



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 1

*По предмету: «Проектирование программного
обеспечения»*

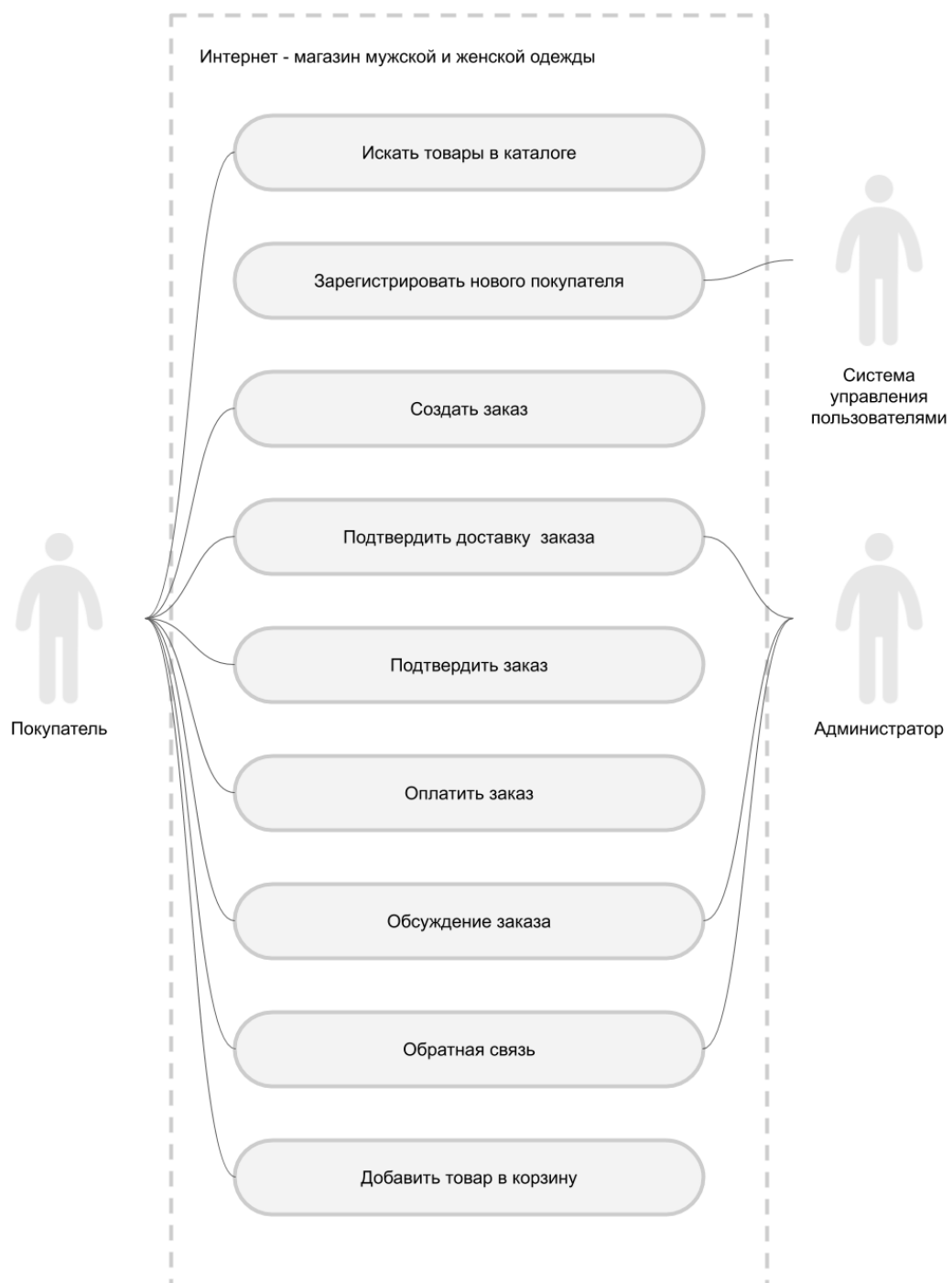
Студент: Юмаев Артур Русланович
Группа: ИУ7-65Б

Москва, 2020 г.

Оглавление

1. Use – Case диаграмма.....	3
2. ER – диаграмма сущностей	4
3. Технологический стек.....	5
4. Структурная схема программы.....	6
5. Программная реализация компонента доступа к данным.....	7

1. Use – Case диаграмма



2. ER – диаграмма сущностей

На данный момент база данных состоит из трех таблиц (рис. 1), в которых хранятся данные о доступном каталоге товаров, а также таблица с корзиной для каждого пользователя. Так как проект в стадии разработки, также будут добавлены таблицы для заказов клиентов с указанием стадии заказа, а также таблица с компаниями, осуществляющими доставку заказа клиенту.

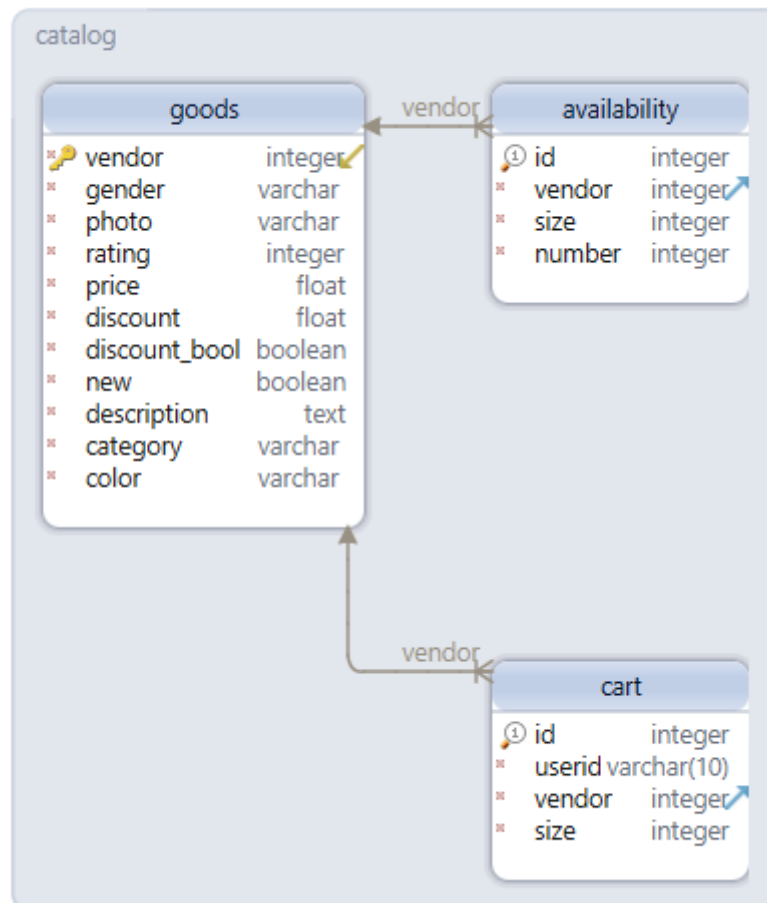


Рисунок 1. ER Диаграмма сущностей

3. Технологический стек

Используемые технологии и подходы

Бэкенд

- Язык программирования Python
- Фреймворк Flask для быстрой реализации простых одностраничных веб приложений на модели MVC (Model – View – Controller)
- Apache HTTP-сервер – кроссплатформенный веб сервер для обработки соединений с юзер-агентом
- SQLite реляционная база данных
- Система контроля версий Git (GitHub)
- Docker — разработка, развертывание на сервере
- Microsoft Azure – облачный сервер с Ubuntu 16.04

Фронтенд

- Верстка макета с помощью HTML, CSS, JavaScript
- Twitter Bootstrap для быстрой и адаптивной разработки интерфейсов

4. Структурная схема программы

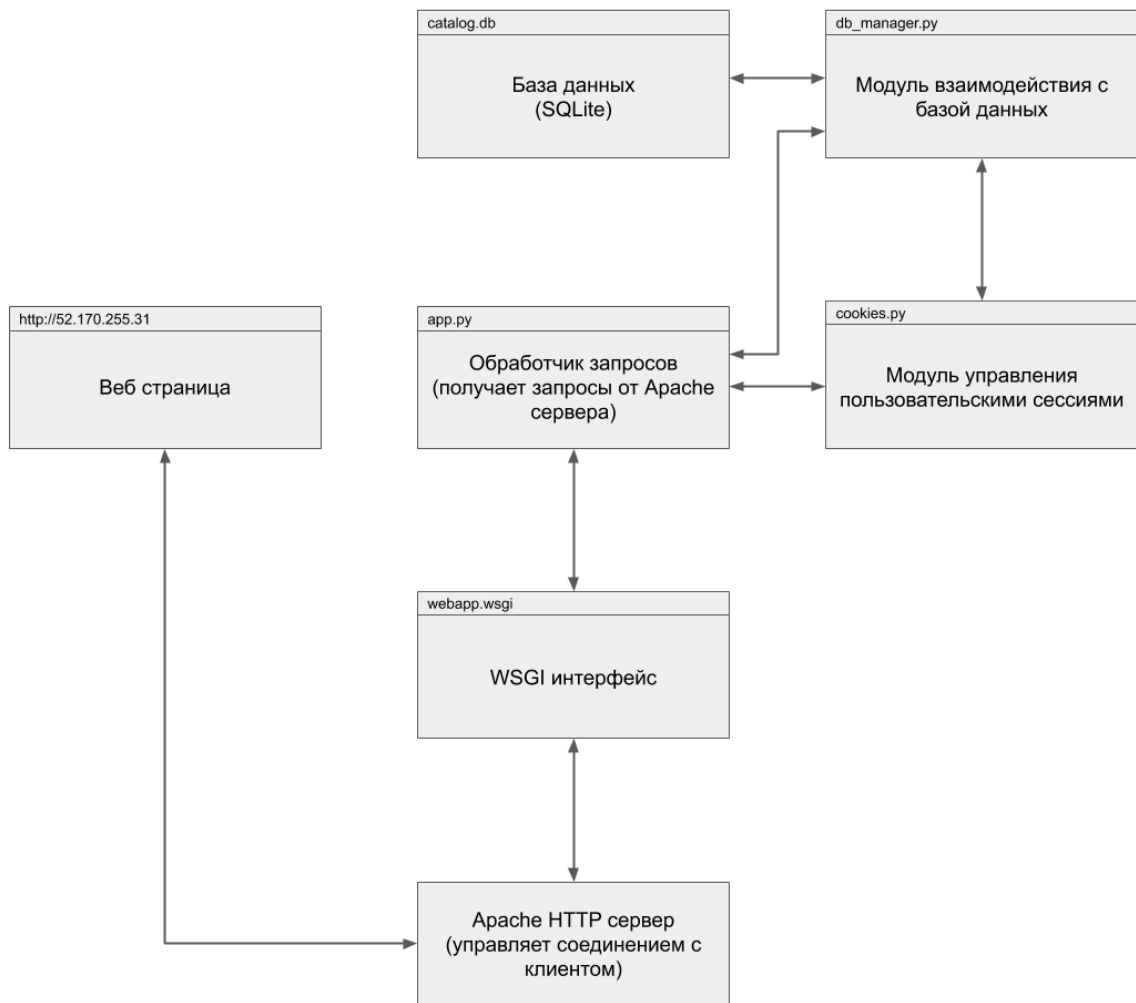


Рисунок 2. Структурная схема программы

5. Программная реализация компонента доступа к данным

В качестве программной реализации доступа к данным был выбран фреймворк `sqlite3` версии 3.8.2 для Python 3.7.1. так как был опыт в использовании. Реализация доступа к базе данных указана на листингах 1-3.

Листинг 1. Подключение библиотеки `sqlite3` и указание пути базы данных

```
import sqlite3

DEV_MODE = 1
DB_PATH = './database/catalog.db' if DEV_MODE else \
          '/var/www/webApp/webApp/database/catalog.db'
```

Листинг 2. Функция добавления товара в пользовательскую корзину

```
def insert_into_cart(userid, vendor, size):
    conn = sqlite3.connect(DB_PATH)
    c = conn.cursor()
    sql_query = 'insert into cart (userid, vendor, size) values '
    sql_query += '(\'{ }\', { }, { });'.format(userid, vendor, size)
    c.execute(sql_query)
    conn.commit()

    sql_query = "select count(userid) from cart where userid = \
                  '{ }';".format(userid)
    goods_amount = c.execute(sql_query)
    goods_amount = [i for i in goods_amount][0][0]
    conn.close()

    return goods_amount
```

Листинг 3. Функция получения количества товаров в корзине

```
def get_items_amount(userid):
    conn = sqlite3.connect(DB_PATH)
    c = conn.cursor()
    goods_amount = c.execute("select count(userid) from cart \
                              where userid = '{ }';".format(userid))
    goods_amount = [i for i in goods_amount][0][0]
    conn.close()

    return goods_amount
```

На листинге 4 показана функция получения каталога товаров. В качестве аргумента она принимает объект `request`, который получает сервер от веб-клиента. В объекте `request` содержатся `url` аргумента, которые представляют собой словарь ключ - значение.

Листинг 4. Функция получения каталога товаров

```
def get_data(request):
    url_args = list(request.args.keys())
    if all(i in url_args for i in ['sort', 'gender', 'cats']):
        sort_by = request.args.get('sort')
        gender = request.args.get('gender')
        categories = request.args.getlist('cats')
    else:
        sort_by = 'Sort by'
        gender = 'forall'
        categories = 'None'

    data = make_sql_query(gender, sort_by, categories)

    return data
```

Листинг 5. Функция получения корзины пользователя

```
def select_cart(userid):
    sql_query = """
    select *, count(*) as amount from (
        select g.description,
               g.category,
               g.gender,
               c.size,
               g.color,
               g.price * (1 - g.discount) as price,
               c.vendor
        from cart c
        join goods g
        on c.vendor = g.vendor
        where userid = '{0}'
    ) as items
    group by vendor, size
    having count(*) >= 1
    """.format(userid)

    conn = sqlite3.connect(DB_PATH)
    c = conn.cursor()
    data = [i for i in c.execute(sql_query)]
    conn.close()

    return data
```