTriggerDisableUpdate* – môže sa týkať celého riadku alebo len vybraných stĺpcov, keď platí podmienka, nepovolí aktualizáciu, v administračných formulároch zakáže meniť dotknuté polia
- *DBTriggerDisableDelete* – v prípade platnosti podmienky sa nepovolí vymazanie riadku z tabuľky

4.3.2 Balíček databázových akcií

Databázové akcie sú takisto ako ostatné akcie odvodené od triedy *Action*. V prípade chyby pri vykonávaní príkazu sa nastaví chybová hláška pomocou statickej metódy *setAlert()* triedy *Alert*.

- *ActionDBInsert* – vloží nový riadok do tabuľky, metódou *setValue()* sa nastavujú jednotlivé vkladané hodnoty
- *ActionDBUpdate* – aktualizuje riadok tabuľky, požadovanými parametrami sú zložky referenčného kľúča, metódou *setValue()* sa nastavujú jednotlivé aktualizované hodnoty

- *ActionDBDelete* – vymaže riadok z tabuľky, požadovanými parametrami sú zložky referenčného kľúča

4.3.3 Balíček formulárov

Formuláre sa vytvárajú pomocou triedy *Form*, ktorej sa nastavuje metóda odoslania dát na server. Táto trieda slúži aj na overenie dát pomocou metódy *validate()*, ktorá volá rovnomennú metódu na všetky polia formulára, overujúcu, či hodnota zadaná užívateľom je pre dané pole platná. Následne je možné bezpečné spracovanie dát.

4.3.3.1 Triedy formulárových prvkov

Formulárové polia odvodené od abstraktnej triedy *FormField* slúžia na získanie údajov od užívateľa. Základnými vlastnosťami sú meno, hodnota, povolenie prázdnej hodnoty, povolenie editácie, menovka a podrobnejší popis. Pred odoslaním formulára sa na strane klienta kontroluje platnosť hodnoty zadanej užívateľom pomocou JavaScriptu. Následne sa kontroluje aj na strane servera metódou *validate()*.

- *FFText* – slúži na vkladanie textových reťazcov rôznych dĺžok, môže byť aj viacrakové
- *FFNumeric* – celé aj desatinné čísla, umožňuje zadať počet všetkých a desatinných miest, minimum, maximum, povolenie záporných čísel
- *FFDate* – dátumové pole
- *FFTime* – pole na zadanie času
- *FFPassword* – slúži na zmenu hesla, vyžaduje zadať nové heslo dva krát
- *FFCheckBox* – zaškrtačacie políčko, je možné zobrazit' viacerými štýlmi
- *FFSelect* – pole na výber jednej hodnoty z vopred definovaných hodnôt, je možné zobrazit' viacerými štýlmi
- *FFMultiSelect* – slúži na výber viacerých hodnôt z vopred definovaných, je možné zobrazit' viacerými štýlmi
- *FForeignField* – určené na zadávanie cudzích kľúčov, podobný *FFSelect*, možnými hodnotami sú riadky z druhej tabuľky

4.3.3.2 Triedy formulárových tlačítiek

Formulárové tlačítka sú odvodené od abstraktnej triedy *FormButton*, slúžia len na vykreslenie.

- *FBSubmit* – potvrdzovacie tlačítko formulára, zašle dáta na server
- *FBReset* – vymaže / nastaví prednastavené hodnoty formulára

- *FBButon* – tlačítko, ktorého funkčnosť je potrebné dedefinovať pomocou JavaScriptu

4.3.4 Balíček administračných tabuliek a formulárov

4.3.4.1 Triedy administračných tabuliek

Administračné tabuľky slúžia na prehľadné zobrazenie databázových tabuliek. Po nastavení databázovej tabuľky umožňujú vybrať z nej stĺpce, ktoré sa majú zobraziť, ako aj stĺpce z iných tabuliek, na ktoré existuje väzba. Je možné nastaviť prednastavený filter a zoradenie, ktoré má užívateľ možnosť meniť. Okrem odkazov na formulár na pridanie a editáciu záznamu sa dajú pridať do tabuľky aj odkazy na zobrazenie ďalších tabuliek a formulárov naviazaných na jednotlivé riadky, rovnako aj volať vopred definované akcie. Štandardnou je akcia vymazania riadku.

- *ADMTable* – administračná tabuľka, zapúzdruje zoznam stĺpcov, odkazov, implicitného filtra a zoradenia
- *ADMTableColum* – stĺpec tabuľky, definuje záhlavie, šírku, zarovnanie
- *ADMTableLink* – odkaz na inú tabuľku, formulár alebo akciu z každého riadku
- *ActionADMTableFilter* – akcia nastaví alebo zruší užívateľský filter nad tabuľkou
- *ActionADMTableOrder* – umožňuje zmeniť zoradenie riadkov tabuľky podľa ľubovoľného stĺpca

4.3.4.2 Triedy administračných formulárov

Administračné formuláre sú stránky odvodené od abstraktnej triedy *ADMForm* so špeciálnymi typmi formulárov slúžiacich na vkladanie a aktualizovanie riadkov v databázach. Zabezpečujú rozmiestnenie prvkov formulára na stránke, kontrolu integritných obmedzení a naplnenie formulárových polí hodnotami. Prvky formulára sú z balíčka formulárov uvedeného vyššie. Taktiež rozširujú triedy databázových akcií k nim prislúchajúcim.

- *ADMFormInsert* – synovská trieda *ADMForm*, pred vykreslením skontroluje podmienky v trigroch *DBTriggerDisableInsert* príslušnej tabuľky, v prípade nemožnosti vloženia nového riadku informuje o tom užívateľa, v opačnom prípade nastaví prvkom formulára implicitné hodnoty z definície stĺpcov tabuľky
- *ActionADMFormInsert* – odvodená od *ActionDBInsert*, skontroluje platnosť hodnôt polí formulára volaním metódy formulára *validate()*, v prípade neúspechu informuje užívateľa o nesprávne vyplnených poliach, v opačnom prípade sa pokúsi vložiť nový riadok do tabuľky, o prípadnom neúspechu informuje užívateľa

- *ADMFormUpdate* – synovská trieda *ADMForm*, pred vykreslením skontroluje podmienky v trigroch *DBTriggerDisableUpdate* príslušnej tabuľky, polia formulára, ktoré nie je možné aktualizovať nastavi ako zakázané, načíta do polí formulára hodnoty stĺpcov vybraného riadku tabuľky
- *ActionADMFormUpdate* – odvodená od *ActionDBUpdate()*, skontroluje platnosť hodnôt polí formulára volaním metódy formulára *validate()*, v prípade neúspechu informuje užívateľa o nesprávne vyplnených poliach, v opačnom prípade sa pokúsi aktualizovať vybraný riadok tabuľky, o prípadnom neúspechu informuje užívateľa

4.4 Užívateľské rozhranie

V rozhraní prístupnom neprihláseným užívateľom a cestujúcim je každá stránka vytvorená samostatne tak, aby sa dosiahla čo najvyššia intuitívnosť a jednoduchosť ovládania. Dáta sa získavajú pomocou vopred pripravených databázových pohľadov, aby nebolo potrebné v skripte vytvárať zložitejšie dotazy. Jedná sa o pohľady na nájdené spojenia, výber tarify a miesta v autobuse, informácie o cestovnom lístku a jeho tarife a prehľad rezervácií.

4.5 Webové prehliadače

Generovanie stránok vo formáte XHTML podľa špecifikácie W3C, používanie kaskádových štýlov CSS taktiež podľa špecifikácie W3C a nepoužívanie neštandardných rozšírení jazyka JavaScript by malo zabezpečovať správnu funkčnosť informačného systému v každom webovom prehliadači. Testované boli prehliadače Microsoft Internet Explorer 6, Mozilla Firefox 2 a Opera 9.

Záver

Práca popisuje návrh a implementáciu informačného systému na predaj a rezerváciu cestovných lístkov pravidelnej autobusovej dopravy. Aplikácia bola navrhnutá, aby sa dala jednoducho rozširovať a udržiavať, taktiež bola snaha zabezpečiť prenositeľnosť medzi viacerými databázovými servermi podporujúcimi jazyk SQL.

Hlavnou výhodou oproti obdobným dostupným systémom je možnosť predaja cestovných lístkov na viacero jász a vedenia okrem bežných taríf aj rôznych časovo obmedzených.

Počas práce vzišli viaceré námety na ďalšie rozšírenie systému. Jednak je to pridanie nových druhov taríf, ako napr. skupinové. Ďalším rozšírením môžu byť vernostné programy, či už typu voľných jász po určitom počte jász, kilometrová banka, apod. Pre dopravcu by mohlo byť užitočné presné evidovanie autobusov, ich jász, vodičov a tvorba turnusov.

Literatura

- [1] CHAPS spol. s r.o.: *Popis formátu a struktury dat pro elektronické zpracování jízdních řádů platných od 1. ledna 2003*, dostupné na URL <http://www.chaps.cz/data/cis/jdf-1.9-a.pdf> (30.7.2007)
- [2] Wikipedia: *Wikipedia, The Free Encyclopedia*, dostupné na URL <http://en.wikipedia.org/> (30.7.2007)
- [3] Achour, M. a kol.: *PHP Manual*, dostupné na URL <http://www.php.net/manual/en/> (30.7.2007)
- [4] MySQL AB: *MySQL 5.0 Reference Manual*, dostupné na URL <http://dev.mysql.com/doc/refman/5.0/en/> (30.7.2007)
- [5] Gutmans, A., Bakken, S. S., Rethans, D.: *Mistrovství v PHP 5*, CP Books, 2005
- [6] Groff, J. R., Weinberg, P. N.: *SQL Kompletní průvodce*, CP Books, 2005
- [7] Musciano, C., Kennedy, B.: *HTML a XHTML Kompletní průvodce*, Computer Press, 2000
- [8] Zendulka, J., Rudolfová, I.: *Databázové systémy IDS – Studijní opora*, FIT VUT v Brně, 2006
- [9] Hruška, T., Křivka, Z.: *Informační systémy (IIS, PIS) – Pojem informačního systému, Data, Procesy, Transakce – Studijní opora*, FIT VUT v Brně, 2006
- [10] Hruška, T., Burget, R.: *Internetové aplikace (WAP) IV. – část JavaScript a PHP – Studijní opora*, FIT VUT v Brně, 2006

Zoznam príloh

Príloha 1: CD s technickou správou vo formáte PDF a zdrojovými kódmi