

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
( **Н И У « Б е л Г У »** )

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА ИНФОРМАЦИОННЫХ И РОБОТОТЕХНИЧЕСКИХ СИСТЕМ

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА РЕГИСТРАТУРЫ  
ПОЛИКЛИНИКИ «НИУ БЕЛГУ»**

Выпускная квалификационная работа студента  
обучающегося по направления подготовки 09.03.02 Информационные системы  
и технологии  
очной формы обучения, группы 07001408  
Гадоева Улугбека Хасан угли

Научный руководитель  
доцент, к.т.н., Шамраева Е.О.

БЕЛГОРОД 2018

## РЕФЕРАТ

Автоматизированная система регистратуры поликлиники НИУ БелГУ – Гадоев Улугбек Хасан угли, выпускная квалификационная работа бакалавра, Белгород, Белгородский государственный национальный исследовательский университет (НИУ «БелГУ»), количество страниц 54, включая приложения 63, количество рисунков 41, количество таблиц 3, количество использованных источников 21.

Медицинская информационная система, регистратура поликлиники НИУ БелГУ, госуслуги.

**ОБЪЕКТ ИССЛЕДОВАНИЯ:** процесс автоматизации регистратуры поликлиники НИУ БелГУ.

**ПРЕДМЕТОМ ИССЛЕДОВАНИЯ:** является спроектированная автоматизированная информационная система (АИС) регистрации пациентов в поликлинике НИУ БелГУ.

**ЦЕЛЬ РАБОТЫ:** автоматизация работы регистратуры поликлиники НИУ БелГУ.

**ЗАДАЧИ ИССЛЕДОВАНИЯ:** провести обзор существующих АИС регистратуры поликлиники, разработать техническое задание, разработать модели для проектирования, программно реализовать разработанные модели, провести тестирование разработанной АИС регистратуры поликлиники НИУ БелГУ.

**ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ:** произведен анализ предметной области; ознакомления с АИС Регистратуры поликлиники; разработан и протестирован АИС Регистратуры поликлиники НИУ БелГУ.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Обзор и анализ существующих АИС «Регистратура поликлиники».....	6
1.1 АИС «Электронная регистратура» в Самарской областной поликлиники.....	6
1.1.1 Общие сведения .....	6
1.1.2 Структура процесса выписки талона к специалисту через «Онлайн-регистратуру» .....	7
1.2 АИС «Регистратуры поликлиники» в Белгородской области.....	11
2 Разработка автоматизированной системы регистратуры «НИУ БелГУ» .....	17
2.1 Техническое задание на разработку автоматизированной системы регистратуры поликлиники «НИУ БелГУ».....	17
2.1.1 Основание для разработки .....	18
2.1.2 Назначение разработки .....	18
2.1.3 Требования к АСРП НИУ БелГУ .....	18
2.1.4 Технико-экономические показатели .....	21
2.1.5 Стадии и этапы разработки.....	21
2.2 Разработка модели использования.....	22
2.3 Моделирование информационной системы.....	24
2.4 Инфологическое проектирование базы данных .....	29
2.5 Даталогическая модель .....	32
3 Программная реализация информационной системы .....	41
3.1 Разработка и тестирование АИС регистратуры поликлиники «НИУ БелГУ» .....	41
ЗАКЛЮЧЕНИЕ.....	50
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	51
Приложение А.....	53

## ВВЕДЕНИЕ

Современные медицинские организации производят и накапливают огромные объемы данных. От того, насколько эффективно эта информация используется врачами, руководителями, управляющими органами, зависит качество медицинской помощи, общий уровень жизни населения, уровень развития страны в целом и каждого ее территориального субъекта, в частности. Поэтому необходимость использования больших, и при этом еще постоянно растущих, объемов информации при решении диагностических, терапевтических, статистических, управленческих и других задач, обуславливает сегодня создание информационных систем в медицинских учреждениях.

До недавнего времени в российском здравоохранении почти полностью отсутствовали хоть какие-то признаки автоматизации. Карты, бюллетени, процедурные отчеты, учет пациентов, лекарственных препаратов – весь документооборот производился на бумаге. Это осложняло руководство лечебно-профилактического учреждения (отсутствие контроля работы подразделений, недостаток оперативной, аналитической информации) и работу контролирующих органов.

В настоящее время госкорпорация «Росатом» совместно с Министерством здравоохранения РФ принимает участие в масштабном проекте «Бережливая поликлиника», который призван повысить эффективность работы лечебно-профилактических учреждений и общую доступность медицинской помощи в РФ. Проект в разы сокращает очереди, ускоряет сдачу и выполнение анализов, упрощает запись к терапевту и позволяет обслуживать гораздо больше пациентов при том же количестве ресурсов. «Бережливая поликлиника» в целом подразумевает целый ряд процессов, которые можно оптимизировать. В частности, в рамках проекта перестраивается работа регистратуры медучреждения с учетом современных стандартов коммуникации, информатизации и формирования доступной среды

для маломобильных групп населения. Внедряется электронная очередь. Врачи освобождаются от несвойственной им работы, в том числе бумажной. Все это позволяет сберечь время как медицинского персонала, так и пациентов, и, как следствие, приведет к повышению удовлетворенности потребителей медицинских услуг [5].

Поэтому актуальной задачей является разработка медицинской информационной системы (МИС) регистратуры поликлиники НИУ БелГУ.

Целью выпускной квалификационной работы (ВКР) является автоматизация работы регистратуры поликлиники «НИУ БелГУ» для упрощения записи пациентов к врачам, улучшения качества обслуживания пациентов и сокращения времени ожидания приема.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) провести обзор существующих МИС регистратуры поликлиники;
- 2) разработать техническое задание;
- 3) разработать модели для проектирования;
- 4) программно реализовать разработанные модели;
- 5) провести тестирование разработанной МИС.

Объектом исследований является регистратура поликлиники «НИУ БелГУ». Предметом исследования является спроектированная автоматизированная информационная система (АИС) регистрации пациентов в поликлинике «НИУ БелГУ».

Первый раздел ВКР посвящен анализу предметной области и моделированию МИС регистратуры поликлиники НИУ БелГУ.

Во втором разделе описано проектирование системы, включающее в себя разработку технического задания, модели использования, схемы базы данных и диаграммы классов.

В третьем разделе проведена программная реализация автоматизированной информационной системы регистратуры поликлиники.

# 1 Обзор и анализ существующих АИС «Регистратура поликлиники»

## 1.1 АИС «Электронная регистратура» в Самарской областной поликлиники

### 1.1.1 Общие сведения

Комплексная информационная автоматизированная система «Электронная регистратура» предназначена для автоматизации процесса записи пациентов на прием в медицинские учреждения в целях улучшения качества медицинского обслуживания населения, в частности для уменьшения очередей в регистратурах поликлинических учреждений, расширения пропускной способности системы записи пациентов на прием за счет ввода в процесс сенсорного терминала записи и онлайн-модуля на Интернет-сайте в качестве инструментов записи. Система позволяет пациенту [1]:

- просмотреть расписание всех врачей медицинского учреждения.

Выбор врача может осуществляться как по специализации, так и пофамильно;

- просмотреть место и время приема интересующего врача;
- просмотреть количество доступных талонов на прием к врачу на заданный период времени;
- самостоятельно без участия сотрудников регистратуры, без очереди записаться на прием к врачу через терминал или из дома через интернет сайт;
- распечатать талон на прием к врачу установленного ЛПУ (лечебно-профилактическое учреждение) образца.

Электронная регистратура обеспечивает:

- полную совместимость с полным программным комплексом АИС «Поликлиники»;

- авторизацию пациентов через сайт и терминал по любым персональным данным, доступным в АИС «Поликлиника» (№ паспорта, № полиса, дата рождения) вручную или с помощью штрих кода;
- контроль прохождения флюорографии или других данных пациента, необходимых для записи на прием;
- автоматическую трансляцию расписание врачей и количества талонов на терминал и интернет сайт без участия ИТ специалистов ЛПУ;
- автоматическое фиксирование и учет талонов в АИС «Поликлиника» при удаленной самостоятельной записи пациента к врачу;
- контроль за распределением и учет талонов для удаленного бронирования.

С помощью модуля «Онлайн регистратура», реализованного на Интернет-сайте медицинского учреждения пациент может просмотреть расписание врачей ЛПУ, зарегистрироваться и выписать талон к интересующему специалисту и распечатать его, не выходя из дома. Все изменения, сделанные пациентом удаленно, фиксируются в базе АИС «Поликлиника». Процесс регистрации осуществляется пошагово, с возможностью возврата как к предыдущему окну, так и к началу (списку специалистов). Интерфейс системы интуитивно понятен и удобен для восприятия. Все шаги процесса выстроены в логической последовательности [1].

#### 1.1.2 Структура процесса выписки талона к специалисту через «Онлайн-регистратуру»

Первоначально пациент должен авторизоваться на сайте. Окно авторизации приведено на рисунке 1.1. Пациент должен ввести такие данные, как номер свидетельства о рождении или паспорта, номер медицинского полиса и дату рождения. Эти данные в системе не запоминаются, а просто в дальнейшем идентифицируют пациента при выдаче талона на прием.

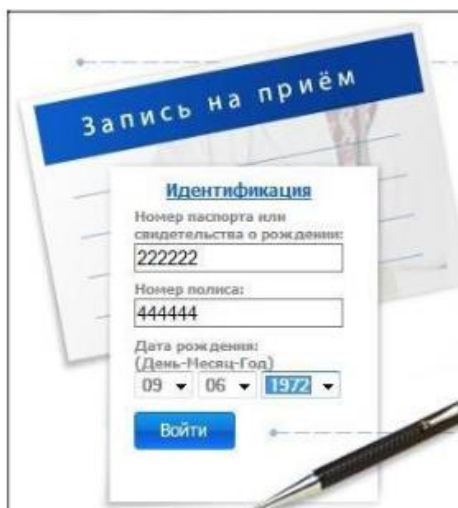


Рисунок 1.1 – Авторизация пациента

После авторизации, пациент попадает на страницу «Моя регистратура», где может выбрать нужную специализацию врача для приема (рисунок 1.2).

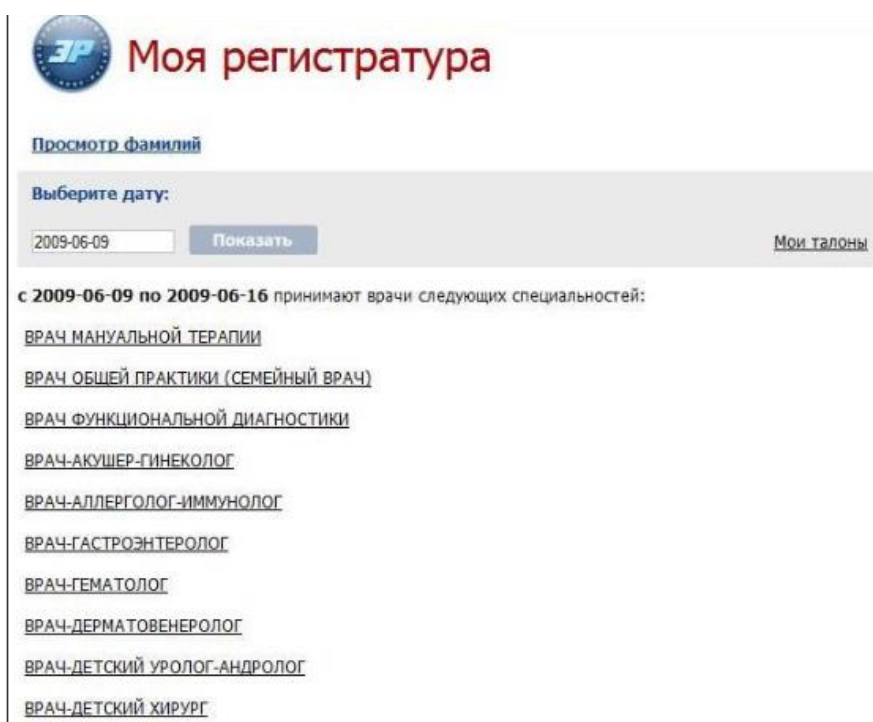


Рисунок 1.2 – Выбор нужной специализации врача

После выбора нужной специализации врача, система выдаст информацию о расписании врачей (дата приема, время приема, количество доступных талонов, кабинет врача), просмотр данных приема приведена на рисунке 1.3.



**Моя регистратура**

[Просмотр специальностей](#) | [Просмотр фамилий](#)

Выберите дату:

2009-06-10  [Мои талоны](#)

2009-06-10 по 2009-06-17 принимают следующие врачи:

В поле "Талоны" указано, общее кол-во талонов и кол-во талонов доступных через интернет.

Врач:	Дата:	Время:	Талоны/Доступно	Кабинет:
<b>БАЛАКИНА НАТАЛЬЯ СЕРГЕЕВНА</b> ВРАЧ-НЕВРОЛОГ	2009-06-14	9:00 - 15:10	28 / 0	204
	2009-06-15	9:00 - 15:10	19 / 0	
	2009-06-16	9:00 - 15:10	20 / 0	
	2009-06-17	9:00 - 15:10	19 / 0	
<b>ВЯТКИНА СВЕТЛАНА ЯКОВЛЕВНА</b> ВРАЧ-НЕВРОЛОГ	2009-06-11	9:00 - 15:10	28 / 0	320
	2009-06-13	9:00 - 15:10	28 / 0	
	2009-06-14	9:00 - 15:10	28 / 0	
	2009-06-15	9:00 - 15:10	28 / 0	
	2009-06-16	9:00 - 15:10	28 / 0	
	2009-06-17	9:00 - 15:10	28 / 0	
<b>ЗЕМСКОВ СЕРГЕЙ ГЕОРГИЕВИЧ</b> ВРАЧ-НЕВРОЛОГ	2009-06-11	10:24 - 15:39	6 / 1	236
	2009-06-15	10:24 - 16:49	4 / 0	
	2009-06-16	8:00 - 14:25	6 / 1	

Рисунок 1.3 – Выбор врача данной специализации, просмотр данных приема

Теперь пациент может выбрать желаемую дату приема к врачу и просмотреть количество свободных талонов по данной специализации.

Выбор даты приема к врачу приведена на рисунке 1.4.

В заключение, пациент подтверждает запись на приём и считается зарегистрированным. Просмотр выписанных талонов приведен на рисунке 1.5.

АИС «Электронная регистратура» Самарской областной поликлиники достаточно удобна и понятна для пациента [1]. Запись через веб-регистратуру экономит время: пациенту не нужно стоять в очереди, а можно записаться в поликлинику, будучи дома, на работе или даже стоя в пробке.

**Моя регистратура**

[Просмотр специальностей](#) / [Список врачей](#) / [Подробное расписание на 2009-06-16](#)

Выберите дату:

2009-06-16 [Показать](#) [Мои талоны](#)

Очистить [Заккрыть](#)

<Пред [Сегодня](#) След>

**Июнь 2009**

Пн	Вт	Ср	Чт	Пт	Сб	Вс
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Талон и кол-во талонов доступных через интернет.

Время приема:	Талоны	
8:00 - 8:35	1   0	
8:35 - 9:10	0   0	
9:10 - 9:45	1   0	
9:45 - 10:20	1   1	<a href="#">Взять талон</a>
10:20 - 10:55	1   0	

Рисунок 1.4 – Выбор желаемой даты приема к врачу, просмотр свободных талонов

**Моя регистратура**

[Просмотр специальностей](#) | [Просмотр фамилий](#)

**талон успешно зарегистрирован**

**ИВАНОВ ИВАН ИВАНОВИЧ**

Забронированные талоны:

Доктор: (42501) ЗЕМСКОВ СЕРГЕЙ ГЕОРГИЕВИЧ ((86) ВРАЧ-НЕВРОЛОГ)  
 Дата: 2009-06-16  
 Время: 9:45  
 Кабинет: 217

[Распечатать](#) | [Отменить талон](#)

Рисунок 1.5 – Просмотр выписанных талонов

## 1.2 АИС «Регистратуры поликлиники» в Белгородской области

Департамент здравоохранения и социальной защиты населения Белгородской области 07.09.2012 издал приказ № 2136 «Об организации записи на прием к врачам», согласно которому утвержден порядок записи на прием к врачу в программно-техническом комплексе «ТрастМед: медицинская информационная система» («ТМ:МИС») в медицинских организациях Белгородской области. Согласно этого приказа должны быть соблюдены следующие пункты:

а) формирование и постоянная актуализация расписания работы врачей всех профилей на срок 4 недели с добавлением 1 дня ежедневно;

б) своевременное внесение изменений в расписание приема в случае необходимости (по причине болезни врача, внепланового отпуска или других обстоятельств), а также уведомление пациентов об изменениях в расписании приема и возможностях переноса даты приема, предполагаемых сроках отсутствия данного врача, о его временных заместителях и графике их приема заранее любым возможным способом;

в) амбулаторный прием пациентов со сроком ожидания приема не более 15 минут от назначенного времени;

г) учет всех принимаемых пациентов в амбулаторных медицинских организациях (подразделениях медицинских организаций) в программно-техническом комплексе «ТМ:МИС» [4].

Для удобства граждан в медицинских организациях Белгородской области запись на прием должна осуществляться одним из способов:

а) посредством обращения в регистратуру медицинской организации;

б) по телефону регистратуры медицинской организации;

в) через электронные терминалы ("Инфоматы");

г) по электронной записи через сеть Интернет;

д) путем записи на прием врачом медицинской организации, назначившим повторный прием;

е) по записи на консультативный прием в другую медицинскую организацию по направлению врача медицинской организации [3].

Следует отметить, что современные МИС «Электронная регистратура» по своей сути – модульные системы, состоящие, в общем, из следующих подсистем (рисунок 1.6):

- регистратура поликлиники;
- электронная регистратура через терминал;
- онлайн регистратура через сеть Интернет.

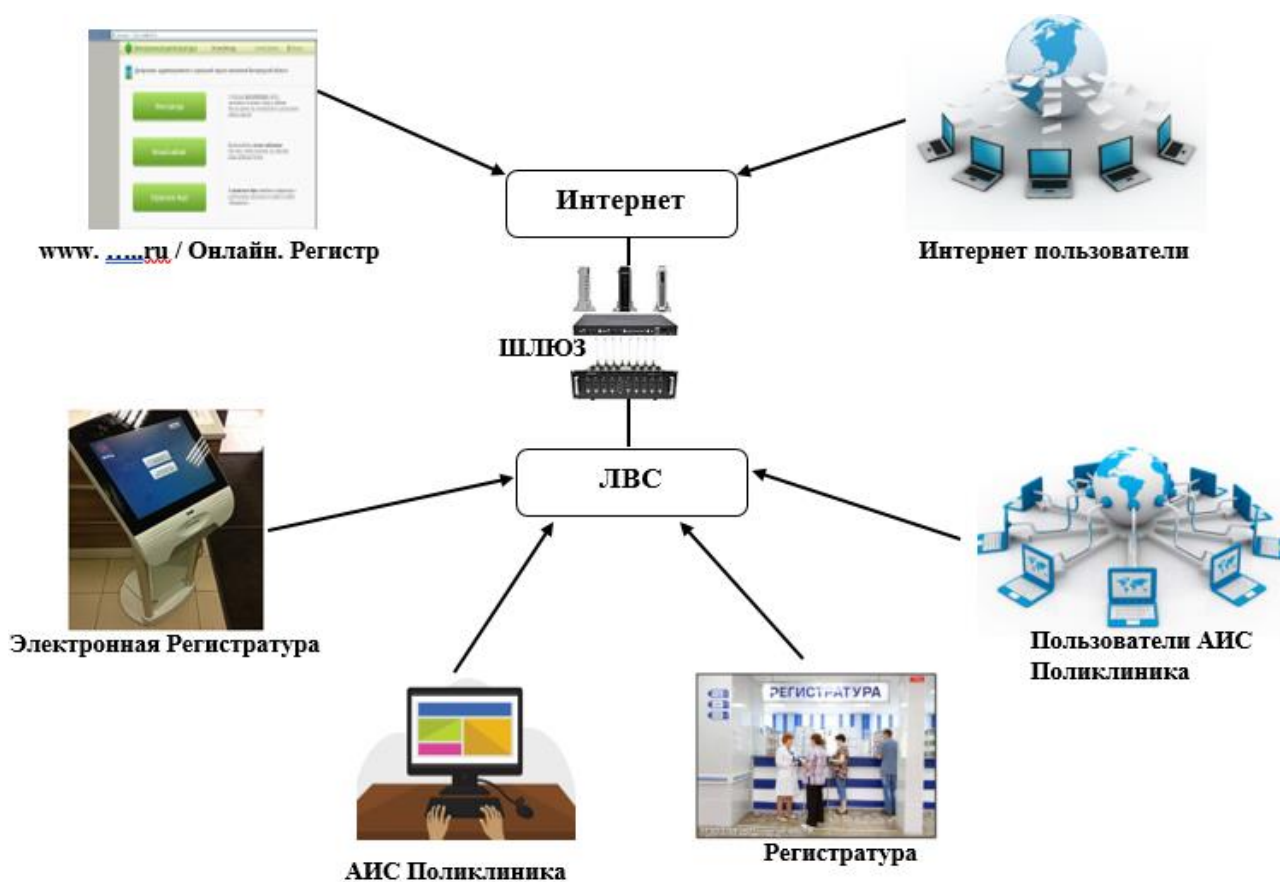


Рисунок 1.6 – Обобщенная функциональная схема МИС «Электронная регистратура»

Все эти подсистемы должны быть связаны и максимально быстро синхронизироваться между собой.

Онлайн-регистратура подразумевает некоторую общую систему для всех медицинских учреждений области, а электронная регистратура через терминал

и регистратура поликлиники – это системы, индивидуальные для каждой поликлиники.

«Электронная регистратура» в Белгородской области [3] – это онлайн-регистратура в виде веб-сайта, которая позволяет пациенту записаться на прием к любому врачу в любой поликлинике (подключенной в этой системе) Белгородской области в удобное для него время (рисунок 1.7). Удобство данной электронной регистратуры связано еще и с тем, что записаться на прием можно через веб-службу Госуслуги. Ограничением записи является только недоступность конкретного медицинского учреждения в сети. Это может быть связано с повреждением компьютерной сети в поликлинике или отсутствием системы электронной регистратуры самой поликлиники, подключенной к веб-регистратуре.

Стартовая веб-страница электронной регистратуры по Белгородской области приведена на рисунке 1.7.

The screenshot shows the 'Электронная регистратура' (Electronic Registration) website interface. At the top, there is a navigation bar with the site name, the date '19 мая 2018 года', and links for 'Личный кабинет' (Personal Cabinet) and 'Помощь' (Help). Below the navigation bar is a progress indicator with six steps: 1. Выбор населённого пункта (Selection of populated point), 2. Выбор мед. учреждения (Selection of medical institution), 3. Выбор специализации (Selection of specialization), 4. Выбор врача и даты (Selection of doctor and date), 5. Выбор времени посещения врача (Selection of doctor's visit time), and 6. Авторизация и получение талона (Authorization and receipt of ticket). The current step is '1. Выбор населённого пункта'. The main content area is titled 'Укажите населенный пункт' (Specify the populated point) and includes a sub-instruction: 'Выберите регион и населенный пункт, в котором предполагается посетить медицинское учреждение.' (Select the region and populated point where you plan to visit the medical institution). There is a search input field with a green button. Below the input field is a dropdown menu showing 'Белгородская область' (Belgorod region) with 22 populated points. The visible options are: Алексеевский район (10 medical institutions), Белгород (34 medical institutions), Белгородский район (17 medical institutions), Борисовский район (1 medical institution), Валуйский район (3 medical institutions), Вейделевский район (10 medical institutions), and Волоконовский район (1 medical institution).

Рисунок 1.7 – Выбор населенного пункта

Электронная регистратура 19 мая 2018 года Личный кабинет ? Помощь

1 Выбор населенного пункта 2 Выбор мед. учреждения 3 Выбор специализации 4 Выбор врача и даты 5 Выбор времени посещения врача 6 Авторизация и получение талона

### Выберите медицинское учреждение

Короткая ссылка на эту страницу: [er.zdrav31.ru/-putde](http://er.zdrav31.ru/-putde)  
Выберите медицинское учреждение из списка или воспользуйтесь поиском.

Белгород

Взрослые Детские Все

- Белгородский кожно-венерологический диспансер  
г. Белгород, Белгородский пр-т, д.97
- Белгородский областной центр по профилактике и борьбе со СПИД и инф. заболеваниями недоступна с  
14.03.2018 10:28
- Белгородский онкологический диспансер  
г. Белгород, ул. Куйбышева, 1
- БОКБ Святителя Иоасафа Центр медицинской реабилитации  
г. Белгород, ул.Костюкова, 67
- Городская поликлиника №1  
г. Белгород, Белгородский пр-т, 95а
- Городская поликлиника №2  
г. Белгород, Народный бульвар, 51
- Городская поликлиника №4  
г. Белгород, ул.Садовая, д. 17а

Рисунок 1.8 – Выбор медицинского учреждения

После выбора пациентом населенного пункта согласно своему месту жительства, открывается вкладка «Выбор медучреждения», приведенная на рисунке 1.8

Далее пациенту необходимо выбрать специализацию врача, к которому он бы хотел записаться на прием, и выбрать время и дату приема врача по свободным талонам. Выбор врача и даты приведен на рисунке 1.9

Как можно увидеть на рисунке 1.9, пациенту по цветовым подсказкам несложно выбрать время и дату приема необходимого врача или выбрать любого врача на удобное время. После сделанного выбора, пациенту нужно авторизоваться и получить талон. Страница авторизации приведена на рисунке 1.10

Авторизация пациента происходит через сайт Госуслуги по нажатию кнопки «Продолжить». Если пациент еще не зарегистрирован на сайте госуслуг, то ему необходимо пройти регистрацию, внести полис ОМС в личном кабинете госуслуг, и вернуться на сайт электронной регистратуры.

**Электронная регистратура** 19 мая 2018 года Личный кабинет ? Помощь

1 Выбор населённого пункта 2 Выбор мед. учреждения 3 Выбор специализации 4 **Выбор врача и даты** 5 Выбор времени посещения врача 6 Авторизация и получение талона

### Выберите врача и дату

■ — запись возможна  
■ — запись невозможна  
■ — запись через интернет невозможна

Уточнить расписание работы специалистов можно на сайте учреждения <http://beldoctor4.narod.ru>

Белгород  
Городская поликлиника №4  
Акушерство и гинекология

Показать только свободные дни с 19 мая по 25 мая Следующая неделя

Врач	Участок	Кабинет	Сб сегодня	Вс 20 мая	Пн 21 мая	Вт 22 мая	Ср 23 мая	Чт 24 мая	Пт 25 мая
Бриль Геннадий Максимович врач-акушер-гинеколог Кабинет акушера-гинеколога		403			8 <sup>00</sup> –14 <sup>00</sup> 0	14 <sup>00</sup> –20 <sup>00</sup> 0	8 <sup>00</sup> –13 <sup>30</sup> 8	14 <sup>00</sup> –20 <sup>00</sup> 1	8 <sup>00</sup> –14 <sup>00</sup> 9
Глоба Наталья Николаевна врач-акушер-гинеколог Кабинет акушера-гинеколога		402			14 <sup>00</sup> –20 <sup>00</sup> 0	8 <sup>00</sup> –14 <sup>00</sup> 0	14 <sup>00</sup> –20 <sup>00</sup> 0	8 <sup>00</sup> –14 <sup>00</sup> 0	14 <sup>00</sup> –20 <sup>00</sup> 0
Зубова Валентина Ивановна врач-акушер-гинеколог Кабинет акушера-гинеколога		404			8 <sup>00</sup> –14 <sup>00</sup> 0	14 <sup>00</sup> –20 <sup>00</sup> 0	8 <sup>00</sup> –13 <sup>30</sup> 0	14 <sup>00</sup> –20 <sup>00</sup> 0	8 <sup>00</sup> –14 <sup>00</sup> 0
Клиасак Ростислав Александрович врач-акушер-гинеколог Кабинет акушера-гинеколога		405			14 <sup>00</sup> –20 <sup>00</sup> 1	8 <sup>00</sup> –14 <sup>00</sup> 6	14 <sup>00</sup> –20 <sup>00</sup> 6	8 <sup>00</sup> –14 <sup>00</sup> 5	14 <sup>00</sup> –20 <sup>00</sup> 9

Рисунок 1.9 – Выбор врача и дата

**Электронная регистратура** 19 мая 2018 года Личный кабинет ? Помощь


1 Выбор населённого пункта 2 Выбор мед. учреждения 3 Выбор специализации 4 Выбор врача и даты 5 Выбор времени посещения врача 6 **Авторизация и получение талона**

### Авторизация

Если Ваши данные изменились (Ф.И.О., страховой медицинский полис, паспорт, СНИЛС, место регистрации или др.) необходимо самостоятельно внести изменения в «Личном кабинете» портала Госуслуг и обязательно обратиться непосредственно в медицинское учреждение, к которому Вы прикреплены, для внесения корректировки

Белгород  
Городская поликлиника №4  
Акушерство и гинекология, кабинет 403  
Бриль Геннадий Максимович  
врач-акушер-гинеколог  
23 мая 2018г., 10:42

Для входа в личный кабинет пациента или записи на прием необходимо авторизоваться через сайт госуслуг.

 **Продолжить**

Пошаговая видео-инструкция по записи на прием

Если Вы еще не зарегистрированы на сайте госуслуг, пройдите регистрацию, внесите полис ОМС в личном кабинете госуслуг, и вернитесь на этот сайт.

Рисунок 1.10 – Авторизация и получения талона

Эффективность «Электронной регистратуры» существенна, и об этом после внедрения сообщают многие пользователи. Главным образом внедрение электронной регистратуры содействует равномерному распределению потока пациентов между медицинскими учреждениями, сокращению времени ожидания записи на прием, времени ожидания приема врачей. Практически все пользователи констатируют, что имеет место существенное сокращение очередей в регистратурах поликлиник, и чем активнее население пользуется сайтом «Электронной регистратуры», тем выше результат пользователей [2].

Как уже отмечалось, на сегодняшний день не каждая поликлиника имеет собственную электронную регистратуру, связанную в электронной регистратурой в Белгородской области. Не имеет такой регистратуры и поликлиника НИУ БелГУ. Поэтому актуальной задачей является разработка автоматизированной системы регистратуры поликлиники НИУ БелГУ с возможностью записи на прием через систему Госуслуги.



## 2 Разработка автоматизированной системы регистратуры «НИУ БелГУ»

В поликлинике «НИУ БелГУ» имеется штат специалистов, которые принимают пациентов по предварительной записи. Оплата осуществляется после приема у врача или проведения лабораторных анализов.

Современная клиника является специализированным лечебно-профилактическим учреждением, предназначенным оказывать медицинскую помощь и осуществлять комплекс профилактических мероприятий по выявлению болезней у пациентов и их лечению.

В ее функции входят:

- оказание медицинской помощи;
- проведения лабораторных анализов;
- преждевременное выявление болезней.

Предоставление услуг пациенту начинается с его регистрации. При регистрации клиент предоставляет необходимую о себе информацию и документы: паспорт, СНИЛС, полис ОМС. Далее, пациент выбирает врача и удобное ему время для приема. При выборе врача и свободного времени, ему предоставляется соответствующая информация. После регистрации пациента и записи его на прием, вся информация заносится в базу данных.

В выпускной квалификационной работе необходимо разработать АИС регистратуры поликлиники НИУ БелГУ.

### 2.1 Техническое задание на разработку автоматизированной системы регистратуры поликлиники «НИУ БелГУ»

Автоматизированная система регистратуры поликлиники (АСРП) НИУ БелГУ предназначена для повышения эффективности работы регистратуры поликлиники.

### 2.1.1 Основание для разработки

Основанием для разработки медицинской информационной системы является необходимость автоматизации процесса регистрации пациентов, ведение медицинских карточек в поликлинике НИУ БелГУ.

### 2.1.2 Назначение разработки

Под автоматизацией регистратуры поликлиники подразумевается решение следующих задач:

а) хранение информации о пациентах (паспортные данные, СНИЛС, полис ОМС), о результатах анализов, истории болезни;

б) запись на прием к врачу;

в) разграничение прав доступа к системе учета для персонала разного уровня, специализации и др.;

г) хранение информации о персонале;

д) интеграция регистратуры поликлиники НИУ БелГУ с онлайн-регистратурой по Белгородской области с обслуживанием через сервис Госуслуги.

### 2.1.3 Требования к АСРП НИУ БелГУ

Система должна обеспечивать следующие функции:

1) регистратура поликлиники НИУ БелГУ должна быть связана с онлайн-регистратурой по Белгородской области через сервис Госуслуги;

2) ввод, вывод, редактирование, хранение информации о пациентах:

- ФИО;
- дата рождения;
- пол;
- СНИЛС;

- ИНН;
- полис ОМС;
- серия и номер паспорта;
- адрес;
- контактная информация;
- соц. статус;
- место учебы/работы.

3) ввод, вывод, редактирование, хранение информации о персонале:

- ФИО;
- логин;
- пароль;
- роль;
- специализация;
- график работы.

4) ввод, вывод, редактирование, хранение информации о записях на прием:

- дата и время;
- ФИО пациента;
- ФИО и специализация врача.

5) ввод, вывод, редактирование, хранение карты пациента:

- дата приема;
- врач;
- симптомы, диагноз, лечение, комментарии.

б) ввод, вывод, редактирование, хранение о результатах анализов:

- дата;
- вид анализа;
- результат.

Входной информацией системы является:

- а) личные данные пациента и информация о нем;
- б) информация о персонале;

в) логины и пароли персонала;

г) заключения врачей;

Выходной информацией системы является:

а) управляющие воздействия на базу данных;

б) квитанции об оплате;

в) заключения врачей;

г) результаты анализов.

По требованиям к надежности система должна:

- проводить контроль вводимой информации;
- блокировать некорректные действия пользователя при работе с системой;
- обеспечивать целостность данных;
- обеспечивать права доступа;
- обеспечивать защиту персональных данных.

В связи с тем, что использовать систему будут пользователи средней и низкой квалификации, то интерфейс системы должен быть простым и интуитивно понятным. Ввод информации должен осуществляться в наиболее унифицированных формах.

Требования к составу и параметрам технических средств: настоящая система должна работать на процессорах совместимых с процессором IBM; оперативная память на каждой ЭВМ не менее 128 Мб; наличие доступа к локальной сети и сети Интернет; свободное место на жестком диске не менее 10Мб.

Требования к информационной и программной совместимости: система должна работать под управлением ОС Windows; СУБД PostgreSQL; другое ПО выбирается по решению разработчика. Основным критерием является низкая стоимость.

#### 2.1.4 Технико-экономические показатели

Затраты на разработку АСРП НИУ БелГУ сведены в таблицу 1.

Таблица 1 – Смета затрат на разработку АСРП НИУ Белгу

Элементы затрат	Стоимость, руб.
Компьютер Pentium G4400	25 000
Хозяйственный инвентарь (мебель)	10 000
Амортизация оборудования и инструментальные средства	8 000
Хоз. материалы	500
Электроэнергия	500
Накладные расходы	10 000
Всего	5400

#### 2.1.5 Стадии и этапы разработки

Общая продолжительность разработки и внедрения системы составляет 4 месяца с 01/03/2018 по 08/06/2018. Общая стоимость работ составляет 54000 руб. График реализации проекта представлен в таблице 2.

Таблица 2 – График реализации проекта

Этапы реализации проекта	Месяцы			
	1	2	3	4
1. Разработка АСРП НИУ БелГУ				
1.1. Техническое задание.	■			
1.2. Эскизный проект.	■	■		
1.3. Технический проект.	■	■		
1.4. Рабочий проект.	■	■		

Продолжение таблицы 2

Этапы реализации проекта	Месяцы			
	1	2	3	4
1.5. Внедрение.				
2. Покупка ЭВМ, оборудования и инструментальных средств для заказчика (осуществляется за средства заказчика).				
3. Обучение персонала (осуществляется за средства заказчика, согласно отдельному договору).				
4. Эксплуатация АСРП НИУ БелГУ (пробная эксплуатация сотрудниками заказчика под контролем разработчика).				

## 2.2 Разработка модели использования

Диаграмма прецедентов определяет границы и контекст АСРП НИУ БелГУ на ранних этапах проектирования, позволяет структурировать основные функции, общие требования к поведению проектируемой системы, т.е. является полностью документированной моделью предполагаемого поведения системы. Базовыми элементами диаграммы являются актер и вариант использования. Актер – согласованное множество ролей, которые играют внешние сущности по отношению к вариантам использования при взаимодействии с ними. Вариант использования, или прецедент, – внешняя спецификация последовательности действий, которые система или другая сущность могут выполнять в процессе взаимодействия с актерами.

Диаграмма прецедентов для моделируемой системы «Регистратуры поликлиники НИУ БелГУ» представлена на рисунке 2.1.

Субъекты – пациент, регистратор, врач и системный администратор.

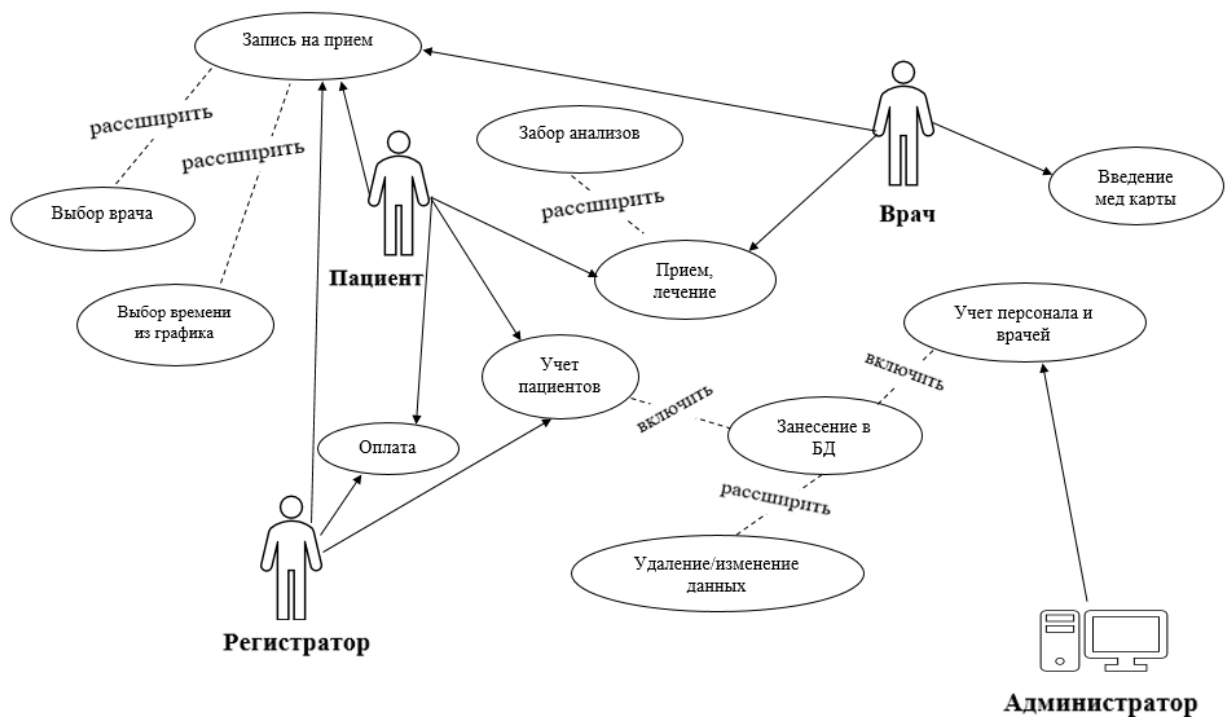


Рисунок 2.1 – Диаграмма вариантов использования

В таблице 3 представлено описание представленных прецедентов, где указаны требования, которые должны выполняться, чтобы участвовали те или иные субъекты и выполнялся связанный с ними прецедент.

Таблица 3 – Распределение требований по субъектам и прецедентам

п/п	Требование	Субъект	Прецедент
1	Пациент записывается на прием к врачу, регистратор или врач вносит данные клиента в базу	Пациент, Регистратор или Врач	Запись на прием
2	Пациент оплачивает оказанные ему услуги, регистратор принимает оплату	Пациент, Регистратор	Оплата

Продолжение таблица 3

3	Врач проводит осмотр, выносит врачебное заключение, выписывает лечение, Пациент получает информацию	Пациент, Врач	Приём, лечение
4	Пациент предоставляет данные, регистратор вносит данные в БД	Пациент, регистратор	Учет пациентов
5	Администратор ведет учет сотрудников	Администратор	Учет персонала и врачей
6	Врач заполняет мед. карту принятого пациента	Врач	Ведение мед карты

### 2.3 Моделирование информационной системы

В качестве средства моделирование АСРП НИУ БелГУ была выбрана программа AllFusion ProcessModeler 7 – мощный инструмент моделирования, который используется для анализа, документирования и реорганизации сложных процессов, в том числе, бизнес-процессов. Модель, созданная средствами AllFusion, позволяет четко документировать различные аспекты деятельности – действия, которые необходимо предпринять, способы их осуществления, требующиеся для этого ресурсы и др.

AllFusion совмещает в одном инструменте средства моделирования функций (IDEF0), потоков данных (DFD) и потоков работ (IDEF3).

Контекстная диаграмма регистратуры поликлиники представляет собой самое общее описание системы и ее взаимодействие с внешней средой. Контекстная диаграмма состоит из одного блока, описывающего функцию верхнего уровня, ее входы, выходы, управления, и механизмы, вместе с формулировками цели модели и точки зрения, с которой строится модель.



Контекстная диаграмма АСРП приведена на рисунке 2.2.



Рисунок 2.2 – Контекстная диаграмма АСРП «НИУ БелГУ»

На контекстной диаграмме входной информацией является информация о пациенте, которую может вводить как персонал поликлиники, так и сам пациент через сервис Госуслуги. Работа регистратуры поликлиники и защита персональных данных регламентируются законами РФ. Результатом работы АСРП НИУ БелГУ является запись пациента на прием к конкретному врачу на определенное время и выдача пациенту в случае необходимости квитанцию на оплату услуг.

Диаграмма декомпозиции предназначена для детализации функций и получается при разбиении контекстной диаграммы на крупные подсистемы (функциональная декомпозиция) и описывает каждую подсистему и их взаимодействие.

Декомпозиция контекстной диаграммы показана на рисунке 2.3. На этой диаграмме было выделено 4 подсистемы:

- регистрация;
- прием у врача;
- сдача анализов;

- оплата услуг.

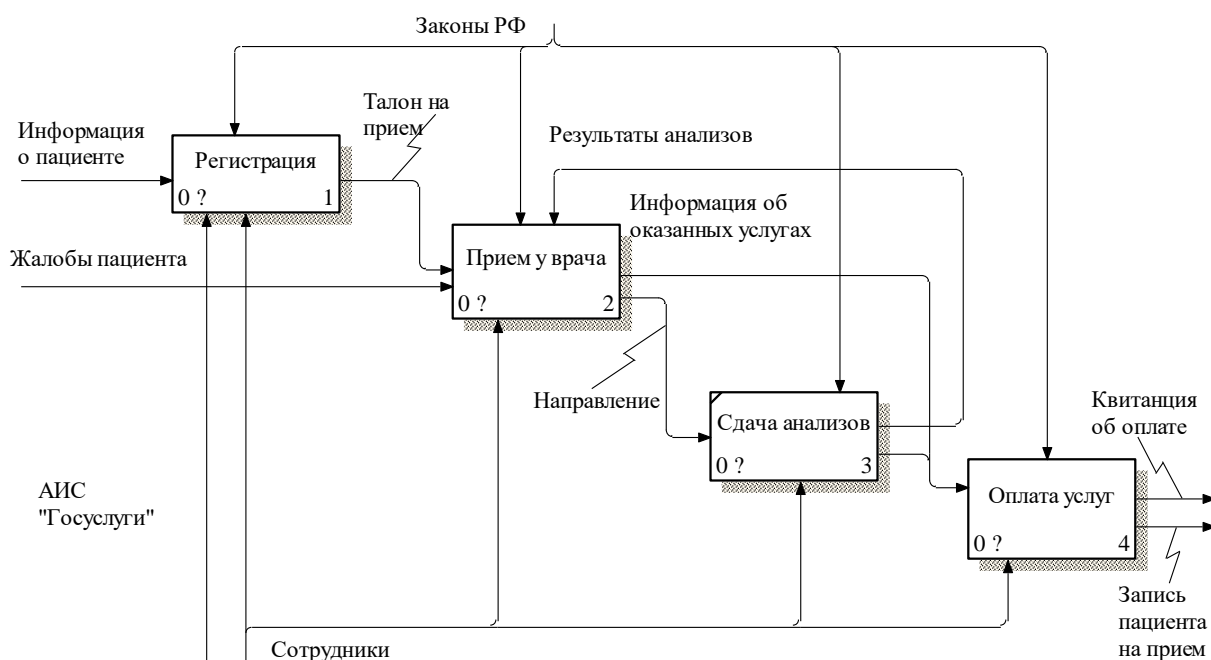


Рисунок 2.3 – Декомпозиция контекстной диаграммы

Внесение личных данных и выбор необходимого врача на определенное время осуществляется персоналом регистратуры поликлиники или лично пациентом через службу Госуслуги (подсистема «Регистрация»). На основании этой информации формируется талон на прием, бумажная копия которого остается у пациента, а электронный вариант – появляется у врача. Врач оказывает услуги пациенту (подсистема «Прием у врача») и в системе отмечает перечень оказанных услуг, а также может сформировать направления на анализы (подсистема «Сдача анализов»). На основании этих данных пациенту выписывается квитанция об оплате услуг (подсистема «Оплата услуг»). При необходимости пациент может записаться к этому же врачу повторно или же выбрать другого врача.

Декомпозиция основных блоков проведена далее. На рисунке 2.4 показана декомпозиция блока «Регистрация».

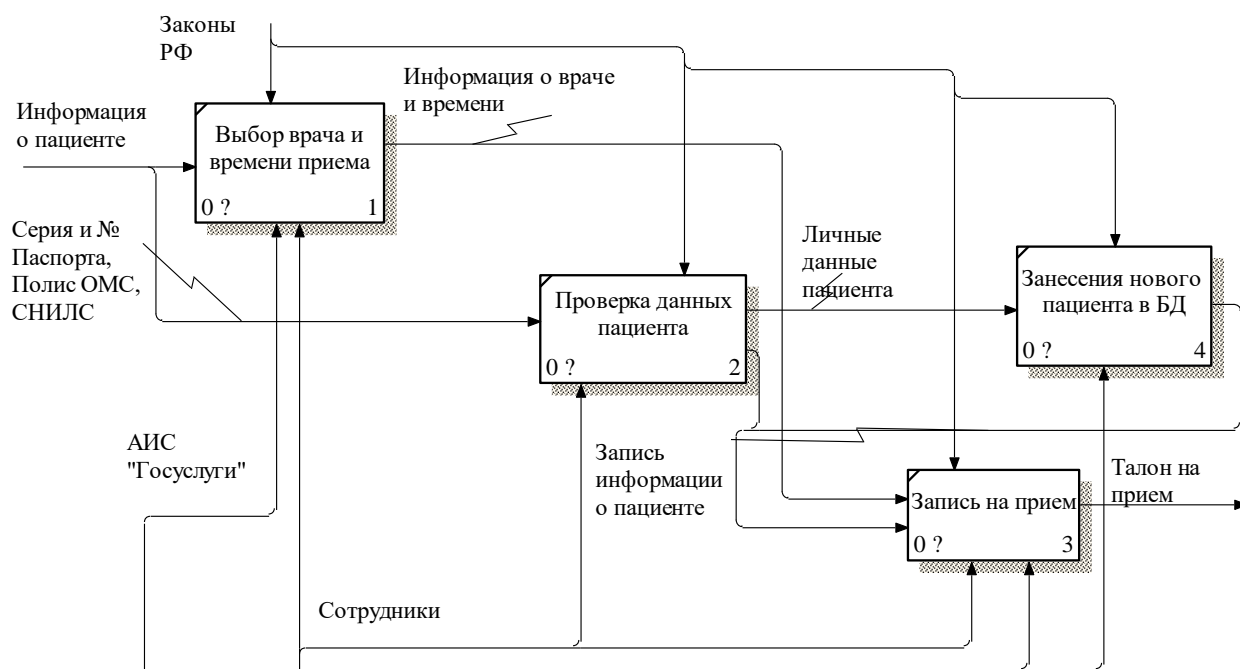


Рисунок 2.4 – Декомпозиция блока "Регистрация"

Декомпозиция блока «Регистрация» включает в себя:

- выбор врача и времени приема;
- проверка данных пациента;
- занесение нового пациента в БД (событие происходит если пациент посещает поликлинику в первый раз);
- запись на прием.

Талон на прием пациенту выписывается на основании пожеланий самого пациента в плане выбора врача и удобного ему времени посещения, а также документов, удостоверяющих личность, СНИЛС, полис ОМС (блок «Выбор врача и времени приема»).

В случае, если пациент первый раз пришел в поликлинику НИУ БелГУ, т.е. регистратор не находит его данные в базе данных, для пациента создается новая электронная медицинская карта (блок «Занесение нового пациента в БД») и только после этого регистратор может записать пациента на прием (блок «Запись на прием»). Если пациент является сотрудником или обучающимся НИУ БелГУ, в медицинской карте делается пометка и в

дальнейшем часть услуг и анализов пациентом не оплачивается или оплачивается со скидкой.

На рисунке 2.5 представлена декомпозиция блока «Прием у врача», которая содержит такие структурные элементы, как:

- осмотр;
- направление на анализы (блок выполняется по усмотрению врача);
- постановка диагноза;
- назначение лечения;
- запись результатов приема в карточку пациента.

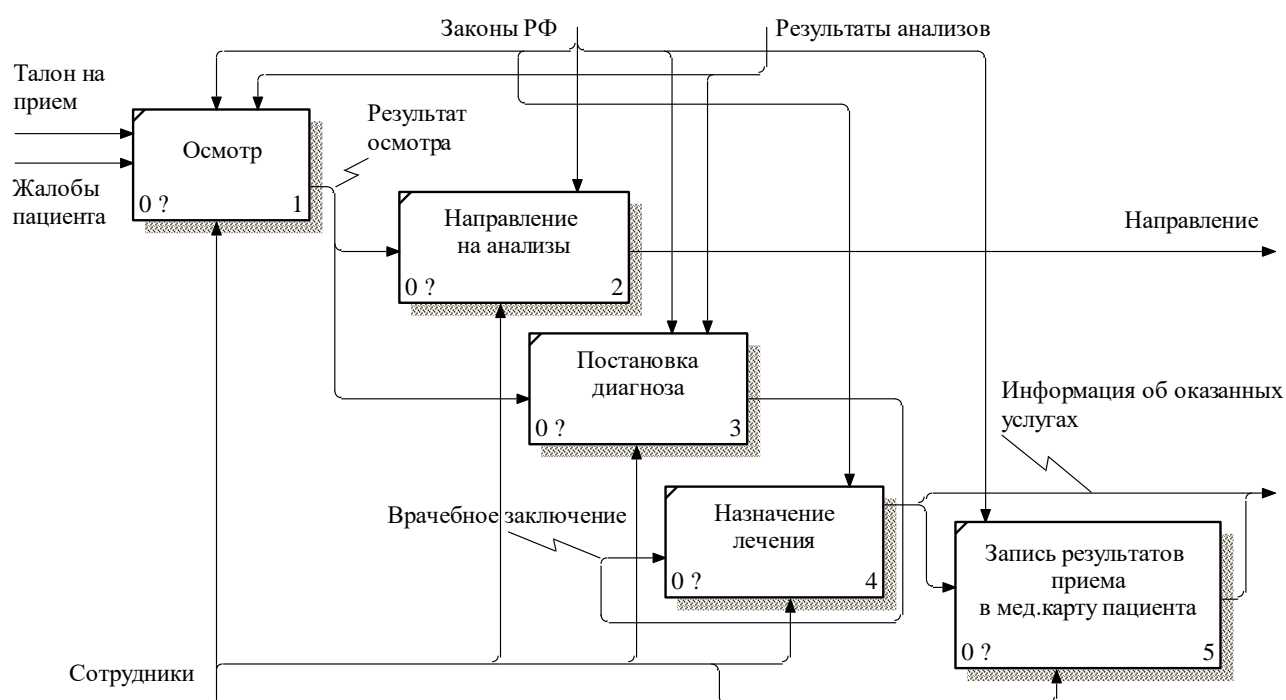


Рисунок 2.5 – Декомпозиция блока "Прием у врача"

Осмотр пациента врачом зависит от жалоб пациента; врач может выдать пациенту направление на анализы, результаты которых при повторном приеме также влияют на результаты осмотра и постановку диагноза. Врач фиксирует в медицинской карте пациента результаты осмотра и указания для дальнейшего лечения, а также отмечает в системе пункты оказанных услуг, которые пациент должен оплатить.

Детализация блока «Оплата услуг» представлена на рисунке 2.6 диаграммой IDEF3.

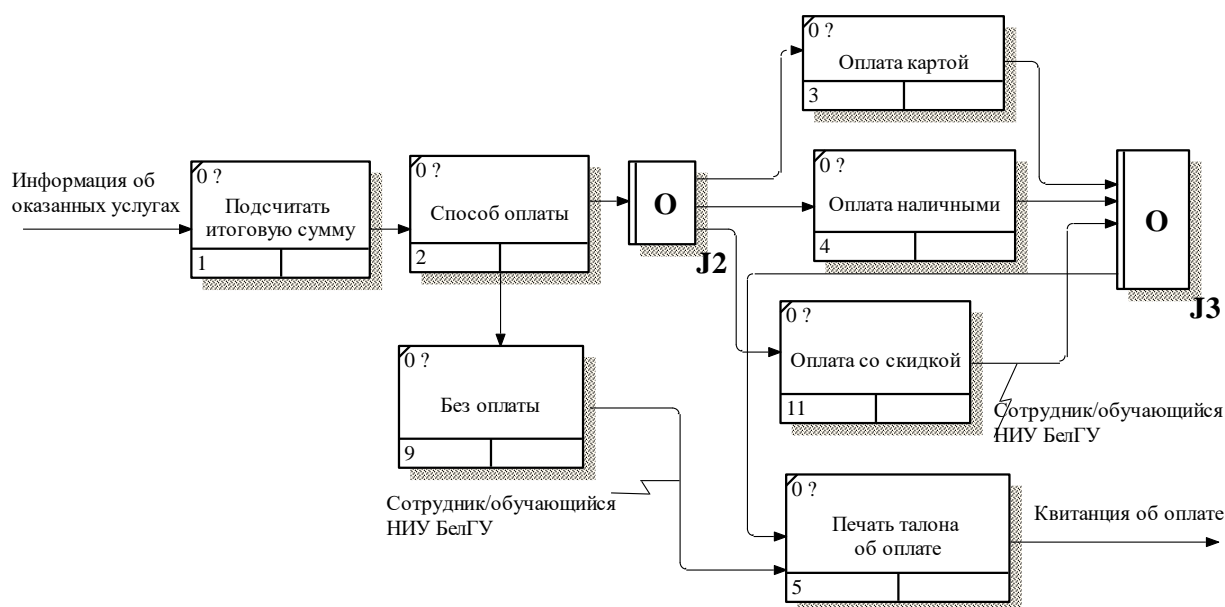


Рисунок 2.6 – Декомпозиция блока "Оплата услуг"

Таким образом, в результате была смоделирована автоматизированная система регистратуры поликлиники НИУ БелГУ, используя нотацию IDEF0 и IDEF3, что позволило структурировать логику и взаимодействие процессов организации. Моделирование началось с наиболее абстрактного уровня описания системы в целом. Было установлено, что входит в систему, а что рассматривается как внешнее воздействие. Далее была произведена декомпозиция выбранных блоков для рассмотрения внутренних процессов. Модель представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм.

#### 2.4 Инфологическое проектирование базы данных

Начальным этапом проектирования базы данных является инфологическое проектирование, основная задача которого состоит в определении предметной области автоматизированной системы и формирование взгляда на информационную систему с позиций пользователей БД.

Инфологическая модель – формализованное описание предметной области, которое будет «легко» читаться не только специалистами по базам данных, но пользователями этой базой.

Инфологическое проектирование отражает в себе смысл базы данных и разрабатывается с помощью информационных объектов. Основными элементами инфологических моделей являются сущности и связи между ними. Сущность моделирует класс однотипных объектов, который имеет уникальное имя, информацию об этих объектах необходимо хранить в базе данных. Под сущностью понимается некоторый объект, идентифицируемый пользователем в предметной области. Сущность имеет уникальное имя в пределах моделируемой системы. Объект, которому соответствует понятие сущности, имеет свой набор атрибутов – характеристик, определяющих свойства данного представителя класса. Между сущностями могут быть установлены связи – бинарные ассоциации, показывающие соотношение или взаимодействие сущностей между собой [7].

Для автоматизированной информационной системы регистратуры поликлиники НИУ БелГУ выделены следующие сущности:

- регистрация;
- персонал;
- пациент;
- прием;
- медицинские карты;
- график работы;
- специализации
- роли;
- роль персонал.

На основании приведенных выше сущностей, построена модель схемы отношений с помощью среды Erwin Data Modeler [8], которая представлена на рисунке 2.7

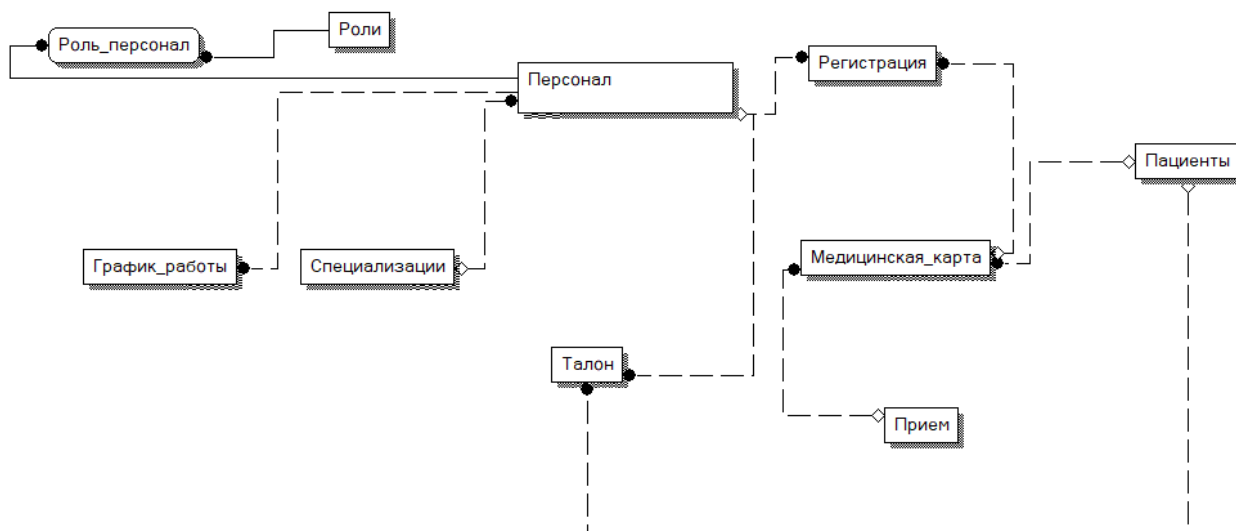


Рисунок 2.7 – Модель схемы отношений

Связь работает путем сопоставления данных в ключевых столбцах; обычно это столбцы с одним и тем же именем в обеих таблицах. Существует три типа связей между таблицами. Тип создаваемой связи зависит от того, как определены связанные столбцы.

Связь «один ко многим» (1 – n) самая распространенная. В этом типе связей у строки таблицы А может быть несколько совпадающих строк таблицы Б, но каждой строке таблицы Б может соответствовать только одна строка из А [9].

На схеме, представленной на рисунке 2.7, такая связь существует между следующими таблицами:

- «персонал» и «регистрация»: поликлиника имеет несколько сотрудников и каждый сотрудник зарегистрирован в регистратуре;
- «регистрация» и «медицинская карта»: в регистратуре хранится несколько мед.карт пациентов поликлиники, но каждый мед.карта принадлежит только одному пациенту;
- «пациенты» и «талон»: каждый пациент может иметь более одного талона, но каждый талон принадлежит только одному пациенту;

- «персонал» и «талон»: каждый сотрудник может создать более одной записи на прием, но запись на прием по талону принадлежит только одному сотруднику;

- «персонал» и «специализации»: несколько сотрудников могут иметь одну специализацию, но каждая специализация относится только одному сотруднику;

- «персонал» и «график работы»: каждый сотрудник имеет динамический график работы, но каждое конкретное время и место работы относится только одному сотруднику;

- «персонал» и «роль»: каждый сотрудник может выполнять несколько ролей, но каждая роль относится только одному сотруднику.

В связи «многие ко многим» ( $n-n$ ) строке таблицы А может сопоставляться несколько строк таблицы Б, и наоборот. Такие связи создаются определением третьей таблицы, которая называется таблицей соединения, чей первичный ключ состоит из внешних ключей А и Б. Такие связи в схеме, представленной на рисунке 2.7 нет.

В связи «один к одному» ( $1-1$ ) строке таблицы А может сопоставляться только одна строка таблицы Б, и наоборот [10]. Такие связи в схеме присутствуют между таблицами «пациент» и «медицинская карта»: каждый пациент имеет только одну мед.карту, и каждая мед.карта может относиться только в одному пациенту.

## 2.5 Даталогическая модель

Следующим шагом в проектировании является даталогическое проектирование. Под даталогической понимается модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. При этом даталогическая модель разрабатывается с учетом конкретной реализации СУБД на основе ее



инфологической модели. Спроектированная даталогическая модель базы данных изображена на рисунке 2.8.

Даталогическое проектирование заключается в разработке схемы базы данных, то есть совокупности схем отношений, которые адекватно моделируют сущности и семантические связи между ними. Основой правильности схемы являются функциональные зависимости между атрибутами БД [11].

В даталогической модели каждой сущности в соответствие ставится отношение, должны быть расставлены первичные и вторичные ключи, все отношения должны быть приведены к нормальной форме.

Все отношения находятся в третьей нормальной форме. Даталогическая модель разрабатываемой системы регистратуры поликлиники НИУ БелГУ представлена на рисунке. 2.8

Для построения инфологической и даталогической модели был использован язык ER-диаграмм.

В базе данных АИС регистратуры поликлиники НИУ БелГУ будет 10 сущностей:

- а) «Пациенты» – содержит данные о пациентах;
- б) «Персонал» – содержит информацию о персонале;
- в) «Специализации» – содержит информацию о существующих специализациях врачей в поликлинике;
- г) «Роли» – содержит данные о существующих ролях сотрудников в поликлинике;
- д) «Роль персонал» – содержит номер роли и номер сотрудника (персонала);
- е) «График работы» – содержит данные о режиме работы персонала поликлиники;
- ж) «Регистрация» – содержит данные о записях пациентов на прием к врачам;
- з) «Прием» – содержит данные о результате приема пациента у врача;

и) «Медицинская карта» – содержит истории болезней пациентов.

к) «Талон» – содержит данные сотрудника, дату и время приема пациента.

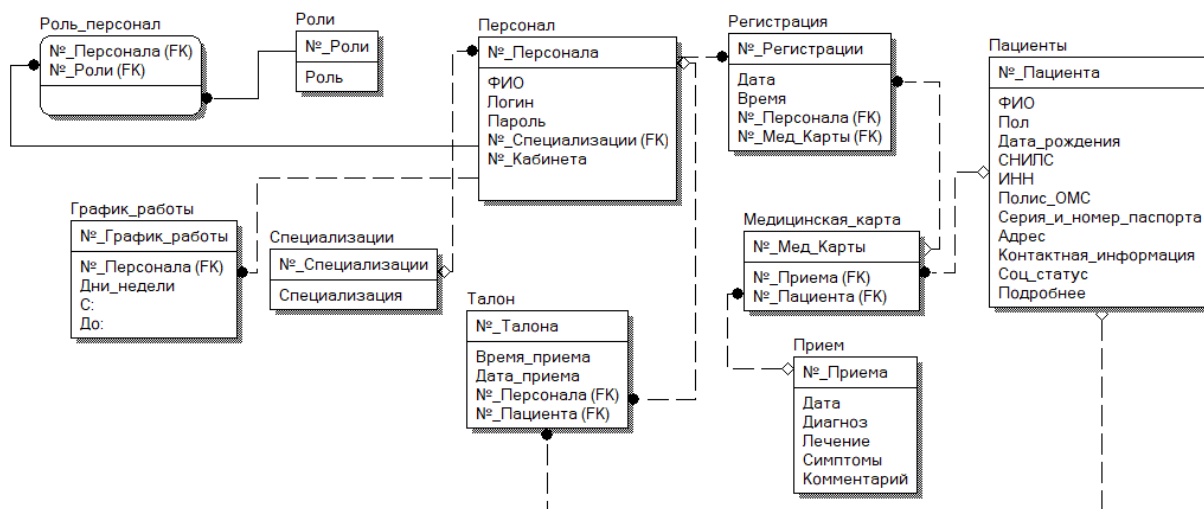


Рисунок 2.8 – Даталогическая модель

Эти сущности и атрибуты являются основой для создания таблиц в базе данных для дальнейшего хранения информации. Изменение данных в одной таблице повлечет за собой изменения в другой, таким образом обеспечивается целостность данных.

Для создания таблиц было создано всего 10 сущностей, поскольку этого достаточно для обеспечения полноценной работы разрабатываемой системы.

Таблица «patient» (рисунок 2.9) – это таблица «Пациентов», которая используется для хранения данных о пациентах.

Содержит следующие поля:

- «id\_patient» – номер пациента;
- «FIO» – ФИО;
- «pol» – пол;
- «INN» – ИНН;
- «SNILS» – СНИЛС;
- «polis\_OMS» – полис ОМС;
- «series\_pas\_num» – серия и номер паспорта;
- «adres» – адрес;

- «date\_of\_birth» – дата рождения;
- «contact\_inform» – контактная информация;
- «sotrudink\_belgu» – сотрудник «НИУ БелГУ»;
- «student\_belgu» – обучающийся «НИУ БелГУ»;
- «soc\_status» – социальный статус;
- «podrobnee» – подробнее.

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_patient	smallint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	info	character varying	100,000		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	pol	character varying	20		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	date_of_birth	date			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	snils	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	inn	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	polis_oms	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	series_pas_num	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	adres	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	contact_inform	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	soc_status	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	podrobnee	character varying	10,000		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	sotrudnik_belgu	character varying	10,000		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	student_belgu	character varying	10,000		<input type="checkbox"/> No	<input type="checkbox"/> No

Рисунок 2.9 – Таблица «Пациента»

Таблица «med\_card» (рисунок 2.10) – это таблица «Медицинской карты», в которой ведется запись истории болезни пациента и назначаемого ему лечения.

Содержит следующие поля:

- «id\_card» – мед.карты;
- «id\_patient» – связь с таблицей «patient» пациент;
- «id\_reception» – связь с таблицей «reception» прием.

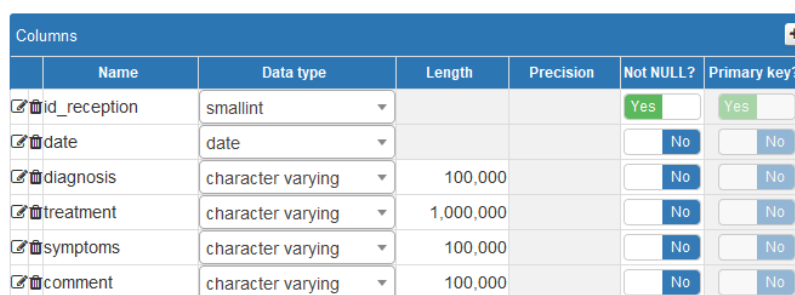
Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_card	smallint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	id_patient	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_reception	integer			<input type="checkbox"/> No	<input type="checkbox"/> No

Рисунок 2.10 – Таблица «Мед.карты»

Таблица «reception» (рисунок 2.11) – это таблица «Приема», в котором хранится запись на прием к врачу.

Содержит следующие поля:

- «id\_reception» – номер приема;
- «date» – дата приема;
- «diagnosis» – диагноз;
- «treatment» – лечение;
- «symptoms» – СИМПТОМЫ;
- «comment» – КОММЕНТАРИИ.



The screenshot shows a window titled 'Columns' with a table defining the structure of the 'reception' table. The table has seven columns: Name, Data type, Length, Precision, Not NULL?, and Primary key?. The 'id\_reception' column is the primary key and is not null. The other columns are not null and have varying lengths and data types.

	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_reception	smallint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	date	date			<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	diagnosis	character varying	100,000		<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	treatment	character varying	1,000,000		<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	symptoms	character varying	100,000		<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	comment	character varying	100,000		<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 2.11 – Таблица «Прием»

Таблица «personal» (рисунок 2.12) – это таблица «Персонал», в котором хранится информация о сотрудниках поликлиники.

Содержит следующие поля:

- «id\_personal» – номер сотрудника;
- «FIO» – ФИО;
- «login» – логин;
- «parol» – пароль;
- «id\_role» – связь с таблицей «role» роль;
- «id\_specialization» – связь с таблицей «specialization» специализация;
- «id\_office» – номер кабинета.

Таблица «specialization» (рисунок 2.13) – это таблица «Специализации», в котором имеется данные о специализации персоналов поликлиники.

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_personal	smallint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	FIO	character varying	10,000		<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	login	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	parol	text			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_role	smallint			<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_specialization	smallint			<input checked="" type="checkbox"/> Yes	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_office	integer			<input type="checkbox"/> No	<input type="checkbox"/> No

Рисунок 2.12 – Таблица «Персонал»

Содержит следующие поля:

- «id\_specialization» – номер специализации;
- «specialization» – номер кабинета.

Таблица «role» (рисунок 2.14) – это таблица «Роли», в которой хранятся данные о ролях сотрудников поликлиники.

Содержит следующие поля:

- «id\_role» – номер роли;
- «roles» – роли.

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_specialization	smallint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	specialization	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No

Рисунок 2.13 – Таблица «Специализация»

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_role	smallint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	roles	character varying	100		<input type="checkbox"/> No	<input type="checkbox"/> No

Рисунок 2.14 – Таблица «Роли»

Таблица «role\_personal» (рисунок 2.15) – это таблица «Роли персонала», в котором хранится данные о роли сотрудников поликлиники.

Содержит следующие поля:

- «id\_personal» – связь с таблицей «personal» персонал;

- «id\_role» – связь с таблицей «role» роли.

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_personal	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	id_role	integer			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Рисунок 2.15 – Таблица «Роль персонал»

Таблица «timeWorks» (рисунок 2.16) – это таблица «График работы», в которой хранится информация о графике работы сотрудников поликлиники.

Содержит следующие поля:

- «id\_time\_works» – номер графика работы;
- «id\_personal» – связь с таблицей «personal» сотрудников;
- «time\_from» – начало приема;
- «time\_to» – конец приема;
- «days\_week» – дни недели;
- «holiday» – отпуск.

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_time_works	smallint			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	id_personal	integer			<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	time_from	time without time zone	6		<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	time_to	time without time zone			<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	days_week	character varying	1,000		<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	holiday	character varying	1,000		<input type="checkbox"/>	<input type="checkbox"/>

Рисунок 2.16 – Таблица «График работы»

Таблица «registration» (рисунок 2.17) – это таблица «Регистратура», в которой хранится информация о графике работы, дате и времени приема пациентов, а также медицинские карты пациентов.

Содержит следующие поля:

- «id\_reg» – номер регистрации;
- «id\_personal» – связь с таблицей «personal» сотрудников;
- «id\_card» – связь с таблицей «med\_card» медицинские карты;
- «date» – дата;

- «time» – время.

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_reg	smallint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	id_personal	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_card	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	date	date			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	time	time without time zone		6	<input type="checkbox"/> No	<input type="checkbox"/> No

Рисунок 2.17 – Таблица «Регистратуры»

Таблица «ticket» (рисунок 2.18) – это таблица «Талон на прием», в котором хранится время и дата приема пациентов.

Содержит следующие поля:

- «id\_ticket» – номер талона на прием;
- «id\_personal» – связь с таблицей «personal» персонал;
- «id\_patient» – связь с таблицей «patient» пациент;
- «date» – дата приема;
- «time» – время приема.

Columns						
	Name	Data type	Length	Precision	Not NULL?	Primary key?
<input checked="" type="checkbox"/>	id_ticket	smallint			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
<input checked="" type="checkbox"/>	id_personal	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	id_patient	integer			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	date	date			<input type="checkbox"/> No	<input type="checkbox"/> No
<input checked="" type="checkbox"/>	time	time without time zone		6	<input type="checkbox"/> No	<input type="checkbox"/> No

Рисунок 2.18 – Таблица «Талон на прием»

Используемая СУБД – PostgreSQL. PostgreSQL является кроссплатформенным продуктом и работает не только в широком диапазоне диалектов UNIX (Linux, FreeBSD, Solaris и т.д.), но и на платформе Microsoft Windows. Работа с pgAdminIII показана на рисунке 2.19.

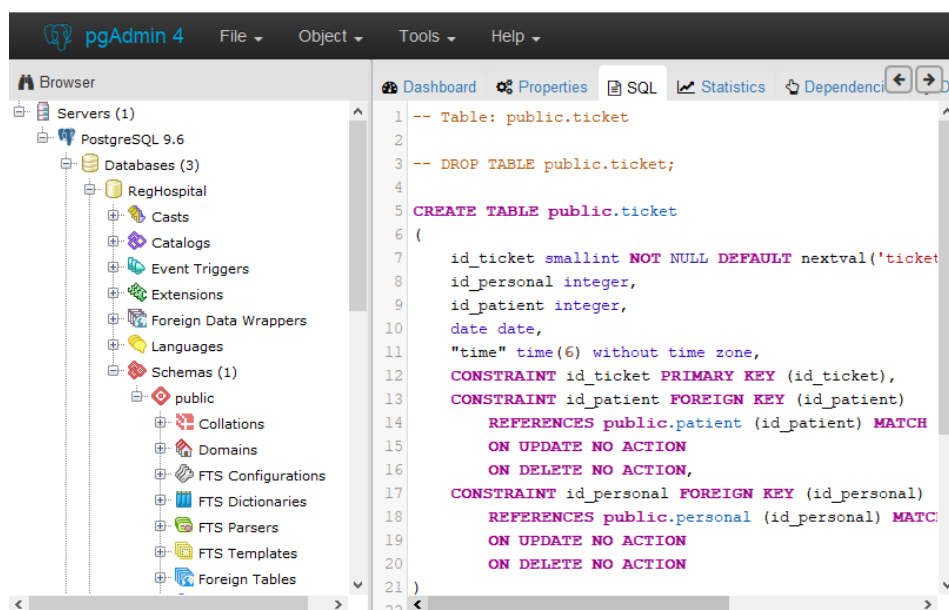


Рисунок 2.19 – Работа с pgAdminIII

С помощью pgAdmin очень удобно администрировать сервера PostgreSQL, а также создавать базы данных (все инструменты, которые нужны для этого, устанавливаются по умолчанию с самой программой) [12].



### 3 Программная реализация информационной системы

#### 3.1 Разработка и тестирование АИС регистратуры поликлиники «НИУ БелГУ»

В качестве средства разработки программы используется язык программирования C# и среда разработки Visual Studio 2015.

При запуске приложения появляется стартовое окошко (рисунок 3.1), в котором пользователь должен ввести свой логин и пароль, это дает возможность разграничить права доступа к разным функциям: в систему можно зайти как системный администратор, как работник регистратуры и как врач. Любой пользователь не может зарегистрироваться сам, это может сделать только системный администратор, путем добавления нового сотрудника в таблицу.

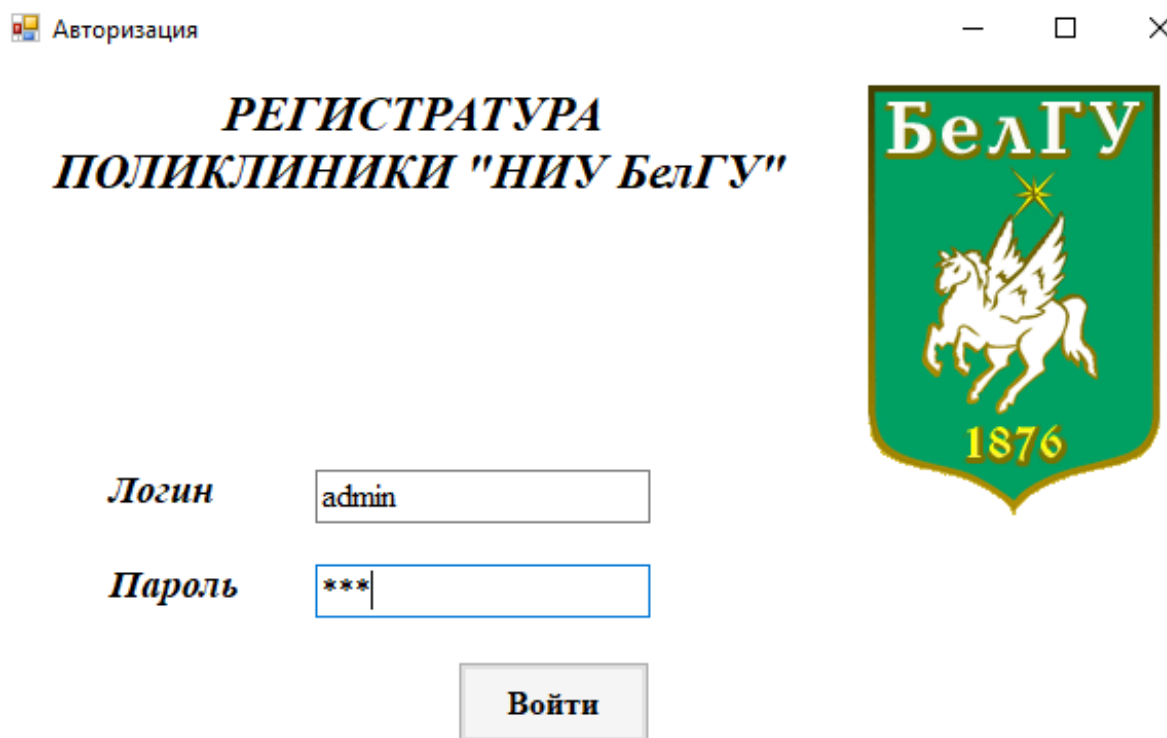


Рисунок 3.1 – Окно авторизации пользователей

В режиме работы от лица регистратора на главной форме появляется всего три кнопки, с помощью которых можно отредактировать таблицы Персонала или Пациентов, а также регистратор может записать на прием пациента. Главная форма для регистратора представлена на рисунке 3.2.

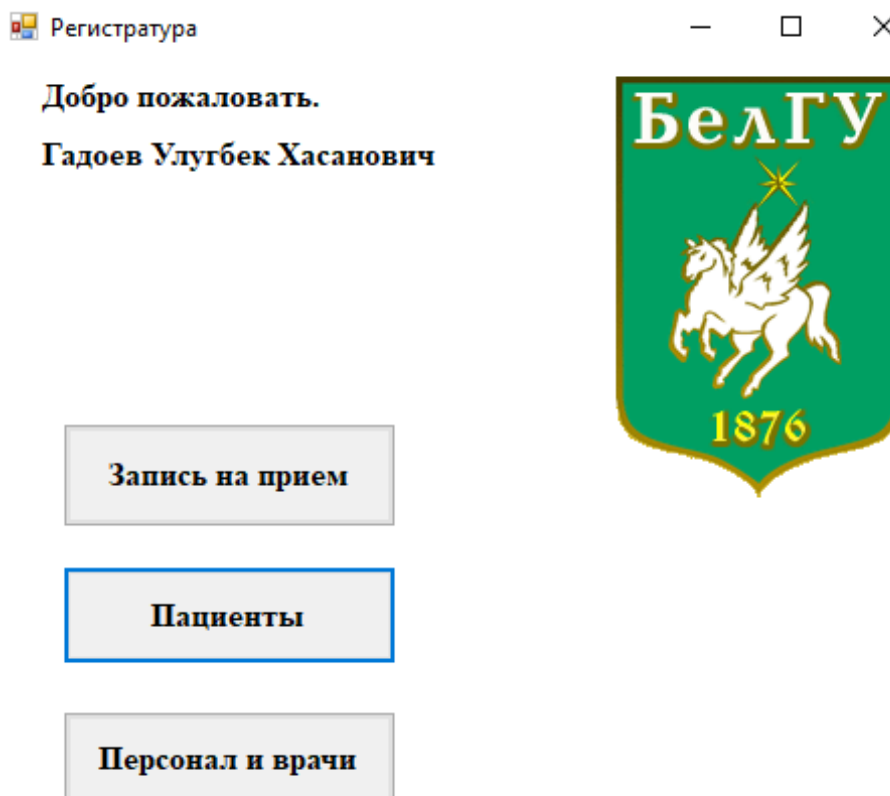


Рисунок 3.2 – Главная форма для пользователя "Регистратор"

На рисунке 3.3 показана форма с таблицей сотрудников для администратора. В левой таблице сотрудники, в правой их режим работы. Нужно заметить, что полный доступ к изменению информации есть только у администратора, у регистратора доступ немного ограничен (не все кнопки активны): не доступны изменения в полях таблицы «Логин» и «Пароль», нельзя добавить и удалить сотрудника.

№ Сотрудников	ФИО Сотрудника	Роль	Специализация	Кабинет	Логин	Пароль	Дни недели	С	До
1	Гадов Улугбе...	Администратор	Системный админис...	223	admin	111	Понедельник - Пятница	09:00:00	18:00:00
6	Бобров Владис...	Регистратор	Сотрудник регистрат	113	bobrov	101	Понедельник - Пятница	09:00:00	18:00:00
2	Плевако Арте...	Врач	Отоларинголог	112	plevako	112	Понедельник - Пятница	09:00:00	18:00:00
10	Нуралиев Али...	Врач	Педиатор	110	nur	02	Понедельник - Пятница	09:00:00	18:00:00
3	Данилова Инн...	Врач	Психолог	114	daniлова	222	Понедельник - Пятница	11:30:00	16:00:00
9	Героев Иван И...	Врач	Психолог	109	ger	01	Понедельник - Пятница	20:00:00	07:00:00
8	Сергеев Никол...	Врач	Физиотерапия	101	nik	000	Понедельник - Пятница	18:00:00	09:00:00
5	Игнатенко Ива...	Врач	Терапевт	222	ignat	444	Понедельник - Пятница	09:00:00	20:00:00
4	Сидоров Игор...	Врач	Хирург	119	sidor	111	Понедельник - Пятница	12:00:00	18:00:00

Назад    Добавить    Изменить ФИО    Изменить логин и    Изменить специализац    Изменить режим работы

Удалить

№ Сотрудников	ФИО Сотрудника	Роль	Специализация	Кабинет	Логин	Пароль	Дни недели	С	До
1	Гадов Улугбе...	Администратор	Системный админис...	223	admin	111	Понедельник - Пятница	18:00:00	09:00:00
6	Бобров Владис...	Регистратор	Сотрудник регистрат...	113	bobrov	101			
2	Плевако Артем...	Врач	Отоларинголог	112	plevako	112			
10	Нуралиев Али...	Врач	Педиатор	110	nur	02			
3	Данилова Инна...	Врач	Психолог	114	daniлова	222			
9	Героев Иван И...	Врач	Психолог	109	ger	01			
8	Сергеев Никола...	Врач	Физиотерапия	101	nik	000			
5	Игнатенко Ива...	Врач	Терапевт	222	ignat	444			
4	Сидоров Игорь...	Врач	Хирург	119	sidor	111			

Назад    Добавить    Изменить ФИО    Изменить логин и    Изменить специализац    Изменить режим работы

Удалить

Рисунок 3.3 – Изменение информации о сотрудниках в режиме администратора

Вторая кнопка на главной форме «Пациенты» открывает таблицу всех пациентов с информацией о них, из этой формы можно также открыть их историю болезни, направить пациента на прием. Эта форма представлена на рисунке 3.4. Здесь также можно искать, добавлять, удалять, изменять информацию о пациентах.

Если пациент пришел записаться на прием, то в первую очередь регистратор задаст вопрос, Вы были у нас уже на приеме? Или в первый раз?

Пациенты

Поиск пациента: ФИО, по Серии и № Паспорта

Поиск:

Найти

Пациенты    Записать на прием    История болезни    Новый пациент

ФИО	Пол	Дата рождения	СНИЛС	ИНН	Полис ОМС	Серия и № паспорта	Адрес	Контактная информация	Соц. статус	Подробнее
Зуев Иван В...	муж	19.11.1992 ...	123-442-333-	231-442-334-	4421-2321-4...	AA3811922	ул.Попова ...	+78541225588	Студент	Уч. в БелГ...
Миронов Вл...	муж	11.11.1990 ...	333-222-111-09	334-334-123-	4276-0700-11...	AA3451111	ул.Студенч...	+79633857414	Рабочий	Слесарь, Фа...
Нгамби Са...	муж	07.12.1993 ...	199-589-789-	145-852-963-	4785-5210-8...	AA1478520	ул.Побед...	+79885558888	Студент	Уч. в БелГ...
Нуралыева ...	жен	04.12.1994 ...	444-555-222-11	123-123-123-	4276-0700-1...	AA3334511	ул.Костюко...	+78454567687	Студенка	Уч. в Мед. ...
Рагачурин Т...	муж	19.02.1995 ...	474-585-896-	114-558-996-	0147-8555-5...	AB1478552	ул. Студен...	+79689141825	Студент	НИУ БелГ...
Самойлов В...	муж	14.05.1993 ...	147-989-569-	147-855-896-	1470-8520-9...	AA2589636	ул.Студенч...	+79563257745	Студент	НИУ БелГ...
Сергеев Па...	муж	09.01.1995 ...	555-555-111-00	123-456-234-	4276-0700-11...	AC1234455	ул.Студенч...	+79955588555	Студент	Уч. в БелГ...

Назад

Рисунок 3.4 – Таблица «Пациента»

Если пациент пришел повторно записаться на прием, то регистратор ищет у себя в таблице пациентов (рисунок 3.4), по «ФИО» или же по «Серии и № Паспорта» (рисунок 3.5)

Пациенты

Поиск пациента: ФИО, по Серии и № Паспорта

Поиск:

Найти

Пациенты    Записать на прием    История болезни    Новый пациент

ФИО	Пол	Дата рождения	СНИЛС	ИНН	Полис ОМС	Серия и № паспорта	Адрес	Контактная информация	Соц. статус	Подробнее
Сергеев Павел ...	муж	09.01.1995 ...	555-555-111-00	123-456-234-	4276-0700-11...	AC1234455	ул.Студенч...	+79955588555	Студент	Уч. в БелГ...

Назад

Рисунок 3.5 – Поиск пациента

Если регистратор нашел пациента у себя в базе, он может оформить запись на прием к врачу, нажав на кнопку «Запись на прием». Система автоматически уже заполнит поля: «ФИО Пациента», «Дата рождения» и «Контактная информация» (рисунок 3.6) и регистратор должен выбрать специализацию и ФИО врача, который может принять по графику работы.

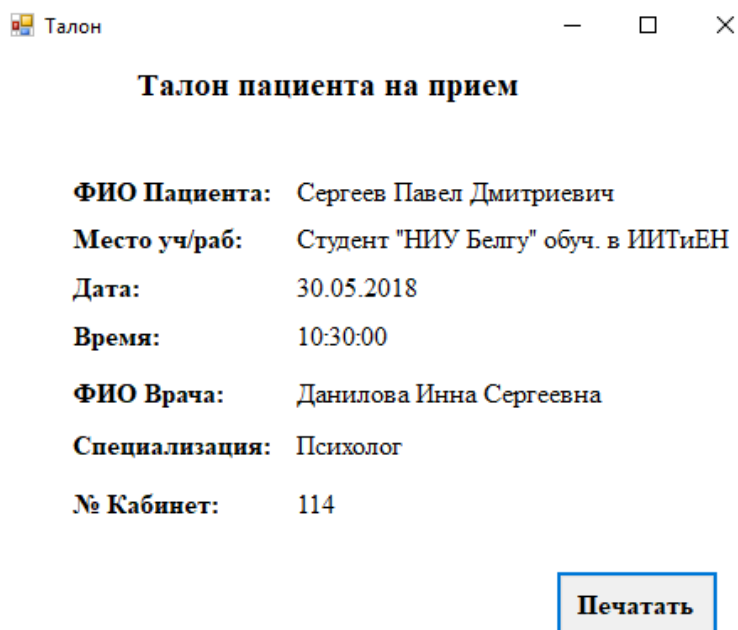
Также пациент может записаться к определенному врачу на прием. Форма таблицы запись на прием приведен на рисунке 3.6

Рисунок 3.6 – Таблица «Запись на прием» к определенному врачу

Как видно из рисунка 3.6, выбранной врач в отпуске с 25.05.2018 по 15.06.2018 гг. Значит регистратору придется направить пациента к врачу, у которого рабочие дни свободные (рисунок 3.7).

Рисунок 3.7 – Таблица «Запись на прием»

После записи пациента на прием, система автоматически формирует талон для пациента на прием к врачу (рисунок 3.8).



<b>ФИО Пациента:</b>	Сергеев Павел Дмитриевич
<b>Место уч/раб:</b>	Студент "НИУ Белгу" обуч. в ИИТиЕН
<b>Дата:</b>	30.05.2018
<b>Время:</b>	10:30:00
<b>ФИО Врача:</b>	Данилова Инна Сергеевна
<b>Специализация:</b>	Психолог
<b>№ Кабинет:</b>	114

[Печатать](#)

Рисунок 3.8 – Талон пациента на прием

После формирования талона на прием, пациент может прийти на прием по указанной дате и времени. В талоне поле «Место уч/раб» указывает, что сотрудники и обучающийся в университете «НИУ БелГУ» проходят осмотр и сдают анализы по скидке или же не оплачивая.

Для врача поликлиники на главной форме исчезает возможность правки информации о сотрудниках и появляется возможность просмотреть список пациентов, записанных к нему на прием, а также просмотреть и вести историю болезни пациентов (кнопка «Прием») (рисунок 3.9).

При нажатии на кнопку «Прием» выведется форма для ведения электронной медицинской карты, записанных на прием пациентов (рисунок 3.10). Нужно отметить, что каждый врач в историю болезни может заносить записи только от своего лица. Но он может посмотреть историю болезни пациента с записями всех врачей (т.е. с симптомами, диагнозами, выписанным лечением).

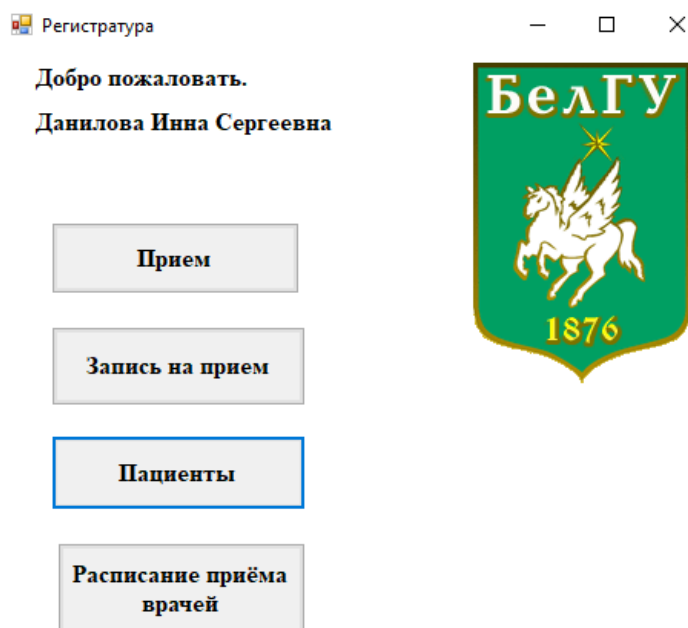


Рисунок 3.9 – Главная форма для пользователя «Врач»

На форме приема врач может выбрать дату (изначально автоматически устанавливается текущая дата), после этого в таблице он увидит, какие пациенты и на какое время записаны к нему в этот день. Далее выбрав одного из пациентов в таблице, врач может начать прием (т.е. в поля справа вводить нужную ему информацию). Нажав на кнопку «Сохранить», он внесет всю введенную информацию в электронную медицинскую карту пациента.

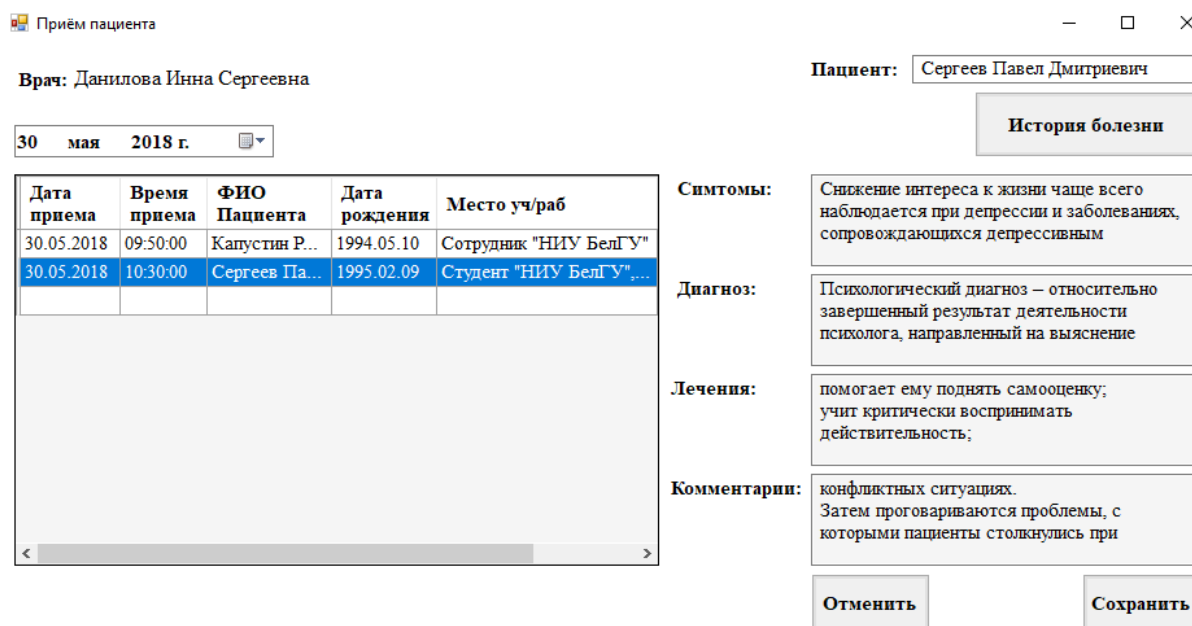


Рисунок 3.10 – Форма приема пациента

В этой же форме врач может открыть историю болезни этого пациента, просмотреть записи других врачей. Форма истории болезни приведена на рисунке 3.11.

**История болезни пациента**

Пациент: Сергеев Павел Дмитриевич

Дата приема	Врач
28.04.2018	Терапевт
01.05.2018	Невролога

**Симптомы:** Повышенная температура тела; Снижение иммунитета (частые ОРЗ, ОРВИ); Нарушения работы желудочно-кишечного тракта (систематическая тошнота, рвота, нарушения стула);

**Диагноз:** Выявленные признаки (симптомы) заболевания терапевт объединяет в синдромы (совокупность симптомов, имеющих единый патогенез), на основании которых делает заключение о предполагаемом заболевании.

**Лечения:** Перейти на здоровое питание, уменьшить содержание соли в рационе, соблюдать питьевой режим (не менее 4-х стаканов чистой воды в день), неизнурительные физические нагрузки (спорт, танцы, активный отдых), ежедневные прогулки на свежем воздухе, полноценный сон, причем ложиться спать следует не позже 22 часов.

**Комментарии:** Нужно изменить образ жизни.

Назад

Рисунок 3.11 – История болезни пациента

В таблице слева показаны все приемы пациента у врачей, нажав по одному из них справа врач увидит записи врача.

Врач также может записать на прием пациента, нажав на кнопку «Запись на прием», также врачу доступен просмотр графика работ врачей поликлиники нажав на кнопку «Расписание приема врачей», чтобы направить пациента к нужному врачу (рисунок 3.12).

Оставшаяся кнопка на главной форме «Пациенты», открывает таблицу всех пациентов с информацией о них, из этой формы можно также открыть их историю о болезни, направит существующего пациента на прием. Эта форма представлена на рисунке 3.4.

Также следует отметить, что пациент также сможет сам записаться на прием к нужному врачу на удобное время, используя «Электронную регистратуру» по Белгородской области, представленную на рисунке 1.8, через систему Госуслуги. Все сведения, внесенные на сайте, будут отображаться в разработанной АИС «Регистратура поликлиники НИУ БелГУ», которая сейчас



проходит тестирование и в ближайшее время будет подключена к системе «Электронная регистратура» по Белгородской области.

Расписание приема врачей: — □ ×

### Расписание приема врачей

Выберите специализацию врача: Хирург ▾

Выберите врача: Сидоров Игорь Сергеевич ▾

Расписание приема:

◀ Май 2018 ▶

Пн	Вт	Ср	Чт	Пт	Сб	Вс
30	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

☐ Сегодня: 28.05.2018

▶	09:50 .свободно
	10:20 .свободно
	10:30 .свободно
	10:40 .свободно
	10:50 .свободно
	11:20 .свободно

Рисунок 3.12 – Таблица «Расписание приема врачей»

Таким образом, была осуществлена программная реализация автоматизированной информационной системы регистратуры поликлиники «НИУ БелГУ», а также был проверен функционал, который система должна выполнять. Функции соответствуют техническому заданию, включены стандартные операции с данными (добавление, изменение, удаление).

Проанализировав выше представленные результаты, можно убедиться, что данная программа работает правильно и готова к дальнейшему использованию.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы была достигнута поставленная цель автоматизации работы регистратуры поликлиники «НИУ БелГУ» для упрощения записи пациентов к врачам, улучшения качества обслуживания пациентов и сокращения времени ожидания приема. Все поставленные для решения цели задачи, были выполнены:

- 1) проведен обзор существующих МИС регистратуры поликлиники;
- 2) разработано техническое задание;
- 3) разработаны модели для проектирования;
- 4) программно реализованы разработанные модели;
- 5) проведено тестирование разработанной МИС.

Проведенный обзор существующих медицинских информационных систем регистратуры поликлиники позволил выделить наиболее удобную с точки зрения сотрудников и пациентов медицинского учреждения структуру приложения. Разработка технического задания определила основные функции, которая должна выполнять система. Программная реализация разработанных проектных моделей позволила получить медицинскую информационную систему регистратуры поликлиники НИУ БелГУ, значительно упрощающую работу сотрудников поликлиники НИУ БелГУ, повышающую удобство записи пациентов на прием, а также позволяющей врачу получить более полную картину болезни пациента за счет возможности просмотра записей других врачей.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Электронная регистратура [Электронный ресурс] / Электрон. текст. данные. – 2016. – Режим доступа: <http://docplayer.ru/38569900-Elektronnaya-registratura.html>. – Дата доступа: 25.04.2018.
2. Гусев, А.В. Обзор решений «Электронная регистратура» [Электронный ресурс] / А.В. Гусев. – Электрон. журнал. – Врач и информационные технологии. — 2010. – Режим доступа: <https://cyberleninka.ru/article/v/obzor-resheniy-elektronnaya-registratura-1>. – Дата доступа: 25.04.2018.
3. Электронная регистратура Белгородской области [Электронный ресурс] / Сервис записи на прием к врачу. – Режим доступа: <https://2dr.ru/>. – Дата доступа: 21.05.2018.
4. Белоусов, Н. Порядок записи к врачам [Электронный ресурс] / Н. Белоусов. – Электрон. текст. данные. – Департамент здравоохранения и социальной защиты населения белгородской области. – 7.09.2012. – Режим доступа: <http://gdp4bel.belzdrav.ru/employees/poryadok-zapisi-k-vracham.php>. – Дата доступа: 10.03.2018.
5. Проект «Бережливая поликлиника» [Электронный ресурс] Электрон. текст. данные. – 2018. – Режим доступа: <http://www.rosatom.ru/social-respons/proekt-berezhlivaya-poliklinika/>. – Дата доступа: 25.04.2018.
6. Репин, В.В. Бизнес-процессы: построение, анализ, регламентация. / В.В. Репин. – М.: РИА «Стандарты и качество», 2007. – 240 с.
7. Голицына, О.Л. СУБД / О.Л. Голицына, В. Максимов, И.И. Попов. – М.: Форум, 2004. – 352 с.
8. Кузнецов, С.Д. Базы данных. Модели и языки. / С.Д. Кузнецов. – М.: Бином пресс, 2008. – 284 с.
9. Маклаков, С.В. Создание информационных систем с AllFusionModeling Suite / С.В. Маклаков. – СПб: Диалог-МИФИ, 2007. – 432 с.

10. Рябова, Ю.С. Моделирование бизнес-процессов производственной компании / Ю.С. Рябова, С. П. Пирогов, – 2007. – 56 с.
11. Джестон, Дж. Управление бизнес-процессами / Дж. Джестон, Й. Нелис, – 2008. – 512 с.
12. Маклаков, С.В. BPwin и ERwin: CASE-средства для разработки информационных систем / С.В. Маклаков. – СПб: Диалог-МИФИ, 2000. – 43 с.
13. Моисеенко, С. SQL: задачи и решения / С. Моисеенко. – СПб: Питер, – 2006. – 256 с.
14. Грекул, В. И. Проектирование информационных систем / В. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. – 2016. – 516 с.
15. Коваленко, В.В. Проектирование информационных систем / В.В. Коваленко. – СПб: Форум, – 2015. – 320 с.
16. Губина Е. А. Проектирование информационной системы на основе связывания CASE-инструментария и реляционной базы данных / Е.А. Губина, Г.Х. Ирзаев Г, М.Г. Адеева. – 2014. – 71 с.
17. Дубейковский В.И. Эффективное моделирование с Allfusion Process Modeler 4.1.4 и Allfusion PM / В.И. Дубейковский. – СПб.: ДИАЛОГ – МИФИ, – 2007. – 384 с.
18. Саймон, Р. Администрирование PostgreSQL / Р. Саймон, Х. Кросинг. – 2013. – 368 с.
19. Галисеев Г.В. Программирование на языке C# / Г.В. Галисеев. – М.: Вильямс, – 2006. – 368 с.
20. Михелёв В.М. Базы данных и СУБД / В.М. Михелёв. – Белгород.: БелГУ, – 2007. – 200 с.
21. Муромцев В.В. Проектирование информационных систем / В.В. Муромцев. – Белгород.: БелГУ, – 2007. – 160 с.

## Приложение А

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Npgsql;
namespace regpolic
{
    public partial class Авторизация : Form
    {
        public Form frm1, frm2, frm3, frm4;
        public static string person;
        public Авторизация()
        {
            InitializeComponent();
        }
        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                string conn = "Server=127.0.0.1;Port=5432;User
Id=postgres;Password=gux19021995;Database=RegHospital";
                NpgsqlConnection connect = new NpgsqlConnection(conn);
                connect.Open();
                NpgsqlCommand command = new NpgsqlCommand("select fio from personal where login = '" +
textBox1.Text + "' and parol = '" + textBox2.Text + "'", connect);
                NpgsqlDataReader dr = command.ExecuteReader();
                while (dr.Read())
                {
                    person = dr[0].ToString();
                }
                connect.Close();
                frm2 = new Form2();
                frm2.Show();
                this.Hide();
            }
            catch (Exception msg)
            {
                MessageBox.Show(msg.ToString());
            }
        }
    }
}

namespace regpolic
{
    public partial class Form2 : Form
    {
        public Form2()
        {
            InitializeComponent();
            label1.Text = Авторизация.person;
        }
        private void label1_Click(object sender, EventArgs e)
        {
        }
    }
}
```

```

{
}
//запись на прием
private void button3_Click(object sender, EventArgs e)
{
    Запись_на_прием form2 = new Запись_на_прием();
    form2.Show();
    this.Hide();
}
//пациенты
private void button1_Click(object sender, EventArgs e)
{
    Пациенты form2 = new Пациенты();
    form2.Show();
    this.Hide();
}
//сотрудники
private void button2_Click(object sender, EventArgs e)
{
    Сотрудники form2 = new Сотрудники();
    form2.Show();
    this.Hide();
}

private void Form2_Load(object sender, EventArgs e)
{
}

private void button2_Click_1(object sender, EventArgs e)
{
    Form4 form2 = new Form4();
    form2.Show();
    this.Hide();
}

private void button4_Click(object sender, EventArgs e)
{
    Сотрудники form2 = new Сотрудники();
    form2.Show();
    this.Hide();
}
}
namespace regpolic
{
    public partial class Сотрудники : Form
    {
        public Сотрудники()
        {
            InitializeComponent();
            dataGridView1.Rows.Clear();
            try
            {
                string conn = "Server=127.0.0.1;Port=5432;User
Id=postgres;Password=gux19021995;Database=RegHospital";
                NpgsqlConnection connect = new NpgsqlConnection(conn);
                connect.Open();
            }
            catch { }
        }
    }
}

```

```

        NpgsqlCommand command1 = new NpgsqlCommand("select
id_personal, fio, roles, specialization, id_office, login, parol from personal, role, specialization where personal.id_role =
role.id_role and personal.id_specialization = specialization.id_specialization", connect);
        NpgsqlDataReader reader;
        reader = command1.ExecuteReader();
        while (reader.Read())
        {
            string[] s = new string[7];
            for (int i = 0; i < reader.FieldCount; i++)
            {
                s[i] = reader[i].ToString();
            }
            dataGridView1.Rows.Add(s);
        }
        reader.Close();
        command1.Dispose();
        NpgsqlCommand command = new NpgsqlCommand("select days_week, time_from, time_to from
timeworks, personal where personal.id_personal=timeworks.id_personal", connect);
        NpgsqlDataReader reader1;
        reader1 = command.ExecuteReader();
        while (reader1.Read())
        {
            string[] s = new string[3];
            for (int i = 0; i < reader1.FieldCount; i++)
            {
                s[i] = reader1[i].ToString();
            }
            dataGridView2.Rows.Add(s);
        }
        reader1.Close();
    }
    catch (Exception msg)
    {
        MessageBox.Show(msg.ToString());
    }
}
private void button1_Click(object sender, EventArgs e)
{
    Form2 frm1 = new Form2();
    frm1.Show();
    this.Hide();
}

private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    dataGridView2.Rows.Clear();
    try
    {
        Int32 rowcount = dataGridView1.SelectedCells[0].RowIndex;
        String identifier = dataGridView1[0, rowcount].FormattedValue.ToString();

        string conn = "Server=127.0.0.1;Port=5432;User
Id=postgres;Password=gux19021995;Database=RegHospital";
        NpgsqlConnection connect = new NpgsqlConnection(conn);
        connect.Open();
        NpgsqlCommand command = new NpgsqlCommand("select days_week, time_from, time_to from
timeworks where timeworks.id_personal = " + identifier + ";", connect);
        NpgsqlDataReader reader1;
        reader1 = command.ExecuteReader();
        while (reader1.Read())

```

```

        {
            string[] s = new string[3];
            for (int i = 0; i < reader1.FieldCount; i++)
            {
                s[i] = reader1[i].ToString();
            }
            dataGridView2.Rows.Add(s);
        }
        reader1.Close();
        connect.Close();
    }
    catch (Exception ex) { MessageBox.Show("" + ex); }
}
private void dataGridView2_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}
}
}
public partial class Form4 : Form
{
    public int personal;
    public NpgsqlConnection connect;
    public Form4()
    {
        InitializeComponent();
    }

    private void monthCalendar1_DateChanged(object sender, DateRangeEventArgs e)
    {
        dataGridView1.Rows.Clear();
        dataGridView1.Rows.Add("09:50 ,свободно");
        dataGridView1.Rows.Add("10:20 ,свободно");
        dataGridView1.Rows.Add("10:30 ,свободно");
        dataGridView1.Rows.Add("10:40 ,свободно");
        dataGridView1.Rows.Add("10:50 ,свободно");
        dataGridView1.Rows.Add("11:20 ,свободно");
        dataGridView1.Rows.Add("11:30 ,свободно");
        dataGridView1.Rows.Add("11:40 ,свободно");
        dataGridView1.Rows.Add("11:50 ,свободно");
    }
    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        string selectedSpec = comboBox1.SelectedItem.ToString();
        switch (selectedSpec)
        {
            case "Терапевт":
                personal = 3;
                break;
            case "Хирург":
                personal = 1;
                break;
            case "Стоматолог":
                personal = 2;
                break;
            case "Физиотерапия":
                personal = 6;
                break;
            case "Психолог":
                personal = 7;
                break;
            case "Педиатор":

```



```

        personal = 8;
        break;
    }
    comboBox1.Items.Clear();
    string conn = "Server=127.0.0.1;Port=5432;User
Id=postgres;Password=gux19021995;Database=RegHospital";
    connect = new NpgsqlConnection(conn);
    connect.Open();
    NpgsqlCommand command2 = new NpgsqlCommand("select fio from personal where
id_specialization = " + personal + "", connect);
    NpgsqlDataReader reader1;
    reader1 = command2.ExecuteReader();
    while (reader1.Read())
    {
        for (int i = 0; i < reader1.FieldCount; i++)
        {
            comboBox1.Items.Add(reader1[i].ToString());
        }
    }
    reader1.Close();
    command2.Dispose();
}
}
}
public partial class Form8 : Form
{
    public Form8()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        Form6 frm1 = new Form6();
        frm1.Show();
        this.Hide();
    }

    private void button2_Click(object sender, EventArgs e)
    {
        Пациенты form2 = new Пациенты();
        form2.Show();
        this.Hide();
    }

    private void button3_Click(object sender, EventArgs e)
    {
        Запись_на_прием form2 = new Запись_на_прием();
        form2.Show();
        this.Hide();
    }
    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            string conn = "Server=127.0.0.1;Port=5432;User
Id=postgres;Password=gux19021995;Database=RegHospital";
            NpgsqlConnection connect = new NpgsqlConnection(conn);
            connect.Open();

```

```

        NpgsqlCommand command = new NpgsqlCommand("insert into patient(fio, pol, date_of_birth,
snils, inn, polis_oms, series_pas_num, adres, contact_inform, soc_status, podrobnee) values(" +
textBox1.Text.ToString() + "," +
        textBox2.Text.ToString() + "," + textBox11.Text.ToString() + "," + textBox3.Text.ToString()
+
        "," + textBox4.Text.ToString() + "," + textBox5.Text.ToString() + "," +
textBox6.Text.ToString() +
        "," + textBox7.Text.ToString() + "," + textBox8.Text.ToString() + "," +
textBox9.Text.ToString() +
        "," + textBox10.Text.ToString() + ")", connect);
        command.ExecuteNonQuery();
        connect.Close();
    }
    catch (Exception msg)
    {
        MessageBox.Show(msg.ToString());
    }
    Пациенты form2 = new Пациенты();
    form2.Show();
    this.Hide();

}

private void button2_Click(object sender, EventArgs e)
{
    Пациенты frm1 = new Пациенты();
    frm1.Show();
    this.Hide();
}

private void textBox11_TextChanged(object sender, EventArgs e)
{
}
}
}
namespace regpolic
{
    public partial class Запись_на_прием : Form
    {
        public int personal;
        public NpgsqlConnection connect;
        public Запись_на_прием()
        {
            InitializeComponent();
            textBox2.Text = Пациенты.fio;
            dateTimePicker1.Text = Пациенты.dob;
            textBox1.Text = Пациенты.contact;

            string conn = "Server=127.0.0.1;Port=5432;User
Id=postgres;Password=gux19021995;Database=RegHospital";
            connect = new NpgsqlConnection(conn);
            connect.Open();
            NpgsqlCommand command1 = new NpgsqlCommand("select specialization from specialization",
connect);
            NpgsqlDataReader reader;
            reader = command1.ExecuteReader();
            while (reader.Read())
            {
                string[] s = new string[8];
                for (int i = 0; i < reader.FieldCount; i++)
                {

```

```

        comboBox1.Items.Add(reader[i].ToString());
    }

}

reader.Close();
command1.Dispose();

}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    string selectedSpec = comboBox1.SelectedItem.ToString();
    switch (selectedSpec)
    {
        case "Терапевт" :
            personal = 3;
            break;
        case "Хирург":
            personal = 1;
            break;
        case "Стоматолог":
            personal = 2;
            break;
        case "Физиотерапия":
            personal = 6;
            break;
        case "Психолог":
            personal = 7;
            break;
        case "Педиатр":
            personal = 8;
            break;
    }
    comboBox3.Items.Clear();

    NpgsqlCommand command2 = new NpgsqlCommand("select fio from personal where
id_specialization = " + personal + "", connect);
    NpgsqlDataReader reader1;
    reader1 = command2.ExecuteReader();
    while (reader1.Read())
    {
        for (int i = 0; i < reader1.FieldCount; i++)
        {
            comboBox3.Items.Add(reader1[i].ToString());
        }
    }
    reader1.Close();
    command2.Dispose();
}

namespace regpolic
{
    public partial class Пациенты : Form
    {
        public static string fio, dob, contact;
        public Пациенты ()
        {
            InitializeComponent();

            dataGridView1.Rows.Clear();
            string conn = "Server=127.0.0.1;Port=5432;User
Id=postgres;Password=gux19021995;Database=RegHospital";

```

```

NpgsqlConnection connect = new NpgsqlConnection(conn);
connect.Open();
NpgsqlCommand command1 = new NpgsqlCommand("select * from patient", connect);
NpgsqlDataReader reader;
reader = command1.ExecuteReader();
while (reader.Read())
{
    string[] s = new string[14];
    for (int i = 0; i < reader.FieldCount; i++)
    {
        s[i] = reader[i].ToString();
    }
    dataGridView1.Rows.Add(s);
}
reader.Close();
command1.Dispose();
}
private void button3_Click(object sender, EventArgs e)
{
    int num = dataGridView1.CurrentCell.RowIndex;
    Запись_на_прием form2 = new Запись_на_прием();
    form2.Show();
    this.Hide();
}

private void button1_Click(object sender, EventArgs e)
{
    Добавления_нового_пациента form2 =new Добавления_нового_пациента();
    form2.Show();
    this.Hide();
}

private void button4_Click(object sender, EventArgs e)
{
    Form2 frm1 = new Form2();
    frm1.Show();
    this.Hide();
}

private void button2_Click(object sender, EventArgs e)
{
    Id=postgres;Password=gux19021995;Database=RegHospital";
    NpgsqlConnection connect = new NpgsqlConnection(conn);
    connect.Open();
    NpgsqlCommand command = new NpgsqlCommand("select * from patient where (fio LIKE '%" +
textBox1.Text + "%' or series_pas_num LIKE '%" + textBox1.Text + "%')", connect);
    NpgsqlDataReader reader;
    reader = command.ExecuteReader();
    while (reader.Read())
    {
        string[] s = new string[12];
        for (int i = 0; i < reader.FieldCount; i++)
        {
            s[i] = reader[i].ToString();
        }
        dataGridView1.Rows.Add(s);
    }
}

private void button5_Click(object sender, EventArgs e)

```

```

        {
            dataGridView1.Rows.Clear();
            string conn = "Server=127.0.0.1;Port=5432;User
Id=postgres;Password=gux19021995;Database=RegHospital";
            NpgsqlConnection connect = new NpgsqlConnection(conn);
            connect.Open();
            NpgsqlCommand command1 = new NpgsqlCommand("select * from patient", connect);
            NpgsqlDataReader reader;
            reader = command1.ExecuteReader();
            while (reader.Read())
            {
                string[] s = new string[12];
                for (int i = 0; i < reader.FieldCount; i++)
                {
                    s[i] = reader[i].ToString();
                }
                dataGridView1.Rows.Add(s);
            }
            reader.Close();
            command1.Dispose();
        }
        private void button6_Click(object sender, EventArgs e)
        {
            Form7 form2 = new Form7();
            form2.Show();
            this.Hide();
        }

int selectedrowindex = dataGridView1.SelectedCells[0].RowIndex;

        DataGridViewRow selectedRow = dataGridView1.Rows[selectedrowindex];

        fio = Convert.ToString(selectedRow.Cells[1].Value);
        dob = Convert.ToString(selectedRow.Cells[3].Value);
        contact = Convert.ToString(selectedRow.Cells[9].Value);

    }
}
}
}

```



Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« \_\_\_\_ » \_\_\_\_\_ Г.

\_\_\_\_\_

\_\_\_\_\_

