



Group 4: Tire Shop

Adamo, Dallas, Daniel, Pablo

Deliverable of the Week

- Reports
- Search page
- Overall flow of the application
- Hotfixes

Code Implementation:

```
@FXML|
private TableView<Employee> employeeTable;
@FXML
private ObservableList<Employee> masterData = FXCollections.observableArrayList();
```

```
String query = "Select * from employees";
ResultSet rs;
try {
    rs = con.createStatement().executeQuery(query);
    while (rs.next()) {
        //Creates an employee using data from the database then adds it to masterData
        Employee test = new Employee(rs.getInt("employeeID"), rs.getString("firstName"),
            rs.getString("lastName"), rs.getString("password"), rs.getDate("startDate"),
            rs.getBoolean("isAdmin"));
        masterData.add(test);
    }
} catch (SQLException e) {
    e.printStackTrace();
}
//Sets the list of employees in to the table
employeeTable.setItems(masterData);
```

Code Implementation:

```
FilteredList<Tire> filteredData = new FilteredList<>(masterData, p -> true);

filterField.textProperty().addListener((observable, oldValue, newValue) -> {
    filteredData.setPredicate(tire -> {
        // If filter text is empty, display all Tires.
        if (newValue == null || newValue.isEmpty()) {
            return true;
        }

        // Compare tire name and tire brand of every Tire with filter text.
        String lowerCaseFilter = newValue.toLowerCase();

        if (tire.getName().toLowerCase().contains(lowerCaseFilter)) {
            return true; // Filter matches tire name
        } else if (tire.getBrand().toLowerCase().contains(lowerCaseFilter)) {
            return true; // Filter matches brand
        }
        return false; // Does not match.
    });
});

SortedList<Tire> sortedData = new SortedList<>(filteredData);

sortedData.comparatorProperty().bind(tireTable.comparatorProperty());

tireTable.setItems(sortedData);
```

Troubleshooting:

- Resizing of all gui windows -- along with allowing for scalable backgrounds.
- Redesigning of every GUI framework from AnchorPane containers to StackPane Containers.
- Understanding how administration rights would be given out via end user after implementation.
- Mysql integration to a cloud server database without using PHP techniques.

Where are we going?

1. Being able to add and remove employees using admin.
2. Validation for customer entry when adding a new customer.
3. Authorization rights with correlating constraints depending on the end user's level of accessibility.

