МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук
Кафедра программирования и информационных технологий

Приложение для расчета себестоимости и конечной стоимости изделий домашнего кондитера

Курсовой проект

09.03.04 Программная инженерия

Информационные системы и сетевые технологии

Зав. Кафедрой _____С.Д. Махортов, д-р физ.-мат. наук, профессор

Обучающийся _____А.С. Елисеев, 3 курс, очное

Обучающийся _____А.Э. Галимов, 3 курс, очное

Обучающийся _____Д.Ю. Скарга, 3 курс, очное

Руководитель _____В.С. Тарасов, ст. преподаватель

Руководитель _____А.В. Москаленко, магистр __. __.20__

Воронеж 2024

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ

Таблица 1 - Определения, сокращения, обозначения

Термин	Определение термина
Apache Maven	Инструмент для автоматической сборки проектов на Java и других языках программирования. Он помогает разработчикам правильно подключить библиотеки и фреймворки, управлять их версиями, выстроить структуру проекта и составить к нему документацию
API	Программный интерфейс приложения. Описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой
Java	Строго типизированный объектно-ориентированный язык программирования общего назначения
Kotlin	Статически типизированный, объектно-ориентированный язык программирования, работающий поверх Java Virtual Machine
Kanban доска	Цифровой инструмент управления проектами, который помогает наглядно представить задачи, ограничить объем незавершенной работы и добиться максимальной производительности
PostgreSQL	Свободно распространяемая объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом
Rest	Архитектурный стиль взаимодействия компонентов распределенного приложения в сети

Термин	Определение термина
Spring	Фреймворк с открытым исходным кодом для языка программирования Java
Автоматизированная система	Представляет собой организационно-техническую систему, обеспечивающую выработку решений на основе автоматизации информационных процессов в различных сферах деятельности
Авторизация	Предоставление определенному лицу прав на выполнение определенных действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий
Авторизованный пользователь	Пользователь, прошедший процесс авторизации
Архитектура приложения	Способ организации и структурирования программного кода, который обеспечивает работу приложения
База Данных (БД)	Упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе
Боковое меню	Меню, которое представляет собой панель, которая находится (или открывается, при помощи, каких-либо кнопок или жестов) снизу, слева или справа от области основного контента приложения, содержащая вертикальную, независимую от основного контента приложения прокрутку, и служит основным инструментом навигации в приложении

Термин	Определение термина
Издержки производства	Затраты, связанные с производством товаров
Интерфейс	Среда, которую пользователь видит и с которой взаимодействует при использовании приложения. Он включает в себя все элементы управления, кнопки, меню, диалоговые окна и другие элементы, которые помогают пользователю управлять приложением и выполнять с его помощью различные задачи
Клиент-серверное взаимодействие	Способ обмена информацией между двумя устройствами, где одна сторона (клиент) запрашивает данные у серверной части. Последняя формирует ответ, направляя его в обмен
Клиентская часть	Часть программного обеспечения, которая взаимодействует непосредственно с пользователем через интерфейс на стороне пользователя
Конечная стоимость	Деньги, которые продавец планирует получить за товар
Коэффициент наценки	Коэффициент, на который умножается себестоимость товара или услуги
Неавторизованный пользователь	Пользователь, не прошедший процесс авторизации
Оперативный отчет	Форма внутренней отчетности, которая характеризует отдельные фрагменты деятельности предприятия или частного лица и используется для нужд текущего управления и контроля

Термин	Определение термина
Пользователь расширенной версии	Авторизованный пользователь, который использует расширенную версию
Расширенная версия приложения	Программное обеспечение, которое представляет собой улучшенную и дополненную версию приложения
Реляционная база данных	Набор данных с предопределенными связями между ними. Эти данные организованы в виде набора таблиц, состоящих из столбцов и строк
Реляционная СУБД	Система управления базами данных. Комплекс программ, позволяющих создать реляционную базу данных (БД) и управлять данными
Себестоимость	Оценка стоимости текущих затрат предприятия на производство и реализацию продукции
Серверная часть	Программно-аппаратная часть сервиса, которая хранится на сервере, обрабатывает полученные данные и отправляет ответ обратно
Фидбек	Ответная реакция, отклик на какое-либо действие, событие, информацию
Фреймворк	Заготовка, готовая модель в программировании для быстрой разработки, на основе которой можно дописать собственный код

Введение

В условиях активного развития рынка услуг домашнего кондитера и увеличения числа людей, занимающихся изготовлением кондитерских изделий для продажи или личного потребления, важность эффективного учета себестоимости и конечной стоимости продукции становится все более актуальной.

Данный курсовой проект посвящен разработке сервиса для расчета себестоимости и конечной стоимости изделий домашнего кондитера с возможностью учета затрат на продукты, электроэнергию, воду и процента аккумулирования. Целью создания приложения является предоставление возможности точного и удобного расчета себестоимости кондитерских изделий, учета всех издержек, связанных с их производством, а также организация системы для отслеживания статуса заказов и составление отчетов о затратах за определенный период времени.

Приложение предназначено для управления состоянием заказов, добавления ингредиентов, используемых в рецептах, создания шаблонов готовых изделий из добавленных ингредиентов, а также для расчета стоимости готовых изделий с учетом издержек на их производство. Оно позволяет объединять пользователей в группы для ведения совместного учета затрат и предоставляет возможность получения оперативных отчетов.

В ходе работы будет проведена разработка и тестирование предложенного сервиса, демонстрирующего его работоспособность и эффективность.

Разработка данного приложения имеет большое практическое значение и может быть использована домашними кондитерами для оптимизации учета затрат и увеличения прибыльности их бизнеса.

1 Постановка задачи

Целью данного курсового проекта является разработка приложения для упрощения расчета себестоимости кондитерских изделий. Приложение должно позволять учитывать все издержки, связанные с производством изделий.

1.1 Требования к функциональной части

Неавторизованным пользователям должна быть предоставлена возможность:

— регистрации;

— просмотра информации о приложении.

Авторизованным пользователям должна быть предоставлена возможность:

— управления заказами клиентов, включая создание новых заказов и отслеживание их статуса;

— управления ингредиентами, необходимыми для рецептов, включая добавление, редактирование и удаление ингредиентов;

— создания шаблонов изделий на основе имеющихся продуктов;

— расчета стоимости готовых изделий с учетом издержек на их производство;

— генерации отчетов по доходам или заказам;

— присоединения к группе;

— создания группы;

— просмотра общих оперативных отчетов по всей группе.

1.2 Технические требования

Данные требования необходимы для обеспечения стабильной работы приложения на целевых устройствах, а также для удовлетворения потребностей конечных пользователей в удобстве и эффективности использования приложения.

1.2.1 Платформа и ОС

Приложение должно корректно функционировать на устройствах с операционной системой Android версии 10 и выше.

1.2.2 Устройства

Мобильное приложение предназначено для мобильных устройств с диагональю экрана не менее 5 дюймов и не более 7 дюймов.

1.3 Требования к интерфейсу

Приложение должно быть выполнено в одной цветовой палитре с использованием ограниченного набора шрифтов. Экраны приложения должны быть оформлены в едином стиле.

2 Анализ предметной области

Данный раздел содержит анализ предметной области.

2.1 Анализ существующих решений

Анализ существующих решений будет проводиться на основе сервисов, приведенных в таблице 2.

Таблица 2 - Примеры существующих решений

Название	Ссылка
Расчет себестоимости десерта	https://naira-arina.ru/calculate
Make Cake	https://webinar.make-cake.net/calculate_cake
Candy School	https://candy-school.ru/candy-calculator
PastryPro	https://pastrypro.ru/
Bake Master	https://bakemaster.ru/

2.1.1 Расчет себестоимости десерта

Данный сервис представляет из себя сайт для расчета себестоимости изготовления десерта. На сайте можно по введенным ингредиентам рассчитать себестоимость готового продукта. Функционал сервиса представлен на рисунке 1.

Преимущества:

— можно быстро и без регистрации посчитать себестоимость десерта;

— простота использования.

Недостатки:

— данные для расчета не сохраняются (при перезаходе на страницу нужно будет вводить все по новой);

— учитываются только затраты на ингредиенты. Невозможно указать издержки на производство (затраты на воду, электричество);

— устаревший дизайн.

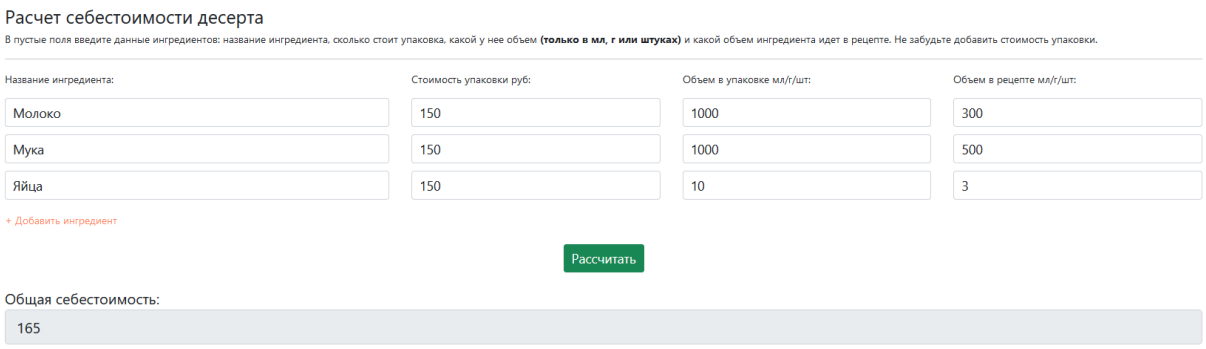

Рисунок 1 - Функционал сайта для расчета себестоимости десерта

2.1.2 Make Cake

Данный сервис представляет из себя сайт для расчета себестоимости изготовления десерта. Для расчета себестоимости необходимо заполнить ингредиенты (граммовки используемые в рецепте, вес упаковки используемого ингредиента, стоимость упаковки ингредиента). Помимо этого, для точно расчета себестоимости 1 кг. торта необходимо указать вес готового десерта, но, если вам нужен предварительный расчет, можно обойтись и без него. Функционал сервиса представлен на рисунке 2.

Преимущества:

— возможность указать вес готового изделия, что позволяет рассчитать себестоимость готового изделия на 1 кг продукта;

— простота расчета;

— современный дизайн.

Недостатки:

— данные расчета не сохраняются;

— невозможно указать издержки для изготовления изделия;

— нельзя посчитать конечную стоимость продукта с учетом наценки.

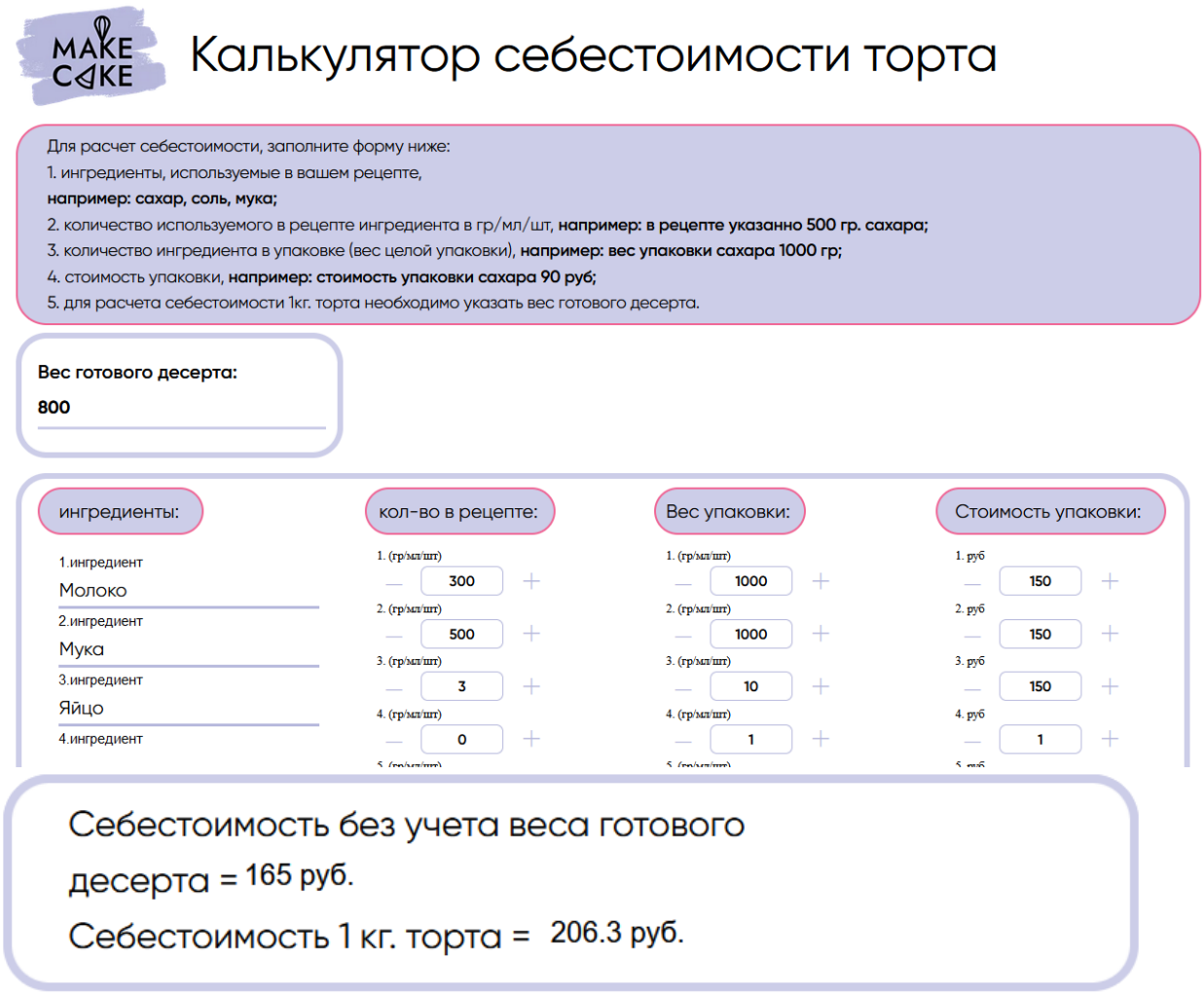

Рисунок 2 - Функционал сайта Make Cake

2.1.3 Candy School

Данный сервис представляет из себя сайт для расчета себестоимости изготовления десертов. На сайте также можно сочетать начинки и крема с самыми популярными бисквитами. Для расчета себестоимости нужно выбрать формулу для расчета. Затем указать граммовку и цены. После внесения исходных данных пользователь получает готовый расчет. Функционал представлен на рисунке 3.

Преимущества:

— возможность учета дополнительных расходов;

— возможность расчета себестоимости сразу нескольких изделий;

— шесть формул для расчета себестоимости;

— современный интерфейс.

Недостатки:

— полностью платный доступ к сервису. Доступ к основным функциям предоставляется только после оплаты на 1, 3, 6 или 12 месяцев;

— невозможно сохранить данные расчета, а также нельзя отдельно хранить ингредиенты.

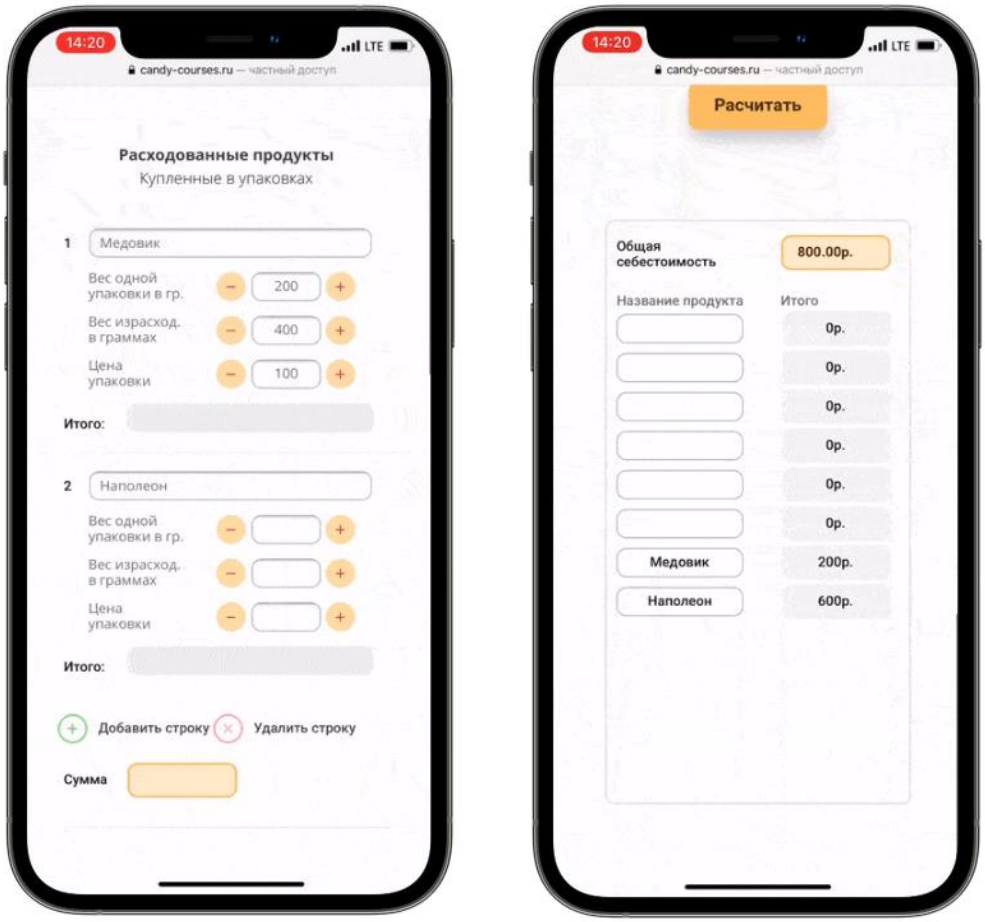

Рисунок 3 - Функционал Candy School

2.1.4 PastryPro

Данный сервис представляет из себя кроссплатформенное приложение для кондитеров. Сервис помогает эффективно вести учет заказов, а также рассчитывать точную себестоимость ингредиентов и рецептов. Функционал приложения представлен на рисунке 4.

Преимущества:

— кроссплатформенность (работает на компьютере, планшете и смартфоне);
— возможность сохранения ранее добавленных ингредиентов;
— возможность сохранения шаблона изделий;
— возможность создания заказов с указанием его статуса, даты и времени, а также срока выполнения заказа;
— возможность ведения учета доходов и расходов.

Недостатки:

— полностью платный доступ к сервису. Доступ к основным функциям предоставляется только после оплаты на 1 месяц или на 1 год;
— невозможность учета издержек (расходы на воду, электричество, газ и т.д.);
— перегруженность функциями.

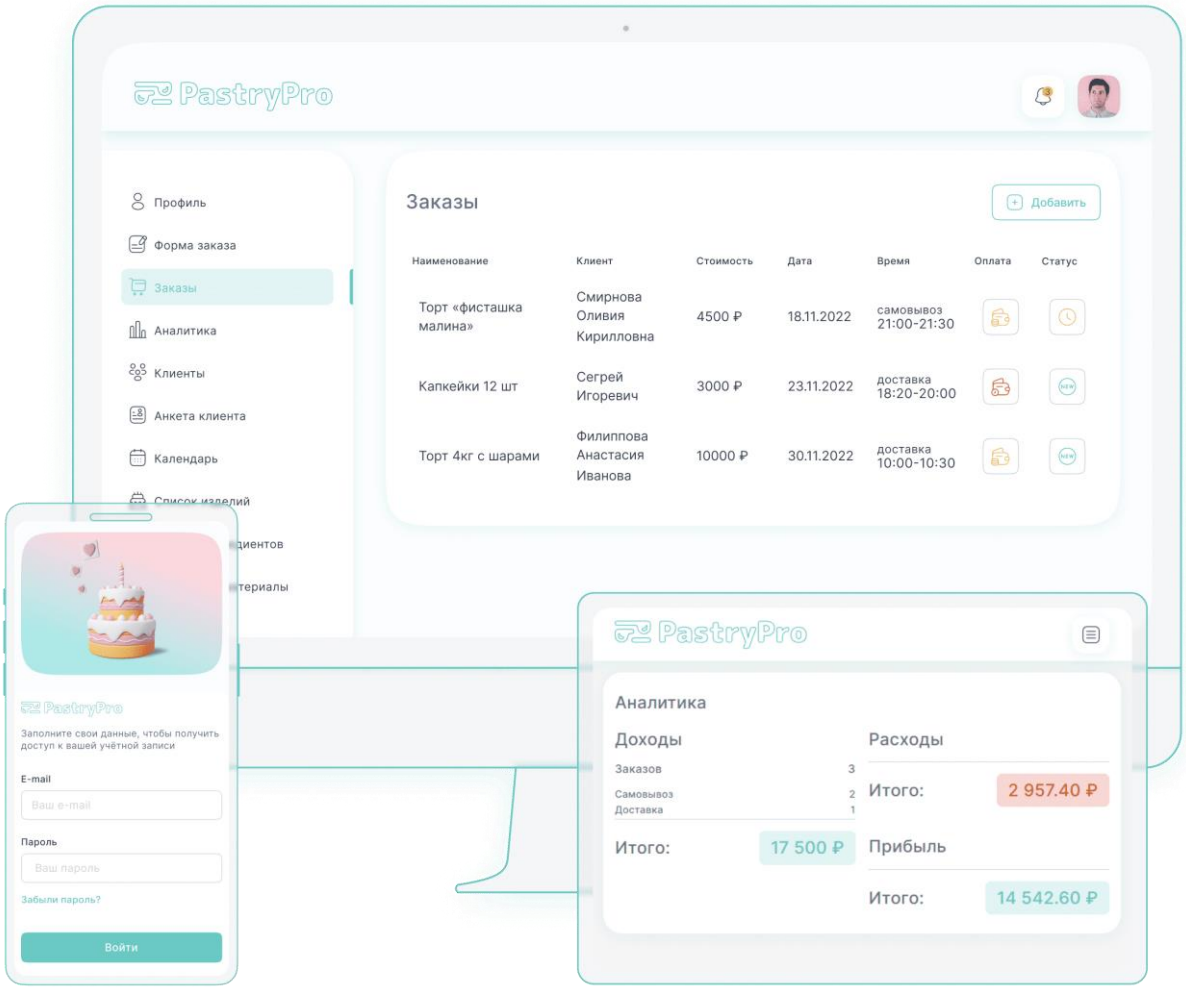

Рисунок 4 - Функционал сервиса PastryPro

2.1.5 BakeMaster

Данный сервис представляет из себя мобильное приложение для кондитеров. В приложении можно вести учет доходов и расходов в самом легком и удобном виде. Функционал приложения приведен на рисунке 5.

Преимущества:

— бесплатное приложение;

— возможность сохранения ингредиентов и рецептов. Это облегчает расчет затрат;

— множество полезных функций.

Недостатки:

— пользователи жалуются на постоянную рекламу в приложении и нестабильную работу.

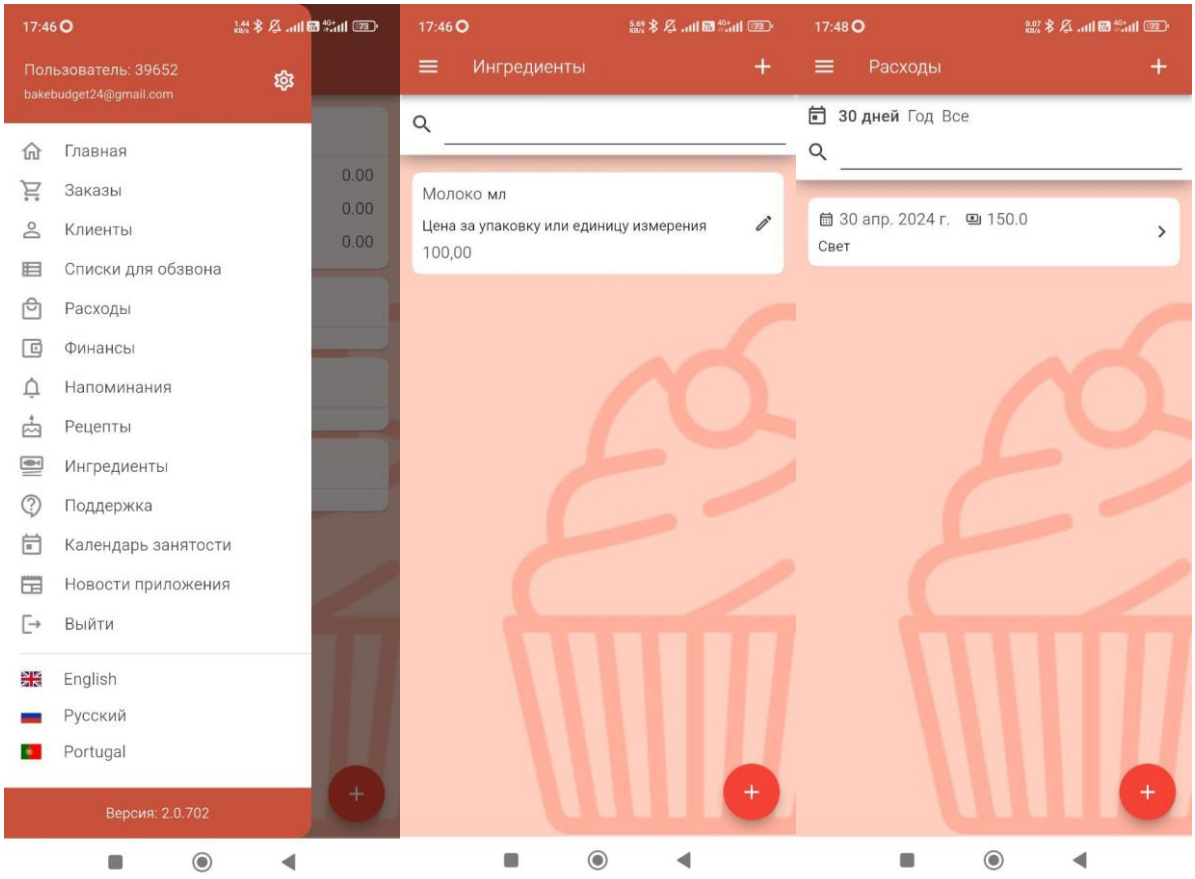

Рисунок 5 - Функционал приложения BakeMaster

2.2 Итоги анализа

В процессе анализа предметной области было установлено, что при разработке приложения следует придерживаться следующих аспектов:

— простой и понятный интерфейс;

— стабильность работы;

— сохранение ингредиентов;

— сохранение шаблонов изделий;

— формирование отчетов по доходам и расходам;

— создание заказов на основе шаблонов изделий;

— учет дополнительных расходов;

Также было установлено, что необходимо избегать таких ошибок, как:

— постоянная реклама;

— низкая, ограниченная функциональность;

— недостаток возможностей для подсчета себестоимости.

3 Графическое описание работы системы

В разделе находится графическое описание системы.

3.1 Диаграммы прецедентов

Диаграмма прецедентов (Use Case) — это графическое представление функциональных требований к системе, показывающее, как различные актеры взаимодействуют с системой для достижения конкретных целей. Она помогает четко определить и описать основные сценарии использования системы и взаимодействие пользователей с ее функционалом.

3.1.1 Диаграмма прецедентов неавторизованного пользователя

Диаграмма прецедентов для неавторизованного пользователя приведена на рисунке 6.

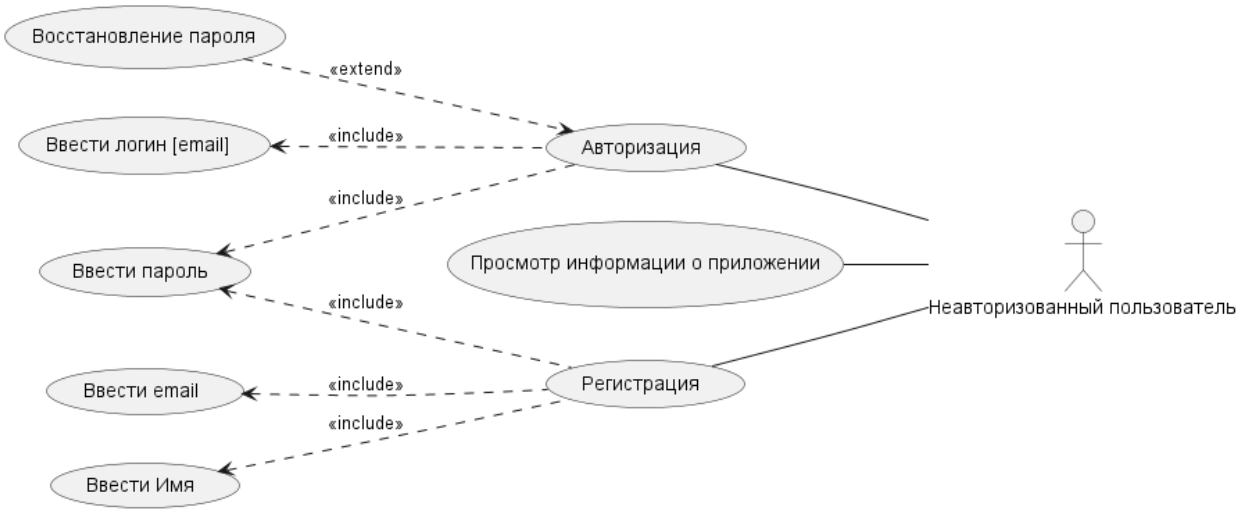

Рисунок 6 - Диаграмма прецедентов неавторизованного пользователя

3.1.2 Диаграмма прецедентов авторизованного пользователя

Диаграмма прецедентов для неавторизованного пользователя приведена на рисунках 7-9.

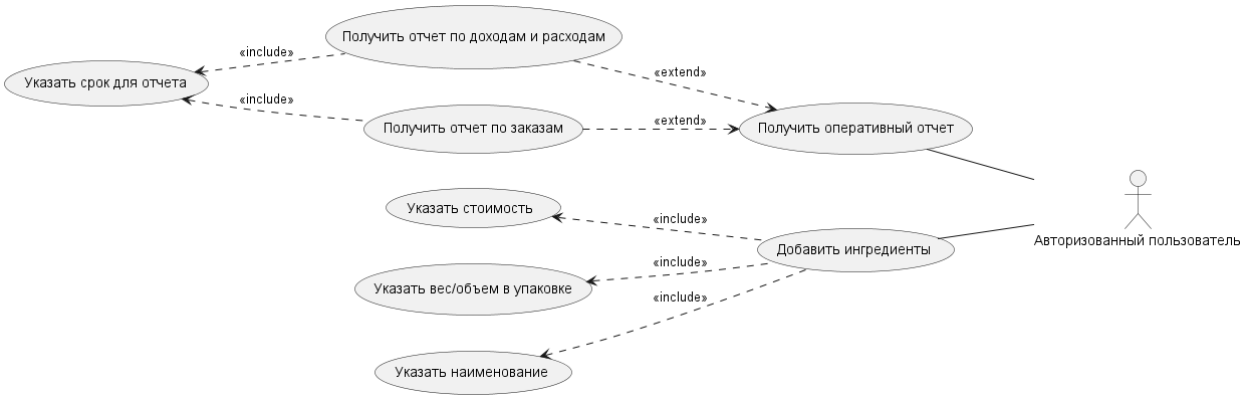

Рисунок 7 - Диаграмма прецедентов авторизованного пользователя часть 1 из 3

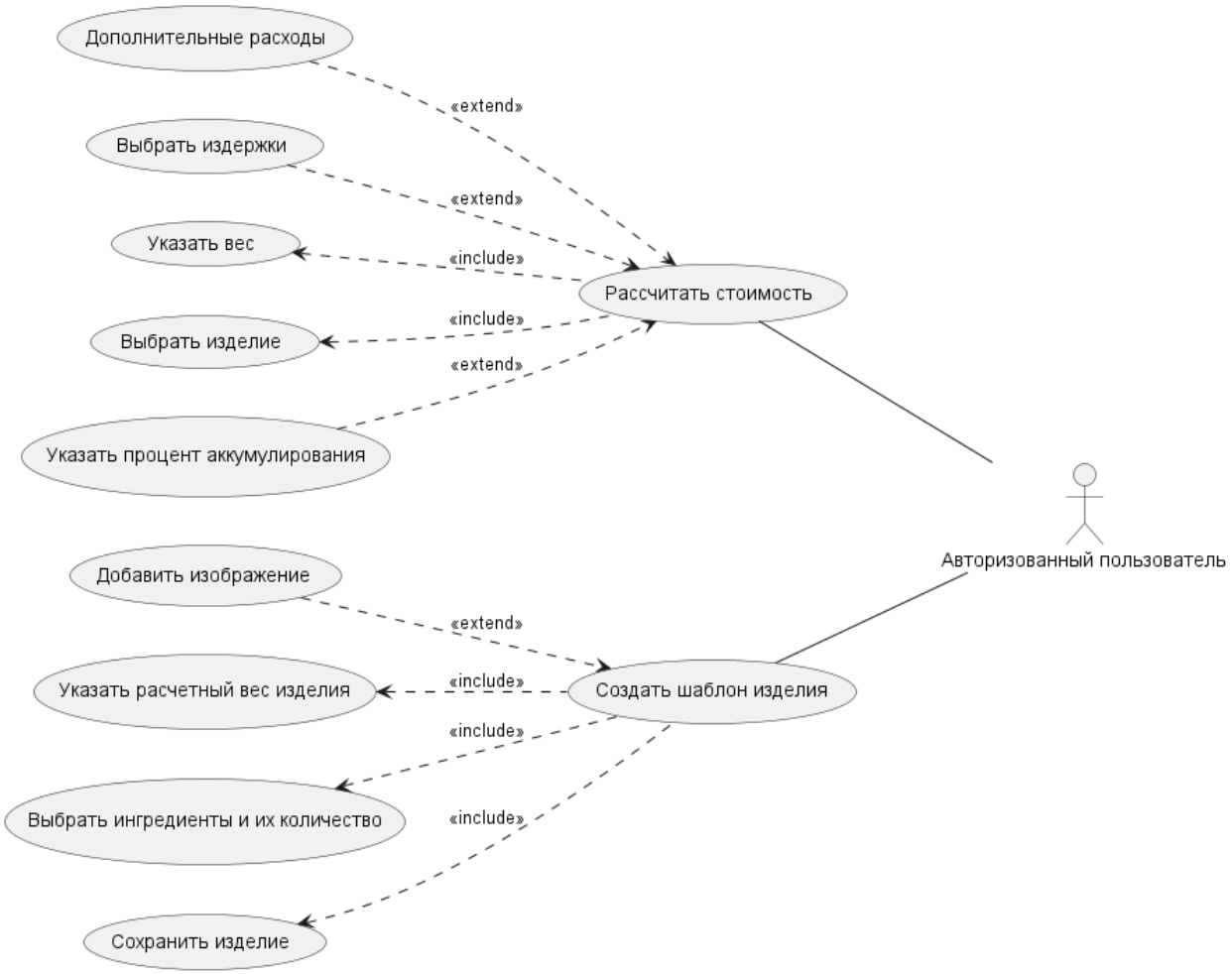

Рисунок 8 - Диаграмма прецедентов авторизованного пользователя часть 2 из 3

Рисунок 9 - Диаграмма прецедентов авторизованного пользователя часть 3 из 3

3.1.3 Диаграмма прецедентов пользователя, владеющего расширенной версией

Диаграмма прецедентов для пользователя, владеющего расширенной версией, приведена на рисунке 10.

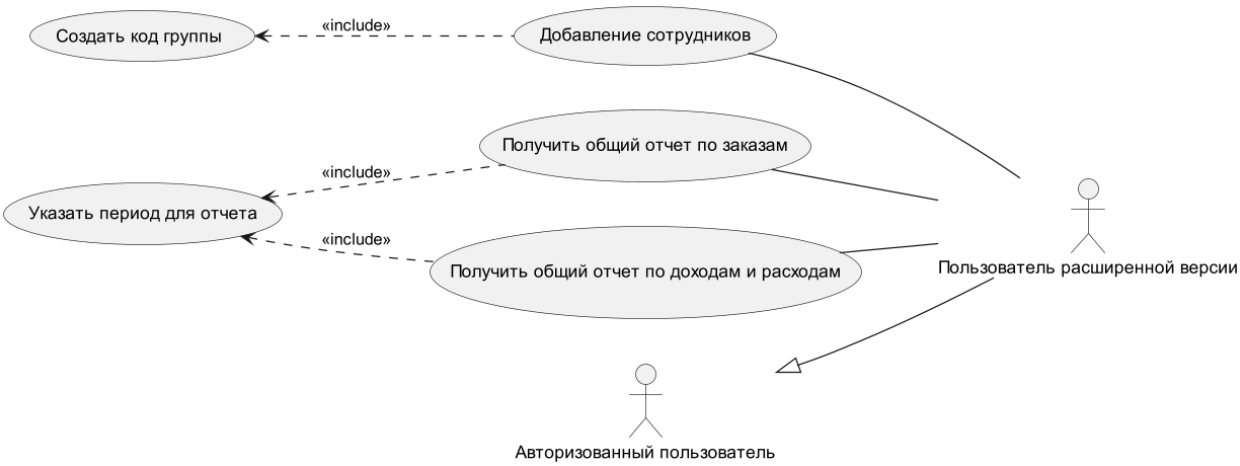

Рисунок 10 - Диаграмма прецедентов пользователя расширенной версии

3.2 Диаграмма развертывания

Проект разделен на 3 архитектурные составляющие: мобильное приложение, серверная часть и база данных. Приложение выполнено в архитектурном стиле REST API.

Для хранения и извлечения данных сервер взаимодействует с базой данных.

С диаграммой развертывания приложения можно ознакомиться на рисунке 11.

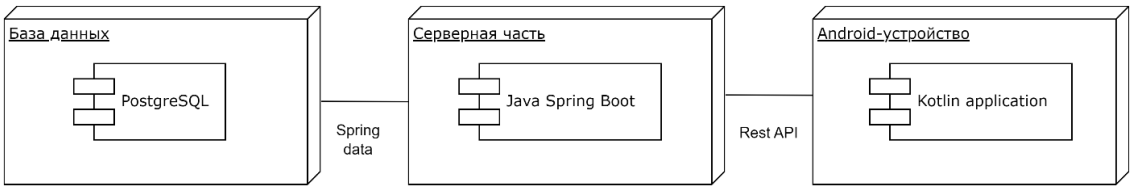

Рисунок 11 - Диаграмма развертывания приложения

3.3 Диаграммы сотрудничества

Диаграмма представляет собой графическую модель, которая показывает взаимодействие объектов в системе и сообщения между ними.

Рисунок 12 - Регистрация пользователя

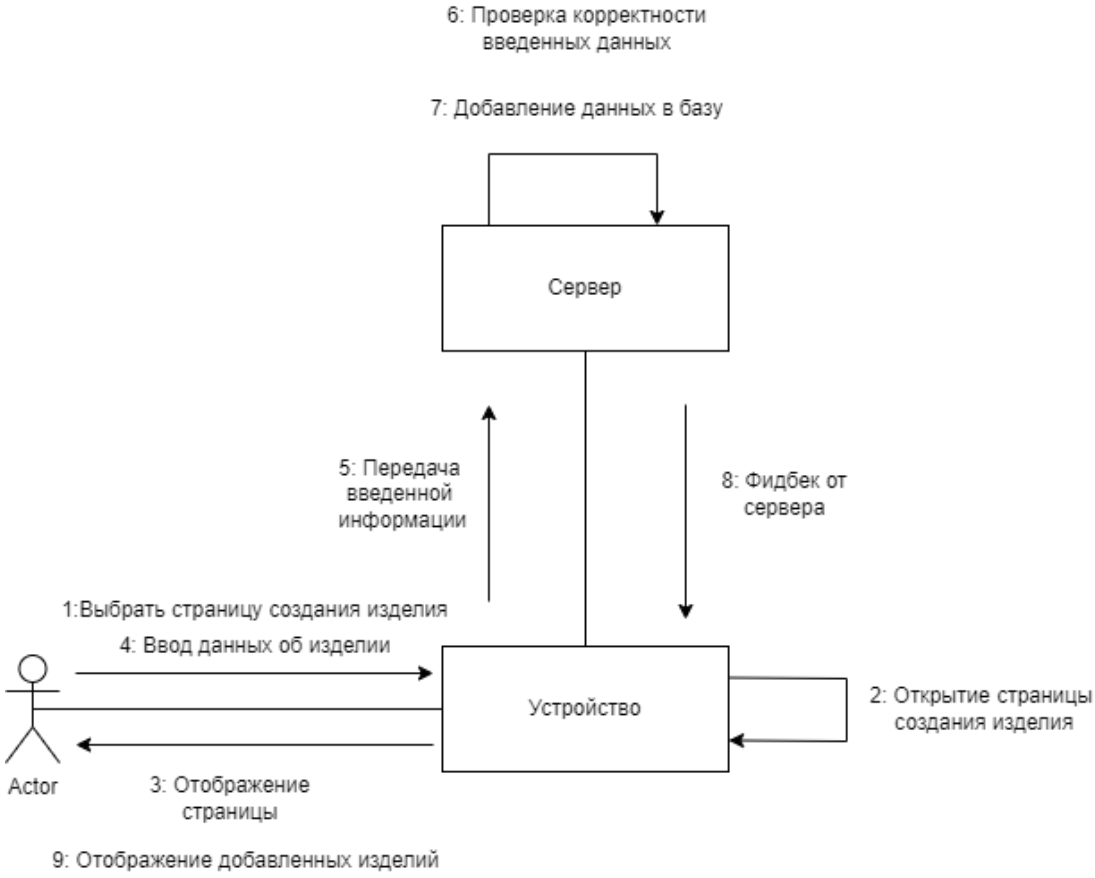

Рисунок 13 - Создание шаблона изделия

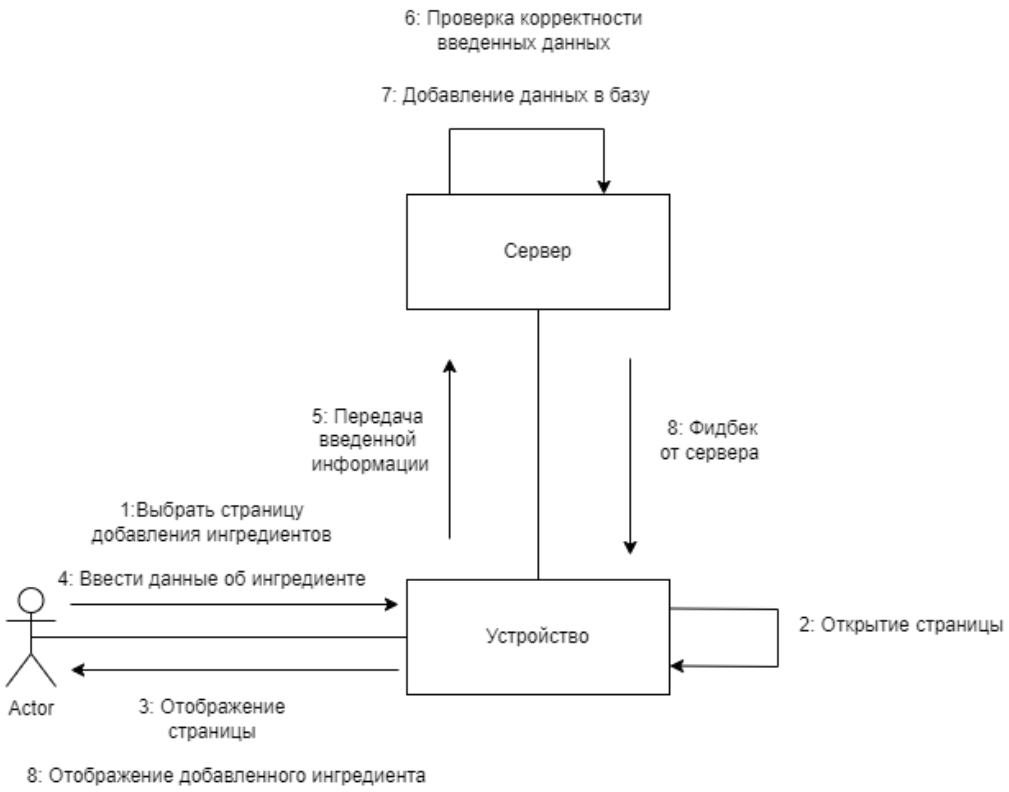

Рисунок 14 - Добавление ингредиента

3.4 Диаграммы последовательности

Диаграмма последовательности позволяет описать взаимодействие между объектами в системе в виде последовательности сообщений, действий и операций, отображая порядок выполнения действий и обмена информацией между объектами во времени.

3.4.1 Диаграмма последовательности для неавторизованного пользователя

Диаграммы последовательности для неавторизованного пользователя приведены на рисунках 15-16

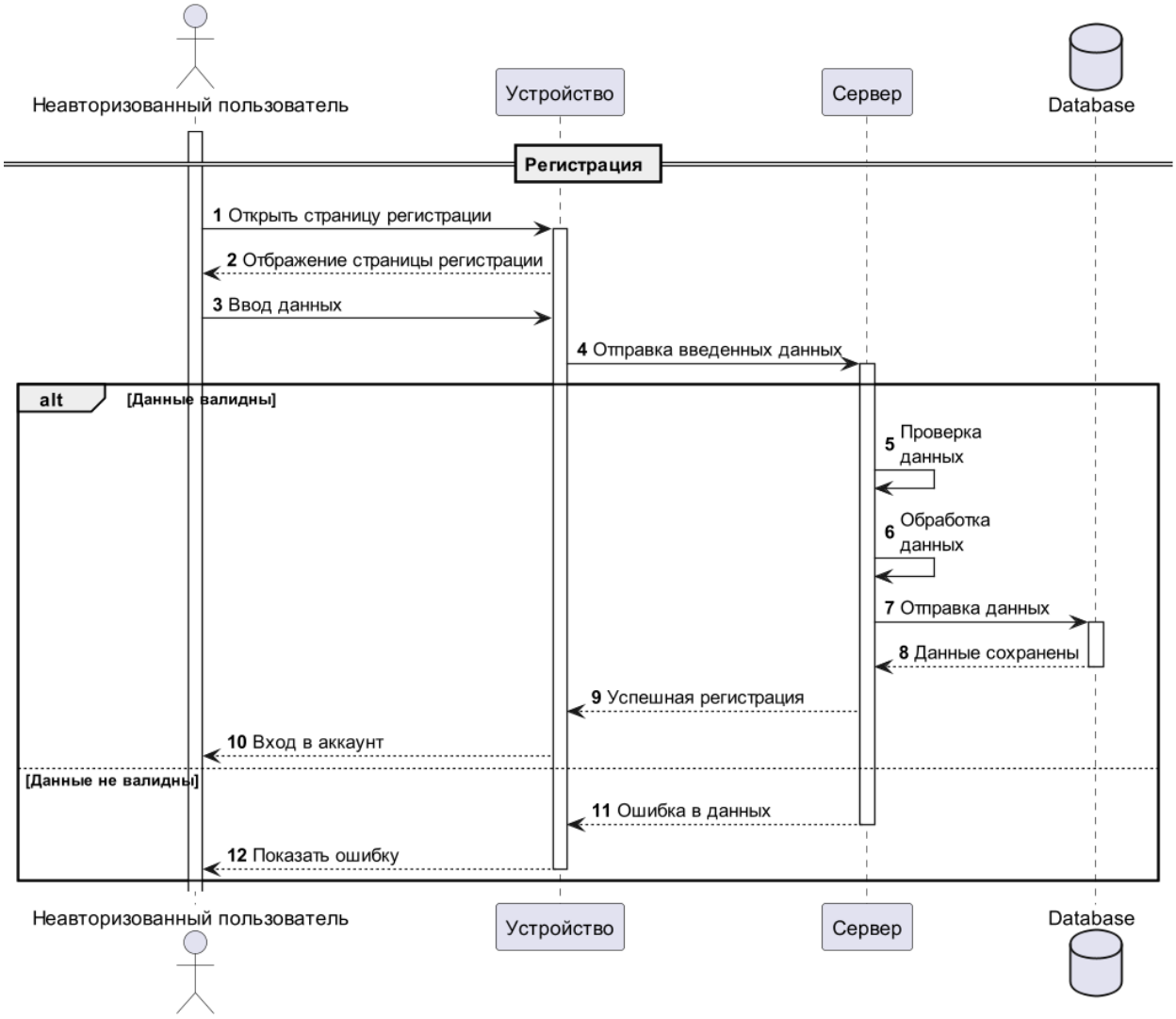

Рисунок 15 - Диаграмма последовательности для регистрации пользователя

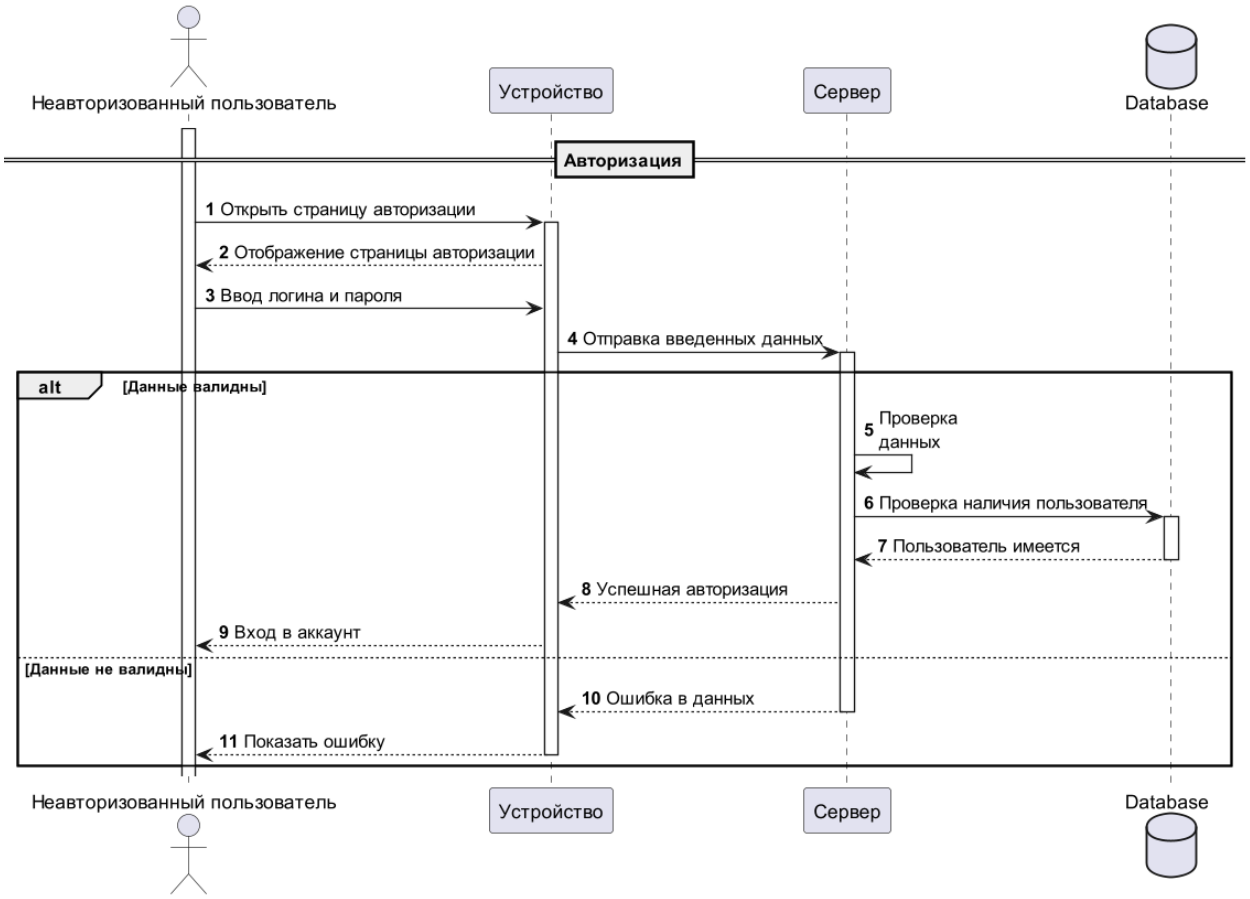

Рисунок 16 - Диаграмма последовательности для авторизация пользователя

3.4.2 Диаграмма последовательности для авторизованного пользователя

Диаграммы последовательности для авторизованного пользователя приведены на рисунках 17-24.

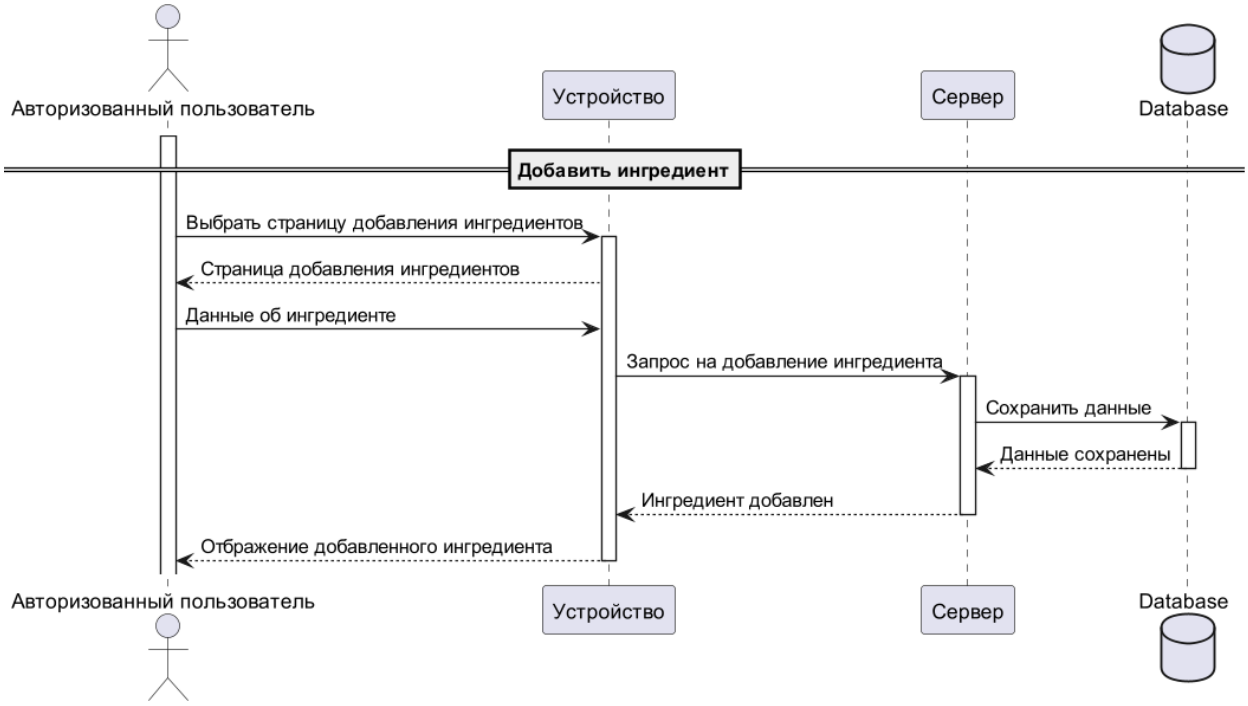

Рисунок 17 - Диаграмма последовательности для добавления ингредиента

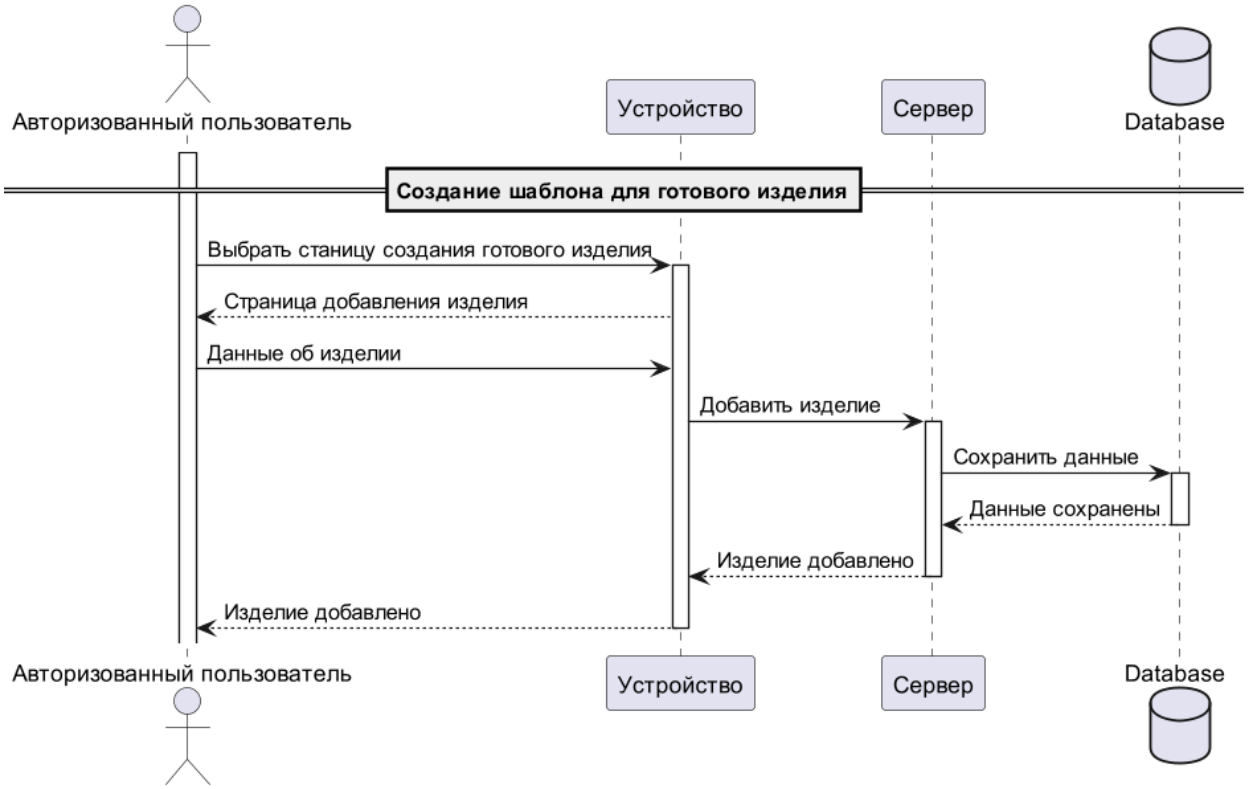

Рисунок 18 - Диаграмма последовательности для добавления изделия

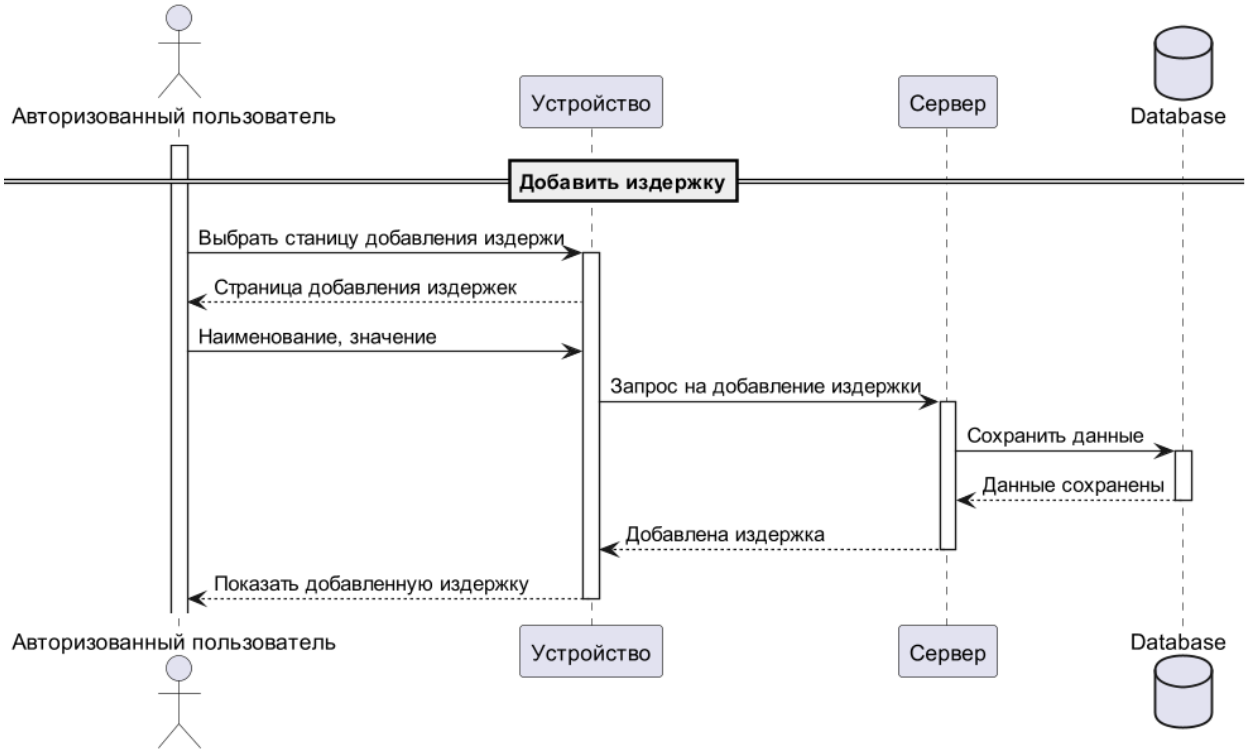

Рисунок 19 - Диаграмма последовательности для добавления издержки

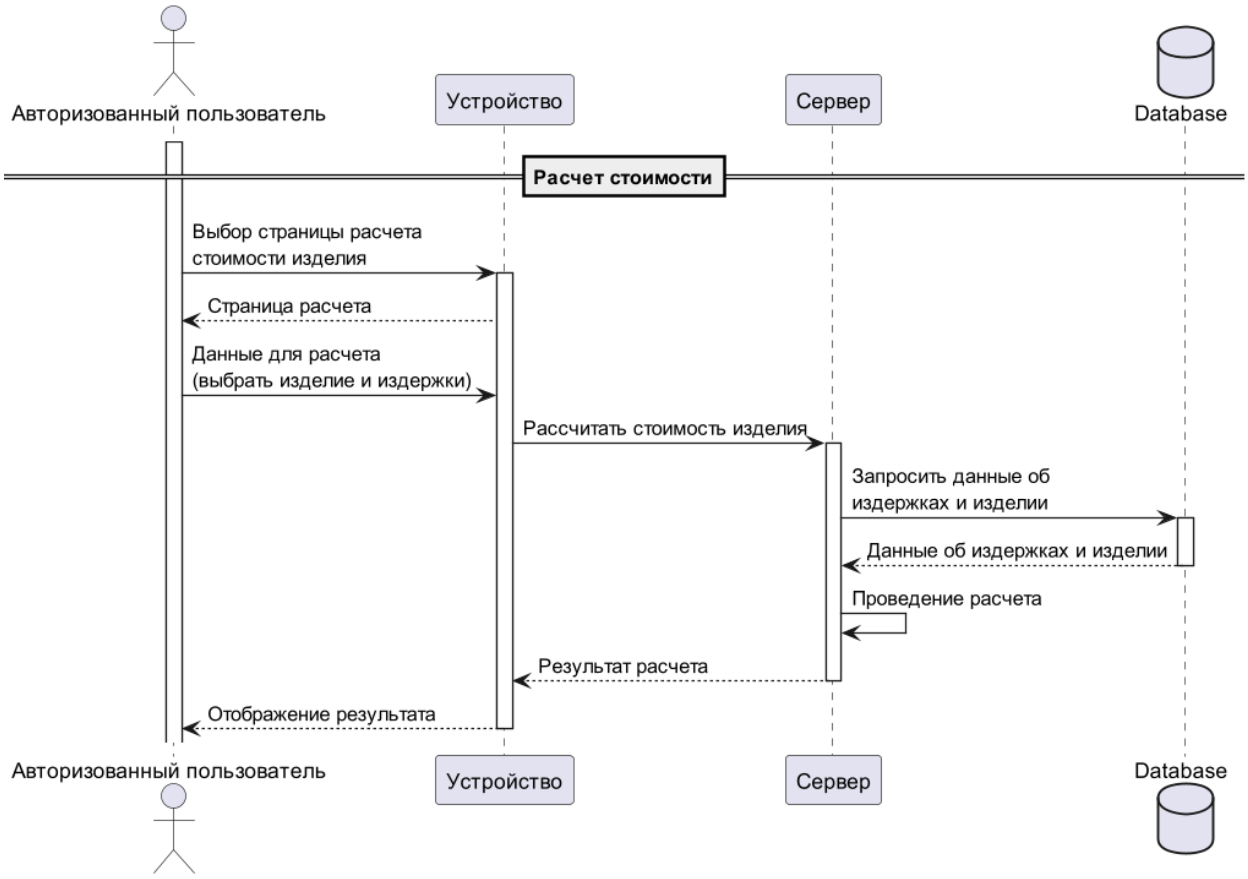

Рисунок 20 - Диаграмма последовательности для осуществления расчета стоимости

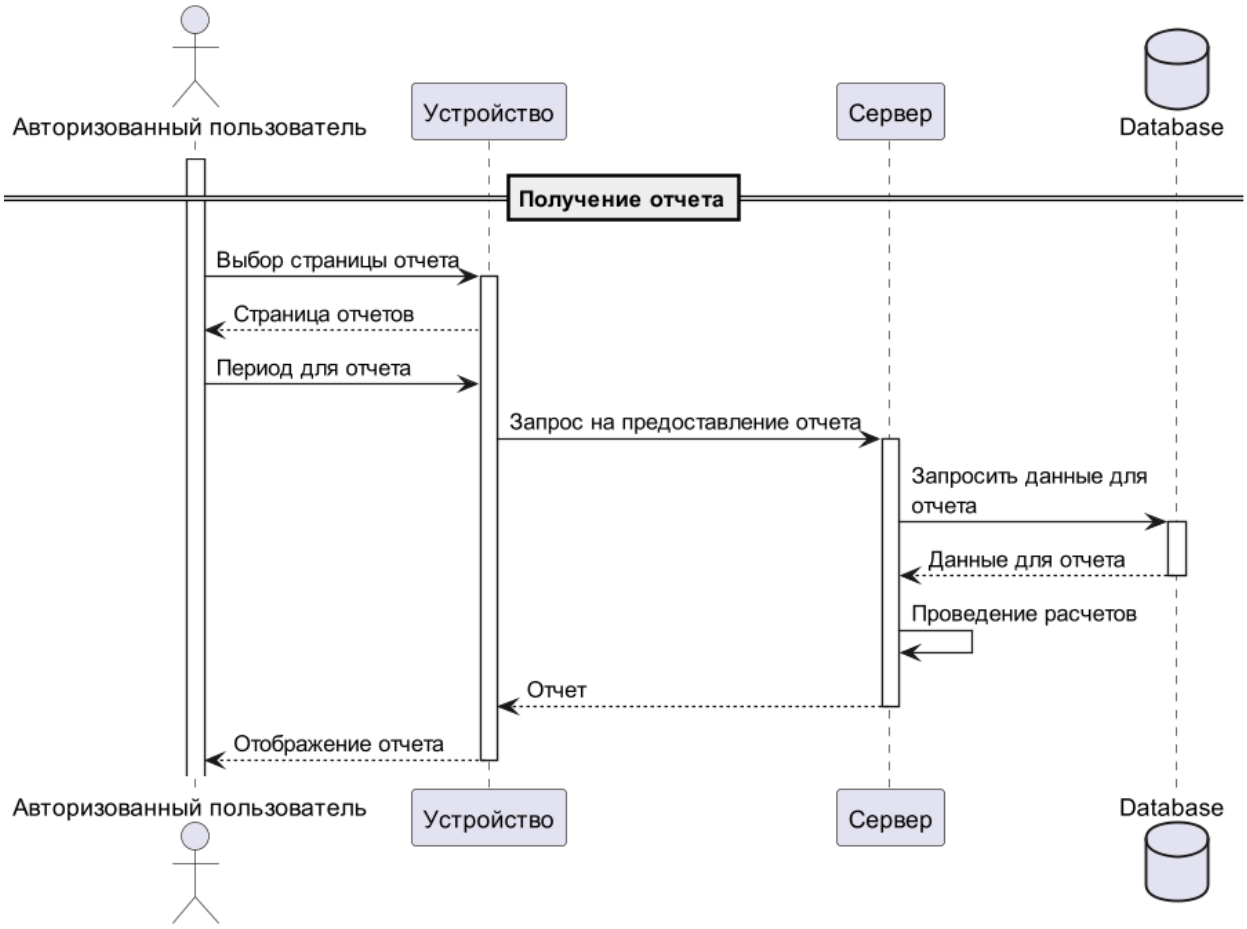

Рисунок 21 - Диаграмма последовательности для получения отчета

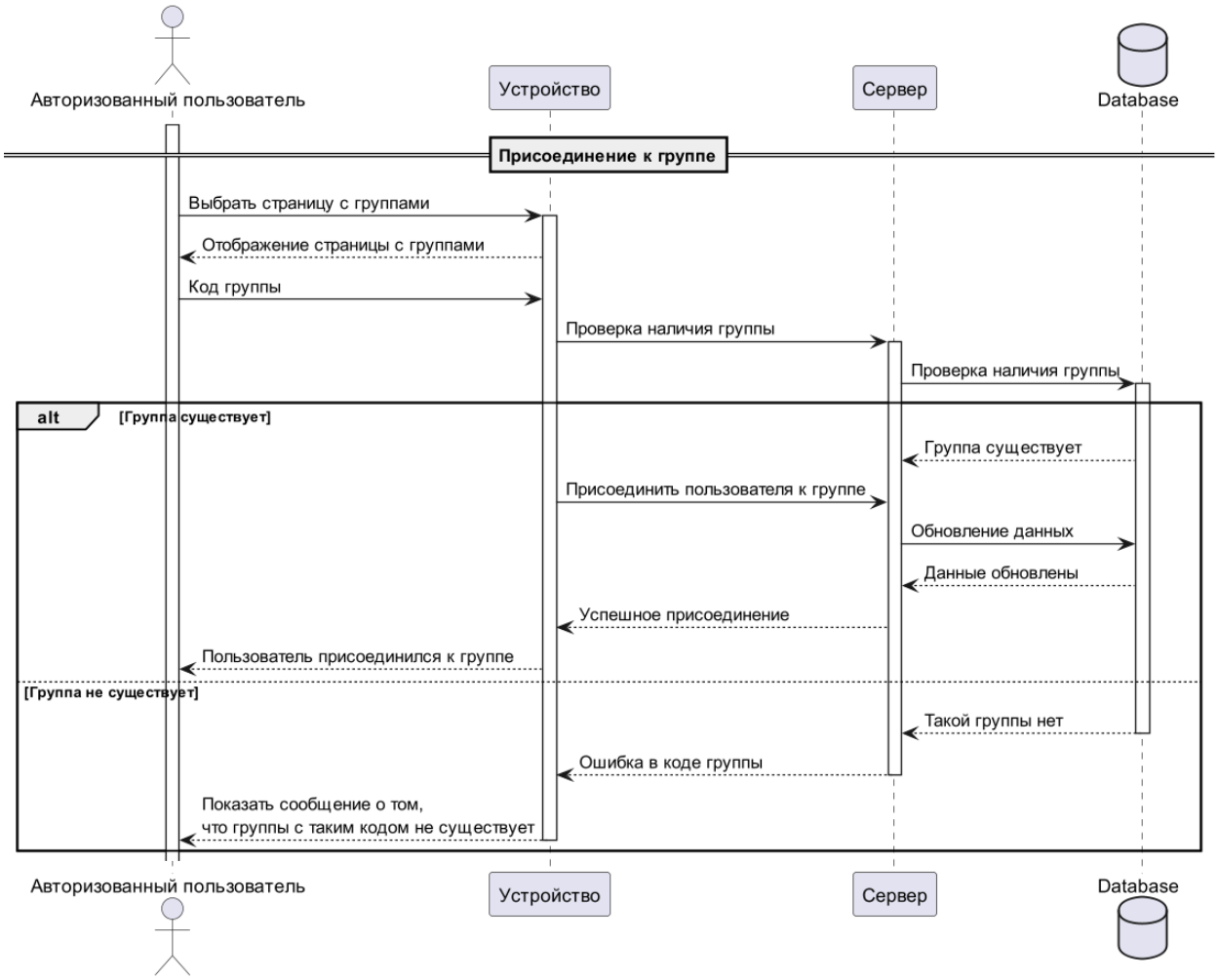

Рисунок 22 - Диаграмма последовательности для присоединения к группе

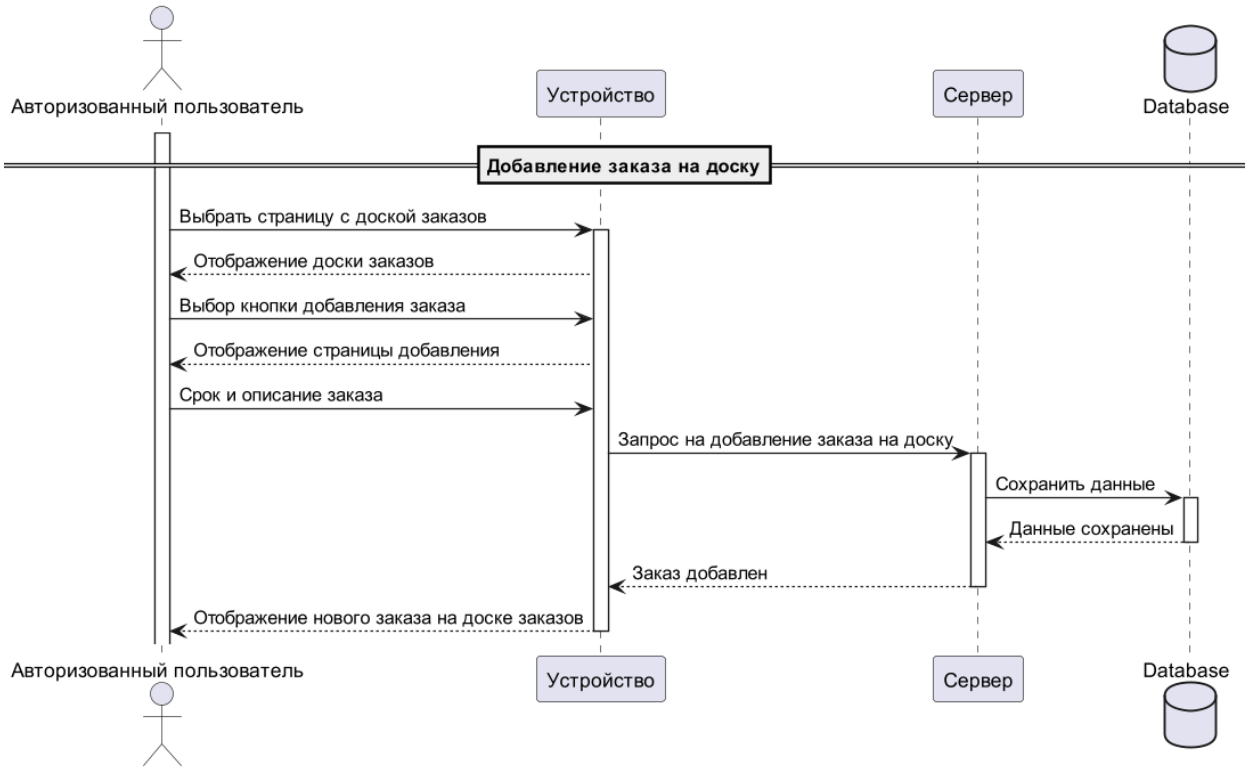

Рисунок 23 - Диаграмма последовательности для добавления заказа на доску

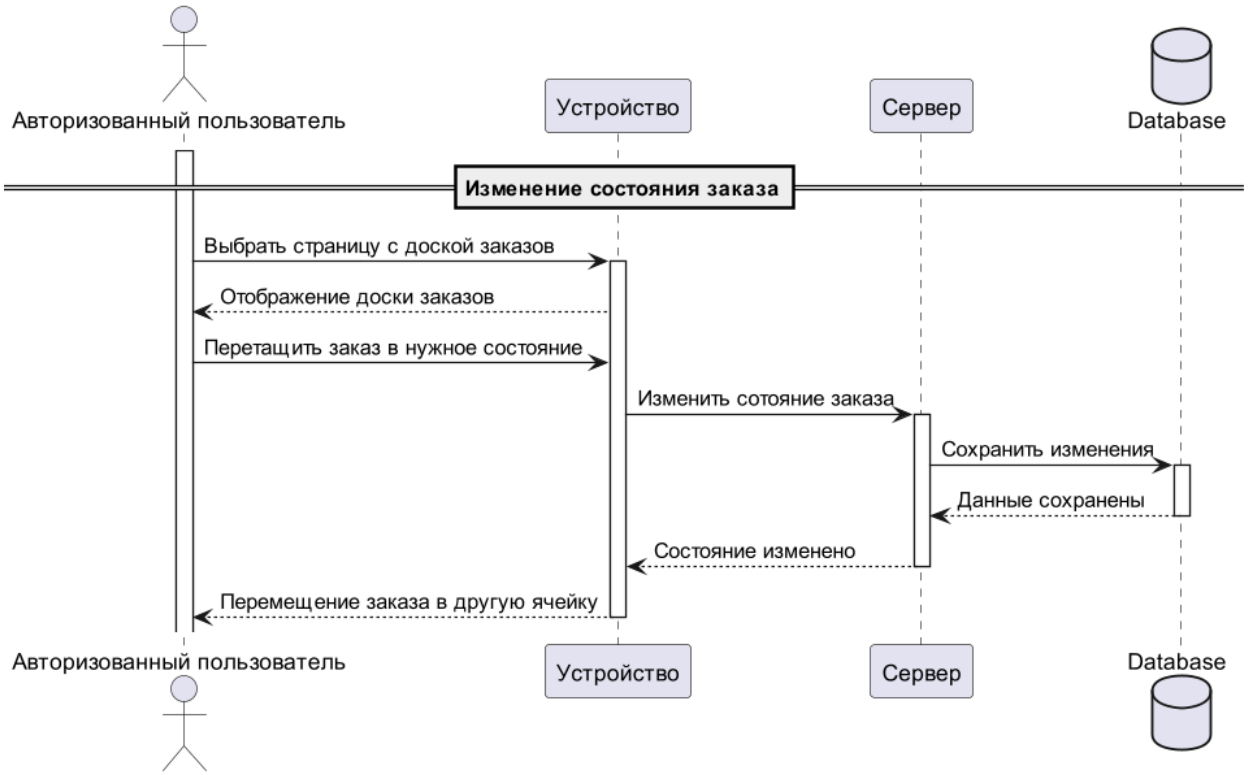

Рисунок 24 - Диаграмма последовательности для изменения состояния заказа

3.4.3 Диаграмма последовательности для пользователя расширенной версии

Диаграмма последовательности для пользователя расширенной приведена на рисунке 25.

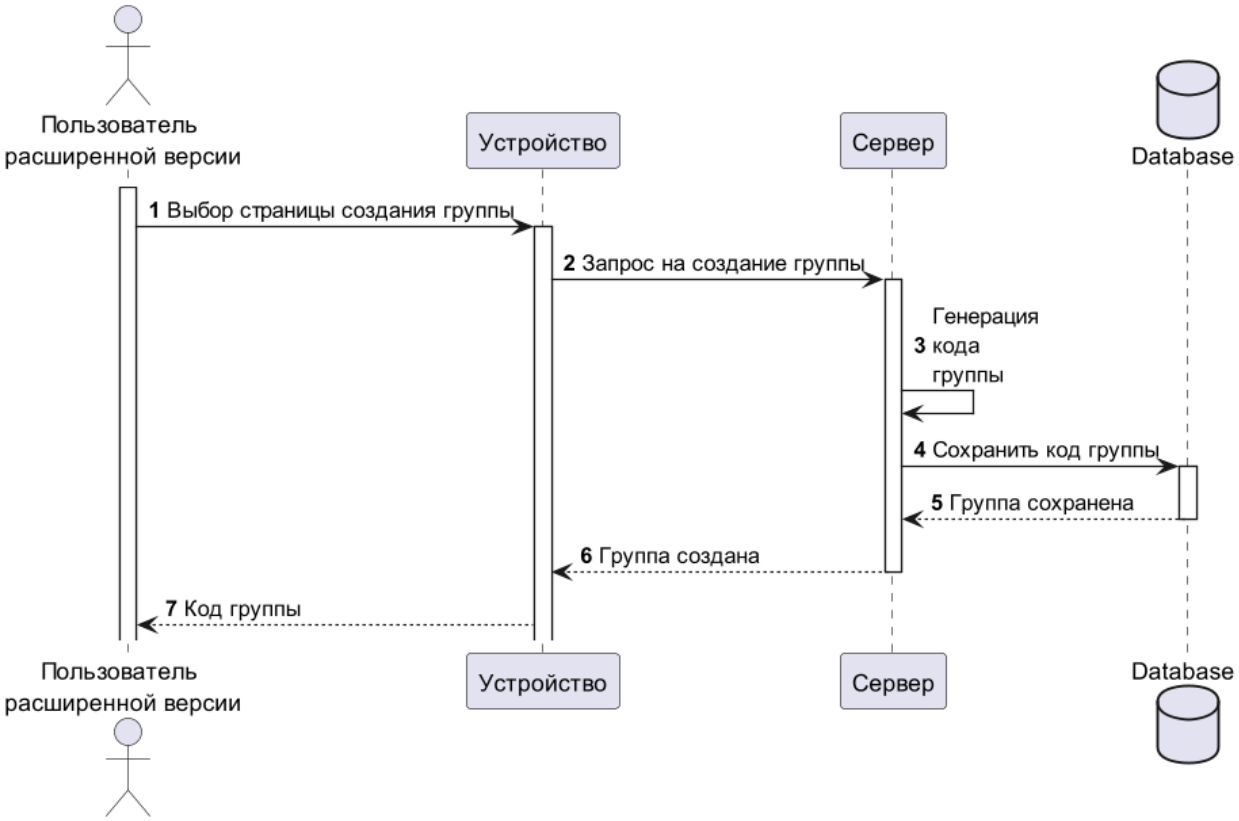

Рисунок 25 - Диаграмма последовательности для создания группы

4 Реализация

В данном разделе будет описана реализация приложения.

4.1 Средства реализации

Приложение должно соответствовать архитектуре клиент-сервер. Для реализации серверной части были выбраны следующие средства:

— ОС Windows 10 / Ubuntu 22.10;

— фреймворк Spring 3.2.4;

— Java версии 17;

— СУБД PostgreSQL 16;

— объектное хранилище данных MinIO;

— система контроля версий Git 2.45.1.

Для серверной части нашего проекта был выбран язык программирования Java в связке с фреймворком Spring и СУБД PostgreSQL. Этот выбор обусловлен несколькими факторами:

— фреймворк Spring предоставляет мощные инструменты для создания серверных приложений, а также обеспечивает удобное управление зависимостями и конфигурацией;

— выбор PostgreSQL в качестве базы данных обусловлен его надежностью, производительностью и расширяемостью;

В качестве средств реализации клиентской части были выбраны:

—Android Studio;

—ЯП Kotlin 1.9.20;

—система контроля версий Git 2.45.1.

Для мобильного приложения был выбран язык программирования Kotlin. Kotlin – язык программирования, полностью совместимый с Java, что обеспечивает плавную интеграцию с серверной частью нашего проекта.

4.2 Реализация базы данных

На рисунке 26 представлена ER диаграмма используемой на проекте базы данных.

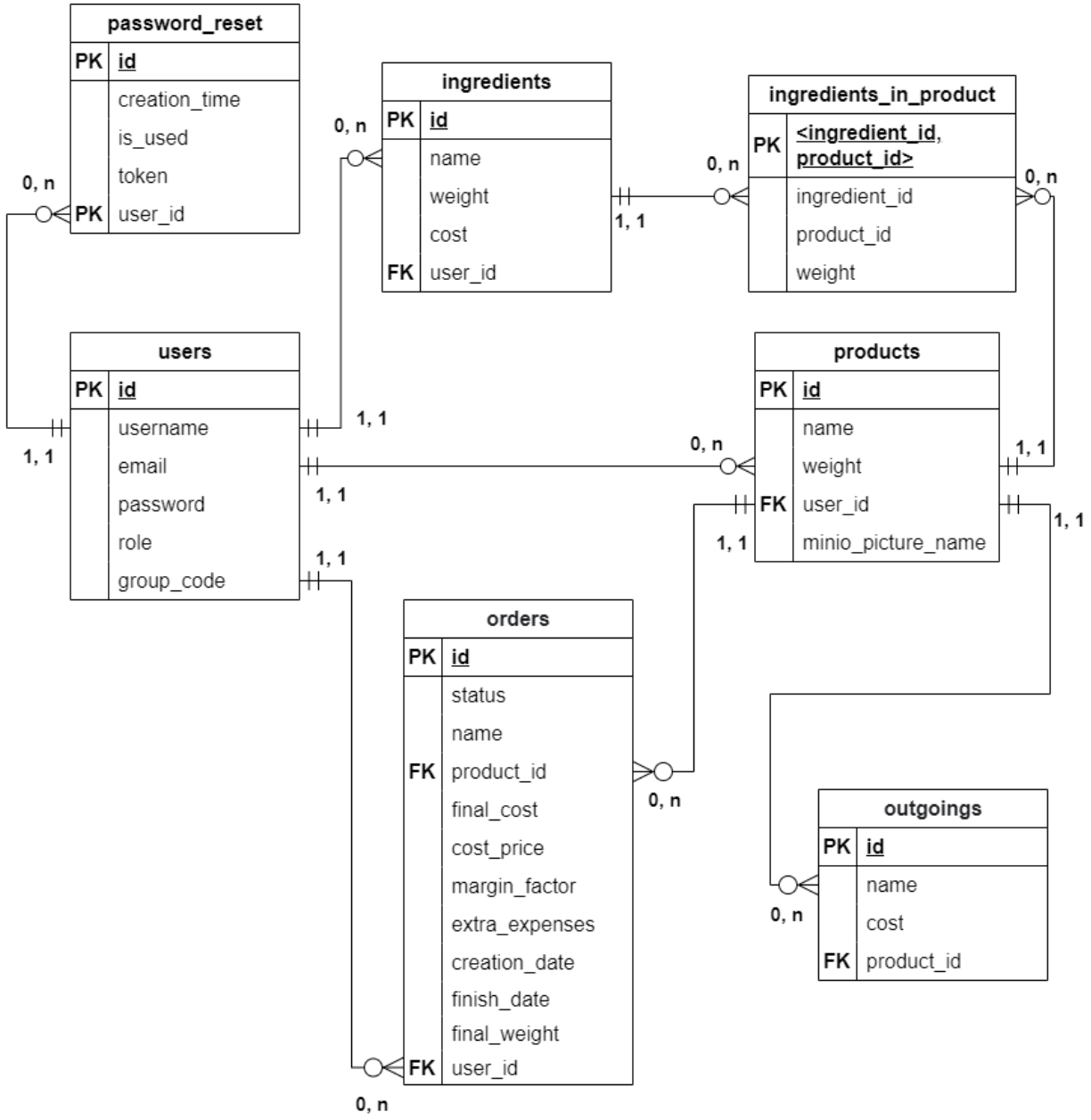

Рисунок 26 - ER диаграмма

Как видно из диаграммы база данных содержит шесть сущностей. Далее будет приведена краткая информация о каждой сущности.

Users – это пользователь, который может быть обычным пользователем (ROLE_USER) или пользователем расширенной версии

(ROLE_ADVANCED_USER). Сущность Users связана с сущностями Orders (заказы), Ingredients (ингредиенты) и Products (готовые изделия) связями 1, 1:0, n. Поскольку у одного пользователя может быть несколько ингредиентов, готовых изделий и заказов, а может и не быть вовсе, но у каждого заказа, готового изделия и ингредиента должен быть свой пользователь.

Orders – это заказы, созданные пользователем. Каждый пользователь может создать один или несколько заказов (а может не создать и ни одного).

Outgoings – это издержки, которые возникают при изготовлении готового изделия. У каждого готового изделия (Products) может быть одна или несколько издержек, а может и не быть вовсе.

Products – это готовые изделия, которые создал пользователь. Готовые изделия связаны с издержками (Outgoings) и ингредиентами (Ingredients in product). У каждого готового изделия может быть один или несколько ингредиентов (Ingredients in product), а может и не быть вовсе.

Ingredients – это ингредиенты, которые добавил пользователь. В дальнейшем пользователь будет добавлять их в готовое изделие. Они связаны с пользователем и ингредиентами в продукте (Ingredients in product). У каждый ингредиент может иметь один или несколько ингредиентов продукте (Ingredients in product), а может и не иметь вовсе.

Ingredients in product – это ингредиенты, используемые в готовом изделии. Они выбираются из добавленных раннее пользователем. В них указывается количество нужное для изготовления готового изделия.

Password reset – это сущность, используемая в процедуре восстановления пароля. В ней хранится временный токен для смены пароля и состояние этого токена (использован или нет).

На рисунке 27 представлена физическая модель базы данных.

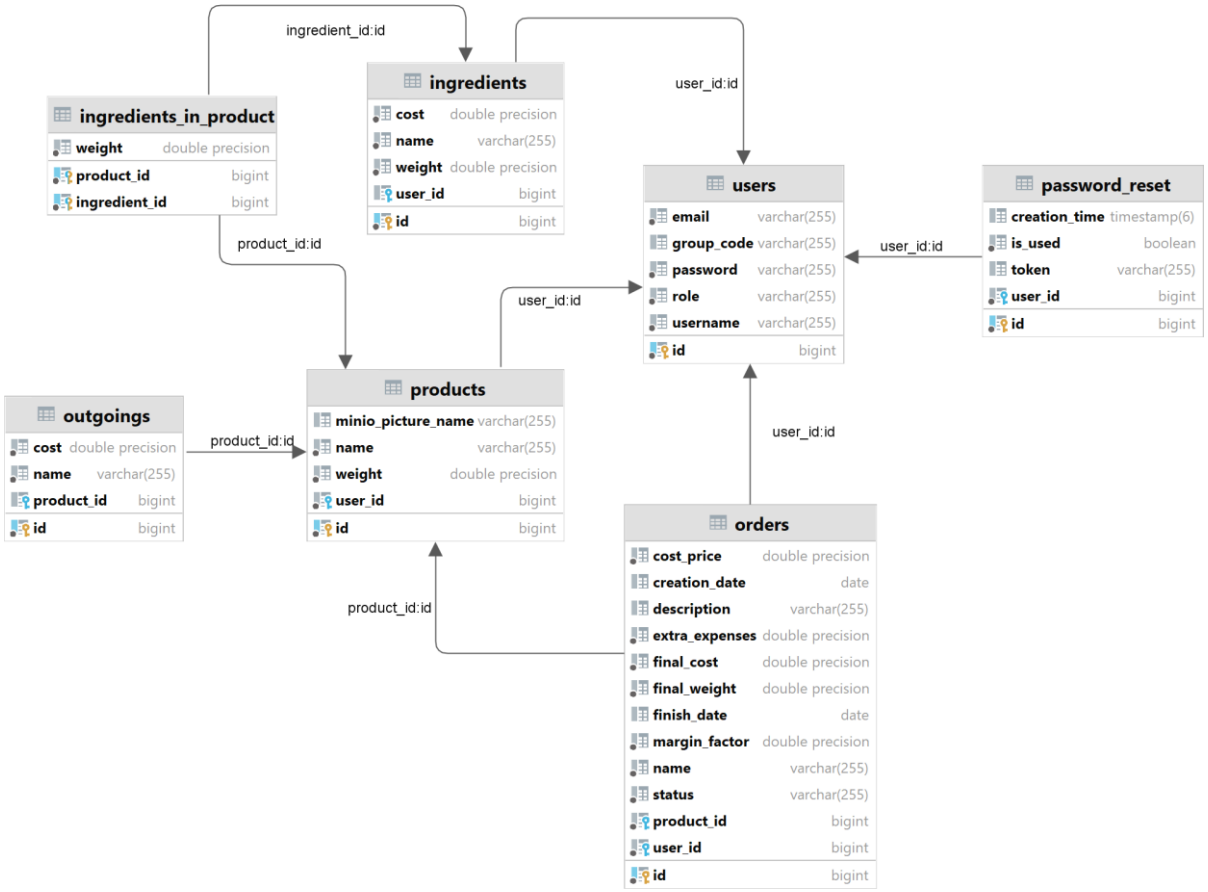

Рисунок 27 - Физическая модель базы данных

4.3 Реализация серверной части приложения

Серверная часть приложения реализована на основе Spring Boot и включает следующие компоненты:

— модели (model) сущностей базы данных, каждая из которых содержит аннотации JPA для определения полей таблицы и связей между таблицами;

— репозитории (repository), которые определяют интерфейсы для взаимодействия с базой данных. Эти интерфейсы наследуются от JpaRepository и обеспечивают CRUD-операции, а также позволяют создавать специальные запросы с использованием названий полей классов и ключевых слов;

— сервисы (service) и их имплементации, взаимодействующие с репозиториями для выполнения операций с данными;
— DTO, используемые для передачи данных между клиентом и сервером. В их реализации используются аннотации для валидации полей;
— мапперы (mapper), используемые для преобразования сущностей в DTO и обратно. С их помощью осуществляется конвертация данных в нужный формат;
— MinIO Template, используемый как компонент для работы с MinIO и определяющий методы для загрузки, скачивания и удаления объектов. А также компонент MinIO Properties, который содержит конфигурационные параметры для подключения к MinIO;
— AuthEntryPoint, AuthTokenFilter, JwtUtils являются компонентами для генерации, валидации и обновления JWT-токенов;
— Enum ролей пользователя, определяющий возможные роли пользователей (ROLE_USER, ROLE_ADVANCED_USER) и их соответствующие права доступа;
— Spring Security, используемый для настройки безопасности приложения. В нем содержатся фильтры для проверки JWT, конфигурация ролей и права доступа;
— контроллеры (controller) обрабатывают HTTP-запросы, вызывая соответствующие методы бизнес-логики через соответствующие сервисы, и возвращают данные и статус клиенту;
— Exception Handler (обработчик исключений), который обрабатывает ошибки, возникающие в приложении;
— Cost Counter, отвечающий за расчет себестоимости и конечной стоимости изделия;
— Report Calculation, отвечающий за генерацию отчетов по доходам или заказам за определенный период;

— email Service, служащий для отправки электронных писем, используемый для процедуры восстановления пароля;
— конфигурационный файл (application properties), содержащий конфигурационные параметры для приложения.

4.4 Реализация клиентской части приложения

В данном разделе будет описана реализация клиентской части.

4.4.1 Общая информация

Клиентская часть приложения реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. Клиентская часть содержит в себе следующие структурные компоненты:

— пакет со всеми моделями – ингредиент, продукт, заказ, пункт меню, издержка, ингредиент в продукте, а также все модели, использующиеся для передачи данных в запросах и получении данных (models);
— пакет, содержащий все http методы обращения к эндпоинтам (api);
— пакет с бизнес-логикой приложения (отправка запросов к серверу и обработка ответов) (services);
— пакет со своими графическими компонентами, которые могут переиспользоваться в разных местах программы (components);
— пакет со всеми экранами приложения (screens);
— пакет с настройками темы и цветов приложения (ui.theme);
— пакет с дополнительными функциями, такими как валидаторы и сообщения пользователю (utils).

4.4.2 Графический интерфейс

Далее будут представлены экраны приложения.

Главная страница содержит логотип приложения, кнопку для просмотра информации о приложении и боковое меню.

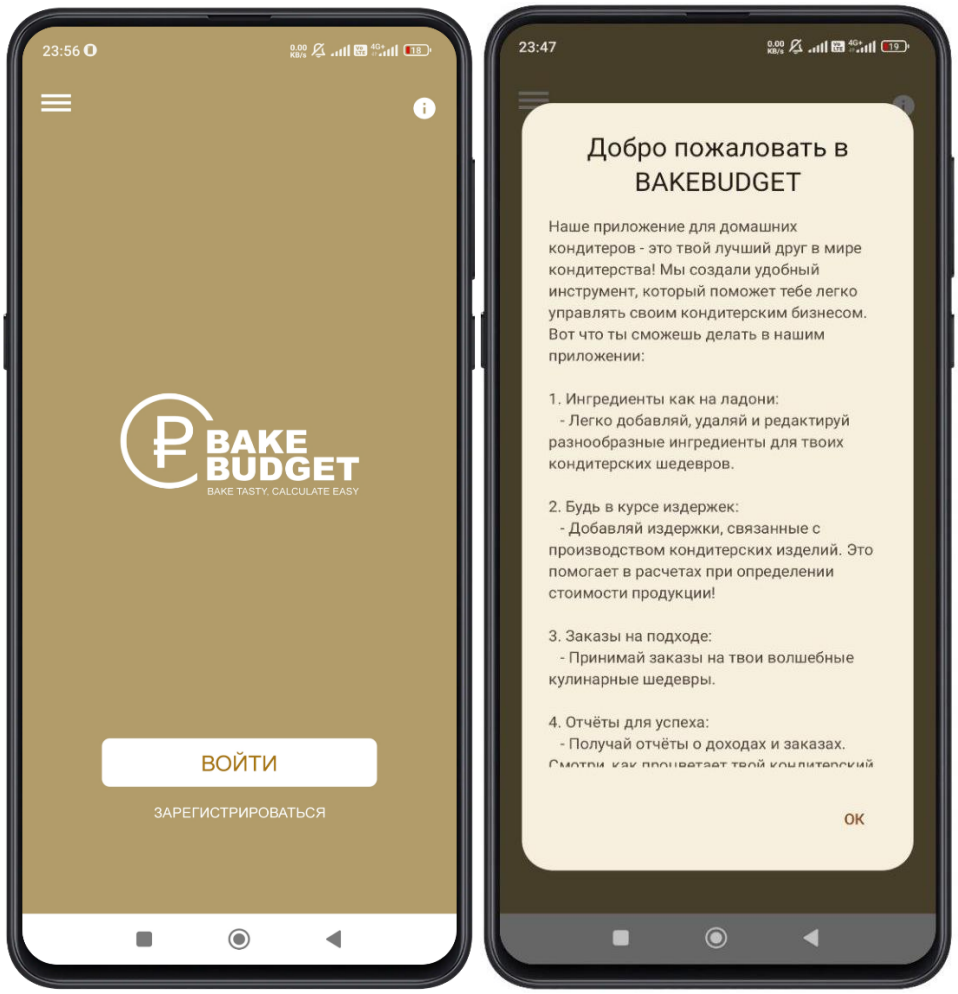

Рисунок 28 - Главная страница и информация о приложении

При свайпе вправо или нажатии кнопки бокового меню появляется боковое меню.

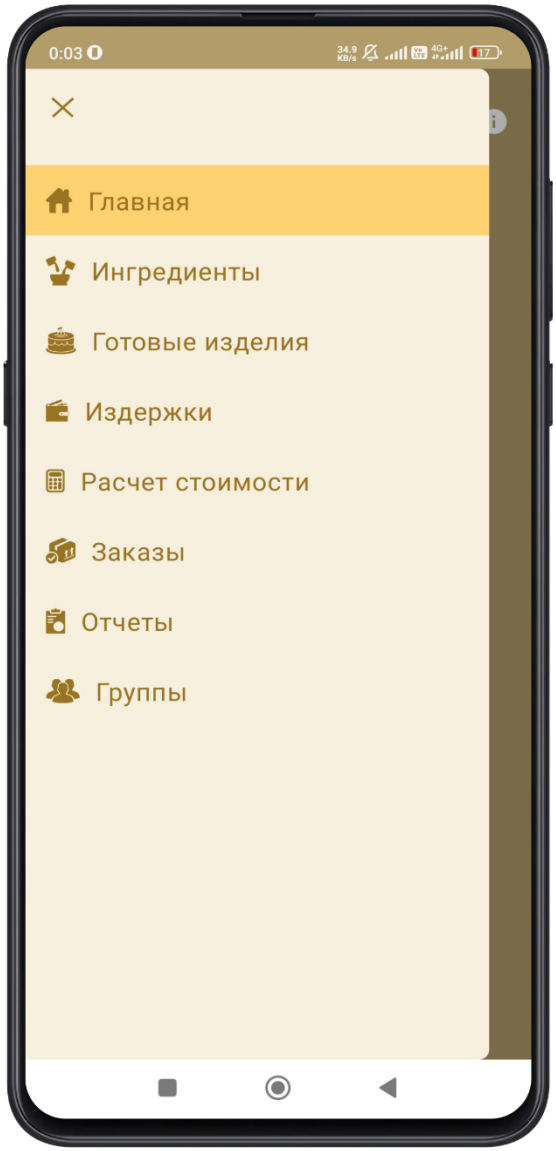

Рисунок 29 - Боковое меню

Приложение позволяет пользователю добавлять ингредиенты и потом использовать их при создании шаблона готового изделия.

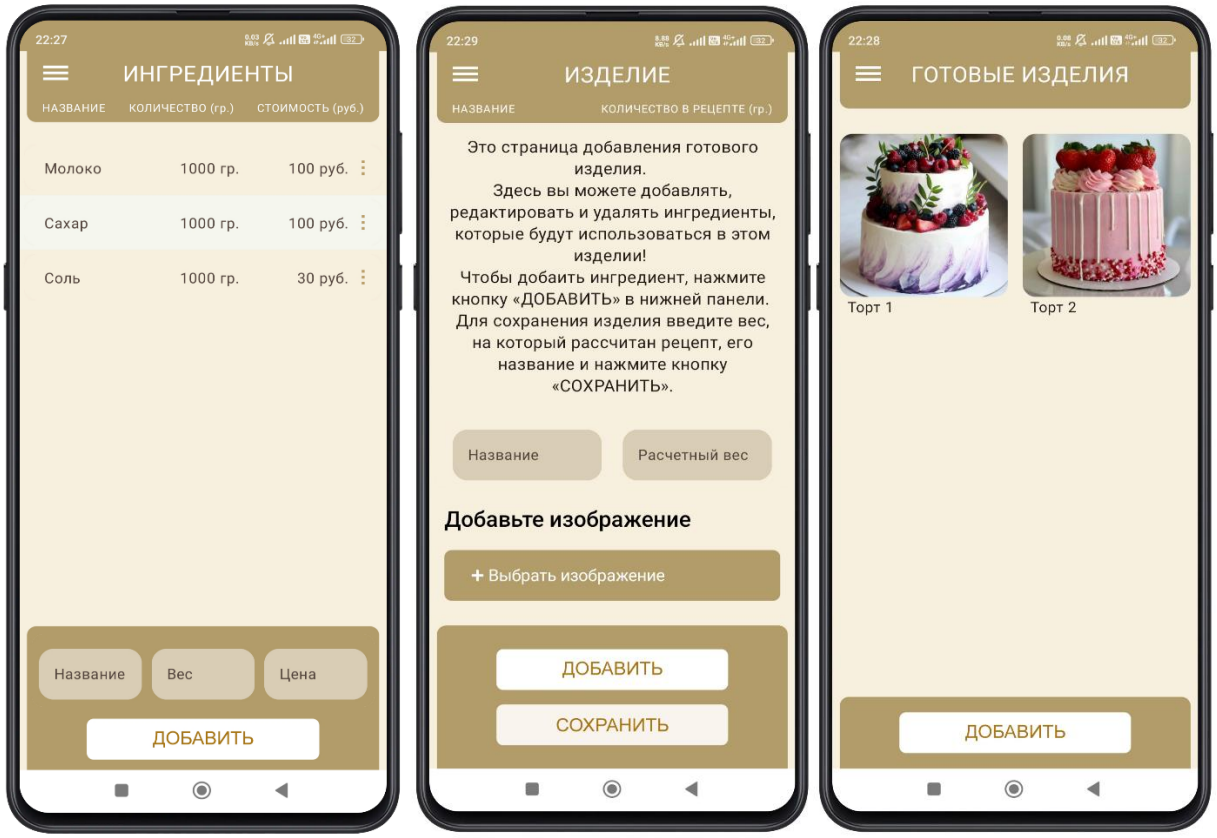

Рисунок 30 - Страницы ингредиентов и готовых изделий

Далее пользователь может указать издержки на странице «Издержки», а затем произвести расчёт стоимости заказа.

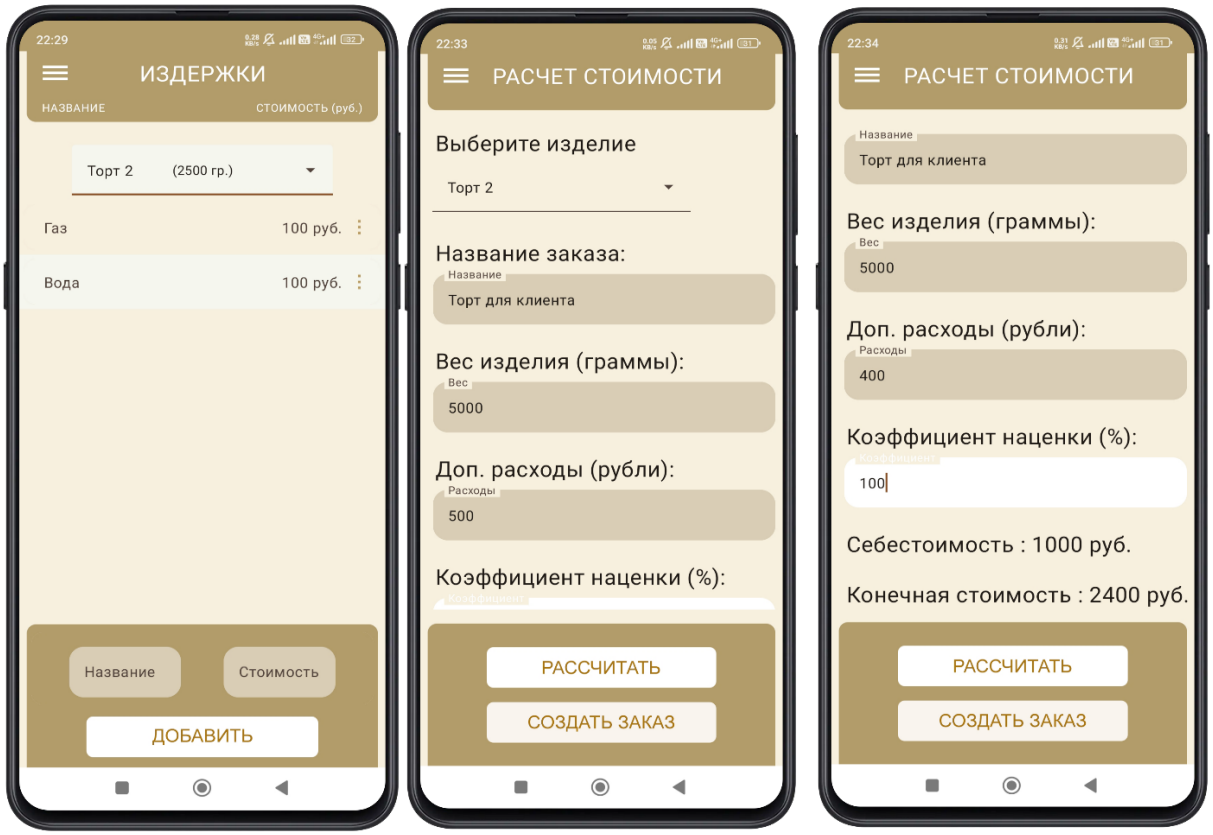

Рисунок 31 - Страницы издержек и расчета стоимости

После создания заказ будет отображаться на странице заказов, где можно изменять его состояние (Не начат, В процессе, Завершен, Отменен). После этого можно сформировать отчет, указав срок, по которому пользователь может его получить.

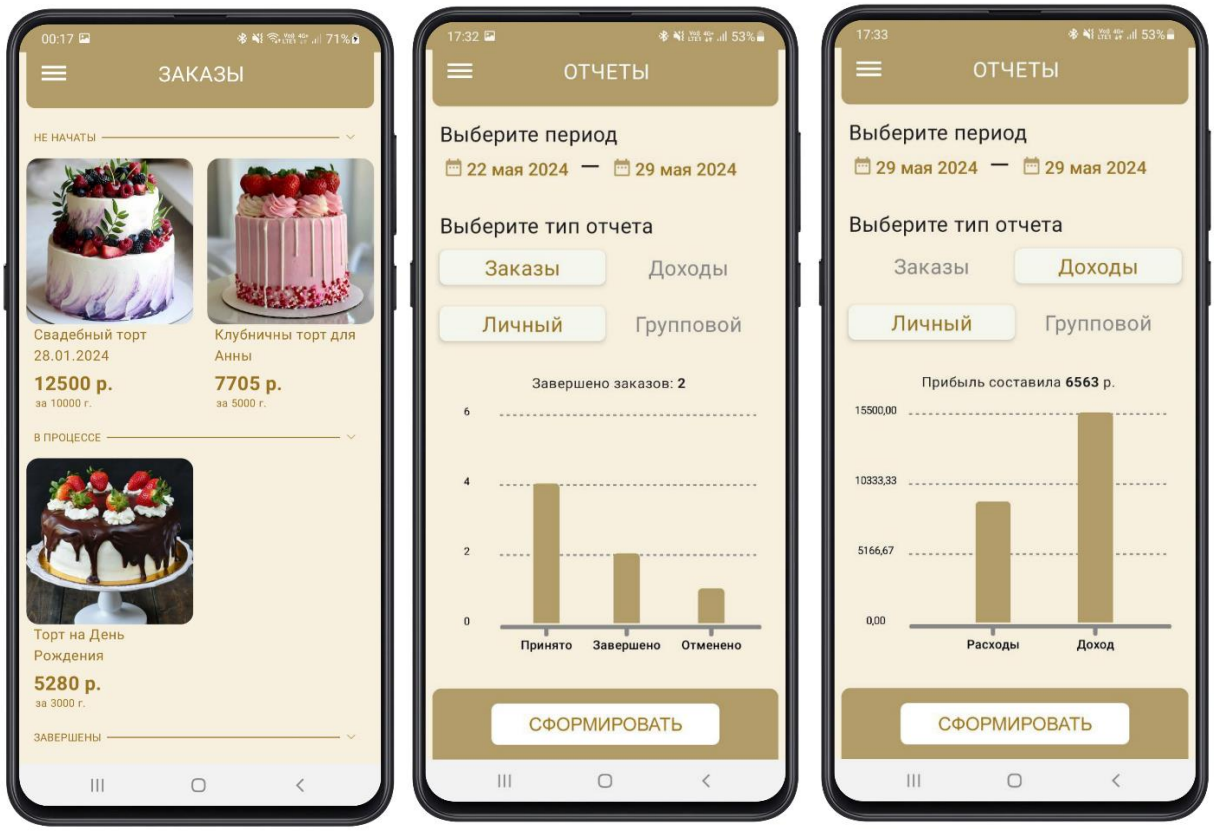

Рисунок 32 - Страницы заказов и отчетов

У пользователя есть возможность вступить в группу, а у пользователя расширенной версии есть возможность создать группу. При создании группы пользователь расширенной версии получает код, который потом могут использовать пользователи обычной версии.

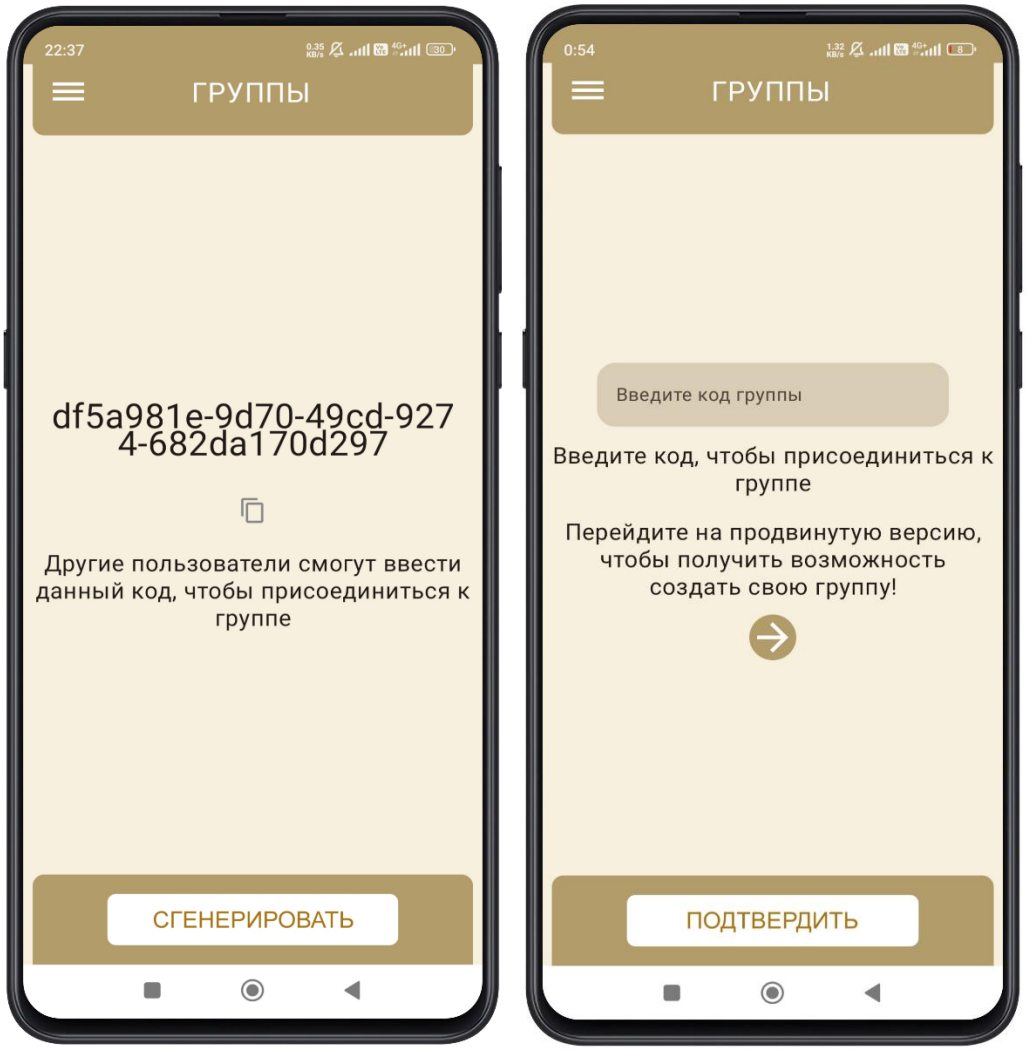

Рисунок 33 - Страница "Группы"

4.5 Методология разработки

При реализации проекта была использована каскадная модель (Waterfall) рисунок 34.

Переход между фазами возможен только после полного и успешного завершения предыдущей. Сначала полностью завершается первый этап ("Анализ требований"), в результате которого появляется полный и исчерпывающий список требований к информационной системе. Только после этого возможен переход к проектированию, в ходе которого разрабатываются дизайн, прототип и отдельные элементы приложения. Далее начинается этап реализации требований в виде программного кода системы.

После того как перечисленные этапы завершены, производится тестирование и последующая отладка продукта. Тут устраняются все недочеты и ошибки, появившиеся ранее. После этого программный продукт внедряется и осуществляется его поддержка - разработка новой функциональности, устранение ошибок и т.д.

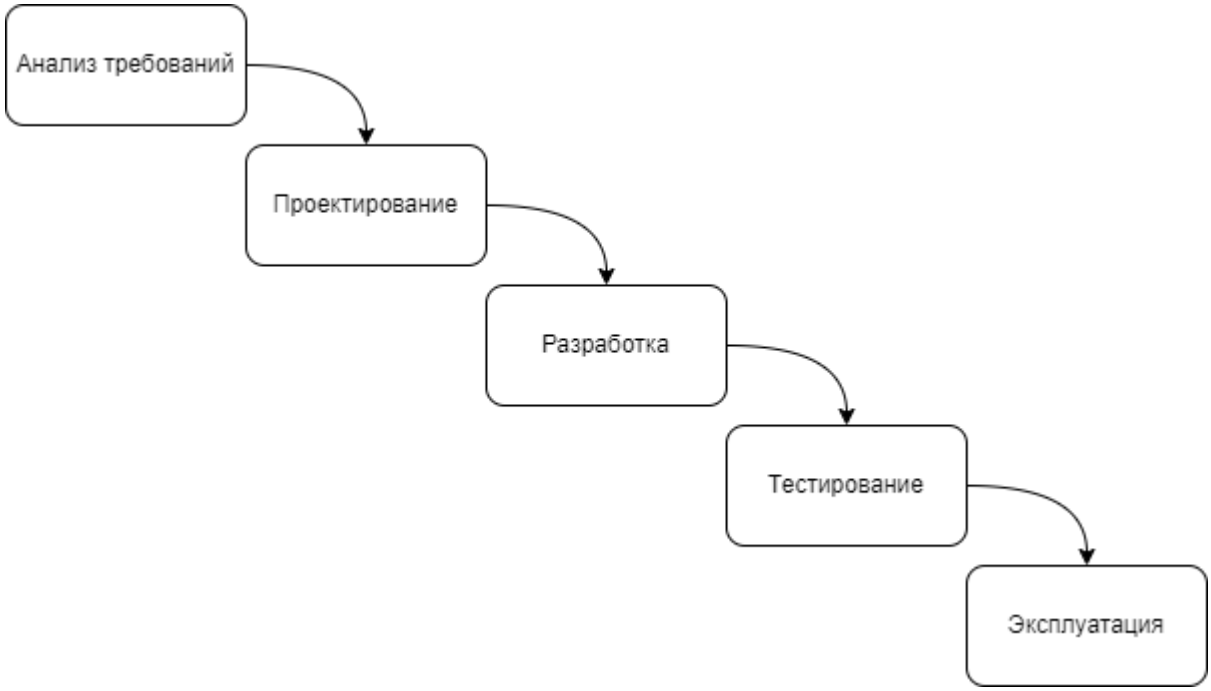

Рисунок 34 - Каскадная модель

5 Тестирование

В разделе находится информация о тестировании системы.

5.1 Дымовое тестирование

Для дымового тестирования необходимо проверить работоспособность основных сценариев. Основные сценарии:

— регистрация;

— авторизация;

— открытие бокового меню;

— добавление нового ингредиента;

— добавление готового изделия;

— проведение расчета стоимости.

В таблице 3 представлены результаты тестирования. Тестирование производилось на Samsung Galaxy A50.

Таблица 3 - Результаты тестирования

Сценарий	**Результат**
Регистрация	Пройден
Авторизация	Пройден
Открытие бокового меню	Пройден
Добавление нового ингредиента	Пройден
Добавление готового изделия	Пройден
Проведение расчета стоимости	Пройден

5.2 UI-тестирование

В таблице приведены результаты UI-тестирования.

Таблица 4 - результаты UI-тестирования

Шаги теста	**Ожидаемый ответ**	**Результат**
Нажатие кнопки «войти» при некорректных входных данных	Вход не удался	Пройден
Нажатие кнопки «войти» при корректных входных данных	Открытие главной страницы	Пройден
Нажатие кнопки «зарегистрироваться» при корректных данных	Открытие страницы входа	Пройден
Открытие бокового меню если пользователь авторизован	Открытие бокового меню	Пройден
Нажатие на кнопку «Главная страница»	Откроется главная страница	Пройден
Нажатие на кнопку «Ингредиенты»	Отроется страница ингредиентов	Пройден
Нажатие на кнопку «Готовые изделия»	Откроется страница готовых изделий	Пройден
Нажатие на кнопку «Издержки»	Откроется страница	Пройден
Нажатие на кнопку «Расчет стоимости»	Откроется страница расчета стоимости	Пройден

Шаги теста	Ожидаемый ответ	Результат
Нажатие на кнопку «Заказы»	Откроется страница заказов	Пройден
Нажатие на кнопку «Отчеты»	Откроется страница отчетов	Пройден
Нажатие на кнопку «Группы»	Откроется страница групп	Пройден
Нажатие на кнопку «Рассчитать» при корректно введенных данных	Выведется результат расчета	Пройден

В результате все сценарии UI-тестирования были пройдены успешно.

Заключение

В ходе реализации приложения были достигнуты поставленные задачи. Разработанное приложение выполняет поставленные задачи, а именно:

— регистрация и авторизация;

— просмотр информации о приложении;

— управление заказами клиентов, включая создание новых заказов и отслеживание их статуса;

— управление ингредиентами, необходимыми для рецептов, включая добавление, редактирование и удаление ингредиентов;

— создание шаблонов изделий на основе имеющихся продуктов;

— расчет стоимости готовых изделий с учетом издержек на их производство;

— генерация отчетов по доходам или заказам;

— присоединение к группе;

— создание группы;

— просмотр общих оперативных отчетов по всей группе.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32. Система стандартов по информации, библиотечному и издательскому делу. Структура и правила оформления: [сайт]. – URL: https://cs.msu.ru/sites/cmc/files/docs/2021-11gost_7.32-2017.pdf – Текст: электронный.

2. В.С. Тарасов, Д.И. Шмойлов, А.В. Москаленко «Методическое пособие Проектирование технического задания». - Воронеж Издательский дом ВГУ, 2024 – Текст: электронный.

3. Документация Spring: [сайт]. – URL: https://docs.spring.io/spring-framework/reference/index.html – Текст: электронный.

4. Система контроля версий Git: [сайт]. – URL: https://ru.wikipedia.org/wiki/Git – Текст: электронный.

5. Документация Swagger: [сайт]. – URL: https://swagger.io/solutions/api-documentation/ – Текст: электронный.

6. Документация PostgreSQL: [сайт]. – URL: https://postgrespro.ru/docs/postgresql/16/index – Текст: электронный.

7. Документация MinIO: [сайт]. – URL: https://min.io/docs/minio/windows/index.html – Текст: электронный.

8. Документация docker: [сайт]. – URL: https://docs.docker.com/ – Текст: электронный.

9. Документация Kotlin: [сайт]. – URL: https://kotlinlang.org/docs/home.html – Текст: электронный.