

ОТЧЕТ о выполнении лабораторной работы №7
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики
Кафедра информационных технологий

Выполнил студент группы МО32/1: Г-А.Ш.Шурлаев

Направление подготовки: 02.03.03 Математическое обеспечение и администрирование информационных систем

Курс: 3

Отчет принял: кандидат педагогических наук, доцент Н. Ю. Добровольская

Краснодар, 2025 г.

СОДЕРЖАНИЕ

1. Введение
 2. Проектирование структур данных
 3. Программные компоненты и диаграмма компонентов UML
 4. Физическая архитектура и диаграмма развертывания UML
 5. Заключение
-

ВВЕДЕНИЕ

Тема: Разработка архитектуры программного продукта

Цель: Приобретение навыков проектирования физической архитектуры ПП на языке UML, а также разработки базы данных для ПП.

Индивидуальная тема: Разработка ПО для автоматизации продаж автомобилей в автосалоне

Задачи:

1. Разработать решения по организации структур данных в ПП на основе диаграммы классов UML.
 2. Выполнить описание структур данных на уровне атомарных атрибутов с указанием типов данных, ограничений и значений по умолчанию.
 3. Представить нормализованную модель данных БД, список таблиц и описание связей между ними.
 4. Перечислить и описать программные компоненты ПП с указанием стереотипов.
 5. Построить диаграмму компонентов UML.
 6. Описать физические элементы комплекса технических средств.
 7. Построить диаграмму развертывания UML.
-

1. ПРОЕКТИРОВАНИЕ СТРУКТУР ДАННЫХ

1.1 Описание структур данных

На основе диаграммы классов UML из лабораторной работы №5 и технического задания разработана реляционная база данных в MySQL, нормализованная до 3НФ.

Таблица 2.6.1 — Структура данных clients (Клиенты)

Атрибут	Тип данных	Размер, байт	Условие на значение	Значение по умолчанию	Примечание
client_id	INT	4	PRIMARY KEY, AUTO_INCREMENT	-	Уникальный идентификатор
full_name	VARCHAR(100)	100	NOT NULL	-	ФИО клиента
passport_data	VARCHAR(20)	20	NOT NULL, UNIQUE	-	Паспортные данные
phone	VARCHAR(15)	15	NOT NULL	-	Контактный телефон
email	VARCHAR(100)	100	UNIQUE	NULL	Электронная почта
address	VARCHAR(200)	200	-	NULL	Адрес проживания
created_at	TIMESTAMP	8	NOT NULL	CURRENT_TIMESTAMP	Дата регистрации

Таблица 2.6.2 — Структура данных cars (Автомобили)

Атрибут	Тип данных	Размер, байт	Условие на значение	Значение по умолчанию	Примечание
car_id	INT	4	PRIMARY KEY, AUTO_INCREMENT	-	Уникальный идентификатор
brand	VARCHAR(50)	50	NOT NULL	-	Марка автомобиля
model	VARCHAR(50)	50	NOT NULL	-	Модель автомобиля
year	YEAR	1	CHECK (year >= 2000 AND year <= 2025)	-	Год выпуска
vin_code	VARCHAR(17)	17	NOT NULL, UNIQUE	-	VIN-номер
color	VARCHAR(30)	30	-	NULL	Цвет
price	DECIMAL(12,2)	8	CHECK (price > 0)	-	Цена в рублях
status	ENUM	1	NOT NULL	'в наличии'	Статус: 'в наличии', 'забронирован', 'продан'
configuration	TEXT	-	-	NULL	Комплектация

Таблица 2.6.3 — Структура данных requests (Заявки)

Атрибут	Тип данных	Размер, байт	Условие на значение	Значение по умолчанию	Примечание
request_id	INT	4	PRIMARY KEY, AUTO_INCREMENT	-	Уникальный идентификатор
client_id	INT	4	FOREIGN KEY (clients)	-	Ссылка на клиента
car_id	INT	4	FOREIGN KEY (cars)	-	Ссылка на автомобиль
request_date	TIMESTAMP	8	NOT NULL	CURRENT_TIMESTAMP	Дата создания заявки
status	ENUM	1	NOT NULL	'создана'	Статус: 'создана', 'прверена', 'забронирован авто', 'оплачена', 'отклонена'
comments	TEXT	-	-	NULL	Комментарии

Таблица 2.6.4 — Структура данных contracts (Договоры)

Атрибут	Тип данных	Размер, байт	Условие на значение	Значение по умолчанию	Примечание
contract_id	INT	4	PRIMARY KEY, AUTO_INCREMENT	-	Уникальный идентификатор
request_id	INT	4	FOREIGN KEY (requests), UNIQUE	-	Ссылка на заявку
contract_number	VARCHAR(20)	20	NOT NULL, UNIQUE	-	Номер договора
contract_date	DATE	3	NOT NULL	-	Дата заключения
status	ENUM	1	NOT NULL	'черновик'	Статус: 'черновик', 'на согласовании', 'согласован', 'исполнен', 'отклонен'
contract_text	TEXT	-	-	NULL	Текст договора
approved_by	INT	4	FOREIGN KEY (employees)	NULL	Кто утвердил

Таблица 2.6.5 — Структура данных payments (Платежи)

Атрибут	Тип данных	Размер, байт	Условие на значение	Значение по умолчанию	Примечание
payment_id	INT	4	PRIMARY KEY, AUTO_INCREMENT	-	Уникальный идентификатор
contract_id	INT	4	FOREIGN KEY (contracts), UNIQUE	-	Ссылка на договор
amount	DECIMAL(12,2)	8	CHECK (amount > 0)	-	Сумма платежа
payment_date	TIMESTAMP	8	NOT NULL	CURRENT_TIMESTAMP	Дата оплаты
payment_method	ENUM	1	NOT NULL	-	Способ: 'наличные', 'карта', 'перевод'
status	ENUM	1	NOT NULL	'ожидает'	Статус: 'ожидает', 'получена', 'возврат'

Таблица 2.6.6 — Структура данных employees (Сотрудники)

Атрибут	Тип данных	Размер, байт	Условие на значение	Значение по умолчанию	Примечание
employee_id	INT	4	PRIMARY KEY, AUTO_INCREMENT	-	Уникальный идентификатор
full_name	VARCHAR(100)	100	NOT NULL	-	ФИО сотрудника
role	ENUM	1	NOT NULL	-	Роль: 'кассир', 'руководитель', 'администратор'
login	VARCHAR(50)	50	NOT NULL, UNIQUE	-	Логин для входа
password_hash	VARCHAR(255)	255	NOT NULL	-	Хэш пароля
is_active	BOOLEAN	1	NOT NULL	TRUE	Активен ли аккаунт

1.2 Нормализованная модель данных

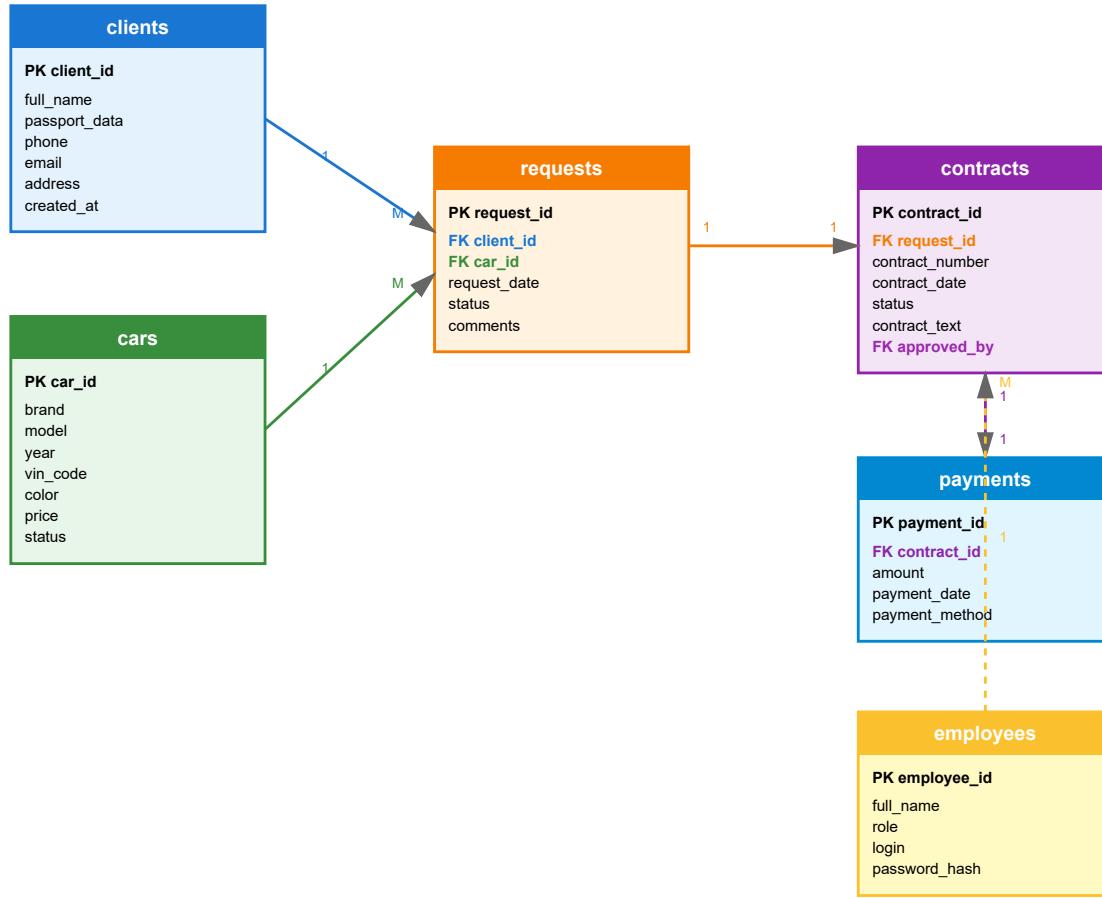
Таблица 2.7 — Список разработанных таблиц БД

№ п/п	Имя таблицы	Описание
1	clients	Данные о клиентах автосалона
2	cars	Каталог автомобилей в наличии
3	requests	Заявки на покупку автомобилей
4	contracts	Договоры купли-продажи
5	payments	Финансовые транзакции
6	employees	Сотрудники автосалона
7	sales_reports	Отчёты о продажах (агрегированные данные)

Таблица 2.8 — Связи между таблицами БД

Родительская таблица	Атрибут PK	Дочерняя таблица	Атрибут FK	Тип связи
clients	client_id	requests	client_id	1:M
cars	car_id	requests	car_id	1:M
requests	request_id	contracts	request_id	1:1
contracts	contract_id	payments	contract_id	1:1
employees	employee_id	contracts	approved_by	1:M

1.3 Схема базы данных Рисунок 1.1 — Схема базы данных для автосалона



Описание: База данных нормализована до 3НФ. Все связи реализованы через внешние ключи с поддержкой ссылочной целостности. Статусы реализованы через ENUM для обеспечения контроля допустимых значений.

2. ПРОГРАММНЫЕ КОМПОНЕНТЫ И ДИАГРАММА КОМПОНЕНТОВ UML

2.1 Описание программных компонентов

Таблица 2.9 — Перечень программных компонентов ПП

№	Имя	Стереотип	Описание
1	ClientForm.jsp	< >	Веб-форма регистрации клиента
2	CarCatalog.jsp	<>	Каталог автомобилей для просмотра
3	RequestForm.jsp	<>	Форма создания заявки
4	ContractForm.jsp	<>	Форма работы с договором
5	PaymentForm.jsp	<>	Форма приема оплаты
6	ReportView.jsp	<>	Интерфейс просмотра отчетов
7	ClientController.java	<>	Контроллер управления клиентами
8	CarController.java	<>	Контроллер управления автомобилями
9	RequestController.java	<>	Контроллер обработки заявок
10	ContractController.java	<>	Контроллер работы с договорами
11	PaymentController.java	<>	Контроллер обработки платежей
12	ReportService.java	<>	Сервис формирования отчетов
13	ValidationService.java	<>	Сервис валидации данных
14	CalculationService.java	<>	Сервис расчета стоимости
15	NotificationService.java	<>	Сервис уведомлений
16	ClientDAO.java	<>	Слой доступа к данным клиентов
17	CarDAO.java	<>	Слой доступа к данным автомобилей
18	RequestDAO.java	<>	Слой доступа к заявкам
19	ContractDAO.java	<>	Слой доступа к договорам
20	PaymentDAO.java	<>	Слой доступа к платежам
21	database.sql	<>	Скрипт инициализации БД
22	config.properties	<>	Конфигурационный файл
23	application.jar	<>	Исполняемый файл приложения
24	mysql-connector.jar	<>	JDBC-драйвер MySQL
25	spring-framework.jar	<>	Библиотека Spring Framework
26	user_manual.pdf	<>	Руководство пользователя

2.1 Описание программных компонентов

Таблица 2.9 — Перечень программных компонентов ПП

№	Имя	Стереотип	Описание
1	ClientForm.jsp	<<form>>	Веб-форма регистрации клиента
2	CarCatalog.jsp	<<form>>	Каталог автомобилей для просмотра
3	RequestForm.jsp	<<form>>	Форма создания заявки
4	ContractForm.jsp	<<form>>	Форма работы с договором
5	PaymentForm.jsp	<<form>>	Форма приема оплаты
6	ReportView.jsp	<<form>>	Интерфейс просмотра отчетов
7	ClientController.java	<<source>>	Контроллер управления клиентами
8	CarController.java	<<source>>	Контроллер управления автомобилями
9	RequestController.java	<<source>>	Контроллер обработки заявок
10	ContractController.java	<<source>>	Контроллер работы с договорами
11	PaymentController.java	<<source>>	Контроллер обработки платежей
12	ReportService.java	<<source>>	Сервис формирования отчетов
13	ValidationService.java	<<source>>	Сервис валидации данных

13	ValidationService.java	<<source>>	Сервис валидации данных
14	CalculationService.java	<<source>>	Сервис расчета стоимости
15	NotificationService.java	<<source>>	Сервис уведомлений
16	ClientDAO.java	<<source>>	Слой доступа к данным клиентов
17	CarDAO.java	<<source>>	Слой доступа к данным автомобилей
18	RequestDAO.java	<<source>>	Слой доступа к заявкам
19	ContractDAO.java	<<source>>	Слой доступа к договорам
20	PaymentDAO.java	<<source>>	Слой доступа к платежам
21	database.sql	<<data>>	Скрипт инициализации БД
22	config.properties	<<config>>	Конфигурационный файл
23	application.jar	<<executable>>	Исполняемый файл приложения
24	mysql-connector.jar	<<library>>	JDBC-драйвер MySQL
25	spring-framework.jar	<<library>>	Библиотека Spring Framework
26	user_manual.pdf	<<document>>	Руководство пользователя

Таблица 2.10 — Типы компонентов приложения

Стереотип	Тип компонента
<>	файл исходного текста программы
<>	файл интерфейсной формы или web-страницы
<>	исполняемый файл
<>	файл конфигурации и/или настроек системы
<>	статическая или динамическая библиотека
<>	файл документации
<>	файл данных
<>	
таблица базы данных	

Таблица 2.10 — Типы компонентов приложения

Стереотип	Тип компонента
<<source>>	файл исходного текста программы
<<form>>	файл интерфейсной формы или web-страницы
<<executable>>	исполняемый файл
<<config>>	файл конфигурации и/или настроек системы
<<library>>	статическая или динамическая библиотека
<<document>>	файл документации
<<data>>	файл данных
<<table>>	таблица базы данных

2.2 Диаграмма компонентов UML Рисунок 2.1 — Диаграмма компонентов UML

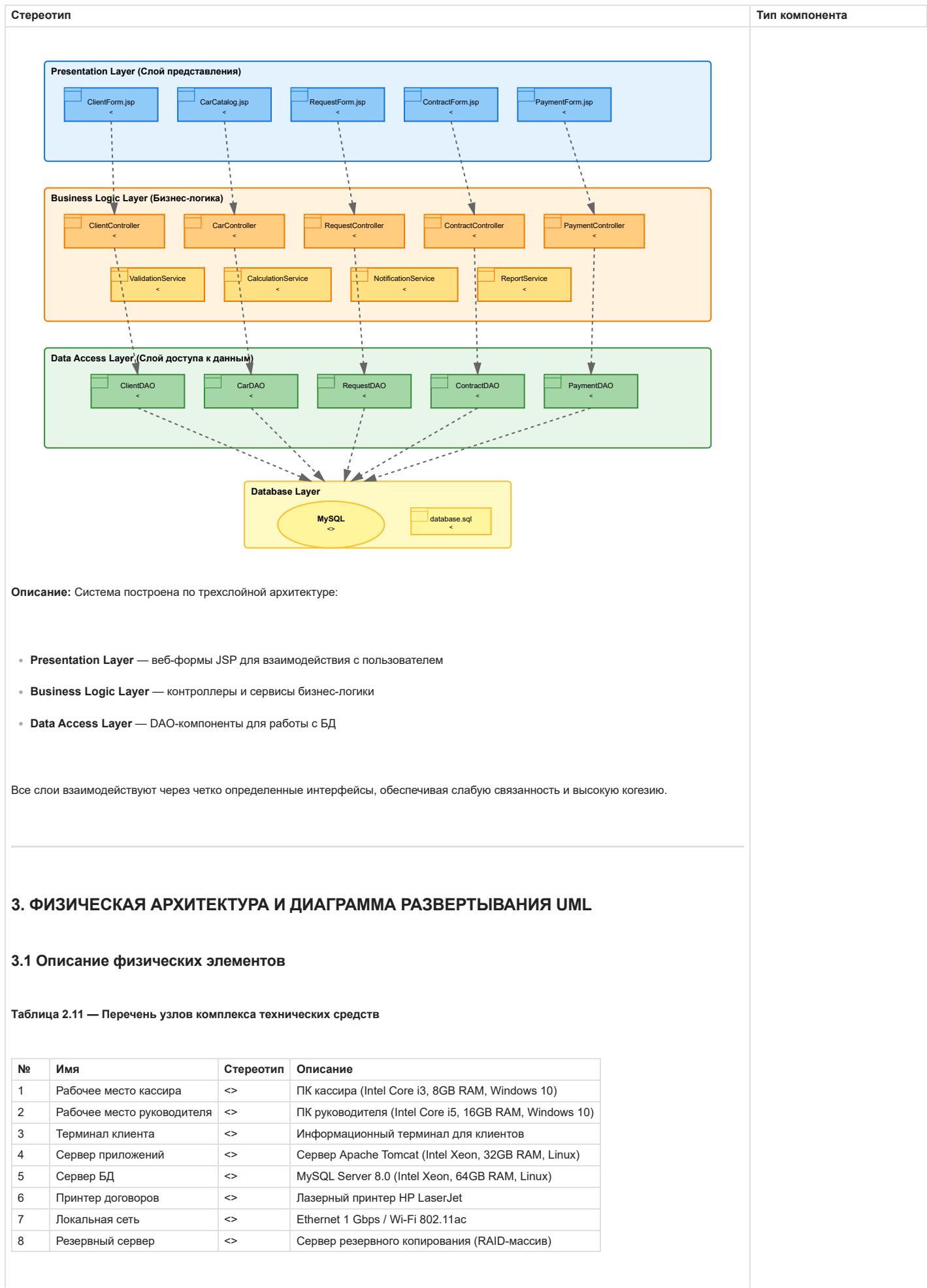
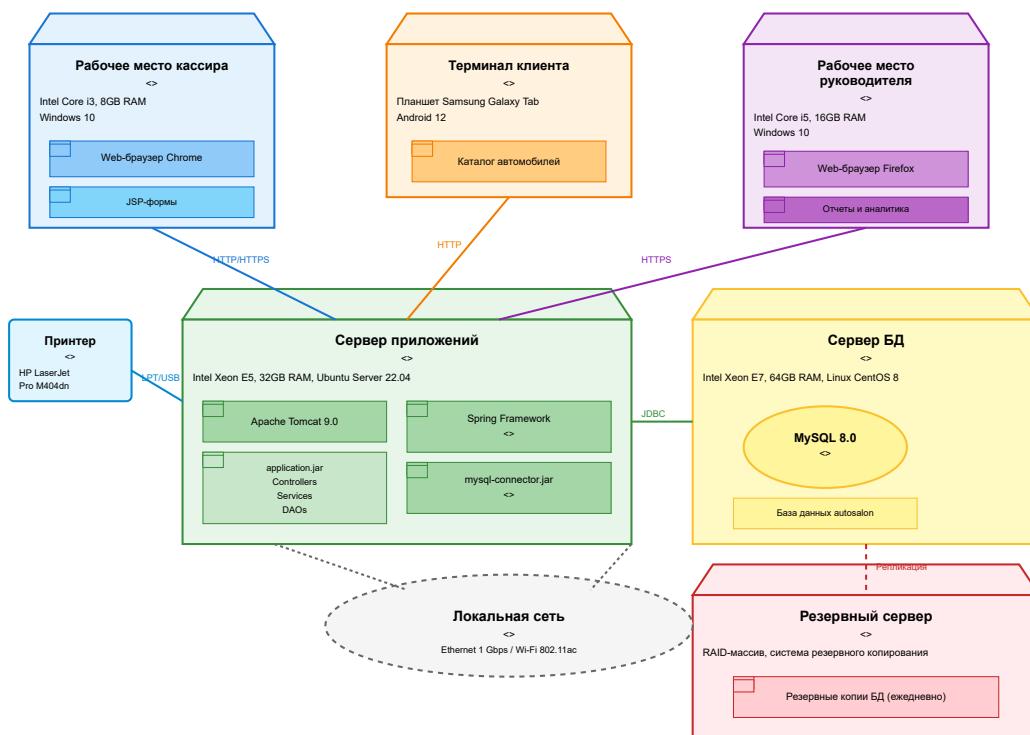


Таблица 2.12 — Типы элементов диаграмм развертывания UML

Стереотип	Тип элемента
<>	Ресурсоемкий узел
<>	Нересурсоемкий узел (устройство)
<>	Измерительное устройство (датчик)
<>	Печатающее устройство (принтер)
<>	Сетевая передающая среда
<>	База данных
<>	Узел, не попадающий под указанные типы

3.2 Диаграмма развертывания UML Рисунок 3.1 — Диаграмма развертывания UML для автосалона

**Описание:**

Физическая архитектура включает:

- **Клиентская часть:** рабочие места кассира и руководителя, информационный терминал для клиентов
- **Серверная часть:** сервер приложений Apache Tomcat с развернутым Java-приложением, сервер БД MySQL
- **Резервирование:** отдельный сервер для резервного копирования данных с ежедневной репликацией
- **Периферия:** принтер для печати договоров
- **Сеть:** локальная сеть Ethernet/Wi-Fi для объединения всех узлов

Взаимодействие осуществляется по протоколам HTTP/HTTPS между клиентами и сервером приложений, JDBC между сервером приложений и БД.

Стереотип	Тип компонента
ЗАКЛЮЧЕНИЕ	
	<p>В ходе выполнения лабораторной работы №7 была успешно разработана архитектура программного продукта для автоматизации продаж автомобилей в автосалоне.</p> <p>Основные результаты работы:</p> <ol style="list-style-type: none"> 1. Спроектирована реляционная база данных из 7 таблиц, нормализованная до ЗНФ, с корректными связями и ограничениями целостности. Разработана схема БД с указанием всех атрибутов, типов данных и связей. 2. Определены программные компоненты системы (26 компонентов) с использованием трехслойной архитектуры: <ul style="list-style-type: none"> • Слой представления (JSP-формы) • Слой бизнес-логики (контроллеры и сервисы) • Слой доступа к данным (DAO-компоненты) 3. Построена диаграмма компонентов UML, отражающая логическую структуру системы и взаимосвязи между компонентами различных слоев. 4. Описан комплекс технических средств (8 узлов), включая рабочие места пользователей, серверы, периферийное оборудование и сетевую инфраструктуру. 5. Построена диаграмма развертывания UML, показывающая физическое размещение программных компонентов на аппаратных узлах и протоколы взаимодействия между ними. <p>Разработанная архитектура обеспечивает:</p> <ul style="list-style-type: none"> • Масштабируемость за счет трехслойной структуры • Надежность благодаря системе резервного копирования • Безопасность через разделение ролей и использование HTTPS • Производительность за счет выделенных серверов для приложения и БД • Сопровождаемость благодаря модульной структуре <p>Архитектура полностью соответствует функциональным и нефункциональным требованиям, сформулированным в техническом задании, и готова для реализации на следующих этапах разработки.</p>