Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

институт

Прикладная информатика

кафедра

**КУРСОВАЯ РАБОТА**

ИС «Закупка запчастей»

тема (вариант)

Преподаватель \_\_\_\_\_\_\_\_\_\_ В.С. Васильев

подпись, дата инициалы, фамилия

Студент КИ23-20Б 32320492 \_\_\_\_\_\_\_\_\_\_ А.В. Илющенко

номер группы номер зачетной книжки подпись, дата инициалы, фамилия

г. Красноярск, 2025

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#__RefHeading___Toc485_4108303272)

[1 Спецификация требований 4](#__RefHeading___Toc487_4108303272)

[1.1 Описание и анализ текущих бизнес-процессов 4](#__RefHeading___Toc489_4108303272)

[1.2 Требования к системе 5](#__RefHeading___Toc491_4108303272)

[1.3 Форматы сущностей 15](#__RefHeading___Toc489_4108303272_Copy_1)

[2 Проектирование 16](#__RefHeading___Toc487_4108303272_Copy_2)

[2.1 Прецедент «Добавить товар в заказ» 16](#__RefHeading___Toc487_4108303272_Copy_1)

[2.2 Прецедент «Обработать заказ Администратора» 17](#__RefHeading___Toc487_4108303272_Copy_1_)

[2.3 Прецедент «Отправить заказ» 18](#__RefHeading___Toc487_4108303272_Copy_11)

[2.4 Диаграмма классов 19](#__RefHeading___Toc487_4108303272_Copy_12)

ВВЕДЕНИЕ

Во время работы магазина продающего запчасти нужно каждый день пополнять запасы товара. Запасы пополняются с помощью поставщиков. Эти поставщики отправляют курьера и те в свою очередь договариваются о поставках с продавцом, через день курьер приезжает с товаром. Из-за этого возникает множество неудобств при заказе деталей. Для решения этих проблем предлагается спроектировать и разработать систему, в которой администраторы магазинов могли бы договариваться о поставках товара с поставщиками без участия курьеров

**Целью** курсовой работы является проектирование и разработка информационной системы «СПАД — Система Покупки Авто-Деталей», позволяющей автоматизировать процесс заказа деталей. Система делается для огранизации взаимодействия администратора магазина и поставщика.

1 Спецификация требований

1.1 Описание и анализ текущих бизнес-процессов

На данный момент процесс управления закупкой деталей в магазине запчастей осуществляется вручную. Администратор магазина на бумаге создаёт список деталей, которые передаёт курьеру поставщика. После этого курьер доставляет заказ поставщику. На следующий день поставщик собирает заказ и отправляет своего курьера отправить его. И администратору магазина и поставщику приходится вручную изменять свои базы данных при работе друг с другом.

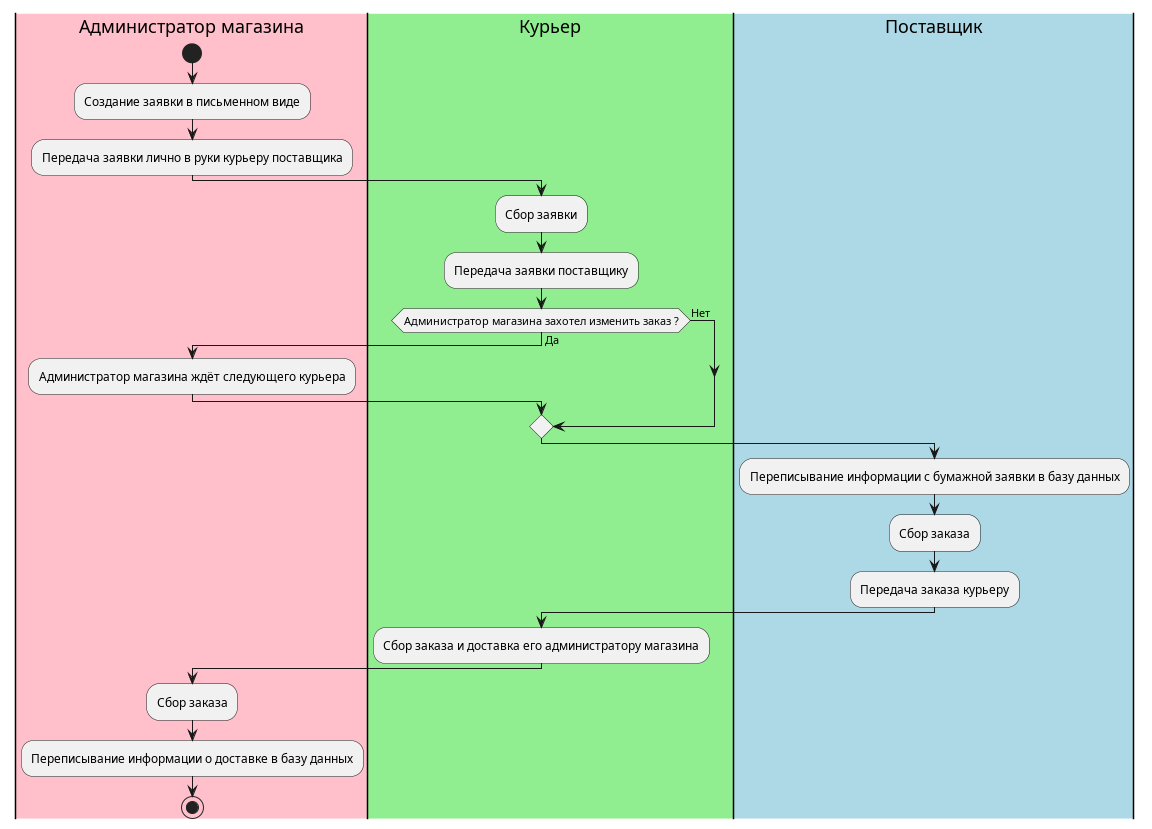


Рисунок 1.1 – Диаграмма деятельности существующего бизнес-процесса

1.2 Требования к системе

Диаграмма прецедентов для Администратора магазина и Поставщика (рисунок 1.2)

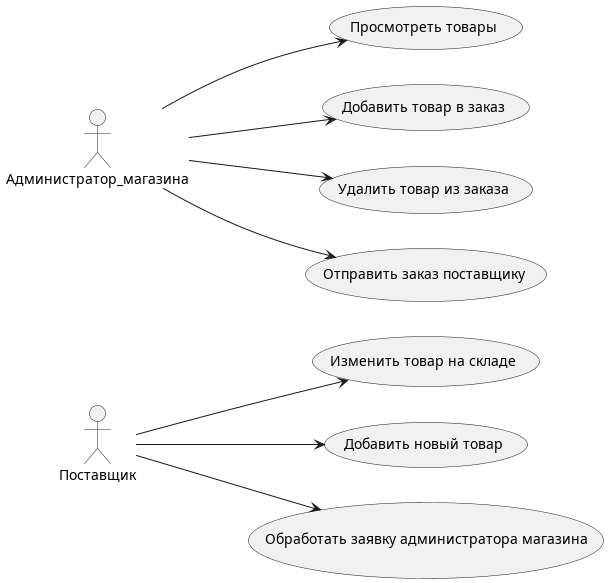


Рисунок 1.2 – Диаграмма прецедентов

Диаграммы интерфейсов для Администратора магазина и Поставщика показывают их работу в ИС с использованием графического интерфейса (user-flow) (рисунки 1.3 – 1.4).

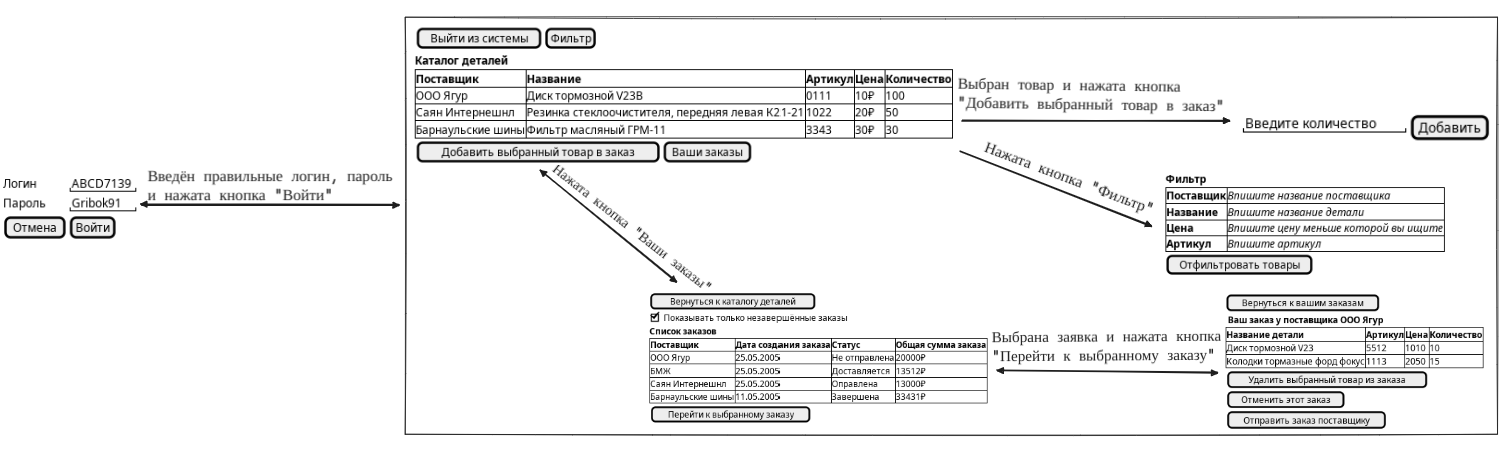


Рисунок 1.3 – Диаграмма интерфейсов Администратора магазина

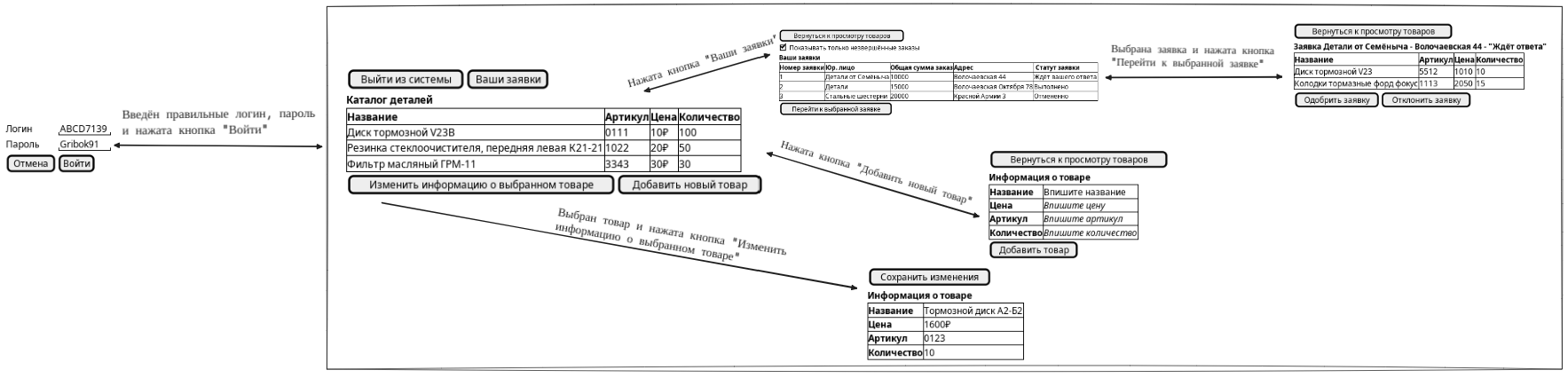


Рисунок 1.4 – Диаграмма интерфейсов Поставщика

**Прецедент** Просмотреть товары

Действующее лицо: Администратор магазина

Цель: Просмотреть доступные для покупки товары

Предусловия: Администратор вошел в систему

Главная последовательность:

1. Администратор находится в окне каталога деталей (рисунок 1.5)

2. Администратор просматривает доступные для покупки детали

3. Администратор нажимает на кнопку «Фильтр» для более точного поиска

4. Появляется всплывающее окно фильтра, в котором Администратор может вписывать различные значения для характеристик деталей (рисунок 1.6)

5. Администратор нажимает на кнопку «Отфильтровать товары»

Постусловие: после нажатия кнопки «Отфильтровать товары» окно фильтра закрывается и Администратор магазина видит только товары соответсвующие условиям фильтрации. Администратор остается в окне просмотра товаров.

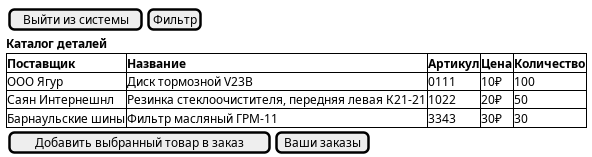


Рисунок 1.5 – Окно просмотра товаров

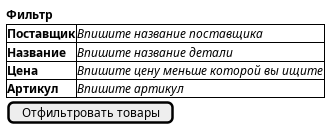


Рисунок 1.6 – Всплывающее окно фильтра

**Прецедент** Добавить товар в заказ

Действующее лицо: Администратор магазина

Цель: Добавить детали которые Администратор хочет приобрести в заказ, для дальнейшей покупки

Предусловия: Администратор вошел в систему

Главная последовательность:

1. Администратор находится в окне каталога деталей

2. Администратор выбирает в таблице товара тот который он хочет приобрести

3. Администратор нажимает на кнопку «Добавить товар в заказ»

4. Появляется всплывающее окно выбора количества товара, в котором Администратор может вписывать то количество товара которое ему нужно приобрести, не превышающее количество товара доступного для продажи (рисунок 1.7)

5. Администратор вводит необходимое количество товара в текстовое поле

6. Администратор нажимает на кнопку «Добавить»

Постусловие: после нажатия кнопки «Добавить» окно выбора количества товара закрывается. Информация о добавленном товаре перенесена в раздел «Ваши заказы». Администратор остается в окне просмотра товаров.

Альтернативная последовательность:

1. Администратор не вводит количество необходимого товара в всплывающем окне выбора количества товара или вводит число превосходящее количество доступного для продажи товара

Постусловие: Всплывающее окне выбора количества товара не закрывается и вместо введённого количества появляется надпись «Введите коректное количество товара»



Рисунок 1.7 – Всплывающие окно выбора количества товара

**Прецедент** Удалить товар из заказа

Действующее лицо: Администратор магазина

Цель: Убрать товар который Администратор добавил в заказ ранее

Предусловия: Администратор вошел в систему и добавил товар в заказ

Главная последовательность:

1. Администратор находится в окне каталога деталей

2. Администратор нажимает на кнопку «Ваши заказы»

3. Администратора переносит в окно заказов, в котором он может просмотреть созданные им ранее заказы (рисунок 1.8). Все добавленные в заказ детали групируются по поставщику. Таким образом один заказ соответсвует только одному поставщику

4. Администратор выбирает заказ со статусом «Не отправлена»

5. Администратор нажимает на кнопку «Перейти к выбранному заказу»

6. Администратора переносит в окно выбранного заказа, в которой он может видеть добавленные детали, поставщика у которого совершается заказ и кнопки управления заказом (рисунок 1.9)

7. Администратор выбирает товар который хочет удалить

8. Администратор нажимает на кнопку «Удалить выбранный товар из заказа»

Постусловие: Из выбраного заказа пропадает выбранный товар



Рисунок 1.8 – Окно заказов

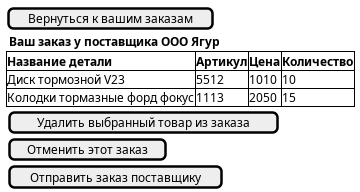


Рисунок 1.9 – Окно выбранного заказа

**Прецедент** Отправить заказ поставщику

Действующее лицо: Администратор магазина

Цель: Отправить сформированный заказ Поставщику, для того что бы Поставщик в последствии одобрил его и привёз заказанные детали

Предусловия: Администратор вошёл в систему, добавил товары в заказ и нажал вошёл в окно заказов

Главная последовательность:

1. Администратор находится в окне заказов

2. Администратор выбирает заказ

3. Администратор нажимает на кнопку «Перейти к выбранному заказу»

4. Администратор нажимает на кнопку «Отправить заказ поставщику»

Постусловие: После нажатия кнопки «Отправить заказ поставщику» заказ передается поставщику и добавляется в базы данных заказов Администратора магазина и Поставщика. Администратора переносит в окно заказов. Переданный заказ меняет статус на «Ожидает ответа»

Альтернативная последовательность:

1. Администратор не нажимает на кнопку «Отправить заказ поставщику», а вместо этого нажимает на кнопку «Отменить этот заказ»

Постусловие: Заказ удаляется из окна «Ваши заказы». Администратора переносит в окно заказов

**Прецедент** Изменить товар на складе

Действующее лицо: Поставщик

Цель: Изменить информацию о товаре на складе поставщика

Предусловия: Поставщик вошел в систему

Главная последовательность:

1. Поставщик находится в окне каталога деталей (рисунок 1.10)

2. Поставщик выбирает товар в таблице товара

3. Поставщик нажимает на кнопку «Информация о товаре»

4. Появляется всплывающее окно информации о товаре, в котором Поставщик может изменить информацию о товаре, вписав новые значения вместо старых (рисунок 1.11)

5. Поставщик изменяет значения товара (Название, Цену, Артикул или Количество)

6. Поставщик нажимает на кнопку «Сохранить изменения»

Постусловие: После нажатия кнопки «Сохранить изменения» всплывающее окно информации о товаре закрывается. Поставщик остаётся в окне каталога деталей. Изменения заносится таблицу товаров поставщика и Администраторов магазинов, кроме этого изменения заносятся в базу данных поставщика

Альтернативная последовательность:

1. Поставщик вводит некоректное значение для характеристики и нажимает на кнопку «Сохранить изменения»

Постусловие: В поле с некоректным значением значение заменятся на текст «Введите корректное значение». Поставщик остаётся в окне информации о товаре

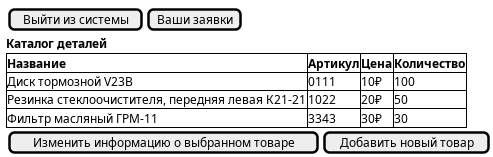


Рисунок 1.10 – Окно каталога деталей

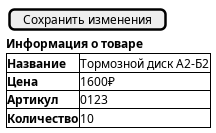


Рисунок 1.11 – Всплывающие окно информации о товаре

**Прецедент** Добавить новый товар

Действующее лицо: Поставщик

Цель: Добавить новую деталь в каталог, что бы Администраторы могли их приобретать

Предусловия: Поставщик вошёл в систему

Главная последовательность:

1. Поставщик находится в окне каталога деталей

2. Поставщик нажимает на кнопку «Добавить новый товар»

3. Появляется всплывающее окно создания нового товара, в котором Поставщик может вписать информацию о новом товаре (рисунок 1.12)

4. Поставщик нажимает на кнопку «Добавить товар»

Постусловие: После нажатия на кнопку «Добавить товар» всплывающее окно создания нового товара закрывается. Поставщик остаётся в окне каталога деталей. Новый товар добавляется в каталог деталей Администратора магазина и поставщика, а также заносится в базу данных Поставщика.

Альтернативная последовательность:

1. Поставщик вводит некоректное значение для характеристики и нажимает на кнопку «Сохранить изменения»

Постусловие: В поле с некоректным значением значение заменятся на текст «Введите корректное значение». Поставщик остаётся в окне создания нового товара

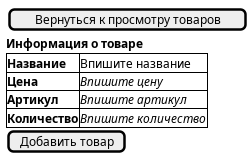


Рисунок 1.12 – Всплывающие окно создания нового товара

**Прецедент** Обработать заказ Администратора магазина

Действующее лицо: Поставщик

Цель: Одобрить или отклонить заказ Администратора и соответсвенно отправить товар Администратору магазина или нет

Предусловия: Поставщик вошел в систему

Главная последовательность:

1. Поставщик находится в окне каталога деталей

2. Поставщик нажимает на кнопку «Ваши заказы»

3. Поставщика переносит в окно заказов, в котором он может просмотреть отправленные ранее Администратором магазина заказы (рисунок 1.13).

4. Поставщик выбирает заказ со статусом «Ждёт вашего ответа»

5. Поставщик нажимает на кнопку «Перейти к выбранному заказу»

6. Поставщика переносит в окно выбранного заказа, в которой он может просмотреть детали которые у него хотят приобрести и одобрить и одобрить этот заказ или нет (рисунок 1.14).

6. Поставщик нажимает на кнопку «Одобрить заказ»

Постусловие: после нажатия кнопки «Одобрить заказ» Поставщика перекидывает в окно заказ. У Администратора магазина изменяется статус заказа в окне заказов. Это значит что Администратор ожидает доставку.

Альтернативная последовательность:

1. Поставщик вместо кнопки «Одобрить заказ» нажимает «Отклонить заказ»

Постусловие: после нажатия кнопки «Отклонить заказ» Поставщика перекидывает в окно заказов. У Администратора магазина изменяется статус заказа в окне заказов. Это значит что Администратор не ожидает доставку.

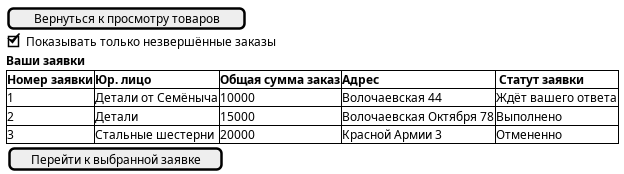


Рисунок 1.13 – Окно заказов



Рисунок 1.14 – Окно выбранного заказ

1.3 Форматы сущностей

Для этого используется PostgreSQL база данных с таблицами, представленными ниже:

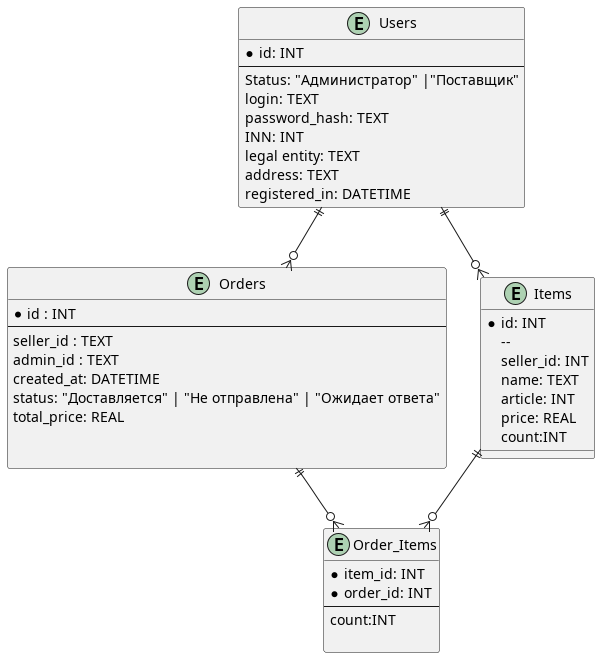


Рисунок 1.15 – Окно выбранного заказ

2 Проектирование

2.1 Прецедент «Добавить товар в заказ»

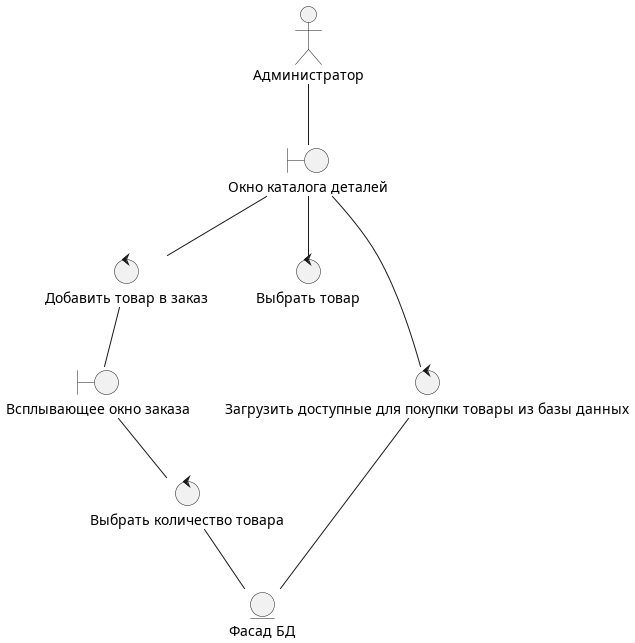


Рисунок 2.1– Диаграмма пригодности «Добавить в заказ»

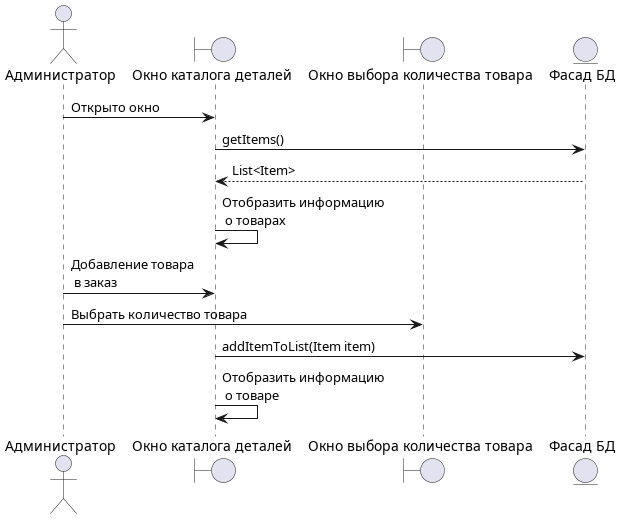


Рисунок 2.2 – Диаграмма последовательности «Добавить в заказ»

2.2 Прецедент «Обработать заказ Администратора»

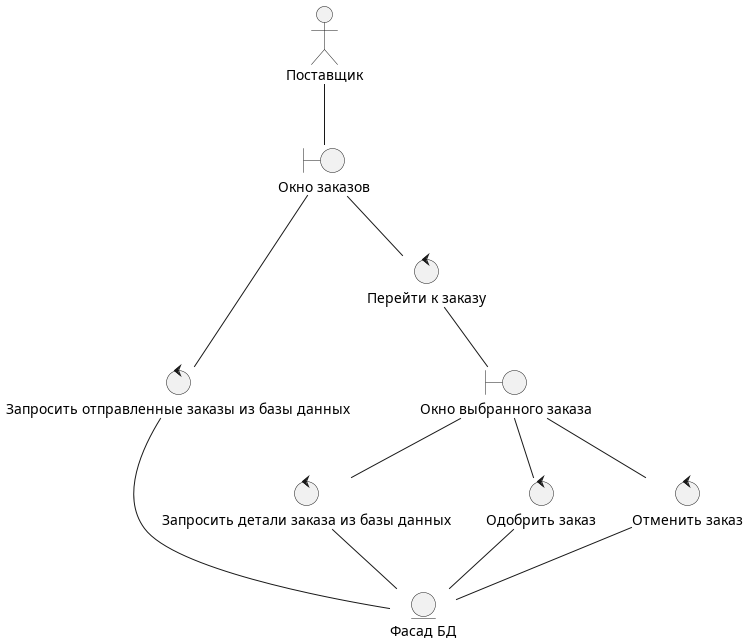


Рисунок 2.3– Диаграмма пригодности «Обработать заказ Администратора»

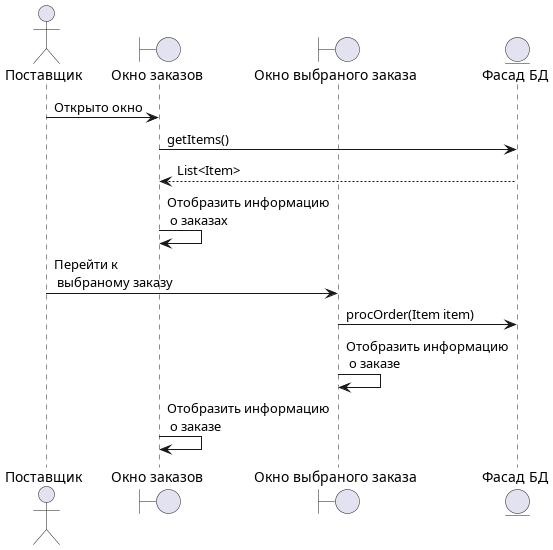


Рисунок 2.4 – Диаграмма последовательности «Обработать заказ Администратора»

2.3 Прецедент «Отправить заказ»

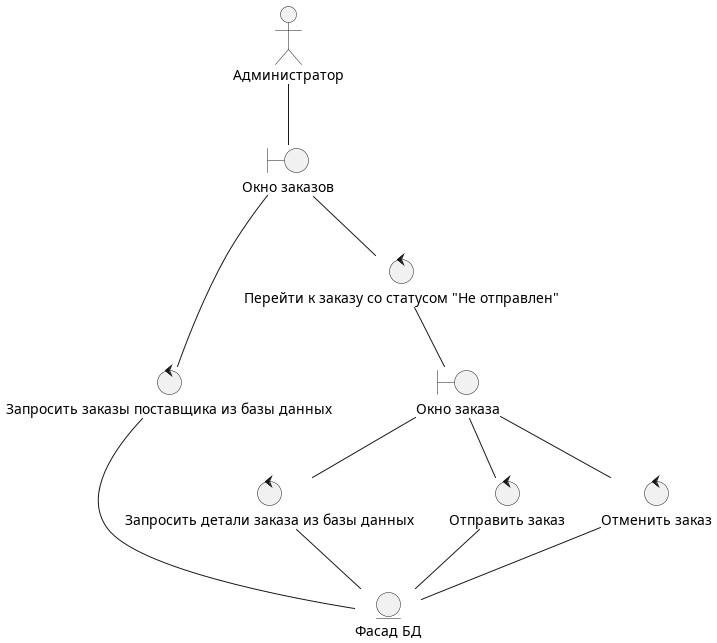


Рисунок 2.5– Диаграмма пригодности «Отправить заказ»

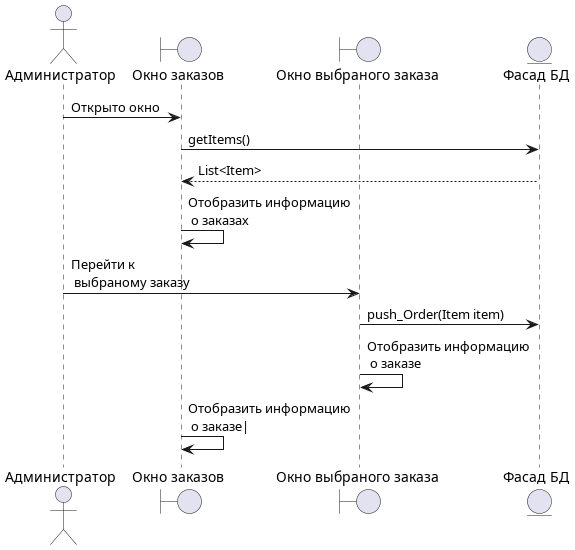


Рисунок 2.6 – Диаграмма последовательности «Отправить заказ»

2.4 Диаграмма классов

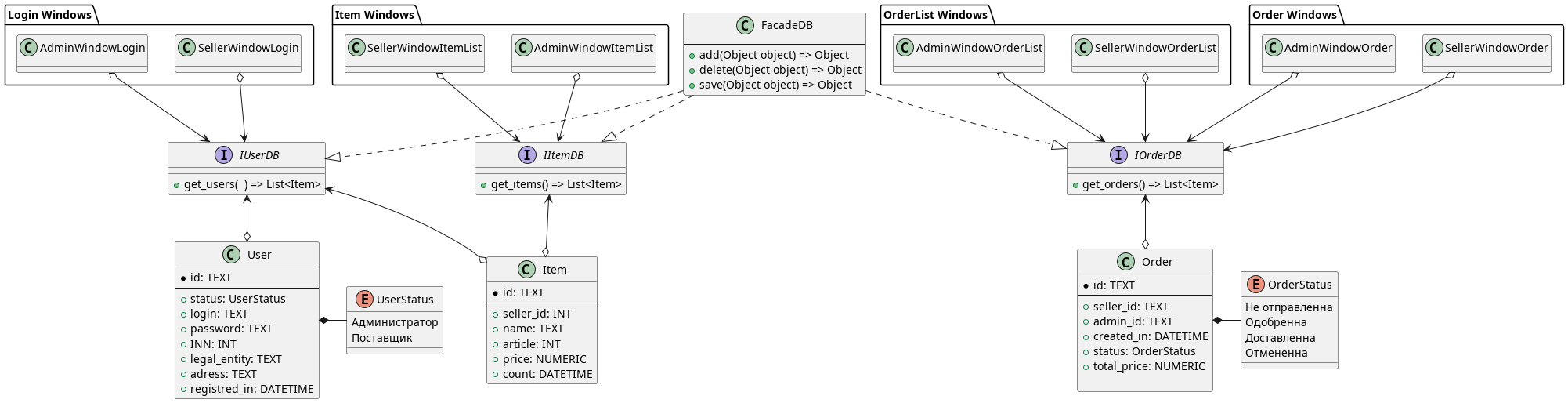


Рисунок 2.7 – Диаграмма классов

3 Разработка и тестирование

3.1 Работа с данными

Подключение к базе происходит внутри файла main.py который обращается к файлу config.yml и берёт от туда строку подключения. Пример содержимого config.yml:

*path: 'postgresql+psycopg2://postgres:password@localhost/name\_of\_base'*

Подключение к базе просходит от лица суперпользователя. Были созданы интерфейсы для всех классов данных: пользователь, товар, заказ, товар в заказе. Эти интерфейся используются интерфейсами-фасадами для доступа к данным в базе. Один класс базы реализует все интерфейсы доступа к данным, он содержит в себе обращения к базе.

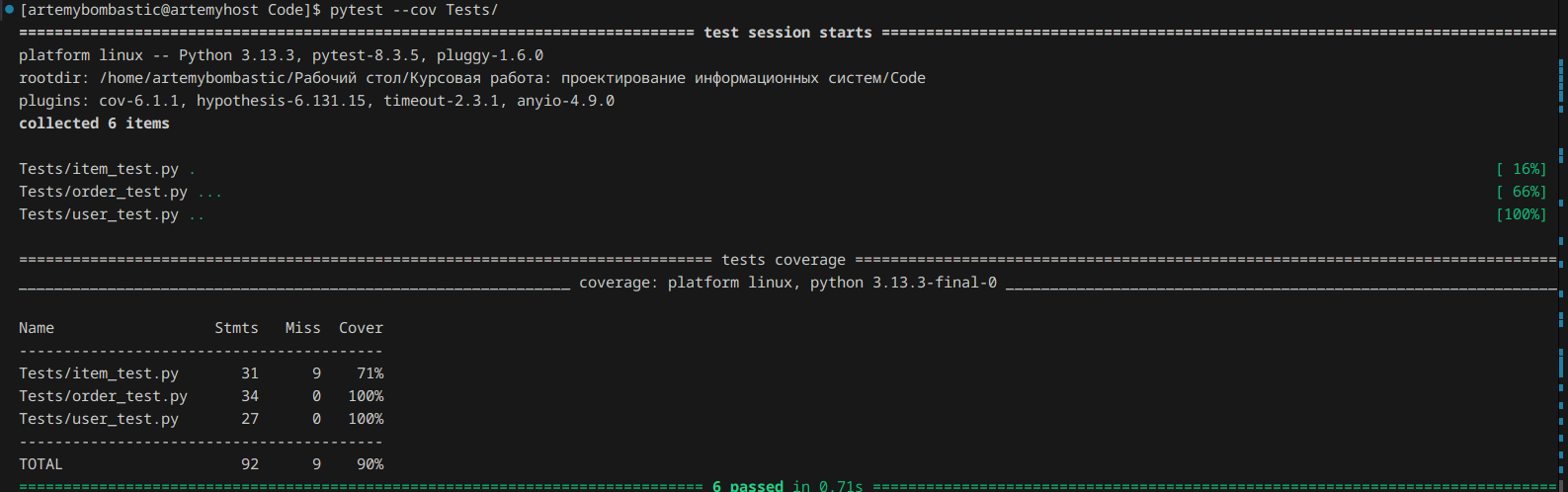
При изменении существующих таблиц нужно поменять классы моделей, если добавляется новая таблица, то следует добавить новую модель, а также добавить эту таблицу в DBFacade.

Для начала работы с ИС создадутся все необходимые базы. Описание таблиц при помощи sqlalchemy приведено в приложении А.

3.2 Тестирование

Для проведения модульных тестов в моём проекте был создан подпроект специально для тестов, тесты реализованны с помощью Pytest[7].

На рисунке 3.1 процент покрытия посчитанный при помощи Pytest Coverage[8]

Рисунок 3.1 – Процент покрытия кода

3.3 Развёртывание проекта

Для развёртывания проекта нужно склонировать репозиторий [9], любым из удобных способов создать пустую базу данных postgres и yml файл с строкой подключение к базе Postgres, далее выполнить несколько скриптов:

1)Скачать зависимости : pip install -r requirements.txt

2)Запустить Controller.py: python Controller.py

Заключение

В рамках работы была спроектированна и реализованна программа, позволяющая работать с базой данных при помощи интерфейса, представленного в виде оконного приложения [9]. Данная программа является крайне упрощённой, например в этом приложении мы не можем зарегестрироваться и пароли хранятятся в незашифрованном виде, что в реальных приложениях встречается крайне редко. Данную программу можно доработать, поменяв подключение базы данных на удалённое. Работа подтверждает получение навыков проектирования, создания модульных тестов, работу с базами данных, работу с ООП.

**Список используемых источников**

1. Нотации модели сущность-связь (ER диаграммы) [Электронный ресурс]. – https://pro-prof.com/archives/8126 (дата обращения 15.05.2025)

2. Создание проекта форм интерфейса и карты диалоговых окон в PLANTUML [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/279373/ (дата обращения 15.05.2025)

3. Процесс ICONIX. Диаграммы пригодности [Электронный ресурс]. – https://pro-prof.com/archives/2723 (дата обращения 15.05.2025)

4. Пример описания многослойной архитектуры, основанной на использовании наборов подслоёв и иерархии моделей данных [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/667922/ (дата обращения 15.05.2025)

5. PostgreSQL [Электронный ресурс]. – Режим доступа: https://www.postgresql.org/ (дата обращения 15.05.2025)

6. SQLAlchemy [Электронный ресурс]. – Режим доступа: https://www.sqlalchemy.org/ (дата обращения 15.05.2025)

7. Pytest [Электронный ресурс]. – Режим доступа: https://docs.pytest.org (дата обращения 15.05.2025)

8. Vitest Coverate [Электронный ресурс]. – Режим доступа: https://pytest-cov.readthedocs.io/en/latest/readme.html (дата обращения 15.05.2025)

9. Репозиторий GitHub [Электронный ресурс]. – Режим доступа: https://github.com/BombasticFantastik/PIS\_Course\_work (дата обращения 15.05.2025)

10. Pip [Электронный ресурс]. – Режим доступа: https://pypi.org/project/pip/ (дата обращения 16.05.2025)

**ПРИЛОЖЕНИЕ А**

**Описание таблиц при помощи sqlalchemy.**

from sqlalchemy import Column,Integer,String,Float,Date,BigInteger

from sqlalchemy.orm import declarative\_base

Base=declarative\_base()

class Item(Base):

\_\_tablename\_\_='items'

id=Column(Integer,autoincrement=True,primary\_key=True)

seller\_id=Column(Integer)

name=Column(String)

article=Column(Integer)

price=Column(Float)

count=Column(Integer)

class User(Base):

\_\_tablename\_\_='users'

id=Column(Integer,primary\_key=True)

status=Column(String)

login=Column(String)

password=Column(String)

INN=Column(BigInteger)

legal\_entity=Column(String)

address=Column(String)

registred\_in=Column(Date)

class Order(Base):

\_\_tablename\_\_='orders'

id=Column(Integer,primary\_key=True)

seller\_id=Column(Integer)

admin\_id=Column(Integer)

created\_in=Column(Date)

status=Column(String)

total\_price=Column(Float)

class Order\_Item(Base):

\_\_tablename\_\_='order\_items'

id=Column(Integer,autoincrement=True,primary\_key=True)

item\_id=Column(Integer)

order\_id=Column(Integer)

count=Column(Integer)