***ŽILINSKÁ UNIVERZITA V ŽILINE***

*Fakulta riadenia a informatiky*



***Pokročilé databázové systémy***

**Správa železničných vozňov**

Bc. Mário Keméň

Bc. Dominik Klučiar

Bc. Lucia Fedoriková

Bc. Martin Novysedlák

**Správa železničných vozňov (4 študenti)**

Navrhnite a implementujte informačný systém (použiteľný pre železničnú spoločnosť, ako pre firmy , ktoré vlastnia železničné vozne) vrátane dátového modelu, ktorého cieľom bude evidencia a správa železničných vozňov. Vzhľadom na komplexnosť reálneho systému bude cieľom tejto práce iba zjednodušený model, ktorý bude zahŕňať nasledovné:

• Evidencia vozňov (pre rôzne spoločnosti) spolu s históriou údržby až po ich vyradenie. Každý vozeň má svoje identifikačné údaje (viď Prílohu) a informácie o vlastníkovi, ...

• Evidencia pravidelných kontrol, ale aj vrátane samotných opráv (záručný, pozáručný servis), nákupu náhradných dielov, doplnkov a spotrebného materiálu k vozňu (vrátane informácie o mieste a čase opravy a o firme, ktorá vykonávala opravu). Je potrebné evidovať kedy sa uskutočnila daná aktivita, cenu, kto ju uskutočnil ( konkrétny opravca, dodávateľ materiálu, a pod.), typ aktivity, stručný popis.

• Správa vozňov určených na vyradenie a ich vyradenie. (likvidácia, predaj, ...)

• Evidencia pracovníkov, ktorí systém obsluhujú, aby sme vedeli, kto zadával dáta do systému

Požadované metódy PL/SQL:

• pridanie vozňa,

• vyradenie vozňa,

• manažment servisu (vlastné personálne zdroje, výkon externou firmou), nákupu spotrebného materiálu a opravy.

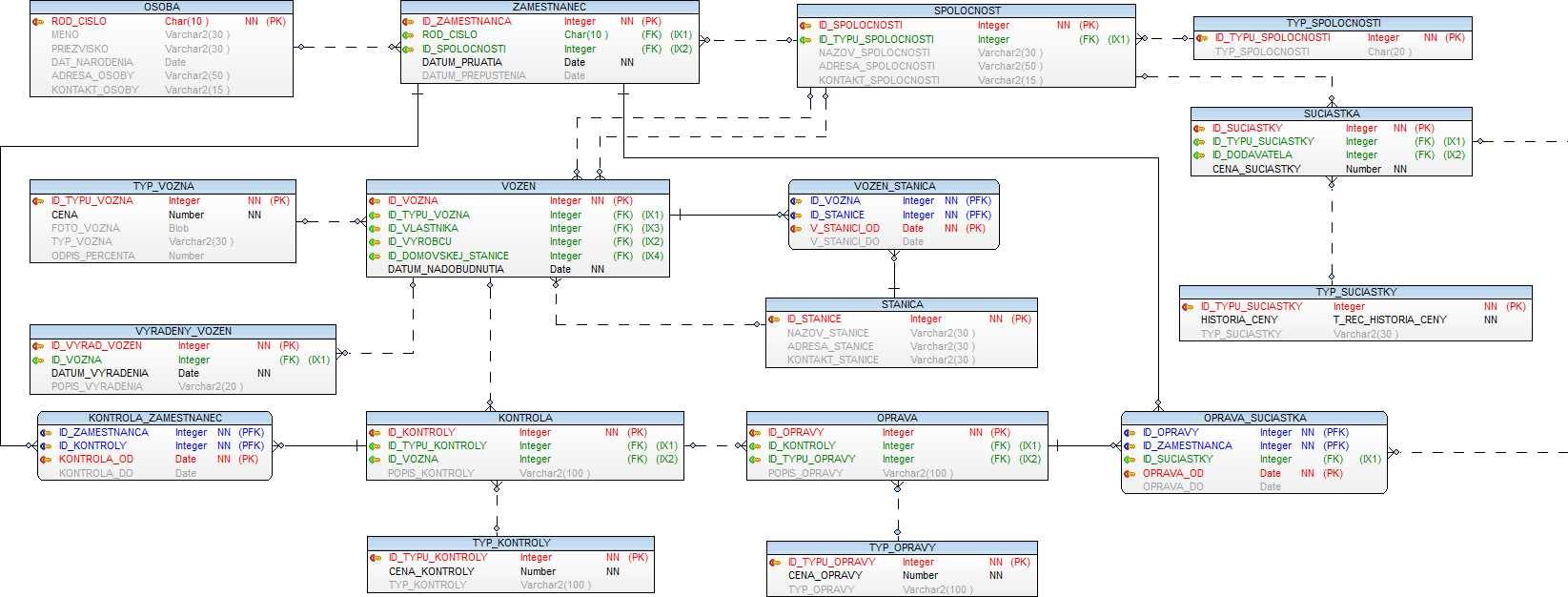
Požadované výstupy:

1. zobrazenie aktuálneho stavu vozňov podľa zadaných kritérií ( podľa typu vozňa, vlastníka, doby nadobudnutia, aktuálnej ceny spolu s prepočtom celkovej ceny vozňa podľa odpisov,
2. zoznam priradeného vozňa konkrétnej železničnej stanici v zadanom čase podľa zadaných kritérií (pozor, jedna stanica môže byť domovskou stanicou viacerých vozňov rôznom v čase) – vypíšte aj informáciu o stave majetku,
3. vypíšte štatistiky o jednotlivých typoch vozňov podľa zadaných kritérií
4. vypíšte životný cyklus vozňa
5. históriu od nadobudnutia, cez informácie o opravách, údržbe až po vyradenie,
6. vypíšte štatistiku nákladov spojených s konkrétnym vozňom alebo za skupiny podľa rôznych kritérií,
7. vypíšte vývoj nákladov na prevádzku jednotlivých typov vozňov za zadané obdobie a zadanou presnosťou (mesačne, polročne, ročne) - výpis troch najporuchovejších vozňov v definovaných kategóriách (typ vozňa, vlastník, výrobca, porucha, obdobie ...),
8. výpis vozňov pre každú organizáciu, ktoré ani po 5tich rokoch prevádzky nepotrebovali žiaden servis (napr. náhradný diel, opravu),
9. výpis vozňov, ktoré musia prejsť v nasledujúcom období (obdobie definované parametrom, napr. 1 mesiac, 3 mesiace, rok, ...) servisnou kontrolou,
10. ku každému vozňu vypísať celkovú cenu opráv, ktoré boli vykonané počas opravy.
11. vyhľadávanie predražených komponentov v definovanom období

porovnanie cien totožných, resp. porovnateľných komponentov, ktoré kúpili jednotlivé organizácie v totožnom období: napr. CARGO kupovalo (totožné) nápravy za polovičnú cenu ako FIRMA1 v roku 2016,

1. výpis vozňov na vyradenie po jednotlivých organizáciách (dôvod vyradenia: vek, opotrebenie, poškodenie, ...),
2. sledovanie počtu vozňov (podľa rôznych kritérií) v definovanej organizácii čase,
3. výpis domovskej stanice, na ktorú je najviac vozňov (po jednotlivých organizáciách a podľa hodnoty),
4. evidencia výkonov zamestnancov podľa rôznych kritérií.

**Dátový model**

****

**Popis tabuliek**

**Osoba –** v tabuľke sú evidované údaje o osobách.

**Zamestnanec** – informácie o ľudoch, ktorí sú zamestnaní ako opravári a kontrolóri.

**Spolocnost** – základné informácie o jednotlivých spoločnostiach.

**Typ\_spolocnosti** – jednotlivé typy spoločností, ktoré sa v modeli vyskytujú (dodávatelia, opravovne, majitelia vozňov).

**Vozen** – informácie, ktoré sa o jednotlivých vozňoch evidujú, vrátane výrobcu a vlastníka vozňa.

**Typ\_vozna** – evidencia informácií o jednotlivých typoch vozňov (lokomotíva, osobný vozeň, ležadlový vozeň, ...) a ich fotografie.

**Vyradeny\_vozen** – informácie o vyradení vozňa (dátum a dôvod vyradenia).

**Stanica** – názov a adresa stanice.

**Vozen\_stanica** – M:N tabuľka, v ktorej sa eviduje v akom období bol vlak v danej stanici.

**Kontrola** – pri vykonávaní kontroly sa evidujú potrebné údaje

**Kontrola\_zamestnanec** – M:N tabuľka, v ktorej sa evidujú informácie o vykonaní kontroly zamestnancom (dátum od kedy do kedy kontrola trvala).

**Typ\_kontroly** – evidované jednotlivé typy kontrol, ktoré vykonávajú zamestnanci.

**Oprava** – informácie o vykonaných opravách, ktoré vychádzajú z kontroly.

**Typ\_opravy** – evidované jednotlivé typy kontrol, ktoré vykonávajú spoločnosti-opravovne.

**Oprava\_suciastka** – M:N tabuľa, v ktorej sa eviduje, aké súčiastky boli v rámci opravy vozňa použité pri oprave.

**Suciastka** – evidované jednotlivé súčiastky a informácie o nich, ktoré dodávatelia dodávajú opravovniam na opravy poškodených vozňov.

**Typ\_suciastky** – zoznam súčiastok, ktoré dodávatelia dodávajú spolu.

**Prepojenie s databázou**

Pomocou nasledovnej metódy sme si v triede DBManager vytvorili spojenie aplikácie s databázou.

public void connect() {

try {

Class.forName("oracle.jdbc.driver.OracleDriver");

conn = DriverManager.getConnection(AppConstants.DB\_CONNECTION\_STRING, AppConstants.DB\_USER, AppConstants.DB\_PASSW);

stmt = conn.createStatement();

System.out.println("Prihlasenie do DB prebehlo uspesne.");

} catch (ClassNotFoundException | SQLException e) {

System.out.println(e.toString());

}

}

**Popis funkcií**

func\_aktCena – funkcia vypočíta aktuálnu cenu vozňa v závislosti od jeho veku a percenta odpisu, o ktorý sa každoročne znižuje cena vozňa.

func\_stavMajetku – funkcia vypočíta aktuálnu cenu majetku (vozňov), ktorý sa aktuálne nachádza v konkrétnej stanici.

func11 – funkcia na výpis predražených komponentov, ktoré boli použité pri opravách vozňov.

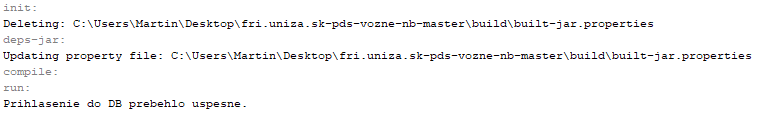
**Popis triggrov**

Triggre sme využili na autoinkrementáciu ID v jednotlivých tabuľkách.

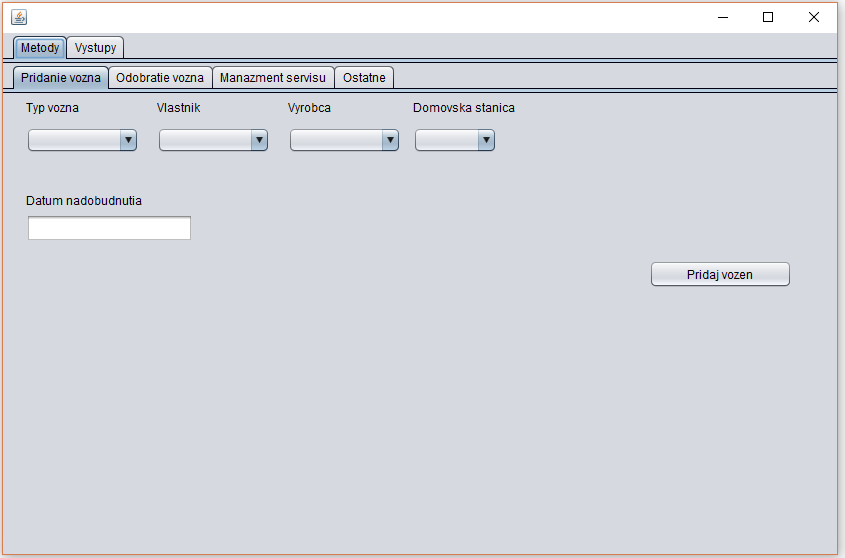
* zamestnanec\_id\_trigger,
* spolocnost\_id\_trigger,
* typ\_spolocnosti\_id\_trigger,
* typ\_vozna\_id\_trigger,
* vozen\_id\_trigger,
* vyradeny\_vozen\_id\_trigger,
* kontrola\_id\_trigger,
* typ\_kontroly\_id\_trigger,
* oprava\_id\_trigger,
* typ\_opravy\_id\_trigger,
* stanica\_id\_trigger,
* suciastka\_id\_trigger,
* typ\_suciastky\_id\_trigger.

**Užívateľská príručka**

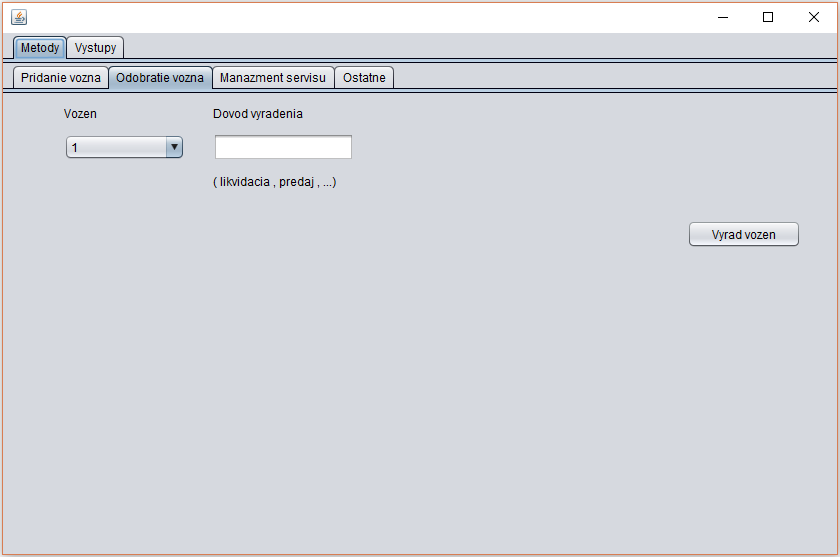
Aplikácia sa po spustení pripojí na základe connectStringu k databáze, po úspešnom prihláseni do databázy sa zobrazí potvrdenie prihlásenia.



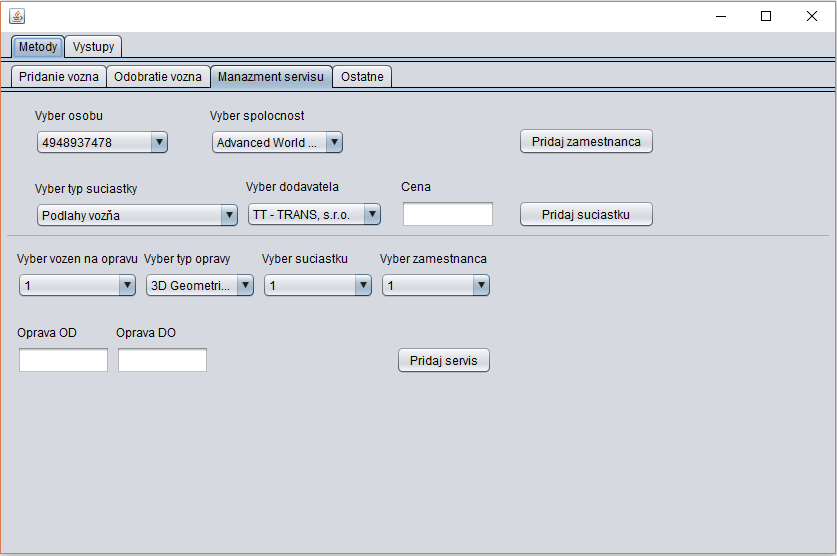
Úvodná obrazovka, ktorá sa zobrazí po spustení aplikácie obsahuje záložku, ktorá umožňuje užívateľovi pridať vozeň. Hodnoty je možné si zvoliť z comboboxov. Tie obsahujú hodnoty nachádzajúce sa v databáze.



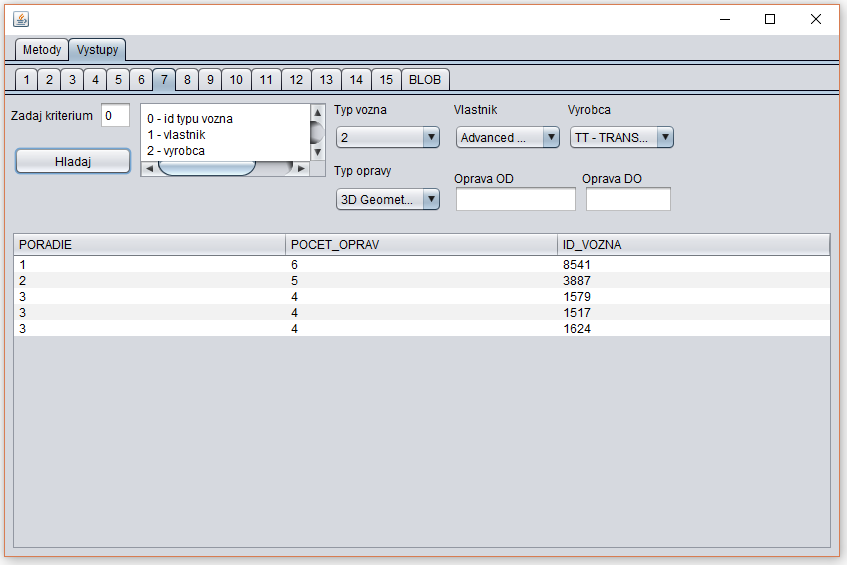
Ďalšia záložka úvodnej strany umožňuje odobratie vozňa – vyradenie vozňa z evidencie. Opäť je možné vybrať si ID vozňa z comboboxu, ktorý obsahuje ID aktuálnych vozňov.



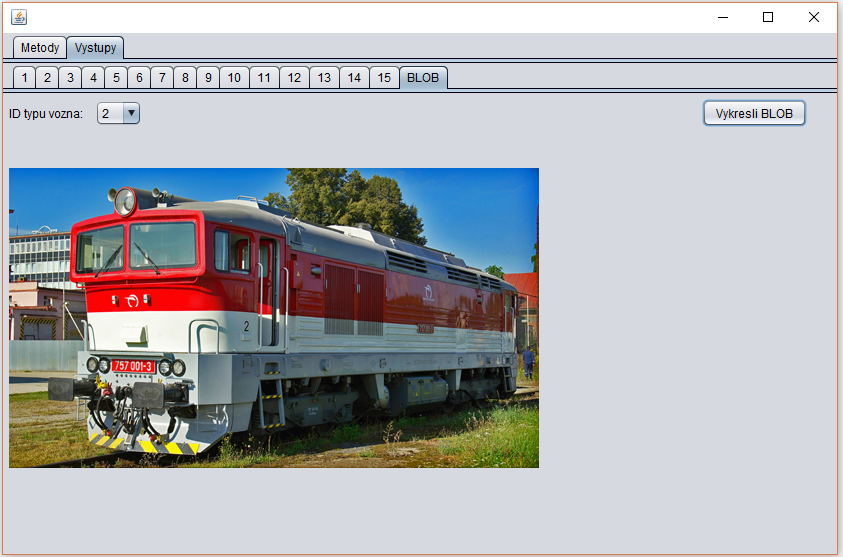
Manažment servisu umožňuje všetky základné funkcie ako pridanie nového zamestnanca do spoločnosti, pridanie novej súčiastky od konkrétneho dodávateľa, vytvorenie záznamu o servise konkrétneho vozňa vrátane použitej súčiastky a zamestnanca, ktorý opravu vykonáva.



Na záložke „Výstupy“ sú pripravené ďalšie záložky, každá pre konkrétny výstup spolu s potrebnými vstupmi, aby boli vypísané vždy správne hodnoty. Používateľ musí zadať kritérium, podľa ktorého chce v databáze vyhľadávať dáta a podľa toho vyplní ďalšie potrebné hodnoty. V tabuľke sa zobrazia výsledky požadovaného výstupu.



Na záložke „BLOB“ sa po zvolení hodnoty „ID typu vozňa“ zobrazí obrázok, konkrétneho typu vozňa.



**Záver**

Cieľom tejto semestrálnej práce bolo vytvoriť dátový model, ktorý bude umožňovať správu železničných vozňov. Správa železničných vozňov predstavuje, ukladanie informácií o vozňoch, spoločnostiach, zamestnancoch, kontrolách a opravách, ktoré sa na vozňoch vykonávajú ale aj o súčiastkach, ktoré sa pri oprave používajú.

Aplikácia bola implementovaná v programovacom jazyku Java vrátane užívateľského prostredia a pripojenia k lokálnej databáze.