

Advanced Programming 2016/17 Project Report

Title: David's Used Cars Shop APP

Student: Savev David

Student ID: 14057

1. Description

The application could be useful in a car shop to manage the collection of cars they sell.

The different sellers can visualize all the cars they are selling and also give information to an eventual buyer filtering by brand, model, color, horse power, production year, price and km; moreover, a seller can add new cars, update same parameters or delete them when are bought.

The admin user can delete all the sellers, while all the other can delete only their own account.

The admin user can not be deleted.

The data (sellers and list of cars) is persistently stored in a Database.

IMPORTANT: the credentials to access the cardealership DB are:

- Username: admin
- Password: admin
- Host: localhost

The DB can be found in the project directory, under name cardealershipdb.sql and it was exported with character set UTF-8.

2. Usage of the programming techniques

Technique	Description
1. Junit Test Cases and Test Suite	Not implemented.
2. Logging	A class named Log was created to manage a proper Logging. Every exception was properly handled and written inside files named log_all (with all the errors encountered) and log_danger (ONLY with serious errors).
3. Exception throwing	I have handled all the exceptions with proper JOptionPane and also by writing the errors inside the log files. Moreover, I created 5 new exceptions, respectively: <ul style="list-style-type: none">• BadlyClosedChooseException (called when an JFileChooser is badly closed);• DeleteAllException (called when all the users or all the cars are being deleted);• ExistingUserException; (called when a someone tries to create a new user with the same name of an existing one);• LoginFailedException (called when a login fails);• WrongFormatException (called when same invalid data is inserted or imported).
4. data I/O (files/JDBC/web)	My application communicates with a Database named cardealership (which contains users table and cars table), by using JDBC driver. The data inside DB can be deleted, updated or new data can be add.
5. Generics and Collections	Inside the class Car I have used an ArrayList<Car> to retrieve the searched cars from the cars table in DB; Inside the class Seller I have an ArrayList<Seller> to retrieve the users from the users table in DB.
6. JavaDoc documentation	JavaDoc properly for each class, constructor, method and variable created. Check inside the doc folder of the project.
7. Design Patterns	Not implemented.

P.S. The information about number of records, deleted records and add records are shown by clicking the "About" Menu Item.

Before running the project, remember that the cardealershipdb.sql file should be imported in the SQL Server! Then, it is enough to launch the Frame.java class of the project (the credentials to access the DB are automatically created and the required external libraries already imported)!