**Федеральное казенное профессиональное образовательное учреждение**

**«Калачевский техникум-интернат»**

**Министерства труда и социальной защиты Российской Федерации**

|  |  |
| --- | --- |
| Допущен к защите «\_\_» \_\_\_\_\_\_2024 г.  Руководитель ЦМК информационных технологий и математических дисциплин  \_\_\_\_\_\_\_\_\_\_ Н.С. Житник | Защищена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_  Председатель ГЭК  Директор ООО «К–Систем»  \_\_\_\_\_\_\_\_\_\_\_\_\_Ю.А. Факов |

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

(дипломный проект)

**РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ «ЭЛЕКТРОННАЯ РЕГИСТРАТУРА»**

**09.02.03 Программирования в компьютерных системах**

**Исполнитель:**

студент группы Т-43

Харитонов Иван Альбертович

**Научный руководитель:**

преподаватель профессионального цикла

Богатырева Маргарита Анатольевна

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

**Рецензент:**

Козлова Ольга Владимировна

Калач-на-Дону

2024

СОДЕРЖАНИЕ

[СПИСОК СОКРАЩЕНИЙ 3](#_Toc169347686)

[ВВЕДЕНИЕ 4](#_Toc169347687)

[1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ ОРГАНИЗАЦИИ. ОБЗОР ОСОБЕННОСТЕЙ И ПРОБЛЕМ ФУНКЦИОНИРОВАНИЯ 7](#_Toc169347688)

[1.1. Краткая характеристика организации 7](#_Toc169347689)

[1.1.1. Регистратура как структурное подразделение ГБУЗ «Калачевская ЦРБ» 9](#_Toc169347690)

[1.2. Обоснование необходимости разработки веб-приложения «Электронная регистратура» 12](#_Toc169347691)

[1.3. Обзор программ-аналогов российского производства 14](#_Toc169347692)

[2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗАЦИИ ЗАПИСИ НА ПРИЁМ 22](#_Toc169347693)

[2.1 Методологии проектирования программного продукта 22](#_Toc169347694)

[2.2 Разработка функциональной модели приложения с помощью методологии IDEF0 «AS-IS» 23](#_Toc169347695)

[2.3 Диаграмма потоков данных 28](#_Toc169347696)

[2.4 Проектирование структуры данных 30](#_Toc169347697)

[2.5 Разработка UML диаграмм 32](#_Toc169347698)

[2.5.1 Диаграмма прецедентов 33](#_Toc169347699)

[2.5.2 Диаграммы активностей 34](#_Toc169347700)

[2.5.3 Диаграммы последовательности 37](#_Toc169347701)

[2.6 Проектирование пользовательского интерфейса 50](#_Toc169347702)

[2.7 Описание инструментальных и языковых средств разработки 56](#_Toc169347703)

[3 РЕАЛИЗАЦИЯ ПРОГРАММЫ 59](#_Toc169347704)

[3.1. Дизайнерское решение разработки веб-приложения 59](#_Toc169347705)

[3.2. Работа пациента с приложением 60](#_Toc169347706)

[3.3. Работа менеджера регистратуры с приложением 65](#_Toc169347707)

[3.4. Тестирование 72](#_Toc169347708)

[ЗАКЛЮЧЕНИЕ 74](#_Toc169347709)

[СПИСОК ИСТОЧНИКОВ 76](#_Toc169347710)

[ПРИЛОЖЕНИЯ 77](#_Toc169347711)

# СПИСОК СОКРАЩЕНИЙ

**ВП** — веб-приложение

**БД** — база данных

**ЭР** — электронная регистратура

**ГБУЗ «Калачевская ЦРБ»**— Государственное Бюджетное Учреждение Здравоохранения «Калачевская Центральная Районная Больница»

**ТЗ** — техническое задание

**МО** — медицинская организация

**МУ** — медучреждение

**ЛПУ** — лечебно-профилактическое учреждение

**ЛК** — личный кабинет

# ВВЕДЕНИЕ

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 сказывалось на скорости и качестве обслуживания пациентов, затрудняло работу врачебного и медицинского персонала, что вело к врачебным ошибкам, большим затратам времени на заполнение карт, составление отчетов.

С началом реализации национального проекта «Здравоохранение» и ряда региональных программ комплексной информатизации муниципальных учреждений процесс информатизации здравоохранения значительно ускорился, а число масштабных проектов в этой сфере заметно увеличилось.

На смену эпохе бумажных носителей информации и самописных программ для медицинских организаций приходит время крупных информационных систем, направленных как на поддержание внутренних функций, так и на решение задач управления медицинской отраслью.

Такие приложения позволяют пациентам самостоятельно записываться на приём в любое время суток, из любой точки, где есть доступ к интернету. Это существенно экономит силы и время пациентов, избавляет от необходимости стоять в очередях, и делает медицинские услуги более доступными.

Электронная регистратура — это комплекс сервисов, предназначенных для осуществления предварительной записи пациентов на амбулаторно-поликлинический прием:

* к врачам общей (семейной) практики;
* к участковым врачам-терапевтам и врачам-педиатрам;
* к врачам акушерам-гинекологам женских консультаций;
* к врачам стоматологам;
* к узким специалистам;
* к медицинским работникам центров здоровья.

Электронная регистратура призвана оптимизировать порядок обращения пациента за медицинской помощью за счет сокращения времени ожидания в очередях, что способствует снижению социального напряжения. Благодаря этому можно контролировать степень загруженности персонала и разделять потоки пациентов.

**Объект исследования:** Государственное бюджетное учреждение здравоохранения «Калачевская районная больница».

Больница — это медицинское учреждение, специализированное на оказании медицинской помощи пациентам, диагностике, лечении и реабилитации. В больнице проводится широкий спектр медицинских услуг и процедур, а также осуществляются различные медицинские деятельности.

**Предмет исследования**: деятельность структурного подразделения "Регистратура" ГБУЗ "Калачевская ЦРБ".

**Цель** данной выпускной квалификационной работы заключается в разработке программного модуля «Электронная регистратура».

**Задачи:**

* исследование предметной области;
* определение особенностей и выявление проблем при работе регистратуры
* формирование функциональных требований к приложению;
* определение инструментальных средств;
* проектирование приложения;
* разработка приложения.

Пояснительная записка состоит из трех глав, введения, заключения и приложений.

Во введении приведены актуальность работы, объект и предмет исследования. Сформулированы цель и задачи курсового проекта, обоснована практическая значимость проекта.

В первой главе описано исследование предметной области: краткая характеристика исследуемой организации, ее деятельности. Проанализированы программы – аналоги, представленные на российском рынке.

Вторая глава посвящена проектированию веб-приложения. Определены методологии проектирования, на основе которых построена модель информационной системы.

В третьей главе описана реализация веб-приложения, продемонстрированы его функциональные возможности.

В заключении подведены итоги выпускной квалификационной работы на соответствие поставленной цели и полученных результатов.

В разделе «Приложения» приведена техническая документация:

* техническое задание (ТЗ);
* руководство пользователя.

# ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ ОРГАНИЗАЦИИ. ОБЗОР ОСОБЕННОСТЕЙ И ПРОБЛЕМ ФУНКЦИОНИРОВАНИЯ

## Краткая характеристика организации

Государственное бюджетное учреждение здравоохранения «Калачевская районная больница» является стационарным медицинским учреждением, направленным на лечение больных или специализированную углубленную дифференциальную диагностику заболеваний в стационарных условиях.

Полное наименование МО: ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ ЗДРАВООХРАНЕНИЯ «КАЛАЧЕВСКАЯ ЦЕНТРАЛЬНАЯ РАЙОННАЯ БОЛЬНИЦА».

Наименование МО (сокращенное): ГБУЗ «Калачевская ЦРБ».

Юридический адрес: 404503, Волгоградская обл., Калачевский район, Калач–на–Дону, ул. Маяковского, 27.

Тип организации: бюджетное учреждение.

Основной вид деятельности: деятельность больничных учреждений широкого профиля и специализированных.

Руководитель МО: Бессолов Виктор Олегович.

ГБУЗ «Калачевская ЦРБ» обеспечивает оказание медицинской помощи на территории всего Калачевского района силами центрального корпуса, расположенного в г. Калач–на–Дону, 2–х участковых больниц, 4–х врачебных амбулаторий и 28 фельдшерско-акушерских пунктов.

В структуру ГБУЗ «Калачевская ЦРБ» входят следующие отделения:

* стационарные: родильное, педиатрическое, инфекционное, анестезиологии и реанимации, терапевтическое и хирургическое;
* поликлинические: стоматологическое, педиатрическое, женская консультация, первичное онкологическое, терапевтическое, кабинеты узких специалистов;
* диагностические: клинико-диагностическая лаборатория, рентген-кабинет, УЗИ–диагностика, функциональная диагностика; эндоскопическое, физиотерапевтическое.

Цель работы МО: оказание первичной медико-санитарной помощи, первичной специализированной медико-санитарной помощи взрослому и детскому населению в рамках государственных гарантий оказания бесплатной медицинской помощи

Задачи:

1. Осуществление доступности медицинской помощи населению в рамках государственных гарантий оказания бесплатной медицинской помощи;
2. Повышение качества оказания медицинской помощи;
3. Снижение общей и первичной заболеваемости;
4. Развитие амбулаторно-поликлинического направления оказания медицинской помощи;
5. Обеспечение достаточного высококвалифицированного медицинского кадрового потенциала;
6. Развитие и поддержание материально-технической базы для оказания медицинской помощи на современном уровне;
7. Внедрение новых медицинских технологий.

Структурная схема ГБУЗ «Калачевская ЦРБ» представлена на рисунке 1.



Рисунок 1. Структурная схема ГБУЗ «Калачевская ЦРБ»

## Регистратура как структурное подразделение ГБУЗ «Калачевская ЦРБ»

Основным документом, регламентирующим деятельность регистратуры поликлиники, является Приложение N 4 к [Положению](https://base.garant.ru/70195856/53f89421bbdaf741eb2d1ecc4ddb4c33/#block_1000) об организации оказания первичной медико-санитарной помощи взрослому населению, утвержденному [приказом](https://base.garant.ru/70195856/) Министерства здравоохранения и социального развития Российской Федерации от 15 мая 2012 г. N 543н «Правила организации деятельности регистратуры поликлиники врачебной амбулатории, центра общей врачебной практики (семейной медицины)».

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

Основными задачами регистратуры медицинской организации являются:

* организация беспрепятственной и безотлагательной предварительной записи больных на прием к врачу, в том числе в автоматизированном режиме, в кабинет медицинской профилактики, кабинет доврачебной помощи (как при их непосредственном обращении в поликлинику, так и по телефону);
* организация и осуществление регистрации вызовов врачей на дом по месту жительства (пребывания) больного;
* обеспечение регулирования интенсивности потока населения с целью создания равномерной нагрузки врачей и распределение его по видам оказываемой помощи;
* систематизированное хранение медицинской документации пациентов, обеспечение своевременного подбора и доставки медицинской документации в кабинеты врачей.

Непосредственное руководство работой регистратуры медицинской организации осуществляет заведующий регистратурой, назначаемый на должность и освобождаемый от должности руководителем медицинской организации.

Визитной карточкой поликлиники ГБУЗ «Калачевская ЦРБ» является отдел регистратуры. Он выполняет функцию регистрации пациентов и обеспечивает их необходимой справочной информацией. Расположена регистратура на первом этаже поликлиники.

Работа регистратуры в поликлинике организована, исходя из принципов медицинского обеспечения прикрепленного контингента и сменного распорядка работы врачей поликлиники.

В целях информационного обеспечения пациентов о порядке работы поликлиники в вестибюле установлен стенд с графиком работы врачей отделений и кабинетов по всем специальностям с указанием часов их приема на все дни недели и номеров кабинетов.

На регистратуру возлагаются следующие задачи:

* учет прикрепленных пациентов;
* обеспечение записи на прием к врачам как при их непосредственном обращении в поликлинику, так и по телефону;
* регулирование движения потоков больных по видам оказания помощи с целью создания равномерной нагрузки на врачей;
* хранение и рациональное размещение амбулаторных карт в регистратуре;
* осуществление своевременной подборки и доставки амбулаторных карт в кабинеты врачей и их возврата в регистратуру после окончания приема больных;
* регистрация и оформление медицинской документации на прикрепляемых пациентов и периодическая перерегистрация прикрепленных;
* обеспечение устной и стендовой информацией, а также справками по телефону о распорядке работы поликлиники, времени приема и номерах кабинетов врачей всех специальностей и др.;
* оформление листов временной нетрудоспособности, подтверждение соответствующими печатями выданных пациентам справок, направлений, рецептов и выписок из амбулаторной карты.

В регистратуре поликлиники имеется возможность получения талона по предварительной записи за 14 дней. Записаться на прием к врачу можно по телефону регистратуры медицинской организации, при личном обращении в регистратуру, непосредственно на приеме у лечащего врача. При возникновении острого состояния медицинская помощь оказывается без предварительной записи в порядке общей очереди или в первоочередном порядке.

## Обоснование необходимости разработки веб-приложения «Электронная регистратура»

Многим памятно то время, когда ежедневно по утрам у поликлиник, порой задолго до начала работы регистратуры, выстраивались длинные очереди из пациентов, жаждущих попасть на приём к врачу. Запись производили день в день, талонов, естественно, всем желающим не хватало. Зачастую, чтобы попасть к узкому специалисту, дежурить приходилось у поликлиники и несколько дней.

Повсеместная модернизация системы учета сказалась и на лечебных учреждениях, где для выдачи талонов на посещение врачей стали использоваться электронные системы [6].

Электронная регистратура, она же — Единая медицинская информационно-аналитическая система (ЕМИАС), а проще, самозапись через интернет, — замечательное решение проблемы. [10]

Виды электронной регистратуры:

1. Электронная очередь. Применяется в клиниках на терминалах выдачи билетов с порядковыми номерами. Пациенты дожидаются приглашения по своему полученному номеру.
2. Запись через сайт. Многие клиники на своём сайте имеют данный модуль, который позволяет их пользователям записаться на посещение, выбрав удобное для себя время.
3. Терминал самозаписи. По сути, это тот же модуль сайта, только в упрощенном виде. Используется в виде терминалов, установленных в клиниках. Если у пациента нет доступа в интернет или он не может им воспользоваться, то в этом случае можно провести запись прямо на терминале клиники.

Электронная регистратура — система, которая позволяет упорядочить посещение врача и практически полностью исключить возможность пройти без очереди. Методик выдачи электронных талонов встречается две. Одна из них заключается в сквозном получении талонов, когда пациенты выписывают его по мере прихода в поликлинику. Другая имеет возможность указать дату и время посещения, естественно, выбрать можно только из того диапазона, что еще остался нерезервированным. [1]

Помимо прочего талоны можно не только выписывать, но и удалять, освобождая, таким образом, время посещения для других пациентов, изменять время и дату посещения, когда изменились личные планы. Когда имеется электронная запись, заказать повторно талон значительно проще, тем более врачу также доступна информация о записавшихся к нему на прием, что позволяет контролировать больных с хроническими заболеваниями.

Работа «Электронная регистратура» должна быть построена на принципе копирования информационных потоков, существующих в любом медицинском учреждении. Именно поэтому ее внедрение не должно требовать значительной реорганизации лечебно-диагностического процесса.

Фактически задача системы — замена бумаги и авторучки на клавиатуру и монитор компьютера. Для соблюдения этого принципа основной минимальной информационной единицей в системе принят электронный документ, который является основной электронной медицинской карты (ЭМК) пациента [13].

Основу электронной медицинской карты составляет документы «Осмотр врача». Эти документы предназначены для автоматизированного внесения в БД системы первичных и контрольных осмотров, включая осмотры врачей в поликлинике и стационаре, контрольные явки пациентов по диспансерному наблюдению, консультации.

Стоить обратить внимание, что основной и главной проблемой саморегистрации на прием является то, что иногда некоторые пациенты просто не приходят. Врач остается без дела на зарезервированное время. Это нарушает эффективность рабочего времени персонала.

Вместо того, чтобы оказывать медицинскую помощь, доктор просто ждет, когда к нему придет следующий пациент в назначенное время очереди.

## Обзор программ-аналогов российского производства

Для создания программного средства, необходимо изучить представленные аналоги и выделить их основные недостатки и преимущества, определить ключевые особенности тенденций в данном направлении.

Для того чтобы собрать максимально объективную информацию, был проведен мониторинг рынка программного обеспечения по теме «Электронная регистратура» [7].

В таблице 1 представлен перечень решений с наиболее часто встречающимися разработками.

Таблица 1

ПО «Электронная регистратура», применяемая на территории России

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Разработчик, регион | Название ПО, приложения | Адрес сайта |
| 1 | Комплексные медицинские информационные системы, Карелия | Электронная регистратура КМИС | https://www.kmis.ru/elektronnaia-registratura-3 |
| 2 | ООО Астрал-Софт, Калуга | Электронная регистратура ЛПУ Волгоградской области | https://registratura.volganet.ru/login |
| 3 | Парус Медиа, Самара | Электронная регистратура — запись на прием к врачу | https://www.regtalon.ru/schedule-adults/ |
| 4 | Торинс, Красноярск | Веб-регистратура | http://www.med-registratura.ru |
| 5 | ООО «Новые информационные технологии», Удмуртия | Электронная регистратура | https://newit-soft.ru/page/products/eregistry |

Рассмотрим каждое программное решение более подробно.

**«Электронная регистратура» компании КМИС (Карелия).** Разработана как дополнительная опция для Карельской медицинской информационной системы, но может поставляться и как отдельный автономный продукт. Электронная регистратура 3 включена в Реестр отечественного программного обеспечения; (Приказ Минкомсвязи России №426 от 06.09.2016 г., регистрационный номер ПО — 1550). Имеет Свидетельство о государственной регистрации программы для ЭВМ №2016615205 Федеральной службы по интеллектуальной собственности от 18 мая 2016 года. [5]

На рисунке 2 представлена начальная страница «Электронной регистратуры».

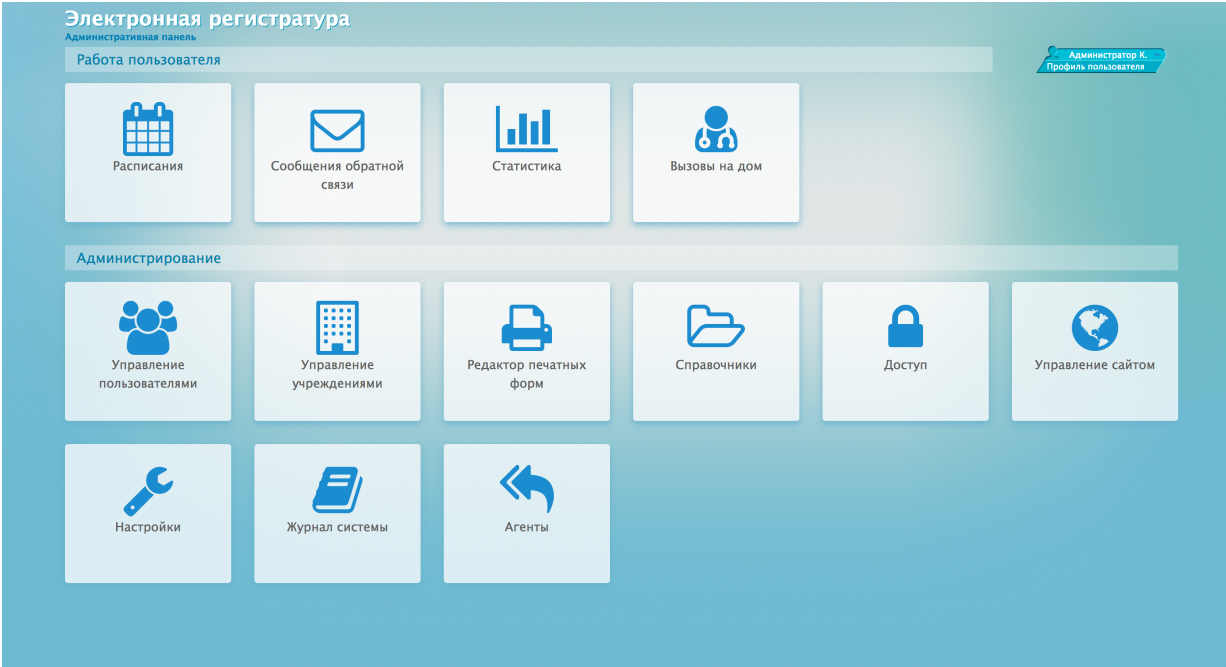


Рисунок 2. Начальная страница Электронной регистратуры КМИС

Поддерживаются все основные функции, обеспечена большая гибкость как в настройках, так и в оформлении. Встроенная интеграция с федеральными сервисами ЕГИСЗ позволяет пациентам производить запись через личный кабинет «Мое здоровье» на Едином портале государственных услуг (ЕПГУ) с авторизацией через Единую систему идентификации и аутентификации (ЕСИА).

Сервисы для пациентов:

* запись на прием к врачу;
* личный кабинет пациента;
* вызов врача на дом;
* заказ справок;
* справочная информация.

Дополнительно реализовано мобильное приложение, что позволяет не только записаться на прием, но и отслеживать статус заявки пациента через мобильный телефон или планшет. В системе реализована возможность уведомления пациентов по электронной почте и SMS.

Предусмотрены гибкие возможности администрирования: настройка внешнего вида, встроенная система управления содержимым сайта, настройка прав доступа с распределением прав по уровню доступа, ведение журнала событий. Формирование статистических отчетов, а также форма обратной связи с пациентами. Сильной стороной решения является кроссплатформенность (поддерживается работа сервера под управлением Windows, Linux, Unix, Mac–OS X и др).

Система внедрена в Ямало-Ненецком автономном округе, Ивановской и Воронежской областях.

**«Электронная регистратура» ЛПУ Волгоградской области, ООО «Астрал-Софт, Калуга.** Реализован личный кабинет пользователя. Для того чтобы записаться на прием к врачу, необходимо пройти процедуру авторизации с использованием логина и пароля Единой системы идентификации и аутентификации (ЕСИА). Если пациент не зарегистрированы на Портале Госуслуг, необходимо пройти регистрацию на странице https://esia.gosuslugi.ru/registration/ и подтвердить учетную запись в одном из Центров обслуживания. В случае отсутствия логина и пароля можно воспользоваться записью через Центр обработки вызовов. [6[1]]

На форме также представлены вкладки «Расписание» и «Поиск участка». Реализовано мобильное приложение. Сайт протестирован и поддерживает основные современные браузеры: Chrome, Safari, Internet Explorer 9+, Opera 11+ и Firefox6+.

Начальная страница портала представлена на рисунке 3.

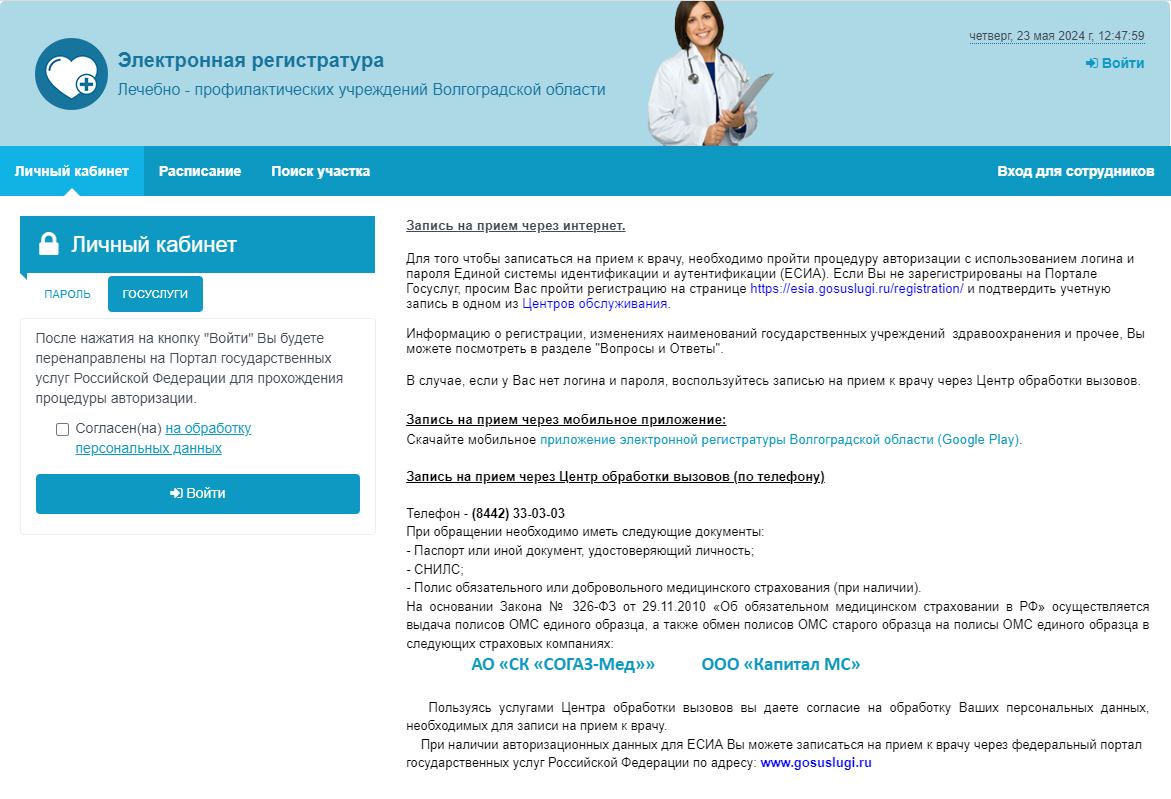
****

Рисунок 3. Начальная страница Электронная регистратура Волгоградской области

**АИС «Электронная регистратура» компании «Парус Медиа», Самара**. Поставляется как программно-аппаратный комплекс или как отдельное программное обеспечение. Поддерживается запись, как на платный, так и на бесплатный прием. Вся информация о выданных через Интернет или информационном терминале талонах в режиме реального времени появляется в информационной среде медучреждения.

Преимущества: универсальность решения, масштабируемость, настраиваемая функциональность (диагностика, диспансеризация, финансирование, платный прием, выбор сопутствующих услуг).

Отдельно стоит отметить поддержку режима записи авторизованным врачам сторонних лечебно-профилактических учреждений.

На рисунке 4 представлена начальная страница интернет-приложения.

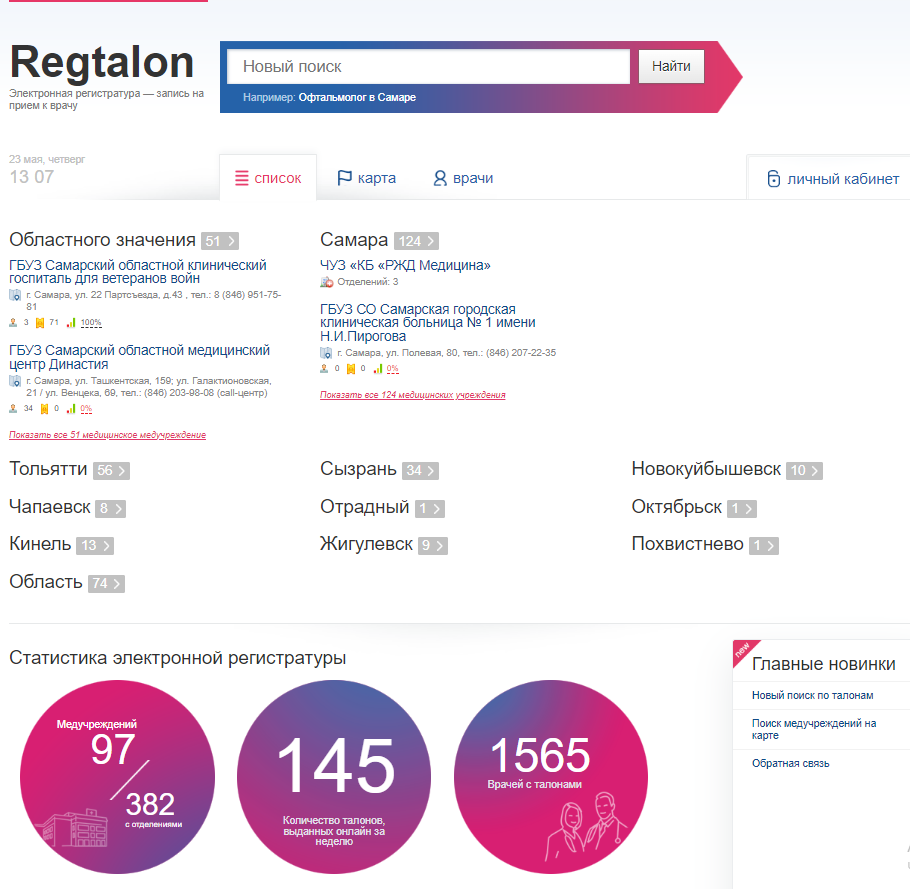


Рисунок 4. Начальная страница АИС «Электронная регистратура», Самара

**Веб-регистратура компании Торинс, Красноярск.** Программа предназначена для автоматизации работы регистраторов и управляющего персонала медучреждений.

Интерфейс приложения представлен на рисунке 5.

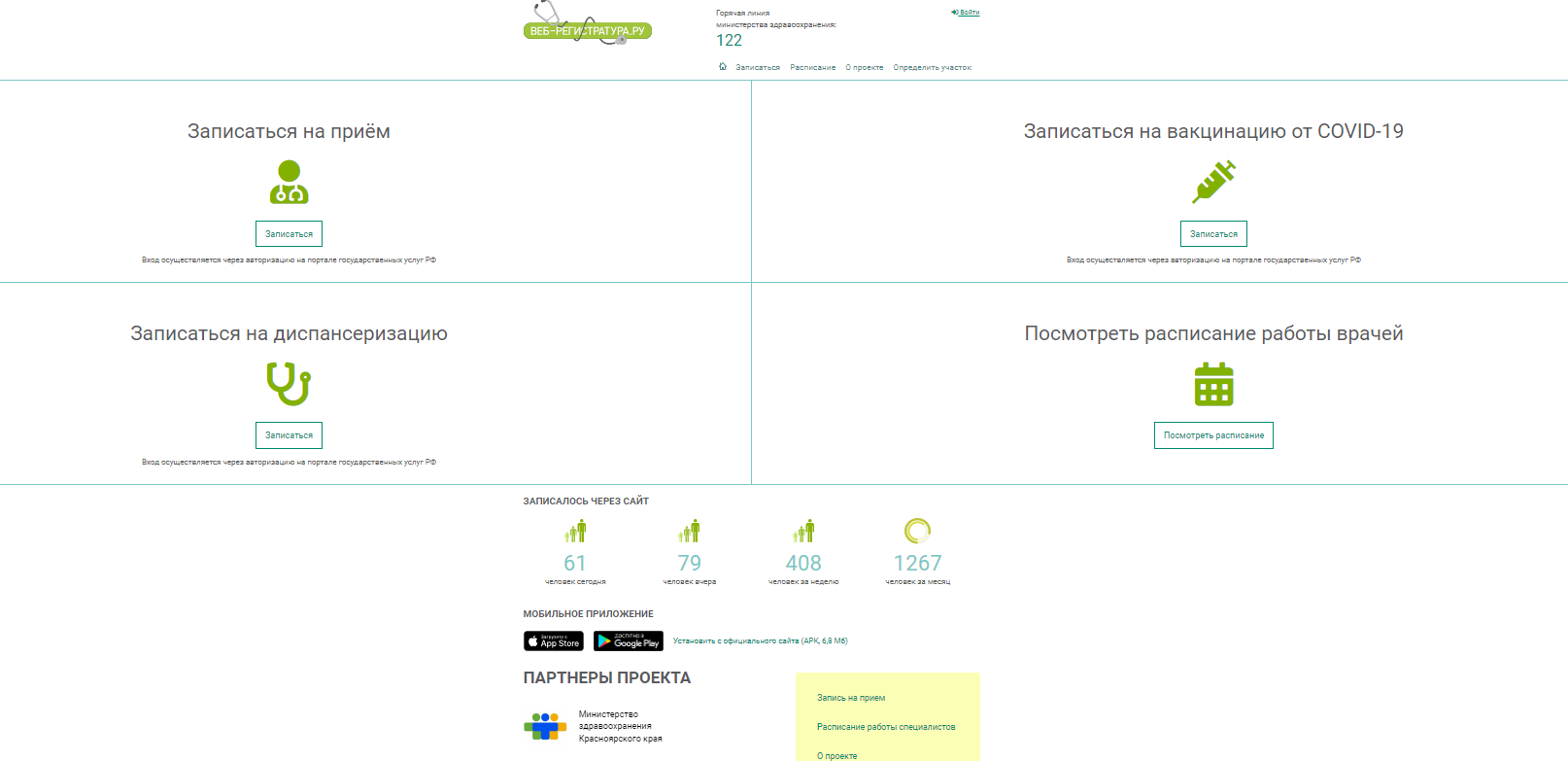


Рисунок 5. Интерфейс приложения Веб-регистратура

Приложение направлено на решение задач регулирования потоков амбулаторных больных, планирования деятельности врачей. Обеспечивает выполнение следующих функций: управление режимом и расписанием врачебных приемов, регистрация и контроль регистрации предварительной записи на прием к врачам, электронная очередь, информационный терминал. Сильными сторонами приложения являются: интуитивно-понятный и привлекательный интерфейс, гибкая работа с расписанием, работа с базой МУ в онлайн режиме, штрихкодирование документации, интеграция с платежными системами для терминалов.

**Электронная регистратура от ООО «Новые информационные технологии», Удмуртия.** Электронная система позволяет просматривать график работы всех врачей с учетом их выходных, праздников и отпусков, а также свободного времени на каждый день, когда возможно посещение. Предоставляет пользователю не только резервировать время, но и распечатать талон через терминал или домашний принтер, в случае, если подача заявки была через интернет-сайт.

На рисунке 6 представлен интерфейс приложения для записи на прием.



Рисунок 6. Запись на прием онлайн веб-приложения ООО НИТ

Для менеджера регистратуры реализованы функции ведения базы пациентов, добавление новых с созданием амбулаторной карты. В дальнейшем при работе врачей поликлиники пациенту можно создавать дополнительные карты — стоматологическую, диспансерную. Заполнение документа основано на применении открытых справочников.

В приложении реализован личный кабинет пациента, в котором помимо выписанных талонов доступна и электронная медицинская карта, где имеются записи врачей.

Анализ функциональных возможностей представленных решений позволяет сформировать типовой функционал, который должна поддерживать современная «Электронная регистратура»:

* создание и ведение «сетки» расписания врачей;
* возможность записи на прием с выбором свободных даты, времени и врача;
* поиск врача по ФИО, и по специальности;
* возможность регистрации пациента на сайте эр и авторизованного доступа пациента;
* возможность для пациента просмотреть записи, на которые он уже записался ранее («личный кабинет» пациента на основе персональных данных);
* расширенная поддержка идентификации пациента по ФИО, полису ОМС;
* формирование талона и его печать;
* возможность отмены записи;
* напоминание пациенту о необходимости прийти на прием (e-mail, sms).

**Краткие выводы по 1 главе**

1. Современные коммуникационные технологии в медицине предоставляют уникальную возможность дистанционно получить всю необходимую информацию не только о той или иной больнице или поликлинике, но и специалистах, которые ведут прием, а также оставить на виртуальном сервисе свой отзыв о полученных медицинских услугах. Последнее, в свою очередь, помогает улучшить качество обслуживания каждого клиента, что также не может не сказать на общем уровне работы медицинского учреждения [6].
2. Основные преимущества онлайн регистратуры: запись через веб-регистратуру экономит время — не нужно стоять в очереди; благодаря онлайн регистратуре можно записаться в поликлинику в любом регионе России, будучи дома, на работе или даже стоя в пробке.
3. Разрабатываемый программный продукт не является уникальной разработкой на данный момент времени. Существует множество аналогов. Но каждый программный продукт создается непосредственно для решения определенного количества задач. Очень сложно найти универсальный программный продукт, который решал бы все необходимые задачи. Нецелесообразно приобретать дорогостоящий программный продукт (стоимость такого программного продукта от 20000 и выше), если в нем использовать только какую-то часть, а остальное останется непригодным. Поэтому желательно иметь программу, реализующую необходимую группу задач.

# 2. ПРОЕКТИРОВАНИЕ АВТОМАТИЗАЦИИ ЗАПИСИ НА ПРИЁМ

## Методологии проектирования программного продукта

При разработке сайтов, приложений и прочих программных продуктов используется достаточно много отличных методологий, такие как: [2]

**SADT** (акроним от англ. Structured analysis and design technique) — методология структурного анализа и проектирования, интегрирующая процесс моделирования, управление конфигурацией проекта, использование дополнительных языковых средств и руководство проектом со своим графическим языком.

**IDEF0** — методология функционального моделирования (англ. *Function modeling*) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временна́я последовательность (поток работ).

**UML**(англ. Unified Modeling Language— унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения, для моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

**ER-модель** (от англ. Entity-Relationship model, *модель «сущность — связь»*) — модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями. [3]

**DFD** — общепринятое сокращение от англ. *Data flow diagrams* — диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

## Разработка функциональной модели приложения с помощью методологии IDEF0 «AS-IS»

На рисунке 7 представлена диаграмма основного процесса «Формирование записи на прием» IDEF0 «Как есть».

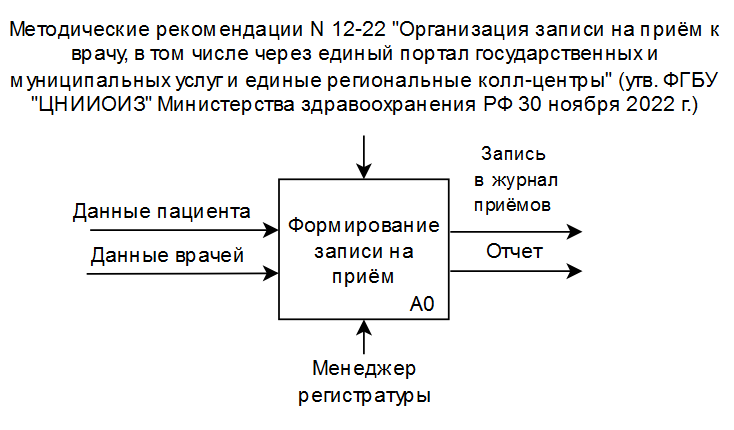


Рисунок 7. Главный процесс IDEF0 As–Is

На диаграмме IDEF0 As–Is, согласно нотации, отображаются:

* входные данные, необходимые для данного процесса: данные пациента и врачей;
* управление: Методические рекомендации N 12-22 «Организация записи на приём к врачу, в том числе через единый портал государственных и муниципальных услуг и единые региональные колл-центры» (утв. ФГБУ "ЦНИИОИЗ" Министерства здравоохранения РФ 30 ноября 2022 г.)
* механизмы, т.е. исполнители и средства для выполнения процесса: Менеджер регистратуры.
* выходные данные: Запись на приём и Отчет.

В дереве процессов главный процесс разделяется на 4 процесса. Процесс А3 декомпозируется на 3 подпроцесса, что представлено на рисунке 8.

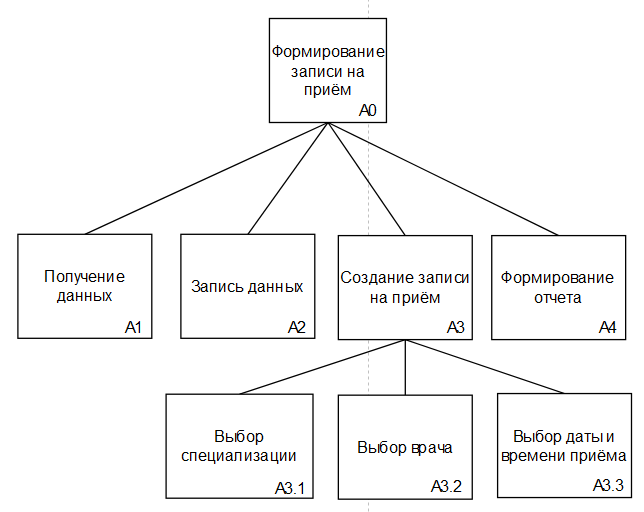


Рисунок 8. Дерево процессов IDEF0 As–Is

На рисунке 9 представлена диаграмма IDEF0 As–Is 1-ый уровень декомпозиции процессов.

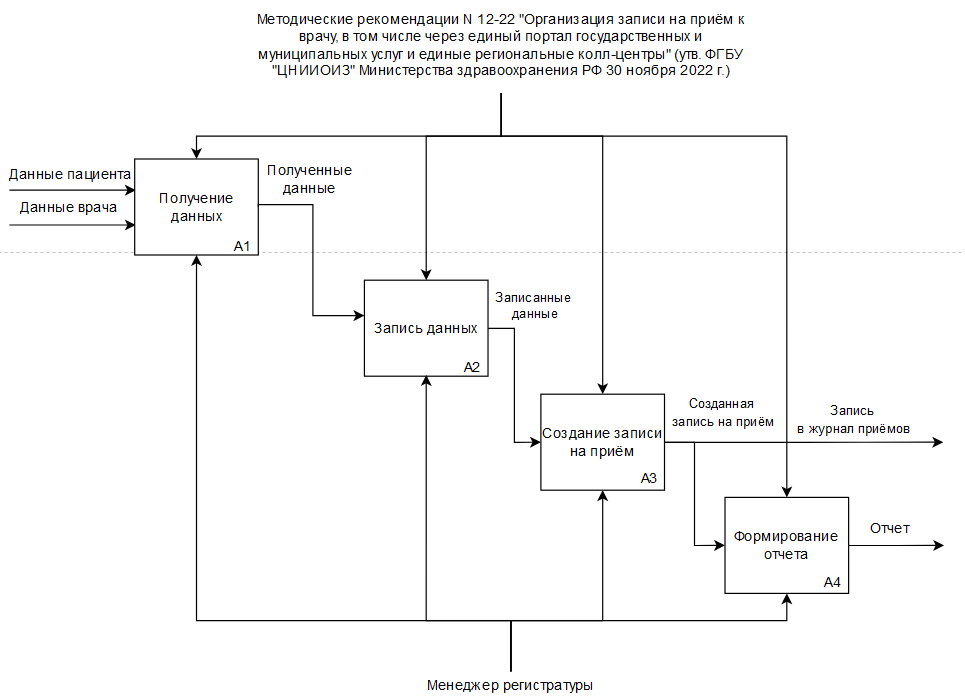


Рисунок 9. Декомпозиция главного процесса IDEF0 As–Is

На рисунке 10 представлена диаграмма IDEF0 As–Is подпроцесса 2-го уровня, а именно процесса создания записи на прием.

При обращении в поликлинику пациент представляет необходимые документы для поиска амбулаторной карты и создания лицевого счета. Менеджер регистратуры записывает пациента на свободную дату и время согласно расписанию врача, создает талон на запись.

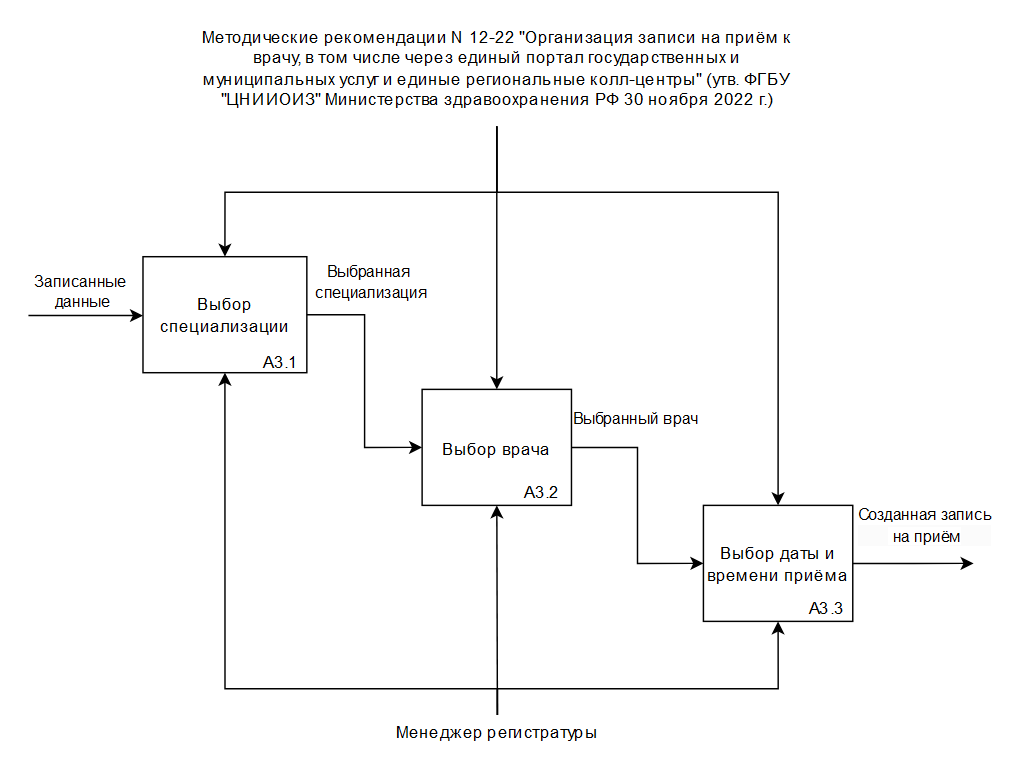


Рисунок 10. Декомпозиция подпроцесса A3 «Создание записи на приём» IDEF0 As–Is

Аналогично опишем функциональную модель IDEF0 To–Be — того, к чему нужно стремиться.

Отличие диаграммы IDEF0 To–Be заключается в том, что в диаграмме IDEF0 As–Is не используется сам программный продукт и его составляющие для функционирования. В главном процессе IDEF0 To–Be участвует новый механизм — «Веб-приложение», то есть, показан процесс с использованием продукта автоматизации, а также появилась запись в базу данных.

На рисунке 11 изображена диаграмма IDEF0 To–Be процесса «Формирование записи на прием».

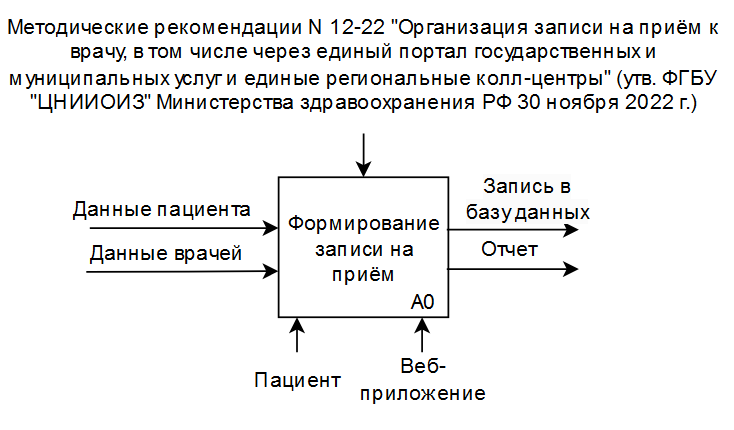


Рисунок 11. Главный процесс IDEF0 To–Be

После автоматизации процесса, дерево процессов IDEF0 To–Be (рис. 12) остается неизменным.

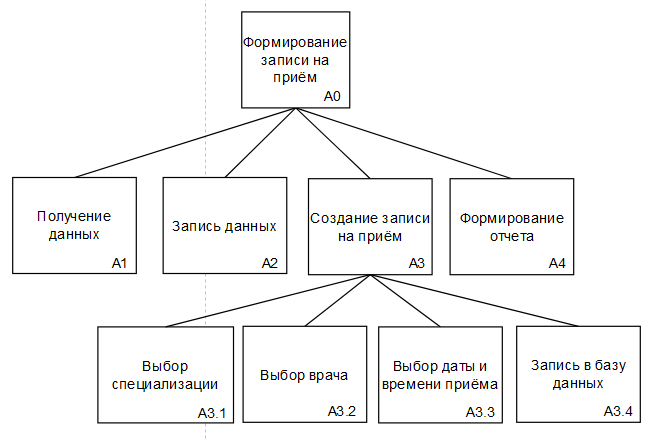


Рисунок 12. Дерево процессов IDEF0 To–Be

В декомпозиции главного процесса IDEF0 To–Be механизм «Веб-приложение» участвует в таких процессах, как: «Получение данных», «Запись данных» и «Формирование отчета». Механизм «Пациент» участвует в процессах «Получение данных» и «Создание записи на приём».

Построение диаграммы осуществляется идентично построению диаграммы IDEF0 As–Is, но уже с учетом автоматизации процесса. Декомпозиция 1-го уровня изображена на рисунке 13.

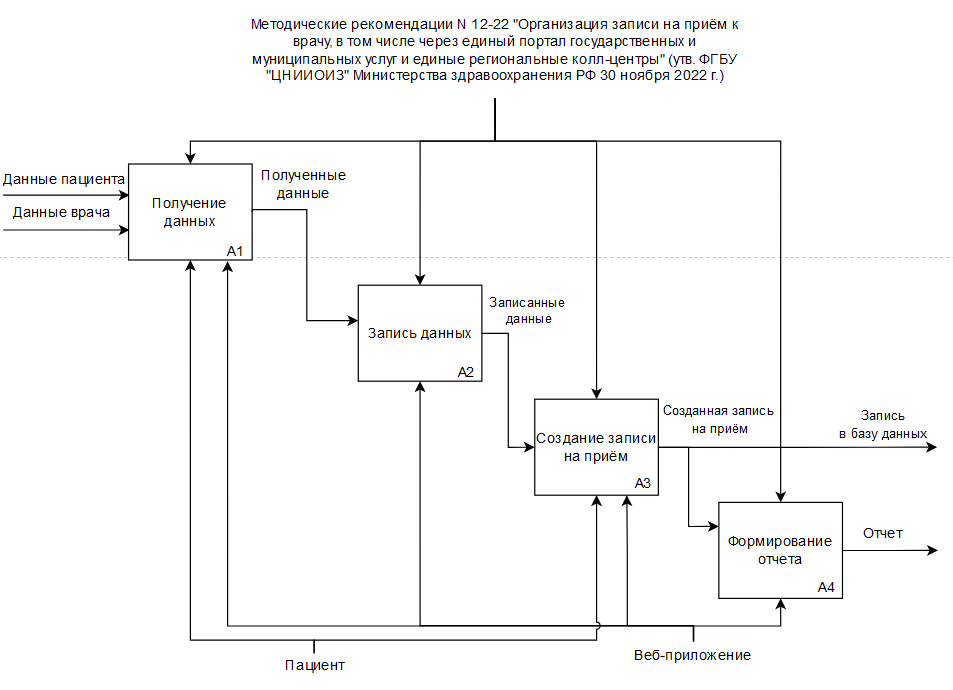


Рисунок 13.Декомпозиция главного процесса IDEF0 «TO-BE»

Декомпозиция 2-го уровня происходит по алгоритму: 1-ый уровень — декомпозиция главного процесса, 2-ой уровень — декомпозиция 1-ого уровня.

На этом этапе пациент сам создает запись к врачу. После того, как будут внесены все необходимые данные, пациент в приложении сможет выбрать специализацию, врача, дату и время. Результатом работы станет электронная запись на прием к врачу.

Декомпозиция 2-го уровня представлена на рисунке 14.

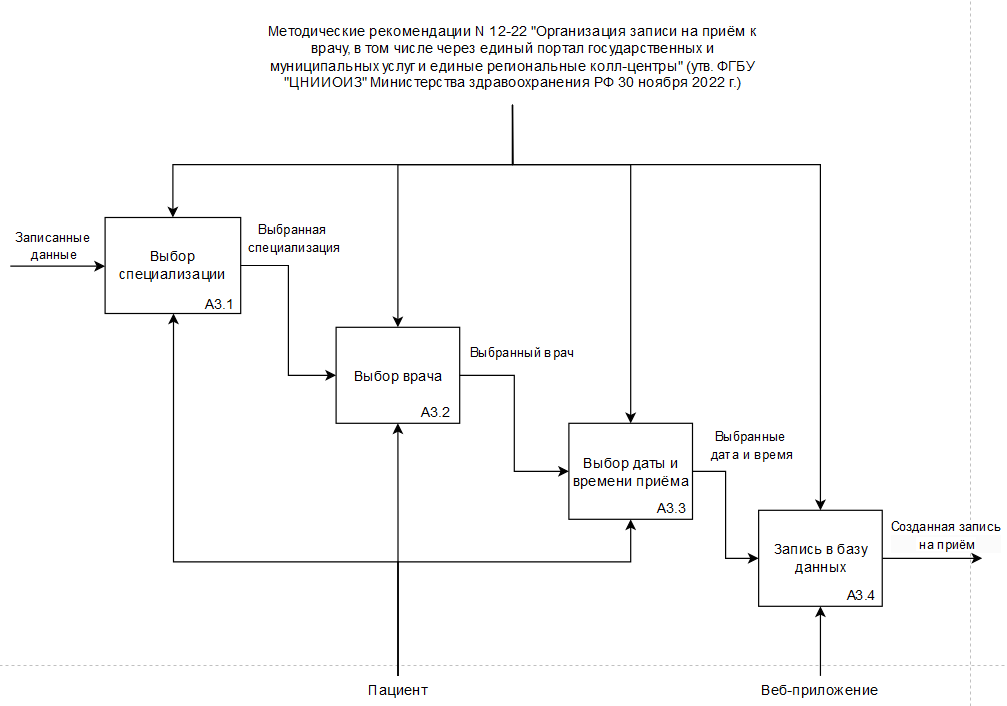


Рисунок 14.Декомпозиция процесса A3 «Создание записи на приём» IDEF0 To–Be

## Диаграмма потоков данных

Диаграмма потоков данных позволяет описать движение и преобразование данных между внешними сущностями, хранилищами и процессами. При этом выходная информация одной внешней сущности или процесса является входной для другого процесса или сущности.

Для того, чтобы продемонстрировать, как каждый подпроцесс бизнес-процесса «Формирование записи на прием» преобразует свои входные данные в выходные, построена диаграмма в нотации DFD (рисунок 15 и 16 соответственно).

На диаграмме показаны следующие процессы: регистрация пациента, авторизация, как пациента, так и менеджера регистратуры, ввод первичных параметров для создания записи на прием, создание записи на прием путем выбора врача, даты и времени, оформление отчета.

Сущностями являются Пациент, Запись на приём, Менеджер регистратуры. В качестве хранилища выступает база данных (БД).

Информационными потоками выступают данные для авторизации, как менеджера, так и зарегистрированного ранее пациента, личные данные пациента для записи на прием, запрос менеджера на создание записи (указание специализации, врача и свободных даты и времени), данные для формирования отчетов.

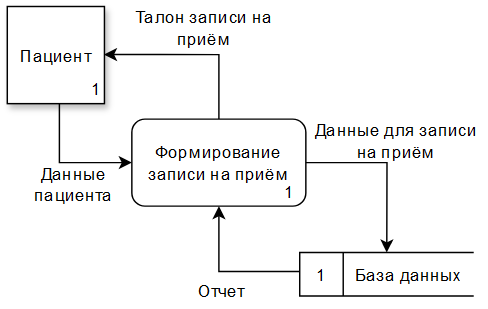


Рисунок 15. Диаграмма DFD «Формирование записи на прием»

Декомпозиция бизнес-процесса «Формирование записи на прием» представлена на рисунке 16

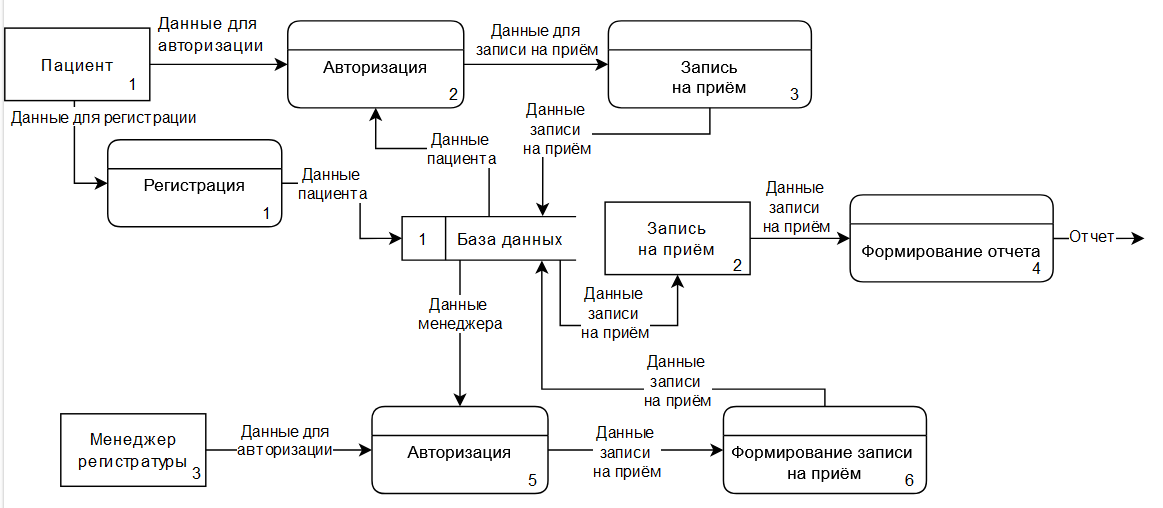


Рисунок 16.Декомпозиция диаграммы DFD «Формирование записи на прием»

## Проектирование структуры данных

На основе анализа деятельности структурного подразделения «Регистратура» ГБУЗ «Калачевская ЦРБ» и выявленных требований к разрабатываемому продукту была составлена структура данных (рисунок 17). Для построения структуры данных был использован язык ER–диаграмм.



Рисунок 17. Структура данных веб-приложения

Структура данных проектируемого веб-приложения включает в себя пять таблиц: «doctors», «specializations», «schedules», «homeCalls», «patients».

Информация о сущностях и их атрибутах представлена в таблице 2.

Таблица 2

Описание сущностей и атрибутов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Наименование сущности | Атрибуты | Тип и длина поля | Назначение |
| 1 | «doctors» | doctorId | Int, (15) | уникальный идентификатор доктора |
| name | varchar | имя врача |
| surname | varchar | фамилия врача |
| patronymic | varchar | отчество врача |
| specializationId | Int, (15) | идентификатор специализации |
| 2 | «specializations» | specializationId | Int, (15) | уникальный идентификатор специализации |
| specializationTitle | varchar | наименование специализации |
| 3 | «schedules» | scheduleId | Int, (15) | уникальный идентификатор времени приёма |
| doctorId | Int, (15) | идентификатор доктора |
| datetimeOfReception | datetime | дата и время приёма |
| patinentId | Int, (15) | идентификатор пользователя |
| 4 | «homeCalls» | homeCallId | Int, (15) | уникальный идентификатор записи о вызове врача на дом |
| dateTimeOfCall | datetime | дата и время вызова |
| reasonOfCall | varchar | причина вызова |
| status | varchar | статус вызова |
| 5 | «patients» | patientId | Int, (15) | уникальный идентификатор пациента |
| login | varchar | логин пациента |
| password | varchar | пароль пациента |
| email | varchar | электронная почта пациента |
| name | varchar | имя |
| surname | varchar | фамилия |
| patronymic | varchar | отчество |
| gender | varchar | пол |
| birthdayDate | varchar | день рождения |
| permanentResidence | varchar | постоянное местожительства |
| actualResidence | varchar | фактическое место жительства |
| passport | varchar | паспортные данные |
| mio | varchar | страховая медицинская организация |
| policyCMI | varchar | полис обязательного медицинского страхования (ОМС) |
| policyPIP | varchar | страховой номер(полис) индивидуального лицевого счета (СНИЛС) |
| disability | varchar | инвалидность |
| phoneNumber | varchar | номер телефона |
| job | varchar | должность |

Типы связей между сущностями указаны в таблице 3.

Таблица 3

Спецификация связей

|  |  |  |
| --- | --- | --- |
| **Связь** | **Объект** | **Отношение** |
| Доктора | Время приёма | 1:M |
| Специализации | Доктора | 1:M |
| Пациенты | Записи на приём | 1:M |

## Разработка UML диаграмм

Унифицированный язык моделирования (UML) является стандартным инструментом для создания «чертежей» программного обеспечения. С помощью UML можно визуализировать, специфицировать, конструировать и документировать артефакты программных систем. UML пригоден для моделирования любых систем: от информационных систем масштаба предприятия до распределенных веб-приложений и даже встроенных систем реального времени [11].

При проектировании приложения были построены диаграммы поведения: Use Case диаграмма (диаграмма прецедентов), Activity диаграмма (диаграмма деятельности) и Sequence диаграммы (диаграммы последовательности) [2].

На диаграмме вариантов использования (Use case diagram) отображается взаимодействие между вариантами использования, представляющими функции системы, и действующими лицами, представляющими людей или системы, получающие или передающие информацию в данную систему. Из диаграмм вариантов использования можно получить довольно много информации о системе. Этот тип диаграмм описывает общую функциональность системы. Пользователи, менеджеры проектов, аналитики, разработчики, специалисты по контролю качества и все, кого интересует система в целом, могут, изучая диаграммы вариантов использования, понять, что система должна делать [4].

Диаграмма деятельности (Activity diagram) представляют переходы потока управления между объектами от одной деятельности к другой внутри системы. На диаграмме деятельности отображается логика или последовательность перехода от одной деятельности к другой, при этом внимание фиксируется на результате деятельности. Сам же результат может привести к изменению состояния системы или возвращению некоторого значения.

Диаграммы последовательностей (Sequence diagram) являются частным случаем диаграмм взаимодействия. Диаграммы последовательности отражают поток событий, происходящих в рамках варианта использования. На этих диаграммах изображаются только те объекты, которые непосредственно участвуют во взаимодействии т.к. ключевым моментом является именно динамика взаимодействия объектов во времени и не используются возможные статические ассоциации с другими объектами.

Остановимся на каждой из них более подробно.

## Диаграмма прецедентов

На рисунке 18 представлена диаграмма прецедентов. В системе «Веб приложение «Электронная регистратура»» участвуют две роли — «Пациент» и «Менеджер регистратуры».

Роль «Пациент» обладает следующими прецедентами: «Регистрация», «Авторизация», «Запись на приём», «Просмотр записей на приём», «Отмена записи на приём», «Редактирование профиля» и «Вызов врача на дом».

Роль «Менеджер регистратуры» обладает следующими прецедентами:

* «Авторизация»;
* «Работа с картотекой пациентов». Прецедент включает: «Создание профиля пациента», «Поиск профиля пациента», «Редактирование профиля пациента»;
* «Регистрация медицинского работника»;
* «Ведение расписания работы медицинского работника»;
* «Перераспределение выданных талонов на приём»;
* «Работа с записями на приём» включает в себя: «Создание записи на приём», «Поиск записей на приём», «Отмена записей на приём»;
* «Формирование отчетно-учетной документации».

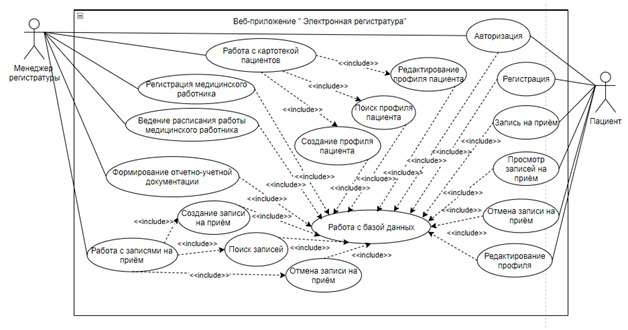


Рисунок 18. Диаграмма прецедентов

## Диаграммы активностей

Как уже было сказано ранее, диаграмма активности показывает, как происходит взаимодействие пользователя с программой и программы с базой данных.

Ниже приведены основные примеры взаимодействия менеджера регистратуры и пациента с программой и взаимодействие программы с базой данных: добавление записи в базу данных;

* просмотр записей, которые находятся в базе данных;
* изменение записи, которая находиться в базе данных;
* удаление записи из базы данных;
* формирование договора (экспорт значений из базы данных в MS Word);
* запрос к программе на выполнение функции кликом по определённой кнопке;
* прохождение валидации. Если валидация не пройдена, то появляется окно об ошибке и предлагается способ её решения, в противном случае происходит переход к формированию программой SQL запроса к базе данных;
* формирование программой SQL запроса к базе данных;
* выполнение SQL запроса;
* просмотр менеджером регистратуры записей, находящихся в базе данных, проходит следующим образом:

1. в главном меню выбрать пункт «Работа с докторами», после чего менеджер регистратуры попадет на страницу с таблицей, содержащей данные докторов.
2. в главном меню выбрать пункт «Работа с пациентами», после чего менеджер регистратуры попадет на страницу с таблицей, содержащей данные пациентов.

* просмотр пациентом записей, находящихся в базе данных, проходит следующим образом: в шапке страницы выбрать пункт «Профиль», после чего пациент попадет в свой профиль, в котором пациент может просмотреть свои личные данные, а также информацию о враче, к которому он записан на приём.

На рисунках 19 — 20 продемонстрированы примеры взаимодействия пациента и менеджера регистратуры соответственно.

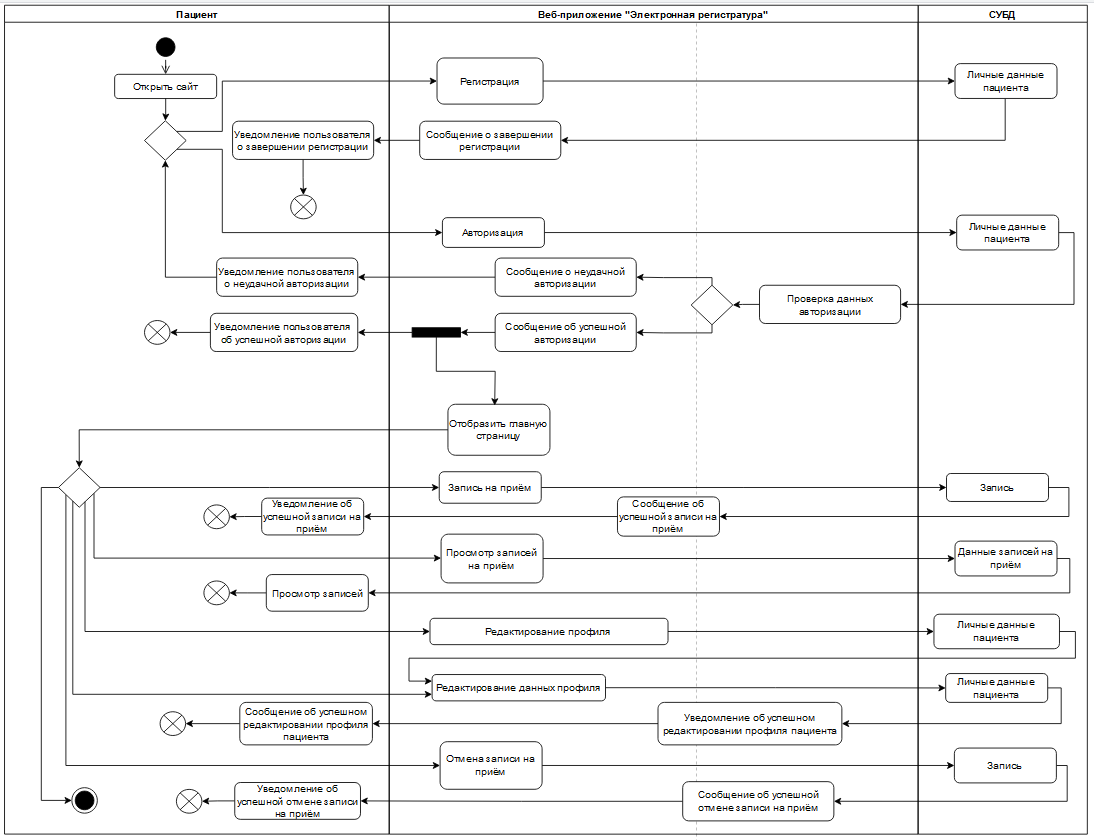


Рисунок 19. Диаграмма активностей, роль — «Пациент»

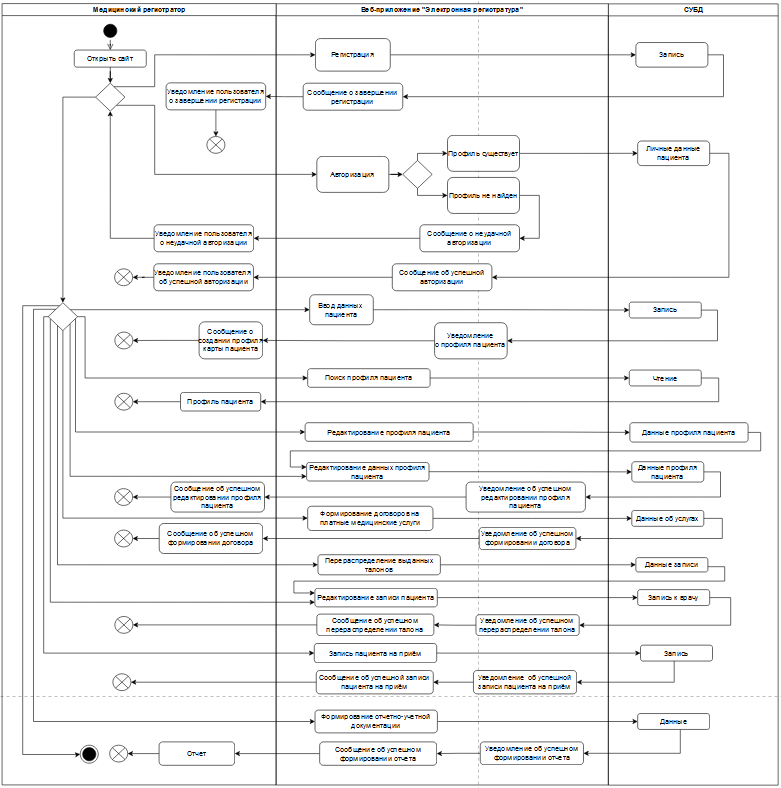


Рисунок 20. Диаграмма активностей, роль — «Менеджер регистратуры»

## Диаграммы последовательности

Диаграммы последовательности отражают поток событий, происходящих в рамках варианта использования. [2]

Диаграмма, представленная на рисунке 21, отображает поток событий в рамках варианта использования «Авторизация менеджера в системе».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Авторизация». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (менеджер регистратуры) и объектом.

Вариант использования начинается, когда менеджер регистратуры открывает форму авторизации и заполняет ее данными. После этого происходит процесс аутентификации пользователя.

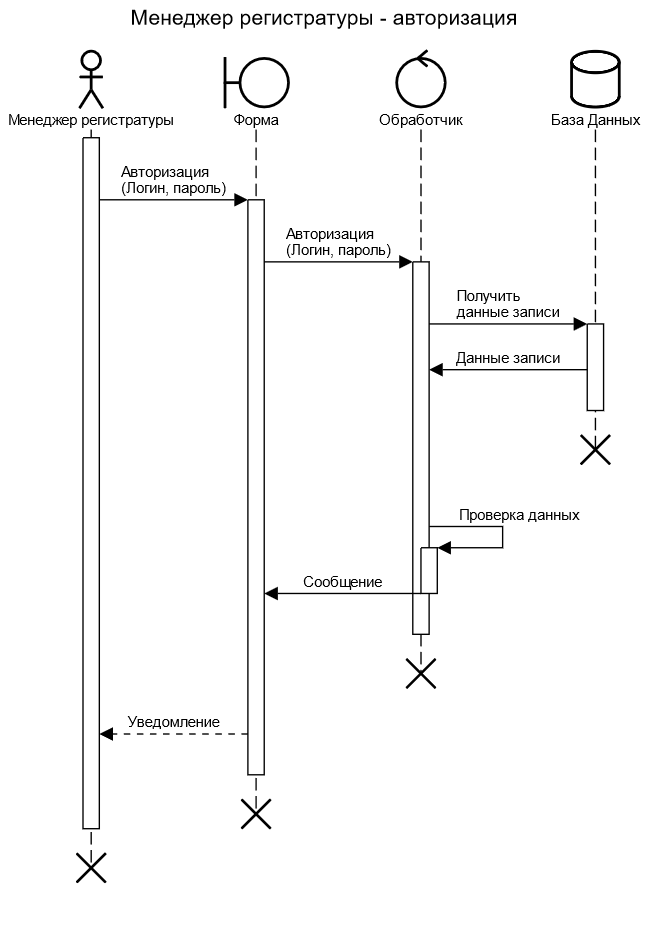


Рисунок 21. Последовательность действий при авторизации менеджера

Диаграмма, представленная на рисунке 22, отображает поток событий в рамках варианта использования «Создание профиля пациента менеджером регистратуры».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Создание профиля пациента». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (менеджер регистратуры) и объектом.

Вариант использования начинается, когда менеджер регистратуры открывает форму создания профиля и заполняет ее данными. После этого происходит процесс создания записи в базе данных.

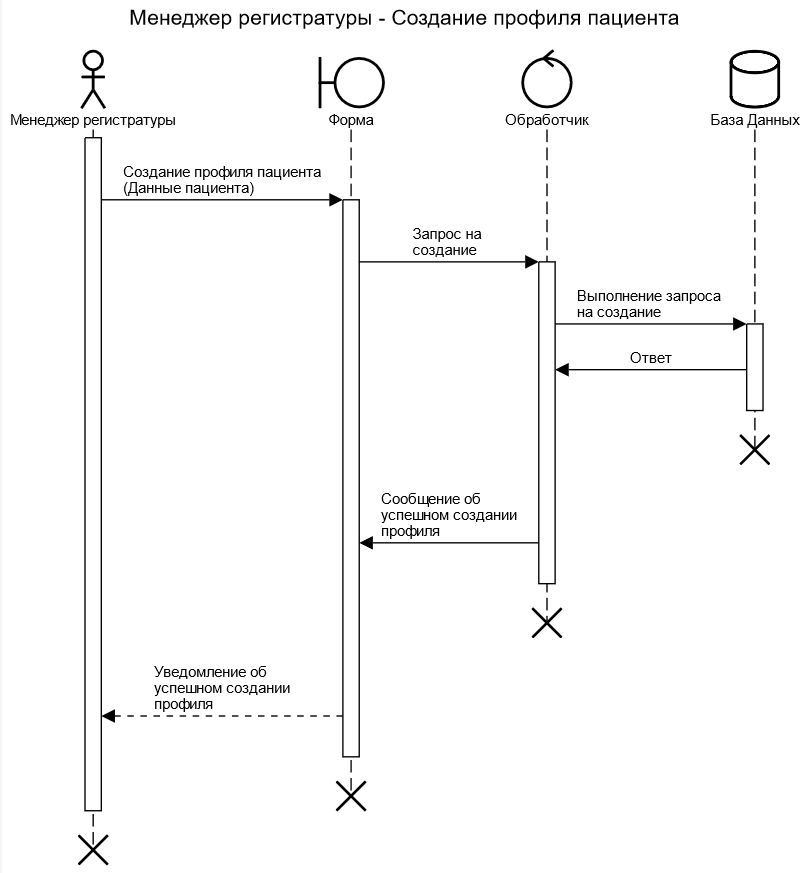


Рисунок 22. Последовательность действий при создании профиля пациента менеджером регистратуры

Диаграмма, представленная на рисунке 23, отображает поток событий в рамках варианта использования «Поиск профиля пациента менеджером регистратуры».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Поиск профиля пациента». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (менеджер регистратуры) и объектом.

Вариант использования начинается, когда менеджер регистратуры открывает форму и вводит личные данные пациента. После этого менеджеру предоставляются данные.

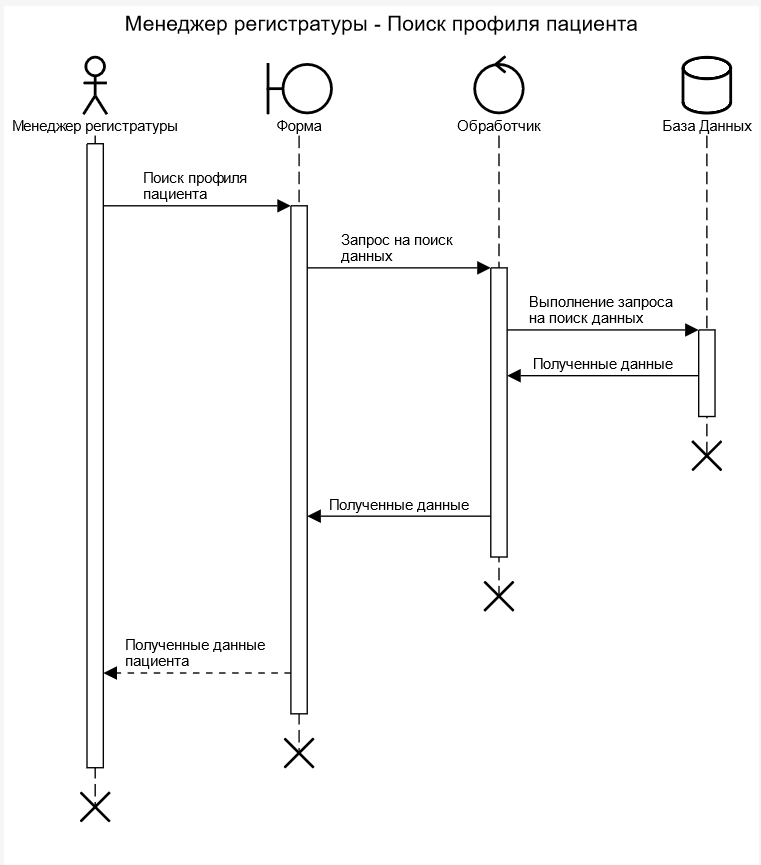


Рисунок 23. Последовательность действий при поиске профиля пациента

Диаграмма, представленная на рисунке 24, отображает поток событий в рамках варианта использования «Редактирование профиля пациента менеджером регистратуры».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Редактирование профиля пациента». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (менеджер регистратуры) и объектом.

Вариант использования начинается, когда менеджер регистратуры открывает форму и выбирает необходимый профиль пациента. После этого менеджер редактирует данные.

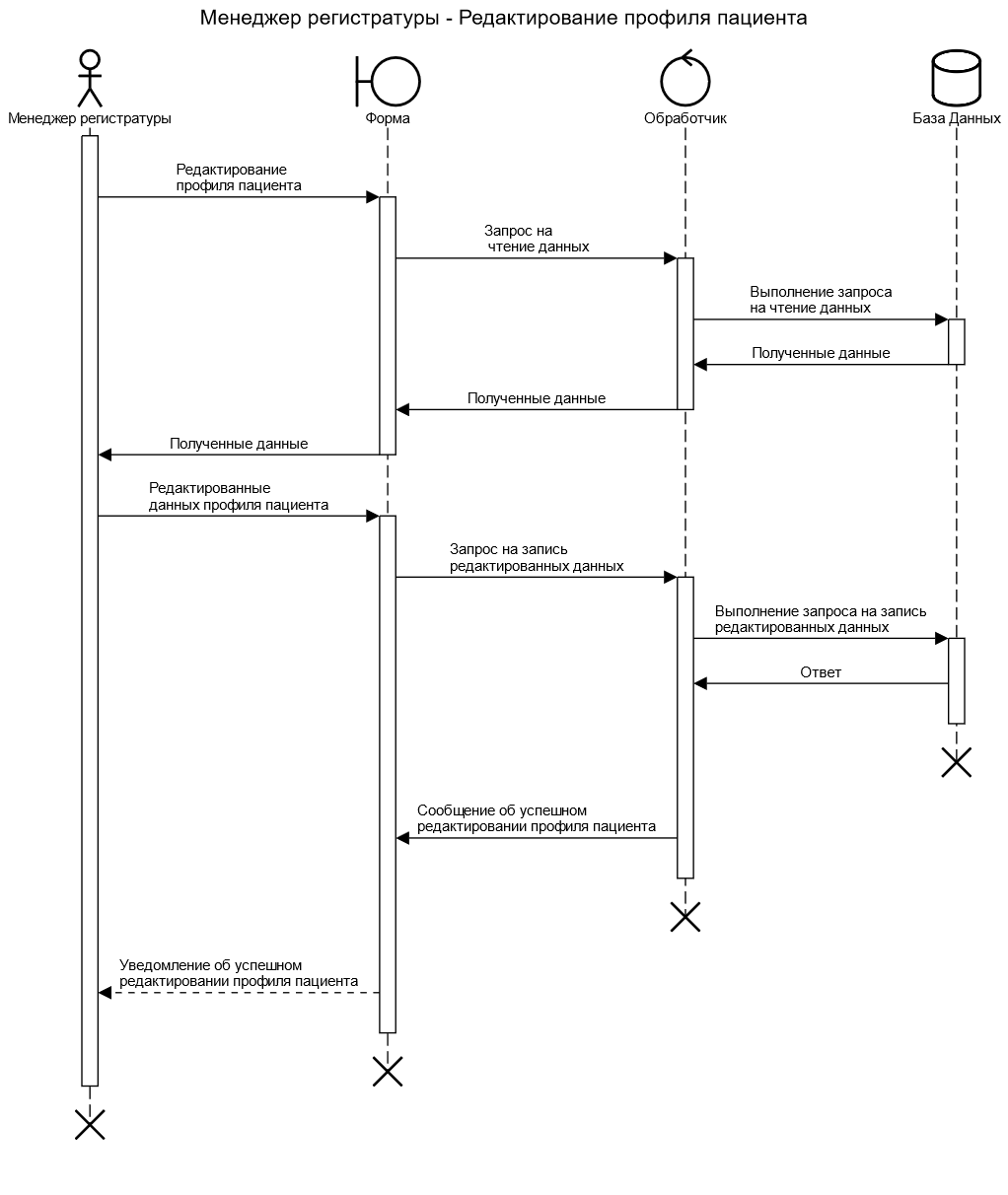


Рисунок 24. Последовательность действий при редактировании профиля пациента менеджером регистратуры

Диаграмма, представленная на рисунке 25, отображает поток событий в рамках варианта использования «Запись пациента на приём менеджером».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Запись на приём». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (менеджер регистратуры) и объектом.

Вариант использования начинается, когда менеджер регистратуры открывает форму записи на приём. После этого он записывает пациента на приём, выбирая необходимого доктора, дату и время.

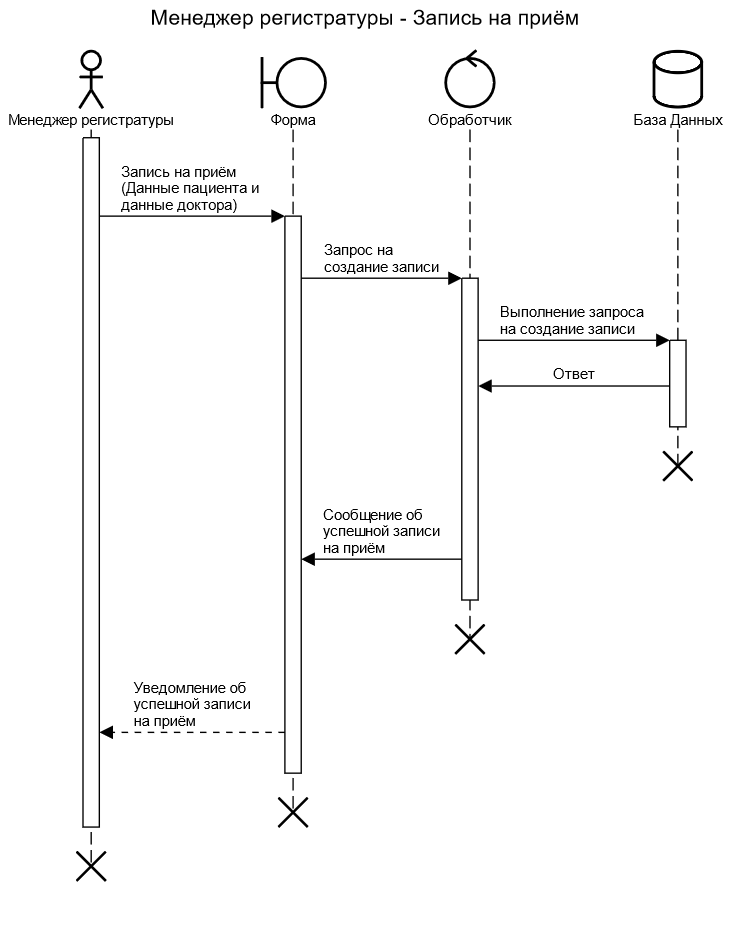


Рисунок 25. Последовательность действий менеджера при записи пациента на приём

Диаграмма, представленная на рисунке 26, отображает поток событий в рамках варианта использования «Регистрация вызова врача на дом».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Авторизация». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (менеджер регистратуры) и объектом.

Вариант использования начинается, когда менеджер регистратуры открывает форму регистрации вызова и заполняет ее данными. После этого происходит процесс аутентификации пользователя.

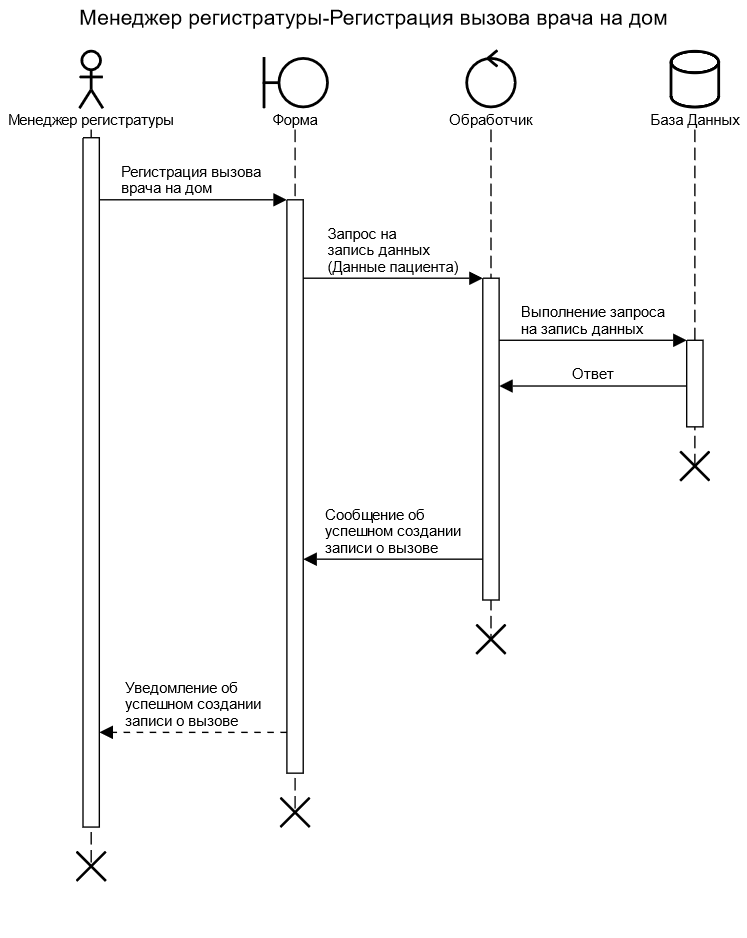


Рисунок 26. Последовательность действий при регистрации вызова врача на дом

Аналогичным образом были проанализированы прецеденты и составлены диаграммы последовательности роли «Пациент».

Диаграмма, представленная на рисунке 27, отображает поток событий в рамках варианта использования «Регистрация пациента в системе».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Регистрация». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (пациент) и объектом.

Вариант использования начинается, когда пациент открывает форму регистрации и заполняет ее данными. После этого происходит процесс внесения данных пользователя.

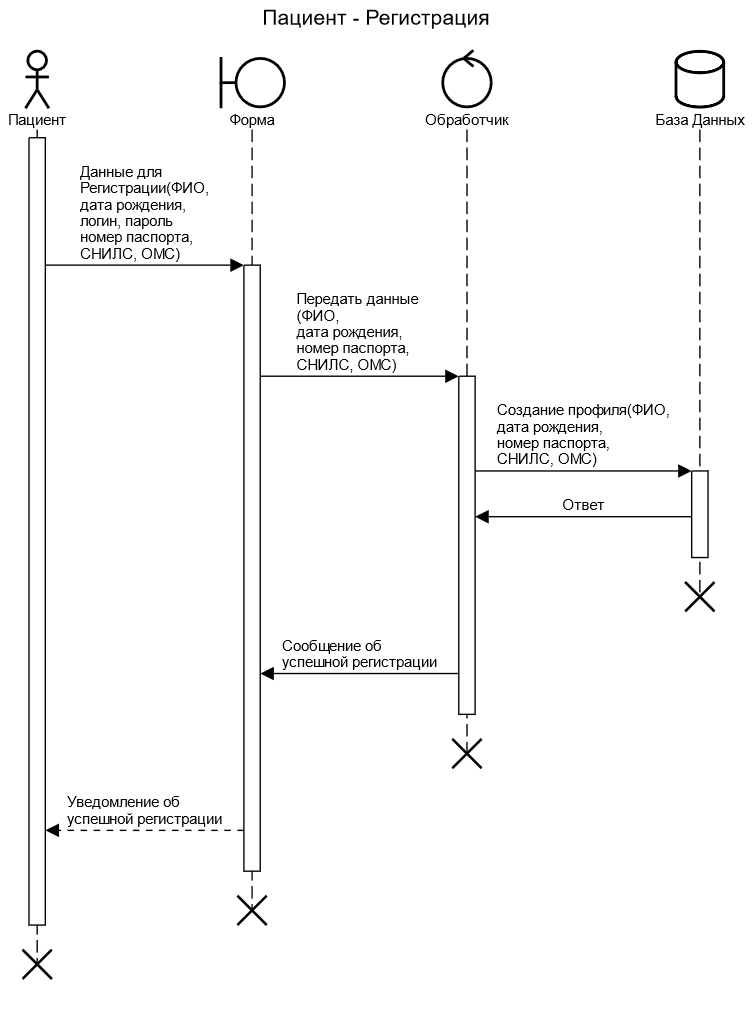


Рисунок 27.Последовательность действий при регистрации

Диаграмма, представленная на рисунке 28, отображает поток событий в рамках варианта использования «Авторизация пациента в системе».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Авторизация». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (пациент) и объектом.

Вариант использования начинается, когда пациент открывает форму авторизации и заполняет ее данными. После этого происходит процесс аутентификации пользователя.

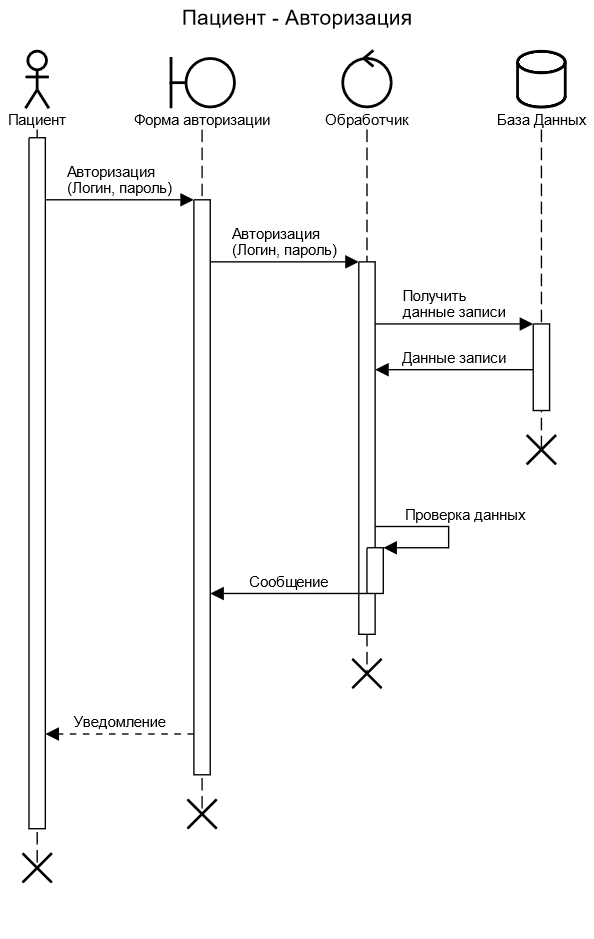


Рисунок 28. Последовательность действий при авторизации

Диаграмма, представленная на рисунке 29, отображает поток событий в рамках варианта использования «Запись на приём пациентом».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Авторизация». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (пациент) и объектом.

Вариант использования начинается, когда пациент открывает форму записи на приём и выбирает необходимого врача, дату и время приёма. После этого происходит процесс записи пациента на приём.

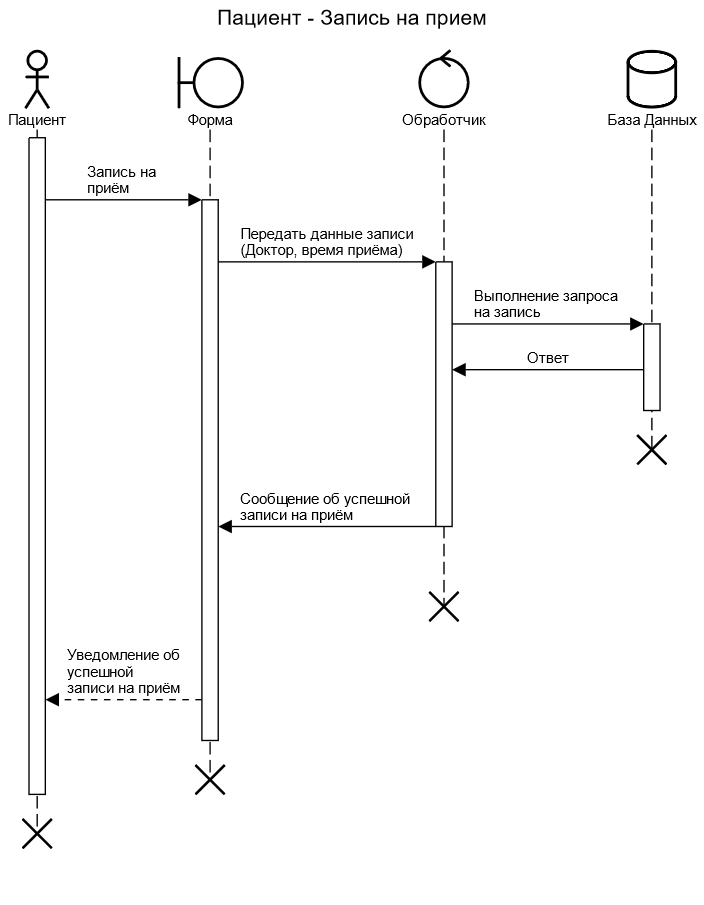


Рисунок 29. Последовательность действий пациента при записи на прием

Диаграмма, представленная на рисунке 30, отображает поток событий в рамках варианта использования «Просмотр пациентом записей на приём».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Просмотр записей на приём». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (пациент) и объектом.

Вариант использования начинается, когда пациент открывает свой профиль. После этого пользователь может просмотреть свои записи на приём.

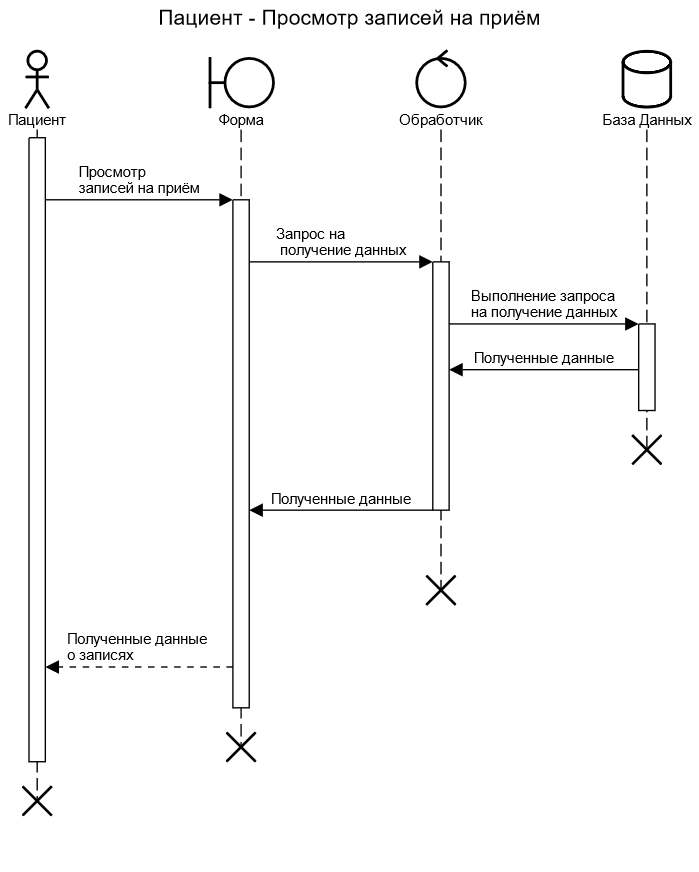


Рисунок 30. Последовательность действий при просмотре записей на приём

Диаграмма, представленная на рисунке 31, отображает поток событий в рамках варианта использования «Отмена пациентом записи на приём».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Отмена записи на приём». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (пациент) и объектом.

Вариант использования начинается, когда пациент открывает свой профиль. После этого пользователь может просмотреть свои записи на приём, и отменить выбранную запись на приём.



Рисунок 31.Последовательность действий при отмене записи на прием

Диаграмма, представленная на рисунке 32, отображает поток событий в рамках варианта использования «Редактирование профиля пациентом».

В верхней части диаграммы показаны все действующие лица и объекты, требуемые системе для выполнения варианта использования «Редактирование профиля». Стрелки соответствуют сообщениям, передаваемым между действующим лицом (пациент) и объектом.

Вариант использования начинается, когда пациент открывает свой профиль. После этого пользователь может просмотреть свои личные данные и изменить их.

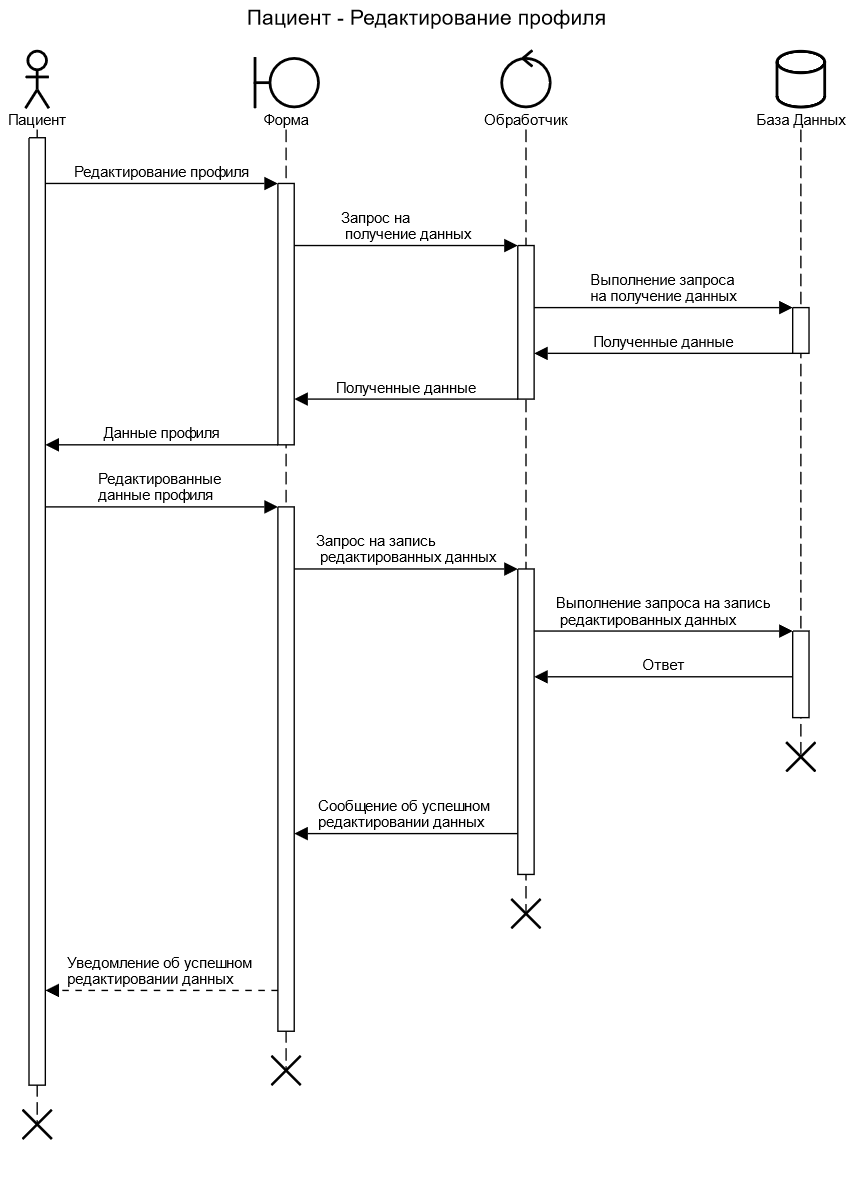


Рисунок 32. Последовательность действий при редактировании профиля

## Проектирование пользовательского интерфейса

Интерфейс пользователя (UI – англ. user interface – пользовательский интерфейс) представляет собой совокупность средств и методов, при помощи которых пользователь взаимодействует с различными устройствами и аппаратурой. Иными словами, это тот набор кнопок, ссылок, форм, диалоговых окон, иконок, пиктограмм, баннеров, ползунков и лент прокрутки, с помощью которого пользователь управляет продуктом. [4]

Одним из этапов проектирования является прототипирование пользовательского интерфейса, который подразумевает создание прототипов экранов системы. Прототипы позволяют обнаружить проблемы функционального характера будущей системы на раннем этапе и устранить их до того, как проект уйдет в разработку к программистам [5].

Прототип страниц — это схематичное представление всех компонентов страницы и их взаимного расположения.

Ранее нами было определено, что в системе будет набор из двух персонажей — менеджера регистратуры и пациент, а также были определены сценарии их взаимодействия с приложением.

Интерфейс главной страницы обоих персонажей идентичен: три основные части — шапка, тело и подвал.

В крайней левой части шапки должен располагаться логотип и название приложения. Справа — кнопки «Регистрации» и «Авторизация».

В нижней части шапки располагается подменю «Запись на прием».

Тело главной страницы состоит из двух элементов:

1. слайдер типа «карусель», который содержит информацию о некоторых врачах;
2. параграф с краткой историей зарождения больницы.

В подвале располагается кнопка авторского права и сокращенное название организации.

На рисунках 33 – 41 представлены макеты страниц приложения.



Рисунок 33. Главная страница веб-приложения

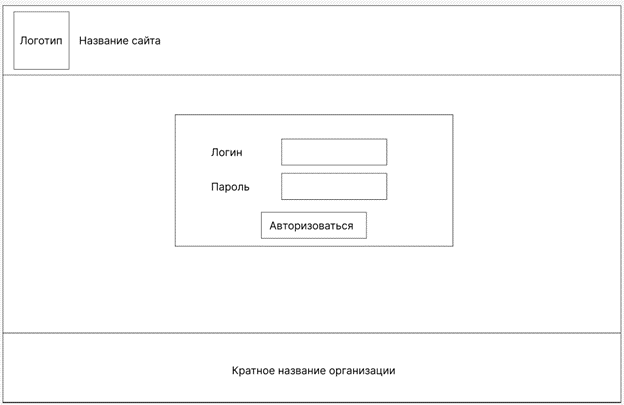


Рисунок 34. Страница с формой авторизации

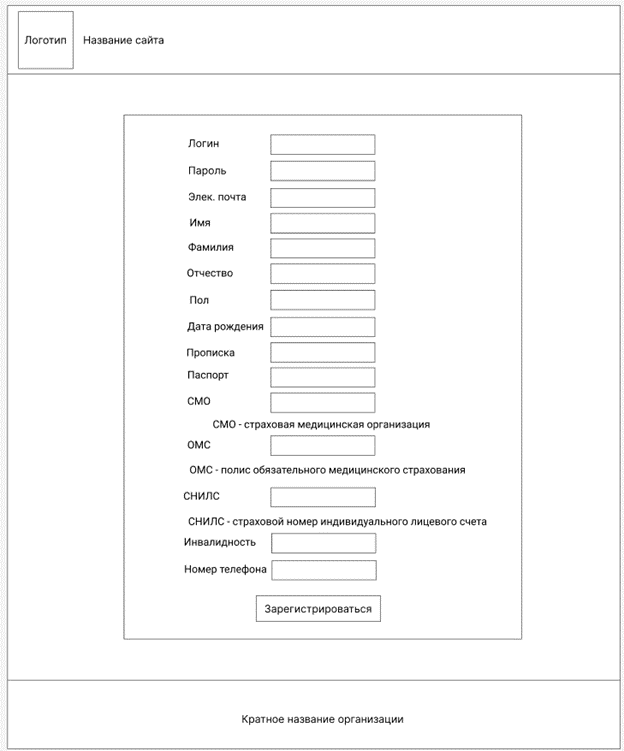


Рисунок 35. Регистрация пациента



Рисунок 36. Форма пациента для записи на приём

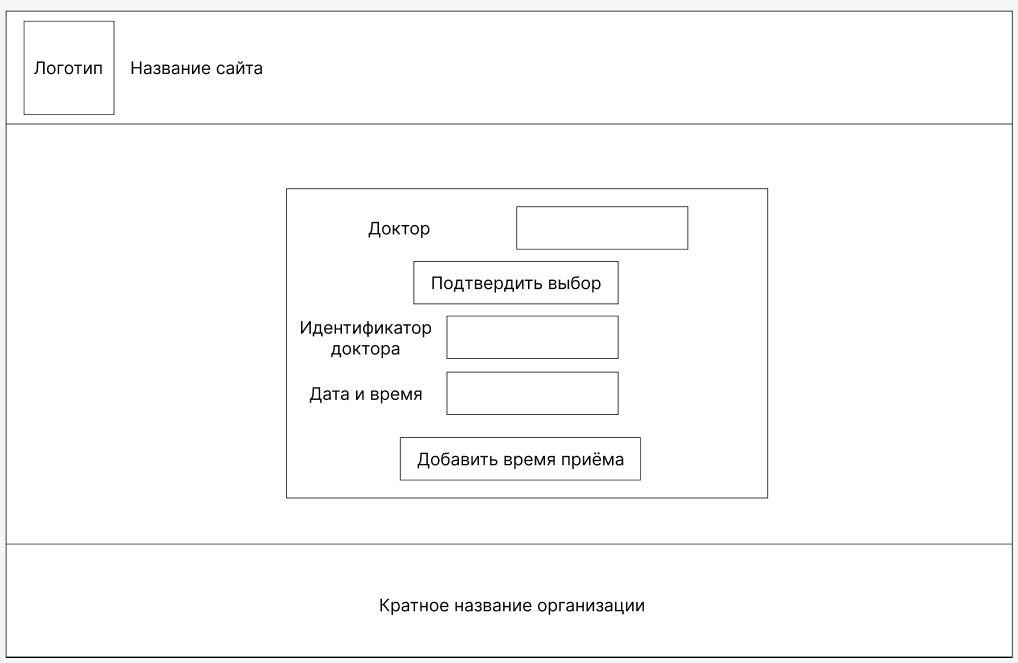


Рисунок 37. Добавление времени приёма доктору

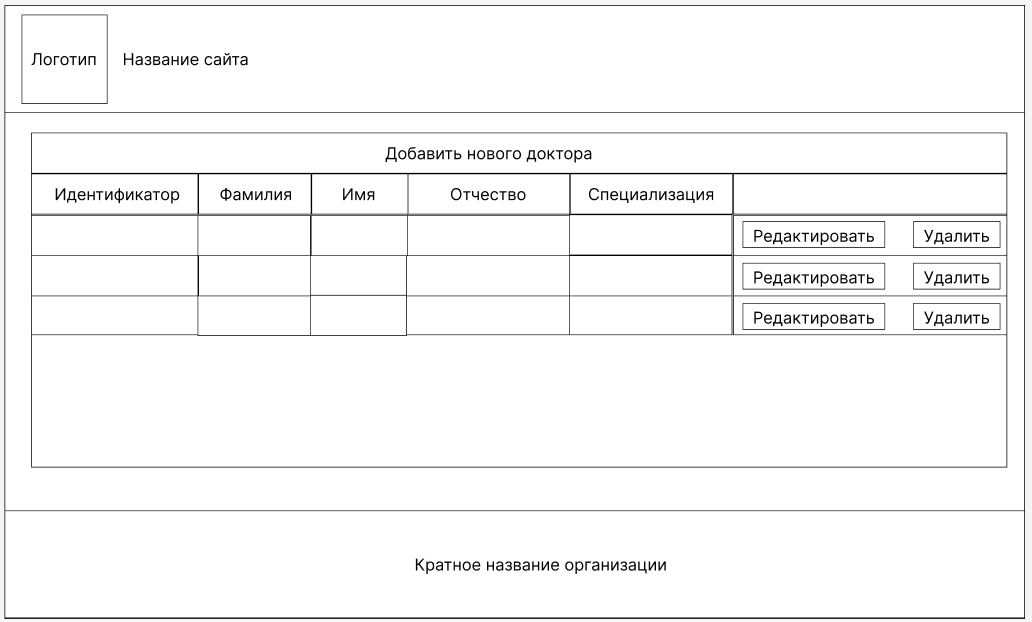


Рисунок 38. Страница работы с докторами

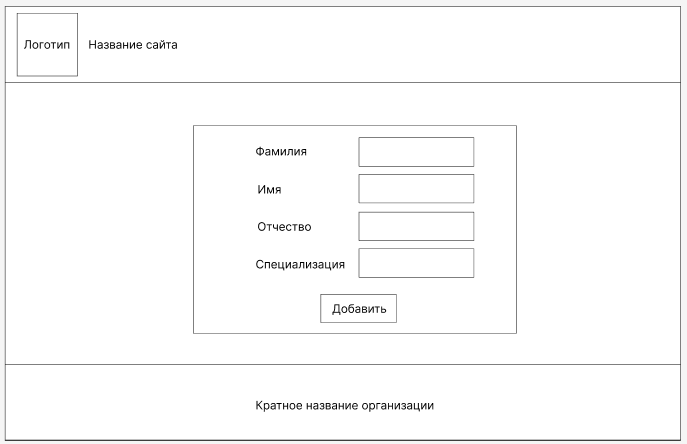


Рисунок 39. Добавление нового доктора



Рисунок 40. Страница работы с пациентами

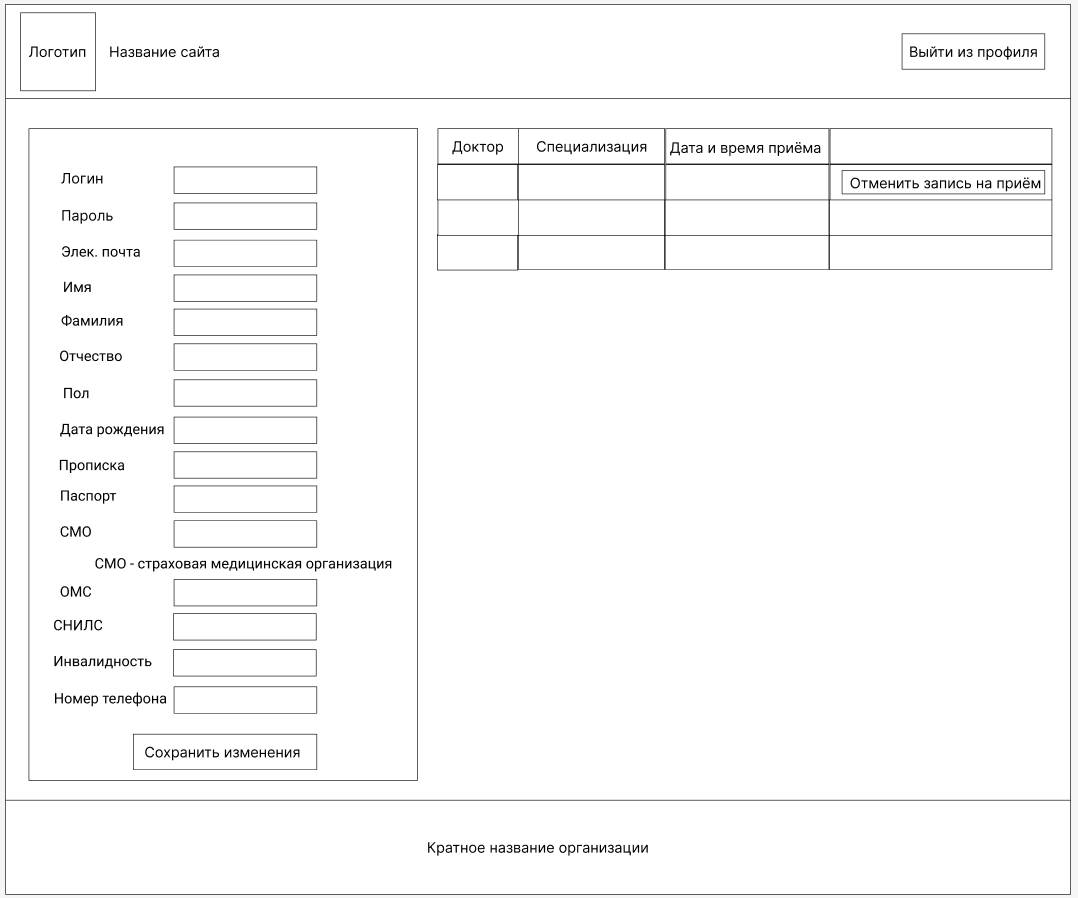


Рисунок 41. Профиль пациента

## Описание инструментальных и языковых средств разработки

**Visual Studio Code** (VS Code) — текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений. Включает в себя отладчик, инструменты для работы с Git, подсветку синтаксиса, IntelliSense и средства для рефакторинга. Имеет широкие возможности для кастомизации: пользовательские темы, сочетания клавиш и файлы конфигурации. Распространяется бесплатно, разрабатывается как программное обеспечение с открытым исходным кодом, но готовые сборки распространяются под проприетарной лицензией.

**XAMPP** — кроссплатформенная сборка локального веб-сервера, содержащая Apache, MariaDB, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер.

**phpMyAdmin** — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL. PhpMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, запускать команды SQL и просматривать содержимое таблиц и баз данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд.

**HTML** (от англ. *HyperText* *Markup* *Language* — «язык гипертекстовой разметки») — стандартизированный язык гипертекстовой разметки документов для просмотра веб-страниц в браузере. Веб-браузеры получают HTML документ от сервера по протоколам HTTP/HTTPS или открывают с локального диска, далее интерпретируют код в интерфейс, который будет отображаться на экране монитора.

**CSS** (англ. *Cascading* *Style* *Sheets* «каскадные таблицы стилей») — формальный язык декодирования и описания внешнего вида документа (веб-страницы), написанного с использованием языка разметки (чаще всего HTML или XHTML). Также может применяться к любым XML-документам, например, к SVG или XUL.

**PHP** (англ. *PHP: Hypertext Preprocessor* — «PHP: препроцессор гипертекста»; первоначально PHP/FI (Personal Home Page / Form Interpreter), а позже названный *Personal Home Page Tools* — «Инструменты для создания персональных веб-страниц») — скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов [11].

**JavaScript** — мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript (стандарт ECMA-262). JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам.

Был использован фреймворк **Bootstrap** (также известен как Twitter Bootstrap) — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения [8].

**Краткие выводы по 2 главе**

Работа в данной главе посвящена проектированию веб-приложения с применением изученных методологий. Методология SADT (IDEF0) используемая для создания функциональной модели, отображает структуру и функции разрабатываемого приложения, а также потоки информации и материальных объектов, связывающих эти функции.

Методология DFD и построенная по ее нотации диаграмма потоков данных показывает, как информация входит в систему и выходит из нее, что изменяет информацию и где она хранится.

Опираясь на ER-методологию, провели проектирование и разработку модели базы данных. Анализ требований к системе позволил определить объекты БД и их атрибуты, первичные и внешний ключи. Структура БД состоит из 5 таблиц.

Использование унифицированного языка моделирования UML демонстрирует взаимодействие пользователей с приложением через диаграмму прецедентов. Диаграмма последовательности необходима для визуализации взаимодействия между объектами в системе. Она показывает порядок сообщений, которыми объекты обмениваются друг с другом во времени и их жизненный цикл. Диаграмма активности же позволила более детально визуализировать конкретный случай использования.

Были разработаны макеты основных и дополнительных страниц для каждого типа пользователя. Прототипы страниц содержат детальное изображение будущих страниц веб-сервиса.

Основываясь на функциональные и пользовательские требования к системе были проанализированы и подобраны языковые, инструментальные средства и среды разработки.

# РЕАЛИЗАЦИЯ ПРОГРАММЫ

## Дизайнерское решение разработки веб-приложения

Интерфейс данного веб-приложения был разработан в соответствии с представленными ранее прототипами.

Логотип веб-приложения изображает стилизованный крест вместе с стетоскопом, которые традиционно ассоциируется с медициной.

Активными элементами на страницах являются: выпадающий список и поля ввода типа «text», «datetime» и «submit».

Цветовая палитра основана на общественных ассоциациях с понятием «медицина». При создании дизайна медицинских учреждений, включающего поликлиники, больницы и другие медицинские объекты, необходимо учитывать специфические требования и ожидания пациентов. Цветовая палитра играет важную роль в создании комфортной и спокойной атмосферы.

Для достижения этой цели, распространенно использование нейтральных цветов, таких как белый, бежевый, серый и светлые оттенки голубого и зеленого цветов. Нейтральные цвета создают эффект тишины, спокойствия и умиротворения.

Важно также учитывать психологические аспекты цвета. Например, голубой цвет ассоциируется с чистотой, свежестью и созиданием. Он помогает создать ощущение доверия и безопасности. Зеленый цвет символизирует природу, жизнь и регенерацию, что помогает создать ощущение спокойствия и покоя.

Все страницы состоят из трех основных частей — шапка, тело (контент) и подвал. Цвет заднего фона – «Призрачный Белый» («Ghost White», HEX-код – F5F7FB). Основным цветом является белый («White» HEX-код – FFFFFF), цвет кнопок – «васильковый» («Cornflower blue», HEX-код– 6B9AFF).

Для контраста и читаемости шрифта был выбран черный цвет (HEX-код: 000000), стиль шрифта – Roboto Sans Serif, шрифт без засечек. Данный шрифт — единственный используемый в данном веб-приложении/

## Работа пациента с приложением

На рисунке 42 представлена главная страница веб-приложения. В шапке сайта находятся логотип, название веб-приложения, кнопки «Регистрация» и «Авторизация».

В теле страницы находятся два элемента — слайдер типа «карусель» и параграф с текстом.

В подвале располагается кнопка авторского права и сокращенное название организации

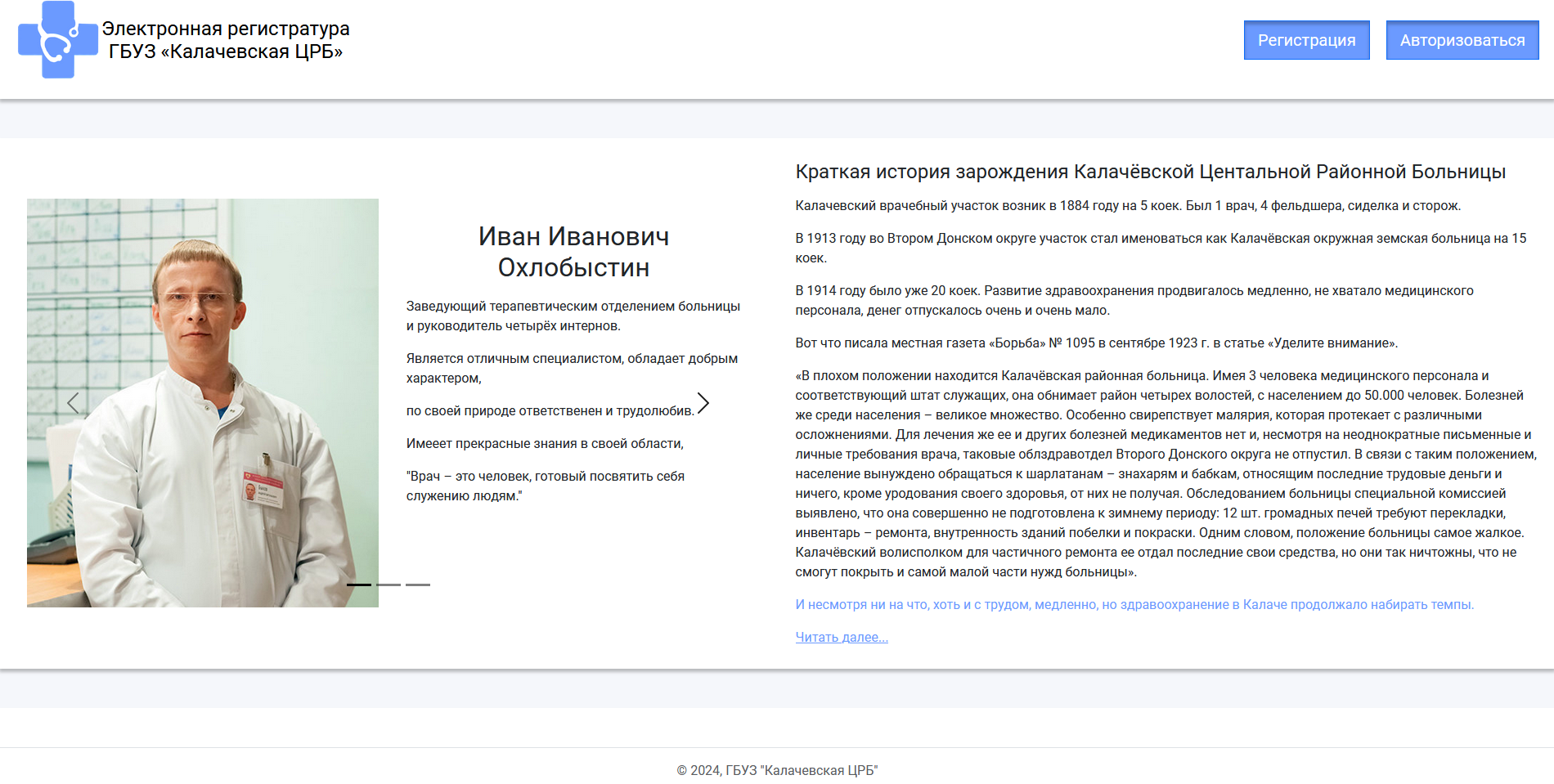


Рисунок 42. Главная страница

Рисунок 43 — страница регистрации пациента, в теле страницы находится форма с полями, такими как: «Логин», «Пароль», «Электронная почта», «Имя», «Фамилия», «Отчество», «Пол», «Дата рождения», «Прописка», «Паспорт», «СМО» (Страховая медицинская организация), «ОМС», «СНИЛС», «Инвалидность», «Номер телефона» и кнопка «Зарегистрироваться».

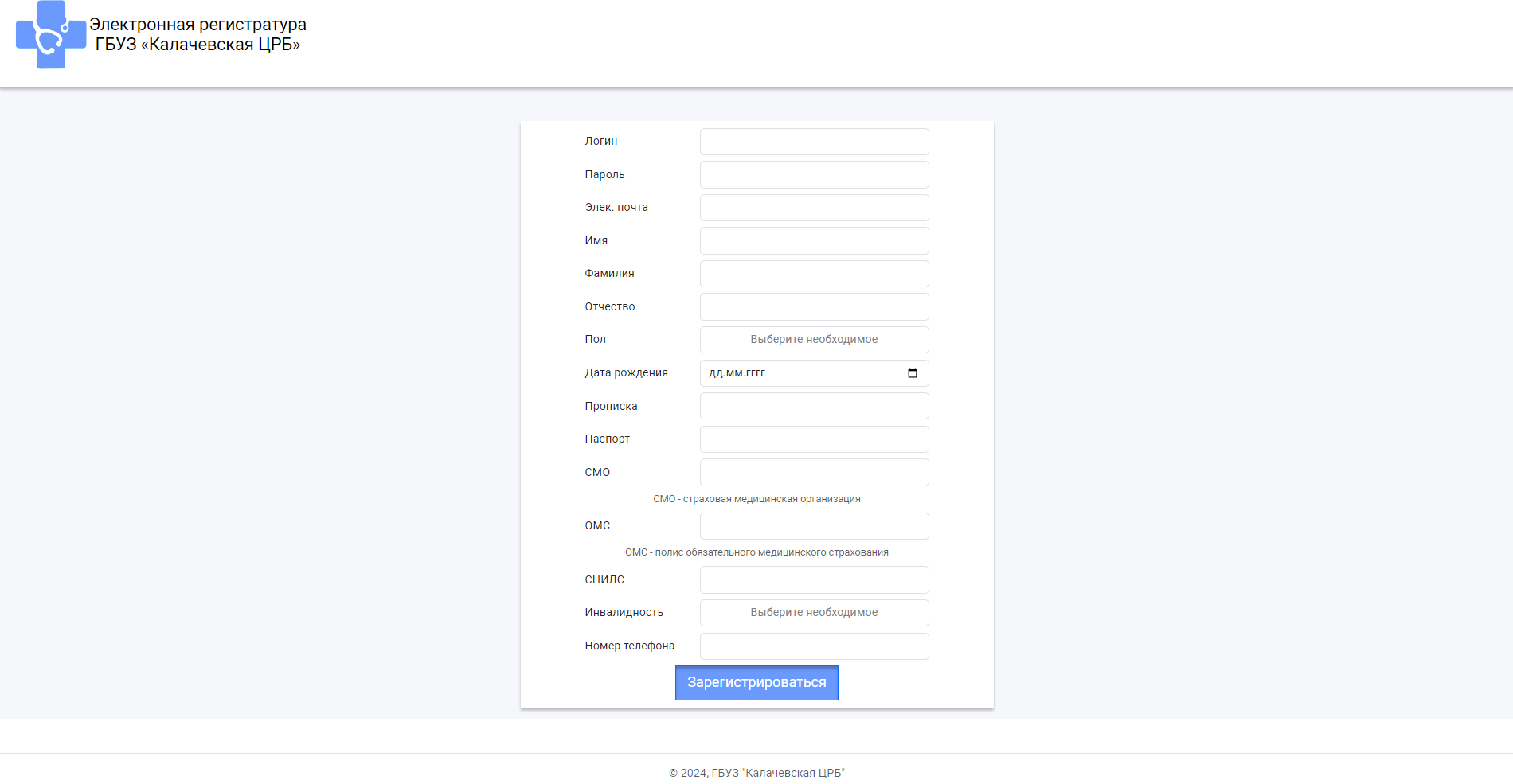


Рисунок 43. Страница регистрации

Фрагмент кода, отвечающий за регистрацию пациента, представлен в Листинге 1.

$query = "INSERT INTO `patients` (

    `login`, `password`,`email`,`name`,  `surname`, `patronymic`,  `gender`, `birthdayDate`, `permanentResidence`, `passport`, `mio`,

    `policyCMI`,`policyPIP`,`disability`, `phoneNumber`)

    VALUES( '$reg\_login','$reg\_password','$reg\_email','$reg\_name','$reg\_surname','$reg\_patronymic','$reg\_gender','$reg\_birthdayDate',

        '$reg\_permanentResidence', '$reg\_passport', '$reg\_mio', '$reg\_policyCMI','$reg\_policyPIP', '$reg\_disability','$reg\_phoneNumber')";

    $res = $conn->query($query);

Листинг 1. Регистрация пациента

Форма авторизации, представленная на рисунке 44, содержит два поля: логин и пароль, кнопку «Авторизоваться». Если пользователь ранее не был зарегистрирован, под кнопкой «Авторизоваться» расположена гиперссылка для регистрации.

Фрагмент кода, отвечающий за авторизацию пациента в приложении, представлен в Листинге 2.

if (isset($\_POST['login\_user'])) {

    $auth\_login = $\_POST['auth\_login'];

    $auth\_password = md5($\_POST['auth\_password']);

    $result = mysqli\_query($conn, "SELECT\*FROM `patients`

    WHERE `login` = '" . $auth\_login ."' and `password` = '" . $auth\_password ."'");

    if(!empty($result)) {

        if ($row = mysqli\_fetch\_array($result)) {

            $\_SESSION['user\_id'] = $row['patientId'];

            $\_SESSION['user\_login'] = $row['login'];

            $\_SESSION['user\_password'] = $row['password'];

            header("Location: ../index.php");

            exit();

Листинг 2. Авторизация пациента в веб-приложении

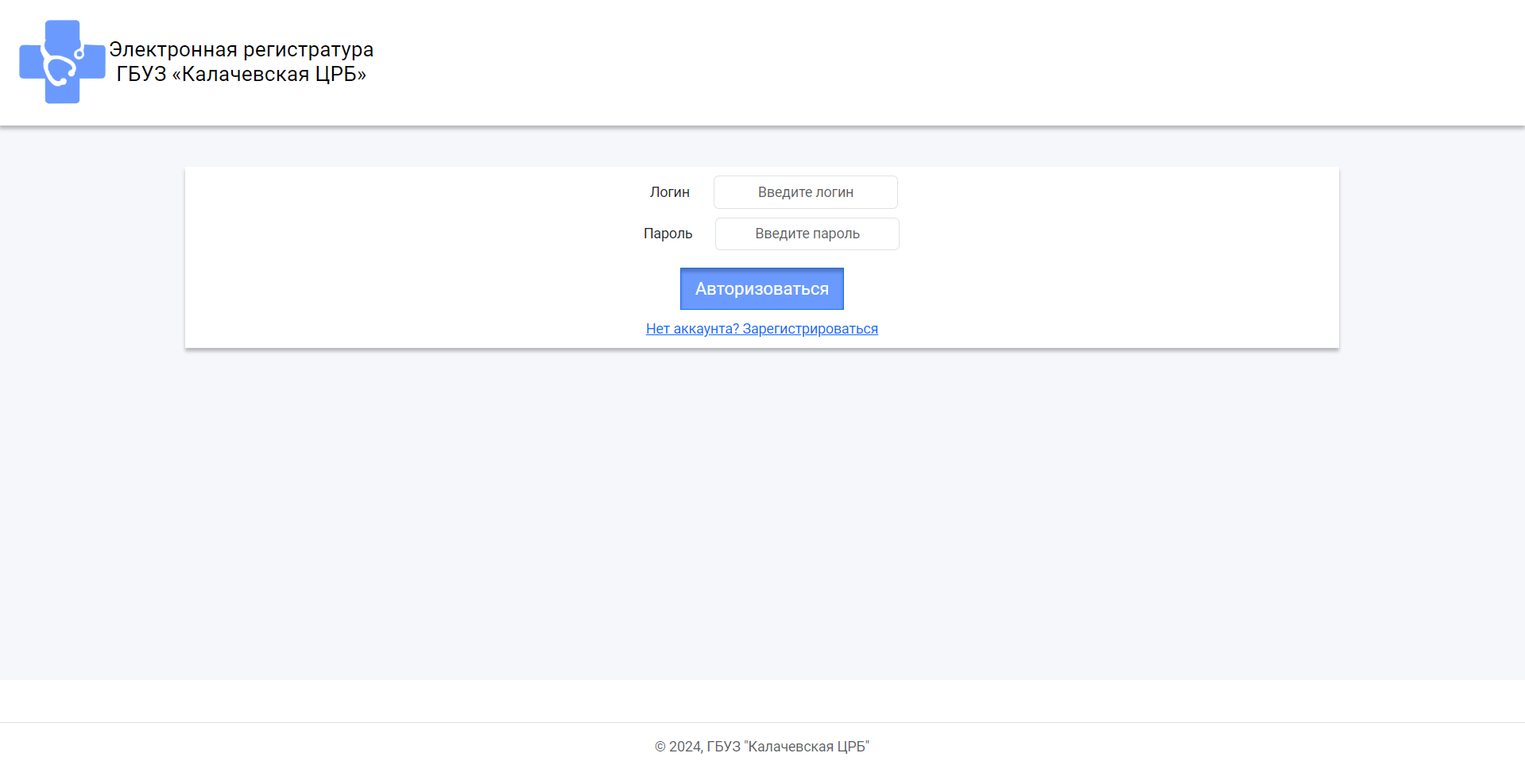


Рисунок 44. Страница авторизации

После успешной авторизации пользователь «Пациент» попадает на главную страницу веб-приложения (рисунок 45), где может ознакомиться с историей медицинского учреждения, а также изучить информацию о врачах, ведущих приемы в поликлинике.

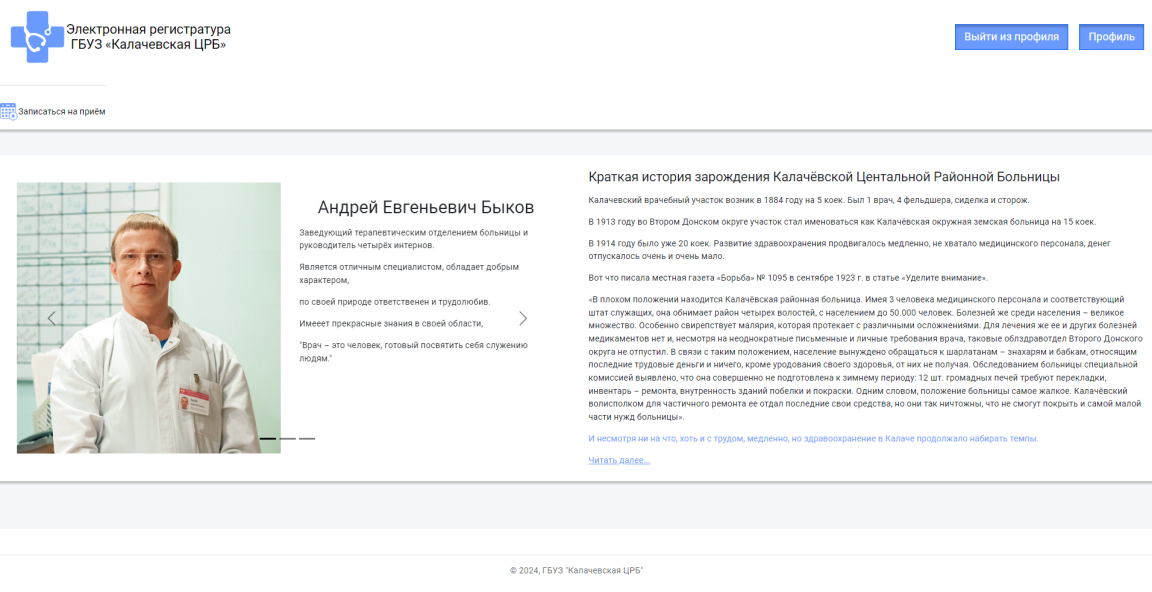


Рисунок 45. Главная страница авторизованного пациента

Запись на прием к врачу — одна из самых востребованных услуг в здравоохранении, система самостоятельной записи через Интернет позволяет получать эту услугу более качественно, с экономией времени, сил, нервов.

Если пациент планирует записаться на прием, он должен быть зарегистрирован в системе. Для того, чтобы пациент мог воспользоваться услугами медицинского учреждения, ему необходимо пройти авторизацию, и только после этого пациент попадает в свой Профиль (рисунок 46).

В верхней части окна располагается строка приветствия. В основном поле контента расположены две формы: 1-я — отображает персональную информацию, 2-я форма отображает запись на прием.

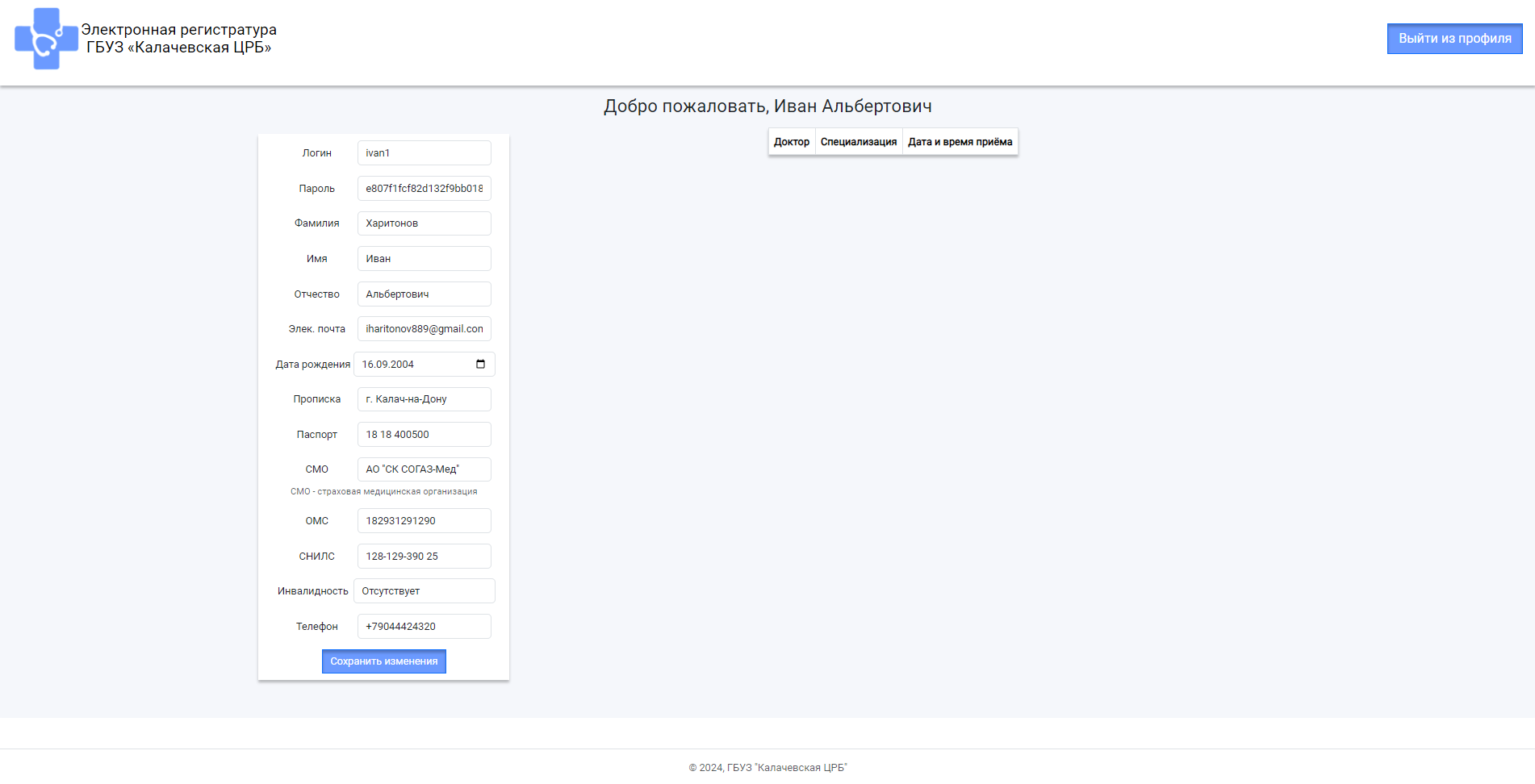


Рисунок 46. Профиль пациента

Поля формы с персональными данными заполняются самим пациентом и, при необходимости, информация может быть отредактирована.

Если по каким-либо причинам пациент не может прийти на прием к врачу, он может отменить существующую запись. Для этого на 2-ой форме присутствует кнопка «Отменить запись».

Фрагмент кода, отвечающий за отмену записи, представлен в Листинге 3.

if (isset($\_GET['scheduleId'])) {

    include ('../../pageController.php');

    $connect = getAbsolutePath('server/connect.php');

    include $connect;

    $scheduleId = $\_GET['scheduleId'];

    $sql = "UPDATE `schedules` SET `patientId`=null

    WHERE `scheduleId` = $scheduleId; ";

    if (mysqli\_query($conn, $sql)) {

        header("Location: ./profile.php");

    }

Листинг 3. Отмена пациентом записи на приём

Для оформления записи на прием пациенту необходимо заполнить форму (рисунок 47). На форме необходимо выполнить следующий порядок действий:

1. из выпадающего списка выбрать специальность врача, который интересует, нажать кнопку «Продолжить»;
2. в поле «Доктор», которое также представляет собой выпадающий список, выбрать лечащего врача и также, нажать кнопку «Продолжить»;
3. в поле «Дата и время» выбрать нужную информацию. После чего необходимо убедиться, что все поля заполнены, информация указана верно и нажать кнопку «Записаться на приём».

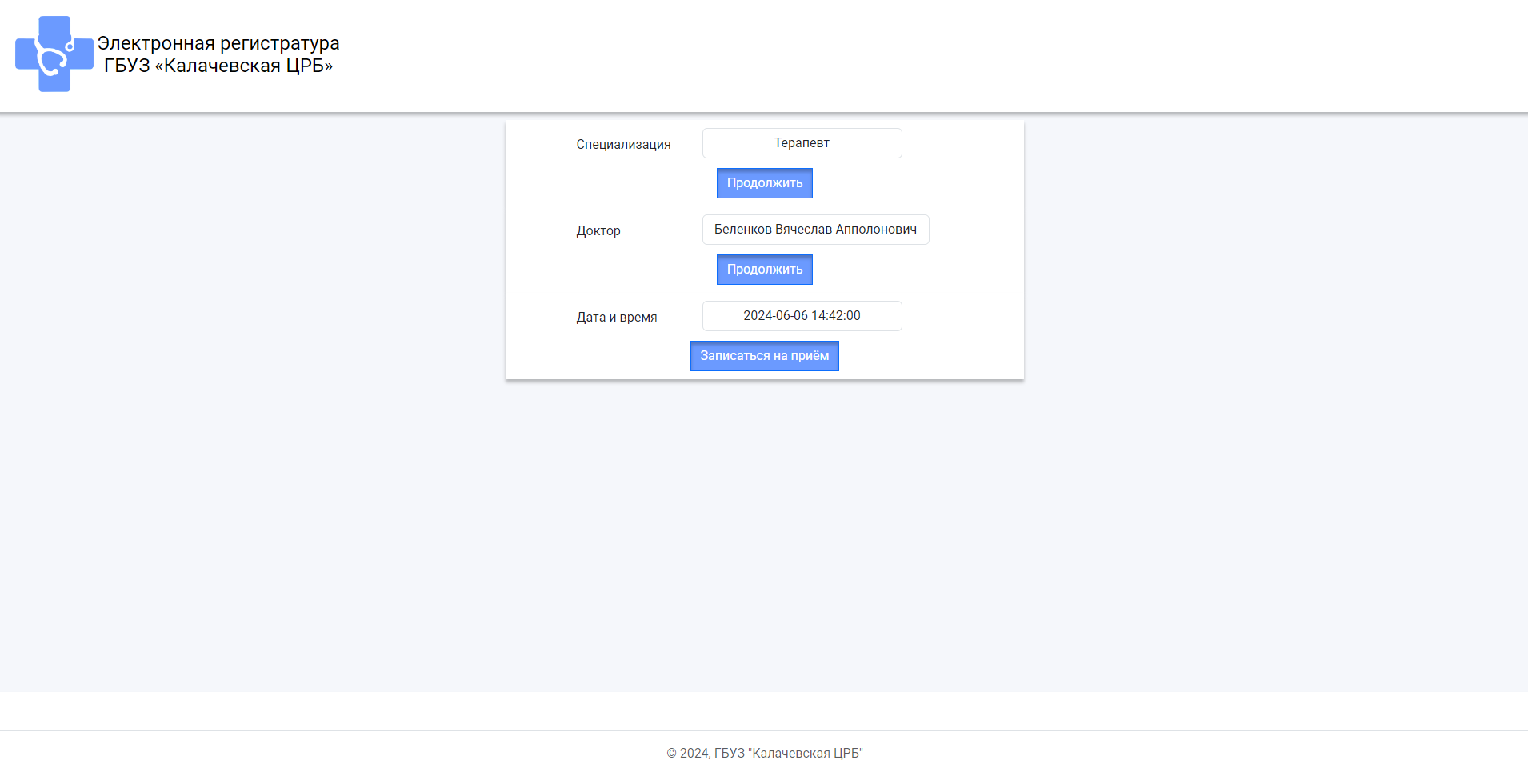


Рисунок 47. Запись пациента на приём

Фрагмент кода, отвечающий за реализацию записи на прием пациентом, представлен в Листинге 4.

if (isset($\_POST['selectSchedule'])) {

  $selectedSchedule = $\_POST['selectSchedule'];

  $query = "UPDATE `schedules` SET `patientId`= '" . $\_SESSION['user\_id'] ."'

  WHERE `scheduleId` = '" . $selectedSchedule ."'";

  $res = $conn->query($query);

  $conn->close();

  unset($\_SESSION['selectedSpecialization']);

  unset($\_SESSION['selectedDoctor']);

  header("Location:./profile.php");

  exit();

}

Листинг 4. Реализация функции пациента «Запись на прием»

## Работа менеджера регистратуры с приложением

После авторизации в приложении, менеджер попадает на главную страницу. Интерфейс формы регистратора изображен на рисунке 48.

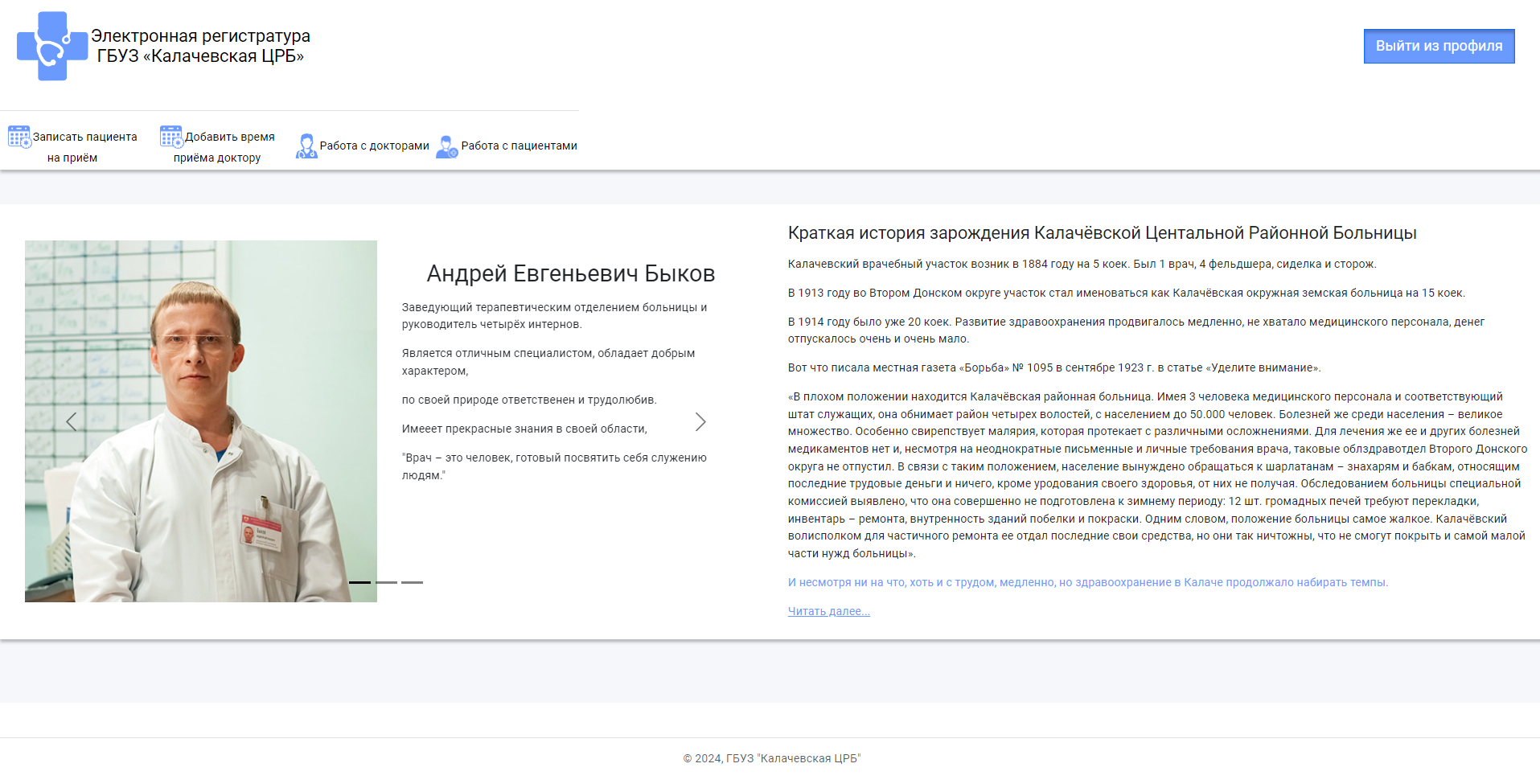


Рисунок 48. Главная страница авторизованного менеджера регистратуры

Основной задачей администратора является эффективное взаимодействие с пациентами с целью их записи на услуги медицинской организации. Отличительной особенностью страницы менеджера регистратуры является расширенное меню, представленное четырьмя кнопками «Записать пациента на прием», «Добавить время приёма доктору», «Работа с докторами», «Работа с пациентами».

Процесс «Запись пациентов на прием» может осуществляться через телефонный звонок или при непосредственном обращении пациента в регистратуру. При первичном обращении пациента по телефону проводит запись и регистрацию пациента, подбирая удобное время для пациента и, при необходимости консультаций нескольких специалистов, приемы ставит последовательно. За день до приема проводит телефонные переговоры с пациентом с целью подтверждения записи пациента на прием к врачу.

С пациентом, впервые обратившимся в медицинское учреждение, заключается договор, оформляется информированное добровольное согласие пациента на медицинское вмешательство, и заводится амбулаторная карта пациента.

В приложении реализована возможность регистрировать вновь поступившего пациента. Форма регистрации пациента представлена на рисунке 49.

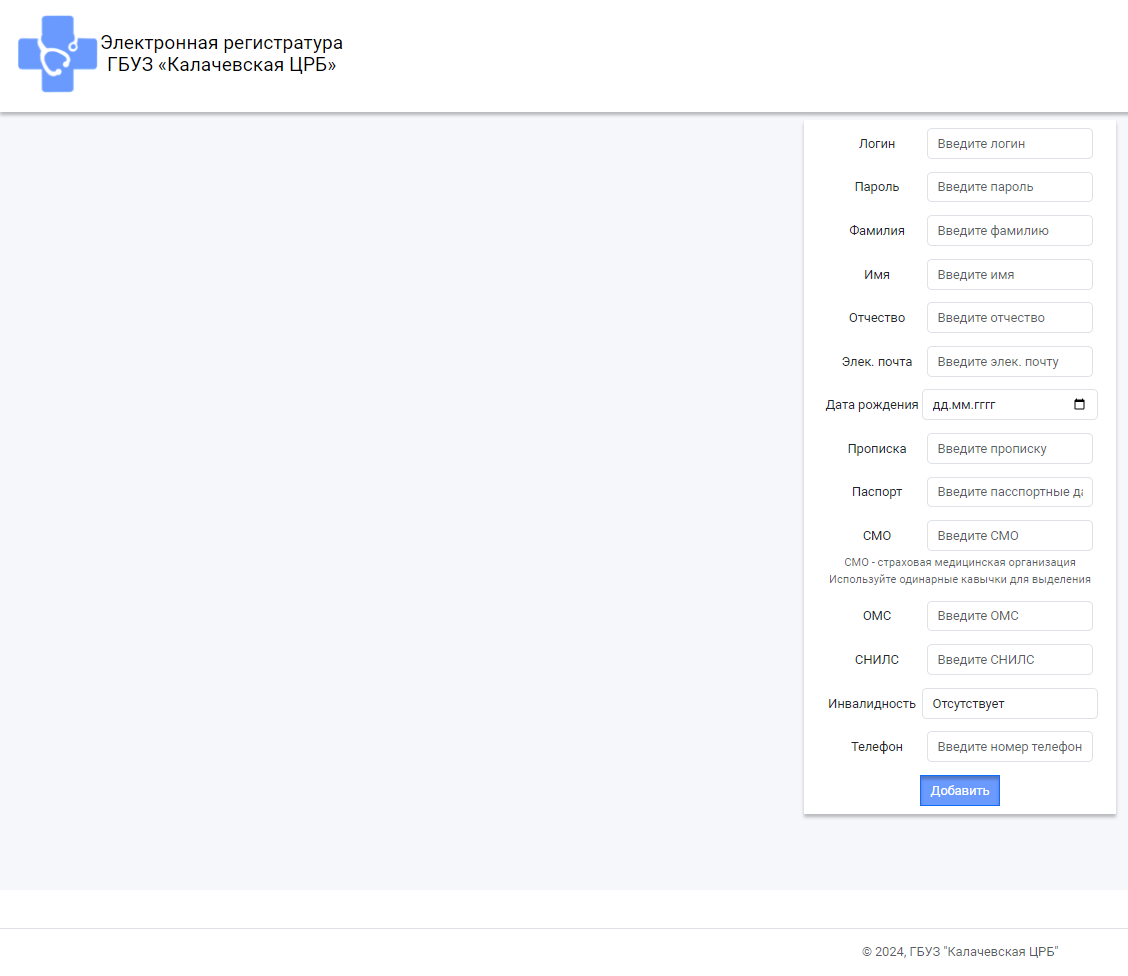


Рисунок 49. Форма менеджера для регистрации пациента

Фрагмент кода, отвечающий за регистрацию нового пациента, представлен в Листинге 5.

$sqlInsert = "INSERT INTO `patients` (`login`, `password`, `name`, `surname`, `patronymic`,

                `email`, `birthdayDate`, `permanentResidence`, `passport`, `mio`, `policyCMI`, `policyPIP`, `disability`, `phoneNumber`)

                VALUES ('" . $login ."','" . $password ."','" . $name ."', '" . $surname ."','" . $patronymic ."',

                '" . $email ."','" . $birthdayDate ."', '" . $permanentResidence ."',

                '" . $passport ."','" . $mio ."', '" . $policyCMI ."','" . $policyPIP ."',

                '" . $disability ."', '" . $phoneNumber ."')";

                $conn->query($sqlInsert);

                $conn->close();

Листинг 5. Добавление пациента менеджером

В случаях, когда у пациента клиники изменились персональные данные (например, смена фамилии, адреса места жительства, замена паспорта и др.), менеджер регистратуры может изменить данные профиля пациента. После внесения всех изменений необходимо на форме нажать кнопку «Сохранить изменения».

Форма для редактирования данных пациента представлена на рисунке 50.

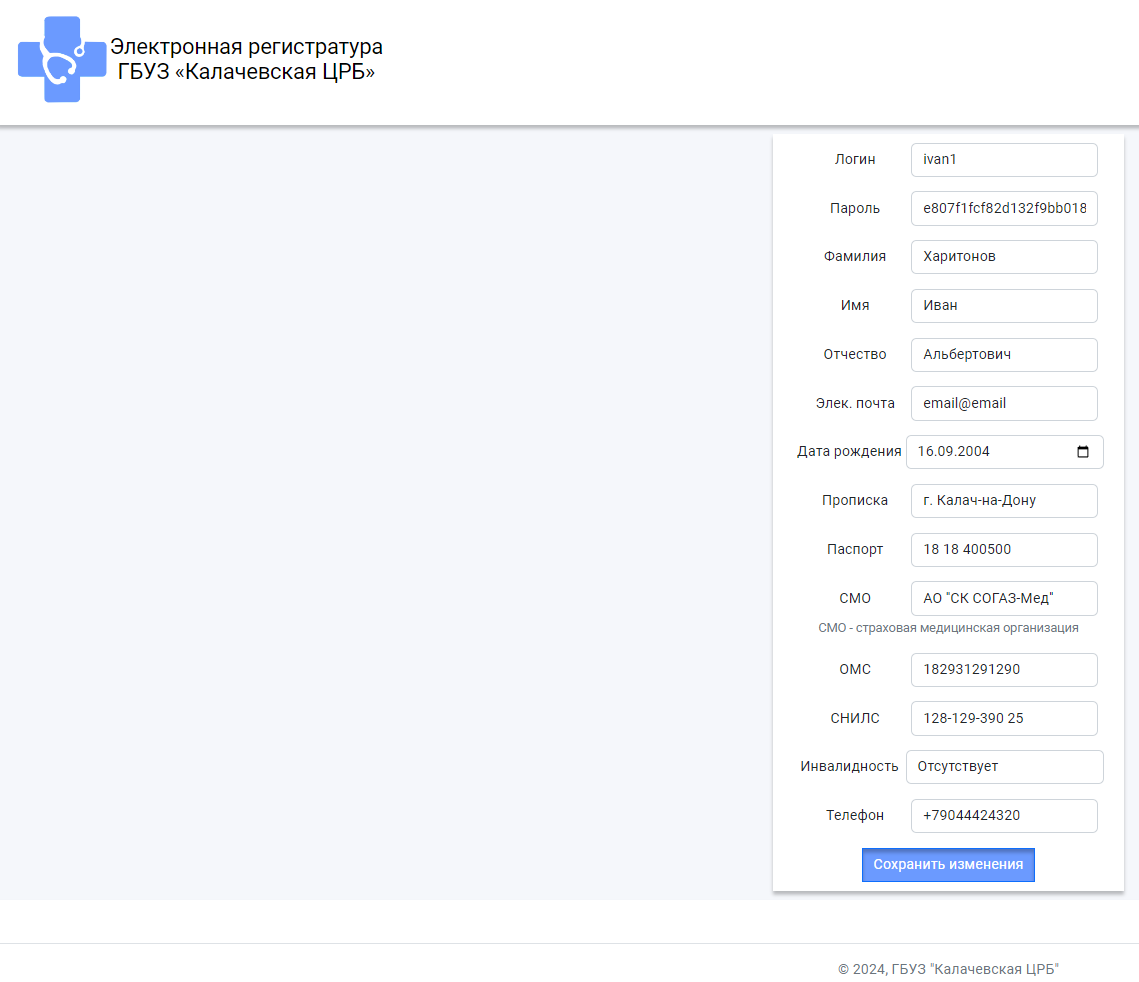


Рисунок . Форма редактирования персональных данных пациента

Фрагмент кода «Редактирование информации о пациенте» представлено в Листинге 6.

    $sqlUpdate = "UPDATE `patients` SET `login`= '" . $login ."',

    `password`= '" . $password ."',`name`= '" . $name ."' ,

    `surname`= '" . $surname ."', `patronymic`='" . $patronymic ."',

    `email`= '" . $email ."' ,`birthdayDate`= '" . $birthdayDate ."' ,

    `permanentResidence`= '" . $permanentResidence ."' ,`passport`= '" . $passport ."' ,

    `mio`= '" . $mio ."' ,`policyCMI`= '" . $policyCMI ."' ,

    `policyPIP`= '" . $policyPIP ."' ,`disability`= '" . $disability ."' ,

    `phoneNumber`= '" . $phoneNumber ."'

    WHERE `patientId` = $\_SESSION[user\_id]";

    $conn->query($sqlUpdate);

    $conn->close();

    echo "<meta http-equiv='refresh' content='0'>";

Листинг 6. Редактирование профиля

В поликлинике участковый врач работает по скользящему графику, ведет прием больных в поликлинике, обслуживает первичные вызовы на дом и планирует активные посещения в зависимости от состояния здоровья больного. Чтобы пациент мог самостоятельно записаться на прием, необходимо, чтобы в приложении отображалась актуальная информация о врачах и их графике приемов.

Менеджер регистратуры должен своевременно обновлять информацию о врачах. В приложении реализованы функции «Работа с докторами». Вся информация об участковых врачах в профиле менеджера структурирована в виде таблицы, которая состоит из следующих столбцов: «Идентификатор», «Фамилия», «Имя», «Отчество», «Специализация». Каждая строчка имеет кнопки управления «Редактировать» и «Удалить». Данная форма представлена на рисунке 51.

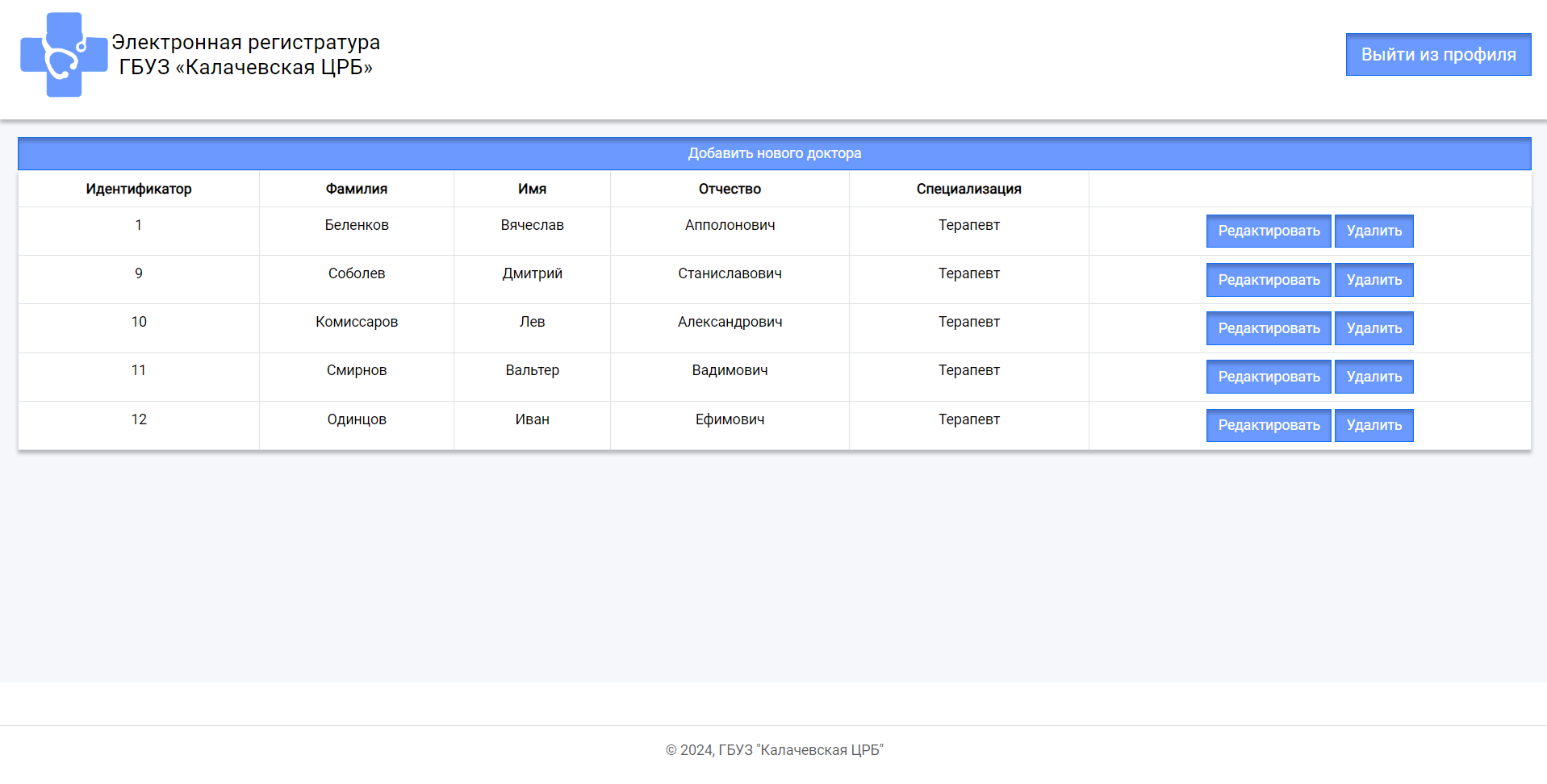


Рисунок . Отображение информации о врачах

При необходимости регистратор может создать запись о враче. Для этого необходимо нажать на кнопку «Добавить нового доктора». В открывшейся форме (рисунок 52) заполнить текстовые поля «Фамилия», «Имя», «Отчество» и выпадающий список «Специализация».

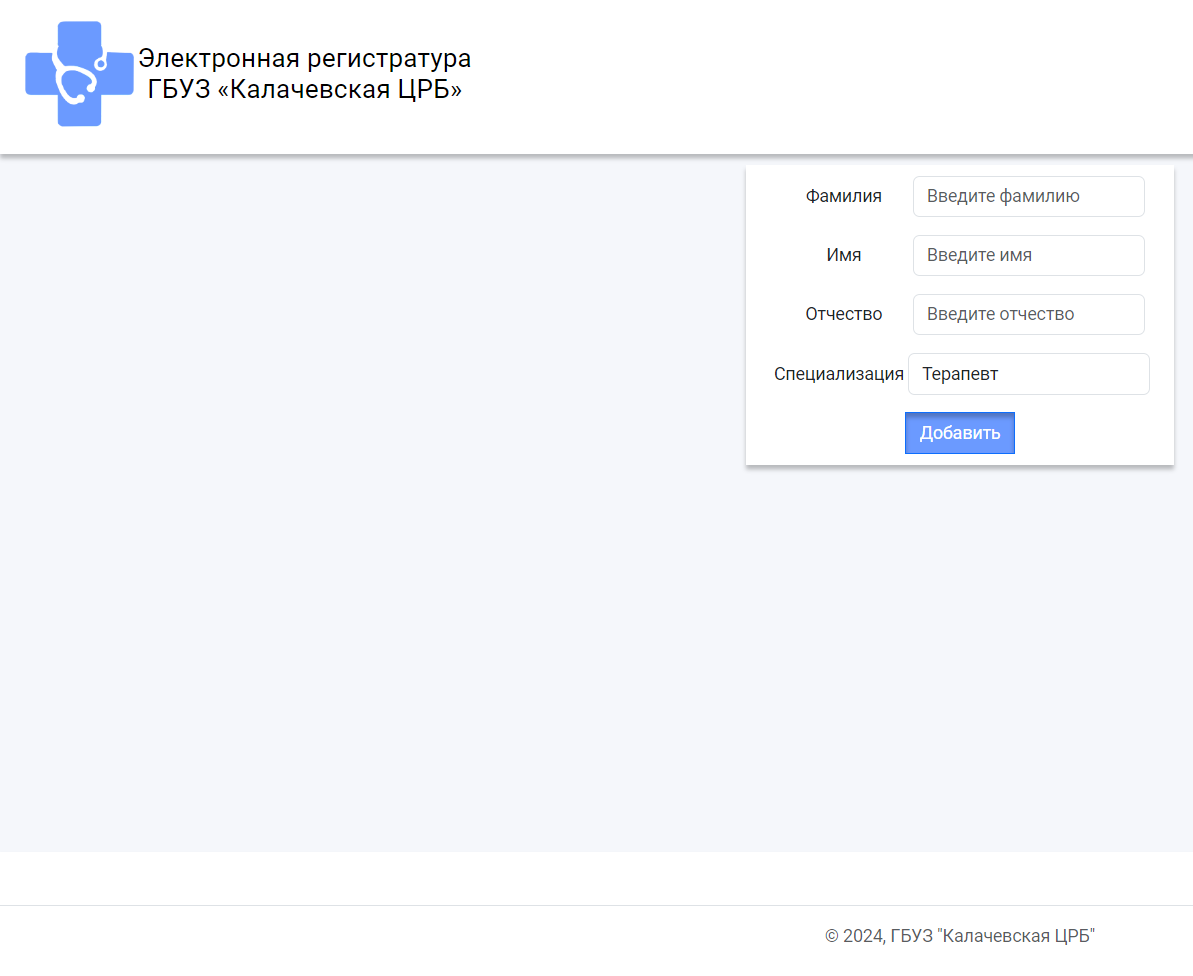


Рисунок . Форма для добавления записи о враче

Фрагмент кода, отвечающий за данную функцию, представлен в Листинге 7.

      if (isset($\_POST["addDoctor"])) {

        $name = mysqli\_real\_escape\_string($conn, $\_POST["name"]);

                $surname = mysqli\_real\_escape\_string($conn, $\_POST["surname"]);

                $patronymic = mysqli\_real\_escape\_string($conn, $\_POST["patronymic"]);

                $specialization = mysqli\_real\_escape\_string($conn, $\_POST["specialization"]);

                $query = "INSERT INTO `doctors` (`name`, `surname`, `patronymic`, `specializationId`) VALUES

                ('" . $name ."','" . $surname ."','" . $patronymic ."', '" . $specialization ."')";

                $conn->query($query);

                $conn->close();

            }

Листинг 7. Добавление записи о враче

Если у врача изменилась какая-либо информация, менеджер регистратуры может редактировать предыдущую запись.

На рисунке 53 представлена форма редактирования записи о враче.

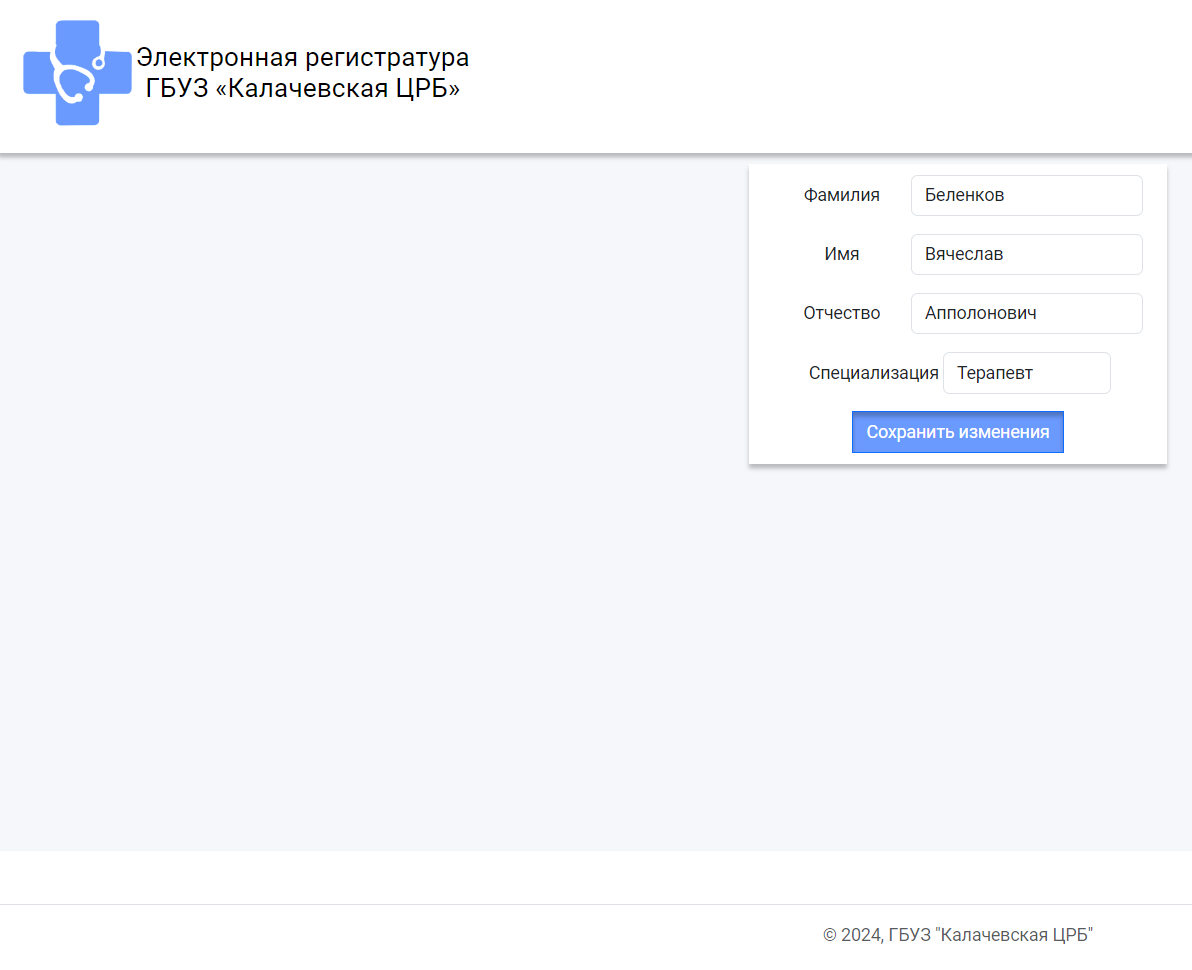


Рисунок . Форма редактирования записи о враче

Фрагмент кода, отвечающий за реализацию данной функции, представлен в Листинге 8.

if (isset($\_POST["edit"])) {

        $name = mysqli\_real\_escape\_string($conn, $\_POST["name"]);

                $surname = mysqli\_real\_escape\_string($conn, $\_POST["surname"]);

                $patronymic = mysqli\_real\_escape\_string($conn, $\_POST["patronymic"]);

                $specialization = mysqli\_real\_escape\_string($conn, $\_POST["selectSpecialization"]);

                $sqlUpdate = "UPDATE `doctors` SET `name`= '" . $name ."' , `surname`= '" . $surname ."',

                `patronymic`='" . $patronymic ."', `specializationId`='" . $specialization ."'

                WHERE `doctorId` = '" . $doctorId ."' ";

                $conn->query($sqlUpdate);

                $conn->close();

                echo "<meta http-equiv='refresh' content='0'>";             }

Листинг 8. Редактирование информации о враче

Формируя оптимальную запись пациентов на осмотры или консультации, регистратор старается минимизировать простои в графике врачей, ведя плотную запись. В случае невыхода врача на прием обеспечивает перезапись пациента на другое время или изыскивает возможность замены врача. Кроме того, пациент может отменить запись к врачу. В таких случаях менеджер регистратуры должен перераспределять талоны на запись: отслеживать электронную запись и, при необходимости, оповещать пациентов об образовавшемся окне.

Реализация описанной выше функции на данном этапе возможна в виде добавления времени приемов врачу. На рисунке 54 представлена страница добавления времени приёма доктору. Тело страницы состоит из выпадающего списка и поля ввода, где можно выбрать время.

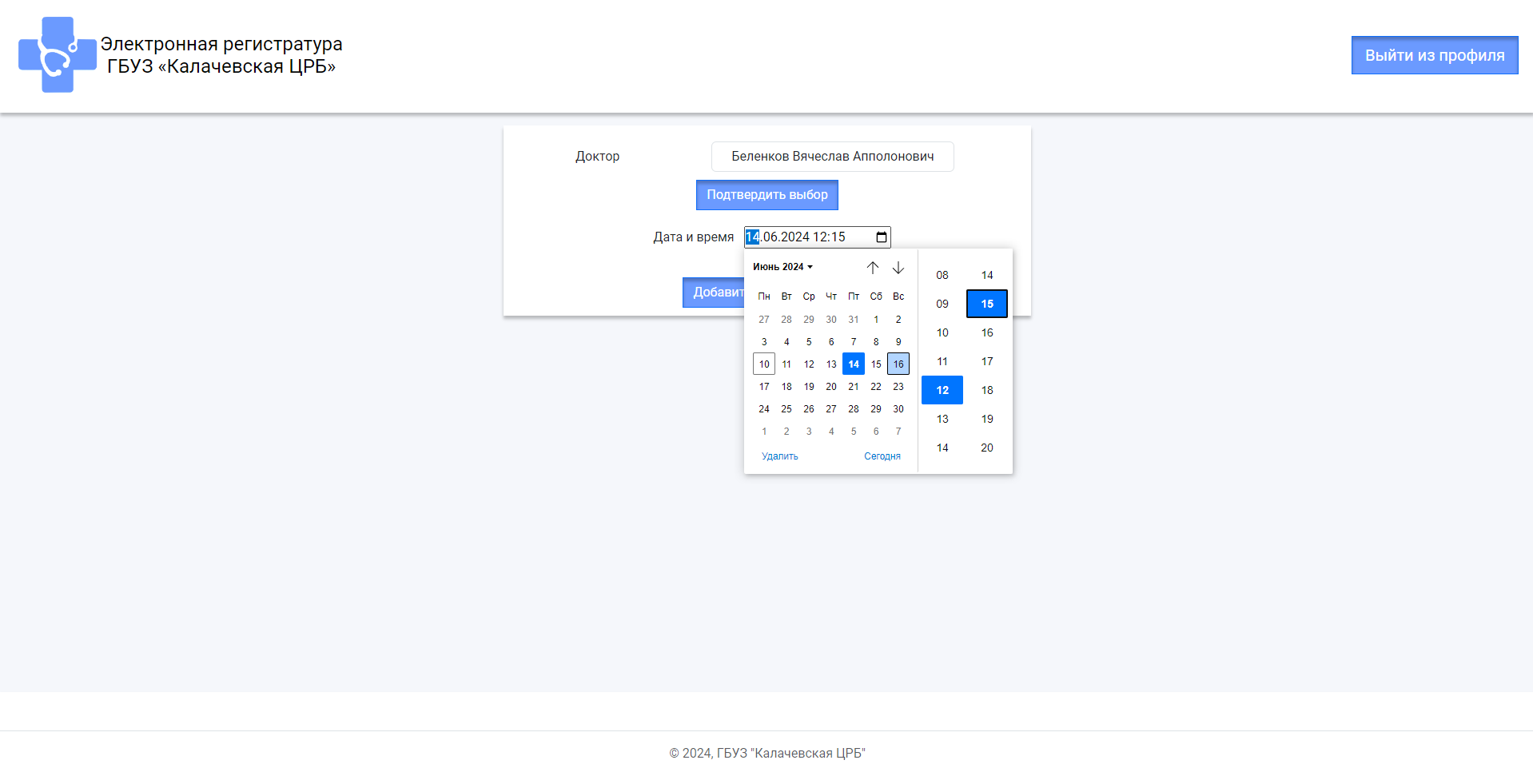


Рисунок 54. Добавление времени приёма доктору

В Листинге 9 представлен фрагмент функции добавления времени приема врачу.

if (isset($\_POST['scheduleDT'])) {

            $selectedDoctor = $\_POST['selectedDoctor'];

            $scheduleDT = $\_POST['scheduleDT'];

            $query = mysqli\_query($conn, "INSERT INTO `schedules` (`doctorId`, `datetimeOfReception`)

            VALUES ('$selectedDoctor','$scheduleDT')");

            $res = $conn->query($query);

            $conn->close();

Листинг 9. Добавление времени приёма доктору

## Тестирование

**Тестирование** — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом. Тестирование играет решающую роль в обеспечении высокого качества и надежности программного обеспечения.

Веб-тестирование -— это тип тестирования программного обеспечения, который включает проверку веб-сайтов или веб-приложений на наличие проблем. Это процесс тщательного тестирования веб-приложений перед их запуском в Сеть. Веб-тестирование проверяет веб-приложение или веб-сайт на функциональность, удобство использования, безопасность, совместимость и производительность.

Тестирование было направлено в первую очередь на проверку работоспособности функциональных возможностей программного продукта, а также для:

* обнаружение ошибок;
* обеспечение качества функций;
* повышение надежности.

Тестирование функций пациента представлено на таблице 4

Таблица 4

Тестирование функций пациента

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Шаги | Ожидаемый результат | Результат тестирования |
| Авторизация | Ввод логина и пароля в форму авторизации | Успешная авторизация пользователя и получение доступа к функциям | Пользователь успешно авторизуется |
| Регистрация | Ввод личных данных пользователя | Успешное создание учетной записи(профиля) пациента | Пользователь успешно прошел регистрацию |
| Запись на приём | Выбор необходимого врача, даты и времени приёма | Пациент зарезервирует дату и время приёма | Пациент успешно записался на приём |
| Отмена записи на приём | Пациент выбирает свою запись, которую хочет отменить | Пациент отменяет запись на приём, освобождая окно приёма | Пациент успешно отменяет свою запись на приём |
| Редактирование профиля пациента | Пациент изменяет необходимые ему данные | Пациент отредактировал и сохранил данные | Данные пациента были успешно изменены |

Аналогичным образом проведено тестирование функций менеджера регистратуры, результаты которого представлены в таблице 5.

Таблица 5

Тестирование функций менеджера регистратуры

|  |  |  |  |
| --- | --- | --- | --- |
| Описание | Шаги | Ожидаемый результат | Результат тестирования |
| Авторизация | Ввод логина и пароля в форму авторизации | Успешная авторизация пользователя и получение доступа к функциям | Пользователь успешно авторизуется |
| Создание профиля пациента | Ввод в форму создания личных данных пациента | Менеджер создает профиль нового пациента | Менеджер успешно создает запись о новом пациенте |
| Редактирование профиля пациента | Изменение необходимых данных в профиле пациента | Менеджер редактирует конкретные данные пациента | Менеджер успешно сохраняет редактированные данные |
| Регистрация медицинского работника | Ввод данных доктора на форму | Менеджер создает нового доктора | Менеджер успешно создает нового доктора |
| Ведение расписания работы медицинского работника | Выбор доктора, выбор даты и времени приёма | Менеджер создает новое окно приёма у доктора | Менеджер успешно создает окно приёма в расписании доктора |

**Краткие выводы по главе 3**

В третьей главе описана реализация программного модуля «Электронная регистратура»:

* дано обоснование выбора цветового решения, шрифтов и иных атрибутов дизайна приложения;
* наглядно представлена реализация клиентской части приложения;
* проведено тестирование веб-приложения.

# ЗАКЛЮЧЕНИЕ

Реализация веб-приложения позволила обеспечить более удобный и быстрый доступ к записи на приём, и другим сервисам, что значительно повышает уровень оказываемых услуг и удовлетворенность пользователей.

Создание системы для хранения, обработки и анализа больших объемов данных обеспечивает менеджерам регистратуры возможность эффективного управления информацией, принятия обоснованных решений и планирования дальнейших действий.

Внедрение автоматизированных процессов существенно сократило временные затраты на выполнение рутинных операций и повысило эффективность работы персонала регистратуры.

Веб-приложение способствует оптимизации процессов внутри регистратуры, улучшает координацию действий между сотрудниками, снижает вероятность ошибок и повышает качество предоставляемых услуг.

Дальнейшая перспектива развития веб-приложения заключается в составлении плана по улучшению функциональности, устранения ошибок, а также внедрение новых функций и возможностей, такие как:

* перераспределение выданных талонов;
* блокировка возможности отмены записи на приём за 24 часа до приёма;
* заключение договоров на платные медицинские услуги с использованием установленных форм и шаблонов;
* ведение не только амбулаторной карты пациента поликлиники, но и стоматологической амбулаторной карты;
* рассылка смс–уведомлений пациентам с напоминанием о посещении приёма;
* вызов врача на дом и ведение журнала вызовов;
* работа с листами нетрудоспособности.

В ходе выполнений выпускной квалификационной работы на тему «Электронная регистратуры» были выполнены все поставленные задачи, а также реализовано веб-приложение.

# СПИСОК ИСТОЧНИКОВ

1. Голицына, О.Л. СУБД / О.Л. Голицына, В. Максимов, И.И. Попов. – М.: Форум, 2014. – 352 с.
2. Гниденко, И.Г. Технология разработки программного обеспечения : учеб. Пособие для СПО / И.Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. – М.: Издательство Юрайт, 2017. – 235с.
3. Кузнецов, С.Д. Базы данных. Модели и языки. / С.Д. Кузнецов. –М.: Бином пресс, 2008. – 284 с.
4. Купер А. Интерфейс. Основы проектирования взаимодействия. – 4-е изд. – Спб.: Питер, 2021. – 720 с.
5. Сысолетин, Е.Г. С95 Проектирование интернет-приложений: учеб. – метод. пособие / Е.Г. Сысолетин, С.Д. Ростунцев. – Екатеринбург : Изд-во Урал. ун-та, 2015. – 92 с.
6. Фейламазова, С. А. Информационные технологии в медицине / С.А. Фейламазова. — Махачкала: Эпоха, 2016. — 173 с.
7. Комплексная медицинская информационная система [Электронный ресурс] – Режим доступа: <https://www.kmis.ru/kmis> – (Дата обращения: 28.03.2024)
8. Начните работу с Bootstrap [Электронный ресурс]: – Режим доступа: <https://bootstrap-4.ru/docs/5.3/getting-started/introduction/> – (Дата обращения: 25.04.2024)
9. Приложение Draw.io [Электронный ресурс]: – Режим доступа: <https://app.diagrams.net/> – (Дата обращения: 25.04.2024)
10. Приложение Figma [Электронный ресурс]: – Режим доступа: <https://www.figma.com/> – (Дата обращения: 29.04.2024)
11. Руководство по PHP [Электронный ресурс]: – Режим доступа: <https://www.php.net/manual/ru/index.php> – (Дата обращения: 20.04.2024)
12. Что такое унифицированный язык моделирования? [Электронный ресурс] – Режим доступа: <https://www.lucidchart.com/pages/ru/uml> – (Дата обращения: 12.04.2024)
13. Электронная регистратура [Электронный ресурс] / Электрон. текст. данные. – 2016. – Режим доступа: <http://docplayer.ru/38569900-Elektronnaya-registratura.html> – (Дата обращения: 20.03.2024)
14. «Электронная регистратура» ЛПУ Волгоградской области [Электронный ресурс]: - Режим доступа: <https://astra.lru/> – (Дата обращения: 04.04.2024)
15. UI/UX-дизайн [Электронный ресурс]: – Режим доступа: <https://media.contented.ru/glossary/ux-dizajn/> – (Дата обращения: 10.04.2024)

**Федеральное казенное профессиональное образовательное учреждение «Калачевский техникум-интернат» Министерства труда и социальной защиты Российской Федерации**

# ПРИЛОЖЕНИЯ

**к выпускной квалификационной работе**

**РАЗРАБОТКА ПРОГРАММНОГО МОДУЛЯ «ЭЛЕКТРОННАЯ РЕГИСТРАТУРА»**

Специальность 09.02.03 Программирование в компьютерных системах

**Исполнитель:**

студент группы Т–43

Харитонов Иван Альбертович

**Научный руководитель:**

преподаватель профессионального цикла

Богатырева Маргарита Анатольевна

Калач-на-Дону, 2024