МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение   
высшего образования «Самарский национальный исследовательский университет   
имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра технической кибернетики

**ОТЧЕТ ПО ПРАКТИКЕ**

Вид практики: *производственная*

Тип практики: *технологическая (научно-технологическая)*

по программе бакалавриата по направлению подготовки   
01.03.02 Прикладная математика и информатика,

профиль «Искусственный интеллект и компьютерные науки»

Сроки прохождения практики: с *14.06.2025* г. по *02.07.2025* г.

|  |  |  |
| --- | --- | --- |
| Обучающийся группы № 6303-010302D | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | М.Д. Шкуркина |
| Руководитель практики  от университета доцент кафедры  технической кибернетики, к.т.н.,  доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | Л.В. Логанова |
| Руководитель практики  от ООО «МедиаСофт Мобайл», администратор | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | М.В. Суханкин |

Дата сдачи 01.07.2025г.

Дата защиты 02.07.2025г.

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2025

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования «Самарский национальный исследовательский университет

имени академика С.П. Королева»

(Самарский университет)

Институт информатики и кибернетики

Кафедра технической кибернетики

Задание по практике для выполнения работ,

связанных с будущей профессиональной деятельностью

обучающемуся Шкуркиной М.Д. группы № 6303-010302D.

Тема задания: *Разработка проекта клиент-серверного мобильного приложения «Онлайн-магазин цветов».*

Направлен на практику приказом по университету от 306-ПР от 11.06.25 в   
ООО «МедиаСофт Мобайл» в соответствии с договором о направлении на практику от 23.05.2023 г. № 148д.

|  |  |  |
| --- | --- | --- |
| Планируемые результаты освоения образовательной программы (*компетенции / индикаторы*) | Выполнение работ, связанных с будущей профессиональной деятельностью | Результаты практики  *(планируемые результаты обучения при прохождении практики)* |
| ПК-1. Способен осуществлять концептуальное моделирование проблемной области и проводить формализацию представления знаний в системах искусственного интеллекта /  ПК-1.1. Демонстрирует способность понимать, совершенствовать и применять современный инструментарий в рамках использования проектной методологии в профессиональной деятельности.  ПК-1.4. Осуществляет информационный поиск и определяет пути решения профессиональных задач в цифровой форме; cпособен использовать цифровые методы в описании и решении социально-гуманитарных задач.  ПК-2. Способен использовать инструментальные средства для решения задач машинного обучения /  ПК-2.1. Демонстрирует способность понимать, совершенствовать и применять современный инструментарий в ходе исследований в рамках профессиональной деятельности.  ПК-2.2. Осуществляет оценку и выбор инструментальных средств для решения поставленной задачи.  ПК-3. Способен осуществлять сбор и подготовку данных для систем искусственного интеллекта /  ПК-3.1. Осуществляет поиск данных в открытых источниках, специализированных библиотеках и репозиториях. | 1. Разработка и описание основной бизнес-логики мобильного приложения, предназначенного для конечных пользователей. Визуализация бизнес-процессов с использованием любой удобной и формально корректной нотации.  2. Составление списка основных пользовательских функций приложения, расположенных **в порядке убывания важности** — от наиболее критичных и часто используемых до второстепенных.  3. Описание того, как осуществляется передача, обновление и удаление данных между клиентской частью (мобильное приложение) и сервером, с использованием следующие средства: UML-диаграммы взаимодействия, описание REST API, ER-диаграмму сущностей.  4. Подготовка **визуального прототипа (макет)** одного из ключевых экранов мобильного приложения. Прототип должен наглядно демонстрировать внешний вид и структуру пользовательского интерфейса, а также отражать основные элементы взаимодействия с системой на выбранном экране.  5. Подробное описание функции покупки товара, которую можно было бы использовать в качестве постановки задачи для разработки (помимо текстового описания использовать UML диаграммы, указать используемые API методы, передаваемые и получаемые параметры, описать процесс хранения информации о покупках пользователя).  6. Формулировка основных сложностей разработки такого приложения.  7. Создание двух взаимосвязанных таблиц по выбранной предметной области с последующим составлением основных SQL-запросов для работы с ними.  8. Изучение основ ООП и спецификации Java. Рассмотрение синтаксиса языка.  9. Формулировка результатов по итогам прохождения технологической практики | *Знает:* основы проектной методологии в системах искусственного интеллекта;  *Умеет:* применять современный инструментарий формализации представления знаний;  *Владеет:* пониманием и навыками совершенствования инструментария в своей профессиональной области.  *Знает:* пути решения профессиональных задач в цифровой форме в системах искусственного интеллекта;  *Умеет:* выполнять информационный поиск;  *Владеет:* цифровыми методами описания и решения социально-гуманитарных задач.  *Знает*: основные методы исследования в области машинного обучения;  *Умеет:* применять современный инструментарий машинного обучения;  *Владеет:* пониманием и навыками совершенствования этого инструментария в своей профессиональной деятельности.  *Знает*: основные инструментальные средства решения задач машинного обучения;  *Умеет*: оценивать и выбирать основные инструментальные средства решения задач машинного обучения;  *Владеет*: навыками решения задач машинного обучения.  *Знает:* основные источники данных для систем искусственного интеллекта;  *Умеет:* осуществлять сбор и подготовку данных для систем искусственного интеллекта;  *Владеет:* навыками поиска данных в открытых источниках, специализированных библиотеках и репозиториях. |

Дата выдачи задания 14.06.2025 г.

Срок представления на кафедру отчета по практике 01.07.2025 г.

|  |  |  |
| --- | --- | --- |
| Руководитель практики  от университета доцент кафедры  технической кибернетики, к.т.н.,  доцент | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | Л.В. Логанова |
| Руководитель практики  от ООО «МедиаСофт Мобайл», администратор | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | М.В. Суханкин |
| Задание принял к исполнению  обучающийся группы № 6303-010302D | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(подпись)* | М.Д. Шкуркина |

**РЕФЕРАТ**

**Отчет по практике:** 28 c., 7 рисунков, 3 таблицы, 6 источников.

МОБИЛЬНОЕ КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ, ОНЛАЙН МАГАЗИН, БИЗНЕС-ЛОГИКА, UML, ER-ДИАГРАММА, REST-API, ПРОТОТИП ИНТЕРФЕЙСА, OOП, JAVA

Данная работа посвящена проектированию и моделированию мобильного клиент-серверного приложения «Онлайн-магазин цветов». В ходе проекта был выполнен анализ бизнес-процессов с использованием нотации BPMN, выделены основные функции и разработана схема взаимодействия клиента и сервера с применением UML-диаграмм, ER-модели и описания REST API. Также создан прототип главного экрана приложения и подробно описан процесс оформления покупки. В заключительной части отчёта рассматриваются ключевые технические трудности, приводятся SQL-запросы для работы с базой данных, а также даётся вводная информация по объектно-ориентированному программированию и основам языка Java.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 7](#_Toc202099007)

[1. Описание бизнес-логики мобильного приложения 8](#_Toc202099008)

[2. Основные функции приложения 11](#_Toc202099009)

[3. Процесс синхронизации данных между клиентом и сервером 12](#_Toc202099010)

[3.1 ER диаграмма сущностей 12](#_Toc202099011)

[3.2 API – методы 12](#_Toc202099012)

[3.3 UML-диаграммы последовательностей 13](#_Toc202099013)

[4. Прототип экрана мобильного приложения 16](#_Toc202099014)

[5. Подробное описание функции покупки товара 17](#_Toc202099015)

[5.1 Текстовое описание функции 17](#_Toc202099016)

[5.2 API – методы 17](#_Toc202099017)

[5.3 UML диаграммы 18](#_Toc202099018)

[5.4 Процесс сохранения данных 20](#_Toc202099019)

[6. Сложности разработки 21](#_Toc202099020)

[7. SQL – запросы для таблиц данной предметной области 22](#_Toc202099021)

[8. Введение в Java 24](#_Toc202099022)

[8.1 Задание 1 по Java 24](#_Toc202099023)

[8.2 Задание 2 по Java 24](#_Toc202099024)

[ЗАКЛЮЧЕНИЕ 26](#_Toc202099025)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 27](#_Toc202099026)

ВВЕДЕНИЕ

Проектирование и моделирование мобильных клиент-серверных приложений становится все более актуальным в связи с широким распространением смартфонов и растущими ожиданиями пользователей в отношении удобных и надёжных инструментов для совместного планирования повседневных дел, например. Особенно востребованы решения, обеспечивающие простой и интуитивный интерфейс для выбора и заказа товаров. Для формализации бизнес-логики и описания взаимодействия между компонентами системы применяются стандартизированные нотации UML и BPMN, при этом согласованность и актуальность данных обеспечиваются за счёт REST-API и механизмов синхронизации между клиентом и сервером. В данной работе рассматривается задача разработки архитектуры и основных компонентов приложения «Онлайн-магазин цветов», включая моделирование бизнес-процессов, проектирование API, построение ER-модели и создание прототипа пользовательского интерфейса.

1. Описание бизнес-логики мобильного приложения

Разработка основной бизнес-логики мобильного приложения, предназначенного для конечных пользователей, предполагает создание последовательной, интуитивно понятной и технически реализуемой модели взаимодействия между клиентом и системой. В данном случае приложение предназначено для заказа цветов и сопутствующих услуг. Пользовательский сценарий начинается с открытия приложения и загрузки каталога товаров, в котором клиент может просматривать ассортимент букетов и композиций. Далее реализована возможность оформления разового заказа: клиент добавляет товар в корзину, указывает адрес и дату доставки, после чего осуществляется оплата. Система фиксирует заказ, обновляет его статус и уведомляет администратора. Отдельной функцией предусмотрена подписка — пользователь может настроить регулярную доставку, выбрав параметры, такие как частота, тип цветов и оформление. После подтверждения сервер создаёт подписку и автоматически формирует первый заказ. Пользователю доступна функция отслеживания заказов через раздел «Мои заказы», где он может получать актуальный статус по каждому заказу. Далее представлен рисунок 1, иллюстрирующий бизнес-процессы.

Диаграмма на рисунке 2 описывает процесс взаимодействия администратора с системой управления интернет-магазином. После запуска приложения администратор переходит к управлению товарами, где он заполняет форму нового товара и добавляет его в каталог. Сервер, в свою очередь, обновляет данные каталога и синхронизирует их с базой данных. Далее администратор может перейти к управлению заказами, где он открывает конкретный заказ и вручную изменяет его статус, например, на «в пути» или «доставлен». После этого сервер обновляет статус заказа в базе данных и отправляет уведомление клиенту о текущем состоянии доставки.

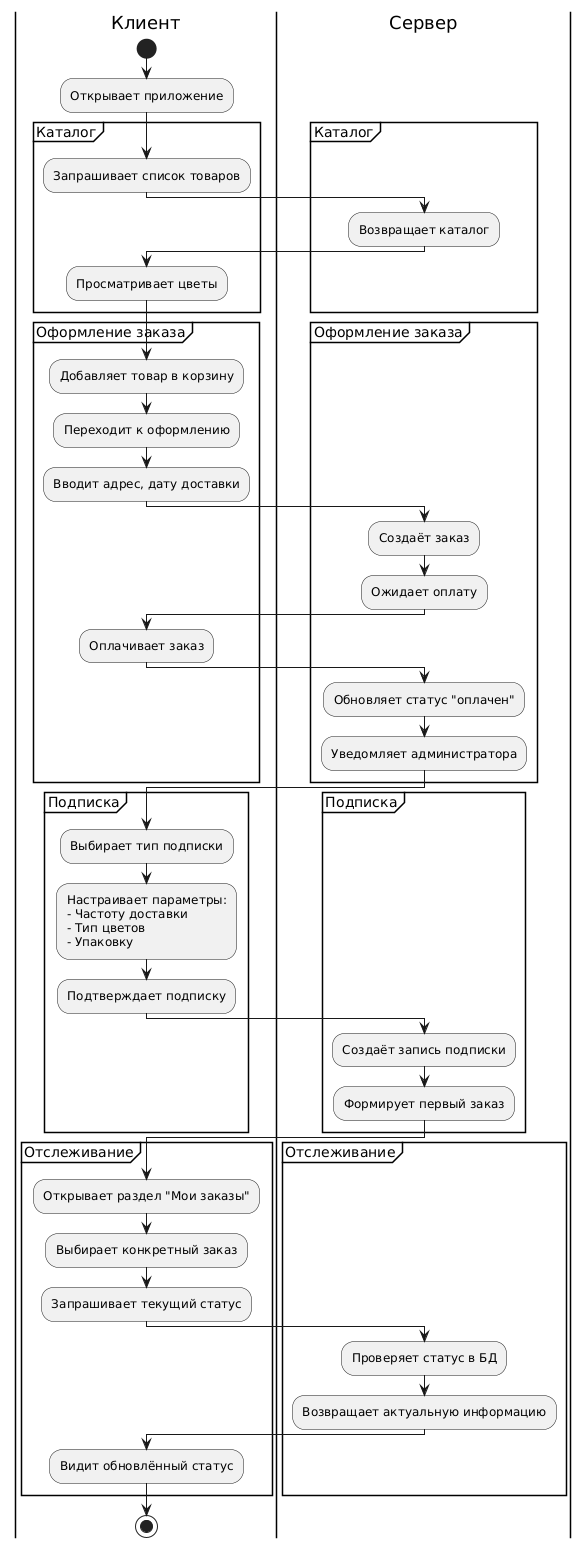


Рисунок 1-Диаграмма, описывающая основные бизнес-процессы со стороны клиента

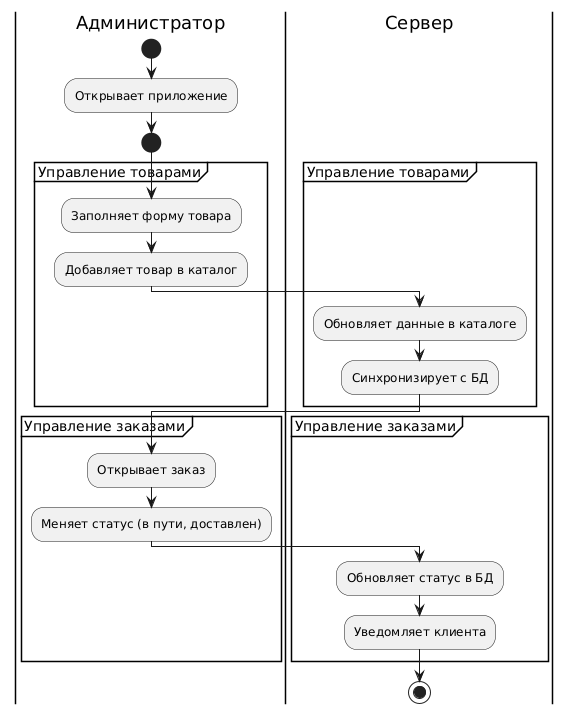


Рисунок 2-Диаграмма, описывающая основные бизнес-процессы со стороны администратора

1. Основные функции приложения

Основные функции приложения (в порядке убывания важности для пользователя):

Просмотр и выбор товаров. У клиента есть возможность просматривать каталог товаров с различными фильтрами (повод, цена, цвет, размер и др.). Каждый товар имеет свою карточку, которая содержит фотографию и описание.

Оформление заказа. После выбора подходящих товаров пользователь может оформить заказ: настроить параметры букета (упаковка, открытка), ввести адрес, дату и время доставки, данные получателя.

Цветочная подписка. Клиент может выбрать частоту доставки цветов, настроить предпочтения, получать напоминания о предстоящих доставках.

Отслеживание заказа. Каждый заказ имеет статус выполнения (принят, обработан, в пути, доставлен). Пользователь будет получать уведомления по этапам доставки

Личный кабинет. Пользователь может обратиться к своей истории заказов, просмотреть избранные товары и т.д.

В этом разделе перечислены основные функции, которые могут быть реализованы в текущей версии приложения. Все дополнительные возможности могут быть рассмотрены и описаны отдельно как предложения для будущих релизов.

1. Процесс синхронизации данных между клиентом и сервером
   1. ER диаграмма сущностей

ER-диаграмма на рисунке 3 подробно описывает предметную область и включает сущности User, Order, Order\_item, Product, Category, Addon, Flower\_subscription и Subscription\_item. User связана с Order и Flower\_subscription по «один-ко-многим» через user\_id. Product имеет связь с объектом Category. Order\_item является промежуточным объектом для реализации связи «многие ко многим» между объектами Order и Product. По мимо этого Order\_item имеет связь с Addon. Аналогично и для реализации связи «многие ко многим» между объектами Flower\_subscription и Product служит объект Subscription\_item.

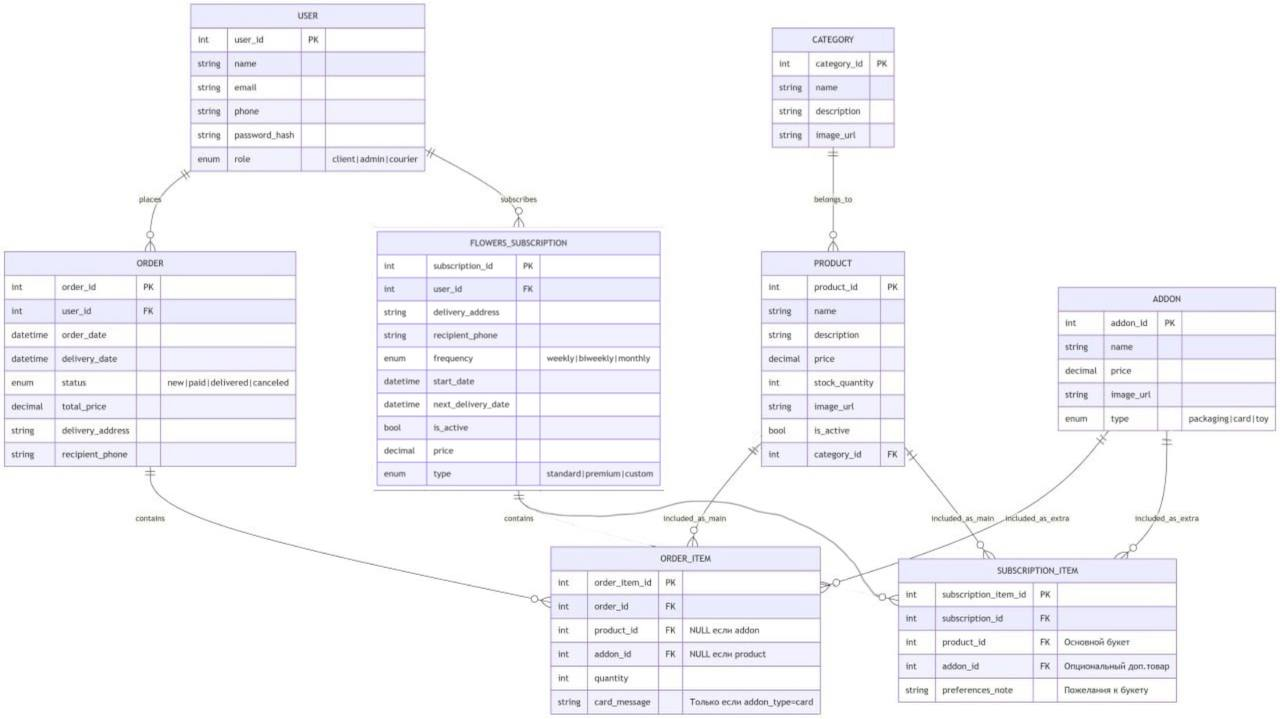
****

Рисунок 3 – ER диаграмма сущностей

* 1. API – методы

Для взаимодействия клиента с сервером реализован REST-API, в котором ключевые операции сгруппированы по областям: аутентификация, товары, корзина, заказы и оплата, цветочная подписка. Пользователь регистрируется и входит в систему через POST-запросы к /auth/register и /auth/login, получая при этом JWT-токен для последующих вызовов. Пользователь может просмотреть свой профиль через GET-запрос /profile.

Администратор может управлять товарами: добавить новый товар через POST-запрос /products, обновить товар через PUT-запрос /products{id} и удалять товары с помощью DELETE-запроса /products{id}.

Обычный пользователь может просматривать список товаров, используя GET-запрос /products, и просмотреть карточку конкретного товара по GET-запросу /products/{id}. Пользователь может добавлять товары в корзину, просматривать корзину и удалять товары из корзины по соответствующим запросам /cart/items, /cart и /cart/items{id}. Для оформления заказа нужно вызвать POST-запрос /order/create, а для оплаты заказа запрос /orders/{id}/pay.

Можно просмотреть свой заказ по запросу /orders/{id}, статус которого меняет администратор PUT-запросом /orders/{id}.

Используя POST-запрос /substructions пользователь может оформить новую подписку, а по такому же GET-запросу можно просмотреть список активных подписок.

* 1. UML-диаграммы последовательностей

Диаграмма последовательности основных процессов представлена на рисунке 4.

Клиент начинает с регистрации в мобильном приложении: вводит свои данные, и после отправки на сервер они проходят проверку и сохраняются. Как только всё успешно, приложение получает доступ через токен.

Когда клиент заходит снова, он проходит авторизацию. Вводит логин и пароль, сервер проверяет и, если всё в порядке, выдаёт доступ. Если данные неверны, авторизация не проходит.

Чтобы выбрать цветы, клиент открывает каталог. Приложение делает запрос на сервер, тот обращается к базе и возвращает список товаров, которые отображаются в приложении.

После выбора клиент добавляет нужные товары в корзину и переходит к оформлению. Он вводит адрес доставки, дату и подтверждает заказ. Когда заказ сформирован, переходит к оплате — и после успешной транзакции сервер меняет статус на «оплачен».

Если клиент хочет получать букеты регулярно, он настраивает подписку: выбирает тип, частоту, предпочтения. Эти данные уходят на сервер, и подписка создаётся — клиент получает подтверждение.

Чтобы узнать, где находится его заказ, клиент заходит в раздел с заказами. Сначала приложение получает список всех заказов, затем при выборе одного — сервер возвращает всю информацию по нему, включая статус.

Администратор со своей стороны может зайти в панель, открыть конкретный заказ и изменить его статус — например, отметить, что он в пути. После обновления в базе система сообщает, что всё прошло успешно.

Также администратор может добавить новый товар. Он заполняет форму, отправляет данные на сервер, и если всё в порядке, товар сохраняется в базе, а админ получает уведомление, что всё создано.

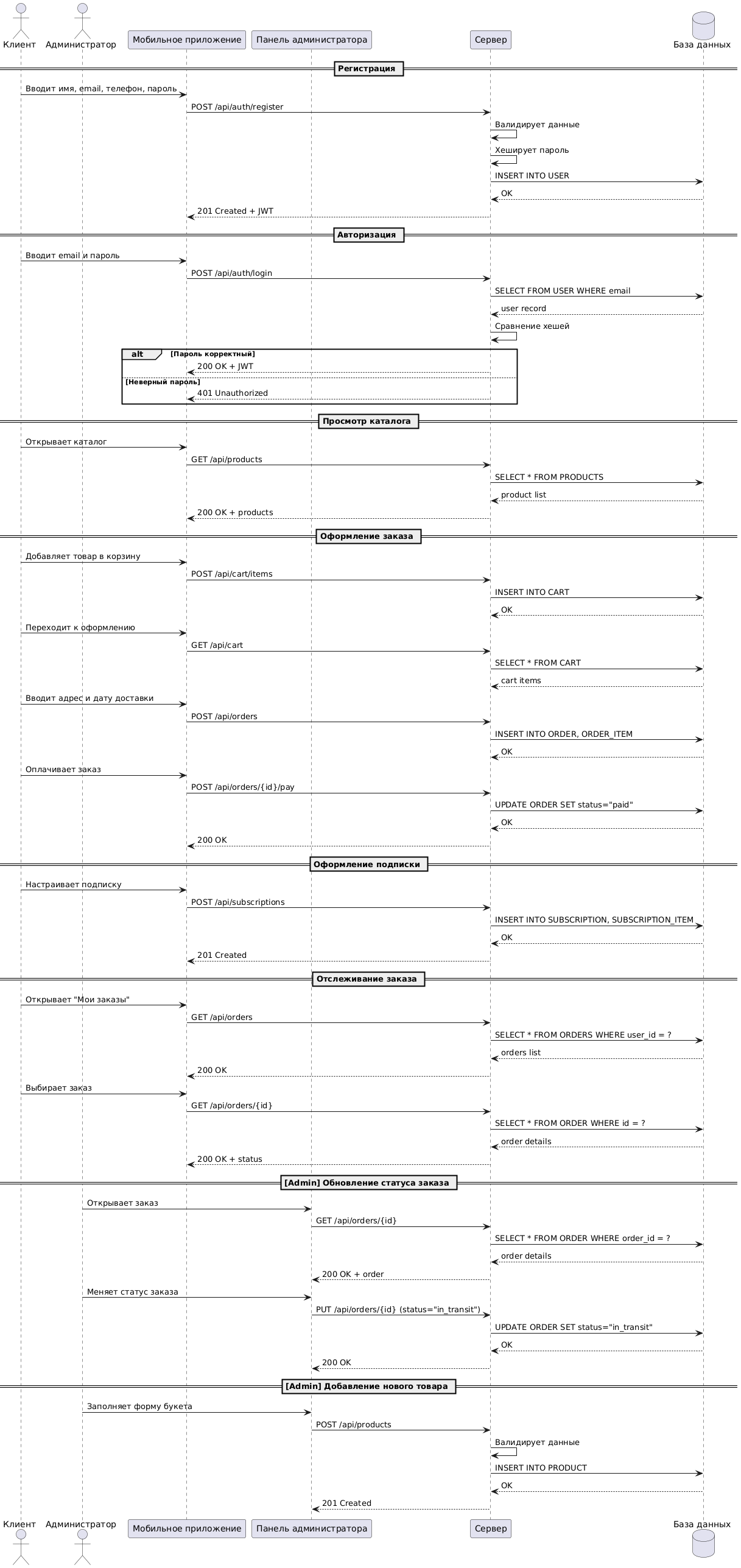


Рисунок 4-Диаграмма последовательности основных процессов

1. Прототип экрана мобильного приложения

На макете (рисунок 5) главного экрана в верхней панели расположено название приложения крупным шрифтом. Ниже находится поисковая строка, которая позволяет искать товары по ключевым словам. Далее располагаются фильтры категорий, по нажатию на которые происходит фильтрация списков товаров ниже. Большую часть занимает сетка товаров в виде двух-колоночной сетки, где каждый товар представлен карточкой с изображением, название, стоимость и кнопкой добавления товара в корзину. Нижняя панель навигации содержит четыре кнопки: главная, избранное, корзина и профиль.

Пользователь видит товары по умолчанию, может отфильтровать по категориям или воспользоваться поиском, может добавить товары в корзину и через нижнюю панель перейти в корзину для оформления заказа.

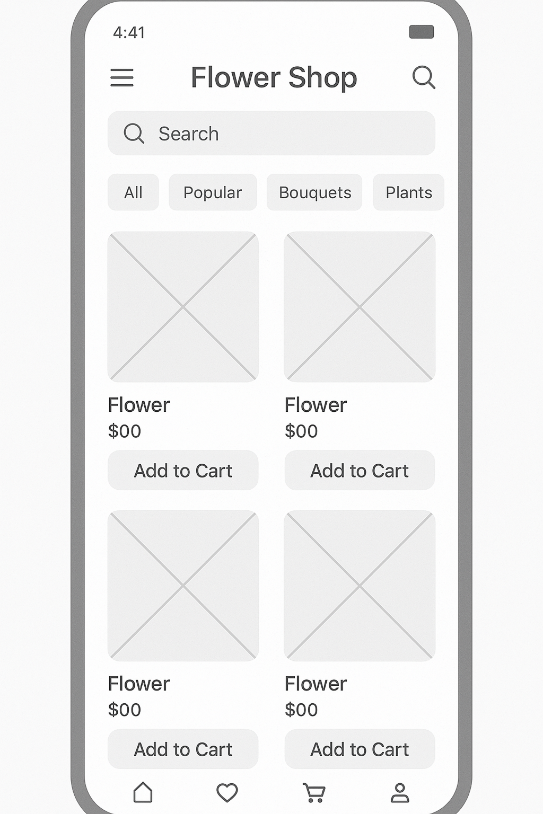
****

Рисунок 5 – Макет главного экрана

1. Подробное описание функции покупки товара

## Текстовое описание функции

Функция покупки товара, в частности букета, позволяет пользователю выбрать один или несколько букетов, добавить их в корзину, указать данные для доставки и оплаты, оформить заказ и получить подтверждение успешной покупки.

Основные этапы:

1. Пользователь просматривает каталог букетов и выбирает нужный.
2. Пользователь добавляет выбранный букет в корзину, при необходимости указывает количество и дополнительные опции (например, открытку, упаковку).
3. Пользователь переходит к оформлению заказа.
4. Вводит данные доставки (адрес, телефон, время доставки и т.д.).
5. Выбирает способ оплаты.
6. Система обрабатывает заказ, сохраняет информацию о покупке и списывает товар со склада.
7. Пользователь получает информацию о текущем статусе заказа.
   1. **API – методы**

В таблице 1 представлены необходимые API-запросы, которые будут использоваться в процессе покупки товара.

Таблица 1- API-запросы для покупки товара

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Метод | URL | Назначение | Тело запроса | Ответ |
| GET | /products | Получить список товаров | ?category=...&price\_min=...&price\_max=...&color=... | Список товаров с id, названием, ценой, изображением и остатком |
| POST | /cart/items | Добавить товар в корзину | {product\_id, quantity, options? } | {success: true, cartItemId: 55} |
| GET | /cart | Получить содержимое корзины |  | {items: [...], total: 2500} |
| POST | /order/create | Оформить заказ | {user\_id, delivery: {address, phone, date, time}} | {order\_id: 789, status: "created"} |
| POST | /orders/{id}/pay | Оплатить заказ | {paymentMethod, cardToken? } | {paymentStatus: "paid", transactionId: "tx\_56789"} |
| GET | /orders/{id} | Получить информацию о заказе |  | {order\_id, status, items, delivery, price} |

* 1. **UML диаграммы**

На рисунке 6 показана диаграмма use-case. Актерами являются пользователь (клиент магазина, оформляющий заказ), администратор, курьерская служба и платежная система. Сценарии: оформить заказ, оплатить заказ, отследить заказ, отменить заказ, управление заказами и обновление статуса доставки. Связь включение показывает, что оформление заказа включает в себя обязательные шаги выбора товара и ввод адреса, связь расширение показывает, что отмена заказа может запустить процесс возврата товара на склад.

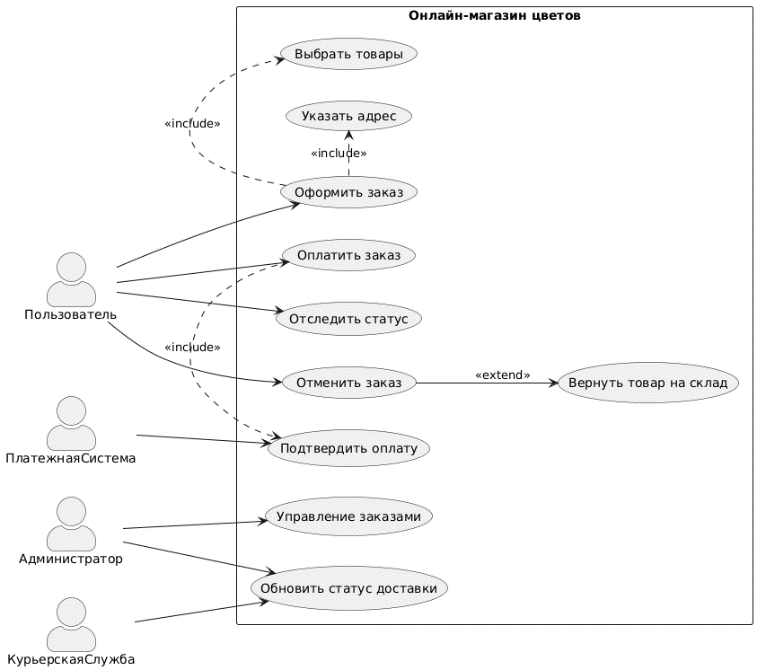


Рисунок 6 – Диаграмма use-case для функции покупки товара

Процесс оформления заказа в мобильном приложении «Flower Shop» начинается с того, что пользователь нажимает кнопку «Оформить заказ», после чего приложение отправляет на сервер запрос на получение содержимого корзины. Получив список товаров и общую сумму, пользователь вводит данные доставки, которые приложение передаёт на сервер через POST-запрос. Сервер проверяет наличие товаров в базе данных, после чего инициирует создание платежа через внешний платежный шлюз. После подтверждения оплаты сервер сохраняет информацию о заказе и его составе в базу данных, формируя записи в таблицах заказов и товаров заказа. В завершение сервер отправляет приложению подтверждение с номером заказа и датой доставки, а приложение уведомляет пользователя об успешном оформлении заказа. Диаграмма последовательностей представлена на рисунке 7.

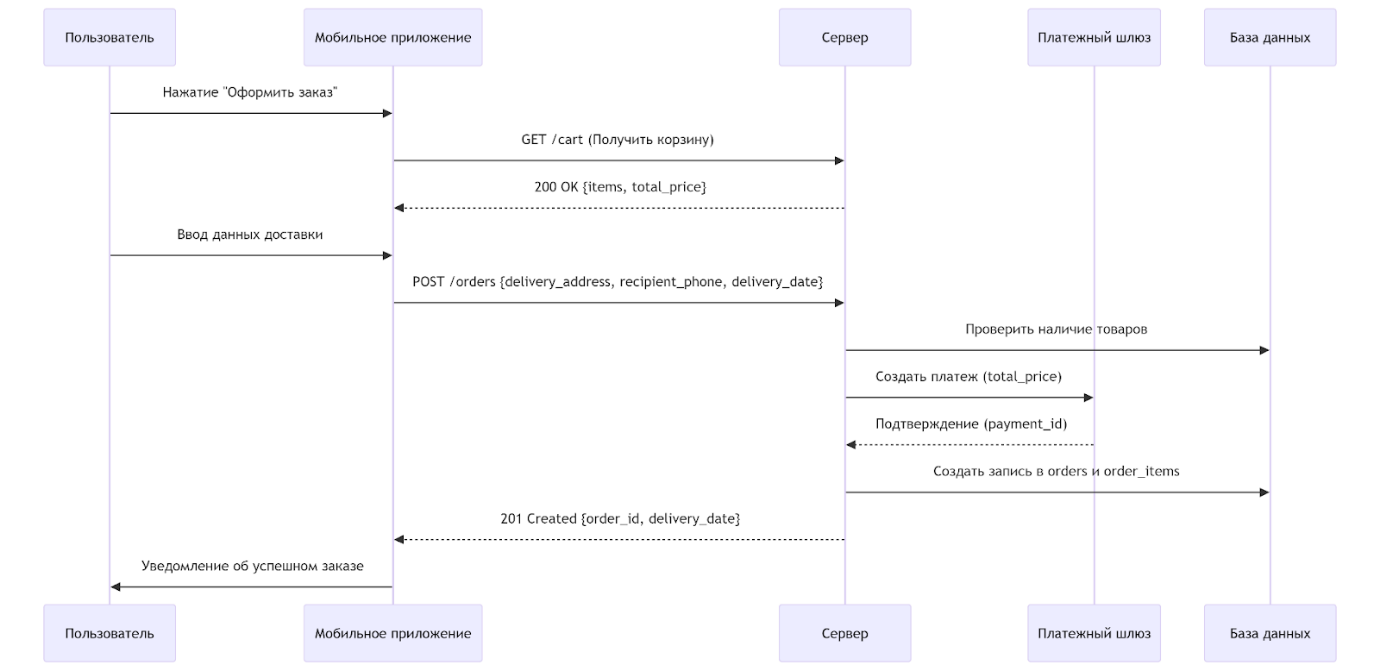


Рисунок 7 – Диаграмма последовательностей для покупки товара

* 1. **Процесс сохранения данных**

1. При создании заказа:

* В orders записывается основная информация (пользователь, сумма, статус);
* В order-items — состав заказа (товары, количество, цена на момент покупки).

1. После оплаты:

* Обновляется status заказа;
* Уменьшается stock\_quantity товаров.

Таблицы подробнее представлены на ER диаграмме на рисунке 3.

1. Сложности разработки

При разработке такого онлайн-магазина с технической точки зрения возникают следующие ключевые сложности.

Во-первых, необходимо реализовать динамический учет наличия товаров: ассортимент часто меняется, а цветы — скоропортящийся продукт, поэтому данные о наличии и сроках годности должны обновляться в реальном времени.

Во-вторых, важна оптимизация и масштабируемость серверной части, поскольку в пиковые периоды, например, в праздничные дни, нагрузка на систему резко возрастает. Для обеспечения комфортного пользовательского опыта необходимо создать отзывчивый и интуитивно понятный интерфейс с оптимизированной загрузкой высококачественных изображений, который будет одинаково хорошо работать как на мобильных устройствах, так и на десктопах.

Кроме того, требуется разработка сложной логики оформления заказов и оплаты, учитывающей множество вариантов доставки, способов оплаты с надежной интеграцией платежных шлюзов и обработкой ошибок. Безопасность данных пользователей и транзакций — еще одна важная задача: система должна защищать от уязвимостей, обеспечивать шифрование платежной информации и корректную аутентификацию.

Хранение и обработка больших объемов данных о пользователях и заказах требуют продуманной архитектуры базы данных с возможностью быстрого поиска, фильтрации и регулярного резервного копирования.

Наконец, необходима многоуровневая административная панель для управления каталогом, заказами, доставкой и аналитикой, а также тщательное тестирование и отладка приложения на различных устройствах и браузерах для предотвращения багов и обеспечения стабильной работы. Все эти технические задачи требуют комплексного подхода и грамотного выбора технологий и архитектурных решений.

1. SQL – запросы для таблиц данной предметной области

Ниже представлены таблица 1 - таблица Users полями id, name, email и phone и таблица 2 – таблица Orders с полями user\_id, order\_date, delivery\_date, price, status, delivery\_address, recipient\_phone.

Таблица 1 – Таблица Users

|  |  |  |  |
| --- | --- | --- | --- |
| id | name | email | phone |
| 1 | Иванова Анна | "anna@example.com" | "+79161234567" |
| 2 | Петров Сергей | "sergey@example.com" | "+79262345678" |
| 3 | Ефимов Андрей | "andre@example.com" | "+79373456789" |
| 4 | Ермакова Мария | "maria@example.com" | "+79484567890" |
| 5 | Смирнова Елена | "elena@example.com" | "+79595678901" |

Таблица 2-Таблица Orders

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id | user\_id | order\_date | delivery\_date | price | status | delivery\_address | recipient\_phone |
| 1 | 1 | "2023-05-10" | "2023-05-12" | 3500.00 | "доставлен" | "ул. Цветочная, 15, кв. 23" | "+79161234567" |
| 2 | 2 | "2023-05-11" | [null] | 4200.00 | "доставлен" | "пр. Розы Люксембург, 42" | "+79262345678" |
| 3 | 3 | "2023-05-12" | "2023-05-13" | 2800.00 | "в обработке" | "ул. Садовая, 7, кв. 12" | "+79373456789" |
| 4 | 1 | "2023-05-13" | [null] | 5100.00 | "новый" | "ул. Лепестковая, 33" | "+79161234567" |
| 5 | 4 | "2023-05-14" | "2023-05-14" | 3900.00 | "отменен" | "ул. Букетная, 18" | "+79484567890" |
| 6 | 5 | "2023-05-15" | "2023-05-15" | 4500.00 | "доставлен" | "пр. Тюльпанов, 5, кв. 9" | "+79595678901" |
| 7 | 3 | "2023-05-16" | "2023-05-17" | 3200.00 | "в обработке" | "ул. Ромашковая, 21" | "+79373456789" |
| 8 | 2 | "2023-05-17" | "2023-05-19" | 4800.00 | "новый" | "ул. Лилейная, 14" | "+79262345678" |

1. Вывести количество заказов пользователя с именем “Петров Сергей”

SELECT

COUNT(o.id) AS total\_orders

FROM orders o

JOIN users u ON o.user\_id = u.id

WHERE u.name = 'Петров Сергей';

1. Найти все заказы пользователя с email = '[anna@example.com](mailto:anna@example.com)'.

SELECT

o.id AS order\_id,

     o.order\_date,

     o.delivery\_date,

     o.total\_price,

     o.status,

     o.delivery\_address

FROM orders o

JOIN users u ON o.user\_id = u.id

WHERE u.email = 'anna@example.com';

1. Вывести количество заказов по статусам.

SELECT

status,

COUNT (\*) AS order\_count

FROM orders

GROUP BY status;

1. Вывести клиентов, сделавших более одного заказа.

SELECT

u.name,

u.email,

COUNT(o.id) AS total\_orders

FROM users u

JOIN orders o ON o.user\_id = u.id

GROUP BY u.id, u.name, u.email

HAVING COUNT(o.id) >1;

1. Вывести суммарную выручку по заказам за конкретную дату.

SELECT

SUM (price) AS revenue\_for\_date

FROM orders

WHERE order\_date = '2023-05-11'

AND status != 'отменен';

1. Введение в Java
   1. **Задание 1 по Java**

В рамках задания был разработан класс BankAccount, представляющий банковский счёт. Класс содержит поля: имя владельца (строка), баланс (целое число), дата открытия (объект LocalDateTime), флаг, указывающий, заблокирован ли счёт (boolean), а также номер счёта в формате восьмизначного числа, генерируемого случайным образом при создании объекта. Реализован конструктор, принимающий только имя владельца и инициализирующий остальные поля начальными значениями. Добавлены методы deposit(), withdraw() и transfer(), возвращающие значение типа boolean, отражающее успешность операции. Методы учитывают ограничения: нельзя снимать или переводить сумму, превышающую текущий баланс. Также переопределён метод toString() для вывода информации о счёте в удобном формате, реализованы методы equals() и hashCode() для корректного сравнения и использования объектов в коллекциях. Реализованный класс был протестирован, и все методы корректно отрабатывают предусмотренные сценарии.

## 8.2 Задание 2 по Java

В ходе выполнения этого задания я вспомнила основнные моменты работы с массивами, подробнее познакомилась с коллекциями, познакомилась со Stream API и вспомнила как и зачем переопределяются методы equals, hashCode() и compareTo() в Java.

Были выполнены задачи на создание и обработку массивов с годами выпуска автомобилей, работа со списками и множествами моделей машин, фильтрация, сортировка и группировка данных с использованием Stream API.

Основное внимание было уделено разработке класса Car, содержащего поля VIN, модель, производитель, год выпуска, пробег, цену и тип (перечисление enum — например, SEDAN, SUV, ELECTRIC и т.д.). В классе переопределены методы equals() и hashCode() таким образом, чтобы машины считались одинаковыми только при совпадении VIN, а также реализован интерфейс Comparable<Car>, что позволило сортировать автомобили по году выпуска от новых к старым. На основе этого класса был реализован основной класс системы — CarDealership, представляющий собой автоцентр с возможностью добавления автомобилей (с проверкой уникальности по VIN), поиска машин по производителю, вычисления средней цены автомобилей заданного типа, сортировки машин по году выпуска и генерации статистики — подсчёта количества машин каждого типа, а также поиска самой новой и самой старой машины. Для взаимодействия с пользователем было реализовано консольное меню на основе класса Scanner, предоставляющее доступ ко всем функциям автоцентра в интерактивном режиме.

# ЗАКЛЮЧЕНИЕ

В рамках выполнения практики было спроектировано мобильное клиент-серверное приложение для онлайн-магазина цветов с акцентом на удобство пользовательского опыта, проработанную бизнес-логику и ключевые функции, необходимые для полноценной работы сервиса. С использованием нотаций BPMN и UML были смоделированы процессы оформления заказа, добавления товаров в корзину, отслеживания статуса и оплаты, что позволило описать логику взаимодействия между пользователем, сервером, базой данных.

Основные процессы были смоделированы в виде последовательных REST API-запросов, каждый из которых документирован с указанием параметров и возвращаемых данных. Диаграмма активности описывает жизненный цикл заказа, от формирования корзины до подтверждения оплаты, а ER-диаграмма отражает структуру ключевых сущностей.

Так же был создан макет главного экрана приложения, демонстрирующий понятную навигацию, фильтры по категориям, карточки товаров и элементы управления корзиной.

Были определены основные технические сложности создания такого приложения: необходимость динамического учета товара, оптимизация и масштабируемость серверной части, разработка сложной логики оформления заказов и оплаты, разработка приложения под клиента и под администратора.

Были созданы таблицы Users и Orders, по которым были написаны SQL-запросы, которые дополнили практическую часть работы.

В завершении практики были изучены основы объектно-ориентированного подхода и базовые принципы языка Java, на котором может быть реализована серверная часть.

Полученный результат представляет собой полный набор материалов, необходимых для старта разработки надёжного и масштабируемого приложения, адаптированного под потребности клиентов.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. **Pautasso C., Ivanchikj A., Schreier S.** Modeling RESTful Conversations with Extended BPMN Choreography Diagrams // Service-Oriented and Cloud Computing. — 2015. — С. 77-94. — DOI: 10.1007/978-3-319-23727-5\_7. — URL: https://www.researchgate.net/publication/281742830\_Modeling\_RESTful\_Conversations\_with\_Extended\_BPMN\_Choreography\_Diagrams(дата обращения: 19.09.2025).
2. **Евдокимова С. А., Новикова Т. П.** Использование языка моделирования UML для проектирования информационных систем // Междисциплинарный научный форум: материалы I Всероссийской научной конференции — Ульяновск: УлГТУ, 2023. — С. 78-83. — URL: https://elibrary.ru/item.asp?id=80664714 (дата обращения: 25.07.2024).
3. **Архипова К. С., Мнацаканян О. Л.** Средства языка UML как точка перехода между этапами проектирования и реализации программного обеспечения // Наука. Производство. Образование - 2024: материалы IV Всероссийской научно-технической конференции (г. Москва, 19 ноября 2024 г.). — М.: РГСУ, 2025. — С. 37-41. — URL: https://elibrary.ru/item.asp?id=80627803(дата обращения: 27.06.2025).
4. Емельянов, А. Н. Разработка интернет-магазина цветов (на примере салона цветов «Лилия») / А. Н. Емельянов, Д. П. Денисова // Наука и молодежь: Материалы XVIII Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых. – Барнаул: Алтайский государственный технический университет им. И.И. Ползунова, 2021. – С. 49-51. – URL: https://elibrary.ru/item.asp?id=46680309 (дата обращения: 27.06.2025).
5. Борисенков, Д. С. Предметно-ориентированный язык описания диаграмм пакетов и диаграмм алгебраических типов данных расширенной нотации UML / Д. С. Борисенков // Перспективы развития информационных технологий. – 2014. – № 19. – С. 7-11. – URL: https://elibrary.ru/item.asp?id=21578990 (дата обращения: 20.06.2025).
6. Чеснаков, М. М. Необходимость UML-диаграмм при описании бизнес-процессов / М. М. Чеснаков // Научный Лидер. – 2024. – № 22(172). – С. 89-92. – URL: https://elibrary.ru/item.asp?id=67356097(дата обращения: 20.06.2025).