|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

**Расчетно-пояснительная записка к курсовому проекту**

по дисциплине: «Базы данных»

|  |  |  |
| --- | --- | --- |
| Студент |  | Долженко Анастасия Тимофеевна |
| Группа |  | РК6-52Б |
| Тема курсовой работы |  | Разработка информационной системы «Интернет-провайдер» |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Долженко А.Т.\_\_\_\_\_**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пивоварова Н. В.**

*подпись, дата фамилия, и.о.*

*Москва, 2024 г.*

# **Аннотация**

Курсовая работа посвящена разработке информационной системы «Интернет-провайдер».

# **Оглавление**

[**Аннотация** 2](#_Toc186268088)

[**Оглавление** 3](#_Toc186268089)

[**Техническое задание** 4](#_Toc186268090)

[**Описание предметной области** 5](#_Toc186268091)

[**Определение конечных пользователей** 5](#_Toc186268092)

[**UML-диаграмма вариантов использования** 5](#_Toc186268093)

[**Стартовая страница** 6](#_Toc186268094)

[**Главное меню** 8](#_Toc186268095)

[**Авторизация** 9](#_Toc186268096)

[**Работа с запросами** 12](#_Toc186268097)

[**Работа с отчётами** 16](#_Toc186268098)

[**Работа с услугами** 21](#_Toc186268099)

[**Личный кабинет** 24](#_Toc186268100)

[**Инфологическая модель** 27](#_Toc186268101)

[**Логическая модель** 28](#_Toc186268102)

# **Техническое задание**

В качестве исходных данных студенты используют описание предметной области, полученное в качестве варианта для лабораторных работ в 4 семестре 2024 года по дисциплине Базы данных.

Техническое задание на этап 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.

Техническое задание на этап реализации состоит из следующих обязательных пунктов:

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

# **Описание предметной области**

Сайт интернет-провайдера для работы с клиентами, заключившими договор.

Интернет-провайдер предоставляет своим клиентам услуги, которыми они пользуются. Клиенты могут пополнять свой баланс и подключать, и отключать услуги из каталога. За каждую подключенную услугу с клиента ежедневно списывается с баланса стоимость услуги.

За работой компании следят менеджер и администратор. Администратор может работать с запросами в базу данных, а также создавать и просматривать отчёты по работе компании. Менеджер может просматривать созданные отчёты.

# **Определение конечных пользователей**

В данной информационной системе в качестве конечных пользователей можно выделить:

* Клиент
* Менеджер
* Администратор

# **UML-диаграмма вариантов использования**

Варианты использования информационной системы (рис. 1):

* Авторизация
* Работа с услугами
* Работа с запросами
* Работа с отчётами
* Личный кабинет

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 1- UML-диаграмма вариантов использования

# **Стартовая страница**

Пользователь попадает на стартовую страницу если, если данных о пользователе нет в сессии. Стартовая страница содержит в себе содержит пункт авторизации. Если пользователь авторизован, то есть данные о нём есть в сессии, пользователь попадает в главное меню. При запуске системы управление передается контроллеру стартовой страницы.

**Предусловия:** пользователь открыл страницу с информационной системой

**Гарантия:** при наличии прав на использование выбранного пункта меню

пользователь переходит на страницу соответствующего варианта использования

**Минимальная гарантия:** пользователь получает сообщение о том, что он не авторизован

**Сценарий работы стартовой страницы:**

1. Пользователь запускает сценарий
2. Система присылает стартовую страницу
3. Пользователь выбирает пунктов авторизации
4. Система передает управление контроллеру пункта авторизации

**Исключения: -**

**Требования к шаблонам:**

1. Статический шаблон «Стартовая страница». Шаблон показывает информацию о том, что для использования системы необходимо авторизоваться, и кнопку со ссылкой на авторизацию.

Шаблон содержит ссылки:

* на контроллер авторизации (адрес ‘/auth’)

**Системная диаграмма последовательностей, соответствующая MVC-паттерну (рис. 2):**

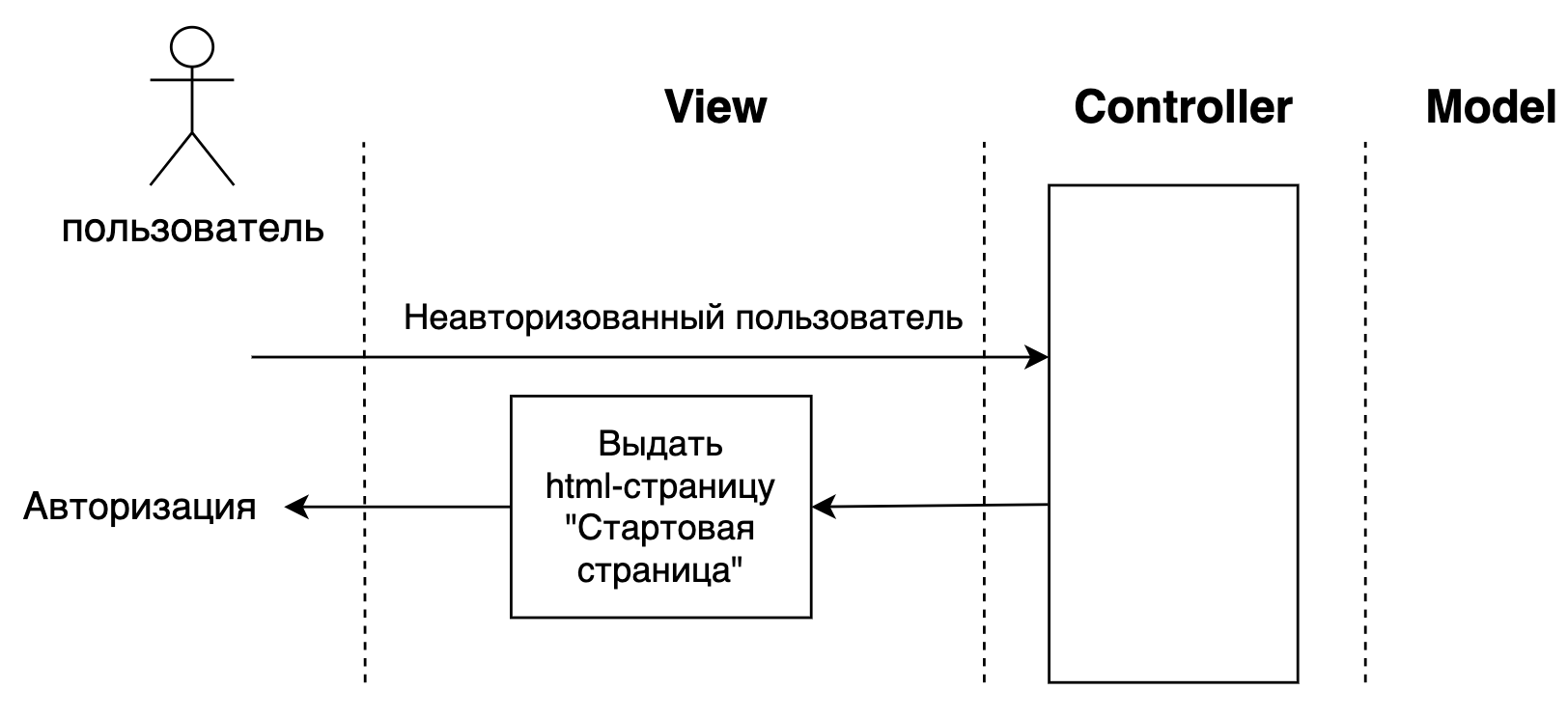


Рисунок 2 - Системная диаграмма последовательностей для варианта использования «Стартовая страница» для неавторизованного пользователя

# **Главное меню**

Если пользователь авторизован, то есть данные о нём есть в сессии, пользователь попадает в главное меню. Пункты главного меню содержат все варианты использования кроме авторизации и пункт для выхода из системы.

**Предусловия:** пользователь открыл страницу с информационной системой

**Гарантия:** при наличии прав на использование выбранного пункта меню пользователь переходит на страницу соответствующего варианта использования

**Минимальная гарантия:** пользователь получает сообщение о том, что у него отсутствуют права на использование выбранного пункта меню

**Пункты главного меню:**

* Работа с услугами («Подключить или отключить услугу»)
* Работа с отчётами
* Работа с запросами
* Выход из системы («Выйти»)

**Сценарий работы главного меню:**

1. Пользователь запускает сценарий
2. Система присылает главное меню
3. Пользователь выбирает один из пунктов вариантов использования
4. Система передает управление контроллеру соответствующего варианта использования

**Исключения:**

1. Пользователь выбирает вариант использования, недоступный для него. Система выводит сообщение об отказе в доступе.

**Требования к шаблонам:**

1. Статический шаблон «Главное меню». Шаблон показывает главное меню со всеми пунктами вариантов использования и пункт для выхода из системы, а также приветственное сообщение для пользователя.

Шаблон содержит ссылки:

* на контроллер авторизации (адрес ‘/auth’)
* на контроллер работы с запросами (адрес ‘/param\_query)
* на контроллер работы с отчетами (адрес ‘/report’)
* на контроллер работы с услугами (адрес ‘/services)
* на выход из системы (адрес ‘/exit’)

**Системная диаграмма последовательностей, соответствующая MVC-паттерну (рис. 3):**

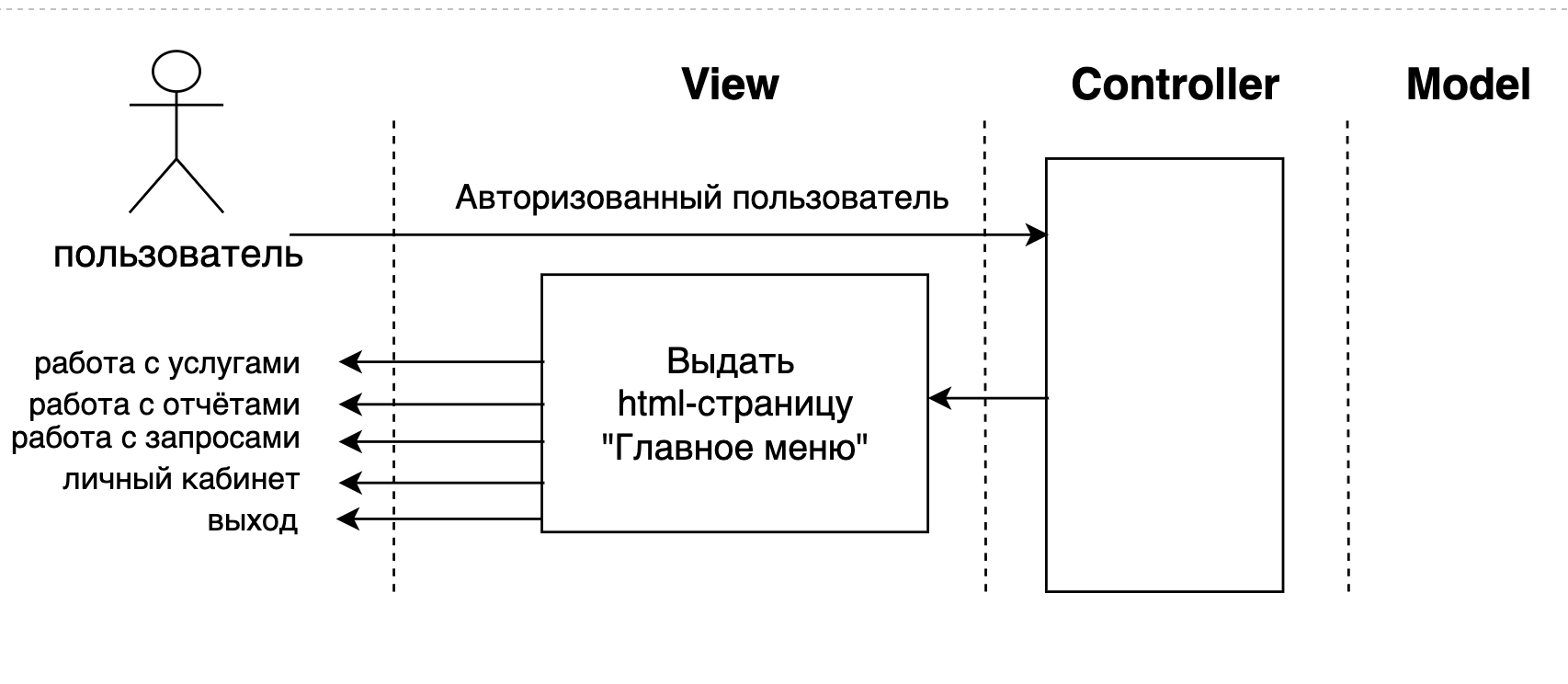


Рисунок 3 - Системная диаграмма последовательностей для варианта использования «Главное меню» для авторизованного пользователя

# **Авторизация**

**Предусловия:** пользователь получил форму для ввода логина, пароля и информации о том, является ли пользователь сотрудником (внутренним пользователем)

**Гарантия:** при корректных данных пользователь авторизовывается в системе и получает права доступа, соответствующие своей группе

**Минимальная гарантия:** пользователь получает сообщение о неверно введенных данных

**Сценарий работы для авторизации:**

* Пользователь запускает сценарий
* Система присылает форму ввода логина, пароля и информации о том, является ли пользователь сотрудником (внутренним пользователем)
* Пользователь осуществляет ввод данных
* Система присылает страницу главного меню, на которой появляется поле с названием группы, присвоенной пользователю.

**Исключения:**

1. Пользователь вводит неверные данные.  
   Система возвращает пользователя на страницу для данных и выводит сообщение о том, что введенные данные не верны.

**Требования к шаблонам:**

1. Статический шаблон для ввода логина, пароля и информации о том, является ли пользователь сотрудником. Шаблон показывает форму для ввода данных и пункт для возврата на стартовую страницу.

Шаблон содержит ссылки:

* для возврата на стартовую страницу (адрес ‘/’)

Шаблон содержит поля для ввода данных:

* поле типа text и именем login для ввода логина
* поле типа password и именем password для ввода пароля
* поле типа checkbox и именем isWorker для ввода информации о том, является ли пользователь сотрудником
* поле типа submit для отправки данных

**Системная диаграмма последовательностей, соответствующая MVC-паттерну для авторизации (рис. 4):**

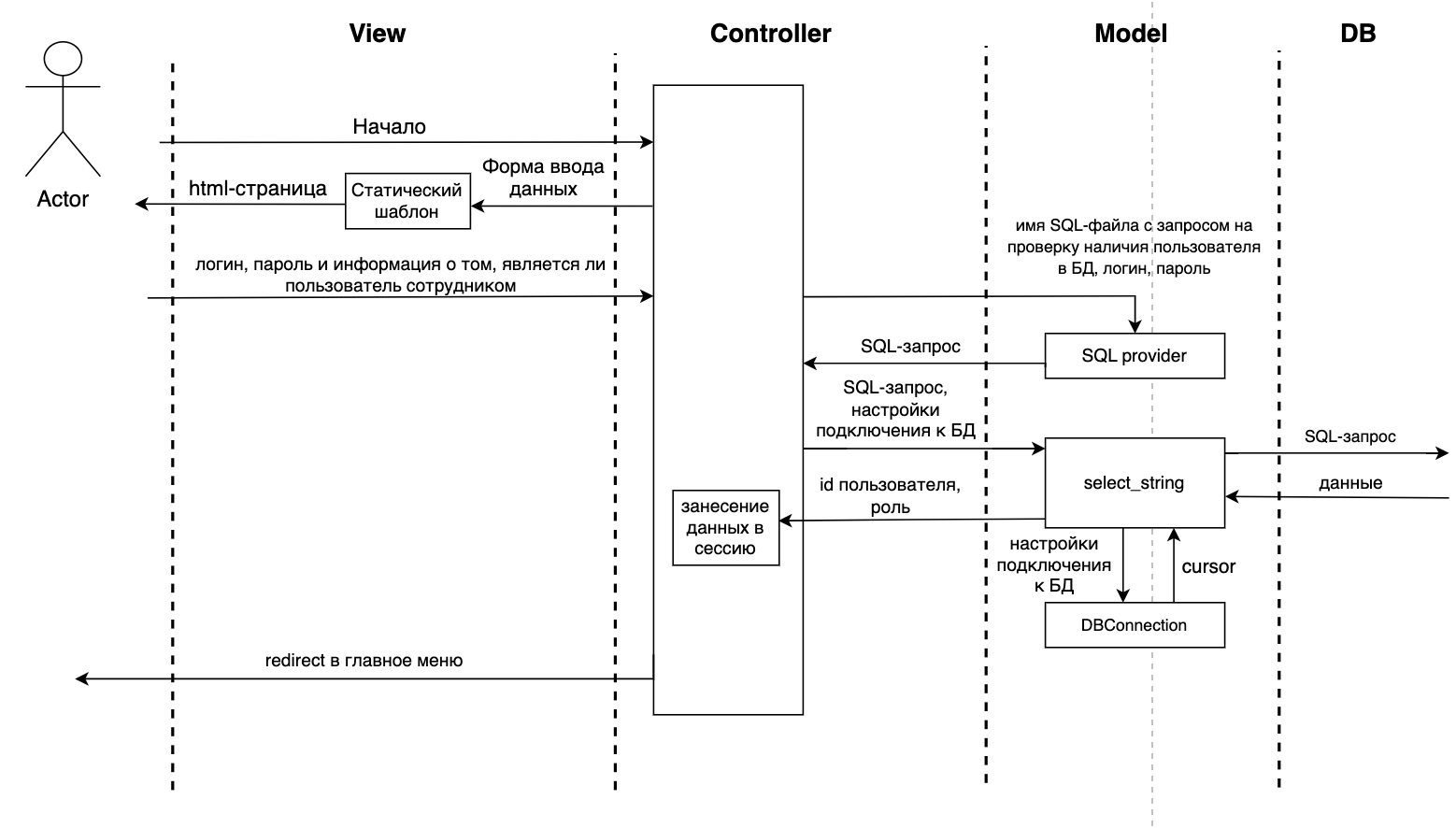
****

Рисунок 4 - Системная диаграмма последовательностей для варианта использования «Авторизация»

**Файловая архитектура:**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 5 - Файловая архитектура для варианта использования "Авторизация"

# **Работа с запросами**

**Предусловия:** пользователь успешно авторизовался и у него есть доступ к этому варианту использования.

**Гарантия:** при корректных параметрах запроса пользователь получает результат выполненного запроса на экране

**Минимальная гарантия:** пользователь получает сообщение о неправильно введенных параметрах запроса или о не полных введенных данных

**Сценарий работы для работы с запросами:**

1. Пользователь запускает сценарий
2. Система присылает меню запросов
3. Пользователь выбирает запрос
4. Система присылает форму для ввода параметров запроса
5. Пользователь вводит параметры запроса
6. Система выполняет запрос и присылает пользователю страницу с результатами

**Исключения: -**

1. Пользователь вводит неверные данные или вводит данные не полностью. Система возвращает пользователя на страницу для ввода параметров запроса и выводит соответствующее информационное сообщение.

**Требования к шаблонам:**

1. Статический шаблон «Меню запросов». Шаблон показывает меню запросов со всеми пунктами – вариантами запросов и кнопку для возврата в главное меню.

Шаблон содержит ссылки:

* На обработчик запроса «Найти услуги по диапазону цен» (адрес ‘/param\_query/find\_services\_by\_cost\_range’)
* На обработчик запроса «Найти услуги по диапазону цен» (адрес ‘/param\_query/ get\_order\_history)
* На возврат в главное меню (адрес ‘/’)

1. Статический шаблон ввода параметров для запроса «Найти услуги по диапазону цен». Шаблон показывает форму для ввода параметров запроса и кнопки для возврата в меню запросов и в главное меню.

Шаблон содержит поля для ввода данных:

* поле типа text и именем low\_lim для нижней границы диапазона цен
* поле типа text и именем high\_lim для верхней границы диапазона цен
* поле типа submit для отправки данных

Шаблон содержит ссылки:

* На возврат в меню запросов (адрес ‘/param\_query’)
* На возврат в главное меню (адрес ‘/’)

1. Статический шаблон для вывода результата запроса запроса «Найти услуги по диапазону цен». Шаблон показывает результаты запроса и кнопки для возврата в меню запросов и главное меню. Вывод результатов осуществляется в виде таблицы. Результаты запросов передаются в шаблон.

Шаблон содержит таблицу с колонками:

* Название услуги
* Цена

1. Статический шаблон ввода параметров для запроса «Получить историю заказов клиента». Шаблон показывает форму для ввода параметров запроса и кнопки для возврата в меню запросов и в главное меню.

Шаблон содержит поля для ввода данных:

* поле типа text и именем num\_d для номера договора клиента
* поле типа submit для отправки данных

Шаблон содержит ссылки:

* На возврат в меню запросов (адрес ‘/param\_query’)
* На возврат в главное меню (адрес ‘/’)

1. Статический шаблон для вывода результата запроса запроса «Получить историю заказов клиента». Шаблон показывает результаты запроса и кнопки для возврата в меню запросов и главное меню. Вывод результатов осуществляется в виде таблицы. Результаты запроса передаются в шаблон.

Шаблон содержит таблицу с колонками:

* Номер заказа
* Номер договора клиента
* Дата заказа
* Услуга
* Действие

**Системные диаграммы последовательностей, соответствующие MVC-паттернам для сценария «Работа с запросами»:**

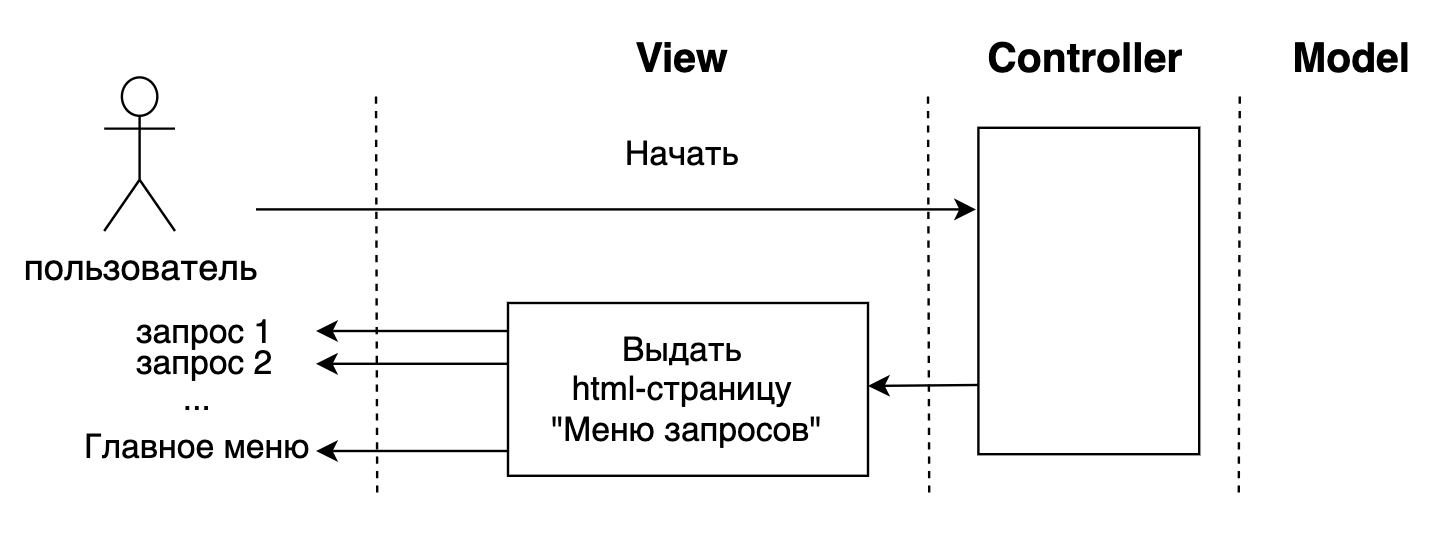


Рисунок 6 - Системная диаграмма последовательностей для меню запросов

**Системная диаграмма последовательностей, соответствующая MVC-паттерну для выполнения запроса (рис. 5.2):**

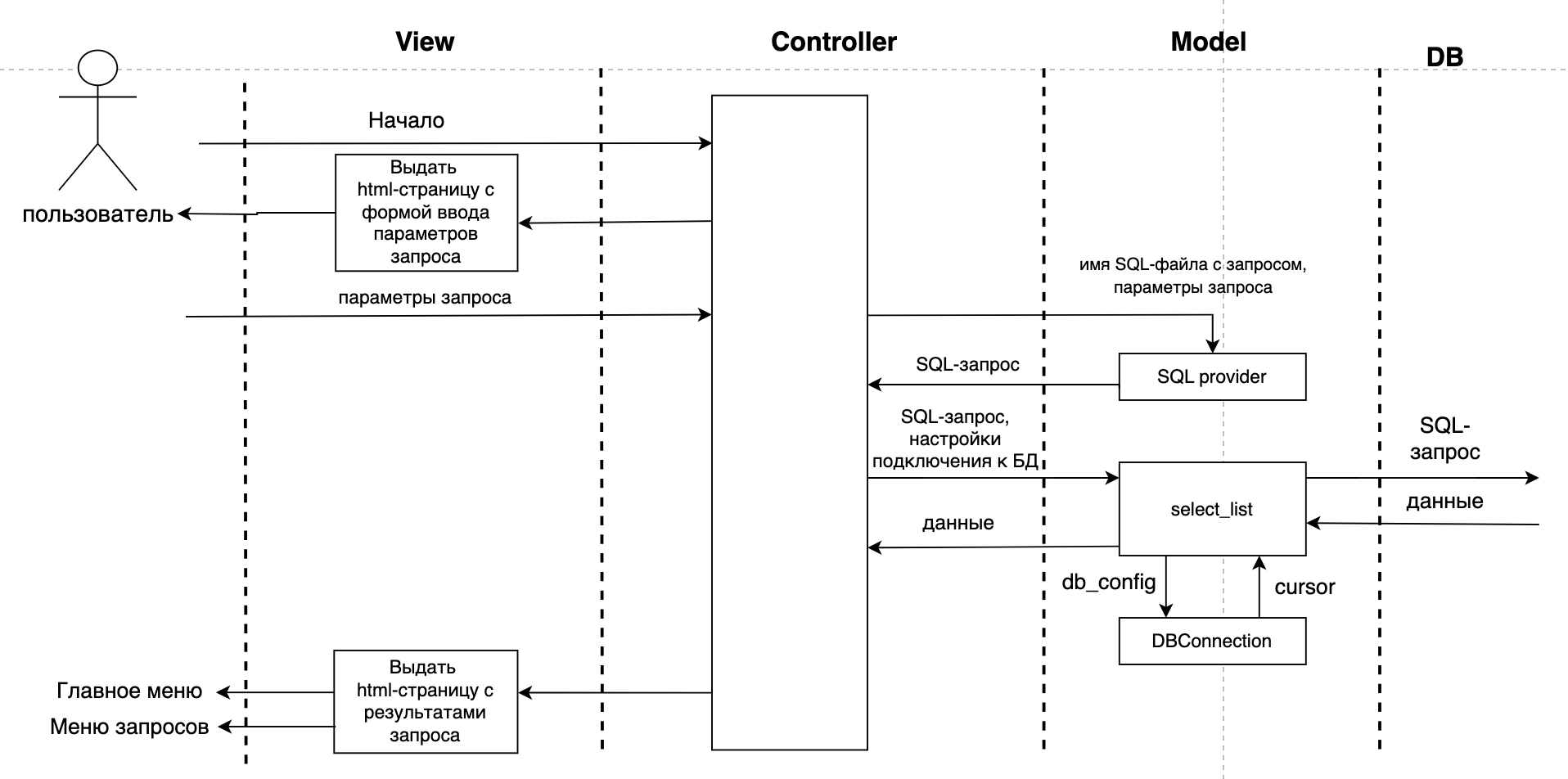


Рисунок 7 - Системная диаграмма последовательностей для выполнения запроса

**Файловая архитектура:**

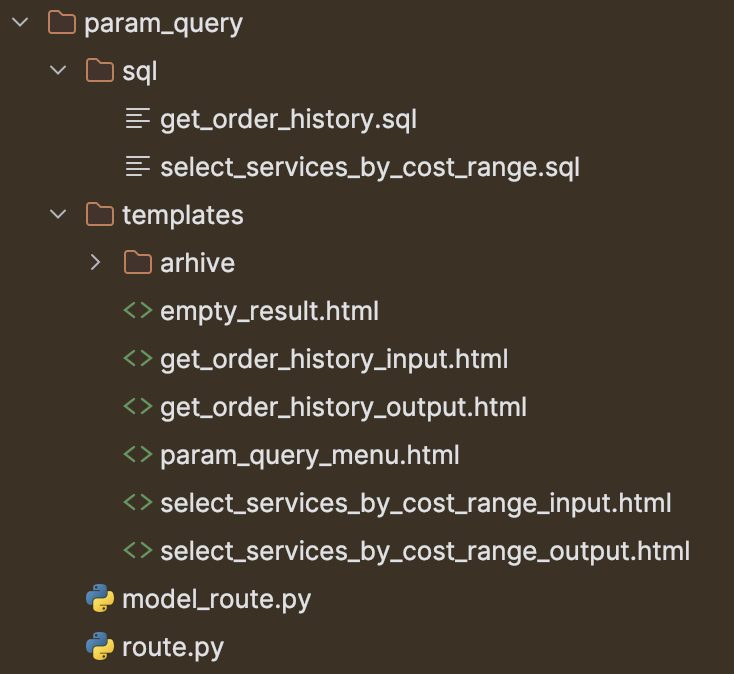


Рисунок 8 - Файловая архитектура для варианта использования "Работа с запросами"

# **Работа с отчётами**

**Предусловия:** пользователь успешно авторизовался и у него есть доступ к этому варианту использования.

**Гарантия:** при корректно выбранных типе и периоде отчёта получает сообщение об успешном создании отчета (при создании отчета), либо данные соответствующего отчета (при просмотре отчета).

**Минимальная гарантия:** пользователь получает сообщение о том, что для данного периода отчёт уже был создан, или, если пользователь ввел не все параметры, он получает сообщение о некорректно введённых данных.

**Сценарий работы для работы с отчётами:**

1. Пользователь запускает сценарий
2. Система присылает форму для ввода параметров отчёта
3. Пользователь вводит параметры отчёта
4. Система выполняет действие и выводит информационное сообщение о результате выполнения действия (если отчёт создается и просматривается несуществующий отчёт). Если действие – просмотр отчёта – система присылает пользователю страницу с отчётом.

**Исключения:**

1. Пользователь вводит данные не полностью. Система возвращает пользователя на страницу для ввода параметров отчёта и выводит соответствующее информационное сообщение.
2. Пользователь вводит данные для создания существующего отчёта или данные для просмотра ещё не существующего отчёта. Система возвращает пользователя на для ввода параметров отчёта и выводит соответствующее информационное сообщение.

**Требования к шаблонам:**

1. Статический шаблон «Меню отчётов». Шаблон показывает форму для ввода параметров запроса: типа и периода, а также кнопку для возврата в главное меню.

Шаблон содержит поля для ввода данных:

* поле типа select с id reportType и именем reportType для выбора типа отчёта
* поле типа select с id year и именем year для выбора года периода отчёта
* поле типа select с id month и именем month для выбора месяца периода отчёта
* поле типа submit для отправки данных на создание отчёта
* поле типа submit для отправки данных на просмотр отчёта

Шаблон содержит ссылки:

* На возврат в главное меню (адрес ‘/’)

1. Статический шаблон для вывода данных отчёта «Отчёт по подключениям». Шаблон показывает данные отчёта и кнопки для возврата в меню отчётов и главное меню. Вывод данных осуществляется в виде таблицы. Данные отчёта передаются в шаблон.

Шаблон содержит таблицу с колонками:

* id клиента
* Дата подключения услуги
* Дата отключения услуги
* id услуги

Шаблон содержит ссылки:

* На возврат в меню запросов (адрес ‘/report)
* На возврат в главное меню (адрес ‘/’)

1. Статический шаблон для вывода данных отчёта «Статистика по услугам». Шаблон показывает данные отчёта и кнопки для возврата в меню отчётов и главное меню. Вывод данных осуществляется в виде таблицы. Данные отчёта передаются в шаблон.

Шаблон содержит таблицу с колонками:

* id услуги
* Название
* Количество подключений
* Количество отключений

Шаблон содержит ссылки:

* На возврат в меню запросов (адрес ‘/report)
* На возврат в главное меню (адрес ‘/’)

**Системные диаграммы последовательностей, соответствующие MVC-паттернам для сценария «Работа с отчётами»:**

Изображение выглядит как текст, линия, диаграмма, Шрифт

Автоматически созданное описание

Рисунок 9 - Системная диаграмма последовательностей для меню запросов

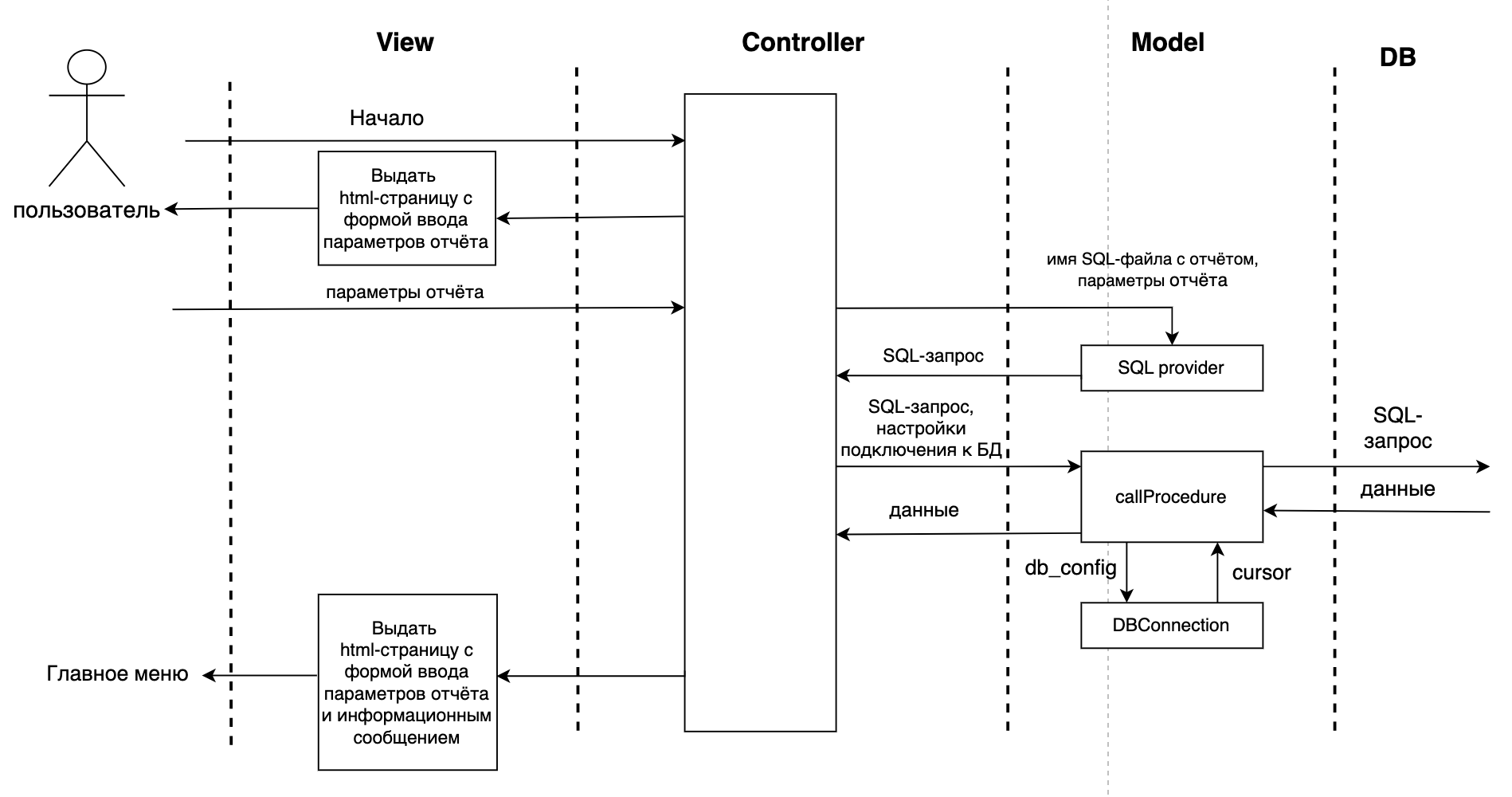


Рисунок 10 - Системная диаграмма последовательности для создания отчёта

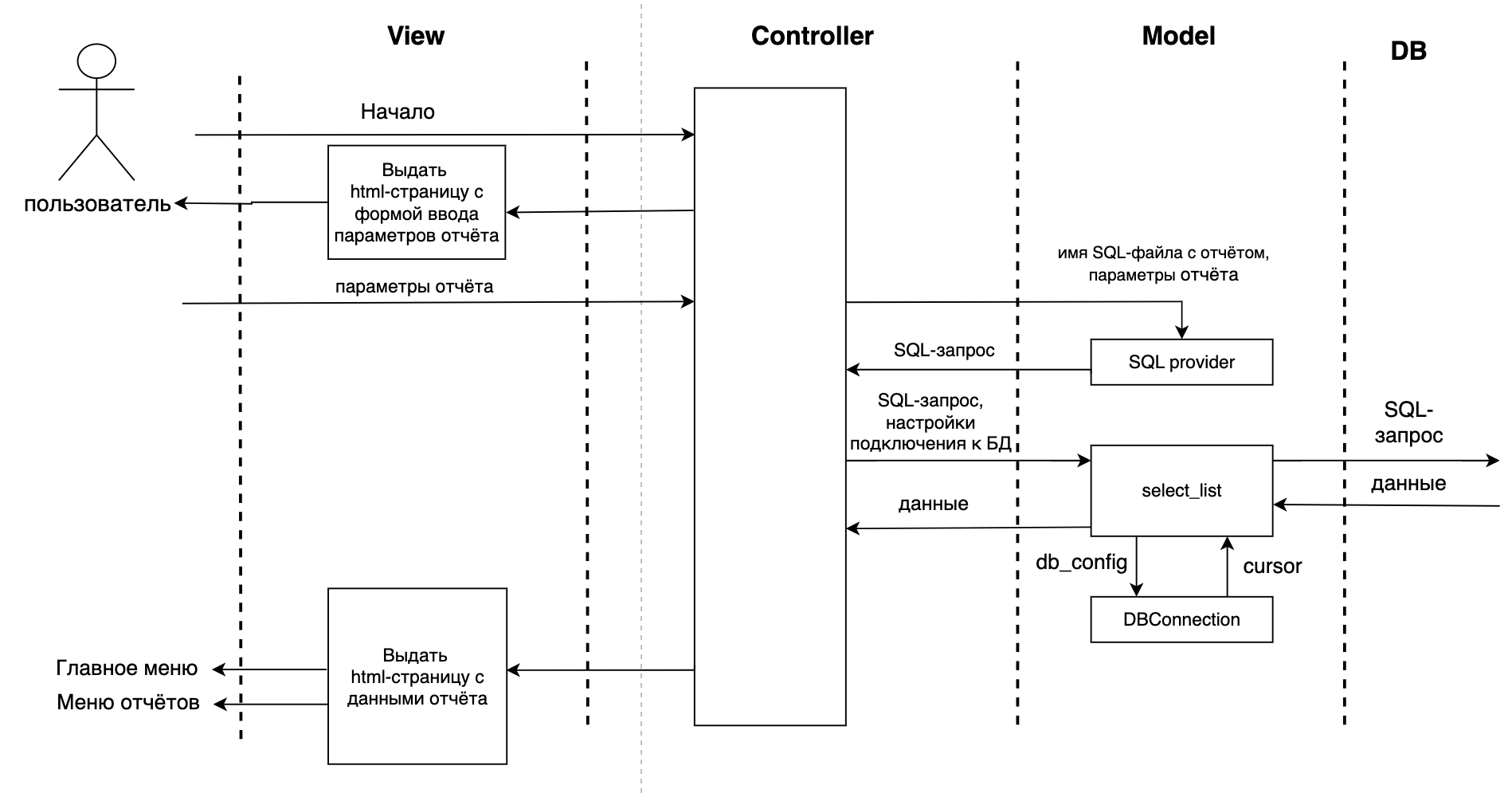


Рисунок 11 - Системная диаграмма последовательности для просмотра отчёта

**Файловая архитектура:**

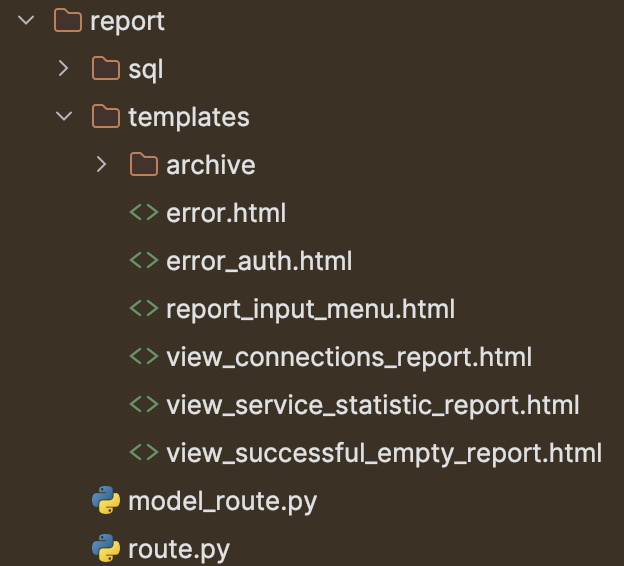


Рисунок 12 - Файловая архитектура для варианта использования "Работа с отчётами"

# **Работа с услугами**

Основной бизнес-процесс. Пользователь составляет заказ на подключение/отключение услуг, предоставляемых компанией.

**Предусловия:** пользователь успешно авторизовался и у него есть доступ к этому варианту использования.

**Гарантия:** при корректном добавлении данных заказ создается, и пользователь получает сообщение об успешном создании заказа

**Минимальная гарантия:** пользователь получает сообщение о том, что произошла ошибка, и кнопку для возврата в главное меню.

**Сценарий работы для работы с отчётами:**

1. Пользователь запускает сценарий
2. Система присылает форму с корзиной для ввода данных
3. Пользователь выбирает услуги и действия (подключение или отключение) с ними из списка предложенных, после чего они попадают в корзину
4. Пользователь оформляет заказ
5. Система записывает информацию о заказе, выполняет заказ и присылает пользователю страницу с сообщением об успешном создании и выполнении заказа и ссылку для возврата в главное меню.

**Исключения:**

1. Происходит ошибка при оформлении заказа. Система присылает страницу с сообщением об ошибке и кнопкой для возврата в главное меню.

**Требования к шаблонам:**

1. Динамический шаблон «Работа с услугами». Шаблон показывает списки подключенных и еще не подключенных услуг клиента, а также корзину и кнопки: для возврата в главное меню, для оформления заказа и работы с корзиной.

Шаблон содержит ссылки:

* На возврат в главное меню (адрес ‘/’)
* На оформление заказа (адрес ‘/save\_order’)

1. Статический шаблон «Оформление заказа». Шаблон показывает статус оформления заказа.

Шаблон содержит ссылки:

* На возврат в главное меню (адрес ‘/’)

**Системная диаграмма последовательностей, соответствующие MVC-паттерну для сценария «Работа с услугами»:**

Изображение выглядит как снимок экрана, текст, лампа, дизайн

Автоматически созданное описание

Рисунок 13 - Системная диаграмма последовательности для создания и оформления заказа

**Файловая архитектура:**

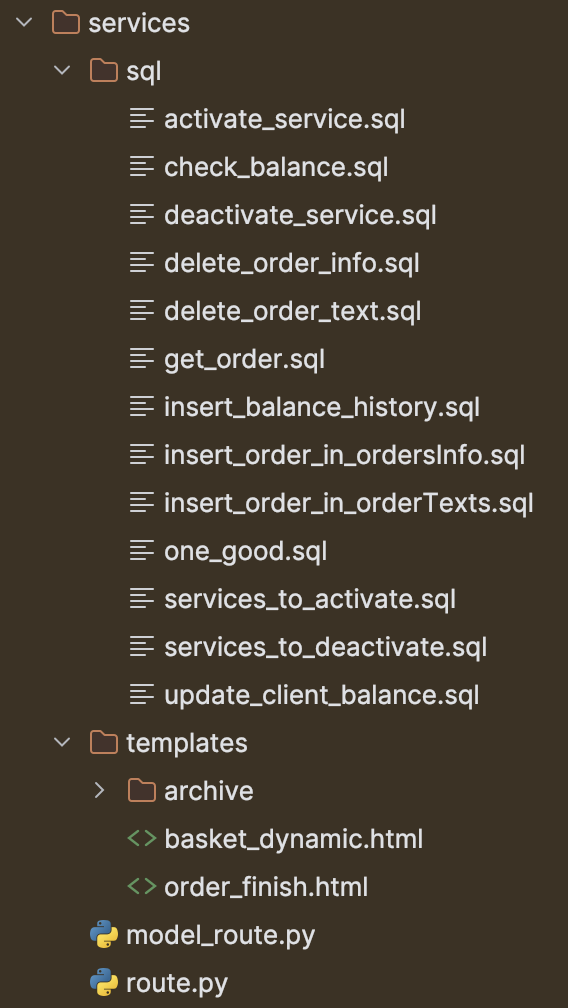
****

Рисунок 14 - Файловая архитектура для варианта использования "Работа с запросами"

# **Личный кабинет**

**Предусловия:** пользователь успешно авторизовался и у него есть доступ к этому варианту использования.

**Гарантия:** при корректном вводе данных пользователь пополняет свой баланс

**Минимальная гарантия:** пользователь получает сообщение о том, что произошла ошибка, и кнопку для возврата в главное меню.

**Сценарий работы для работы с отчётами:**

1. Пользователь запускает сценарий
2. Система присылает страницу со списком подключенных услуг и ссылкой на пополнение баланса
3. Пользователь переходит на страницу пополнения баланса
4. Пользователь вводит сумму для пополнения баланса
5. Система обновляет баланс пользователя и присылает страницу с сообщением об успешном пополнении баланса и ссылкой в главное меню

**Исключения:**

1. Происходит ошибка при пополнении баланса. Система показывает страницу с информационным сообщением об ошибке и ссылкой для перехода в главное меню.

**Требования к шаблонам:**

1. Статический шаблон «Личный кабинет». Шаблон показывает список подключенных услуг клиента, его баланс, кнопки: для возврата в главное меню, для пополнения баланса.

Шаблон содержит ссылки:

* На возврат в главное меню (адрес ‘/’)
* На оформление заказа (адрес ‘/top\_up\_balance)

1. Статический шаблон «Пополнение баланса». Шаблон показывает форму для ввода суммы для пополнения, а также кнопки для возврата в главное меню и личный кабинет.

Шаблон содержит поля для ввода данных:

* поле типа text и именем summ и id summ для ввода суммы для пополнения
* поле типа submit для отправки данных

Шаблон содержит ссылки:

* На возврат в главное меню (адрес ‘/’)
* На возврат в личный кабинет (адрес ‘/lk’)

**Системные диаграммы последовательностей, соответствующие MVC-паттернам для сценария «Личный кабинет»:**

Изображение выглядит как снимок экрана, текст, дизайн

Автоматически созданное описание

Рисунок 15 - Системная диаграмма последовательности для личного кабинета

**Файловая архитектура:**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 16 - Файловая архитектура для варианта использования "Работа с запросами"

# **Инфологическая модель**

Изображение выглядит как текст, снимок экрана, Шрифт, Параллельный

Автоматически созданное описание

Рисунок 17 - Инфологическая модель БД

# **Логическая модель**

**Dogovor**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| num\_in | date\_in | date\_out | client\_id | passport\_seria | passport\_number | date\_of\_birth | FIO |
| PK |  |  | FK |  |  |  |  |

**Client**

|  |  |  |
| --- | --- | --- |
| idClient | FIO | balance |
| PK |  |  |

**Balance\_history**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | data\_last\_balance\_edit | client\_id | balance\_before | balance\_after | type |
| PK |  | FK |  |  |  |

**External\_users**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| idExternalUsers | idClient | login | password | role | type |
| PK | FK |  |  |  |  |

**Services\_list**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | s\_id | date\_in | date\_out | client\_id |
| PK | FK |  |  | FK |

**Service**

|  |  |  |
| --- | --- | --- |
| idService | name | cost |
| PK |  |  |

**OrdersInfo**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| idOrdersInfo | s\_id | date\_in | date\_out | client\_id |
| PK | FK |  |  | FK |

**OrderTexts**

|  |  |  |  |
| --- | --- | --- | --- |
| idOrderTexts | idOrder | idService | type |
| PK | FK | FK |  |

**Reports**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| idReport | data | yearReport | monthReport | reportType |
| PK |  |  |  |  |

**InternalUsers**

|  |  |  |  |
| --- | --- | --- | --- |
| idUser | login | password | role |
| PK |  |  |  |