**Repository Location: https://github.com/awbriskman/Backend-programming-week-18-assignment.git**

**Tasks (1.2)**

Theresa:

* Create SQL schema and starter data for vehicles (min 4 rows) and customers (min 5 rows) tables
* Create starter data for transactions table (3 rows, 2 are unique cars sold to different customers by same employee and dealership)
* Create controller, service and dao layers for Insert operations on customers table
* Participated in debugging sessions on Tuesday and Thursday

Jennifer:

* Create file structure for MySQL
* Create SQL schema for transactions table
* Create controller, service, and dao layers for all CRUD operations on locations table
* Participated in debugging sessions on Tuesday and Thursday

Melissa:

* Create SQL schema, and starter data for employees and locations tables
* Create controller, service and dao layers for Delete operations on customers table
* Participated in debugging sessions on Tuesday and Thursday

Alex:

* Create file structure for project
* Create controller, service, and dao layers for all CRUD operations on transactions table
* Create entity for all tables (vehicles, customers, transactions, locations, employees)
* Create controller, service, and dao layers for Update and Get operations on customers table
* Participated in debugging sessions on Tuesday and Thursday

Robert:

* Create controller, service, and dao layers for all CRUD operations on employees table
* Participated in debugging sessions on Tuesday and Thursday

Phil:

* Create controller, service, and dao layers for all CRUD operations on vehicles table
* Participated in debugging sessions on Tuesday and Thursday

**Table schema info (1.2)**

|  |  |
| --- | --- |
| vehicles table   * vehicleID (pk) * locationID (fk) * is\_sold * price * model * make   customers table  - customerID (pk)  - name  - address  - phone  employees table  - employeeID (pk)  - locationID (fk)  - name | transactions table  - transactionID (pk)  - vehicleID (fk)  - customerID (fk)  - locationID (fk)  - datestamp  locations table  - locationID (pk)  - location\_name |

**API Documentation (1.0)**

vehicles table API documentation

|  |  |
| --- | --- |
| **HTTP Verb and Method Name** | **What it does** |
| GET---- | Get all vehicles |
| GET: getVehicle | Get a vehicle by id |
| PUT: updateVehicle | Update a vehicles info by id |
| DELETE: deleteVehicle | Delete a vehicle by id |
| POST: newVehicle | Create a new vehicle |

customers table API documentation

|  |  |
| --- | --- |
| **HTTP Verb and Method Name** | **What it does** |
| GET: listCustomers | Get all customers |
| GET---- | Get a customer by id |
| PUT: updateCustomer | Update a customer’s info by id |
| DELETE: deleteCustomer | Delete a customer by id |
| POST: newCustomer | Create a new customer |

employees table API documentation

|  |  |
| --- | --- |
| **HTTP Verb and Method Name** | **What it does** |
| GET: listEmployees | Get all employees |
| GET---- | Get an employee by id |
| PUT: updateEmployee | Update an employee’s info by id |
| DELETE: deleteEmployee | Delete an employee by id |
| POST: newEmployee | Create an new employee |

transactions table API documentation

|  |  |
| --- | --- |
| **HTTP Verb and Method Name** | **What it does** |
| GET---- | Get all transactions |
| GET: getTransactions (sic) | Get a transaction by id |
| PUT: updateTransaction | Update a transaction’s info by id |
| DELETE: deleteTransaction | Delete a transaction by id |
| POST: newTransaction | Create a new transaction |

locations table API documentation

|  |  |
| --- | --- |
| **HTTP Verb and Method Name** | **What it does** |
| GET ---- | Get all locations |
| GET: getLocation | Get a location by id |
| PUT: updateLocation | Update a location’s info by id |
| DELETE: deleteLocation | Delete a location by id |
| POST: newLocation | Create a new location |

Get vehicle

Graphical user interface, text, application

Description automatically generated

Update vehicle

Graphical user interface, text

Description automatically generated

Delete vehicle

A screenshot of a computer

Description automatically generated with medium confidence

New vehicle

Graphical user interface, text

Description automatically generated

List customers

A picture containing text, screenshot, screen

Description automatically generated

Update customer

A screenshot of a computer

Description automatically generated with medium confidence

Delete customer

Graphical user interface, text

Description automatically generated

New customer

A screenshot of a computer

Description automatically generated with medium confidence

List employees

Graphical user interface, text, application

Description automatically generated

Update employee

Graphical user interface, text, application

Description automatically generated

Delete employee

Graphical user interface, text, application

Description automatically generated

New employee

Graphical user interface, text, application

Description automatically generated

Get transaction

Graphical user interface, text, application

Description automatically generated

Update transaction

Graphical user interface, text

Description automatically generated

Delete transaction

A screenshot of a computer

Description automatically generated with medium confidence

New transaction

Graphical user interface, text

Description automatically generated

Get location

Graphical user interface, text, application, email

Description automatically generated

Update location

A screenshot of a computer

Description automatically generated with medium confidence

Delete location

A screenshot of a computer

Description automatically generated with medium confidence

New location

A screenshot of a computer

Description automatically generated with medium confidence