|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_ ***ИУК «Информатика и управление»\_\_\_\_\_\_\_\_\_***

**КАФЕДРА** \_\_ ***ИУК5 «Системы обработки информации»***

**ОТЧЕТ**

Студент гр. ИУК5-51Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Губарев П.П.)

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (Кириллов В.Ю.)

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

6-10 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

6-10 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2021

**1. Общие сведения.**

**1.1 Полное наименование системы и ее условное обозначение.**

Разработка прикладного программного обеспечения для записи и последующего обслуживания пациентов учреждения здравоохранения (поликлиники).

**1.2 Шифр темы или шифр (номер) договора.**

Шифр темы и договора отсутствует.

**1.3 Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты.**

Высшее учебное заведение КФ МГТУ имени Н. Э. Баумана

**1.4** **Перечень документов, на основании которых создается система, кем и когда утверждены эти документы.**

На основании задания курсовой работы по учебному плану.

**1.5 Плановые сроки начала и окончания работы по созданию системы.**

Начало работы: 01.09.2021

Окончание работы: 10.12.2021

**1.6 Сведения об источниках и порядке финансирования работ.**

Проект осуществляется без финансирования.

**2. Назначение и цели создания (развития) системы.**

**2.1 Назначение системы.**

Прикладное программное обеспечение необходимо для создания записей пациентов на прием к специалистам и последующего приема по заданному расписанию.

**2.2 Цели создания системы.**

Целью курсовой работы является формирование практических навыков по разработке и реализации программного приложения с использованием баз данных.

**3. Характеристика объектов автоматизации.**

**3.1 Краткие сведения об объекте автоматизации или ссылки на документы, содержащие такую информацию.**

Объектом автоматизации приложения является расписание специалистов; правила направления пациентов к специалистам.

**4. Требования к системе.**

**4.1 Требование к структуре и функционированию системы.**

Функциональная структура системы должна включать основные методы для регистрации пользователей; записи пациентов на прием в сетку расписания в базе данных; ввод/импорт расписания специалистов; добавление специальностей или специалистов; ввод карточек пациентов; запись пациента на прием; отмена приема; перенос приема; аутентификация пользователя.

**4.2 Требования к защите информации от несанкционированного доступа.**

Требования к защите информации от несанкционированного доступа, на текущем этапе работы отсутствуют. Защита информации от несанкционированного доступа к данным осуществляется с помощью аутентификации.

**4.3 Требования к составу и параметрам технических средств.**

Для использования программы необходимы средства ввода и вывода информации (мышь, Touch Screen или клавиатура, монитор).

**5. Состав и содержание работ по создание системы.**

*Таблица 1.*

|  |  |  |
| --- | --- | --- |
| **Стадии** | **Этапы работ** | **Сроки исполнения** |
| 1. Исследовательская часть | 1.1 Исследования предметной области.  1.2 Выбор средств разработки.  1.3 Рассмотрение аналогов программного продукта.  1.4 Оформление отчёта о выполненной работе. | 30.09.2021 |
| 2. Проектно-конструкторская часть | 2.1 Разработка структуры приложения.  2.2 Разработка архитектуры приложения.  2.3 Реализация рабочего прототипа приложения. | 30.10.2021 |
| 3. Проектно-технологическая часть | 3.1 Тестирование и отладка макета рабочей программы. | 30.11.2021 |
| 4. Защита курсовой работы | 4.1 Подготовка презентации и речи для защиты курсовой работы  4.2 Защита курсовой работы | 20.12.2021 |

**6. Порядок контроля и приемки системы.**

**6.1 Виды испытаний**

Для тестирования приложения будут выполнятся приемо-сдаточные испытания.

**6.2 Общие требования к приемке работ.**

При провидении приемки работ должна быть произведена проверка выполненного задания на соответствие требованиям текущего «Технического задания».

**7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие.**

Для работы приложения необходимо установить программный комплекс; выполнить первоначальное наполнение базы данных; подготовить аппаратный комплекс.

**8. Требование к документированию.**

По окончанию работы предъявлена расчетно-пояснительная записка в состав которой входят:

* техническое задание;
* научно-исследовательская часть;
* проектно-конструкторская часть;
* проектно-технологическая часть.

Также должна быть предоставлена графическая часть работы, выполненная формате А1 на 2 листах, в которую входят:

* структурные схемы;
* алгоритмические схемы.

**9. Источники разработки.**

1. Е.В. Вершинин, Р.Б. Бобров “Методические указания к выполнению курсовой работы” Калуга Кф МГТУ им. Баумана 2021.

2. ГОСТ 34.602-89.

3. ГОСТ 34.601-90.

**Научно-исследовательская часть**

1. **Постановка задачи проектирования.**

Задачей проектирования является формирование практических навыков по разработке и реализации программного обеспечения для записи и последующего обслуживания пациентов учреждения здравоохранения (поликлиники) с использованием баз данных.

Необходимо разработать программное обеспечение для создания записей пациентов на прием к специалистам и последующего приема по заданному расписанию.

1. **Описание предметной области.**

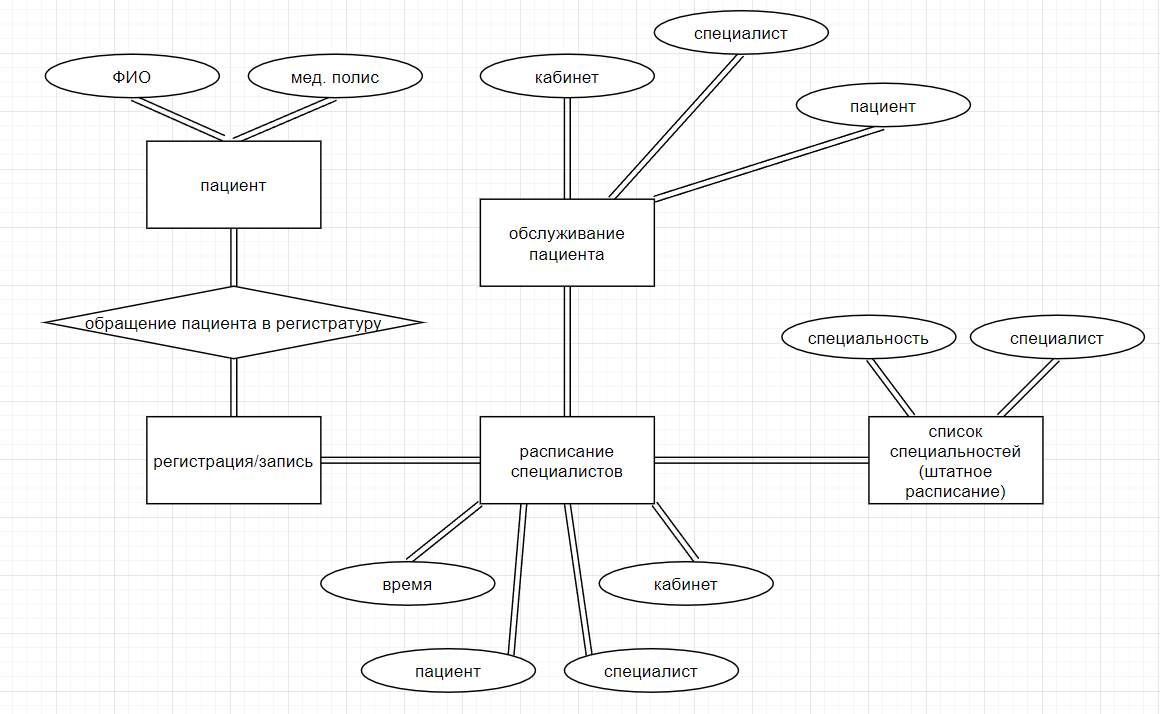
**Регистратура** является структурным подразделением поликлиники и врачебной амбулатории, обеспечивающим формирование и распределение потоков пациентов, своевременную запись и регистрацию больных на прием к врачу, в том числе с применением информационных технологий.

Рисунок 2.1 – Концептуальная схема

Основными задачами регистратуры медицинской организации являются:

1.Организация беспрепятственной и безотлагательной предварительной записи больных на прием к врачу в автоматизированном режиме.

2.Обеспечение регулирования интенсивности потока населения с целью создания равномерной нагрузки врачей и распределение его по видам оказываемой помощи.

3.Систематизированное хранение медицинской документации пациентов, обеспечение своевременного подбора и доставки медицинской документации в кабинеты врачей.

4.Информирование населения в доступной форме:

- о времени приема врачей всех специальностей, режиме работы лабораторий, кабинетов, дневного стационара и иных подразделений медицинской организации, в том числе субботу и воскресенье, с указанием часов приема, расположения и номеров кабинетов помещений;

- о правилах и порядке предоставления платных медицинских услуг;

- о правилах работы в системе Единой электронной регистратуры.

5. Направление в установленном порядке граждан, обратившихся в поликлинику, на профилактические обследования и осмотры.

6. Ведение первичной медицинской документации (заполнение паспортной части медицинских карт амбулаторных больных, талона амбулаторного пациента, журналов и т.д.).

Порядок работы регистратуры:

Прием плановых больных в поликлинике и амбулаторно-поликлинических подразделениях (врачебная амбулатория) осуществляется при предоставлении пациентом страхового медицинского полиса и документа, подтверждающего личность

Направление пациентов на прием к врачу осуществляется по талонной системе (предварительной записи, в том числе электронной).

В поликлинике предусматривается возможность развития информационных технологий для организации предварительной записи на прием к врачам по сети Интернет и через информационный киоск. Предварительная запись осуществляется в соответствии с Единым республиканским регламентом организации записи на прием к врачу. Перед приемом регистратор относит в кабинет подобранные медицинские карты и листы предварительной записи на прием, по окончании приема медицинская сестра кабинета возвращает амбулаторные карты в регистратуру. Пациент, предварительно записавшийся на приём к врачу, обращается непосредственно в кабинет врача к установленному времени, минуя регистратуру.

Медицинская карта амбулаторного больного должна содержать («Медицинская карта амбулаторного больного» форма 025/у-04, утвержденная приказом Министерства здравоохранения и социального развития Российской Федерации от 22.11.2004 г. № 255 «О порядке оказания первичной медико-санитарной помощи гражданам, имеющим право на получение набора социальных услуг»):

- паспортную часть;

- информированное согласие на медицинское вмешательство;

- информированное согласие пациента на обработку персональных данных;

- лист уточненных диагнозов;

- лист профилактических осмотров;

- записи амбулаторных посещений, патронажей;

- лечебные назначения;

- результаты дополнительных обследований и консультаций;

- эпикризы;

- выписки из истории болезни (копии) (в случае стационарного лечения);

-медицинская карта амбулаторного больного, имеющего право на получение набора социальных услуг, должна быть промаркирована литерой «Л».

В медицинской карте амбулаторного больного соответствующим образом должны быть оформлены:

- данные о прививках;

- сведения о выдаче листка нетрудоспособности;

- заключения ВК, МСЭ;

- данные о направлении на госпитализацию;

- данные о направлении на санаторно-курортное лечение;

- данные о выписке льготных рецептов;

- сведения о выдаче справок и выписок из амбулаторной карты и др.

Все записи лечащего врача заверяются его подписью.

Медицинские карты амбулаторных больных хранятся в регистратуре поликлиники или врачебной амбулатории. Работники регистратуры несут ответственность за сохранность медицинских карт амбулаторных больных, прикрепленных к лечебно-профилактическому учреждению.

Амбулаторные карты систематизируются в стеллажах по участкам, а внутри них — по улицам, домам и квартирам.

При выдаче медицинских карт амбулаторных больных на руки пациентам (с учетом соблюдения требований федерального законодательства по защите персональных данных; при наличии заявления на имя главного врача и письменного согласия пациента на передачу его персональных данных) или передаче в другие лечебно-профилактические учреждения, данный факт фиксируется в «Журнале регистрации выдачи «Медицинской карты амбулаторного больного» (форма 025/у-04) в другие лечебно-профилактические учреждения и на руки пациентам (Приложение 2).

В регистратуре располагается информация с указанием времени и места приема населения главным врачом и его заместителями, поэтажный указатель врачебных кабинетов; номера телефонов вызова врачей на дом, часы работы кабинета доврачебного приема и кабинета неотложной помощи (при их наличии).

Для хранения медицинской и общей документации в поликлинике создается медицинский архив. В медицинском архиве производятся прием, учет, классификация, хранение и выдача документальных материалов как персоналу поликлиники, так и больным (в случае изменения их места жительства или по запросам различных организаций в установленном законодательством порядке). Медицинские карты амбулаторных больных поступают в архив из регистратуры в том случае, если больной в течение 5 лет не обращался в поликлинику и хранятся в течение 25 лет

1. **Анализ аналогов и прототипов.**

**1)Госуслуги**

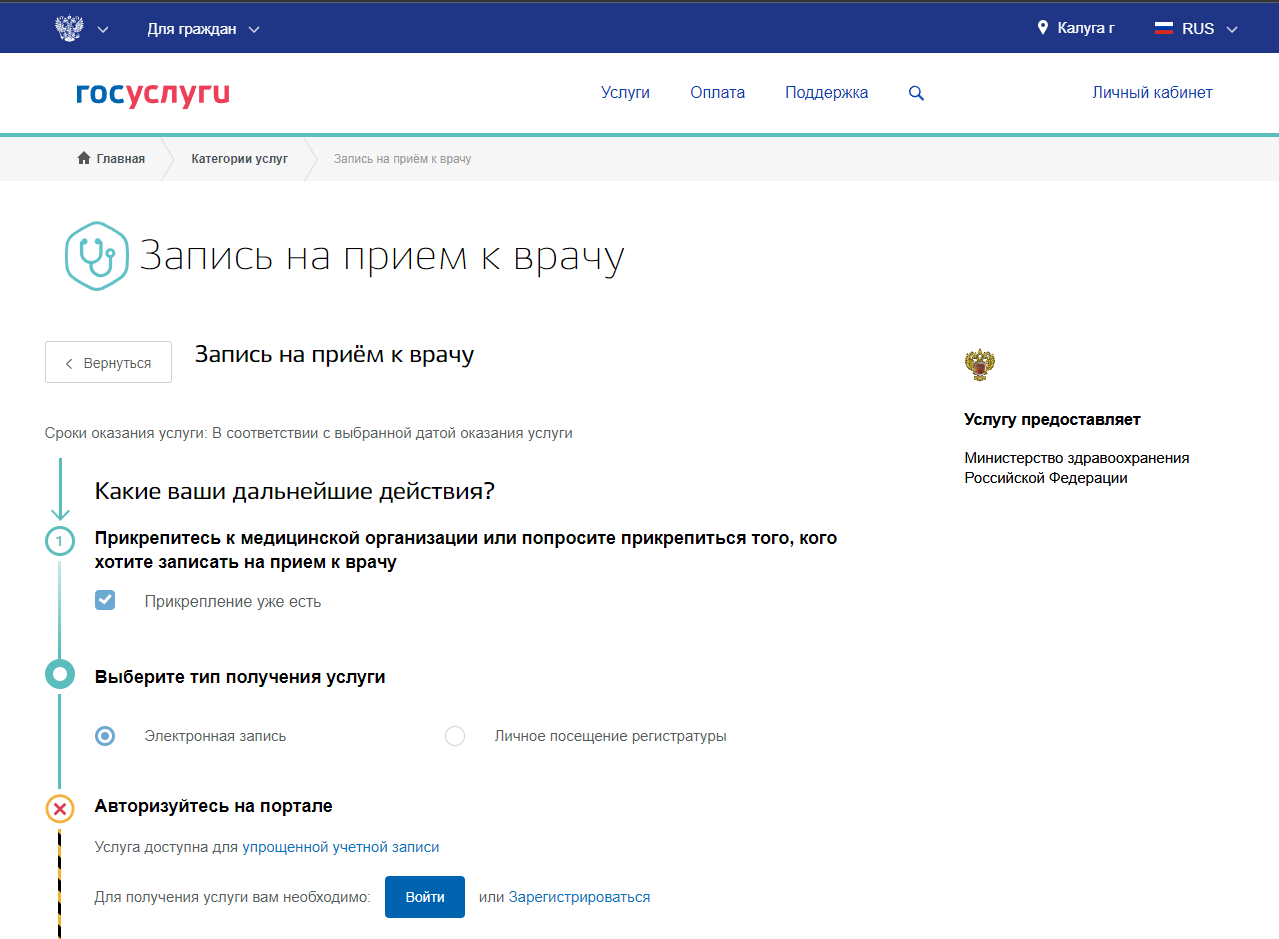
Услугу предоставляет [Министерство здравоохранения Российской Федерации](https://www.gosuslugi.ru/structure/10000001077).

Рисунок 2.2 - Госуслуги

Выберите поликлинику и запишитесь на прием к врачу, заполнив электронную форму записи. Для этого вам потребуется ваш полис обязательного медицинского страхования. Вы можете записать на прием другого человека – для этого достаточно указать в форме записи его ФИО, дату рождения, номер полиса ОМС и СНИЛС (при наличии). В случае необходимости вы можете перенести время приема или отменить запись.

**Получение медицинской услуги:**

Распечатайте информацию (при необходимости), подтверждающую запись на прием, и посетите медицинское учреждение в назначенное время. Распечатать подтверждение можно на экране с результатами записи, который появится после заполнения и отправки заявления на запись к врачу – просто нажмите на ссылку «Распечатать» в правой части страницы. Эта же страница будет доступна в вашем Личном кабинете на портале.

Преимуществом такой записи является то, что большинство людей уже зарегистрированы в портале госуслуг из-за его распространённости во многих сферах жизнедеятельности, связанных с государственными учреждениями. Но из-за частой высокой загруженности портала нередко возникают сбои в его работе, что является ощутимым недостатком.

**2)Регистратура40.рф**

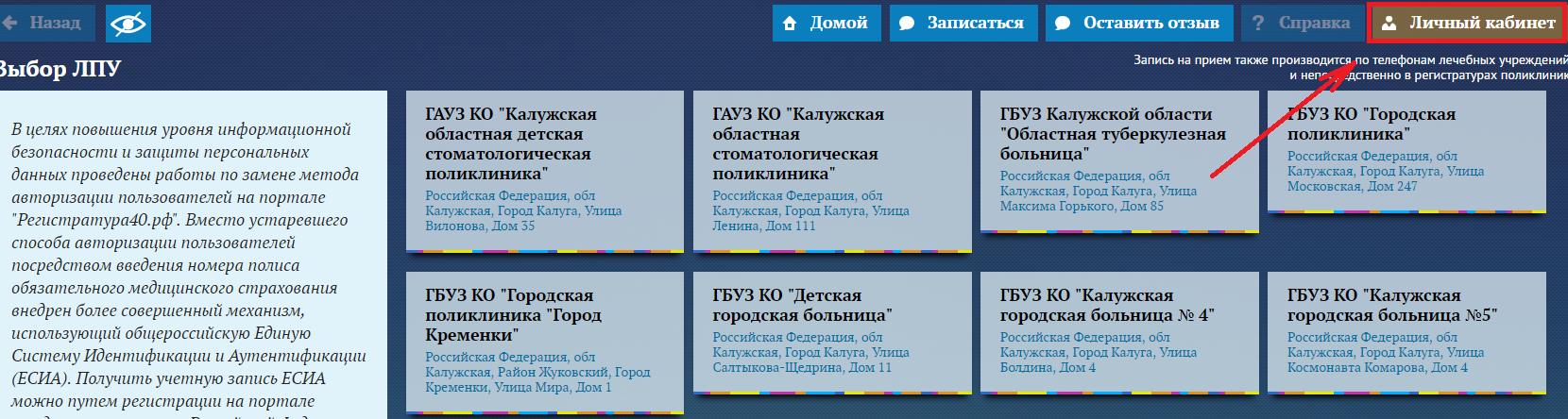
На главной странице портала пациента необходимо пройти авторизацию через «Личный кабинет» (Рисунок 2.3):

Рисунок 2.3 - В открывшемся окне нажмите кнопку «Войти через ЕСИА»

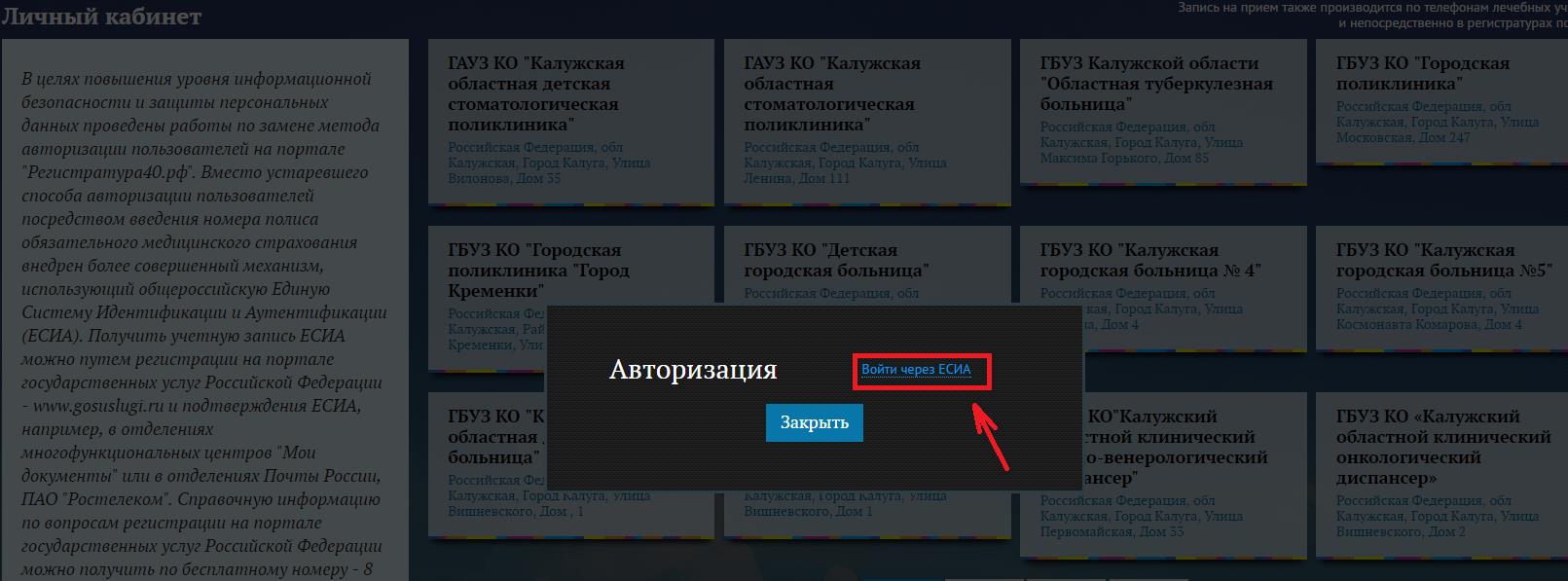


Рисунок 2.4 – Оно для авторизации.

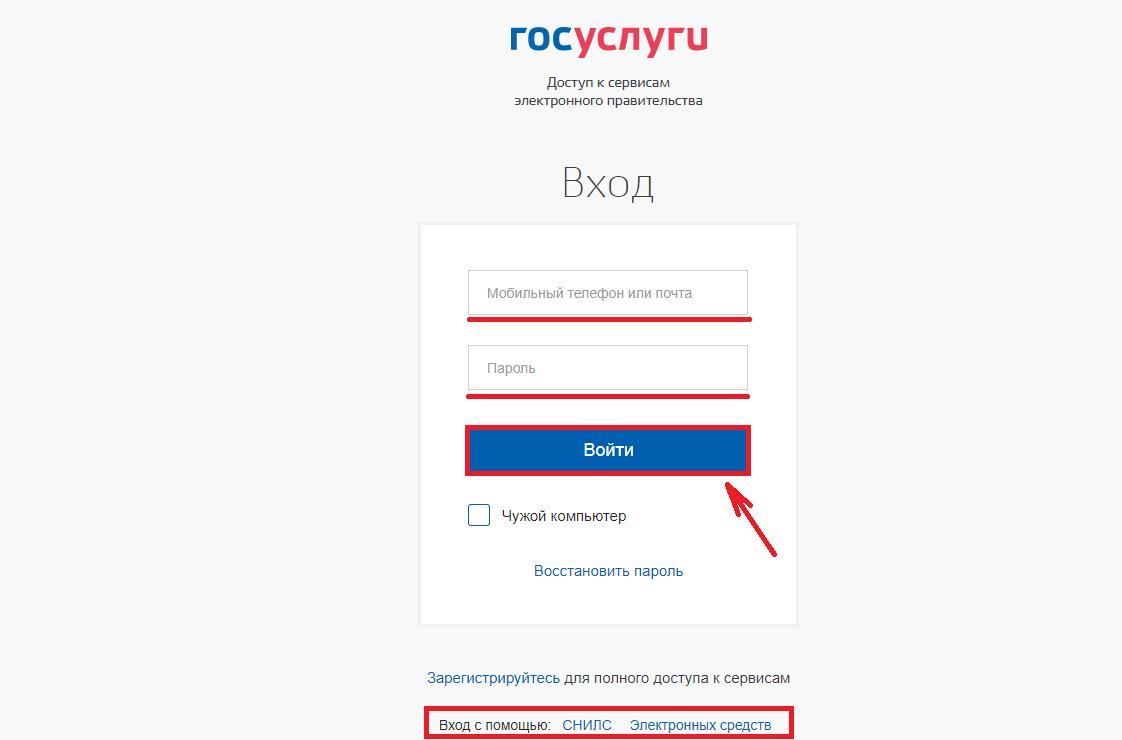


Рисунок 2.5 - В открывшемся окне авторизации введите свои учетные данные – логин и пароль. Авторизация возможна по номеру мобильного телефона, электронной почты, СНИЛС.

После успешного прохождения процедуры авторизации в правом верхнем углу вместо надписи «Личный кабинет» появится Ваша фамилия и список медицинских организаций. (Рисунок 2.6)

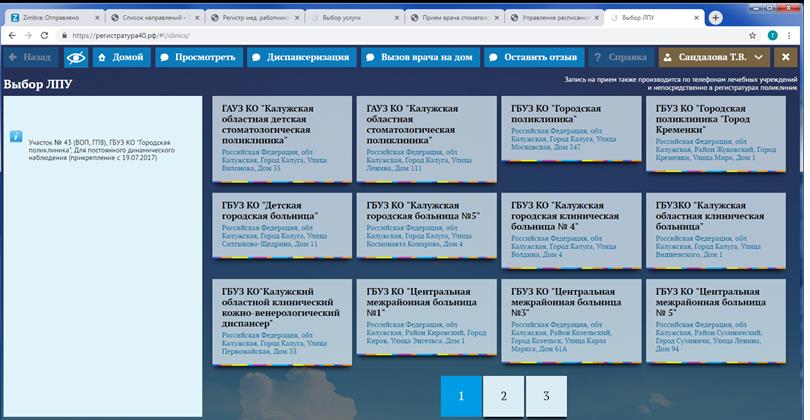


Рисунок 2.6 – Окно после входа

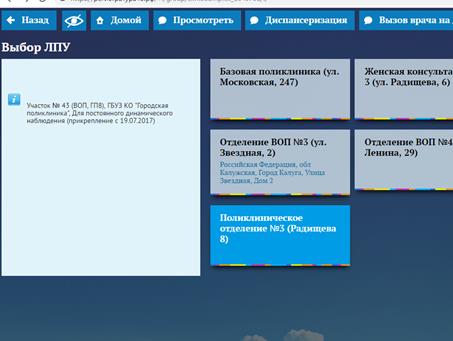


Рисунок 2.7 - Выберите поликлинику (больницу), отделение



Рисунок 2.8 – Выбор специалиста

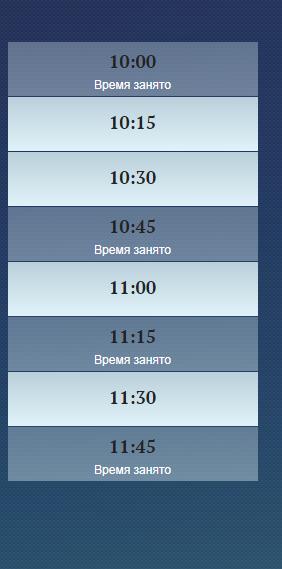


Рисунок 2.9 - При наличии у врача свободных талонов, выберите подходящий временной интервал.

Недостатком данного способа является то, что и в этом случае потребуется быть зарегистрированным на сайте госуслуг. Сайт предназначен для жителей Калуги и соответственно охватывает не настолько большое число людей как портал госуслуг, в следствие чего меньше подвержен сбоям в работе.

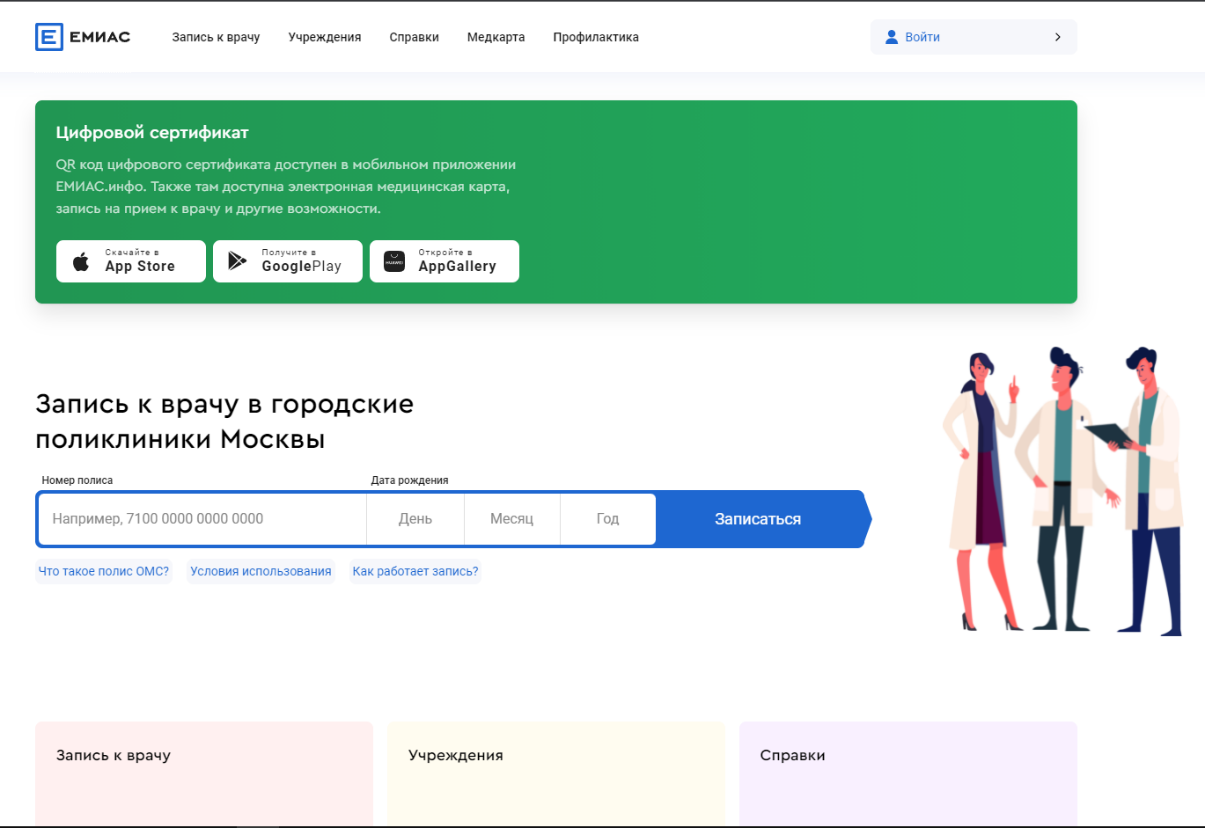
**3)emias.info**

Рисунок 2.10 – Версия сайта в браузере

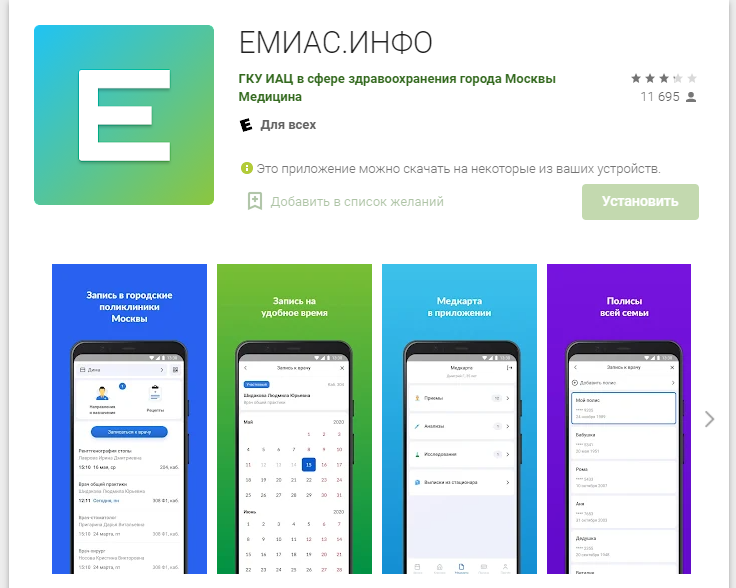


Рисунок 2.11 - Версия мобильного приложения

Приложение ЕМИАС.ИНФО предоставляет запись к врачам в городские поликлиники Москвы и доступ к своей электронной медкарте. Для просмотра медкарты необходимо оформить доступ к карте на портале mos.ru.  
  
Возможности:  
- запись на прием к врачу, просмотр/отмена/перенес записей;  
- запись к специалистам по направлениям;  
- запись на лабораторные исследования по назначениям;  
- просмотр выписанных рецептов;  
- автоматические напоминания о записях на прием;  
- возможность подключить несколько полисов ОМС.  
- электронная медицинская карта (осмотры врача, анализы, исследования, выписки)  
  
Для записи к врачу необходимо иметь:  
1. Полис ОМС, зарегистрированный в Москве;  
2. Прикрепление в любой Московской поликлинике.

Этот способ имеет такое же преимущество в виде не столь большого охвата пользователей как госуслуги. Также есть возможность воспользоваться мобильным приложением для записи. Недостатком является дополнительная регистрация на портале mos.ru.

* 1. **Перечень задач, подлежащих решению в процессе разработки.**
* Реализация взаимодействия приложения с базой данных;

Реализация методов записи пациентов на прием в сетку расписания в базе данных; добавления специальностей или специалистов; ввод карточек пациентов;

Разработка пользовательского интерфейса программы, соответствующего заданным требованиям.

**2.4 Обоснование выбора инструментов и платформы для разработки.**

Средой проектирования, позволяющими реализовать подобное программное обеспечение, была выбрана Microsoft Visual Studio. Среда разработки Visual Studio представляет собой полный набор средств разработки для создания веб-приложений ASP.NET, XML (веб-службы), настольных приложений и мобильных приложений. В этих языках используются функциональные возможности платформы .NET Framework, которая позволяет получить доступ к ключевым технологиям, упрощающим разработку веб-приложений ASP и XML (веб-службы). Можно создать обычные приложения Microsoft Windows и приложения с архитектурой "клиент-сервер" с помощью использования конструкторов в Visual Studio.

Преимуществами выбранной среды программирования являются удобство интерфейса и сравнительно невысокие требование к системе.

В качестве БД была выбрана Microsoft SQL Server 2019. Это популярная свободная объектно-реляционная система управления базами данных. Она базируется на языке SQL и поддерживает многочисленные возможности:

* поддержку БД неограниченного размера;
* мощные и надёжные механизмы транзакций и репликации;
* удобное использование хранимых процедур
* наследование;
* легкая расширяемость.

Платформой для разработки была выбрана Windows 10. Это доминирующая и наиболее распространённая операционная система, которая имеет наибольшую поддержку работы с различным ПО. Windows предоставляет четыре основные платформы приложений, каждая из которых имеет свои преимущества: [универсальная платформа Windows (UWP)](https://docs.microsoft.com/ru-ru/windows/apps/desktop/choose-your-platform#uwp). На этой платформе предоставляется система общих типов, интерфейсы API и модель приложений для всех устройств под управлением Windows 10. Приложения UWP могут быть нативными или управляемыми.

[WPF](https://docs.microsoft.com/ru-ru/windows/apps/desktop/choose-your-platform#wpf) и [Windows Forms](https://docs.microsoft.com/ru-ru/windows/apps/desktop/choose-your-platform#windows-forms). Эти платформы на базе .NET предоставляют систему общих типов, интерфейсы API и модель приложений для управляемых приложений.

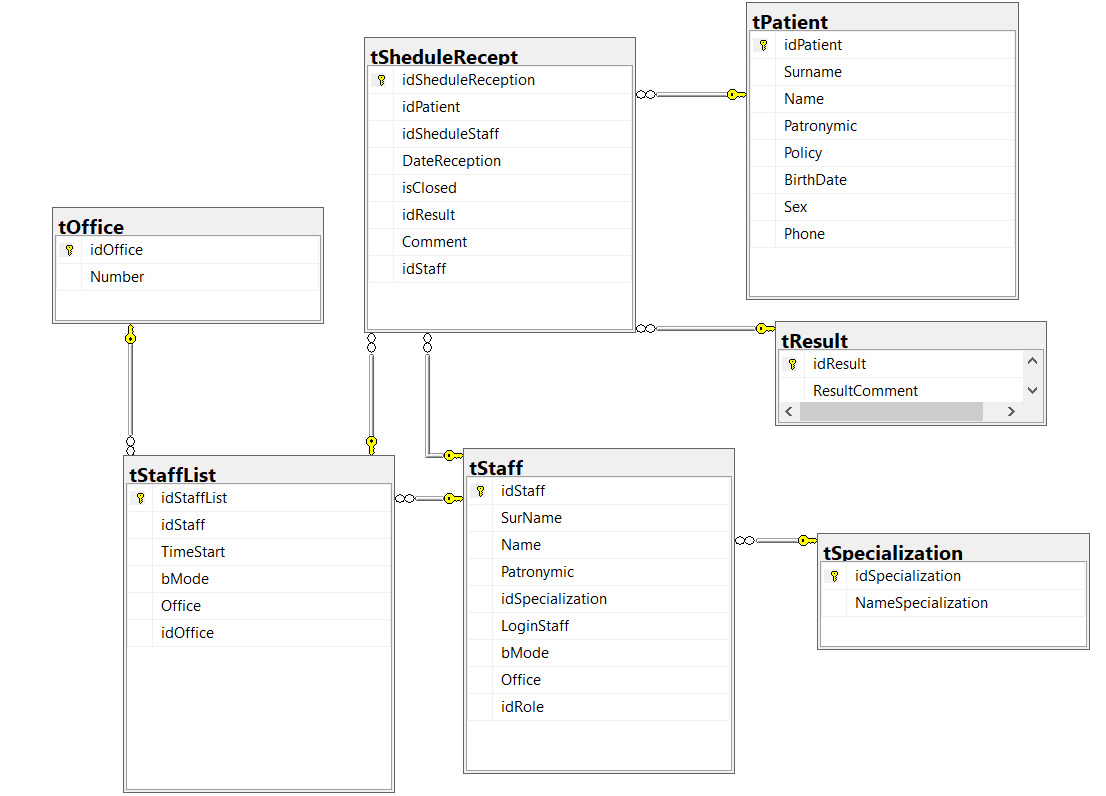
В качестве языка программирования для реализации проекта был выбран С#. C# — это объектно- и компонентно-ориентированный \_ язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Программы C# выполняются в .NET, виртуальной системе выполнения, вызывающей общеязыковую среду выполнения (CLR) и набор библиотек классов. Благодаря этому C# подходит для создания и применения программных компонентов.

Рисунок 2.12 – Схема базы данных

**Проектно-конструкторская часть**

1. **Разработка структуры системы**

Программа создана как Win32-приложение, проект WinForm.

Приложение должно давать возможность организовывать работу поликлиники в части регистрации пациентов, хранения их персональных данных, регистрации пациентов при обращении и организовывать их фактический прием, согласно сформированному расписанию. Потребителем является пациент, взаимодействующий с регистратором. Регистратор, в свою очередь, работает непосредственно в приложении. Приложение выполнено графическим интерфейсом, в котором пользователь выполняет запись к врачу в соответствии с расписанием.

Все компоненты разрабатываются на Windows 10 в среде разработки Visual Studio 2019, используя C# в качестве языка программирования. СУБД MS SQL Server EXPRESS 2019.

В Microsoft visual studio community создается приложение Windows с помощью «мастера классических приложений Windows». В приложении проект организован в виде визуальных форм по смыслу (справочники, форма входа и т.д.)

1. **Разработка алгоритмов обработки информации.**

Вход в приложение осуществляется под выбранным пользователем. Для записи пациента выбирается вкладка «Работа», затем в выпадающем списке «Выдача талонов». В открывшемся окне отображается список существующих записей. Правой кнопкой мыши открывается контекстное меню, выбрав пункт «Новый талон на прием к врачу», открывается окно для записи. Из списка зарегистрированных пациентов выбирается человек и далее заполняется карточка записи.

1. **Разработка архитектуры приложения**

**Используемые элементы интерфейса:**

В интерфейсе задействован стандартный набор элементов Windows Forms. Можно перечислить (Buttons, TextBox …)

**Получение данных:**

В качестве хранилища используется реляционная СУБД MS SQL EXPRESS 2019. База приложения – «polyclinic». В ней созданы 7 связанных таблиц. Для управления данными в базе созданы 26 хранимых процедур. Для организации бизнес-логики приложения в базе используются таблицы справочники и таблицы хранения результата производственной деятельности.

**Создание элементов интерфейса:**

Создание проекта и выбор шаблона «Приложение Windows Forms». В полученном проекте появляется одна форма и модуль (класс) Program.cs. По смыслу задания нам понадобятся формы для следующих задач:

* справочники: «Специальности», «Сотрудники», «Пациенты», «Кабинеты»;
* рабочие формы: вход в систему, «Расписание работы сотрудника», «Расписание приема пациентов» и форма для выдачи талонов, она же собственно прием врача.

На перечисленных формах в конструкторе в удобном порядке располагаются необходимые органы управления. В нашем случае использовались ярлыки (Label), текстовые окна (TextBox), выпадающие списки (ComboBox), кнопки (Button), календарь (DateTimePicker) и сетки данных (DataGrid). Этих типов элементов достаточно, чтобы отражать на экране и зачитывать необходимую информацию.

Получение данных:

Получение и хранение данных в приложение обеспечивается взаимодействием с СУБД MS SQL EXPRESS 2019. В C# за это отвечает класс SqlData. Он подключается директивами using System.Data и using System.Data.SqlClient;

Для соединения с нашей базой описывается строка соединения и передается в дочерние классы SqlDataAdapter – если нас интересует возвращенный набор DataSet (select) или SqlCommand – если мы только воздействуем на данные, не возвращая набор в приложение (update, delete, insert).

Для удобства все функции и процедуры по управлению данными собраны в одном не визуальном классе ClassMy.

Ниже некоторые примеры из него в коде C#:

Добавление пациента.

public static long InsertPatient(string Surname, string Name, string Patronymic, DateTime BirthDate, bool Sex, string Policy, string Phone)

{

long i = 0;

try

{

SqlConnection conn = new SqlConnection(ConnectString);

conn.Open();

SqlCommand cmdSel = new SqlCommand("sp\_InsertPatient", conn)

{

CommandType = CommandType.StoredProcedure,

CommandTimeout = 600

};

cmdSel.Parameters.Add("@SurName", SqlDbType.VarChar, 100);

cmdSel.Parameters["@SurName"].Value = Surname;

cmdSel.Parameters.Add("@Name", SqlDbType.VarChar, 100);

cmdSel.Parameters["@Name"].Value = Name;

cmdSel.Parameters.Add("@Patronymic", SqlDbType.VarChar, 100);

cmdSel.Parameters["@Patronymic"].Value = Patronymic;

cmdSel.Parameters.Add("@Policy", SqlDbType.VarChar, 100);

cmdSel.Parameters["@Policy"].Value = Policy;

cmdSel.Parameters.Add("@birthdate", SqlDbType.DateTime);

cmdSel.Parameters["@birthdate"].Value = BirthDate;

cmdSel.Parameters.Add("@sex", SqlDbType.Bit);

cmdSel.Parameters["@sex"].Value = Sex;

cmdSel.Parameters.Add("@idpatient", SqlDbType.BigInt);

cmdSel.Parameters["@idpatient"].Direction = ParameterDirection.Output;

cmdSel.Parameters.Add("@phone", SqlDbType.VarChar,10);

cmdSel.Parameters["@phone"].Value = Phone;

cmdSel.ExecuteNonQuery();

i = Convert.ToInt64(cmdSel.Parameters["@idPatient"].Value);

conn.Close();

cmdSel.Dispose();

conn = null;

return i;

}

catch (Exception e)

{

MessageBox.Show(e.Message);

return i;

}

}

Сама хранимая процедура MS SQL сервера:

ALTER PROCEDURE [dbo].[sp\_InsertPatient]

(

@Surname varchar(100),

@Name varchar(100),

@Patronymic varchar(100),

@Policy varchar(100),

@BirthDate datetime,

@Sex bit,

@Phone varchar(10),

@idPatient bigint output

)

AS

BEGIN

SET NOCOUNT ON;

INSERT INTO [dbo].[tPatient]

([Surname]

,[Name]

,[Patronymic]

,[Policy]

,[BirthDate]

,[Sex]

,Phone)

VALUES

(@Surname,

@Name,

@Patronymic,

@Policy,

@BirthDate,

@Sex,

@Phone

)

select @idPatient = @@IDENTITY

END

Здесь видно, что в процедуру MS SQL передается ровно тот набор параметров, который необходим для создания записи о новом пациенте.

**При работе с пользователем:**

Перечисленные выше элементы управления в режиме конструктора генерируют необходимые события, что освобождает нас от необходимости подробно самостоятельно создавать их в коде. Нами используются, для примера, следующие события:

* нажатия кнопок
* изменение выбранного элемента в выпадающих списках
* навигация по сетке с данными
* выбор даты календаря

Факт генерации таких событий удобен, чтобы на них «повесить» необходимую бизнес-логику в нужных функциях и процедурах.

Ниже некоторые примеры таких событий из кода:

Пример смены в списке специальностей на форме выдачи талона к врачу, в результате чего произойдет заполнение списка врачей строго по выбранной специальности:

private void cmbSpec\_SelectedIndexChanged(object sender, EventArgs e)

{

if (cmbSpec.SelectedIndex == -1)

{

idSpec = 0;

cmbStuff.DataSource = null;

cmbStuff.DisplayMember = "Fio";

cmbStuff.ValueMember = "idStaff";

return;

}

idSpec = Convert.ToInt64(cmbSpec.SelectedValue);

this.cmbStuff.SelectedIndexChanged -= new System.EventHandler(this.cmbStuff\_SelectedIndexChanged);

dt2 = ClassMy.SelectStaff4spec(idSpec).Tables[0];

if (dt2.Rows.Count == 0)

{

cmbStuff.DataSource = null;

}

else

{

cmbStuff.DataSource = dt2;

}

cmbStuff.DisplayMember = "Fio";

cmbStuff.ValueMember = "idStaff";

this.cmbStuff.SelectedIndexChanged += new System.EventHandler(this.cmbStuff\_SelectedIndexChanged);

}

1. **Разработка и описание графического интерфейса взаимодействия пользователя с сис­темой**

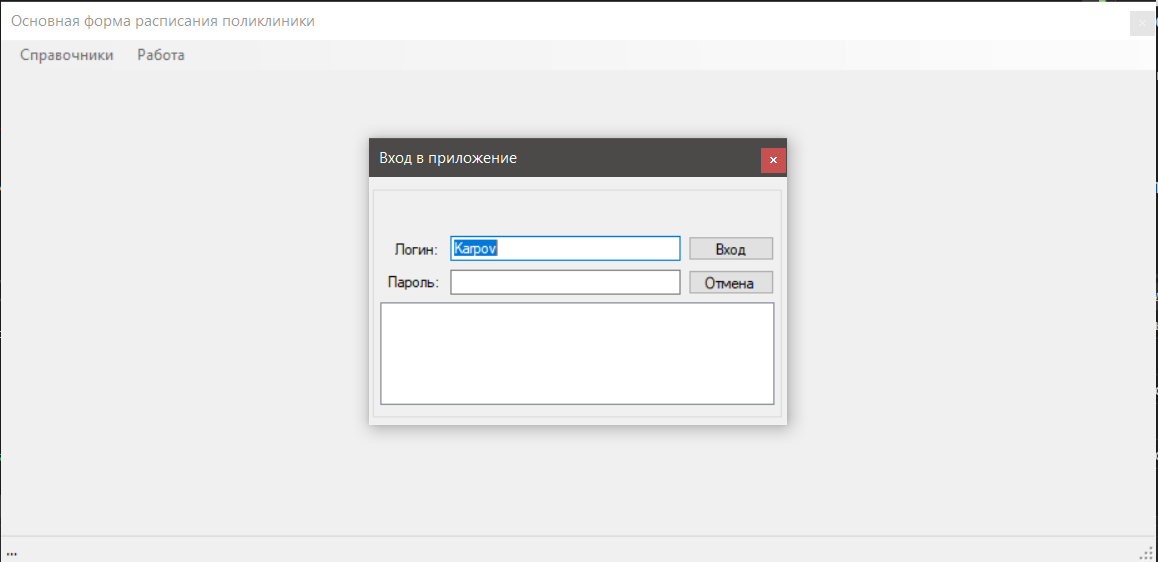


Рисунок 3.1 – окно для входа в профиль

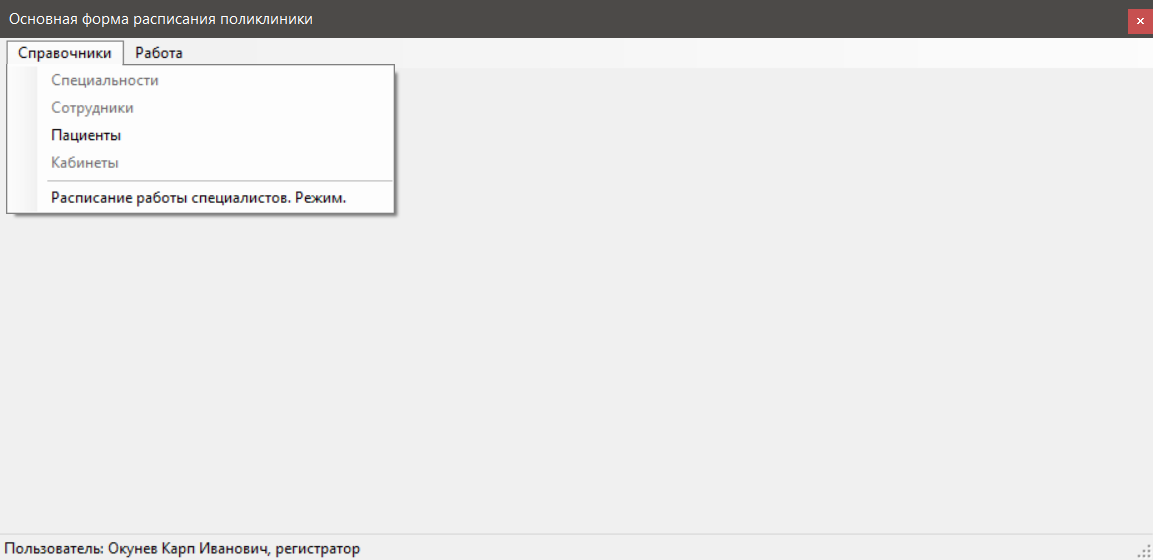


Рисунок 3.2 – Главное окно приложения с открытой вкладкой «Справочники»

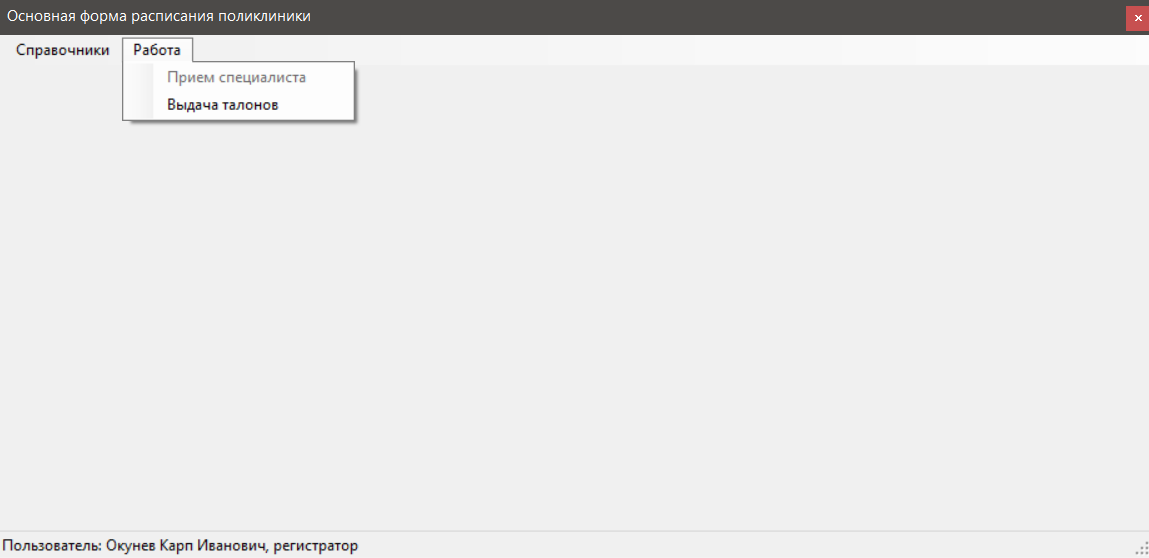


Рисунок 3.3 – Главное окно приложения с открытой вкладкой «Работа»

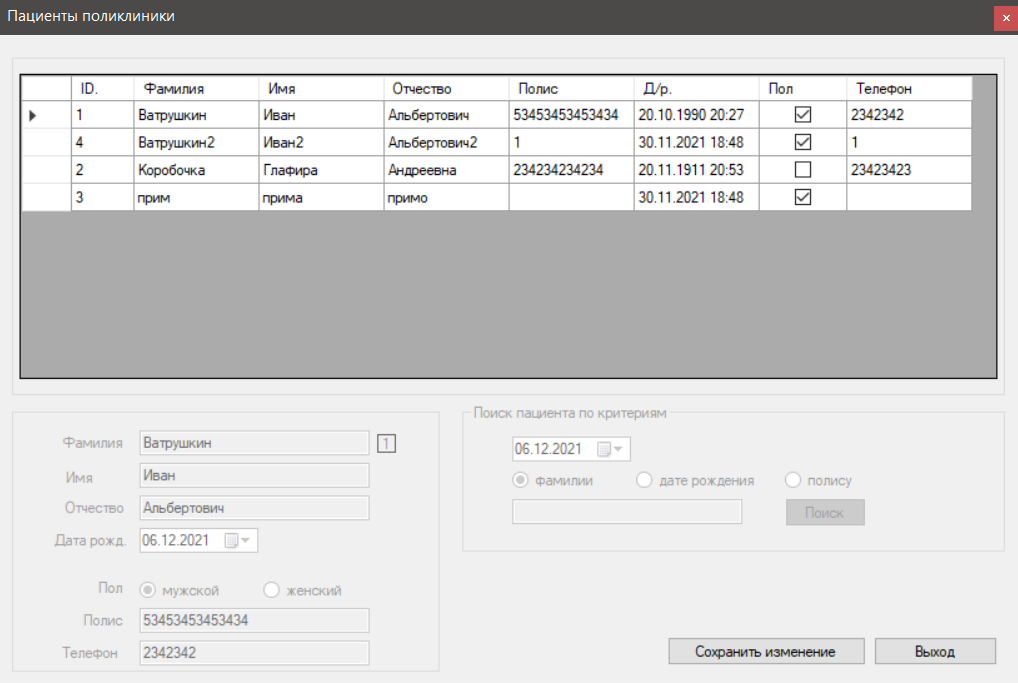


Рисунок 3.4 – Окно справочника «Пациенты»

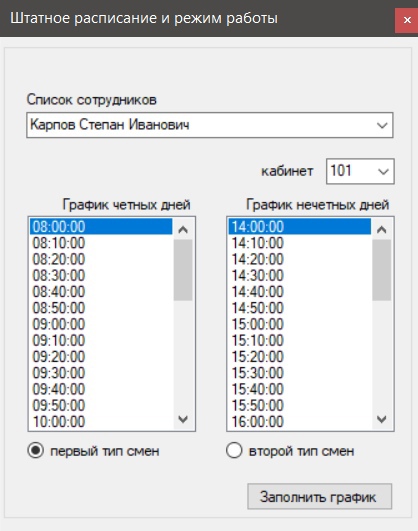


Рисунок 3.5 – Окно справочника «Расписание работы специалистов. Режим.»

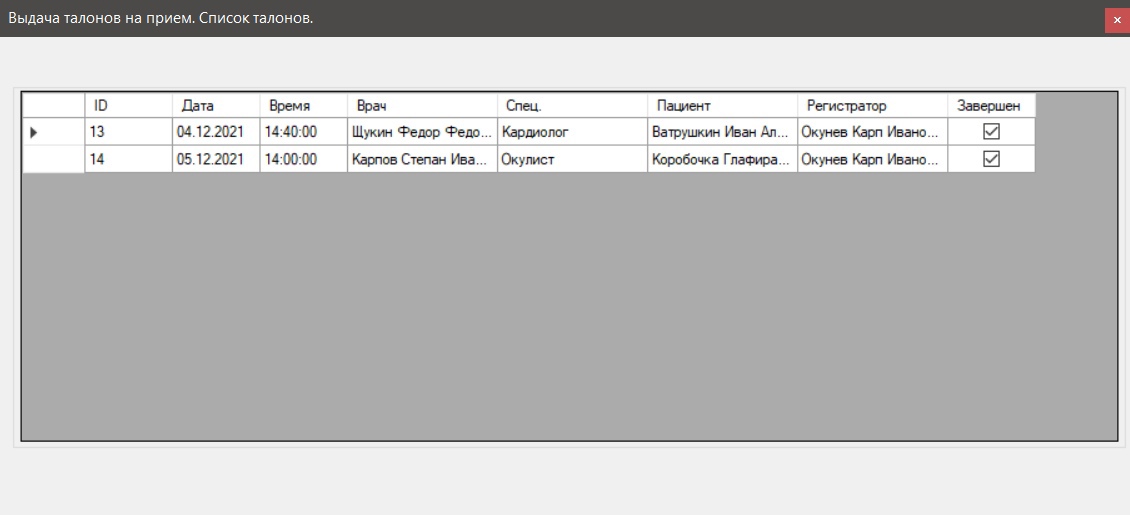


Рисунок 3.6 – Окно выдачи талонов со списком уже существующих записей

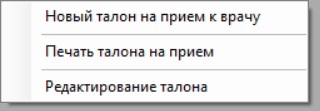


Рисунок 3.7 – Контекстное меню в окне для выдачи талонов

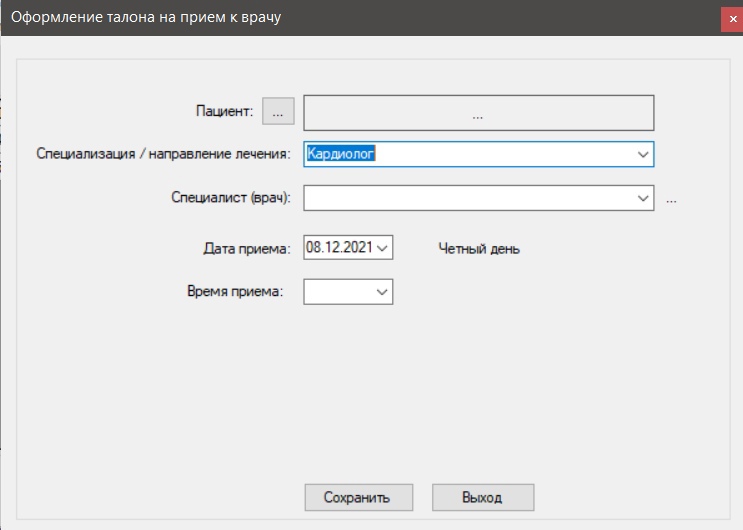


Рисунок 3.7 – Окно заполнения талона

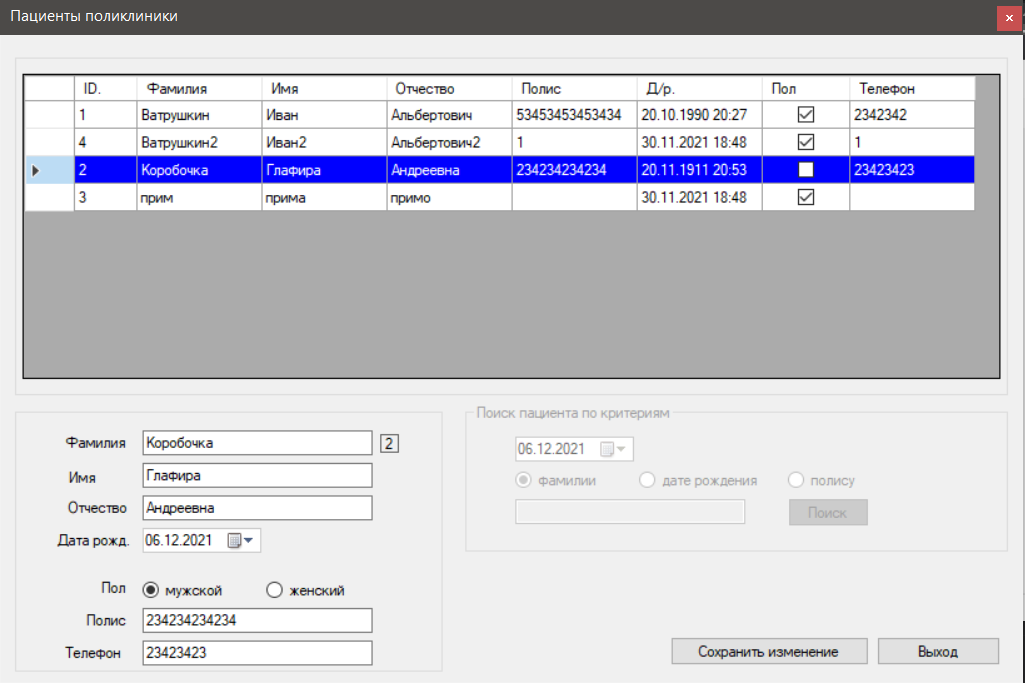


Рисунок 3.8 – Окно справочника пациентов.

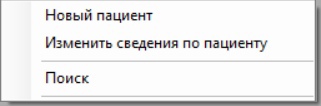


Рисунок 3.9 – Контекстное меню в справочнике пациентов

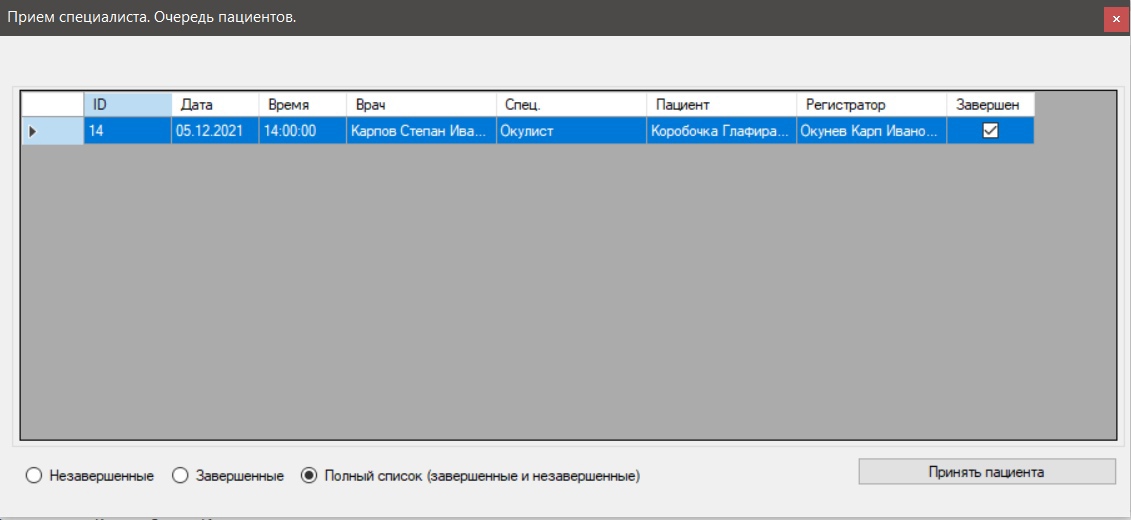


Рисунок 3.10 – Окно приема пациентов для специалиста

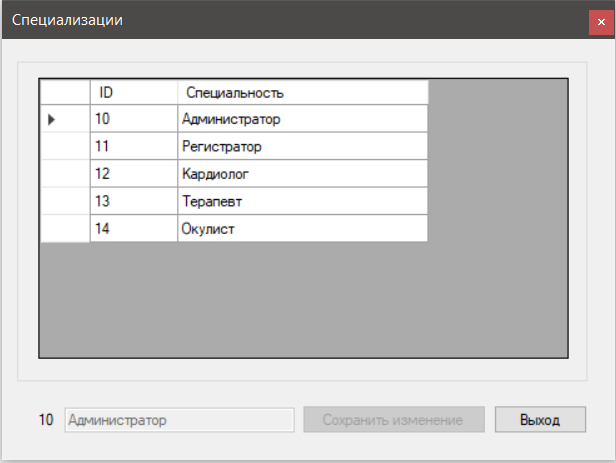


Рисунок 3.11 – Окно справочника специализаций

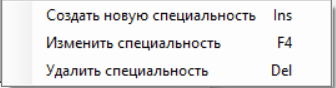


Рисунок 3.12 – Контекстное меню справочника специализаций

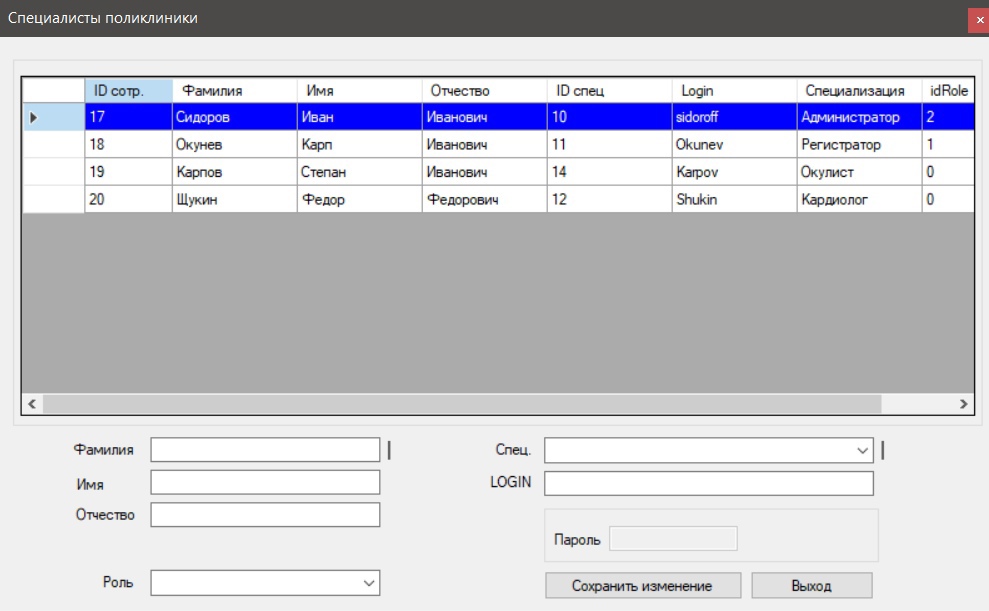


Рисунок 3.13 – Окно справочника сотрудников

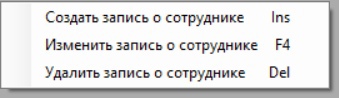


Рисунок 3.14 – Контекстное меню справочника сотрудников

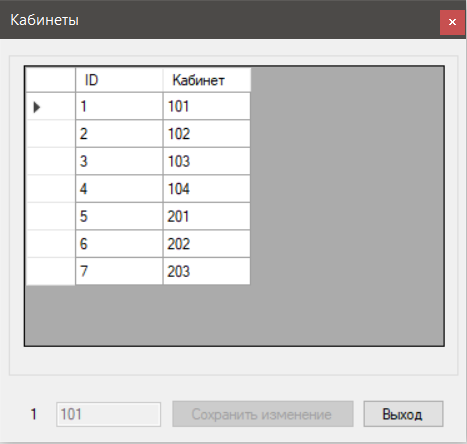


Рисунок 3.15 – Окно справочника кабинетов

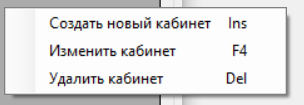


Рисунок 3.16 – Контекстное меню справочника кабинетов

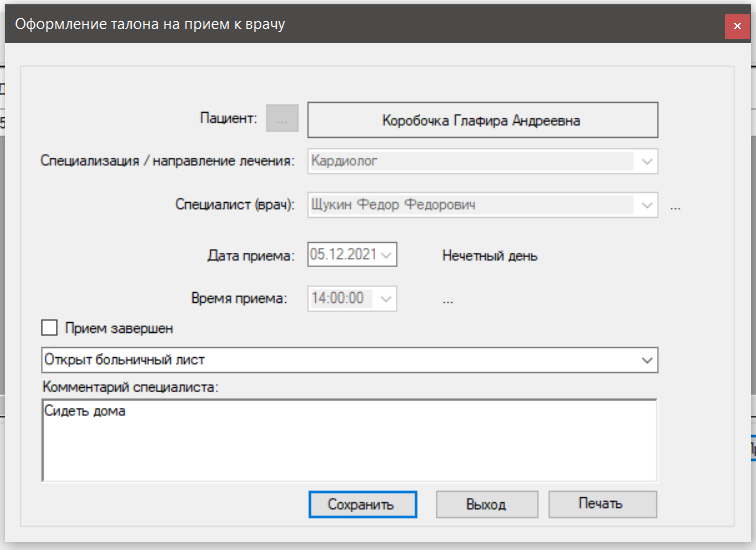


Рисунок 3.17 – Окно приема пациента у специалиста

**Проектно-технологическая часть**

1. **Тестирование и отладка макета рабочей программы**

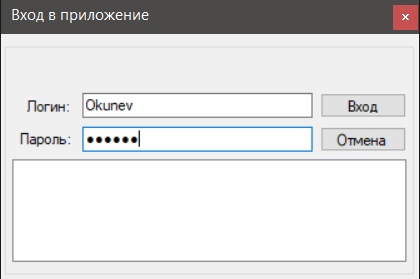


Рисунок 4.1 – Вводим данные от профиля пользователя (в данном случае регистратора)

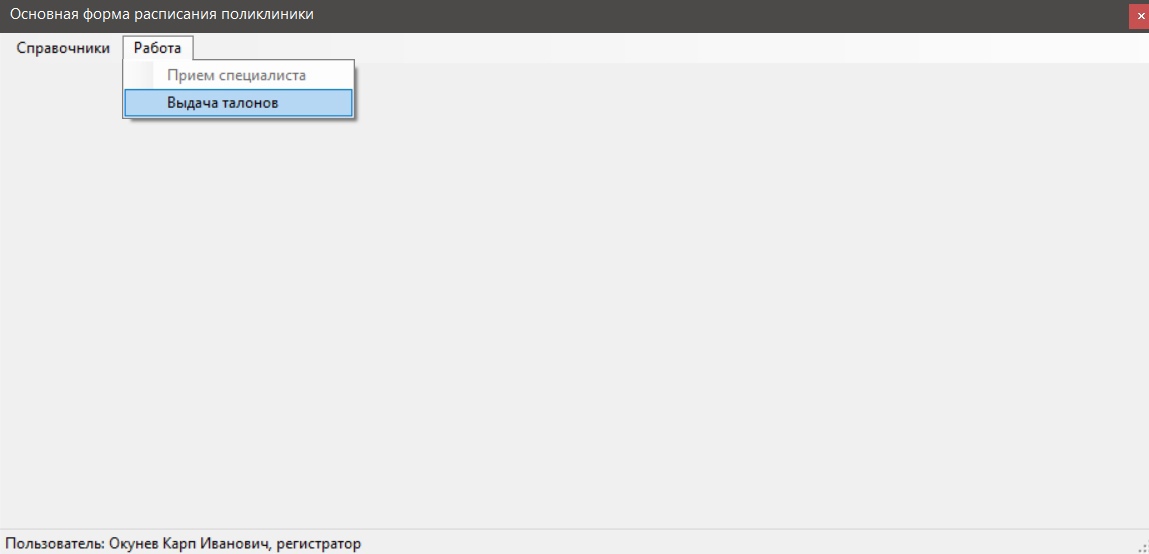


Рисунок 4.2 – В главном окне выбираем пункт «Работа», затем «Выдача талонов»

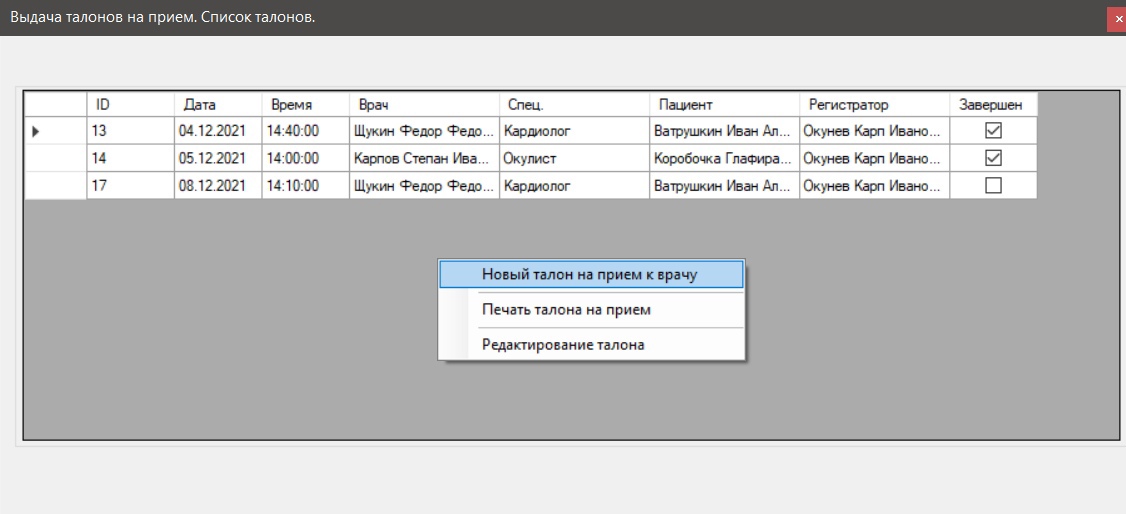


Рисунок 4.3 – В открывшемся окне нажимаем пкм и из списка выбираем пункт «Новый талон на прием к врачу»

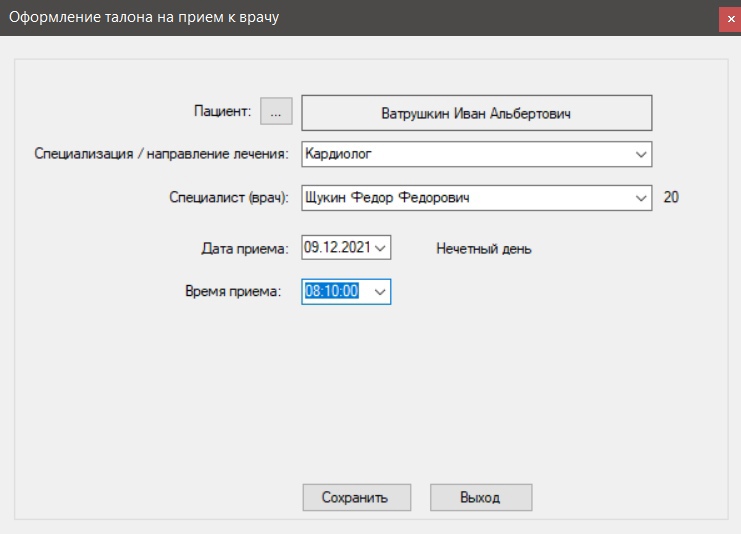


Рисунок 4.4 – Заполняем все необходимые поля в карточке талона и нажимаем кнопку «Сохранить»

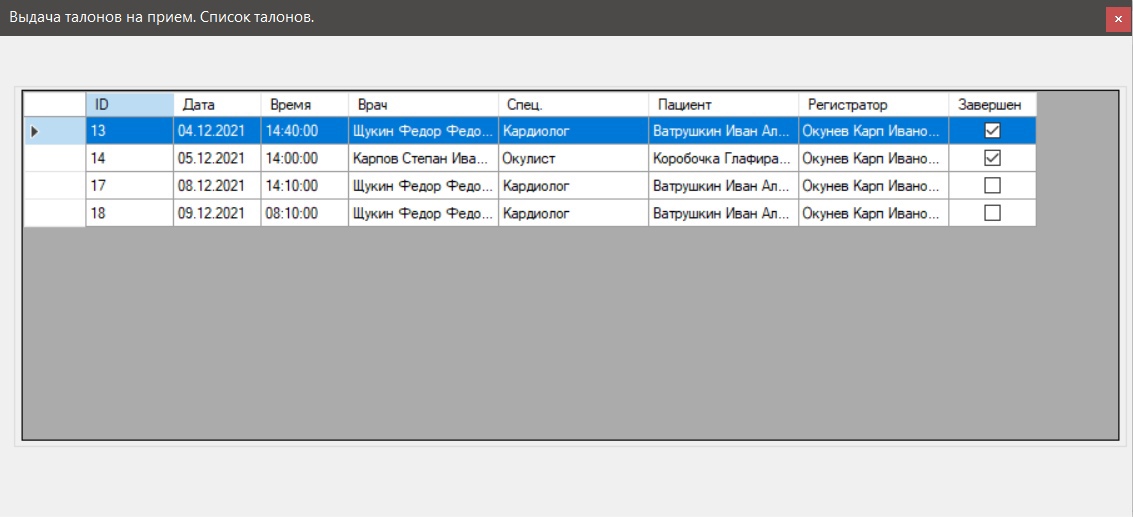


Рисунок – 4.5 Теперь в списке существующих талонов появился только что созданный талон на прием.

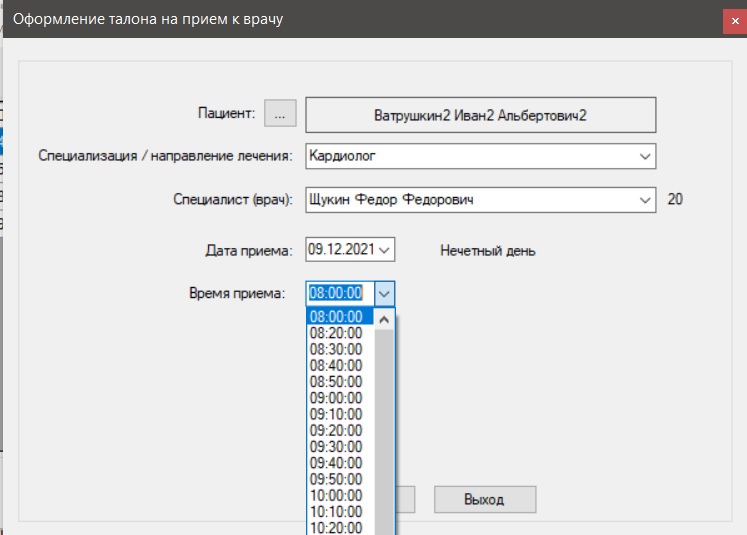


Рисунок – 4.6 Теперь если попытаться создать новый талон с тем же врачом и той же датой, ранее выбранное время приема не будет доступным для выбора. Талон был создан успешно

1. **Разработка руководства пользователя**

После запуска программы необходимо ввести данные логина и пароля для входа в профиль со своей определенной ролью. После входа появляется главная панель с вкладками «Справочники» и «Работа». Для записи пациента на прием необходимо выбрать вкладку «Работа» и в выпадающем списке выбрать «Выдача талонов». В открывшемся окне отображается список существующих талонов, нажав правой кнопкой мыши по пустому месту, откроется контекстное меню, в которой необходимо выбрать пункт «Новый талон на прием к врачу», после этого заполняется карточка и необходимо нажать кнопку «Сохранить». Если пациент не зарегистрирован в базе, то на этапе выбора пациента при создании талона необходимо нажать правую кнопку мыши и в появившемся меню выбрать «Новый пациент» и заполнить его данные. Для добавления нового специалиста необходимо на главном экране открыть вкладку «Справочники» и выбрать пункт «Сотрудники». В открывшемся списке, нажав по пустому месту пкм, в списке выбираем «Создать запись о сотруднике». Далее заполняем данные сотрудника и жмем кнопку «Сохранить изменение». Аналогичные действия выполняются для добавления пациентов через вкладку «Пациенты».

**Заключение**

В ходе выполнения курсовой работы была выполнена задача проектирования по формированию практических навыков по разработке и реализации программного приложения с использованием баз данных.

В курсовой работе было разработано программное обеспечение, позволяющее записывать и выдавать талоны для пациентов учреждения здравоохранения (поликлиники) на прием к специалистам и последующего приема по заданному расписанию.

# Список источников

1. Шнырёв, С.Л.  Базы данных: учебное пособие для вузов - М. : НИЯУ МИФИ, 2011. — 224 с. — Режим доступа: <http://e.lanbook.com/book/75809>

2. Ревунков, Г.И. Базы и банки данных - М.: МГТУ им. Н.Э. Баумана, 2011. — 68 с. — Режим доступа: <http://e.lanbook.com/book/52425>

3. Ревунков, Г.И. Проектирование баз данных - М.: МГТУ им. Н.Э. Баумана, 2009. — 20 с. — Режим доступа: <http://e.lanbook.com/book/52390>

4. Кудрявцев, К.Я. Создание баз данных: учебное пособие — М.: НИЯУ МИФИ, 2010. — 155 с. — Режим доступа: <http://e.lanbook.com/book/75822>

5**.** Сидоров В.Н., Сломинская Е.Н., Полникова Т.В., Макарова О.Ю. Оформление графической части выпускной квалификационной работы. Учебное пособие. М.: МГТУ им. Н.Э. Баумана, 2016.

6. C# SqlDataAdapter Класс

https://docs.microsoft.com/ru-ru/dotnet/api/system.data.sqlclient.sqldataadapter?view=dotnet-plat-ext-6.0

7. C# SqlCommand Класс

https://docs.microsoft.com/ru-ru/dotnet/api/system.data.sqlclient.sqlcommand?view=dotnet-plat-ext-6.0

8. C# Элемент управления DataGridView (сетка данных)

https://docs.microsoft.com/ru-ru/dotnet/desktop/winforms/controls/datagridview-control-code-directory-windows-forms?view=netframeworkdesktop-4.8

9. MS SQL создание логина для входа

https://docs.microsoft.com/ru-ru/sql/t-sql/statements/create-login-transact-sql?view=sql-server-ver15

10. MS SQL создание пользователей базы данных

https://docs.microsoft.com/ru-ru/sql/relational-databases/security/authentication-access/create-a-database-user?view=sql-server-ver15