|  |  |
| --- | --- |
| 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)*

**РАСЧЁТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОМУ ПРОЕКТУ**

по дисциплине: «Разработка информационных систем»

|  |  |  |
| --- | --- | --- |
| Студент |  | Ядрихинская Варвара Евгеньевна |
| Группа |  | РК6-53Б |
| Тема курсовой работы |  | Разработка информационной системы «Поликлиника» |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_Ядрихинская В.Е.**

*подпись, дата фамилия, и.о.*

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

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2023 г.*

# Аннотация

Курсовая работа посвящена разработке информационной системы для поликлиники.

**Оглавление**

[Аннотация 2](#_Toc150294240)

[Техническое задание 4](#_Toc150294241)

[Определение конечных пользователей 6](#_Toc150294242)

[UML-диаграмма вариантов использования 6](#_Toc150294243)

[Вариант использования «Работа с запросами» 6](#_Toc150294244)

[Вариант использования «Авторизация» 8](#_Toc150294245)

# Техническое задание

**Описание предметной области**

Поликлиника состоит из *отделений*. Каждое отделение имеет уникальный номер, название, этаж, на котором оно расположено, известна фамилия заведующего. В каждом отделении работают многие врачи. При этом каждый *врач* по основной специальности работает только в одном отделении.

Каждому врачу присвоен уникальный идентификационный номер. О каждом враче поликлиники известны его паспортные данные, адрес, год рождения, специализация, дата приема на работу, дата увольнения. Для работающих врачей дата увольнения остается не заполненной.

В каждом отделении находится несколько *кабинетов*. Каждый кабинет имеет номер, тип (обычный, процедурный, физиотерапевтический и т.д.), известна его площадь.

Поликлинику посещают *пациенты*. На каждого пациента заводится карта с уникальным номером, известны его паспортные данные, адрес, дата рождения, дата создания карты.

Прием пациентов врачами поликлиники ведется по *расписанию*. В расписании указываются номер врача, номер кабинета, дата приема, время и номер пациента (номер карты), записанного на это время. Есть также отметка о явке/не явке пациента на прием в указанное время.

Каждое *посещение* пациентом поликлиники заносится в его карту.

Фиксируется дата посещения, врач, принимавший пациента, диагноз, поставленный в результате посещения, основные жалобы и назначения, сделанные врачом.

Для ускорения работы врача экземпляры посещений создаются специальной процедурой автоматически, как только пациент записывается на прием в расписании. При явке пациента врачу остается только внести основные жалобы, диагноз и назначения.

При неявке пациента соответствующие экземпляры удаляются другой специальной процедурой.

**Техническое задание на этап проектирования**

1. Определить конечных пользователей будущей системы.
2. Составить UML-диаграмму вариантов использования.
3. Придумать и согласовать с преподавателем основной вариант использования (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**

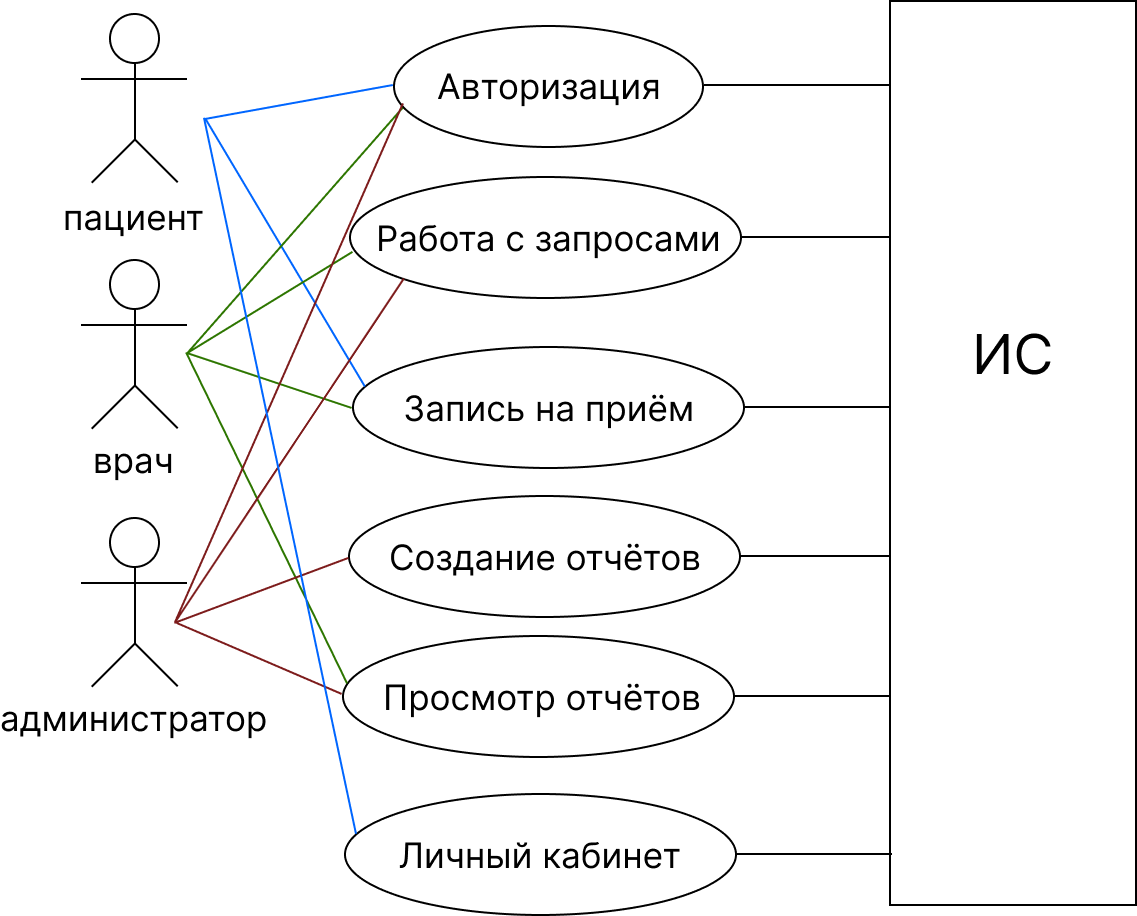
1. Реализовать разработанную на этапе проектирования информационную систему на языке Python в среде фреймворка Flask.
2. Каждый вариант использования оформить, как блюпринт.
3. Доступ конечных и внешних пользователей к вариантам использования реализовать с помощью декораторов.

# Определение конечных пользователей

# 

* Врачи
* Пациенты
* Управляющий

# UML-диаграмма вариантов использования



# Вариант использования «Работа с запросами»

**Карточка варианта использования**

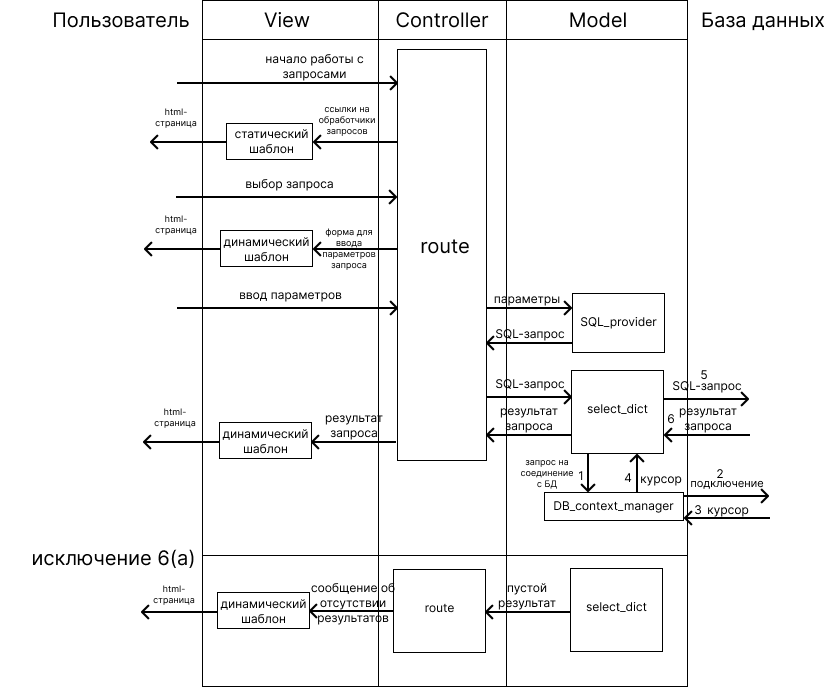
1. Название: «Работа с запросами»
2. Предусловия: пользователь авторизован и имеет доступ к работе с запросами
3. Гарантия: пользователь должен увидеть результат запроса при заданных параметрах
4. Минимальная гарантия: если пользователь не ввёл параметры, то система должна предложить их ввести

**Сценарий**

1. Пользователь начинает работу с запросами;
2. Система отправляет меню запросов;
3. Пользователь выбирает запрос;
4. Система присылает форму для ввода параметров;
5. Пользователь вводит параметры;
6. Система выполняет запрос и присылает пользователю страницу с результатами запроса и ссылкой для возврата в меню запросов.

Исключения:  
6(а). Система выполняет запрос и присылает пользователю страницу с сообщением о том, что в базе данных не найдена информация, соответствующая введенным параметрам

**MVC паттерн**



**Требования к шаблонам**

1. Статический шаблон меню запросов search\_result.html

Включает 4 ссылки (2 для пациента и 2 для врача) на запросы и ссылку на возврат на главную страницу

1. Динамические шаблоны ввода параметров для запроса input\_docs.html, input\_doc\_tmtbl.html, input\_tmtbl.html, input\_patient\_hist.html. В каждом шаблоне содержится:
   * Форма для ввода параметров
   * Кнопка для отправки данных
   * Кнопка для возвращения на главную страницу
   * Кнопка для возвращения на страницу выбора запроса
2. Динамические шаблоны вывода результатов поиска в виде таблицы result\_docs.html, result\_doc\_tmtbl.html, result\_tmtbl.html, result\_patient\_hist.html. Результаты запросов передаются в шаблоны. В каждом шаблоне содержится:
   * Таблица с результатами
   * Кнопка для возвращения на главную страницу
   * Кнопка для возвращения на обработчик запроса, который выполнялся

# Вариант использования «Авторизация»

**Карточка варианта использования**

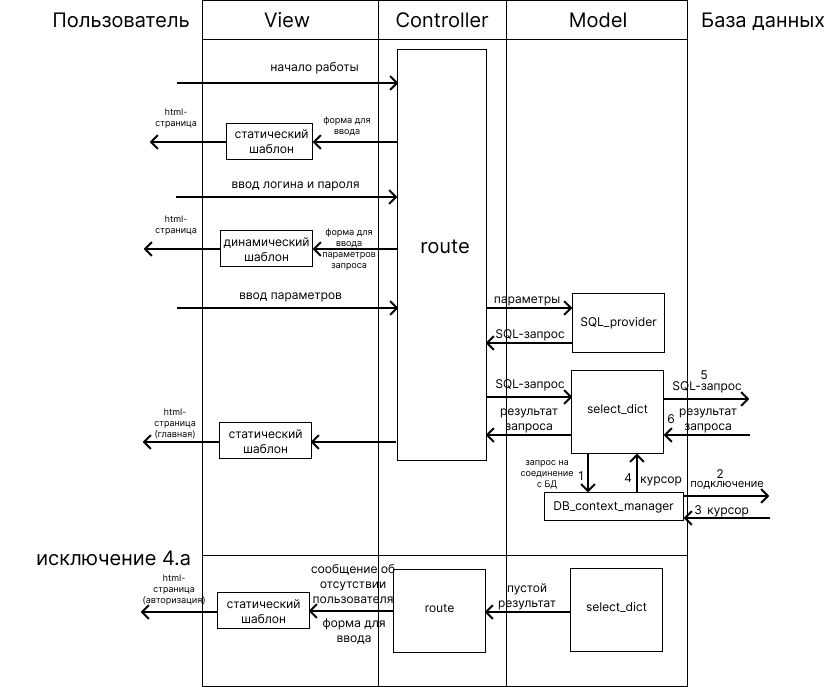
1. Название: «Авторизация»
2. Предусловия: в базе данных существуют таблицы с данными внутренних и внешних пользователей
3. Гарантия: пользователь может войти в свой аккаунт или выйти из него, при этом база данных остаётся в согласованном состоянии
4. Минимальная гарантия: если пользователь не ввёл параметры, то система предусмтаривает продолжение работы по алгоритму

**Сценарий**

1. Пользователь начинает авторизацию;
2. Система отправляет форму для ввода логина и пароля;
3. Пользователь вводит логин и пароль;
4. Система авторизует пользователя и перенаправляет на главную страницу.

Исключения:  
4(а). Пользователь вводит данные, которых нет в базе данных, и система отправляет сообщение с ошибкой и возвращает к пункту 2.

**MVC паттерн**



**Требования к шаблонам**

1. Динамический шаблон ввода логина и пароля auth.html. В нём содержится:
   * поля для ввода параметров
   * кнопка для отправки данных
   * кнопка для возвращения на главную страницу

В шаблон может передаваться:

* сообщение о том, что пользователь не найден;
* сообщение о том, что необходима авторизация

# Вариант использования «Работа с отчётами»

**Карточка варианта использования**

1. Название: «Работа с отчётами»
2. Предусловия: пользователь авторизован и имеет доступ к соответствующим вариантам работы с отчётами
3. Гарантия: пользователь должен создать или посмотреть отчёт за выбранный период
4. Минимальная гарантия: если пользователь не ввёл дату, то система должна предложить их ввести

**Сценарий**

* 1. Пользователь запускает сценарий работы с отчётами;
  2. Система присылает меню отчётов;
  3. Пользователь выбирает создание или просмотр отчёта;

С)

* + 1. Пользователь выбирает создание отчёта;
    2. Система присылает форму для ввода параметров отчёта;
    3. Пользователь вводит параметры отчёта;
    4. Система создаёт отчёт с введёнными параметрами в базе данных;
    5. Система перенаправляет пользователя в меню отчётов;

П)

* + 1. Пользователь выбирает просмотр отчёта;
    2. Система присылает форму для ввода параметров отчёта;
    3. Пользователь вводит параметры отчёта;
    4. Система присылает таблицу с данными отчёта;

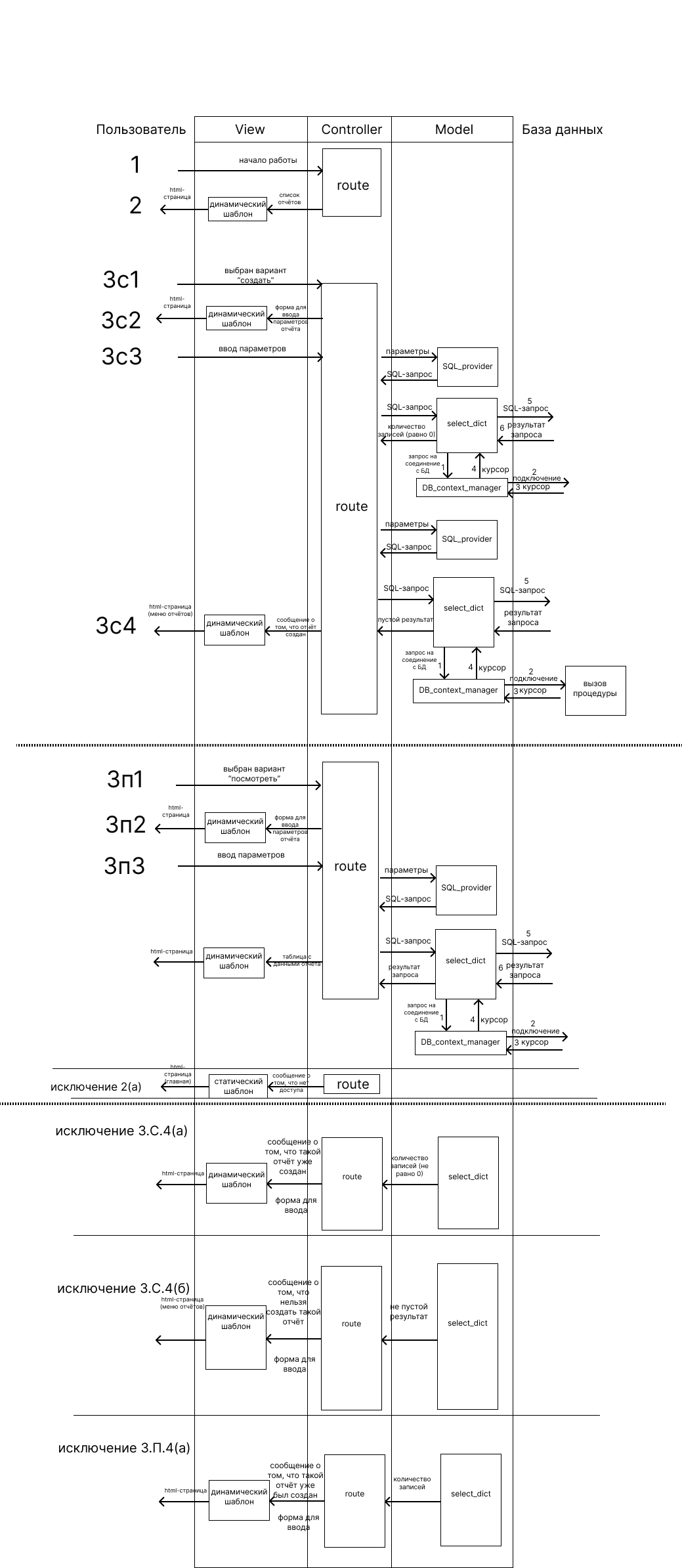
Исключения:  
2(а). Система перенаправляет пользователя на главную страницу с сообщением о том, что у пользователя нет доступа к этому сценарию.

3.С.4(а). Система присылает сообщение о том, что нельзя создать отчёт с такими параметрами. Возврат к пункту 3.С.2.

3.С.4(б). Система присылает сообщение о том, что такой отчёт уже создан. Возврат к пункту 3.С.2.

3.П.4(а). Система присылает сообщение о том, что такой отчёт уже был создан. Возврат к пункту 3.П.2.

**MVC паттерн**



**Требования к шаблонам**

1. Динамический шаблон меню отчётов. Содержит:

* Список всех видов отчётов. Каждый элемент списка содержит кнопки:
  + Создать
  + Посмотреть
* Ссылку на выход в главное меню (адрес ‘/’)

1. Динамический шаблон для ввода параметров для создания отчёта. Каждый содержит:

* Поля для ввода параметров
* Кнопку «Создать отчёт»
* Ссылку на выход в главное меню
* Ссылку на возврат к меню отчётов

1. Динамический шаблон для выбора имеющихся отчётов. Каждый содержит:
   * + Поля для ввода параметров
     + Кнопку «Посмотреть отчёт»
     + Ссылку на выход в главное меню
     + Ссылку на возврат к меню отчётов
2. Группа динамических шаблонов для вывода данных отчёта. Содержит:
   * + Таблицу с данными отчёта
     + Ссылку на выход в главное меню
     + Ссылку на выбор имеющихся отчётов