



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

«Робототехника и комплексная автоматизация»

КАФЕДРА

«Системы автоматизированного проектирования (РК-6)»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОМУ ПРОЕКТУ

НА ТЕМУ:

«Разработка информационной системы склада»

Студент РК6-53Б
(Группа)

(Подпись, дата)

Олейников А.А.
(Фамилия И.О.)

Преподаватель

(Подпись, дата)

Пивоварова Н.В.
(Фамилия И.О.)

2024 г.

РЕФЕРАТ

Пояснительная записка к курсовой работе «Разработка информационной системы склада» с использованием языка программирования Python и фреймворка Flask. включает 38 страниц машинописного текста и 16 иллюстраций. В ходе выполнения работы использовался 1 источник, а также пошагово реализовывались все пункты технического задания.

В пояснительной записке приведено: техническое задание на разработку информационной системы, определение конечных пользователей, варианты использования: выбор варианта использования в главном меню, авторизация пользователя, работа с запросами, работа с отчетами, работа с накладными.

В процессе разработки были созданы: архитектура приложения, проектная документация, схемы базы данных и интерфейсы пользовательских экранов. Система протестирована на соответствие функциональным и нефункциональным требованиям.

Курсовая работа позволила автору приобрести навыки работы с веб-фреймворком Flask, проектирования веб-приложений, взаимодействия с базами данных и подготовки проектной документации.

Ключевые слова: склад, поставки, информационная система, Python, Flask, MySQL.

СОДЕРЖАНИЕ

1. Описание предметной области.....	4
2. Техническое задание.....	5
2.1 Техническое задание на этап проектирования.....	5
2.2 Техническое задание на этап реализации.....	5
3. Проектирование информационной системы	6
3.1 Конечные пользователи информационной системы	6
3.2 UML-диаграмма вариантов использования	7
3.3 Вариант использования "Авторизация пользователя"	8
3.4 Вариант использования "Выбор варианта использования в главном меню"	11
3.5 Вариант использования "Работа с запросами: Выбор запроса в меню запросов"	13
3.6 Вариант использования "Работа с запросами: Выполнение запроса"	15
3.7 Вариант использования "Работа с отчётами"	20
3.8 Вариант использования "Работа с накладными: Выбор варианта использования в меню накладных"	25
3.9 Вариант использования "Работа с накладными: Заполнение накладной" ..	27
3.10 Вариант использования "Работа с накладными: Подтверждение поставки по накладной"	31
4. ER-модель базы данных	36
5. Заключение	37
6. Использованные источники	38

1. Описание предметной области

Поставки на склад.

На складе хранятся товары, разбитые на группы товаров, например канцелярские товары, спортивный инвентарь, бижутерия и т.д.

Все товары, прежде чем их принимают на склад, регистрируются. При этом для каждого товара указывается ID, к какой группе товаров он относится, единицу измерения.

Товары на склад доставляют различные поставщики, о которых известны их уникальный номер, название, из какого они города и название банка, в котором у них находятся счета, номер счета в этом банке.

Поступление товаров на склад оформляется Накладной.

Каждая накладная имеет уникальный номер, содержит общую стоимость всех поставленных по ней товаров и оформляется на одного Поставщика в определенную дату. По каждой накладной можно поставить множество наименований ранее зарегистрированных товаров в любом количестве. После совершения реальной поставки менеджер подтверждает накладную и товары зачисляются в систему.

Цена товара устанавливается поставщиком в момент поставки.

На складе ведется строгий учет запасов товаров. Товары с одинаковым шифром, но поставленные по разным ценам, суммируются отдельно. Для всех запасов товаров указывается дата последнего изменения величины запаса.

2. Техническое задание

2.1 Техническое задание на этап проектирования

Техническое задание на этап проектирования состоит из следующих 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.

2.2 Техническое задание на этап реализации

Техническое задание на этап реализации состоит из следующих задач:

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.

3. Проектирование информационной системы

3.1 Конечные пользователи информационной системы

В данной системе пользователи разделяются на 3 категории:

1. Поставщик (внешний пользователь):

- Авторизация пользователя;
- Выбор варианта использования в главном меню;
- Выбор варианта использования в меню накладных;
- Заполнение накладной;

2. Менеджер (внутренний пользователь):

- Авторизация пользователя;
- Выбор варианта использования в главном меню;
- Выбор варианта использования в меню накладных;
- Подтверждение поставки по накладной;

3. Админ (внутренний пользователь):

- Авторизация пользователя;
- Выбор варианта использования в главном меню;
- Создание отчётов;

4. Директор (внутренний пользователь):

- Авторизация пользователя;
- Выбор варианта использования в главном меню;
- Просмотр отчётов;
- Выбор запроса в меню запросов;
- Выполнение запроса;

3.2 UML-диаграмма вариантов использования

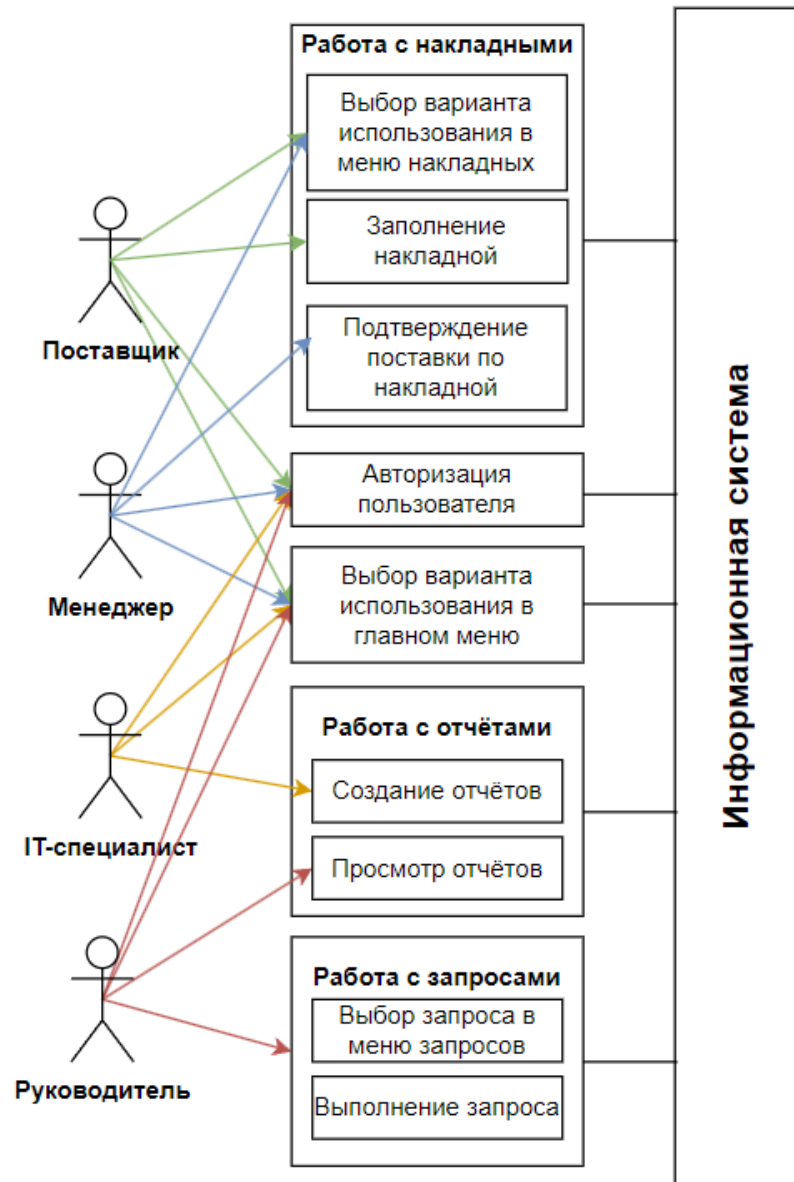


Рис.1 UML-диаграмма вариантов использования

3.3 Вариант использования "Авторизация пользователя"

Карточка варианта:

Название: Авторизация пользователя.

Предусловие: Не авторизованный пользователь пытался получить доступ к системе и был автоматически перенаправлен на этот вариант использования.

Гарантия: Пользователь прошел авторизацию и перешел на страницу доступного ему меню.

Минимальная гарантия: Пользователь получает сообщение об ошибке и может продолжить работу. База данных остается в согласованном состоянии.

Основной успешный сценарий:

1. Не авторизованный пользователь был перенаправлен на страницу авторизации при попытке доступа к системе.
2. Система отправляет форму для ввода данных учётной записи (логин и пароль).
3. Пользователь вводит эти данные, а также указывает является ли он сотрудником и нажимает кнопку "Войти".
4. Система проверяет введенные данные на корректность.
5. Если данные верны, система создает сессию для пользователя и перенаправляет его в Главное меню.

Исключения:

3а Пользователь не ввёл данные или ввел некорректные данные (неверный логин или пароль):

- Система отправляет сообщение об ошибке авторизации и возвращает его к пункту 2 для повторного ввода учетных данных.

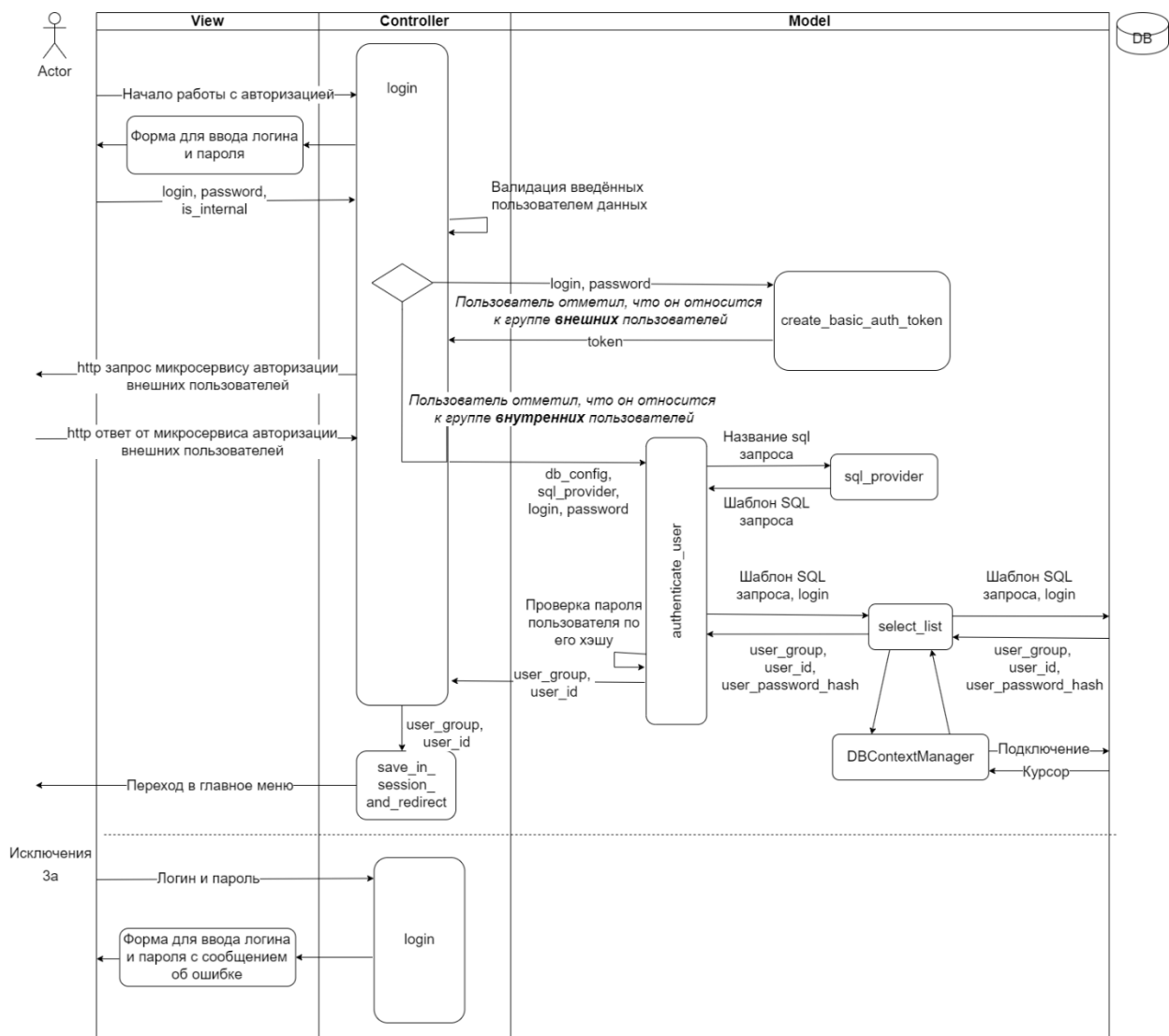


Рис.2 - Системная диаграмма последовательности «Авторизация пользователя»

После авторизации каждый пользователь соотноситься к одной из групп пользователей, описанных выше, и каждая из них имеет собственного пользователя в базе данных со специфицированными правами на выполнение действий в базе данных:

Supplier (Поставщик) - *SELECT, INSERT* – для получения данных о продуктах, сохранения неподтверждённых накладных.

Admin (Админ) – *EXECUTE* для запуска хранимых в базе процедур для генерации ежемесячных отчётов.

Director (Директор) – *SELECT* для работы с параметризованными запросами и хранимыми ежемесячными отчётами.

Manager (Тестировщик) – *SELECT, INSERT, UPDATE* – для получения неподтверждённых накладных, товаров в них и последующего подтверждения.

Файловая архитектура реализации варианта использования

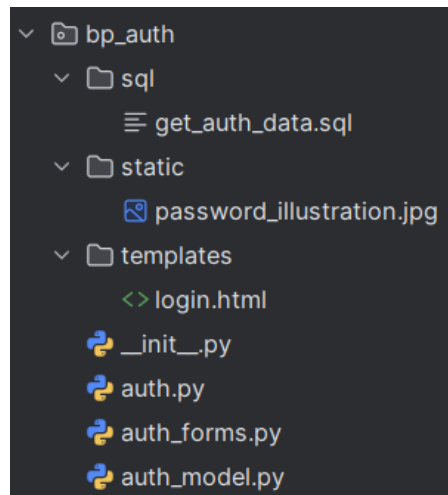


Рис.3 - Файловая архитектура реализации варианта использования «Авторизация пользователя»

Список необходимых шаблонов с требованиями к каждому из них

1. Динамический шаблон *login.html*

Назначение: Страница входа в систему.

Должен содержать:

- Навигационную панель с названием информационной системы.
- Секцию для flash-сообщений об ошибках.
- Форму для ввода данных пользователя:
 - Поле для ввода логина *login*.
 - Поле для ввода пароля *password*.
 - Переключатель типа пользователя (внешний/внутренний) *is internal*.
 - Кнопка "Войти" для отправки формы.
 - Кнопка "Очистить" для сброса формы.

3.4 Вариант использования "Выбор варианта использования в главном меню"

Карточка варианта:

Название: Выбор варианта использования в главном меню.

Предусловие: Пользователь успешно вошёл в систему и авторизовался.

Гарантия: Пользователь получает меню с доступными, согласно его пользовательской группе, действиями.

Минимальная гарантия: Пользователь получает сообщение об ошибке и может продолжить работу. База данных остается в согласованном состоянии.

Основной успешный сценарий:

1. Система присылает пользователю меню с доступными ему, согласно его пользовательской группе, действиями

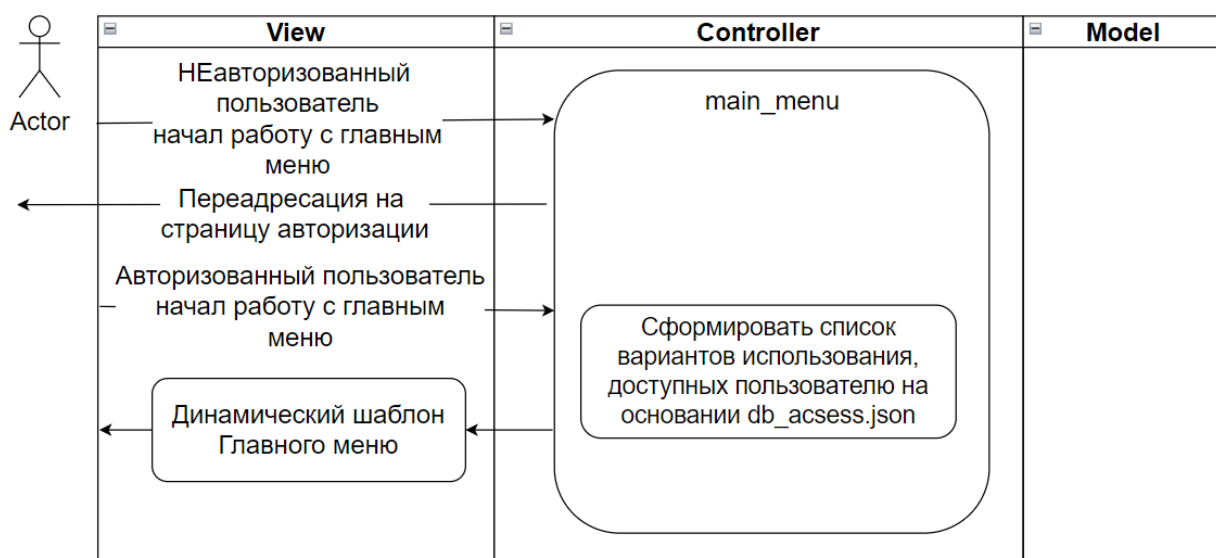


Рис.4 - Системная диаграмма последовательности «Выбор варианта использования в главном меню»

Список необходимых шаблонов с требованиями к каждому из них

1. Динамический шаблон *main_menu.html*

Назначение: Главное меню пользователя.

Должен содержать:

- Навигационную панель, которая содержит:
 - Название информационной системы.
 - Кнопку "ВЫЙТИ", которая переадресует пользователя на де-авторизацию *url_for('bp_auth.logout')*
 - Блок, отображающий ID пользователя *user_id* и группой пользователя *user_group*.
- Секцию для flash-сообщений.
- Блок с доступными вариантами использования:
 - Ряд кнопок, каждая из которых ведёт на доступный пользователю вариант использования системы. Кнопки генерируются динамически по списку *accessible_blueprints*, который содержит словари вида *{'url', 'name'}*.

3.5 Вариант использования "Работа с запросами: Выбор запроса в меню запросов"

Карточка варианта:

Название: Выбор запроса в меню запросов.

Предусловие: Пользователь вошел в систему, успешно авторизовался и выбрал вариант использования "Выбор запроса в меню запросов" при условии, что он имеет доступ к работе с этим вариантом использования.

Гарантия: Пользователь перенаправлен на работу с выбранным запросом.

Минимальная гарантия: Пользователь получает сообщение об ошибке и может продолжить работу. База данных остается в согласованном состоянии.

Основной успешный сценарий:

1. Пользователь начинает сценарий, выбирая вариант использования " Выбор запроса в меню запросов", при условии, что он имеет доступ к работе с этим вариантом использования.
2. Система присылает меню доступных пользователю запросов.
3. Пользователь выбирает запрос и перенаправляется на работу с ним.

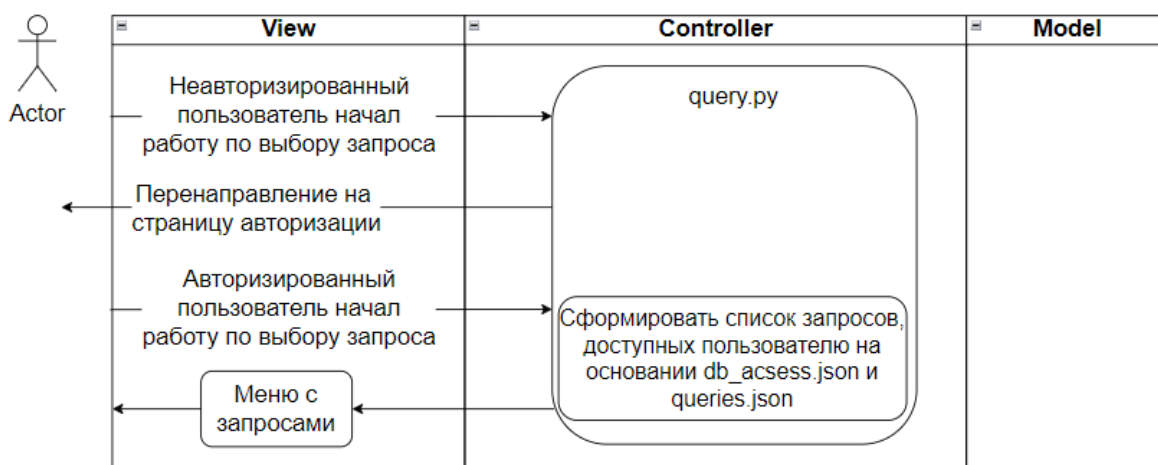


Рис.5 - Системная диаграмма последовательности «Выбор запроса в меню запросов»

Файловая архитектура реализации варианта использования

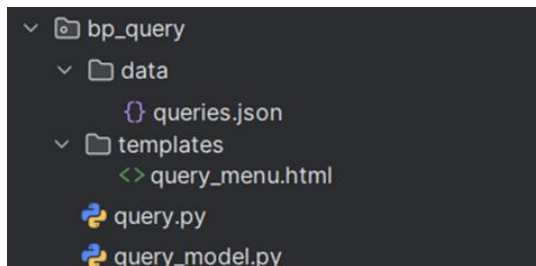


Рис.6 - Файловая архитектура реализации варианта использования «Выбор запроса в меню запросов»

Список необходимых шаблонов с требованиями к каждому из них

2. Динамический шаблон *query_menu.html*

Назначение: Меню запросов.

Должен содержать:

- Навигационную панель, которая содержит:
 - Название информационной системы, является также кнопкой переадресации в главное меню *url_for('main_menu')*
 - Кнопку "ВЫЙТИ", которая переадресует пользователя на деавторизацию *url_for('bp_auth.logout')*
 - Блок, отображающий ID пользователя *user_id* и группой пользователя *user_group*.
- Секцию для flash-сообщений с ошибками.
- Блок с доступными запросами:
 - Ряд кнопок, каждая из которых ведёт на доступный пользователю запрос. Кнопки генерируются динамически по списку *accessible_queries*, который содержит словари вида *{'query_id', 'query_name', 'query_controller'}*.

3.6 Вариант использования "Работа с запросами: Выполнение запроса"

Карточка варианта:

Название: Выполнение запроса.

Предусловие: Пользователь вошел в систему, успешно авторизовался и выбрал запрос в меню запросов, при условии, что он имеет доступ к работе с выбранным запросом запросам.

Гарантия: Пользователь получил результат запроса на экран.

Минимальная гарантия: Пользователь получает сообщение об ошибке и может продолжить работу. База данных остается в согласованном состоянии.

Основной успешный сценарий:

1. Пользователь начинает сценарий, нажимая на соответствующую кнопку в меню запросов.
2. Система присылает форму для ввода параметров запроса.
3. Пользователь вводит параметры и отправляет их системе.
4. Система присылает результат выполнения запроса.

Исключения:

6a Пользователь ввел некорректные данные или не ввел данные вовсе:

Система присылает сообщение об ошибке ввода и возвращает его к пункту 2 для повторного ввода параметров запроса.

6b Результатом запроса оказался пустой *result-set*:

Система присылает сообщение об отсутствии информации по данному запросу.

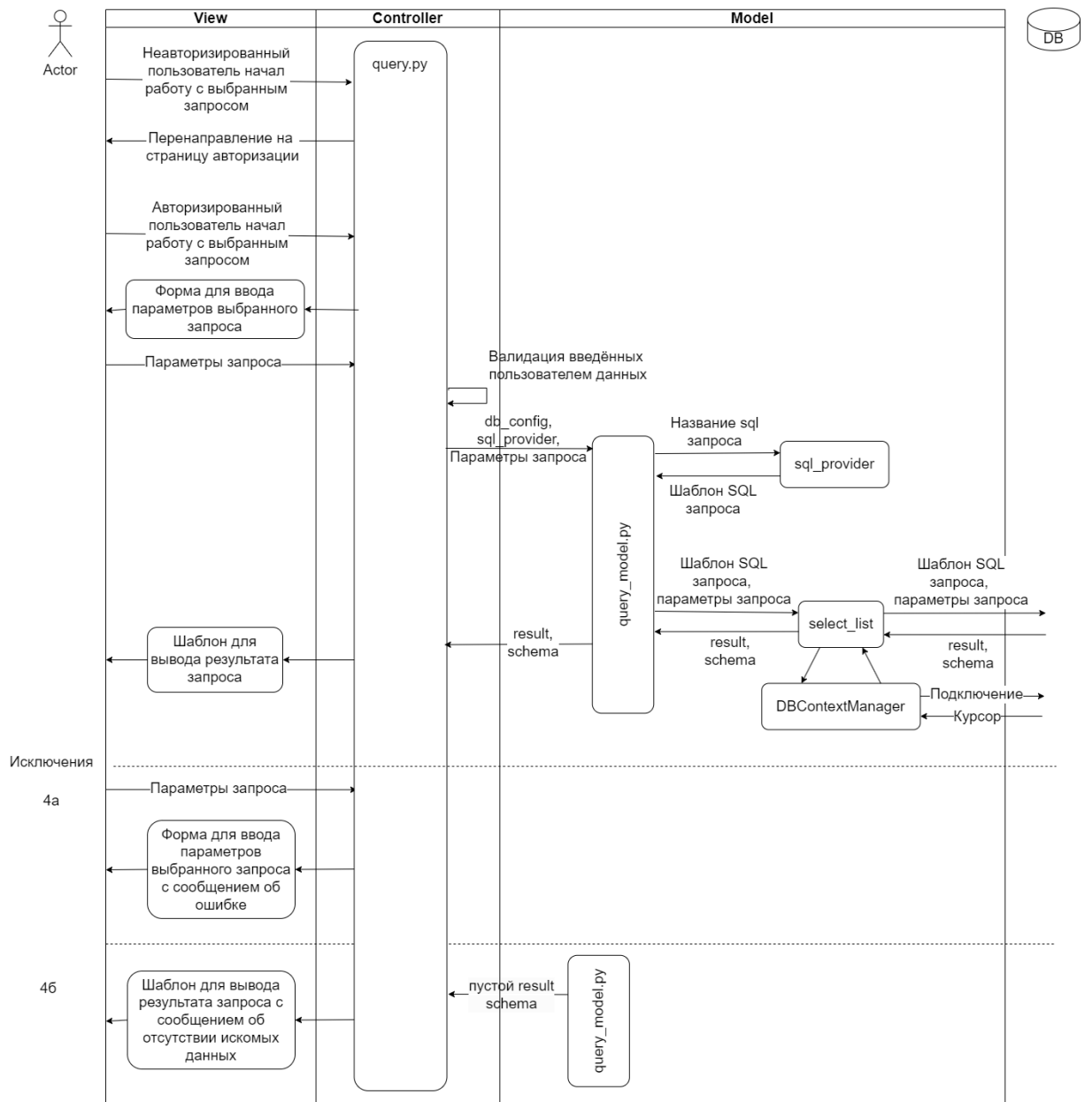


Рис.7 - Системная диаграмма последовательности «Выполнение запроса»

Список необходимых шаблонов с требованиями к каждому из них

1. Динамический шаблон *query_input.html*

Назначение: Ввод параметров запроса.

Должен содержать:

- Навигационную панель, которая содержит:
 - Название информационной системы, является также кнопкой переадресации в главное меню *url_for('main_menu')*
 - Кнопку "ВЫЙТИ", которая переадресует пользователя на де-авторизацию *url_for('bp_auth.logout')*.
 - Блок, отображающий ID пользователя *user_id* и группой пользователя *user_group*.
- Секцию для flash-сообщений.
- Форма ввода параметров:
 - Динамическая форма, генерируемая на основе объекта *form* (предоставленного Flask-WTF):
 - Каждое поле для ввода параметров запроса генерируется с использованием Flask-WTF по атрибутам *field.id* и *field.label.text* путём итерирования по *form* объектом *field*.
 - Если поле содержит ошибку *field.errors*, выводится сообщение об ошибке *field.errors[0]*.
 - Скрытый *query_id* для передачи идентификатора запроса.
 - Скрытый *csrf-token* для защиты от атак (сгенерирован Flask-WTF).
- Кнопка "Войти" для отправки формы на контроллер-источник.
- Кнопка "Очистить" для сброса формы.

2. Динамический шаблон *query_result.html*

Назначение: Отображение результатов выполнения параметризованного запроса.

Должен содержать:

- Навигационную панель, которая содержит:
 - Название информационной системы, является также кнопкой переадресации в главное меню *url_for('main_menu')*
 - Кнопку "ВЕРНУТЬСЯ К ЗАПРОСУ", которая возвращает на страницу запроса *url_for(query_controller)*.
 - Кнопку "МЕНЮ ЗАПРОСОВ", которая переадресует пользователя в меню запросов *url_for("bp_query.query_menu")*.
 - Кнопку "ВЫЙТИ", которая переадресует пользователя на де-авторизацию *url_for('bp_auth.logout')*.
 - Блок, отображающий ID пользователя *user_id* и группой пользователя *user_group*.
- Секцию для flash-сообщений.
- Результаты выполнения запроса:
 - Если данные доступны:
 - Заголовок: отображает название параметризованного запроса (*query_name*).
 - Таблица результатов:
 - Генерируется динамически с помощью функции *render_sortable_searchable_table(headers, results)* из модуля *sort_utils* по спискам *headers, results*.
 - Позволяет сортировать и искать данные.
 - Если данных нет:
 - Сообщение о пустых результатах:
 - Выводится заголовок "Ничего не найдено".
 - Краткое пояснение, что результаты отсутствуют, с рекомендацией изменить критерии поиска.

Файловая архитектура реализации варианта использования

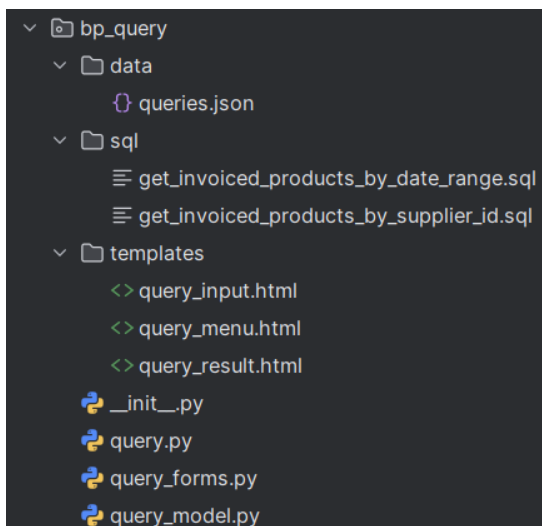


Рис.8 - Файловая архитектура реализации варианта использования «Выполнение запроса»

На текущий момент пользователям доступны запросы:

1. Посмотреть все поставки за период.
2. Посмотреть все поставки поставщика.

Для задания запросов в queries.json используется универсальная структура вида:

```
{
  "query_id": "1",
  "query_name": "Посмотреть все поставки за период",
  "query_controller": "universal_query_controller",
  "query_model": "universal_query_model",
  "query_form": "SearchInvoicedProductsByDateRangeForm",
  "sql": "get_invoiced_products_by_date_range.sql",
  "sql_params": [
    "start_date",
    "end_date"
  ],
  "schema_ru": [
    "id поставки",
    "Дата поставки",
    "id поставщика",
    "id категории продукта",
    "категория продукта",
    "id продукта",
    "продукт"
  ]
}
```

3.7 Вариант использования "Работа с отчётами"

Карточка варианта:

Название: Работа с отчётами.

Предусловие: Пользователь вошел в систему, успешно авторизовался и выбрал вариант использования "Работа с отчётами" в Главном меню при условии, что он имеет доступ к работе с этим вариантом использования.

Гарантия: Пользователь увидит меню отчётов с доступными ему вариантами использования (создать/просмотреть). При создании – будет создан отчёт, при просмотре – будет отображено содержимое отчёта в человекочитаемом формате в виде таблицы.

Минимальная гарантия: Пользователь получает сообщение об ошибке и может продолжить работу. База данных остается в согласованном состоянии.

Основной успешный сценарий:

1. Пользователь начинает сценарий, выбирая вариант использования "Работа с отчётами" в Главном меню.
2. Система присылает меню отчётов с формой для ввода действия, которое хочет совершить пользователь (создать или просмотреть отчёт), отчётного периода и типа отчёта из доступных ему.
3. Пользователь вводит параметры и отправляет их системе.

Если было выбрано «создать отчёт»

4. Система заносит соответствующий отчёт в базу данных и присылает сообщение, что отчёт успешно создан.

Если было выбрано «просмотреть отчёт»

4. Система выводит пользователю отчёт в виде таблицы в человекочитаемом формате.

Исключения:

7а Пользователь ввел некорректные данные или не ввел данные вовсе:

Система присылает сообщение, что поле формы должно быть заполнено, и возвращает его к пункту **2**.

7б Пользователь пытался создать отчёт за уже существующий отчётный период:

Система присылает сообщение, что такой отчёт уже существует, при этом не создаёт данный отчёт в базе данных, и возвращает его к пункту **2**.

7в Пользователь пытался открыть отчёт за несозданный отчётный период:

Система присылает сообщение, что нет данных за выбранный период, и возвращает его к пункту **2**.

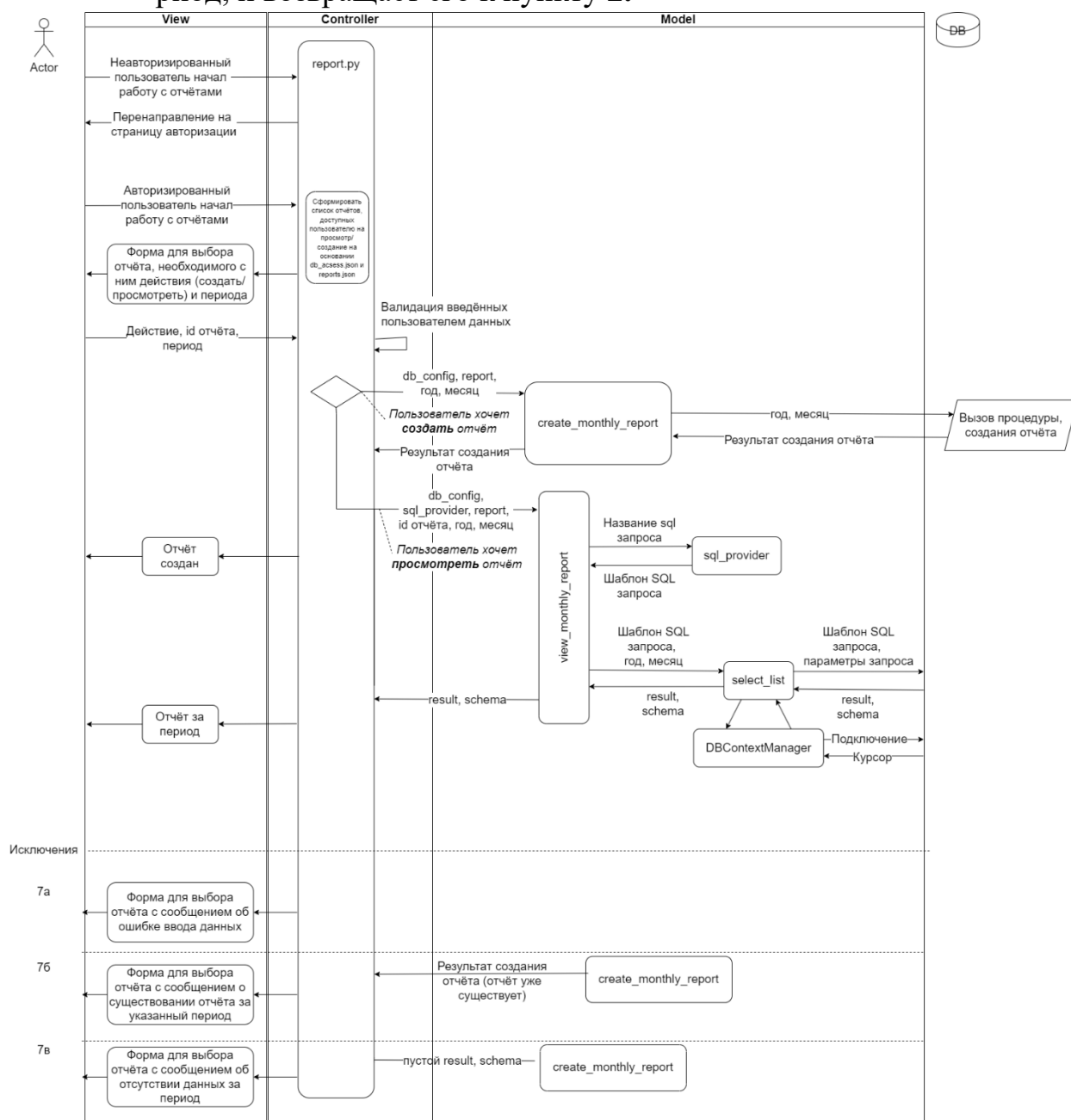


Рис.9 - Системная диаграмма последовательности «Работа с отчётами»

Список необходимых шаблонов с требованиями к каждому из них

1. Динамический шаблон *report_menu.html*

Назначение: Ввод параметров запроса.

Должен содержать:

- Навигационную панель, которая содержит:
 - Название информационной системы, которое является кнопкой переадресации в Главное меню *url_for('main_menu')*.
 - Кнопку "ВЫЙТИ", которая переадресует пользователя на де-авторизацию *url_for('bp_auth.logout')*.
 - Блок, отображающий ID пользователя *user_id* и группой пользователя *user_group*.
- Секцию для flash-сообщений.
- Форма ввода параметров:
 - Динамическая форма, генерируемая на основе объекта *form* (предоставленного Flask-WTF):
 - Каждое поле для ввода параметров отчёта генерируется с использованием Flask-WTF по атрибутам *field.id* и *field.label.text* путём итерирования по *form* объектом *field*.
 - Если поле содержит ошибку *field.errors*, выводится сообщение об ошибке *field.errors[0]*.
 - Динамическая форма содержит:
 - Радиокнопки для выбора действия *action*: "Создать отчёт" или "Просмотреть отчёт". Изменение значения вызывает обновление доступных вариантов в списке отчётов.
 - Выпадающий список *report_type* с доступными отчётами на выбранное действие:
 - Создать – *createChoices*
 - Просмотреть - *viewChoices*

- Поле для ввода года year
 - Поле для выбора месяца из списка month
 - Скрытый csrf-token для защиты от атак (сгенерирован Flask-WTF).
 - Динамические списки (createChoices и viewChoices), используемые для выбора отчётов.
- Скрипт для обновления доступных отчётов:
 - Реализован на JavaScript:
 - Обновляет содержимое выпадающего списка отчётов report_type в зависимости от выбранного действия action.
- Кнопка "Совершить" для отправки формы на контроллер-источник.
- Кнопка "Очистить" для сброса формы.

2. Динамический шаблон report_view.html

Назначение: Отображение результатов отчёта в удобном формате с поддержкой сортировки и поиска.

Должен содержать:

- Навигационную панель, которая содержит:
 - Название информационной системы, которое является кнопкой переадресации в Главное меню url_for('main_menu').
 - Кнопку "МЕНЮ ОТЧЁТОВ", которая переадресует пользователя в меню запросов url_for("bp_report.report_menu").
 - Кнопку "ВЫЙТИ", которая переадресует пользователя на деавторизацию url_for('bp_auth.logout').
 - Блок, отображающий ID пользователя user_id и группой пользователя user_group.
- Секцию для flash-сообщений.
- Таблица с результатами (results), сгенерированной функцией render_sortable_searchable_table(headers, results) из модуля sort_utils. Таблица поддерживает сортировку и поиск.

Файловая архитектура реализации варианта использования

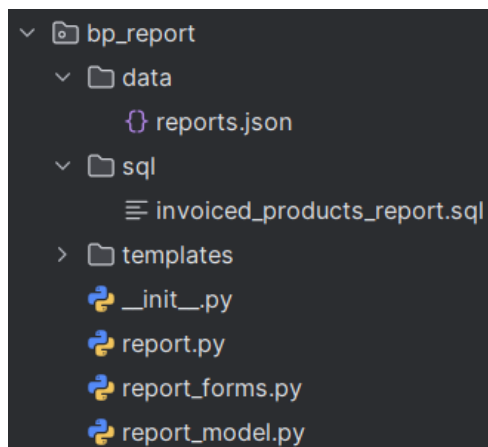


Рис.10 - Файловая архитектура реализации варианта использования «Работа с отчётами»

На текущий момент пользователям доступны отчёты:

1. Месячный отчёт по поставкам продуктов

Для задания отчётов в reports.json используется универсальная структура вида:

```
[
  {
    "rep_id": "1",
    "rep_name": "Месячный отчёт по поставкам продуктов",
    "proc_name": "generate_invoiced_products_report",
    "proc_table": "invoiced_products_report",
    "sql_get": "invoiced_products_report.sql",
    "sql_check": "invoiced_products_report_check.sql",
    "schema": [
      "id категории продукта",
      "категория продукта",
      "id продукта",
      "продукт",
      "количество поставленных единиц",
      "средняя цена поставки"
    ]
  }
]
```


3.8 Вариант использования "Работа с накладными: Выбор варианта использования в меню накладных"

Карточка варианта:

Название: Выбор варианта использования в меню накладных.

Предусловие: Пользователь вошел в систему, успешно авторизовался и выбрал вариант использования "Выбор варианта использования в меню накладных" в Главном меню при условии, что он имеет доступ к работе с этим вариантом использования.

Гарантия: Пользователь получит меню действий с накладными согласно его правам доступа.

Минимальная гарантия: Пользователь получает сообщение об ошибке и может продолжить работу. База данных остается в согласованном состоянии.

Основной успешный сценарий:

1. Пользователь начинает сценарий, выбирая вариант использования "Выбор варианта использования в меню накладных" в Главном меню.
2. Система присылает пользователю меню с доступными ему, согласно его пользовательской группе, действиями с накладными.

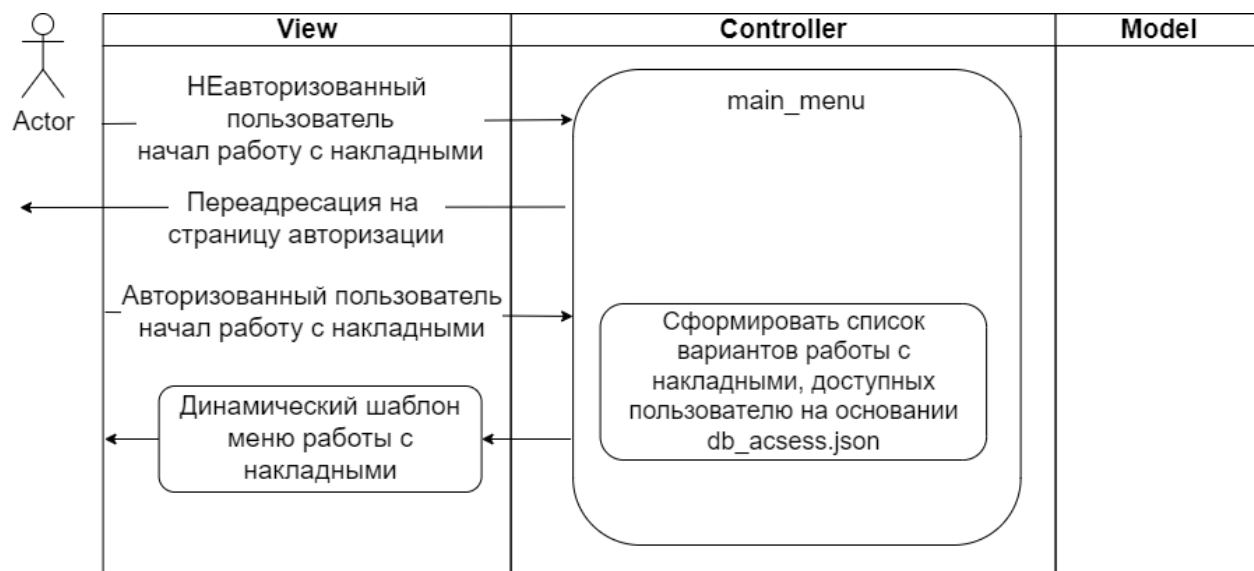


Рис.11 - Системная диаграмма последовательности «Выбор варианта использования в меню накладных».

Список необходимых шаблонов с требованиями к каждому из них

1. Динамический шаблон *main_menu.html*

Назначение: Главное меню пользователя.

Должен содержать:

- Навигационную панель, которая содержит:
 - Название информационной системы.
 - Кнопку "ВЫЙТИ", которая переадресует пользователя на де-авторизацию *url_for('bp_auth.logout')*
 - Блок, отображающий ID пользователя *user_id* и группой пользователя *user_group*.
- Секцию для flash-сообщений.
- Блок с доступными вариантами использования:
 - Ряд кнопок, каждая из которых ведёт на доступный пользователю вариант работы с накладными. Кнопки генерируются динамически по списку *available_actions*, который содержит словари вида *{'url', 'name'}*.

3.9 Вариант использования "Работа с накладными: Заполнение накладной"

Карточка варианта:

Название: Заполнение накладной о поставке.

Предусловие: Пользователь вошел в систему, успешно авторизовался и выбрал вариант использования "Заполнение накладной" в меню работы с накладными при условии, что он имеет доступ к работе с этим вариантом использования.

Гарантия: Пользователь создаст неподтвержденную накладную, которая будет сохранена в базе данных.

Минимальная гарантия: Пользователь получает сообщение об ошибке и может продолжить работу. База данных остается в согласованном состоянии.

Основной успешный сценарий:

1. Пользователь начинает сценарий, выбирая вариант использования "Заполнение накладной" в меню работы с накладными.
2. Система присылает меню товаров, зарегистрированных в системе.
3. Пользователь выбирает товар, который хочет добавить в накладную, количество поставляемого товара и цену за единицу. Нажимает "Сохранить накладную".
4. Система сохраняет накладную в базе данных и присылает сообщение пользователю об успешном совершении операции.

Исключения:

9a Пользователь не выбрал ни одного товара или не указал количество/стоимость за единицу:

Система присылает сообщение об ошибке ввода и возвращает его к пункту 3 для продолжения ввода информации в накладную.

Список необходимых шаблонов с требованиями к каждому из них

1. Динамический шаблон *make_invoice.html*

Назначение: отображение интерфейса выбора товаров для составления накладной.

Должен содержать:

- Навигационную панель, которая содержит:
 - Название информационной системы, которая переадресует пользователя в главное меню *url_for('main_menu')*.
 - Кнопку "ВЕРНУТЬСЯ В МЕНЮ ПОСТАВОК", которая перенаправляет пользователя в меню поставок (*url_for('bp_supply.supply_menu')*).
 - Кнопку "ВЫЙТИ", которая переадресует пользователя на деавторизацию *url_for('bp_auth.logout')*.
 - Блок, отображающий ID пользователя *user_id* и группой пользователя *user_group*.
- Секцию для flash-сообщений.
- Секцию "Возможные к поставке товары":
 - Поле ввода, позволяющее пользователю вводить текст для поиска товаров в таблице.
 - Таблица отображает данные о товарах *products* с поддержкой сортировки. Колонки:
 - ID товара. (*product.product_id*)
 - Название товара. (*product.product_name*)
 - Категория товара. (*product.product_category_name*)
 - Единицы измерения. (*product.product_units_measure*)
- Секцию "Товары в накладной":
 - Поле ввода, позволяющее пользователю вводить текст для поиска товаров в таблице.
 - Таблица отображает данные о товарах в накладной *products* с поддержкой сортировки. Колонки:

- ID товара. (*product.product_id*)
 - Название товара. (*product.product_name*)
 - Категория товара. (*product.product_category_name*)
 - Единицы измерения. (*product.product_units_measure*)
 - Цена и количество (*invoiced_products.price* и *invoiced_products.count*)
- Таблицы реализованы с возможностью сортировки каждой колонки, используя JavaScript-функции из модуля *sort_utils*.
- Форма выбора товара в накладную и удаления из накладной:
 - В каждой строке таблицы колонка "ID" содержит кнопку. Нажатие на кнопку отправляет POST-запрос с ID (*product.product_id*) выбранного товара в обработчик *url_for('bp_supply.update_invoice')* и *url_for('bp_supply.delete_invoice_product')*.

Файловая архитектура реализации варианта использования

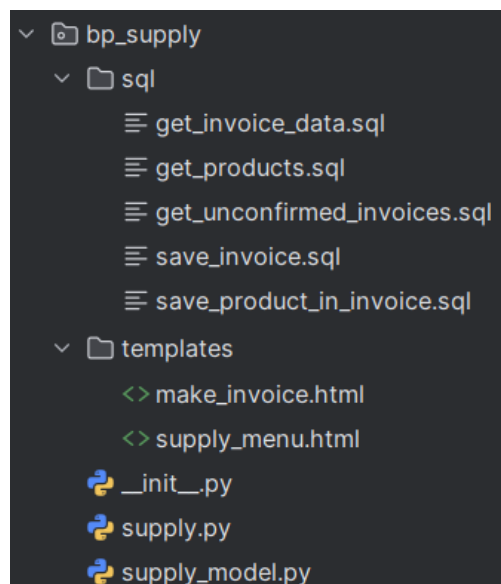


Рис.12 - Файловая архитектура реализации варианта использования «Заполнение накладной о поставке»

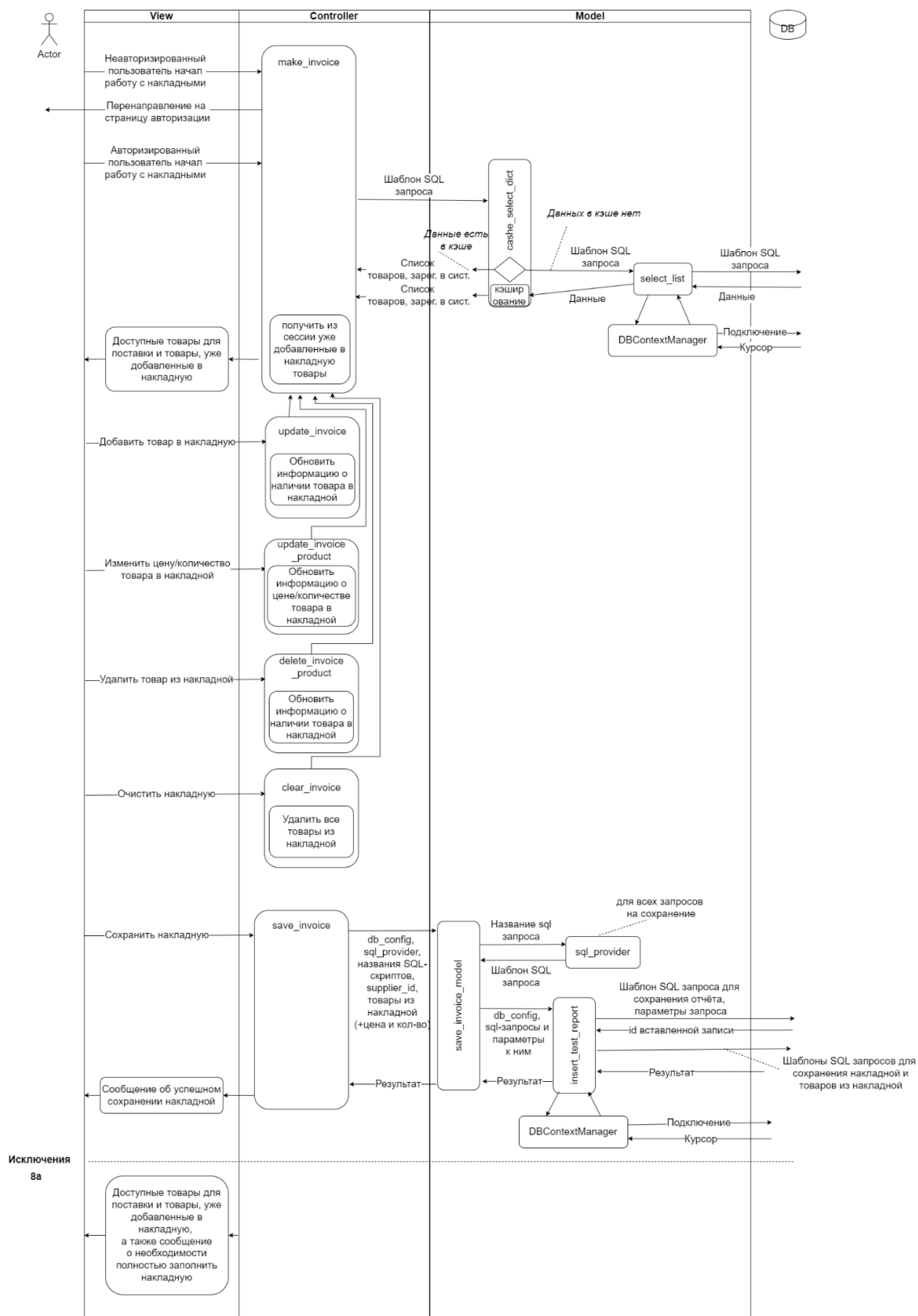


Рис.13 - Системная диаграмма последовательности «Заполнение накладной о поставке»

3.10 Вариант использования "Работа с накладными: Подтверждение поставки по накладной"

Карточка варианта:

Название: Подтверждение поставки по накладной.

Предусловие: Пользователь вошел в систему, успешно авторизовался и выбрал вариант использования "Подтверждение поставки по накладной" в меню работы с накладными при условии, что он имеет доступ к работе с этим вариантом использования.

Гарантия: Пользователь подтверждает ранее созданную накладную .

Минимальная гарантия: Пользователь получает сообщение об ошибке и может продолжить работу. База данных остается в согласованном состоянии.

Основной успешный сценарий:

1. Пользователь начинает сценарий, выбирая вариант использования "Подтверждение поставки по накладной" в меню работы с накладными.
2. Система присылает список неподтверждённых накладных, зарегистрированных в системе.
3. Пользователь выбирает накладную, которую хочет просмотреть.
4. Система присылает страницу с полным списком поставляемых товаров по накладной, их количеством и стоимостью.
5. Пользователь нажимает "Подтвердить поставку".
6. Система обновляет статус накладной и количество остатков товара в базе данных и присылает сообщение пользователю об успешном совершении операции.

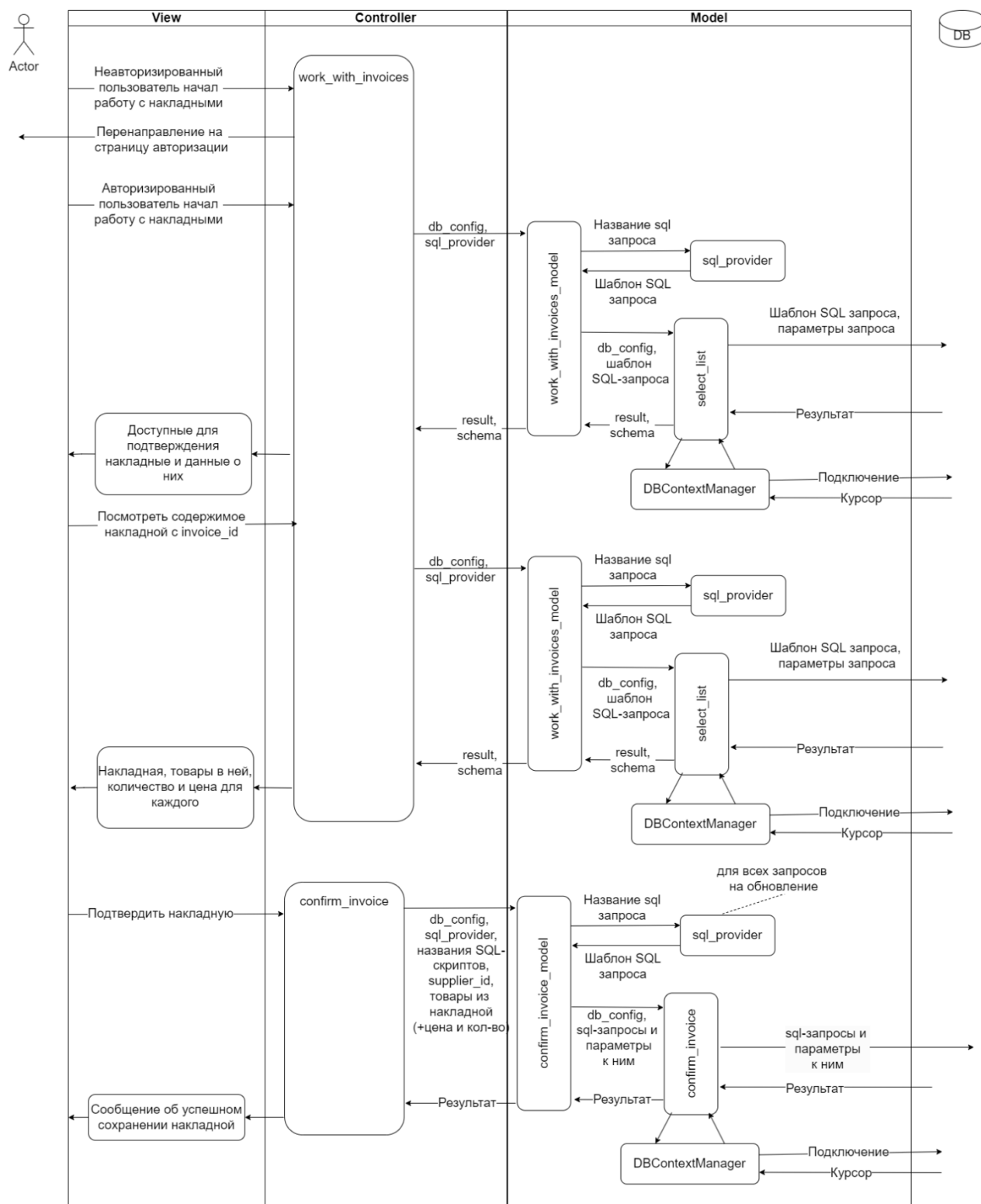


Рис.14 - Системная диаграмма последовательности «Подтверждение поставки по накладной»

Список необходимых шаблонов с требованиями к каждому из них

1. Динамический шаблон *work_with_invoices.html*

Назначение: отображение интерфейса выбора накладной для дальнейшего подтверждения.

Должен содержать:

- Навигационную панель, которая содержит:
 - Название информационной системы, которая переадресует пользователя в главное меню *url_for('main_menu')*.
 - Кнопку "ВЕРНУТЬСЯ В МЕНЮ ПОСТАВОК", которая перенаправляет пользователя в меню поставок (*url_for('bp_supply.supply_menu')*).
 - Кнопку "ВЫЙТИ", которая переадресует пользователя на деавторизацию *url_for('bp_auth.logout')*.
 - Блок, отображающий ID пользователя *user_id* и группой пользователя *user_group*.
- Секцию для flash-сообщений.
- Секцию "Неподтверждённые накладные":
 - Поле ввода, позволяющее пользователю вводить текст для поиска накладных в таблице.
 - Таблица отображает данные о накладных *invoices* с поддержкой сортировки. Колонки:
 - ID накладной. (*invoice[0]*)
 - Дата поставки. (*invoice[1]*)
 - Общая стоимость. (*invoice[2]*)
 - ID поставщика. (*invoice[3]*)
 - Телефон поставщика. (*invoice[4]*)
 - Город поставщика. (*invoice[5]*)
 - Фамилия поставщика. (*invoice[6]*)
 - Имя поставщика. (*invoice[7]*)
 - Отчество поставщика. (*invoice[8]*)

- Название банка поставщика. (invoice[9])
- Номер счёта в банке поставщика. (invoice[10])
- Форма выбора накладной:
 - В каждой строке таблицы колонка "ID накладной" содержит кнопку. Нажатие на кнопку отправляет POST-запрос с ID (invoice[0]) выбранной накладной в обработчик url_for('bp_supply.work_with_invoices').

2. Динамический шаблон *confirm_invoice.html*

Назначение: отображение списка товаров в накладной.

Должен содержать:

- Навигационную панель, которая содержит:
 - Название информационной системы, которая переадресует пользователя в главное меню url_for('main_menu').
 - Кнопку "ВЕРНУТЬСЯ В МЕНЮ ПОСТАВОК", которая перенаправляет пользователя в меню поставок (url_for('bp_supply.supply_menu')).
 - Кнопку "ВЫЙТИ", которая переадресует пользователя на де-авторизацию url_for('bp_auth.logout').
 - Блок, отображающий ID пользователя user_id и группой пользователя user_group.
- Секцию для flash-сообщений.
- Секцию "Товары в накладной":
 - Поле ввода, позволяющее пользователю вводить текст для поиска накладных в таблице.
 - Таблица отображает данные о накладных *invoices* с поддержкой сортировки. Колонки:
 - ID продукта. (invoice_product[0])
 - Название продукта. (invoice_product[1])
 - ID категории продукта (invoice_product[2])

- Название категории продукта. (invoice_product /3/)
 - Количество. (invoice_product /4/)
 - Единица измерения. (invoice_product /5/)
 - Цена. (invoice_product /6/)
- Кнопку “Подтвердить накладную”, которая отправляет POST запрос с invoice_id на обработчик url_for(‘bp_supply.confirm_invoice’)

Файловая архитектура реализации варианта использования

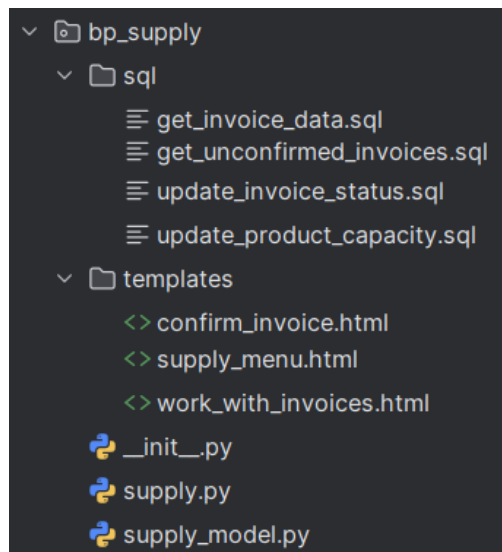


Рис.15 - Файловая архитектура реализации варианта использования «Подтверждение поставки по накладной»

4. ER-модель базы данных

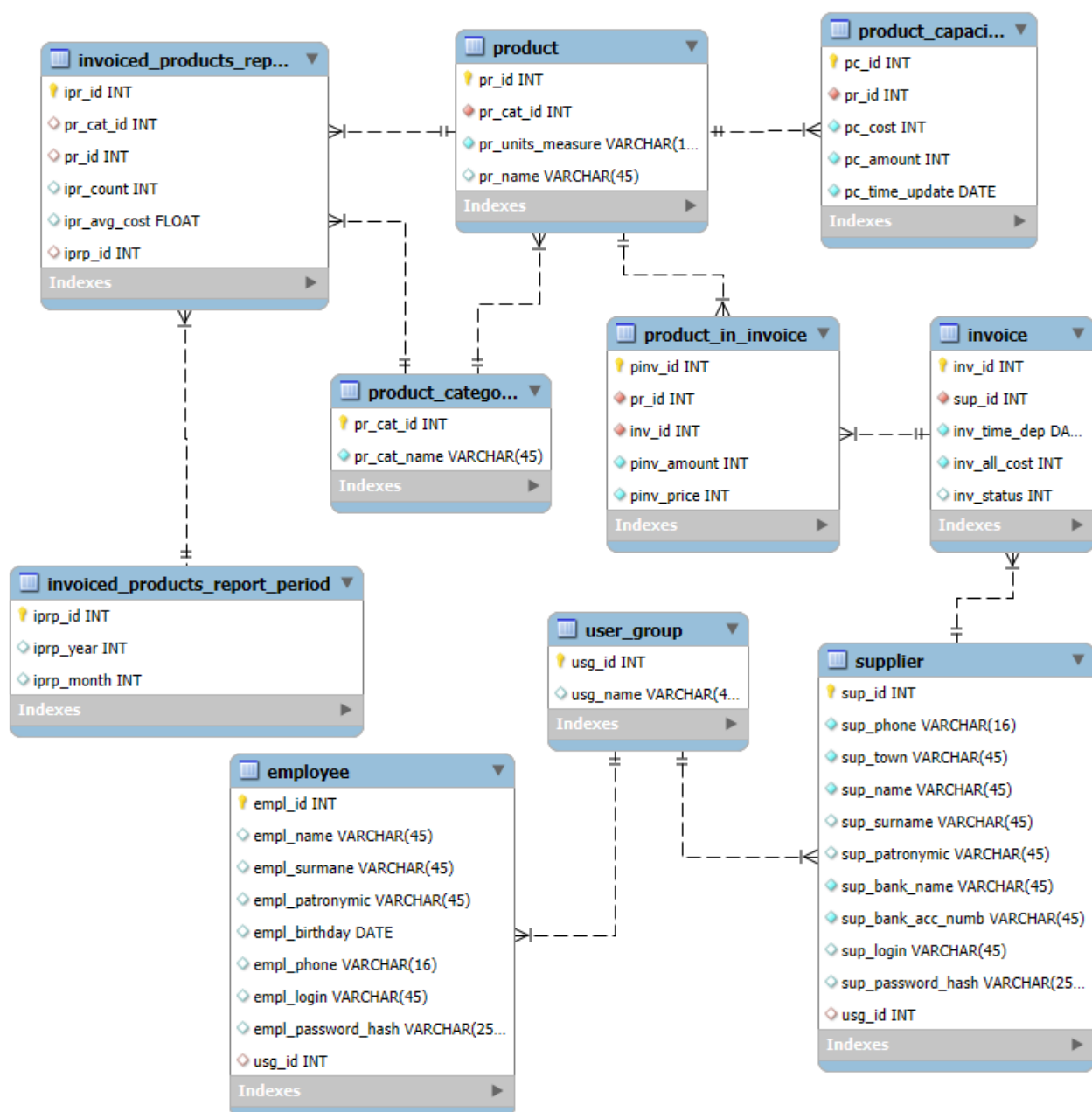


Рис.16 - ER-модель базы данных

5. Заключение

В ходе выполнения курсовой работы была создана информационная система «Склад». Для работы с системой разработан веб-интерфейс на языке программирования Python с использованием фреймворка Flask. В процессе разработки выполнено предварительное проектирование, составлена вся необходимая документация и проработана архитектура приложения.

Система охватывает ключевые аспекты бизнес-процесса. В приложении реализована система авторизации с разграничением прав доступа, что обеспечивает безопасную и эффективную работу с данными. Среди функциональных возможностей системы были разработаны следующие варианты использования: выбор варианта использования в главном меню, авторизация пользователя, работа с запросами, работа с отчетами, работа с накладными.

Созданное приложение предоставляет конечным пользователям удобный веб-ориентированный интерфейс для взаимодействия с базой данных, упрощая процессы работы и анализа. В процессе выполнения курсовой работы приобрел навыки программирования на Flask, проектирования веб-приложений и подготовки проектной документации. Полученный опыт может быть применен в будущей профессиональной деятельности для разработки аналогичных систем.

6. Используемые источники

1. Пивоварова Н. В., Марков И. С. Базы данных. Курсовая работа: учебно-методическое пособие / Пивоварова Н. В., Марков И. С. [Электронный ресурс] // Библиотека МГТУ им. Н. Э. Баумана: [сайт]. — URL: <https://press.bmstu.ru/catalog/item/7744/> 20.12.2023).