



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

ФАКУЛЬТЕТ «Робототехники и комплексной автоматизации»
КАФЕДРА «Системы автоматизированного проектирования (РК-6)»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОЙ РАБОТЕ
по дисциплине «Базы данных»

Студент:	Гавриш Александр Александрович
Группа:	РК6-51Б
Тип задания:	Курсовая работа
Тема:	Разработка информационной системы аптеки

Студент

подпись, дата

Гавриш А.А.

Фамилия, И.О.

Преподаватель

подпись, дата

Пивоварова Н.В.

Фамилия, И.О.

Москва, 2023

Содержание

1	Техническое задание	4
1.1	Этап проектирования	4
1.2	Этап реализации	4
2	Описание предметной области	5
3	Описание конечных пользователей	5
4	UML-диграмма вариантов использования информационной системы	6
5	Системная диаграмма классов	7
6	Общая файловая структура	8
7	Вариант использования «Работа с запросами»	8
7.1	Карточка варианта	8
7.2	Сценарий	8
7.3	Системная диаграмма последовательности	10
7.4	Файловая структура	11
7.5	Требования к шаблонам	11
8	Вариант использования «Авторизация»	12
8.1	Карточка варианта	12
8.2	Сценарий	12
8.3	Системная диаграмма последовательности	13
8.4	Файловая структура	14
8.5	Требования к шаблонам	14
9	Вариант использования «Работа с отчетами»	15
9.1	Карточка варианта	15
9.2	Сценарий	15
9.3	Системная диаграмма последовательности	16
9.4	Файловая структура	18
9.5	Требования к шаблонам	19
10	Вариант использования «Формирование заказа»	20
10.1	Карточка варианта	20
10.2	Сценарий	20
10.3	Системная диаграмма последовательности	21
10.4	Файловая структура	23
10.5	Требования к шаблонам	23
11	Заключение	24

Курсовая работа на тему «Разработка информационной системы аптеки».

Руководитель - Пивоварова Н.В.

Консультант - Марков И.С.

Пояснительная записка содержит техническое задание с 2 этапами – этап планирования и этап реализации. В ходе работы была проведена пошаговая реализация пунктов технического задания для двух этапов. В разделах проектирование и реализация информационной системы представлены вся документация разработанной информационной системы. Работа содержит 24 листа, 15 рисунков.

1 Техническое задание

1.1 Этап проектирования

Техническое задание на этапе проектирования состоит из следующих обязательных пунктов:

1. Определить конечных пользователей будущей системы.
2. Составить UML-диаграмму вариантов использования.
3. Выделить основной вариант использования информационной системы (основной бизнес-процесс в предметной области).
4. Разработать систему авторизации пользователей ИС.
5. Разработать системную архитектуру ИС.
6. Для всех вариантов использования разработать главные успешные сценарии и расширения к ним.
7. Разработать системные UML-диаграммы последовательности для всех сценариев с использованием MVC-паттерна.
8. Разработать требования ко всем шаблонам для каждого варианта использования.
9. Разработать инфологическую модель предметной области в форме UML-диаграммы классов.
10. Разработать логическую модель будущей базы данных.

1.2 Этап реализации

Техническое задание на этап реализации должно быть следующим:

1. Реализовать разработанную на этапе проектирования информационную систему на языке Python в среде фреймворка Flask.
2. Каждый вариант использования оформить, как blueprint.
3. Доступ конечных и внешних пользователей к вариантам использования реализовать с помощью декораторов.

2 Описание предметной области

Аптека продаёт лекарства покупателям и постоянно пополняет их запас, пользуясь услугами нескольких поставщиков.

О каждом поставщике известно: уникальный шифр, название фирмы, город, дата заключения договора на поставку лекарств.

Также известен список лекарств, доступных для заказа у поставщиков.

Про каждое лекарство известно: шифр, название, группа лекарств, фирма производитель, страна производитель. Для пополнения запасов лекарств в аптеке составляется заказ. В заказе указывается поставщик, дата составления заказа и ожидаемая дата получения заказа, и его стоимость. Также предусмотрено поле для проставления даты реального получения заказа.

Каждый заказ содержит множество строк заказа. В каждой строке указано конкретное лекарство и его количество.

В базе данных аптеки хранятся также данные о имеющихся в наличии лекарствах. При этом одноимённые лекарства, полученные по разной цене, учитываются отдельно. Про каждое лекарство известно количество упаковок на фиксированную дату.

3 Описание конечных пользователей

В разрабатываемое информационной системе определены 5 видов конечных пользователей (4 внутренних и 1 внешний):

1. Директор - внутренний пользователь информационной системы, имеет доступ к просмотру отчетов.
2. Менеджер - внутренний пользователь информационной системы, имеет доступ следующим вариантам использования: создание/просмотр отчетов и выполнение запросов (может добавлять новых внутренних пользователей).
3. Товаровед - внутренний пользователь информационной системы, имеет доступ к выполнению запросов, связанных с поставщиками, поставками и складом.
4. Фармацевт - внутренний пользователь информационной системы, имеет доступ к выполнению запросов, связанных с классификацией медикаментов.
5. Покупатель - внешний пользователь информационной системы, имеет доступ к формированию заказов препаратов у аптеки.

4 UML-диаграмма вариантов использования информационной системы

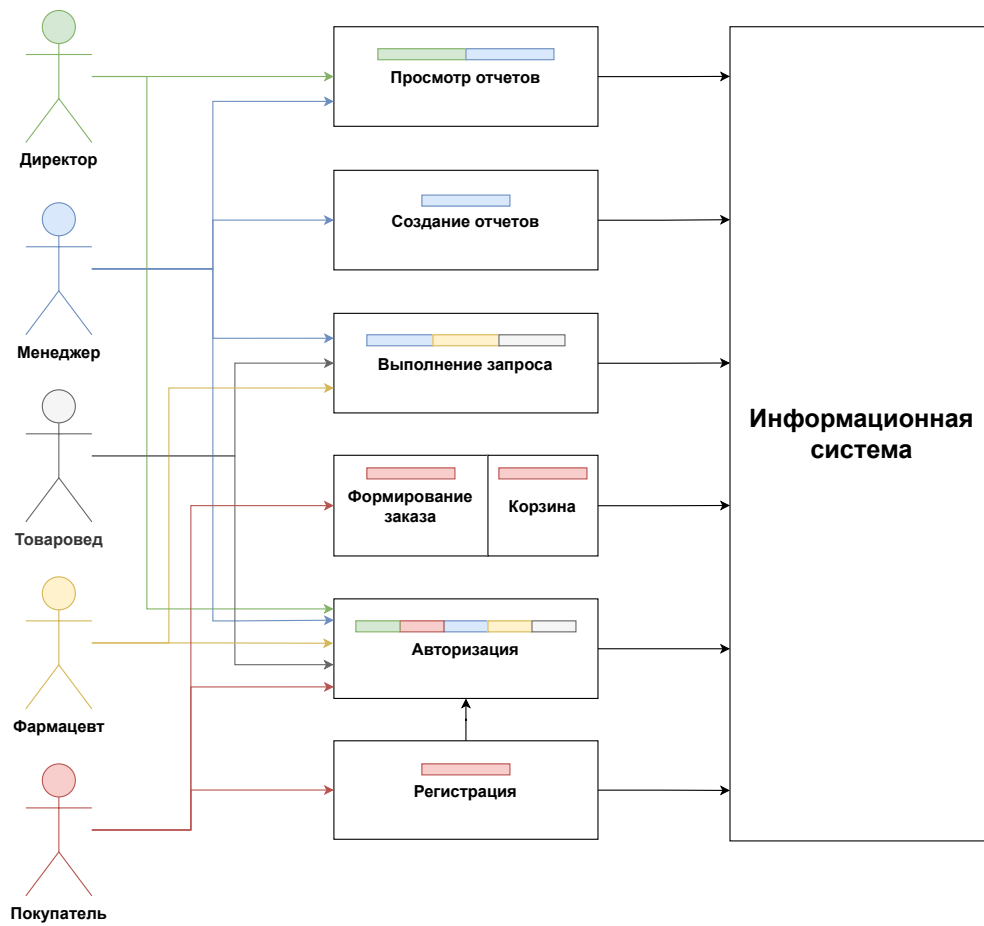


Рис. 1. UML-диаграмма вариантов использования информационной системы

5 Системная диаграмма классов

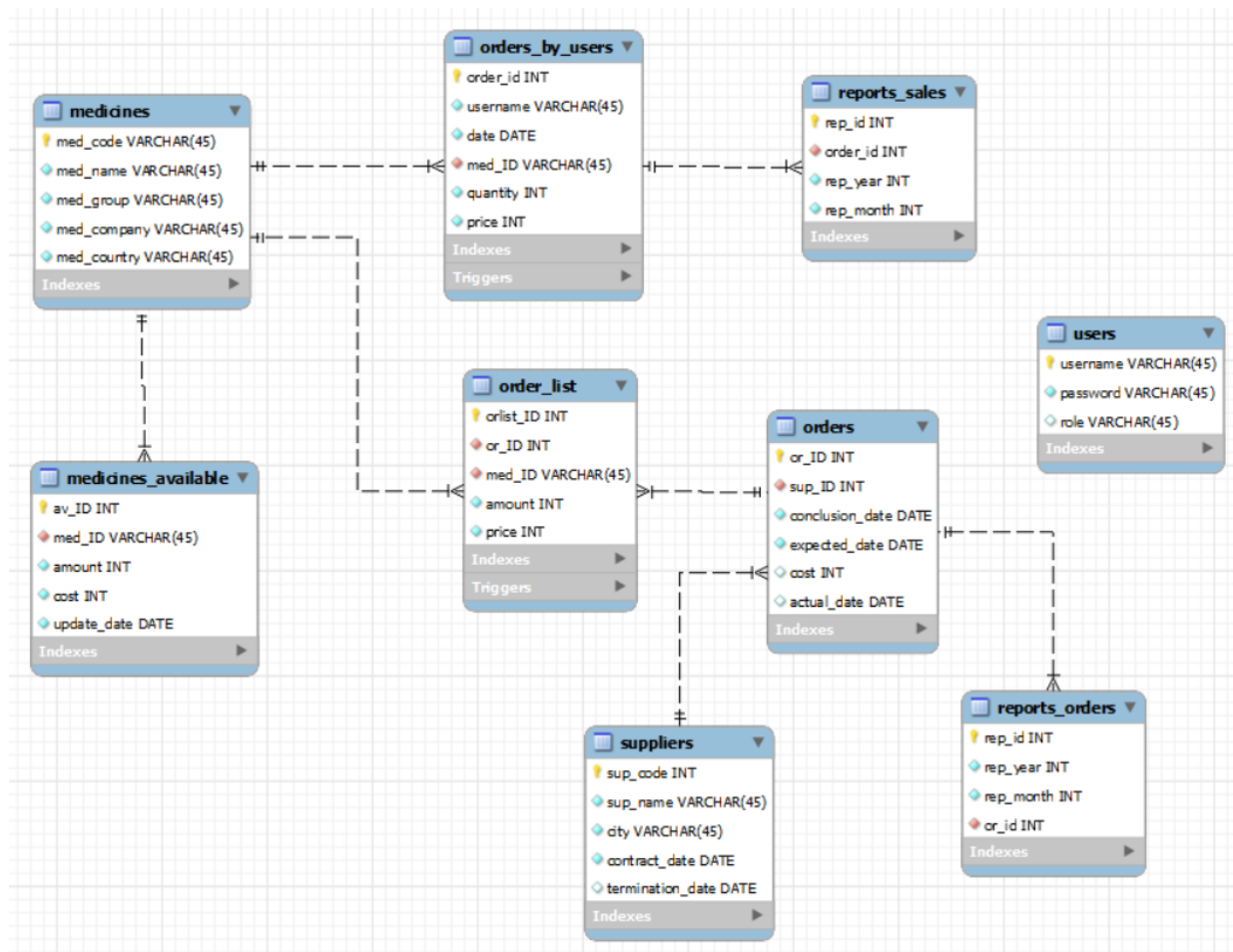


Рис. 2. Системная диаграмма классов

6 Общая файловая структура

```

Pharmacy
├─ access
├─ basket
├─ configs
│  ├─ access.json
│  ├─ db_config.json
│  └─ reports.json
├─ database
│  ├─ connection.py
│  ├─ operations.py
│  └─ sql_provider.py
├─ decorators
│  └─ routes.py
├─ queries
├─ reports
├─ static
│  ├─ css
│  │  └─ styles.css
│  ├─ images
│  │  ├─ icon.svg
│  │  └─ logo.svg
│  └─ js
│     └─ pageEventHandlers.js
├─ templates
│  ├─ header.html
│  └─ index.html
└─ app.py

```

Рис. 3. Общая файловая структура проекта

7 Вариант использования «Работа с запросами»

7.1 Карточка варианта

Предусловие: пользователь успешно авторизовался в системе.

Гарантия: пользователь получил результат SQL-запроса во вкладке.

Минимальная гарантия: пользователь получил сообщение об ошибке, база данных осталась в согласованном состоянии.

7.2 Сценарий

Главный успешный сценарий:

1. Пользователь начинает работу с запросом
2. Система присылает форму для ввода параметров запроса

3. Пользователь вводит параметры и отправляет их системе
4. Система присылает результат выполнения запроса

Исключения:

- 4.a Неверный формат ввода параметров. Система снова присылает форму для ввода параметров запроса с сообщением об ошибке ввода.
- 4.6 Результатом запроса оказался пустой `result_set`. Система снова присылает форму для ввода параметров запроса с сообщением об отсутствии результатов предыдущего запроса.

7.3 Системная диаграмма последовательности

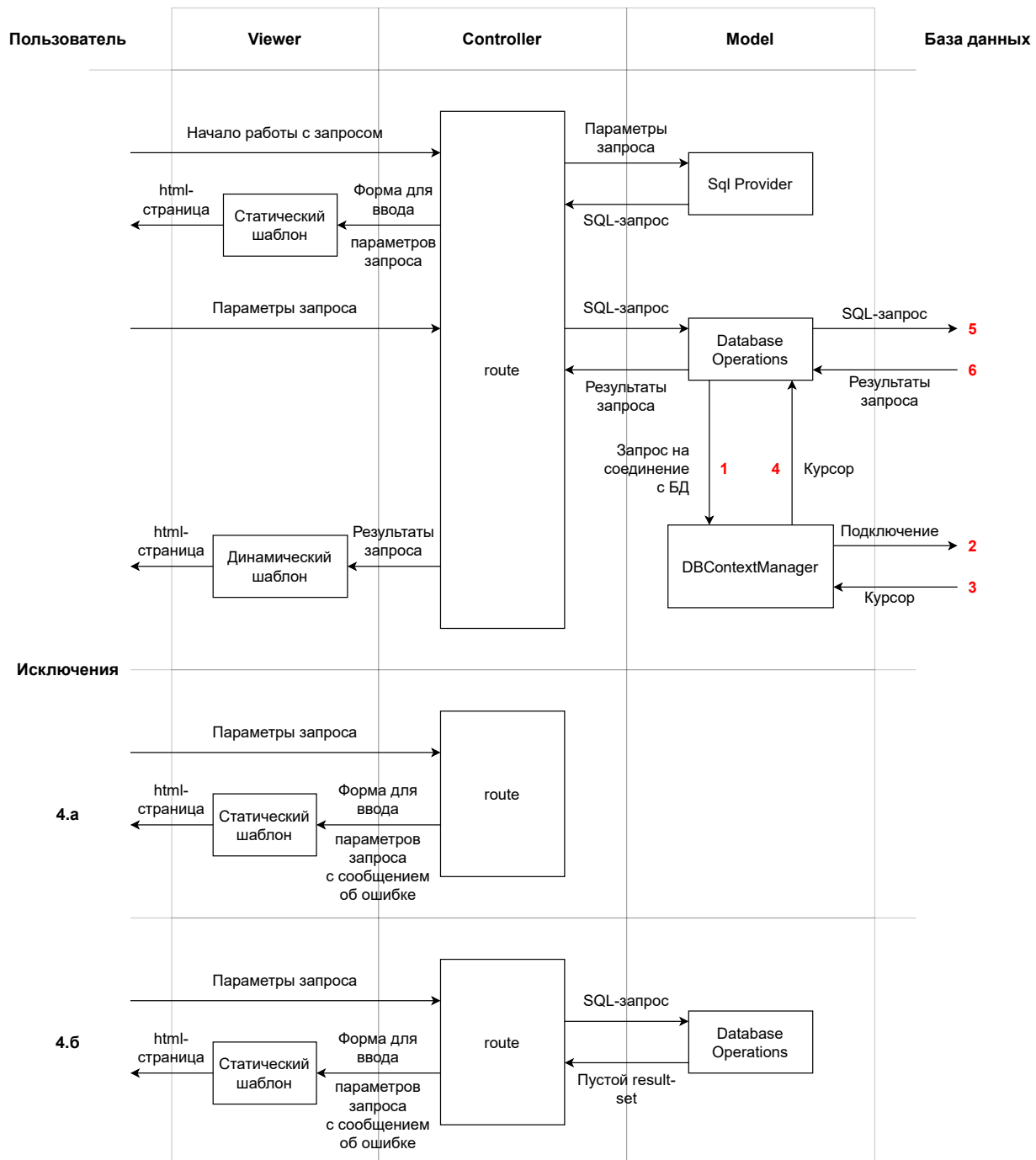


Рис. 4. MVC системная диаграмма последовательности для варианта использования «выполнение запроса»

7.4 Файловая структура

```

queries
├─ sql
│  ├─ check_user_doesnt_exist.sql
│  ├─ insert_new_internal_user.sql
│  ├─ search_med_by_country.sql
│  ├─ search_med_by_group.sql
│  ├─ search_med_by_name.sql
│  ├─ search_supplier_by_city.sql
│  ├─ show_balance_by_group.sql
│  ├─ show_ord_by_date.sql
│  └─ show_rating_ord_by_month.sql
├─ templates
│  ├─ add_new_internal_user.html
│  ├─ base_for_search.html
│  ├─ queries_index.html
│  ├─ search_med_by_country.html
│  ├─ search_med_by_group.html
│  ├─ search_med_by_name.html
│  ├─ search_supplier_by_city.html
│  ├─ show_balance_by_group.html
│  ├─ show_ord_by_date.html
│  └─ show_rating_ord_by_month.html
└─ routes.py

```

Рис. 5. Файловая структура запросов

7.5 Требования к шаблонам

1. Статический шаблон с выбором шаблона запроса – `queries_index.html`
 - Страница предназначена для выбора выполняемого запроса.
 - Для каждого шаблона запроса на странице должна быть размещена ссылка для перехода на страницу с формой ввода параметров запроса.
 - Рядом с каждой ссылкой должно быть расположено пояснение запроса, к которому принадлежит ссылка.
 - Под всеми вариантами запросов должна располагаться ссылка для перехода в главное меню.
2. Динамические шаблоны для форм запросов – `search_*.html`, `show_*.html`
 - Шаблон должен содержать в себе формы для ввода параметров соответствующего запроса.
 - Формы для ввода должны располагаться в верхней части страницы.
 - Все формы для ввода должны быть обязательны для заполнения.
 - Под формами должна располагаться кнопка «Поиск», осуществляющая передачу введенных данных на сервер с последующей переадресацией на страницу просмотра результатов запроса.

- Шаблон должен предусматривать область для вывода сообщений об ошибке под формами.
- Страница также предназначена для отображения результатов выполненного запроса.
- Результат запроса должен быть представлен в виде таблицы по центру страницы.
- Под полями ввода параметров для запроса должна располагаться ссылка для возврата к выбору шаблона запроса.

8 Вариант использования «Авторизация»

8.1 Карточка варианта

Предусловие: пользователь успешно открыл форму для ввода логина и пароля.

Гарантия: пользователь успешно авторизовался в системе как внутренний или внешний и получил доступ к соответствующему меню.

Минимальная гарантия: база данных осталась в согласованном состоянии, пользователь получил сообщение об ошибке авторизации с кнопкой для повторного ввода.

8.2 Сценарий

Главный успешный сценарий:

1. Пользователь начинает авторизацию
2. Система присылает форму для ввода логина и пароля
3. Пользователь вводит свой логин и пароль и отправляет их системе
4. Система создает для пользователя сессию с его группой доступа и перенаправляет его в главное меню.

Исключения:

- 4.a Неверный формат ввода параметров. Система снова присылает форму для ввода логина и пароля с сообщением об ошибке ввода.
- 4.6 Пользователь с таким логином и паролем не найден в системе. Система снова присылает форму для ввода логина и пароля с сообщением о неуспешной авторизации.

8.3 Системная диаграмма последовательности

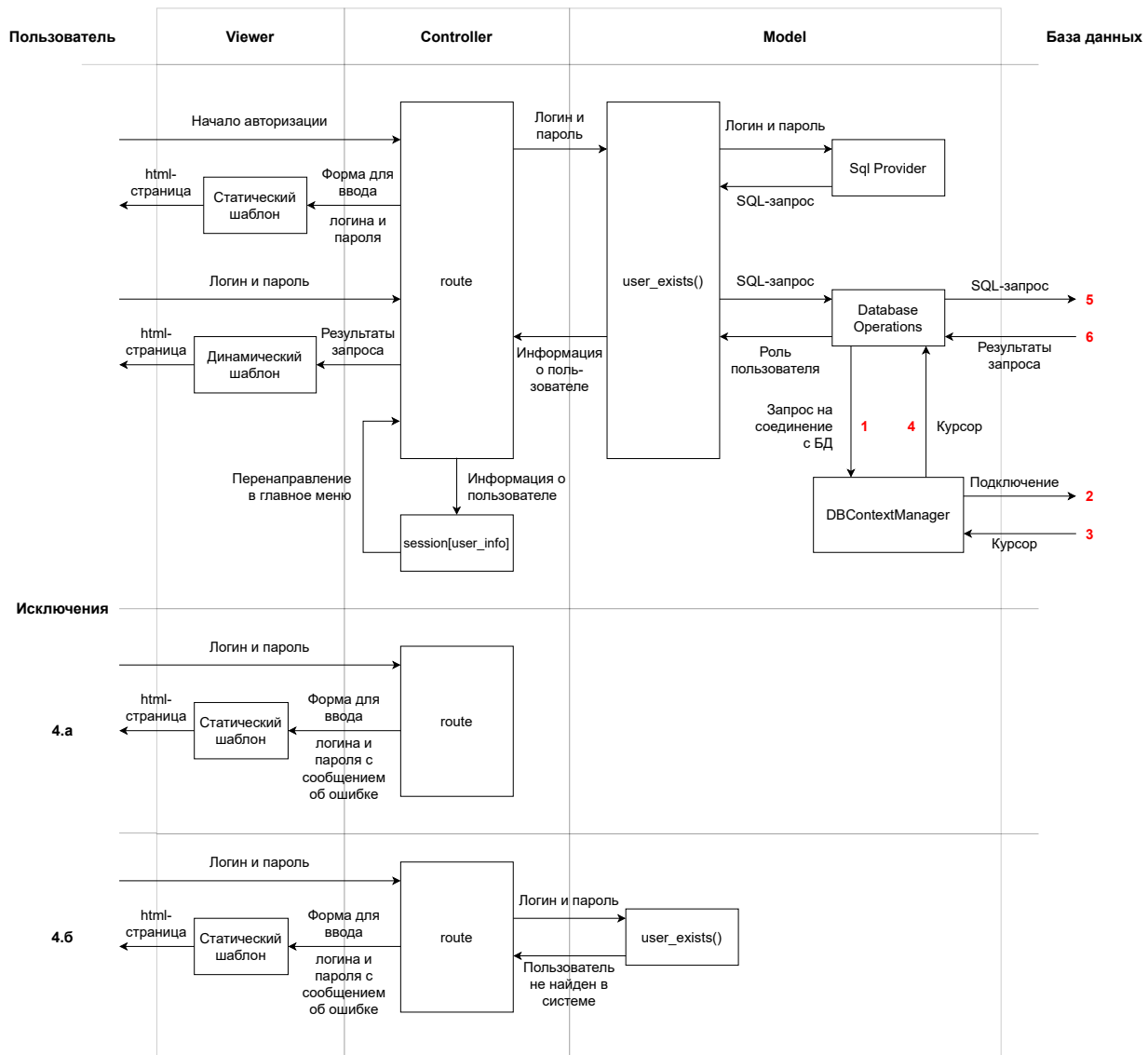


Рис. 6. MVC системная диаграмма последовательности для варианта использования «авторизация»

8.4 Файловая структура

```

access
├─ sql
│  ├─ add_user.sql
│  ├─ check_role.sql
│  └─ check_user_doesnt_exist.sql
├─ templates
│  ├─ access_index.html
│  └─ register.html
└─ routes.py

```

Рис. 7. Файловая структура авторизации

8.5 Требования к шаблонам

1. Статический шаблон для ввода логина и пароля – access.html

- Шаблон предназначен для ввода идентификационных данных пользователя и последующей авторизации в системе.
- В верхней части страницы должны располагаться формы для ввода логина и пароля.
- Обе формы должны быть обязательными для заполнения.
- Под формами должна располагаться кнопка «Отправить», осуществляющая передачу данных на сервер с последующей проверкой успешной авторизации.
- Ниже шаблон должен предусматривать область для вывода сообщения об ошибке.
- Внизу страницы должна располагаться ссылка для перехода в главное меню.

2. Статический шаблон для ввода логина и пароля – register.html

- Шаблон предназначен для ввода идентификационных данных пользователя и последующей регистрации в системе.
- В верхней части страницы должны располагаться формы для ввода логина и пароля.
- Обе формы должны быть обязательными для заполнения.
- Под формами должна располагаться кнопка «Отправить», осуществляющая передачу данных на сервер с последующей проверкой успешной регистрации.
- Ниже шаблон должен предусматривать область для вывода сообщения об ошибке.
- Внизу страницы должна располагаться ссылка для перехода к авторизации.

9 Вариант использования «Работа с отчетами»

9.1 Карточка варианта

Предусловие: в базе данных для каждого типа отчета создана соответствующая таблица. В базе данных разработана и сохранена процедура для создания отчетов. Пользователь успешно авторизовался как внутренний.

Гарантия: при введении нового отчетного периода, будут создаваться новые отчетные записи в таблице отчетов.

Минимальная гарантия: при попытке создать отчет второй раз выводится ошибка.

9.2 Сценарий

Главный успешный сценарий:

1. Пользователь начинает работу с отчетами.
2. Система присылает страницу со списком доступных отчетов и соответствующими им кнопками «создать» и «просмотр».
3. Пользователь выбирает функцию создания необходимого ему отчета.
4. Система присылает форму для ввода параметров отчетного периода.
5. Пользователь вводит параметры и отправляет их системе.
6. Система присылает страницу с сообщением об успешном создании отчета и предложением вернуться в меню отчетов.
7. Пользователь выбирает функцию просмотра необходимого ему отчета.
8. Система присылает форму для ввода параметров отчетного периода.
9. Пользователь вводит параметры и отправляет их системе.
10. Система присылает динамический шаблон с визуализацией отчета и предложением вернуться в меню отчетов.

Исключения:

- 4.а, 8.а Не введен отчетный период. Система снова присылает форму для ввода параметров отчетного периода с сообщением об ошибке ввода.
- 4.б Отчет за указанный отчетный период уже существует в БД. Система снова присылает форму для ввода параметров отчетного периода с соответствующим сообщением.
- 8.б Отчет за указанный отчетный период отсутствует в системе. Система снова присылает форму для ввода параметров отчетного периода с соответствующим сообщением.

9.3 Системная диаграмма последовательности

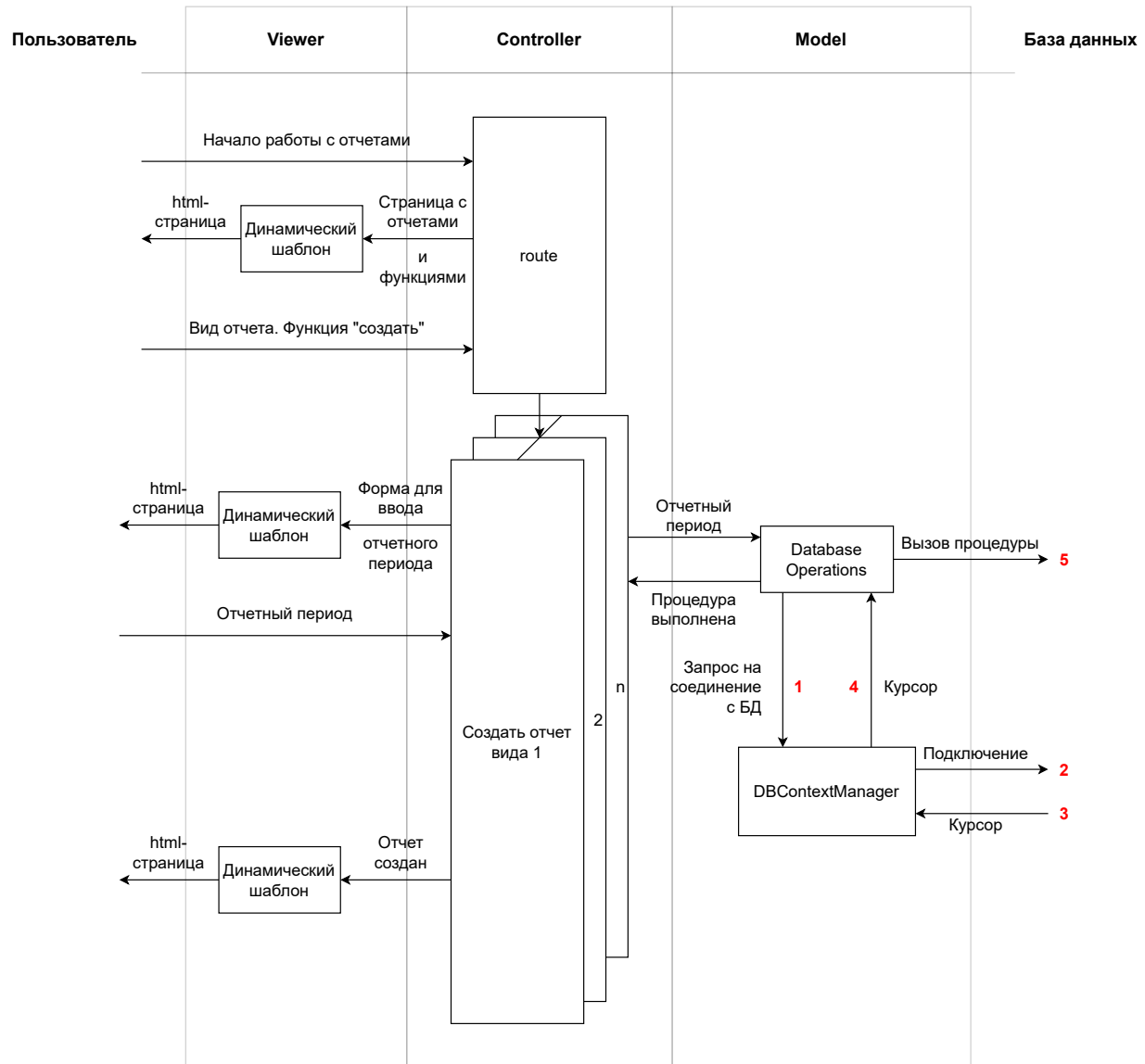


Рис. 8. MVC системная диаграмма последовательности для функции «создание отчета» варианта использования «работа с отчетами»

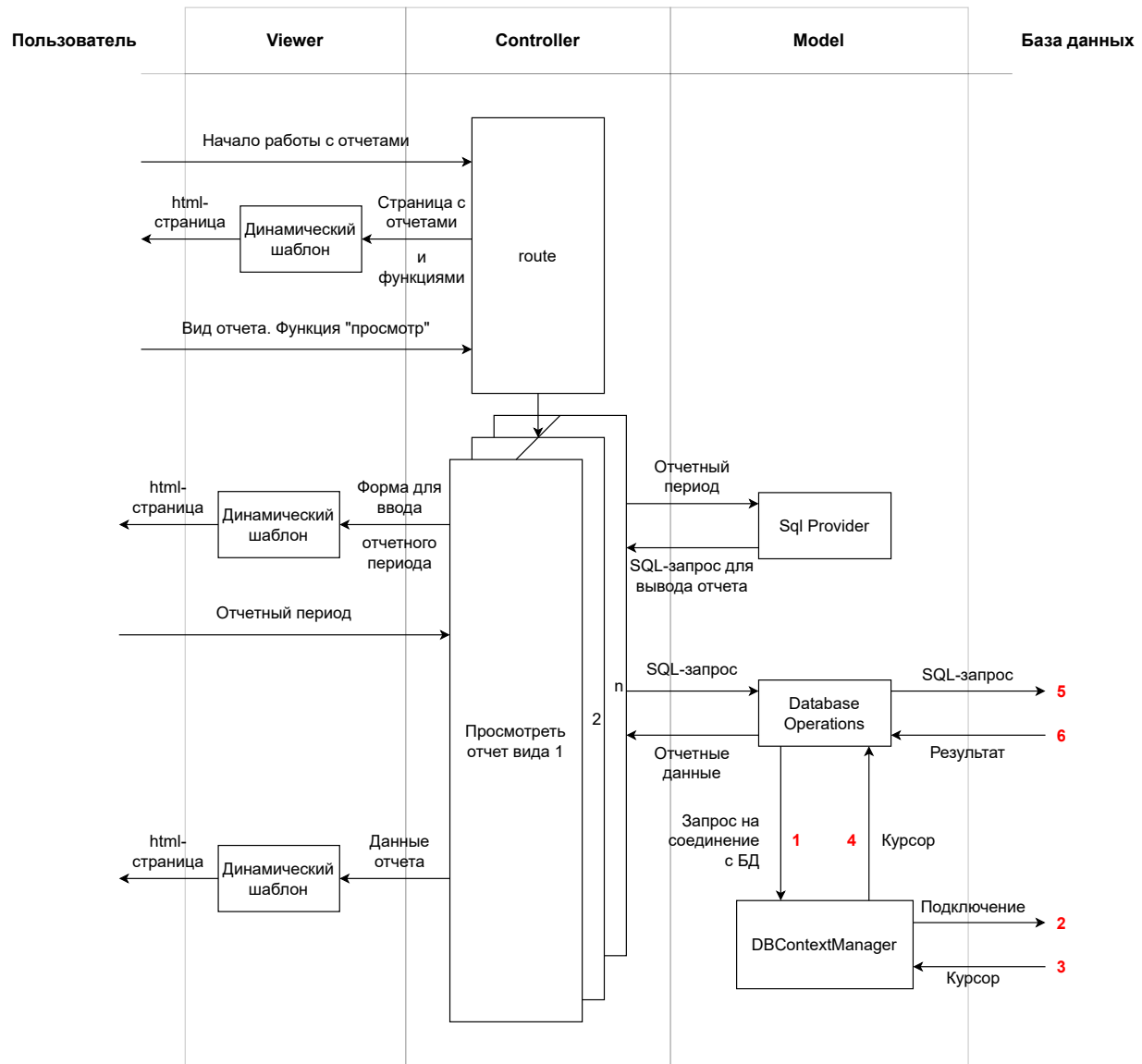


Рис. 9. MVC системная диаграмма последовательности для функции «просмотр отчета» варианта использования «работа с отчетами»

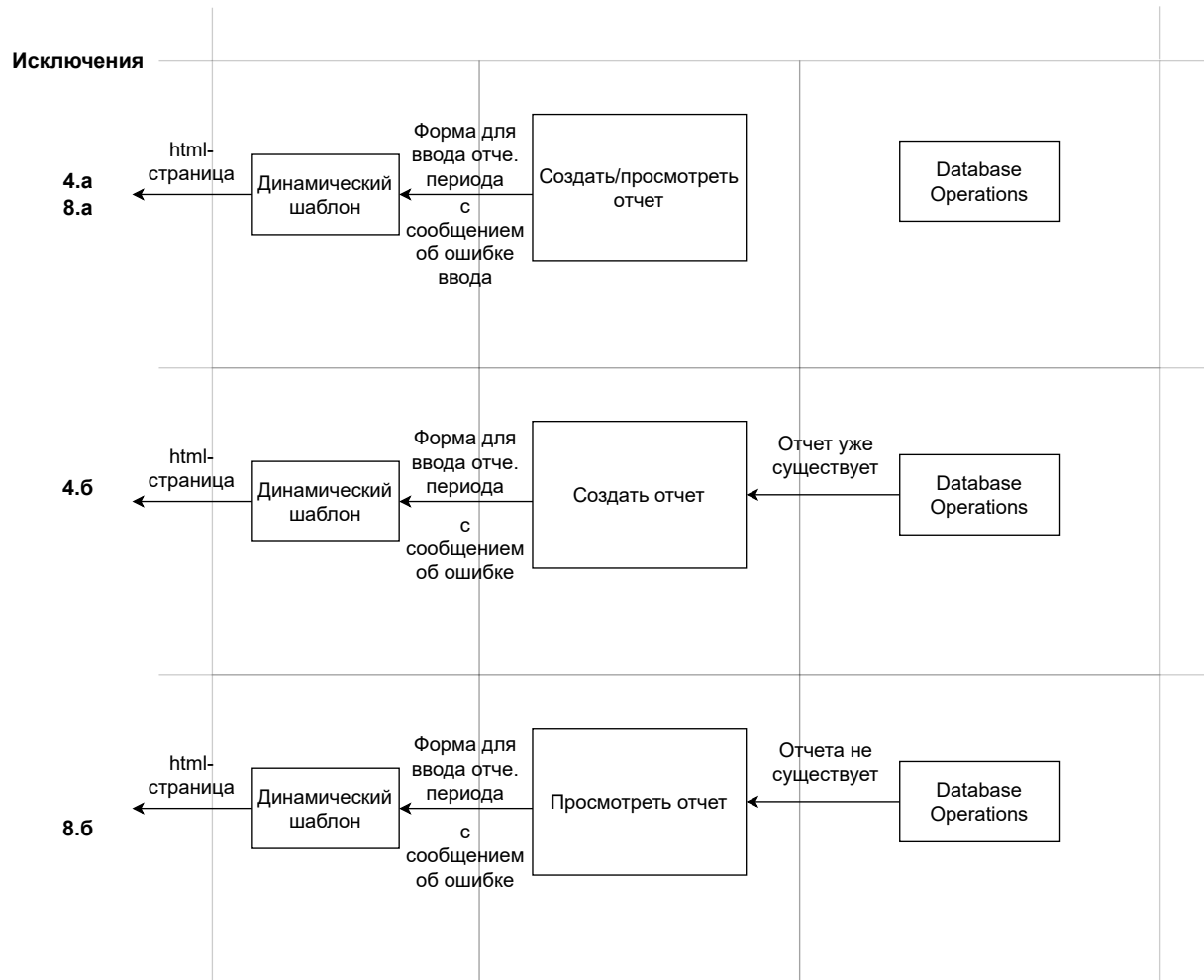


Рис. 10. MVC системная диаграмма последовательности исключений варианта использования «работа с отчетами»

9.4 Файловая структура

```

reports
├─ sql
│  ├─ show_orders_report.sql
│  └─ show_sales_report.sql
├─ templates
│  ├─ create_report.html
│  ├─ reports_index.html
│  └─ view_report.html
├─ routes.py
└─ utils.py
  
```

Рис. 11. Файловая структура отчетов

9.5 Требования к шаблонам

1. Динамический шаблон с выбором вида отчета и функции, осуществляемой над ним - reports_index.html.
 - Шаблон предназначен для отображения доступных видов отчета и предоставления кнопок-ссылок для перехода на страницы создания и просмотра соответствующих отчетов.
 - В центральной части страницы должен располагаться первый блок с доступными в системе отчетами.
 - Блок должен представлять из себя столбец, в котором в виде кратких пояснений о типе отчета перечислены все доступные отчеты.
 - Доступные в системе отчеты передаются в шаблон в виде списка (имя списка – "report_list").
 - Элементом списка является словарь (имя словаря – "report"), содержащий в себе ключи "rep_name" (строковое пояснение отчета) и "rep_id" (ID отчета в системе).
 - В правой части страницы должен располагаться второй блок с кнопками "Создать" и "Просмотр". При этом кнопки должны быть расположены напротив каждого отчета из первого блока.
 - Под формой ввода должна располагаться ссылка для перехода в главное меню.
 - При этом необходимо предусмотреть, чтобы для пользователя, авторизованного как "Директор" напротив каждого отчета отображалась только кнопка "Просмотр" а для пользователя, авторизованного как "Менеджер" – обе кнопки "Создать" и "Просмотр".
2. Статический шаблон для создания отчета – create_report.html.
 - Шаблоны предназначены для ввода отчетного периода просматриваемого или создаваемого отчета.
 - В верхней части страницы должно быть расположено пояснение просматриваемого или создаваемого отчета.
 - Ниже должны быть расположены необходимые формы для ввода отчетного периода.
 - Под формами должна быть расположена кнопка отправить передающая данные на сервер и инициирующая создание или просмотр отчета. В любом случае кнопка должна осуществлять переадресацию на следующую страницу.
 - Под кнопкой шаблон должен предусматривать область для вывода сообщения об ошибке (имя поля – "error").

- Внизу страницы должна располагаться ссылка для перехода в меню выбора отчетов.

3. Динамический шаблон для визуализации отчетов – view_report.html.

- Шаблоны предназначены для визуализации данных просматриваемого отчета.
- В верхней части страницы должно быть расположено пояснение просматриваемого отчета.
- Ниже должна быть расположена таблица, содержащая в себе все данные просматриваемого отчета.
- Поля шапки таблицы должны быть заданы в шаблоне индивидуально для каждого отчета.
- Вид таблицы должен быть разработан для каждого вида отчета отдельно.
- Под полями ввода должна располагаться ссылка для перехода в меню выбора отчетов.

10 Вариант использования «Формирование заказа»

10.1 Карточка варианта

Предусловие: авторизованный внешний пользователь решает оформить заказ на препараты.

Гарантия: пользователь создает новый заказ.

Минимальная гарантия: при попытке создать заказ с неверным содержимым (некорректное количество товара) выводится ошибка.

10.2 Сценарий

Главный успешный сценарий:

1. Пользователь заходит в корзину.
2. Система выдает страницу со списком доступных товаров.
3. Пользователь нажимает кнопку «Добавить товар».
4. Система присылает страницу со списком доступных товаров с информационным сообщением о том, что товар добавлен в корзину.
5. Пользователь нажимает кнопку «Перейти в корзину».
6. Система присылает страницу для отображения товаров в корзине и общей итоговой стоимостью товаров.

7. Пользователь нажимает кнопку «Очистить корзину».
8. Система присылает страницу со списком доступных товаров с информационным сообщением, что корзина очищена.
9. Пользователь нажимает кнопку «Оформить заказ».
10. Система возвращается на страницу с выбором товара.

Исключения:

- 3.a Товар не может быть добавлен в корзину. Пользователь получает сообщение об ошибке.
- 9.a Заказ не был создан в системе. Пользователь получает сообщение об ошибке создания заказа.

10.3 Системная диаграмма последовательности

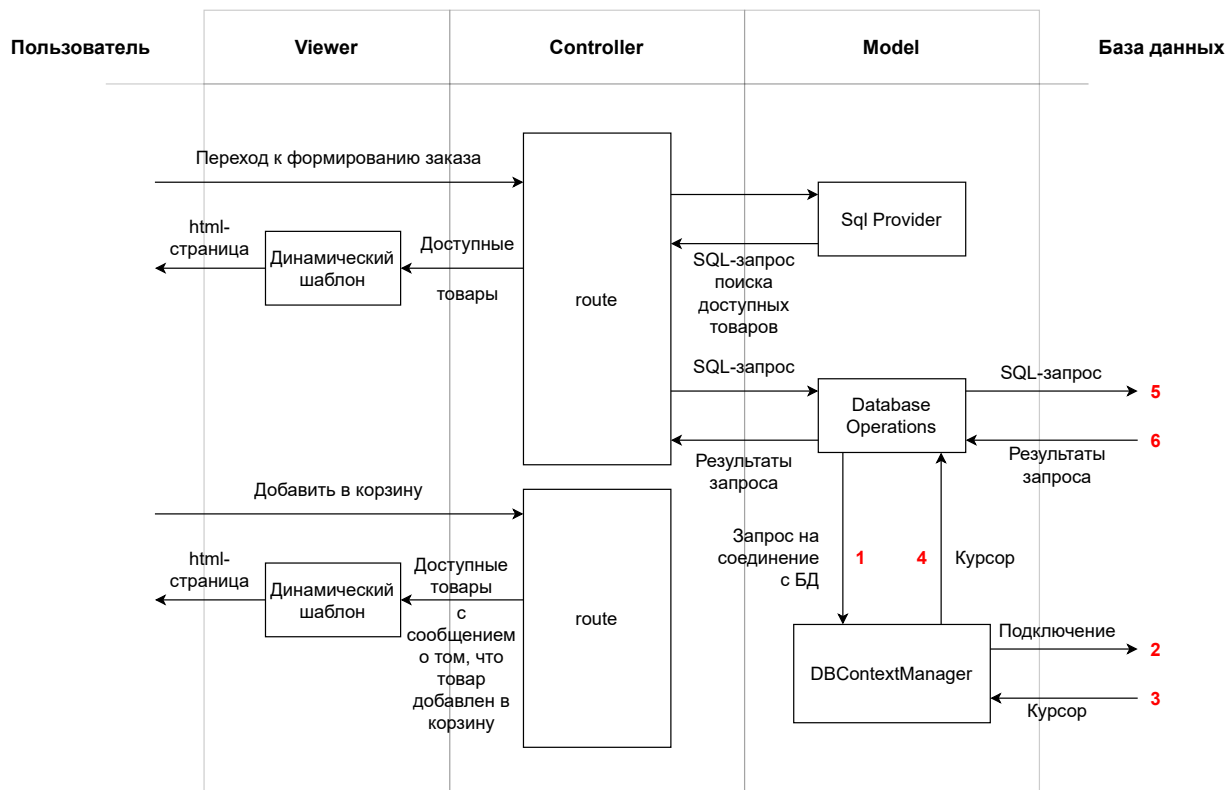


Рис. 12. MVC системная диаграмма последовательности варианта использования «формирование заказа»

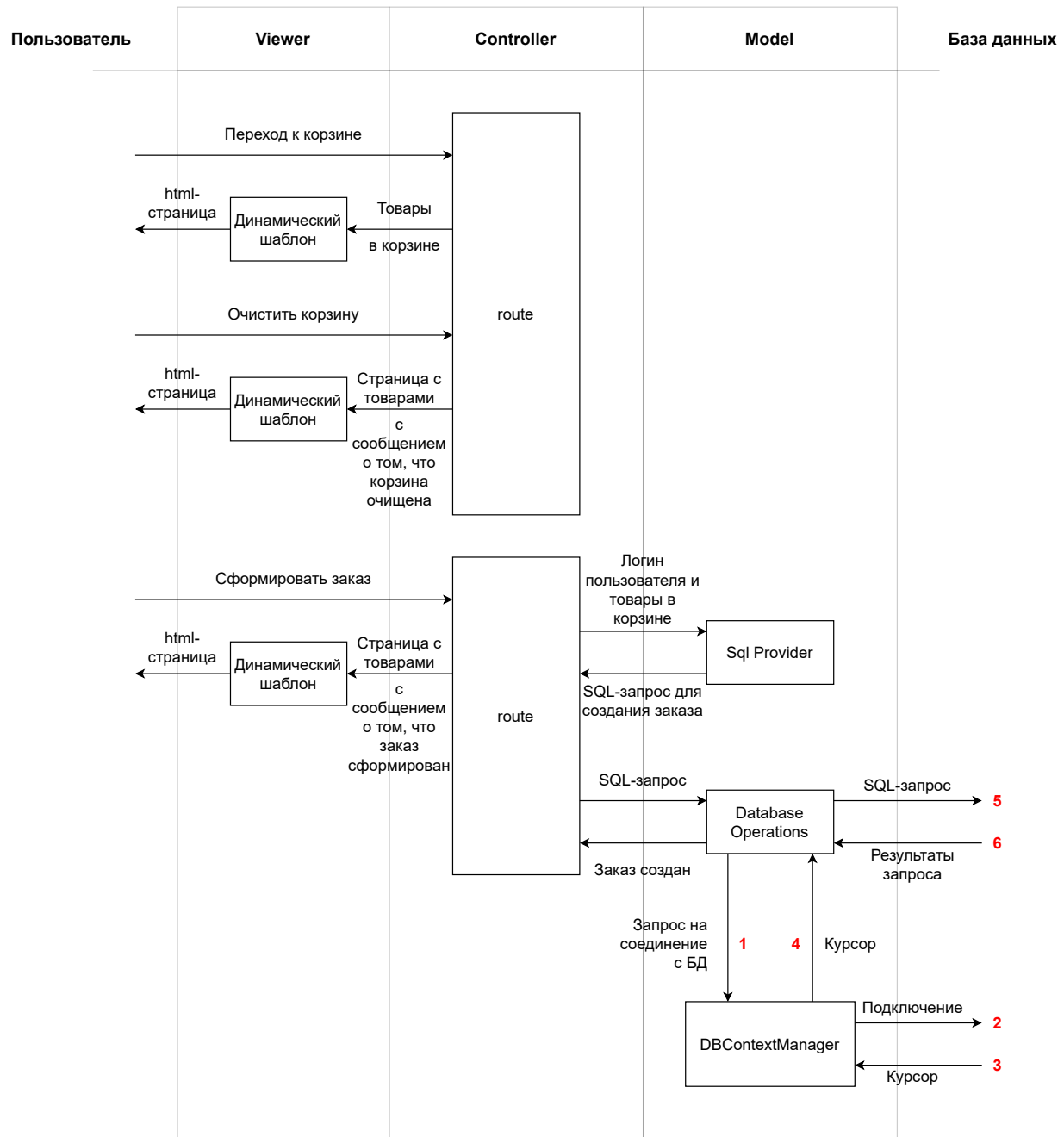


Рис. 13. MVC системная диаграмма последовательности варианта использования «корзина»

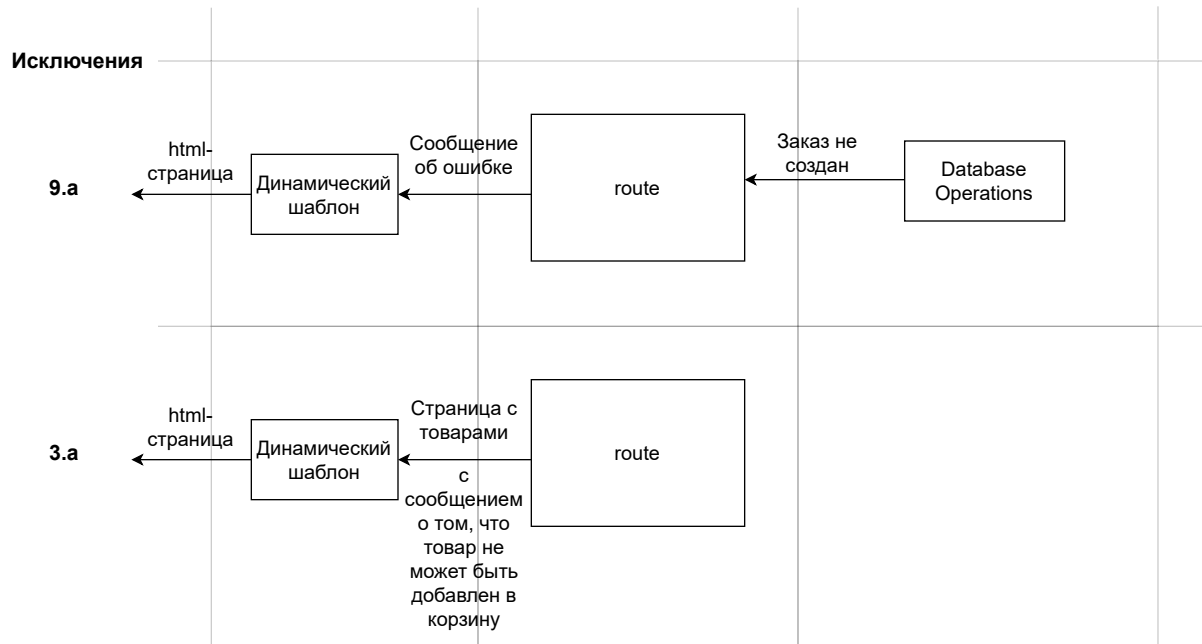


Рис. 14. MVC системная диаграмма последовательности исключения варианта использования «формирование заказа»

10.4 Файловая структура

```

basket
├─ sql
│  ├─ create_order.sql
│  └─ select_all_med.sql
├─ templates
│  ├─ basket_index.html
│  └─ show_basket.html
└─ routes.py
  
```

Рис. 15. Файловая структура формирования заказа

10.5 Требования к шаблонам

1. Динамический шаблон с доступными товарами – `basket_index.html`.
 - В центральной части страницы должен быть расположен блок с доступными товарами.
 - В нижней части страницы должны располагаться ссылки для возврата в главное меню и перехода к корзине.
 - Товар в блоке с доступными товарами должен представляться в виде строки с необходимой информацией. В строке товара также должна располагаться

кнопка "Добавить".

- В каждой строке в блоке доступного товара должно располагаться поле ввода количества товара.

2. Динамический шаблон с содержимым корзины – `show_basket.html`.

- В центральной части страницы должен быть расположен блок с товарами в корзине.
- В нижней части страницы должны располагаться ссылки для возврата к странице формирования заказа (доступные товары).
- Под блоком содержимого корзины должны располагаться кнопки: оформить заказ и очистить корзину.

11 Заключение

В результате выполнения данной курсовой работы была разработана информационная система аптеки на языке программирования Python в среде фреймворка Flask. При разработке было проведено предварительное проектирование и была составлена вся необходимая документация.

В процессе разработки были приобретены навыки программирования на языке Python, JavaScript, HTML и CSS на примере разработки клиент-серверного приложения. В процессе проектирования системы были получены навыки составления и систематизации требований и подготовки проектной документации.

Разработанное приложение позволяет пользователям информационной системы взаимодействовать с базой данных посредством web-ориентированного интерфейса. Система предоставляет пользователям возможности реализации различных вариантов использования с поддержанием бизнес-логики процессов и согласованием с базой данных.