



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:	Гапеев Дмитрий Андреевич
Группа:	РК6-51Б
Тип задания:	Курсовая работа
Тема:	Разработка информационной системы учёта поставок на склад

Студент

\_\_\_\_\_  
подпись, дата

Гапеев Д.А.  
\_\_\_\_\_  
Фамилия, И.О.

Преподаватель

\_\_\_\_\_  
подпись, дата

Пивоварова Н.В.  
\_\_\_\_\_  
Фамилия, И.О.

Москва, 2023

## Содержание

1	Техническое задание . . . . .	4
2	Описание предметной области . . . . .	4
3	Описание конечных пользователей . . . . .	5
4	UML-диграмма вариантов использования информационной системы . . . . .	6
5	Инфологическая модель предметной области в форме UML-диаграммы классов . . . . .	7
6	Логическая модель БД . . . . .	8
7	Общая файловая структура . . . . .	9
8	Вариант использования «Работа с запросами» . . . . .	9
	8.1 Карточка варианта . . . . .	9
	8.2 Сценарий . . . . .	9
	8.3 Системная диаграмма последовательности . . . . .	11
	8.4 Файловая структура . . . . .	12
	8.5 Требования к шаблонам . . . . .	12
9	Вариант использования «Авторизация» . . . . .	13
	9.1 Карточка варианта . . . . .	13
	9.2 Сценарий . . . . .	13
	9.3 Системная диаграмма последовательности . . . . .	14
	9.4 Файловая структура . . . . .	15
	9.5 Требования к шаблонам . . . . .	15
10	Вариант использования «Работа с отчетами» . . . . .	16
	10.1 Карточка варианта . . . . .	16
	10.2 Сценарий . . . . .	16
	10.3 Системная диаграмма последовательности . . . . .	18
	10.4 Файловая структура . . . . .	20
	10.5 Требования к шаблонам . . . . .	21
11	Вариант использования «Формирование накладной» . . . . .	22
	11.1 Карточка варианта . . . . .	22
	11.2 Сценарий . . . . .	22
	11.3 Системная диаграмма последовательности . . . . .	23
	11.4 Файловая структура . . . . .	25
	11.5 Требования к шаблонам . . . . .	25
12	Заключение . . . . .	26
13	Список использованной литературы . . . . .	26

## Аннотация

Курсовая работа на тему «Разработка информационной системы учёта поставок заготовок на склад малого предприятия».

Выполнил - Гапеев Д.А.

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 26 листа, 16 рисунков.

## 1 Техническое задание

Этап проектирования:

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.

Этап реализации:

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

На малом предприятии (МП) из заготовок изготавливают детали.

Заготовки, хранящиеся на складе МП, характеризуются уникальным кодом заготовки, весом, материалом, ценой, количеством, датой последнего обновления количества. Одни и те же заготовки, поставленные по разной цене, учитываются отдельно.

Заготовки на предприятие поставляются поставщиками, о которых известны уникальный номер, название, город, дата заключения контракта с предприятием. Прежде чем начать поставки заготовок поставщик должен заключить договор с МП и зарегистрироваться в базе данных последнего.

При поставке очередной партии заготовок поставщиком, создается накладная. Каждая накладная имеет уникальный номер, для нее известна дата поставки, поставщик и общая стоимость всех поставленных заготовок.

По одной накладной один и тот же поставщик может поставить произвольное количество различных заготовок. Поэтому каждая накладная может в общем случае содержать несколько строк накладной. В каждой строке указывается код заготовки, ее цена и поставленное количество.

На предприятии ведется учет хранящихся заготовок. При поступлении новой партии заготовок от конкретного поставщика обновляется количество соответствующих заготовок на складе и проставляется дата последнего обновления количества заготовок.

### 3 Описание конечных пользователей

В данной системе пользователи разделяются на следующие категории:

1. Директор - внутренний пользователь информационной системы, варианты использования: авторизация, просмотр отчетов.
2. Менеджер - внутренний пользователь информационной системы, варианты использования: авторизация, создание и просмотр отчетов.
3. Товаровед - внутренний пользователь информационной системы, варианты использования: авторизация, работа с запросами, связанных со складом и поставками, подтверждение накладных.
4. Логист - внутренний пользователь информационной системы, варианты использования: авторизация, работа с запросами, связанных с поставщиками и поставками.
5. Поставщик - внешний пользователь информационной системы, вариант использования: авторизация, формирование накладных для склада.

#### 4 UML-диаграмма вариантов использования информационной системы

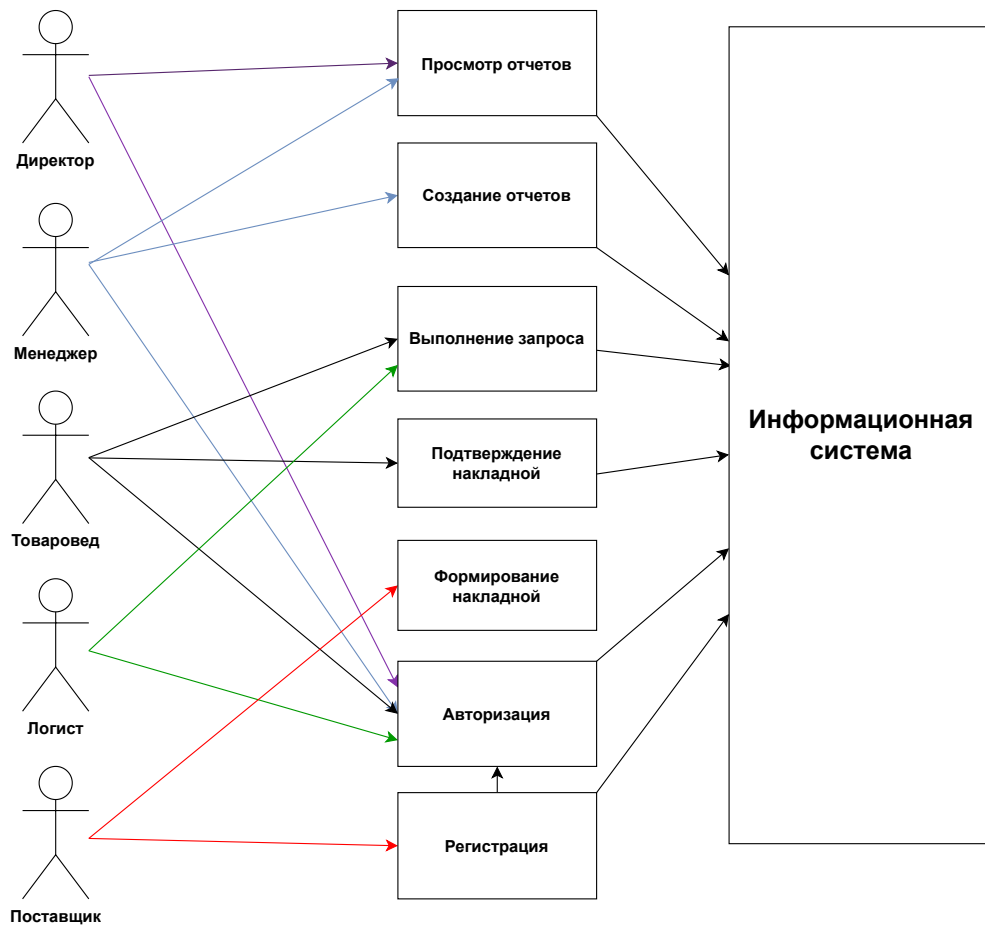


Рис. 1. UML-диаграмма вариантов использования информационной системы

## 5 Инфологическая модель предметной области в форме UML-диаграммы классов

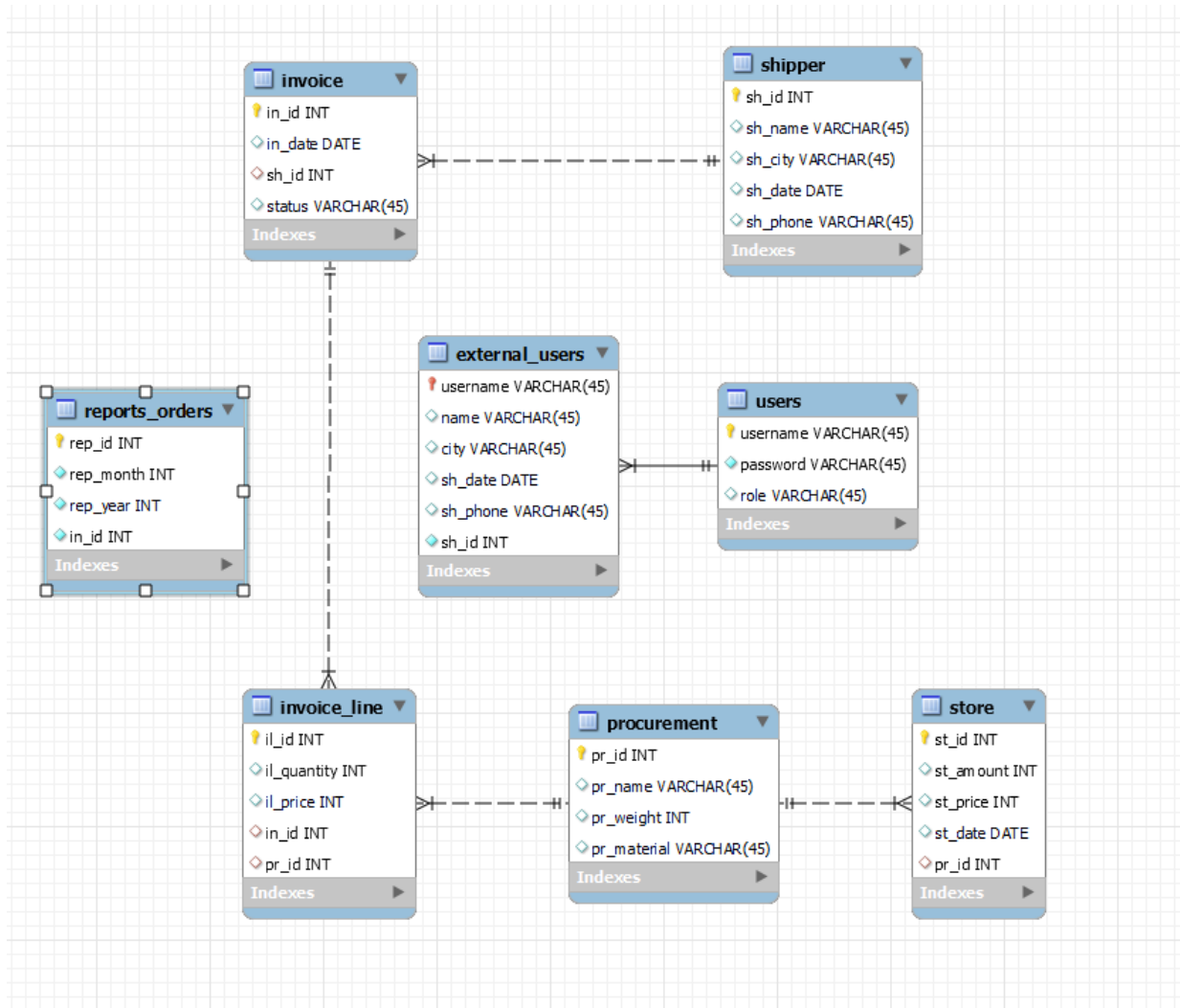


Рис. 2. Инфологическая модель предметной области в форме UML-диаграммы классов

## 6 Логическая модель БД

invoice					
in_id	in_date	sh_id	status		
PK		FK(shippers)			
shipper					
sh_id	sh_name	sh_city	sh_date	sh_phone	
PK					
external_users					
username	name	city	sh_date	sh_phone	sh_id
PK,FK(users)					
users					
username	password	role			
PK					
invoice_line					
il_id	il_quantity	il_price	in_id	pr_id	
PK			FK(invoice)	FK(procurement)	
procurement					
pr_id	pr_name	pr_weight	pr_material		
PK					
store					
st_id	st_amount	st_price	st_date	pr_id	
PK				FK(procurement)	
reports_orders					
red_id	rep_month	rep_year	in_id		
PK			FK(invoice)		

Рис. 3. Логическая модель БД



## 7 Общая файловая структура

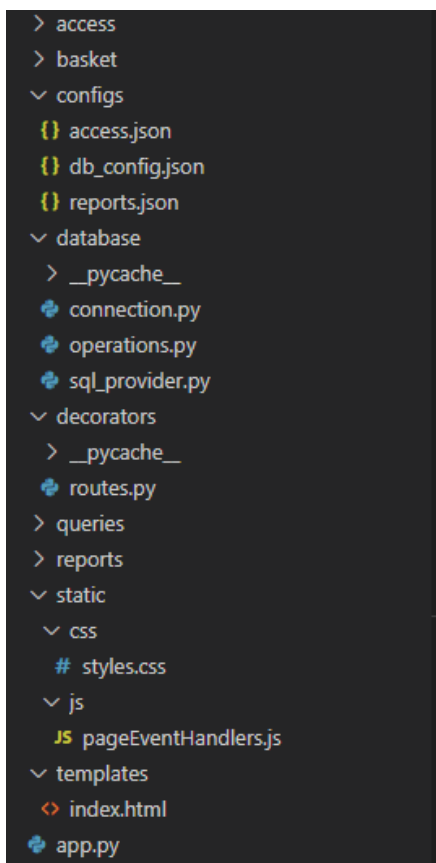


Рис. 4. Общая файловая структура проекта

## 8 Вариант использования «Работа с запросами»

### 8.1 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.

### 8.2 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.б Результатом запроса оказался пустой `result_set`. Система снова присылает форму для ввода параметров запроса с сообщением об отсутствии результатов предыдущего запроса.

### 8.3 Системная диаграмма последовательности

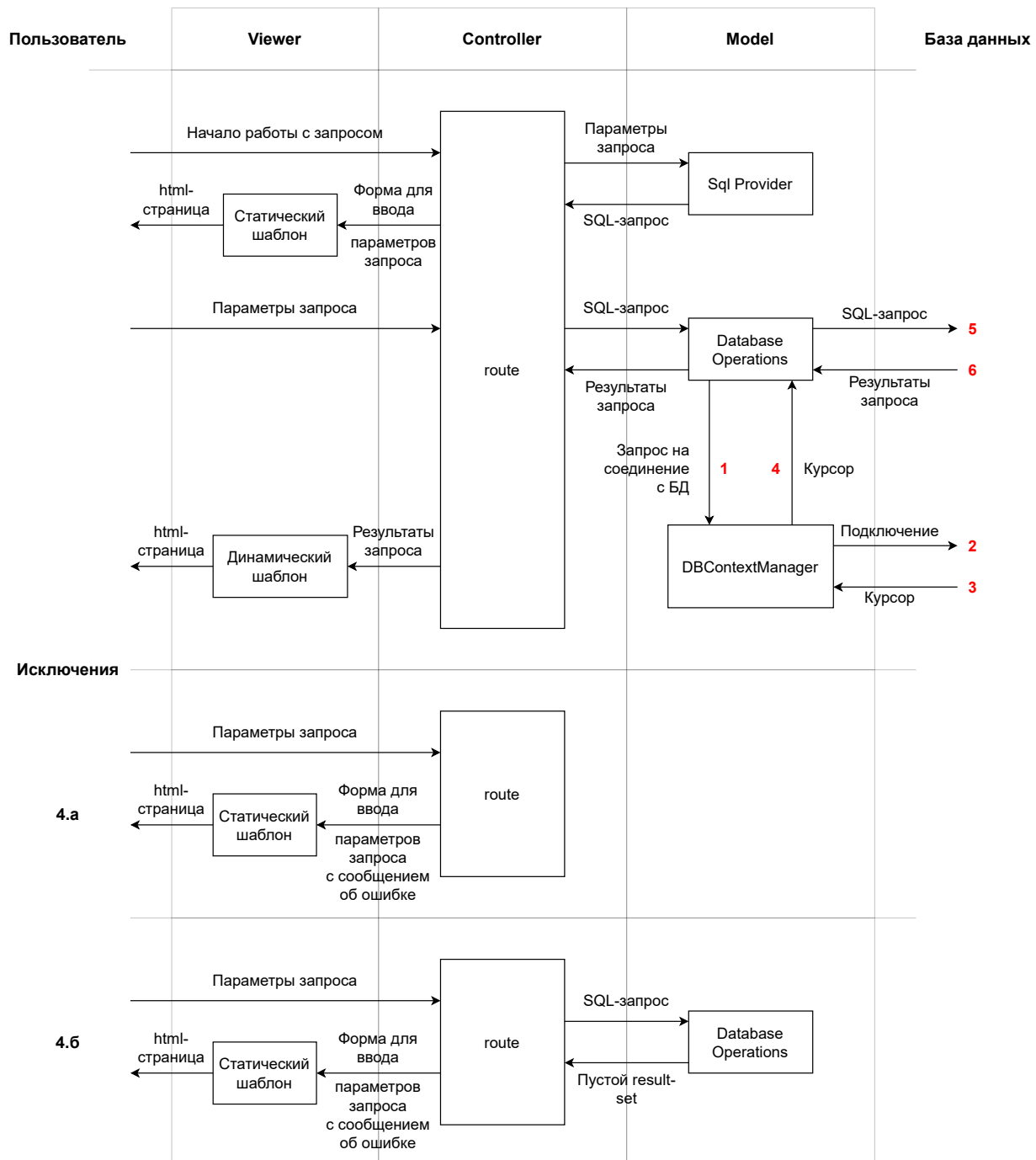


Рис. 5. MVC системная диаграмма последовательности для варианта использования «выполнение запроса»

## 8.4 Файловая структура

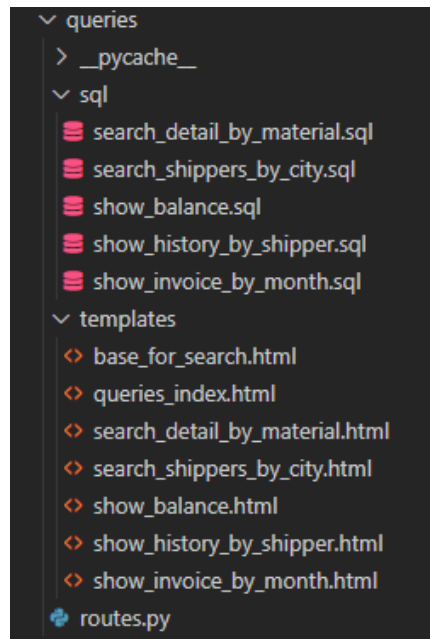


Рис. 6. Файловая структура запросов

## 8.5 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.

## 9 Вариант использования «Авторизация»

### 9.1 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 повтора ввода.

### 9.2 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.

### 9.3 Системная диаграмма последовательности

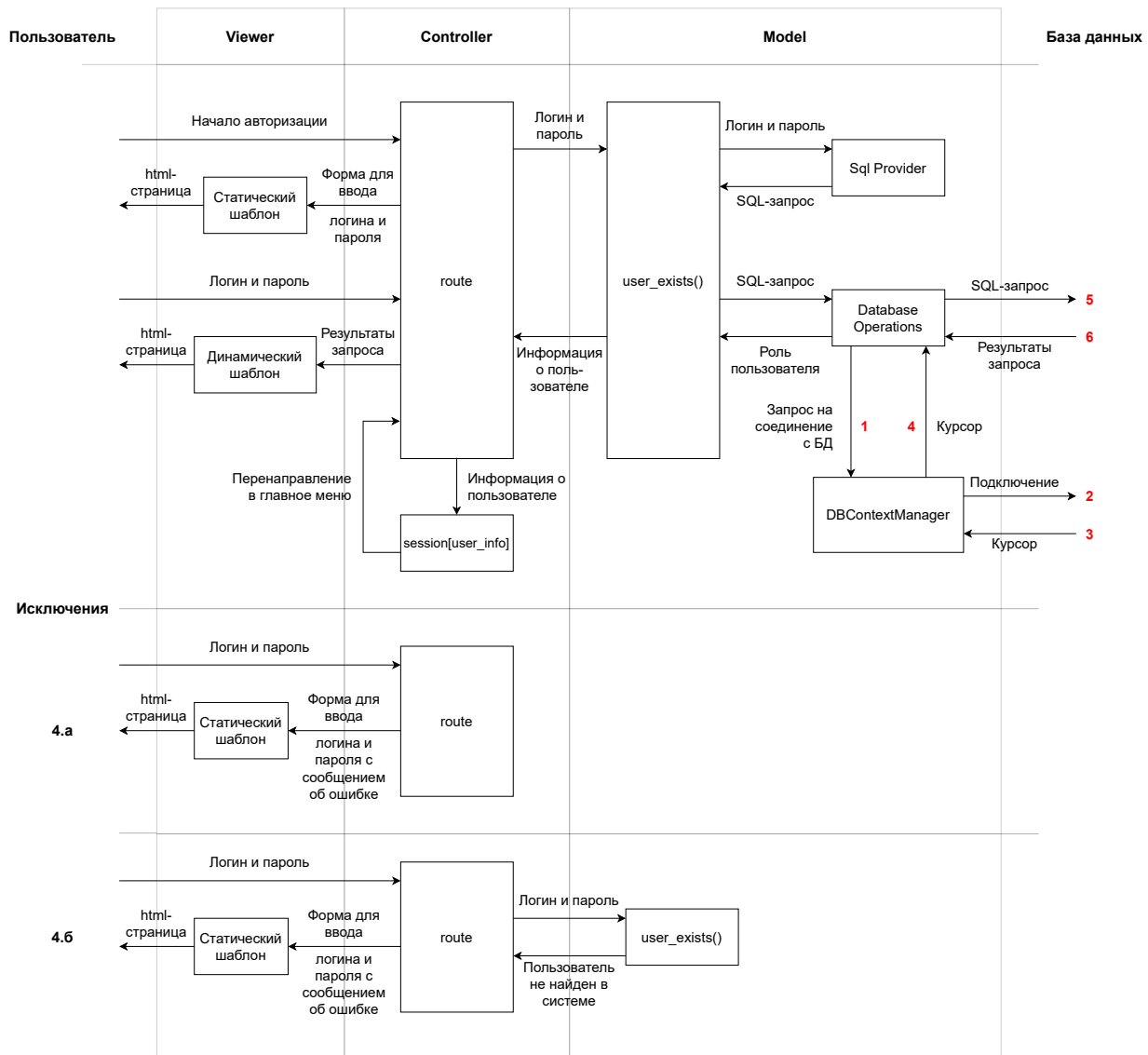


Рис. 7. MVC системная диаграмма последовательности для варианта использования «авторизация»

## 9.4 Файловая структура

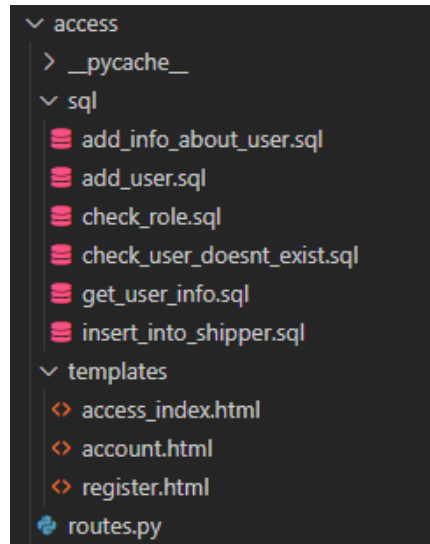


Рис. 8. Файловая структура авторизации

## 9.5 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.





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

### 10.3 Системная диаграмма последовательности

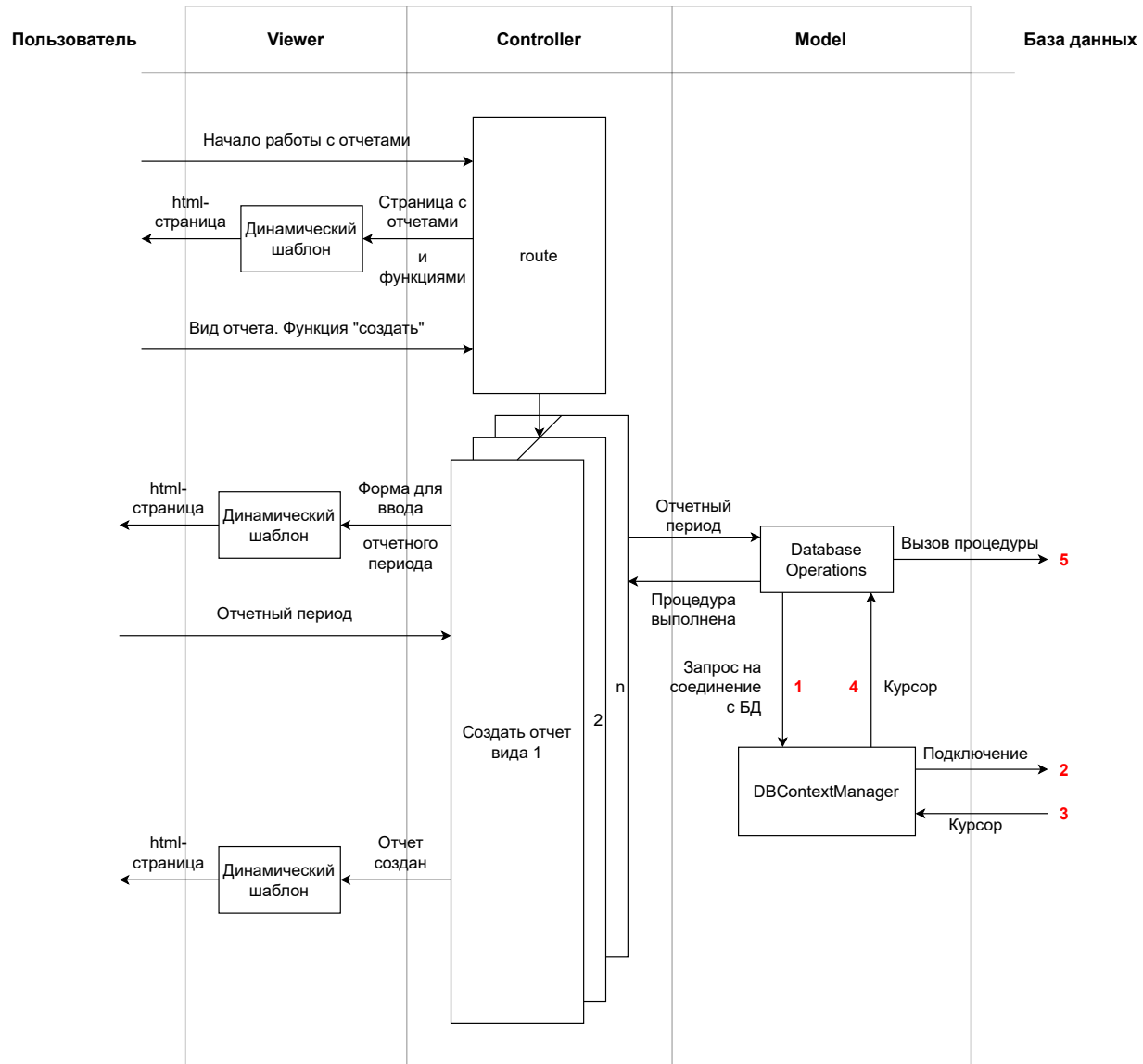


Рис. 9. MVC системная диаграмма последовательности для функции «создание отчета» варианта использования «работа с отчетами»

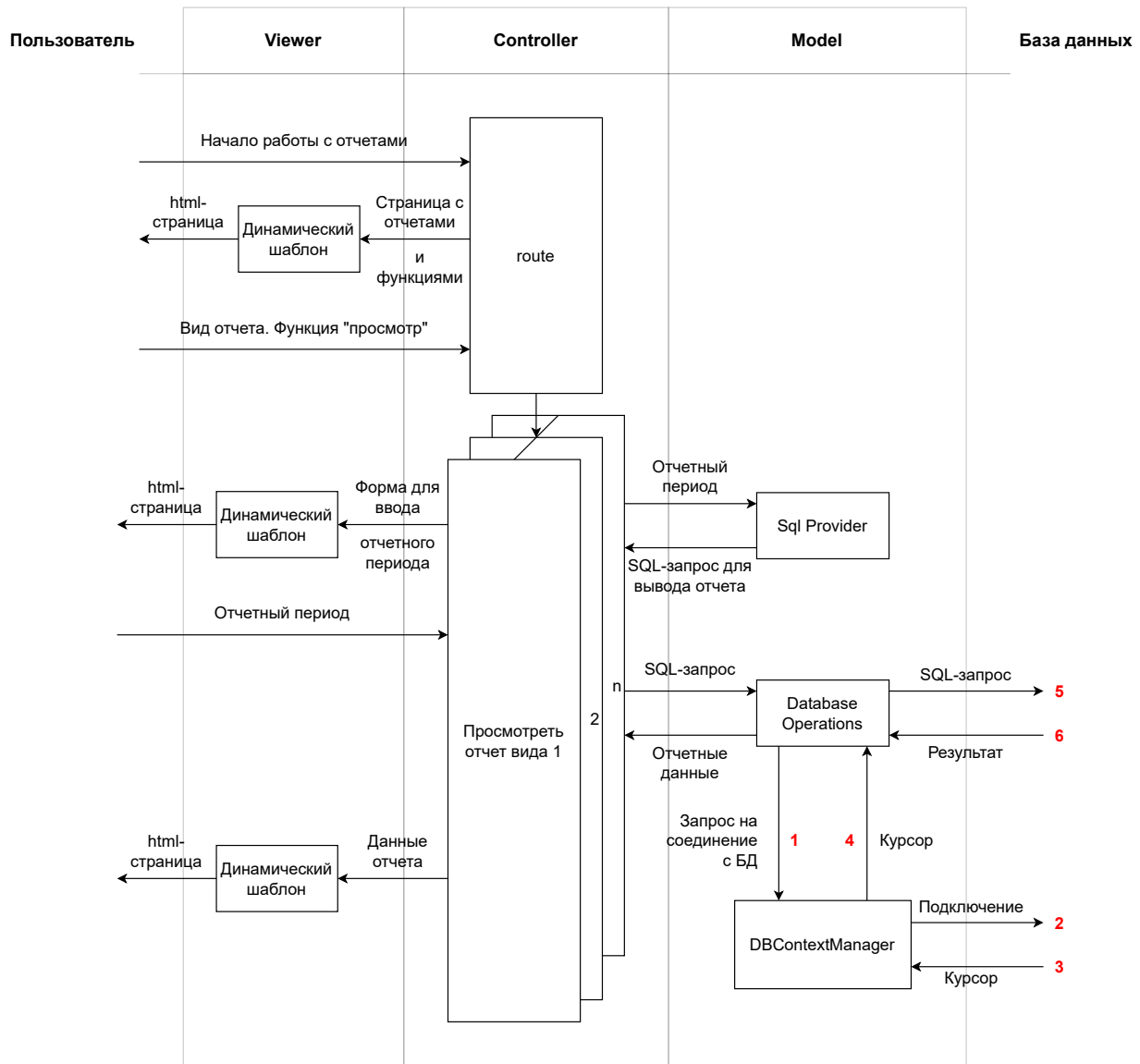


Рис. 10. MVC системная диаграмма последовательности для функции «просмотр отчета» варианта использования «работа с отчетами»

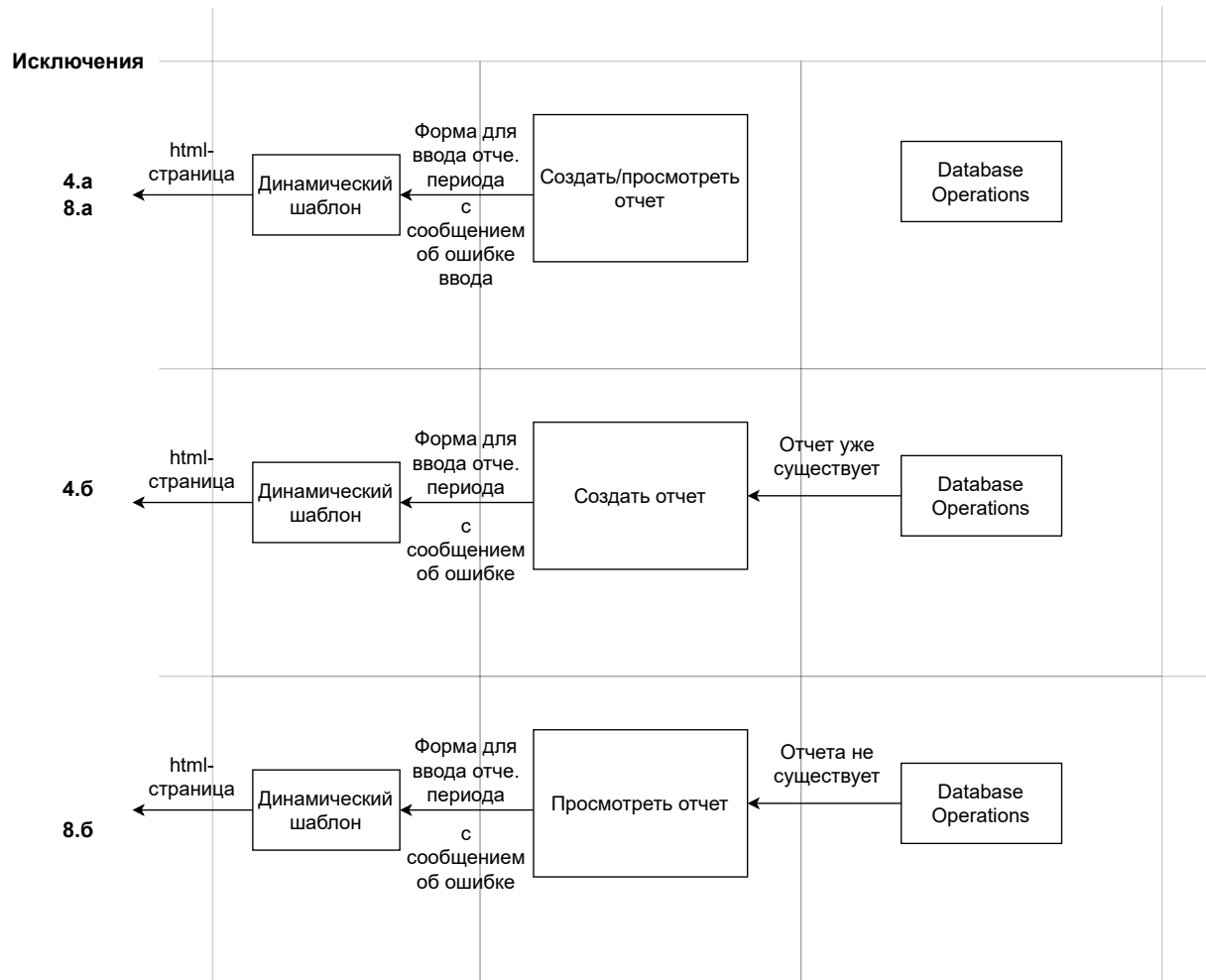


Рис. 11. MVC системная диаграмма последовательности исключений варианта использования «работа с отчетами»

## 10.4 Файловая структура

```

v reports
> __pycache__
v sql
  show_orders_report.sql
v templates
  create_report.html
  reports_index.html
  view_report.html
routes.py
utils.py
  
```

Рис. 12. Файловая структура отчетов

## 10.5 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". 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.

## 11 Вариант использования «Формирование накладной»

### 11.1 Карточка варианта

Предусловие: авторизованный внешний пользователь оформляет накладную на поставленные им товары.

Гарантия: пользователь оформил накладную.

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

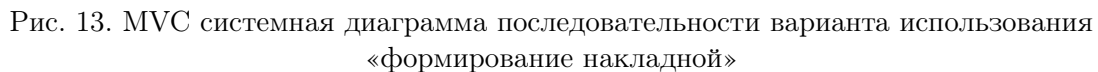
### 11.2 Сценарий

Главный успешный сценарий:

1. Пользователь заходит в "Формирование заказа на поставку".
2. Система выдает страницу со списком доступных заготовок.
3. Пользователь нажимает кнопку «Добавить».
4. Система присылает страницу со списком доступных товаров с информационным сообщением о том, что товар добавлен в накладную.
5. Пользователь нажимает кнопку «Просмотр накладной».
6. Система присылает страницу для отображения заготовок в накладной и общей итоговой стоимостью заготовок.

- Исключения:

- ### 11.3 Системная диаграмма последовательности



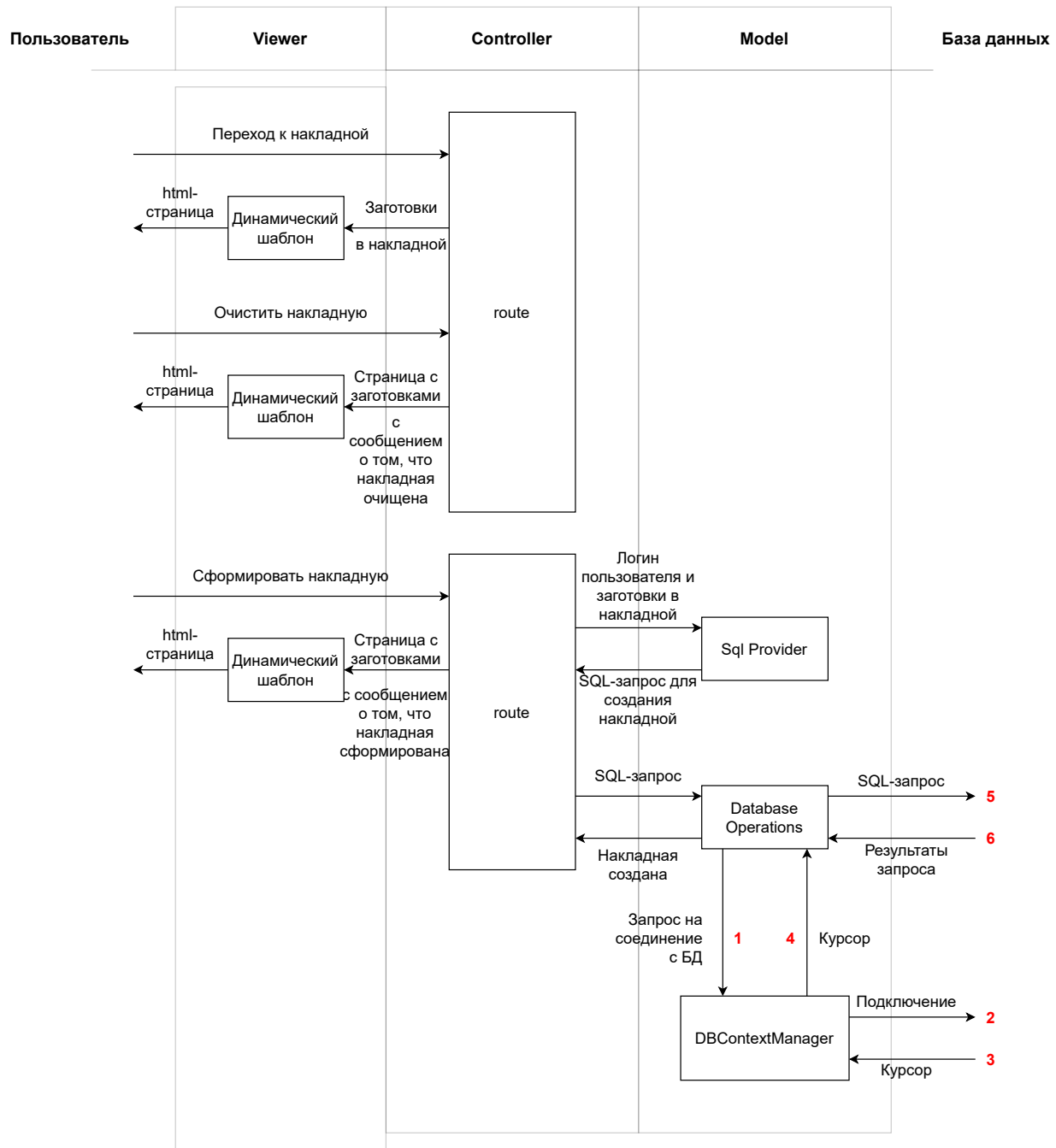


Рис. 14. MVC системная диаграмма последовательности варианта использования «накладная»



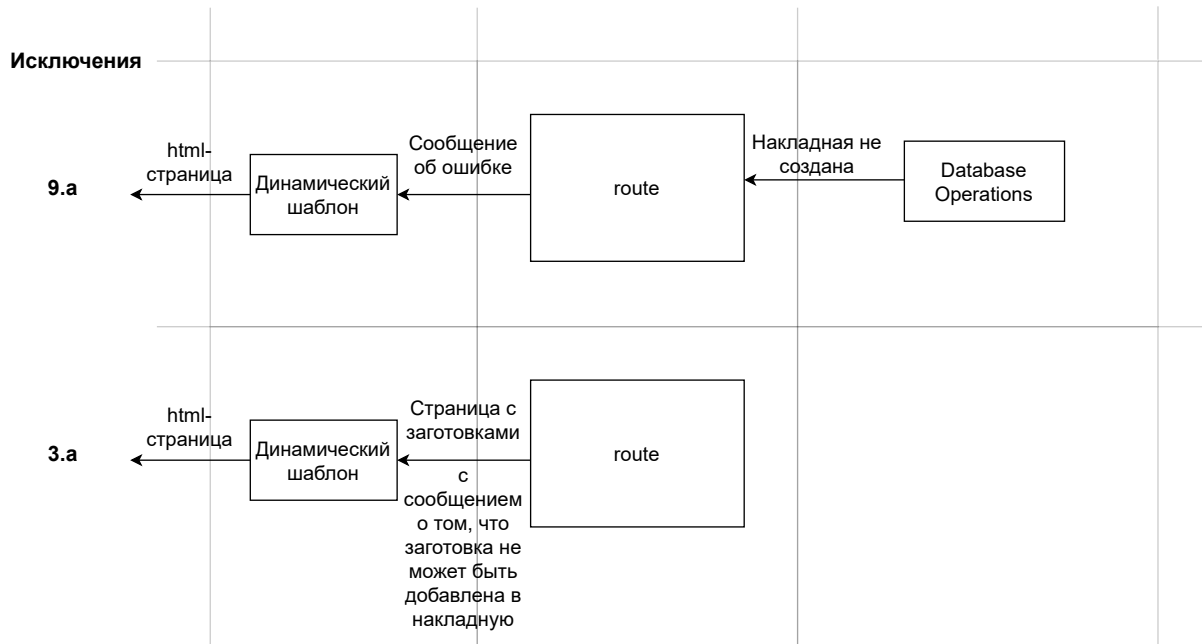


Рис. 15. MVC системная диаграмма последовательности исключения варианта использования «формирование накладной»

## 11.4 Файловая структура

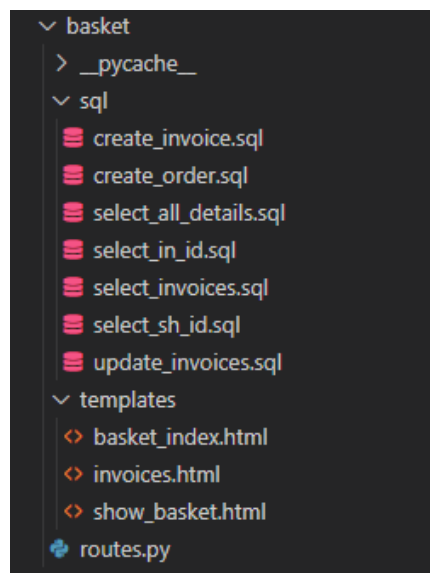


Рис. 16. Файловая структура формирования накладной

## 11.5 Требования к шаблонам

1. Динамический шаблон с доступными заготовками – `basket_index.html`.

- В центральной части страницы должен быть расположен блок с доступными заготовками.
- В нижней части страницы должны располагаться ссылки для возврата в главное меню и перехода к накладной.
- Товар в блоке с доступными заготовками должен представляться в виде строки с необходимой информацией. В строке товара также должна располагаться кнопка "Добавить".
- В каждой строке в блоке доступной заготовки должно располагаться поле ввода количества товара.

## 2. Динамический шаблон с содержимым накладной – `show_basket.html`.

- В центральной части страницы должен быть расположен блок с заготовками в накладной.
- В нижней части страницы должны располагаться ссылки для возврата к странице формирования накладной (доступные заготовки).
- Под блоком содержимого накладной должны располагаться кнопки: Сформировать накладную и очистить всю накладную.

## 12 Заключение

В результате курсовой работы была создана информационная система для учёта поставок на склад, написанная на Python и использующая фреймворк Flask. Проект начинался с этапа предварительного планирования, в рамках которого была разработана вся требуемая документация.

В ходе работы над проектом был получен опыт в программировании на Python, JavaScript, а также в верстке на HTML и CSS через разработку клиент-серверного приложения. Также были освоены навыки формулирования и структурирования требований к системе и создания проектной документации.

Созданное приложение обеспечивает пользователям информационной системы возможность взаимодействия с базой данных через веб-интерфейс. Оно предлагает пользователям функционал для осуществления множества сценариев использования, поддерживая бизнес-процессы и интегрируясь с базой данных.

## 13 Список использованной литературы

1. Марков И.С. Конспекты лекций по курсу «Разработка информационных систем». Москва, 2023
2. Пивоварова Н.В. Конспекты лекций по курсу «Базы данных». Москва, 2022.