



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 (САПР)

# Расчетно-пояснительная записка к курсовой работе по дисциплине "Разработка информационных систем"

Студент:	Дунайцев Александр Иванович
Группа:	РК6-54Б
Тема:	Разработка информационной системы "История болезни"

Студент

\_\_\_\_\_  
подпись, дата

Дунайцев А. И

Преподаватель

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
Фамилия, И. О.

Москва, 2021

# Содержание

<b>1</b>	<b>Аннотация.</b>	<b>3</b>
<b>2</b>	<b>Задание. Описание предметной области.</b>	<b>3</b>
<b>3</b>	<b>Определение конечных пользователей.</b>	<b>3</b>
<b>4</b>	<b>Разработка UML диаграммы вариантов использования.</b>	<b>4</b>
<b>5</b>	<b>Вариант использования "Главное меню".</b>	<b>4</b>
5.1	Сценарий работы главного меню. . . . .	4
5.2	BRMN диаграмма контроллера главного меню. . . . .	5
5.3	Требования к шаблонам. . . . .	5
5.4	Программная архитектура реализации главного меню. . . . .	6
<b>6</b>	<b>Вариант использования «Работа с запросами».</b>	<b>6</b>
6.1	Сценарий использования "Работа с запросами". . . . .	6
6.2	BRMN диаграмма контроллера меню запросов. . . . .	7
6.3	Требования к шаблонам. . . . .	7
6.4	Программная архитектура реализации работы с запросами. . . . .	8
<b>7</b>	<b>Вариант использования «Авторизация».</b>	<b>9</b>
7.1	Сценарий работы «Авторизация». . . . .	9
7.2	BRMN диаграмма контроллера авторизации. . . . .	9
7.3	Требования к шаблонам. . . . .	9
7.4	Программная архитектура реализации авторизации. . . . .	10
<b>8</b>	<b>Вариант использования "Запись пациентов на госпитализацию".</b>	<b>10</b>
8.1	Сценарий использования "Запись пациентов на госпитализацию". . . . .	10
8.2	BRMN диаграмма контроллера записи пациентов. . . . .	11
8.3	Требования к шаблонам. . . . .	11
8.4	Программная архитектура реализации сценария "Запись пациента на госпитализацию". . . . .	12
<b>9</b>	<b>Вариант использования "Лечение пациента".</b>	<b>12</b>
9.1	Сценарий использования "Лечение пациента". . . . .	12
9.2	BRMN диаграмма контроллера "Лечение пациента". . . . .	13
9.3	Требования к шаблонам. . . . .	13
9.4	Программная архитектура реализации сценария "Лечение пациента". . . . .	14
<b>10</b>	<b>Логическая модель базы данных</b>	<b>15</b>

## 1 Аннотация.

Курсовой проект включает в себя реализацию параметризованных запросов через пользовательский интерфейс, авторизацию пользователей, проект и реализацию основного бизнес-процесса, а также работу с хранимыми процедурами.

Была составлена UML диаграмма ролей конечных пользователей. Для каждой роли предусмотрены соответствующие варианты использования информационной системы.

Для разработки информационной системы, в рамках данного курсового проекта, были использованы следующие программные средства, языки программирования и технологии: язык программирования общего назначения python, фреймворк для создания веб-приложений на языке python Flask, язык гиперразметки HTML, декларативный язык программирования запросов к реляционным базам данных SQL, система управления базами данных MySQL, набор инструментов для создания сайта Bootstrap.

## 2 Задание. Описание предметной области.

В рамках курсового проекта необходимо разработать информационную систему, состоящую из базы данных и интерфейса конечного пользователя. В интерфейсе конечного пользователя должно быть предусмотрено 4 варианта использования: Главное меню, работа с запросами, авторизация и основной бизнес-процесс. Для каждого варианта использования необходимо сделать:

- Сценарий
- BPMN-диаграмму контроллера
- Требования к шаблонам
- Программная архитектура реализации вариантов использования

## 3 Определение конечных пользователей.

В разрабатываемой информационной системе определим 3 вида конечных пользователей:

- Администратор
- Главный врач
- Врач

## 4 Разработка UML диаграммы вариантов использования.

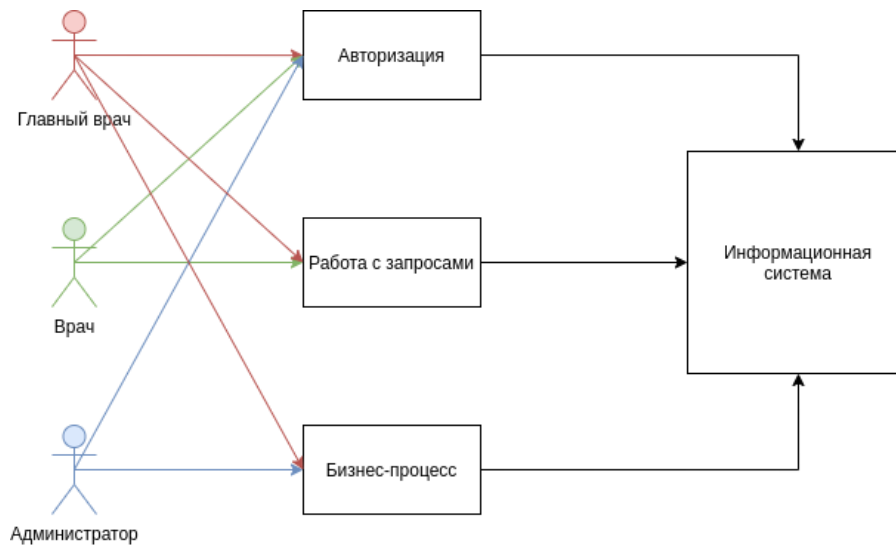


Рис. 1: UML диаграмма вариантов использования.

## 5 Вариант использования "Главное меню".

Количество пунктов в главном меню соответствует количеству вариантов использования плюс пункт для выхода из системы.

При запуске ИС управление передается контроллеру главного меню.

### 5.1 Сценарий работы главного меню.

- Пользователь запускает сценарий.
- Система присылает главное меню.
- Пользователь выбирает один из пунктов (вариантов использования).
- Система передаёт управление контроллеру соответствующего варианта использования.

## 5.2 BPMN диаграмма контроллера главного меню.

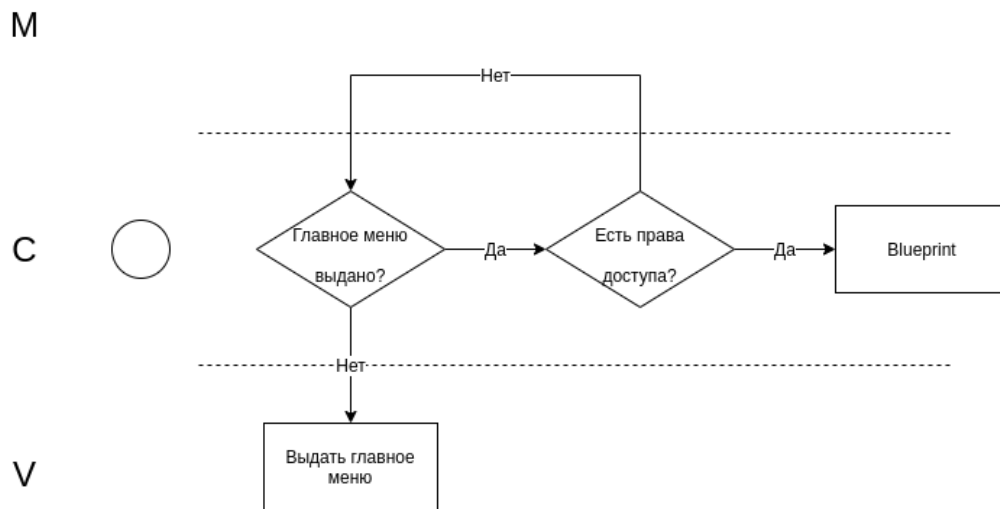


Рис. 2: BPMN диаграмма контроллера главного меню.

## 5.3 Требования к шаблонам.

Статический шаблон Главное меню.

Меню содержит следующие ссылки:

- На контроллер работы с запросами (адрес `'/db_requests'`)
- На выход из системы (адрес `'/exit'`)

## 5.4 Программная архитектура реализации главного меню.

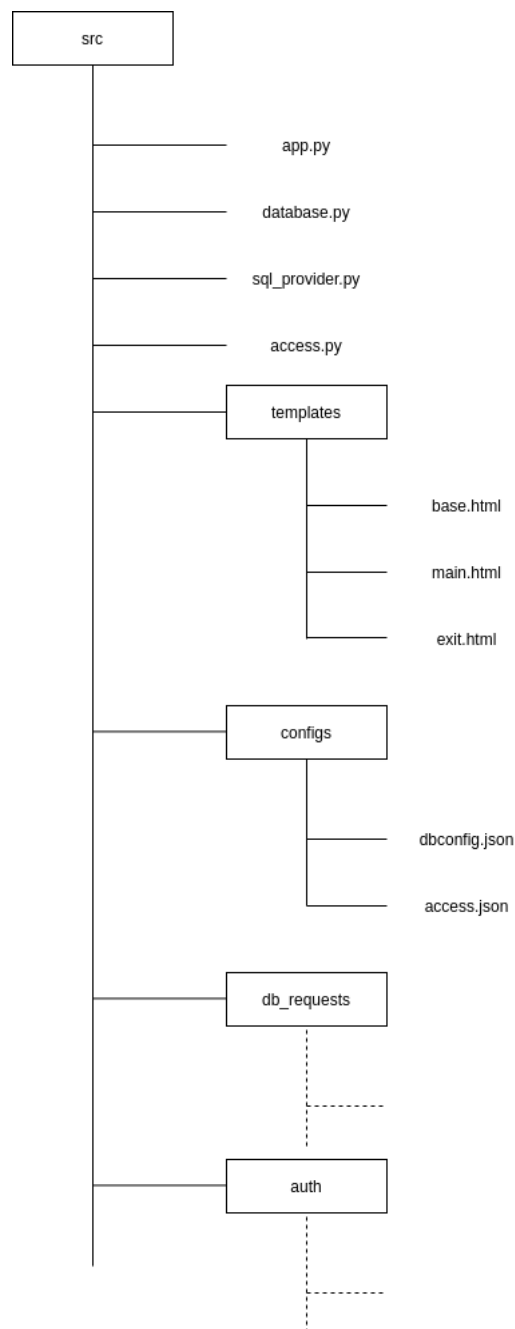


Рис. 3: Программная архитектура реализации главного меню.

## 6 Вариант использования «Работа с запросами».

### 6.1 Сценарий использования "Работа с запросами".

- Пользователь запускает сценарий.
- Система присылает меню запросов.
- Пользователь выбирает запрос.

- Система присылает форму для ввода параметров.
- Пользователь вводит параметры.
- Система выполняет запрос и присылает пользователю страницу с результатами запроса и ссылкой для возврата в меню запросов.

## 6.2 BPMN диаграмма контроллера меню запросов.

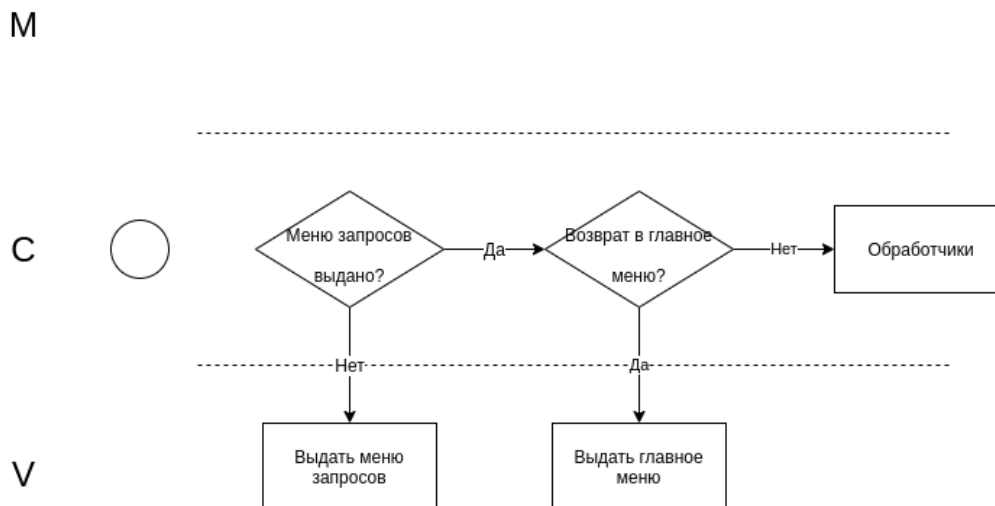


Рис. 4: BPMN диаграмма контроллера меню запросов.

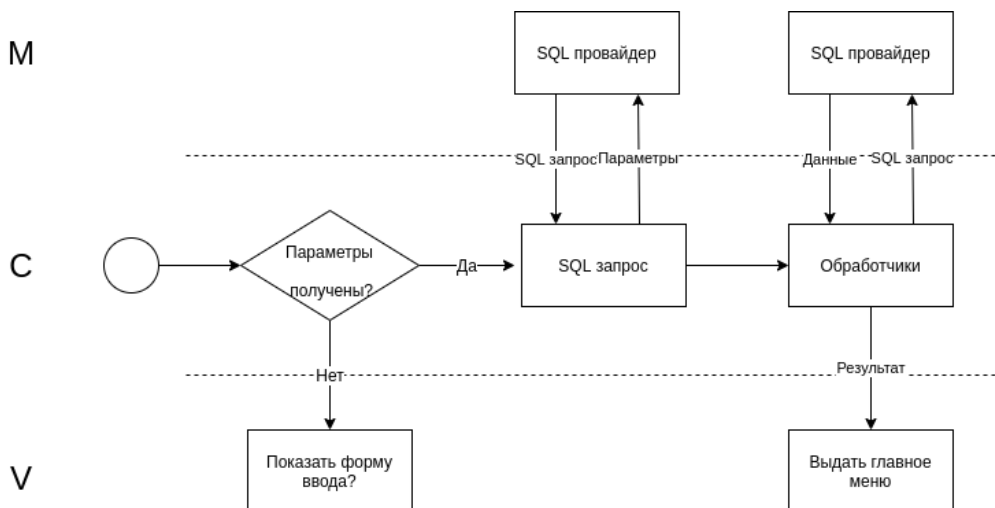


Рис. 5: BPMN диаграмма обработчика конкретного запроса.

## 6.3 Требования к шаблонам.

Меню запросов соержит следующие ссылки:

- На страницу для выполнения запроса 1 (адрес '/requests/1')
- На страницу для выполнения запроса 2 (адрес '/requests/2')

- На страницу для выполнения запроса 3 (адрес '/requests/3')
- На страницу для выполнения запроса 4 (адрес '/requests/4')
- На возврат в главное меню (адрес '/')

Страница каждого из запросов содержит форму для ввода параметров запроса и ссылку (адрес '/requests') на возврат в меню запросов. Форма в свою очередь должна иметь кнопку для отправки введённых параметров и перехода по ссылке '/requests/result'.

Страница, отображающая результаты запроса, должна представлять их в виде таблицы. Под таблицей должна находиться ссылка (адрес '/requests') для возврата в меню запросов.

#### 6.4 Программная архитектура реализации работы с запросами.

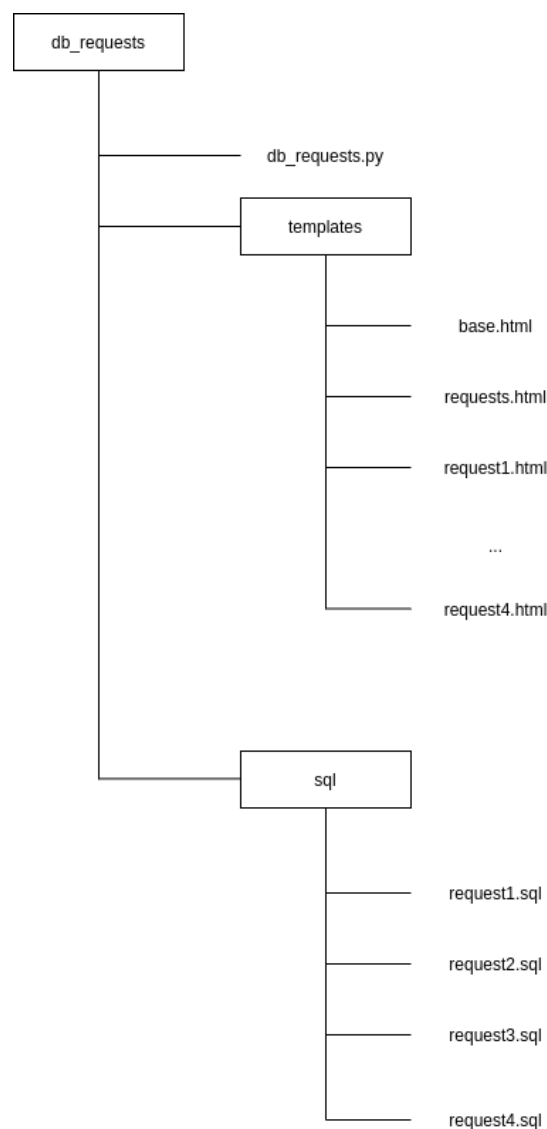


Рис. 6: Программная архитектура реализации работы с запросами.



## 7 Вариант использования «Авторизация».

### 7.1 Сценарий работы «Авторизация».

- Пользователь запускает сценарий.
- Система возвращает форму ввода логина и пароля.
- Пользователь вводит данные и нажимает кнопку отправить.
- Система проверяет введённые данные и возвращает сообщение об успешной авторизации, если такие логин и пароль существуют в БД, и сообщение об ошибке, если пользователя с таким логином и паролем в БД нет.

### 7.2 BPMN диаграмма контроллера авторизации.

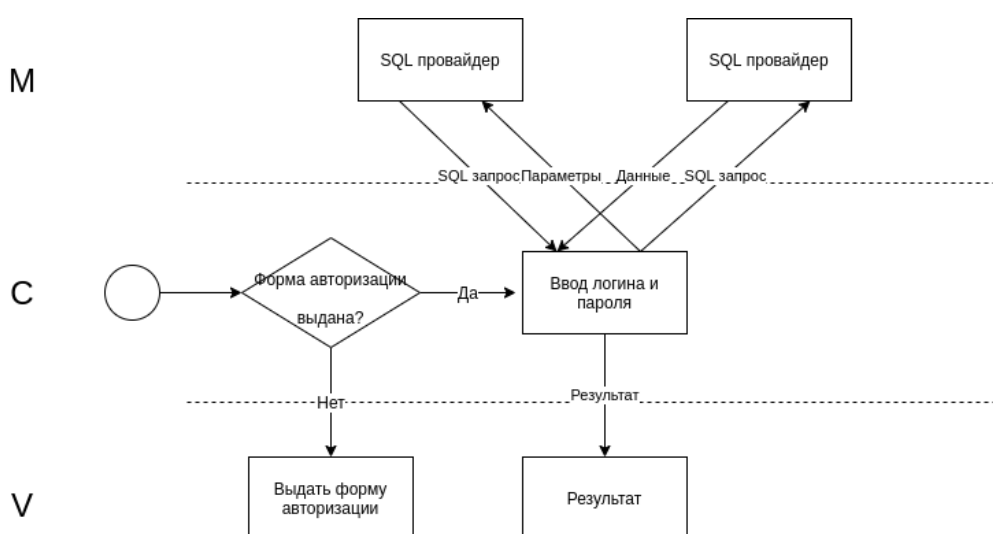


Рис. 7: BPMN диаграмма контроллера авторизации.

### 7.3 Требования к шаблонам.

Страница авторизации содержит форму для ввода логина и пароля, кнопку для отправки обработчику введённых данных, кнопку для очистки сессии (переход по адресу ‘/login/clearsession’) и ссылку на возврат в главное меню (адрес ‘/’).

При вводе верного логина и пароля после нажатия кнопки отправки должна отображаться страница с подтверждением, содержащая ссылку для перехода в главное меню.

При вводе неверного логина или пароля после нажатия кнопки отправки должна отображаться страница с сообщением об ошибке, содержащая ссылку для перехода в меню запросов (адрес ‘/login’), для повторного ввода данных, и ссылка для перехода в главное меню.

При нажатии кнопки для очистки сессии должна отобразиться страница с сообщением об успешной очистке сессии и ссылкой для перехода в главное меню.

## 7.4 Программная архитектура реализации авторизации.

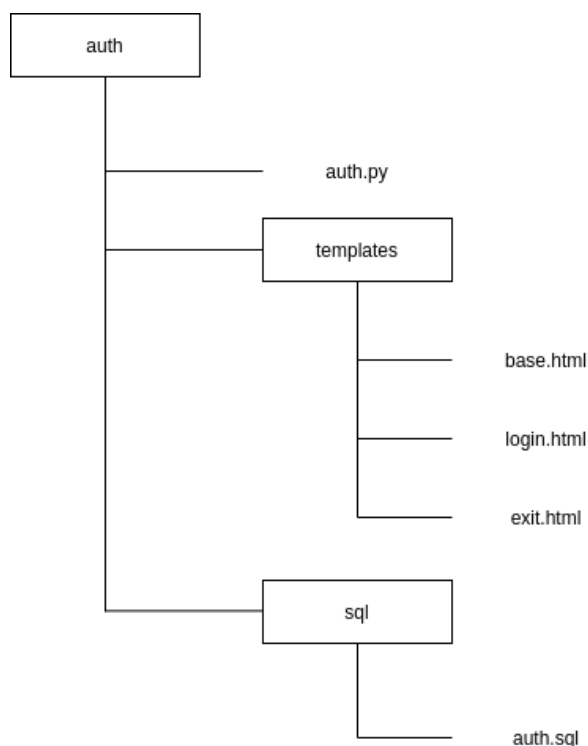


Рис. 8: BPMN диаграмма контроллера авторизации.

## 8 Вариант использования "Запись пациентов на госпитализацию".

### 8.1 Сценарий использования "Запись пациентов на госпитализацию".

- Пользователь запускает сценарий.
- Система присылает форму записи пациентов, имеющих направление на госпитализацию.
- Пользователь выбирает пациента и проверяет его паспортные данные.
- Система выдает список врачей и доступных палат, в которых можно разместить пациента.
- Пользователь выбирает для пациента врача и палату из доступных.
- Система выполняет запрос и вносит изменения в базу данных. Создается новая запись о пациенте и его истории болезни в соответствующих таблицах.

В случае, если мест в палатах нет, пациент сможет быть госпитализирован только после того, как места в палатах освободятся.

## 8.2 BPMN диаграмма контроллера записи пациентов.

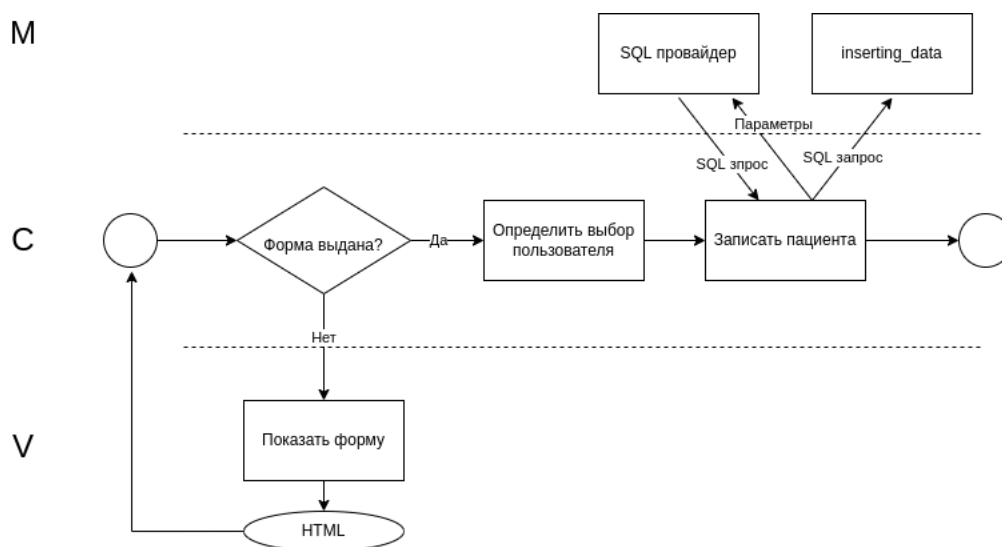


Рис. 9: BPMN диаграмма контроллера записи пациентов на госпитализацию.

## 8.3 Требования к шаблонам.

Страница с сценарием "Запись пациента на госпитализацию" содержит три поля, обязательных для выбора. Первое поле содержит список фамилий и паспортных данных пациентов, имеющих направление на госпитализацию. Второе поле позволяет сделать выбор палаты, в которой есть место, чтобы разместить пациента. Также для каждой палаты отображено, какое количество людей в ней уже есть. Третье поле позволяет выбрать врача для пациента. Также, чтобы передать выбранные параметры в систему, на странице предусмотрена соответствующая кнопка.

При отсутствии свободных мест в платах, пользователь не может записать пациента на госпитализацию до тех пор, пока место в одной из палат не освободится.

## 8.4 Программная архитектура реализации сценария "Запись пациента на госпитализацию".

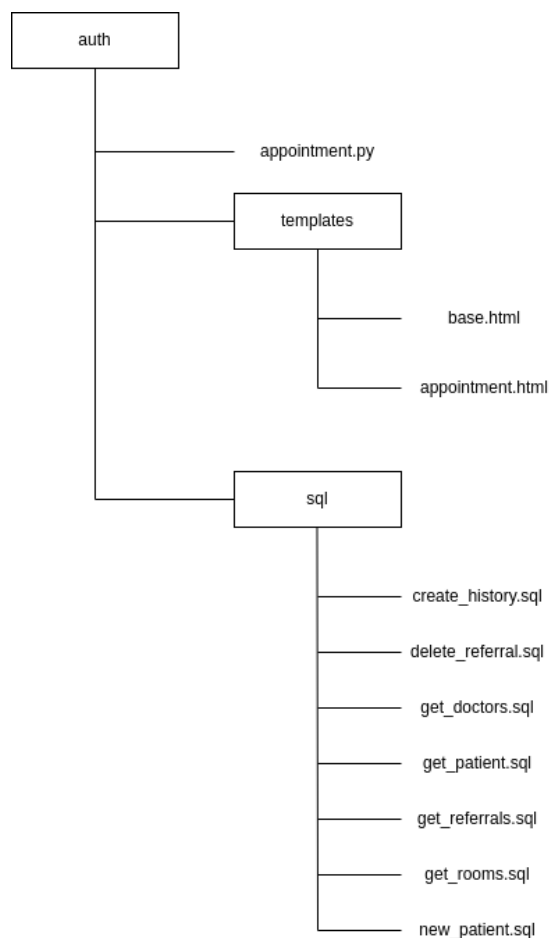


Рис. 10: Программная архитектура реализации сценария "Запись пациента на госпитализацию".

## 9 Вариант использования "Лечение пациента".

### 9.1 Сценарий использования "Лечение пациента".

- Пользователь запускает сценарий.
- Открывает форму выбора пациента.
- Пользователь выбирает пациента для проведения осмотра.
- Система получает информацию о выбранном пациенте и отображает страницу с историей болезни пациента.
- Пользователь вносит записи в историю болезни.
- Система отображает внесенные записи в соответствующем поле на странице истории болезни.

- Пользователь выписывает пациента.
- Система вносит изменения в историю болезни и закрывает ее. Затем система возвращает пользователя к странице выбора пациентов.

## 9.2 BPMN диаграмма контроллера "Лечение пациента".

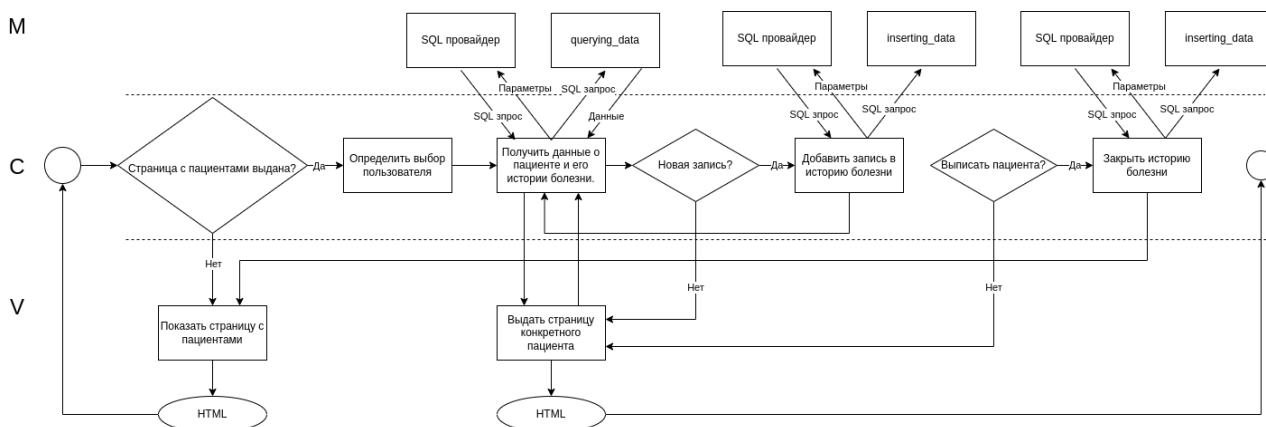


Рис. 11: BPMN диаграмма контроллера "Лечение пациента".

## 9.3 Требования к шаблонам.

Данный сценарий реализации подразумевает использование двух основных динамических таблиц, а именно: страница выбора пациента, страница лечения выбранного пациента.

Страница выбора пациента должна содержать информацию о всех пациентах, для которых на данный момент открыта история болезни. Кроме того, данная страница должна предусматривать разделение пациентов на две группы. Первая группа содержит пациентов, которые относятся к самому пользователю, а вторая, пациентов, относящихся к другим пользователям. Данные о каждом пользователе представлены в виде некоторого визуального объекта, содержащего кнопку "Провести осмотр". Нажав на эту кнопку пользователь передает параметры о выбранном пользователе в систему, которая в свою очередь открывает страницу лечения выбранного пациента.

Страница лечения выбранного пациента должна содержать всю информацию о пациенте, его враче и истории болезни. Кроме того, на данной странице должен размещаться список записей в истории болезни.

На странице лечения выбранного пациента должно быть две формы. Первая форма позволяет провести осмотр и внести новую запись в историю болезни. Эта форма должна содержать в себе текстовое поле ввода и кнопку для отправки данных в систему. Вторая форма необходима для выписки пациента. Она также содержит текстовое поле для внесения диагноза при выписке, а и кнопку для отправки данных в систему. При использовании первой формы, пользователь должен увидеть отображение новой, внесенной им, записи на странице пациента. При использовании второй формы пользователя вернут на страницу выбора пациента. Однако, только что выписанного пациента в списке уже быть не должно, так как его история болезни закрыта.

#### 9.4 Программная архитектура реализации сценария "Лечение пациента".

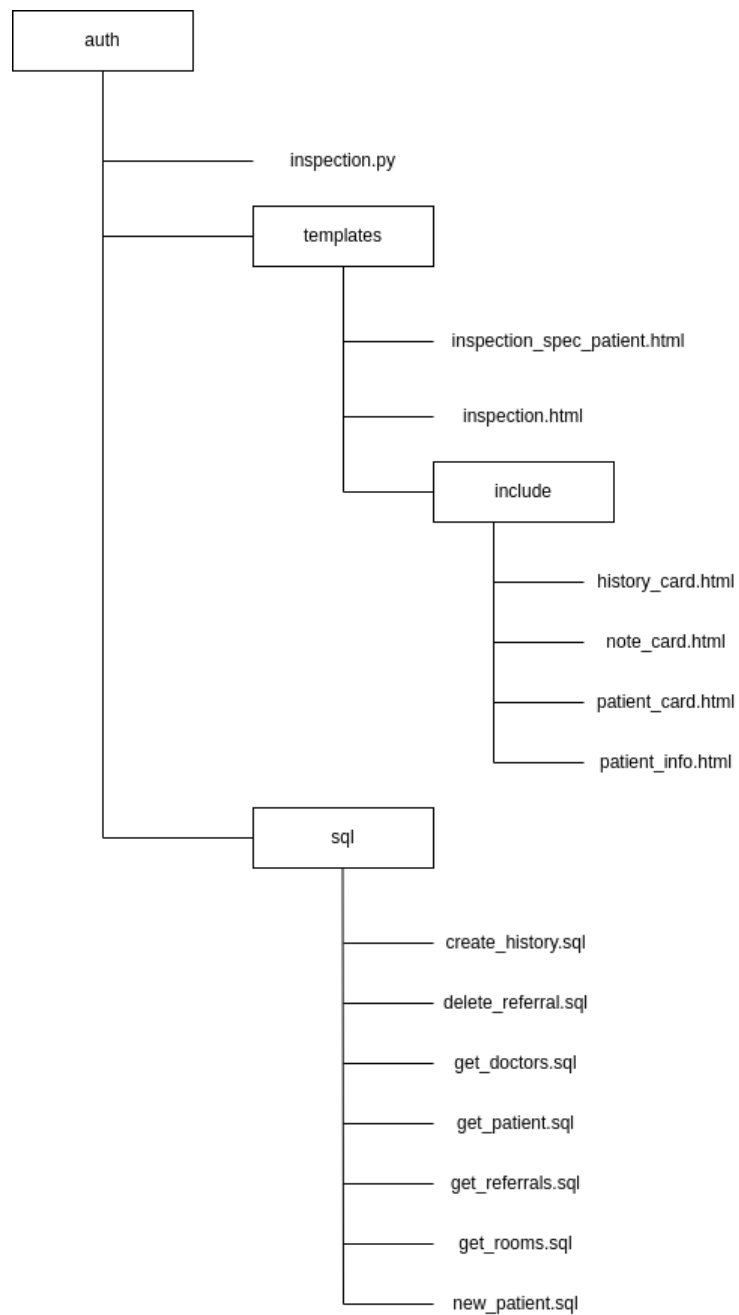


Рис. 12: Программная архитектура реализации сценария "Лечение пациента".

## 10 Логическая модель базы данных

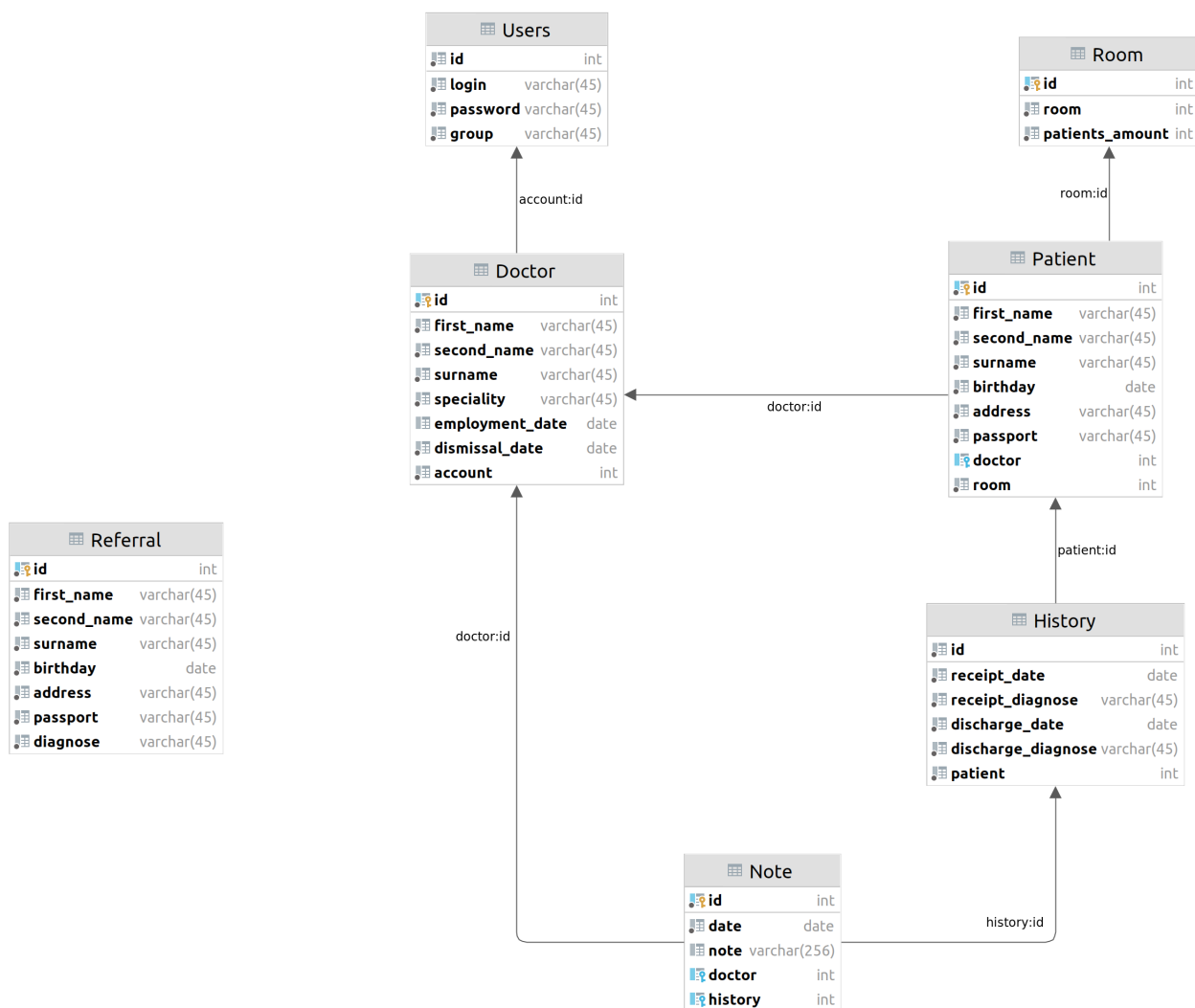


Рис. 13: Логическая модель базы данных информационной системы "История болезни".  
Диаграмма.

Таблица 1: Таблица Doctor

id	f_name	s_name	surname	speciality	empl_date	dis_date	account
pk							fk

Таблица 2: Таблица Patient

<b>id</b>	f_name	s_name	surname	birthday	address	passport	doctor	room
pk							fk	fk

Таблица 3: Таблица History

<b>id</b>	receipt_date	receipt_diagnose	discharge_date	discharge_diagnose	patient
pk					fk

Таблица 4: Таблица Note

<b>id</b>	date	note	doctor	history
pk				fk

Таблица 5: Таблица Users

<b>id</b>	login	password	group
pk			

Таблица 6: Таблица Room

<b>id</b>	room	patients_amount
pk		