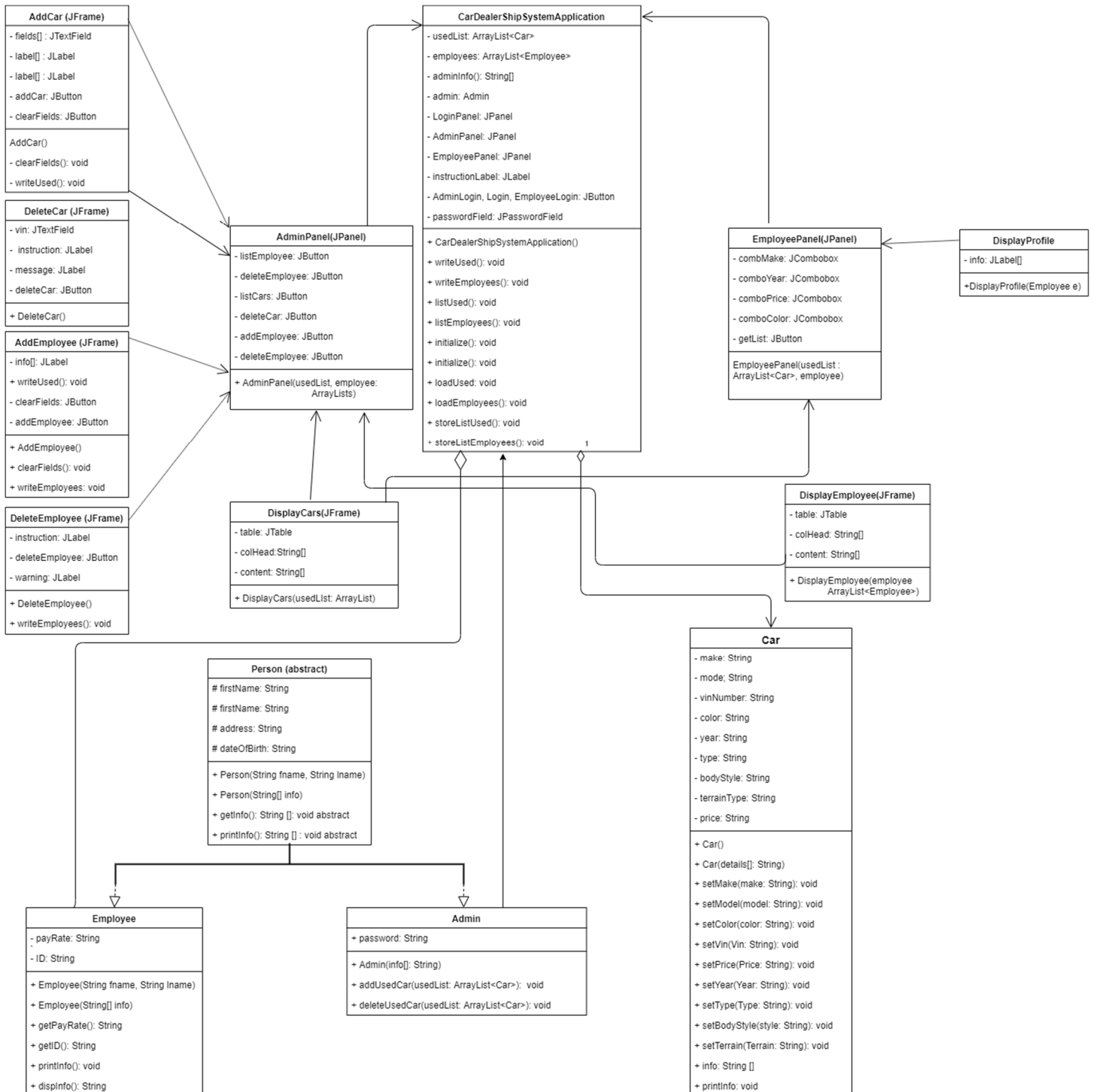


Car Dealership Management System

CSCI 24000 Final Project

Dax Patel - Fall 2019

UML



Goals of the System:

- I. System will have an **Administrator Mode** to handle the following:
 - Inventory
 - Add/Delete/View New Cars
 - Add/Delete/View Used Cars
 - Employee Information:
 - Add Employees.
 - List of employees currently working at the dealership
 - Their pay rate
 - Employee ID Number.
- II. System will have **Employee Mode** which will allow the following:
 - Search for cars using filters like cost, year of manufacture, mileage, etc.
- III. **GUI** : My project is fully functional GUI project. User Will Interact through GUI.

Data Handling:

- The system is heavily reliant on data management. The most important data of my system is car and employee details. My plan to implement car related data is as follows:
 - I. Car Class – which will have following data containers:
 - a. Car body type – Sedan, SUV, Coupe etc.
 - b. Brand Ford, Toyota, Honda, Chevrolet, Lexus etc.
 - c. Car model - Camry, Accord, Focus etc.
 - d. Mileage
 - e. Year of Manufacture
 - f. Sale Price
 - g. VIN number
 - h. Color

- i. Terrain Type
- j. Condition type New or Used
- k. Getters and setters for above fields

II. Person Class (abstract) –

- a. FirstName
- b. Last Name
- c. Address
- d. Date of birth
- e. Getters and setter for above

III. Employee Class – extends person

- a. Employee Identification Number
- b. Pay Rate

IV. Filer class

- a. Contains static methods to sort and filter array list of data. Essentially static class

V. Admin Class – extends person

- a. Password field
- b. Unique methods

Tools Required:

- Tesla To Compile
- Visual Studio Code
- R – for gathering and manipulating car data
- Git bash and java JRE 8 for GUI.
- Eclipse (if needed).

Algorithm (for more details about components look at GUI) :

- Java Window Application
 - Login/Logout capability (As admin and employee)
 - 3 Panels stacked on application window with card layout.
 - Admin Panel – if admin logs in. Will have buttons such as add/delete car, add/delete employee, list inventory, list employee.
 - Employee Panel – If any employee logs in. Will have options to filter inventory in different ways using make, price, year and color.
 - Login Panel – To login as employee or admin and when either logs back out, they will be taken back to sign in panel.
- Add/delete cars will open their own frames. Similarly add/delete employees will open up their individual frames, and so will list car/employee.
- Admin and Employee class will inherit from person. This is done in order for me to expand project by including customer and transaction history of the dealership. I did not intend this idea for final project. I plan to implement it in my free time.
- UsedInventory.csv and Employee.csv will be two main files to be managed.
- Data will be stored and retrieved using array list serialization. If there exists no prior serialization, two functions createSerialUsed and createSerialEmployee will create them. These are called on every program run.
- When changes made, csv files will be rewritten to keep up to date info using storeList methods.
- **Employee, Admin and Car class will have methods that return array of string containing their info. This is especially useful to create tables for better view.**