

Отчет по практикуму по web-технологиям.

В данной работе было реализовано веб-приложение, выполняющее функцию информационной системы автосалона. Работа была проделана на основе фреймворка spring с использованием технологии selenium. Для базы данных была выбрана СУБД postgresql. Работа с базой данных и менеджмент сессий производится с помощью библиотеки hibernate. С помощью данного приложения администратор может вести учет клиентов, заказов и автомобилей в автосалоне. Возможно изменение информации, добавление новых объектов, удаление старых и выборка по различным признакам. При удалении возможен каскадный эффект, например, при удалении марки автомобилей удаляются все автомобили этой марки или при удалении клиента удаляются все его заказы.

Архитектура приложения:

При деплое приложения пользователь первым делом попадает на домашнюю страницу

Car showroom Information

[Cars](#)
[Orders](#)
[Clients](#)

Ссылки ведут на страницы со списком всех автомобилей, заказов и клиентов. Принцип работы с каждой из этих трех сущностей почти одинаков, поэтому рассмотрим поближе только сценарии взаимодействия с сущностью “cars”.

При нажатии на соответствующую ссылку пользователь перенаправляется на страницу со списком всех автомобилей.

All present cars. Its count is 10

Registration_number	Model	Developer	Price
<input type="text" value="Search by registration number.."/>	<input type="text" value="Search by model.."/>	<input type="text" value="Search by developer.."/>	<input type="text" value="Search by price .."/>
hhsh99	Kalina	Lada	267654.0
rewe31	Felicia	Lada	1000000.0
fdwg54	Kalina	Lada	421113.0
dsdw23	Fusion	Lada	544321.0
gasw23	Civic	Honda	3212123.0
gez62	Felicia	Lada	678921.0
sqwe13	Civic	Honda	900123.0
zaf78	Fusion	Lada	651876.0
trew54	Qashqai	Nissan	523323.0
zasw12	Accord	Honda	554311.0

Можно производить фильтрацию по любым полям, например, по полю developer.

Registration_number	Model	Developer	Price
<input type="text" value="Search by registration number.."/>	<input type="text" value="Search by model.."/>	<input type="text" value="H"/>	<input type="text" value="Search by price .."/>
gasw23	Civic	Honda	3212123.0
sqwe13	Civic	Honda	900123.0
zasw12	Accord	Honda	554311.0

Значение номера является ссылкой на страницу с информацией о данном автомобиле и формой для ее редактирования.

Car with registration number qasw23

Model	Registration number	Number of seats	Upholstery type	Price	Last maintance date	Test-drive clients	Mileage	Car colour	Engine power	Number of doors	Developer	Devices
Civic	qasw23	5	Olefin	3212123.0	2017-09-09 00:00:00.0	Rodion Sulzhenko	5673.0	Green	75.0	4	Honda	video conditioner window regulator

Update car

Registration number

aa000aa

Devices

window regulator

airbag

massage chair

Test-drive clients

Vadim Vakhrushev

Rodion Sulzhenko

Alexey Kondashin

Upholstery type

Leather

Car colour

Green

Last maintance date

yyyy-mm-dd

Car price

0.0

Engine power

0.0

Number of doors

0

Number of seats

0

Mileage

0.0

Update car

Delete this car

Delete all cars with the same model

Home

Ввод некорректных данных контролируется при создании базы данных путем наложения ограничения на значения столбцов. Например, если попробовать ввести отрицательную цену, при попытке сабмита будет сгенерирована страница, сообщающая об ошибке.

Internal server error occured, sorry

Внизу страницы находятся кнопки удаления данного автомобиля, удаления модели, к которой данный автомобиль принадлежит и кнопка возврата на главную страницу сайта. Также на странице со списком автомобилей присутствует кнопка Add car, перенаправляющая на страницу добавление новой машины и кнопка Add model, перенаправляющая на страницу добавления новой модели.

Add new car

Registration number	<input type="text" value="aa000aa"/>
Devices	<div><div>GPS-navigator</div><div>radio</div><div>video</div></div>
	<div><div>Vadim Vakhrushev</div><div>Rodion Sulzhenko</div><div>Alexey Kozdetskiy</div></div>
Test-drive clients	
Developer	<input type="text" value="Shkoda"/>
Model	<input type="text" value="Octavia"/>
Upholstery type	<input type="text" value="Leather"/>
Car colour	<input type="text" value="Green"/>
Last maintance date	<input type="text" value="yyyy-mm-dd"/>
Car price	<input type="text" value="0.0"/>
Engine power	<input type="text" value="0.0"/>
Number of doors	<input type="text" value="0"/>
Number of seats	<input type="text" value="0"/>
Mileage	<input type="text" value="0.0"/>
<input type="button" value="Add car"/>	

Add new model

Model title	<input type="text" value="enter the title.."/>
Developer	<input type="text" value="Shkoda"/>
<input type="button" value="Add model"/>	

Схожим образом происходит взаимодействие с клиентами и заказами.

Данное приложение тестировалось в два этапа:

Сначала для каждого метода каждого DAO-класса был написан модульный тест, проверяющий его корректность. Это первая тестовая “сюита”.

Далее было написано пять системных тестов: три из них полностью покрывают логику работы с клиентами, заказами и автомобилями соответственно, а оставшиеся два – тестируют реакцию сервера на некорректное обращение клиента.

Итак, исходя из тестов, можно сделать вывод, что данное приложение удовлетворяет бизнес-логике небольшого автосалона, а тесты обеспечивают корректность сценариев взаимодействия пользователя с приложением.