Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра экономической информатики

Дисциплина: Программирование сетевых приложений

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

**Система учетов пациентов в поликлинике**

БГУИР КП 1-40 05 01 02 026 ПЗ

Студент группы 873603 Хомич А.А.

Руководитель ассистент кафедры ЭИ

Жадинец

Минск 2022

**СОДЕРЖАНИЕ**

Введение 4

1 Использование информационных технологий в сфере здравоохранения 6

2 Постановка задачи по автоматизированию системы учёта оказания платных услуг поликлиникой и обзор методов её решения 12

2.1 Постановка задачи учёта оказания платных услуг поликлиникой 12

2.2 Обзор методов решения поставленной задачи 15

3 Функциональное моделирование процесса оказания платной услуги пациенту в поликлинике 15

4 Построение информационной модели системы учёта оказания платных услуг поликлиникой 20

5 Спецификация вариантов использования системы учёта оказания платных услуг поликлиникой 22

6 Модели представления системы учёта оказания платных услуг поликлиникой 24

6.1 Диаграмма состояний 26

6.2 Диаграмма последовательностей 26

6.3 Диаграмма классов 27

6.4 Диаграмма компонентов 28

6.5 Диаграмма развёртывания 28

7 Описание применения паттернов проектирования 28

8 Описание алгоритмов, реализующих бизнес-логику серверной части проектируемой системы 30

8.1 Схема алгоритма клиент-серверного соединения 30

8.2 Схема алгоритма поиска пользователя по фамилии 31

9 Руководство пользователя по развёртыванию системы учёта оказания платных услуг поликлиникой 32

10 Результаты тестирования разработанной системы учёта оказания платных услуг поликлиникой 39

Заключение 45

Список использованных источников 46

Приложение А (обязательное) Диаграммы UML 47

Приложение Б (обязательное) Схемы алгоритмов работы программы 50

# **ВВЕДЕНИЕ**

Информационные технологии в современном мире интенсивно развиваются и применяются повсеместно. Наиболее широко применяемые средства в наше время – это сотовая связь и интернет, мобильные телефоны и компьютеры. Тем не менее, каждая узкая отрасль науки и производства имеет своё специфическое оборудование, а также специально разработанное программное обеспечение, обеспечивающее работу устройств.

Внедрение современных информационных технологий в столь важной сфере, как здравоохранение, является не просто закономерным, это выводит здравоохранение на новую ступень развития, так как оперативный доступ к информации и обмен ею существенно сокращает временные затраты на поиск решений проблемы, а время зачастую является решающим фактором при спасении жизни человека [1].

Прогресс в информационных технологиях положительно сказался на развитии новых направлений организации медицинской помощи населению. Проведение онлайн-консультаций для пациентов, наблюдение и контроль в реальном времени, системы, позволяющие дистанционно фиксировать и транслировать физиологические параметры пациента – все эти и многие другие средства уже активно применяются в регулярной практике в сфере здравоохранения во многих развитых странах [2].

Очень важную роль играют различные электронные базы, в которых хранится информация о пациентах (истории болезни, результаты обследований), материальных ресурсах, трудовых ресурсах, данные о лекарственных препаратах, стандарты диагностики и лечения, а также экспертные системы. Интерфейс таких систем доступен и интуитивно понятен и неподготовленным пользователям, что помогает персоналу организаций здравоохранения без труда осваивать новые технологии.

Как показывает общемировая практика, внедрение информационных технологий в сферу здравоохранения позволяет повысить качество обслуживания пациентов и уменьшить их расходы, а также существенно упростить работу медицинского персонала. В настоящее время перечисленные преимущества становятся доступны многим медицинским учреждениям. Современные программные средства позволяют вывести сферу на принципиально новый уровень работы [3].

Общеизвестно, что в сфере услуг приходится иметь дело с огромными потоками информации. Здравоохранение исключением не является. Несмотря на широкое распространение новейших технологий в медицине, на сегодняшний день в поликлиниках нередко можно встретить такие случаи, когда учёт данных о пациентах, обследованиях и назначениях ведётся в бумажной форме. Естественно, при таком способе сохранения информации не исключены ошибки и потеря данных. Автоматизация является тем самым процессом, который может существенно облегчить указанные процессы и минимизировать вероятность ошибок.

Параллельно с экономическим прогрессом в нашей стране развивается практика оказания платных медицинских услуг. Поэтому становится резонным вопрос о создании такой системы, которая позволяла бы производить как учёт данных о самих пациентах, так и учёт оказанных им услуг. Именно этим и обусловлена актуальность выбранной темы.

Целью данного курсового проекта является сокращение непроизводительных трудозатрат сотрудников учреждений здравоохранения за счёт автоматизации процесса учёта платных услуг, оказываемых пациентам.

Разрабатываемая система должна обеспечить возможность вести учёт пациентов поликлиники, сохранять всю необходимую информацию о врачах, проведённых обследованиях и назначенных курсах лечения. Пользователю будут предоставлены специальные возможности для внесения информации в систему, а также для поиска интересующей его информации. Помимо этого, программное средство должно позволить систематизировать информацию об общем количестве и стоимости оказанных платных услуг.

Для достижения поставленной цели необходимо решить следующие задачи:

* осуществить анализ использования информационных технологий в сфере здравоохранения;
* проанализировать логическую и физическую модель представления данных;
* создать базу данных для учёта оказания платных услуг поликлиникой;
* реализовать серверную часть, которая позволит работать с базой данных и упростить работу сотрудников медицинских учреждений;
* разработать для пользователя простой и удобный интерфейс приложения;
* протестировать разработанное приложение;
* разработать и описать руководство пользователя.

# **1 ИСПОЛЬЗОВАНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В СФЕРЕ ЗДРАВООХРАНЕНИЯ**

Главное богатство любого государства – это его население. Люди создают государство и являются двигателем его развития и процветания. Но для того, чтобы иметь возможность активной деятельности, человек должен быть здоров. Поэтому одна из важнейших функций государства – поддержание здоровья своего населения. Такая поддержка осуществляется через систему здравоохранения.

Здравоохранение – это совокупность мер политико-правового, социально-экономического, научного, медицинского, санитарно-гигиенического и культурного характера, направленных на сохранение и укрепление физического и психического здоровья каждого человека, поддержание его долголетней активной жизни, предоставление ему медицинской помощи в случае ухудшения здоровья [4].

Система здравоохранения Республики Беларусь включает в себя следующие подразделения:

* государственная система здравоохранения;
* негосударственные организации здравоохранения;
* другие организации, которые наряду с основной деятельностью также осуществляют медицинскую или фармацевтическую деятельность в порядке, установленном законодательством РБ;
* индивидуальные предприниматели, осуществляющие медицинскую или фармацевтическую деятельность в порядке, установленном законодательством РБ.

Основой здравоохранения нашей страны является государственная система здравоохранения, состоящая из:

* Министерства здравоохранения РБ и подчиненных ему государственных организаций здравоохранения;
* управлений здравоохранения областных исполнительных комитетов и Комитета по здравоохранению Минского городского исполнительного комитета;
* органов управления здравоохранением других республиканских органов государственного управления и подчиненных им государственных организаций здравоохранения;
* органов управления здравоохранением государственных организаций, подчиненных Правительству, и подчиненных им государственных организаций здравоохранения;
* государственных организаций здравоохранения, подчиненных местным исполнительным и распорядительным органам;
* государственных учреждений, обеспечивающих получение медицинского (фармацевтического) образования и (или) повышение квалификации и переподготовку медицинских (фармацевтических) работников;
* государственных медицинских (фармацевтических) научных организаций;
* государственных организаций, обеспечивающих функционирование здравоохранения (медицинских информационно-аналитических центров, ремонтно-технических организаций, транспортных организаций и др.) [5].

При этом непосредственное оказание медицинской помощи населению осуществляют организации здравоохранения. К ним относятся поликлиники, больницы, диспансеры, амбулатории, фельдшерско-акушерские пункты и медицинские центры [6].

Наиболее популярным местом у граждан страны для получения медицинской помощи являются, безусловно, поликлиники. Это обусловлено, в первую очередь, их доступностью и приемлемой скоростью оказания необходимых услуг, в том числе и на платной основе.

Поликлиника – это термин, обозначающий многопрофильное учреждение здравоохранения, в котором оказывается амбулаторная (внебольничная) медицинская помощь пациентам, пришедшим на прием или же на дому.

В Беларуси поликлиники подразделяются по территориальному принципу. В зависимости от территории обслуживания, данные учреждения здравоохранения могут быть городские, районные и межобластные. Иногда они находятся на базе больниц. По возрастному принципу все поликлиники подразделяются на взрослые и детские.

В поликлинике оказывается первая медицинская и неотложная помощь в случае острого или внезапно развившегося заболевания, при травмах и отравлениях, проводится плановая диспансеризацию. В случае необходимости пациенту даётся направление на госпитализацию.

Кроме диагностики и лечения заболеваний в функции поликлиники входит экспертиза временной нетрудоспособности, медицинские освидетельствования, профилактическая работа с гражданами по предотвращению заболеваний и их осложнений, а также формирование положительного отношения к здоровому образу жизни. Выполняются противоэпидемические мероприятия: профилактические прививки и выявление инфекционных заболеваний.

В состав поликлиники, в зависимости от ее структуры, могут входить следующие специализированные кабинеты (отделения): терапевтический, хирургический, офтальмологический, оториноларингологический, неврологический, травматологический, эндокринологический, женская консультация, травматологический, стоматологический и многие другие [7].

Также поликлиника предоставляет услуги по проведению различных процедур как на бесплатной, так и на платной основе. В перечень таких процедур обычно входят рентгенография, электрокардиография (ЭКГ), ультразвуковое исследование органов (УЗИ), физиотерапия, общий анализ крови, биохимический анализ крови, вакцинация. Платные услуги становятся всё более популярными по причине более оперативного их оказания по сравнению с бесплатными.

Специалисты, работающие в поликлинике, ведут приём пациентов по талонам. Их заказывают в справочной службе регистратуры или с помощью установленных в фойе поликлиник инфокиосков. Помимо этого, заказать талон можно по интернету с помощью сервиса [Talon.by](https://talon.by/). Если талона нет, но срочно необходима консультация врача, следует обратиться к дежурному врачу. На рисунке 1.1 изображён пример талона, заказанного с помощью интернет-сервиса [Talon.by](https://talon.by/).

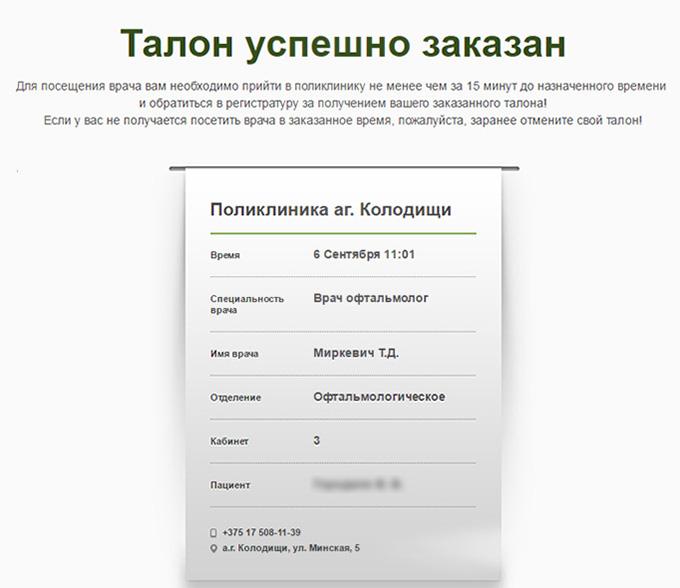




Рисунок 1.1 – Пример талона, заказанного с помощью интернет-сервиса Talon.by

В некоторых поликлиниках осуществляется приём с помощью электронного терминала (электронная очередь). В этом случае, предварительный заказ талона не требуется. Пациент самостоятельно берет талон с номером в терминале данного отделения. Электронное табло и звуковой сигнал пригласят пациента в соответствующий кабинет. Пример реализации системы электронной очереди приведён на рисунке 1.2. Но ни для кого не является секретом кадровый дефицит в поликлиниках, что ведёт к недостатку талонов и провоцирует огромные очереди в поликлиниках. Зачастую приём вынуждены вести молодые специалисты, у которых недостаточно опыта и им необходима консультация с более опытным специалистом для более точной постановки диагнозов.





Рисунок 1.2 – Пример реализации системы электронной очереди в поликлинике

Вдобавок к этому, текущий способ ведения учёта пациентов в поликлиниках объективно можно назвать устаревшим и несостоятельным. Сотрудники регистратуры вручную выписывают талоны на приём, переносят информацию о пройденных пациентом исследованиях в амбулаторную карту. Бухгалтера вынуждены получать информацию об оказанных платных услугах путём сбора кассовых чеков. Врачу выделяется всего 10-15 минут на осмотр пациента, изучение его анамнеза, назначение исследований или лечения. Разумеется, этого времени недостаточно, учитывая, что врач обязан внести запись в амбулаторную карту пациента, а также в свой журнал учёта и отчётную документацию.

Использование современных технологий позволит существенно сократить время на «бумажную» работу. Составление электронных историй болезни позволит каждому работнику системы здравоохранения моментально получать полную информацию обо всех заболеваниях пациента, отслеживать изменения таких показателей как частота сердечных сокращений, артериальное давление, уровень гемоглобина или сахара в крови, иметь представления, какие препараты принимает пациент и насколько они эффективны в конкретном случае. Это особенно удобно, если человеку срочно требуется медицинская помощь в другом городе (например, его сбила машина и он находится в коме), и нет никакой возможности узнать вышеперечисленную информацию.

Запись пациента на приём будет сопровождаться внесением необходимой информации в систему. В случае оказания платной услуги наряду со стандартными сведениями в систему будет вноситься стоимость оказанной услуги.

Исходя из вышеперечисленных проблем, с которыми сталкивается сфера здравоохранения, применение информационных технологий позволит решить следующие задачи:

* ведение учёта пациентов поликлиник;
* дистанционное наблюдение за состоянием больных;
* контроль назначенного способа лечения;
* сохранение и передача результатов диагностики;
* консультирование и обучение молодых специалистов;
* эффективное планирование работ и контроль их выполнения;
* ведение учёта лекарственных средств и медицинского оборудования и регистрация приходно-расходных операций;
* передача отчётной документации контролирующим органам.

Информатизация здравоохранения — это достаточно обширное понятие, которое также включает в себя мероприятия, направленные на информирование специалистов с помощью ИТ о научных достижениях в мире в области медицины. Таким образом, это является эффективным способом обучения и повышения квалификации персонала больниц и поликлиник.

С помощью таких технологий врачи могут быстро получать информацию о новых разработках и открытиях, которые помогут им работать эффективнее. Особенно актуальна эта проблема для медработников, которые трудятся в удалённых населенных пунктах.

На сегодняшний день некоторые сервисы уже нашли активное применение в медицинской сфере, например:

* функционирует интернет-сайт Talon.by, посредством которого можно заказать талон на любое удобное время, не обращаясь непосредственно в регистратуру поликлиники;
* функционирует интернет-сайт Tabletka.by, на котором есть можно узнать информацию о наличии в продаже и стоимости любого лекарственного препарата в любой аптеке Беларуси;
* во многих поликлиниках внедрена система электронной очереди;
* создана единая система электронных рецептов, в пределах которой взаимодействуют многие аптеки и поликлиники страны;
* проводятся врачебные консилиумы с участием опытных специалистов из специализированных научно-практических центров.

Руководствуясь выводами, сделанными при изучении предметной области, обозначим проблему, требующую автоматизации. Поликлиника должна хранить большие потоки данных о пациентах, об их заболеваниях, о назначенных исследованиях и их стоимости. Хранение указанной информации в бумажной форме является устаревшим и непрактичным. Поэтому создание автоматизированной системы учёта пациентов и оказания платных услуг поможет значительно упростить перечисленные выше процессы.

Таким образом, необходимо создать программное средство, которое будет отвечать следующим требованиям:

* пользователь должен затрачивать минимальное количество времени для записи пациента на приём;
* пользователю должны быть предоставлены широкие возможности для внесения и поиска необходимой информации о пациентах и стоимости оказанных услуг;
* интерфейс приложения должен быть понятным и удобным;
* для организации эффективного поиска должна быть создана база данных.

Такая система могла бы хранить большое количество данных в удобной для восприятия форме, позволять быстро производить запись пациента на приём без многократного ввода контактных данных, вносить всю необходимую информацию о пройденных обследованиях, а также вычислять общую стоимость оказанных услуг за определённый период времени.

# **2 ПОСТАНОВКА ЗАДАЧИ ПО АВТОМАТИЗИРОВАНИЮ СИСТЕМЫ УЧЁТА ОКАЗАНИЯ ПЛАТНЫХ УСЛУГ ПОЛИКЛИНИКОЙ И ОБЗОР МЕТОДОВ ЕЁ РЕШЕНИЯ**

**2.1 Постановка задачи учёта оказания платных услуг поликлиникой**

В данном курсовом проекте поставлена цель сократить трудозатраты сотрудников учреждений здравоохранения, которые ведут учёт оказания платных услуг. Для этого необходимо разработать такое приложение, которое предоставит возможность многим пользователям вносить и просматривать необходимую информацию. Пользоваться им смогут различные сотрудники учреждений здравоохранения. Например, медицинский регистратор будет иметь возможность оформлять пациентам талоны на приём к врачу. Врач сможет вносить информацию о проведённых приёмах и процедурах. Сотрудники финансового отдела смогут использовать приложение для получения информации о некоторых финансовых показателях, таких как помесячный доход или же общая сумма дохода за определённый промежуток времени.

Предлагается разработать приложение в архитектуре клиент-сервер. Так как необходимость воспользоваться приложением может возникать сразу у нескольких пользователей, сервер должен поддерживать возможность одновременно обрабатывать запросы этих пользователей. Реализация серверного приложения будет осуществлена в виде консольного приложения, в котором будет отображаться информация о подключениях клиентов и параметрах этих подключений. Это приложение должно производить обработку следующих запросов:

* авторизация пользователя;
* регистрация пользователя;
* получение списка пользователей;
* получение списка пациентов;
* получение списка врачей;
* получение списка процедур;
* получение списка оформленных талонов;
* получение списка проведённых приёмов пациентов;
* добавление новых записей в базу данных;
* редактирование существующих записей;
* удаление записей из базы данных;
* получение отчёта о сумме дохода за определённый промежуток времени;
* формирование текстового отчёта.

Для максимального удобства клиентское приложение должно быть реализовано в виде оконного приложения с графическим интерфейсом пользователя. Функционал администратора отличается от функционала простого пользователя, поэтому это будет отражено в клиентском приложении. Администратору будут доступны все функции для работы со списками пользователей, пациентов, врачей и процедур. Простой пользователь сможет работать с талонами, приёмами, а также просматривать статистику по оказанию платных услуг и доходу, полученному от их оказания.

**2.2 Обзор методов решения поставленной задачи**

Для программной реализации системы в данном курсовом проекте был выбран объектно-ориентированный язык программирования Java.

Основные преимущества Java как языка программирования:

* является объектно-ориентированным;
* платформонезависимость;
* простота;
* безопасность;
* портативность;
* интерпретируемость;
* высокая производительность.

Ещё одним неоспоримым преимуществом языка Java является его многопоточность. Эта функция позволяет писать программы, которые могут выполнять множество задач одновременно. Применение этой конструктивной особенности позволит реализовать все поставленные требования к функционалу системы.

Графический интерфейс клиентской части реализуется с помощью платформы JavaFX. Она предоставляет большие возможности по сравнению с рядом других подобных платформ, в частности, по сравнению со Swing. Это и большой набор элементов управления, и возможности по работе с мультимедиа, двухмерной и трехмерной графикой, декларативный способ описания интерфейса с помощью языка разметки FXML, возможность стилизации интерфейса с помощью CSS, интеграция со Swing и многое другое.

Для визуального редактирования файлов пользовательского интерфейса будет использоваться редактор SceneBuilder. Интерфейс создаётся с помощью файлов разметки FXML. Этот способ хорошо подходит для отделения контроллеров от представлений, а использование SceneBuilder позволит избежать прямой работы с XML.

Соединение между серверной и клиентскими частями должно устанавливаться с помощью протокола TCP/IP. Этот протокол обладает одним важным преимуществом: он обеспечивает аппаратную независимость. Так как в сетевых протоколах определяется только блок передачи и способ его отправки, TCP/IP не зависит от особенностей сетевого аппаратного обеспечения, позволяя организовать обмен информацией между сетями с различной технологией передачи данных. Система IP-адресов, в свою очередь, позволяет без затруднений установить соединение между любыми двумя машинами сети. Взаимодействие между устройствами в рамках стека TCP/IP осуществляется с помощью связки IP-адреса и номера порта.

Эта связка образует сокет – программный интерфейс, который обеспечивает обмен данными между устройствами на низком уровне. Протокол TCP/IP основывается на соединениях, устанавливаемых между двумя компьютерами, обычно называемых клиентом и сервером. Поэтому различают клиентский сокет и серверный сокет. Для организации взаимодействия клиент должен знать IP-адрес и номер порта сервера, по которым он осуществляет подключение к удаленному устройству.

Для хранения информации используется система управления реляционными базами данных MySQL. В реляционной базе данных информация хранится в отдельных таблицах, благодаря чему достигается выигрыш в скорости и гибкости. Таблицы связываются между собой при помощи отношений, благодаря чему обеспечивается возможность объединять при выполнении запроса данные из нескольких таблиц. Для упрощения работы с сервером MySQL используется MySQL Workbench. Он представляет графический клиент для работы с сервером, через который в удобном виде можно создавать, удалять, изменять необходимые таблицы и управлять их наполнением.

Описанные технологии и программные средства характеризуются в первую очередь надёжностью и простотой использования. Совместное их применение позволит создать удобное многофункциональное приложение, отвечающее всем поставленным требованиям.

# **3 ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ ПРОЦЕССА ОКАЗАНИЯ ПЛАТНОЙ УСЛУГИ ПАЦИЕНТУ В ПОЛИКЛИНИКЕ**

Проведённый анализ предметной области даёт возможность разработать функциональную модель процесса оказания платной услуги пациенту в поликлинике на основе методологии IDEF0.

На рисунке 3.1 представлена контекстная диаграмма верхнего уровня. Как можно заметить, на входе расположены пациент, денежные средства пациента и незаполненная амбулаторная карта. Необходимыми для успешного выполнения процесса ресурсами являются медицинский регистратор и врач. Управление процессом приёма пациента в поликлинике осуществляется нормативно-правовыми актами Министерства здравоохранения РБ, а также внутренним регламентом поликлиники. Результатом выполнения процесса являются пациент, получивший услугу, прибыль поликлиники и заполненная амбулаторная карта пациента.

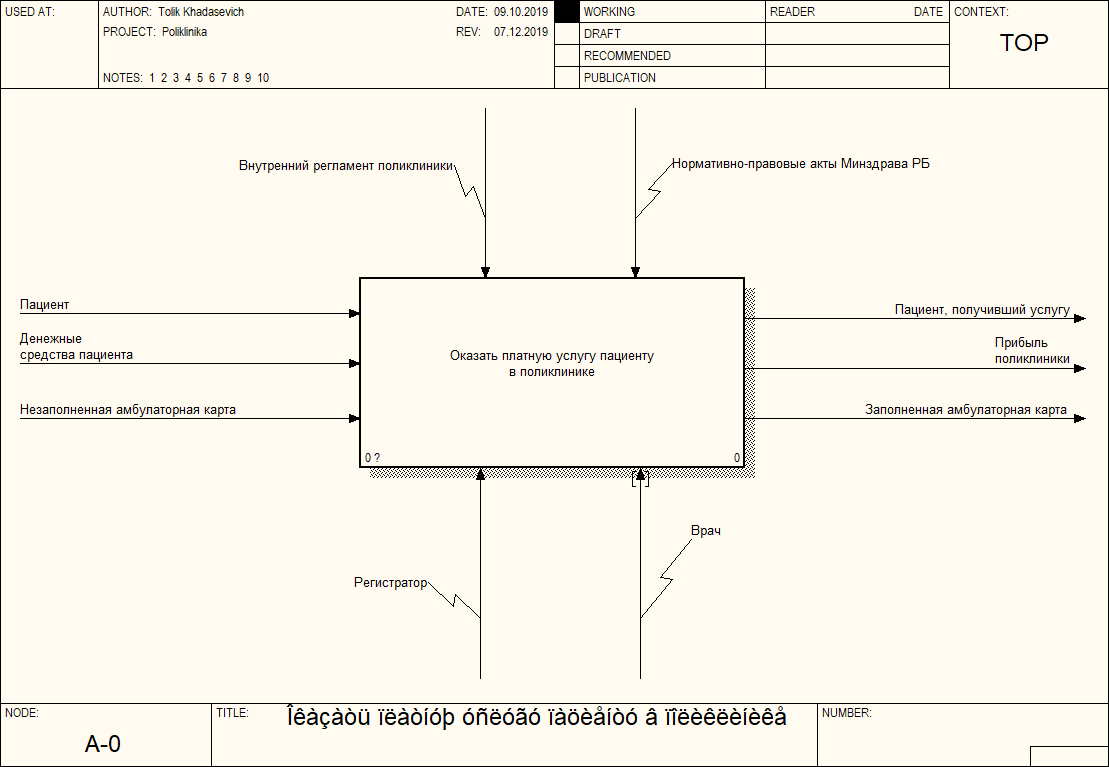


Рисунок 3.1 – Контекстная диаграмма верхнего уровня

После декомпозиции контекстной диаграммы получаем следующие блоки, показанные на рисунке 3.2:

* выбрать дату и время приёма;
* записать пациента на приём;
* провести приём пациента;
* внести запись в амбулаторную карту.

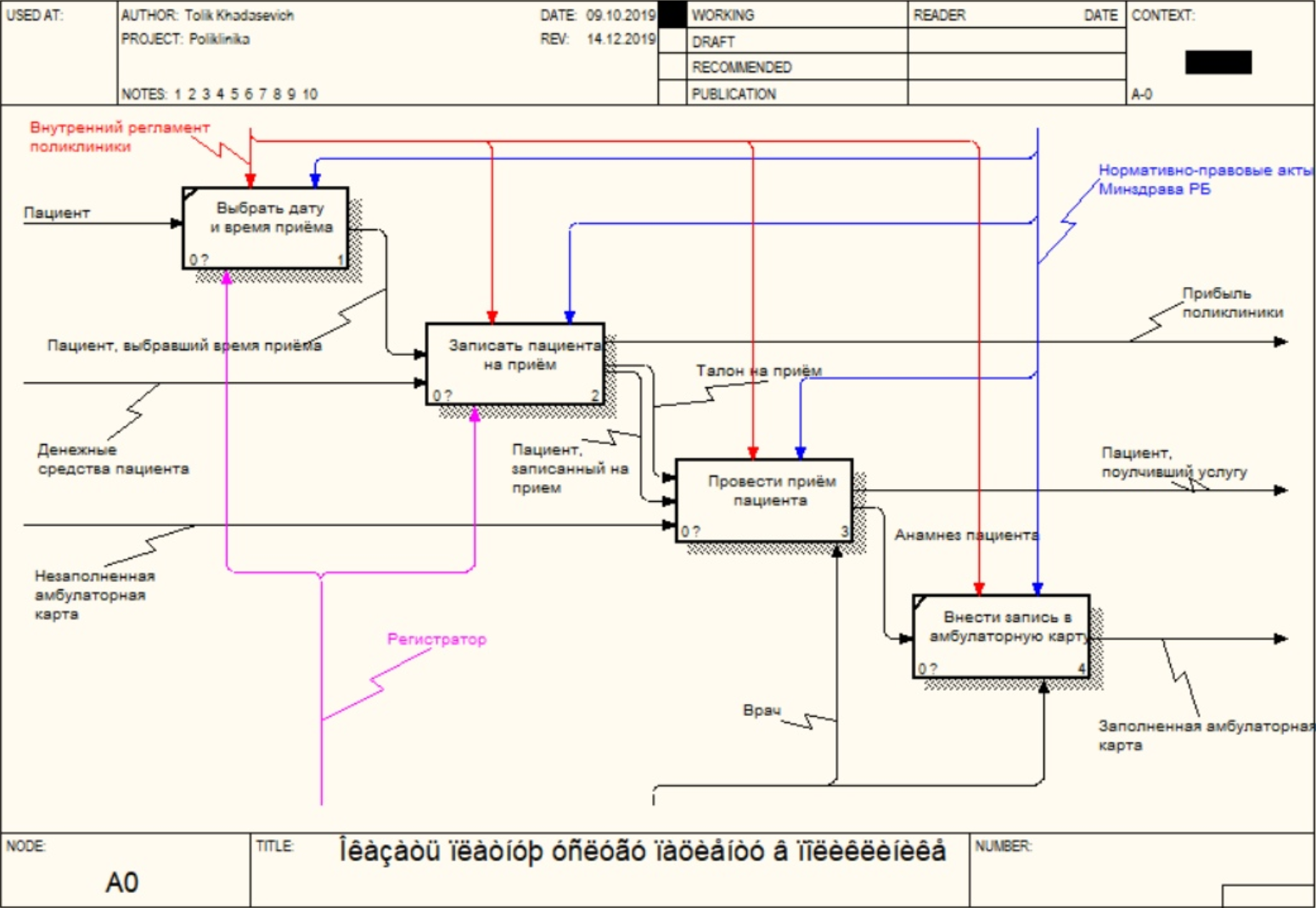


Рисунок 3.2 – Декомпозиция контекстной диаграммы верхнего уровня

Когда медицинский регистратор получает от обратившегося в регистратуру пациента информацию о желаемых дате и времени приёма, он может записать пациента на приём и зафиксировать это документально в форме талона. После этого он обязан принять от пациента оплату услуги. Результатом выполнения блока «Записать пациента на приём» является прибыль поликлиники, талон на приём и записанный пациент. Последние два элемента поступают на вход в блок «Провести приём пациента», наряду с незаполненной амбулаторной картой. Выполнение этого блока осуществляет врач. В ходе приёма пациента врач получает анамнез, который далее вносит в амбулаторную карту. В результате выполнения всех описанных действий пациент получает необходимую ему услугу, заполняется его амбулаторная карта, а поликлиника получает прибыль, что и является главной целью осуществления коммерческой деятельности.

Декомпозиция блока «Записать пациента на приём» содержит три блока:

* + внести данные в журнал предварительной записи;
  + оформить талон;
  + принять оплату услуги от пациента.

Данная декомпозиция отражена на рисунке 3.3.

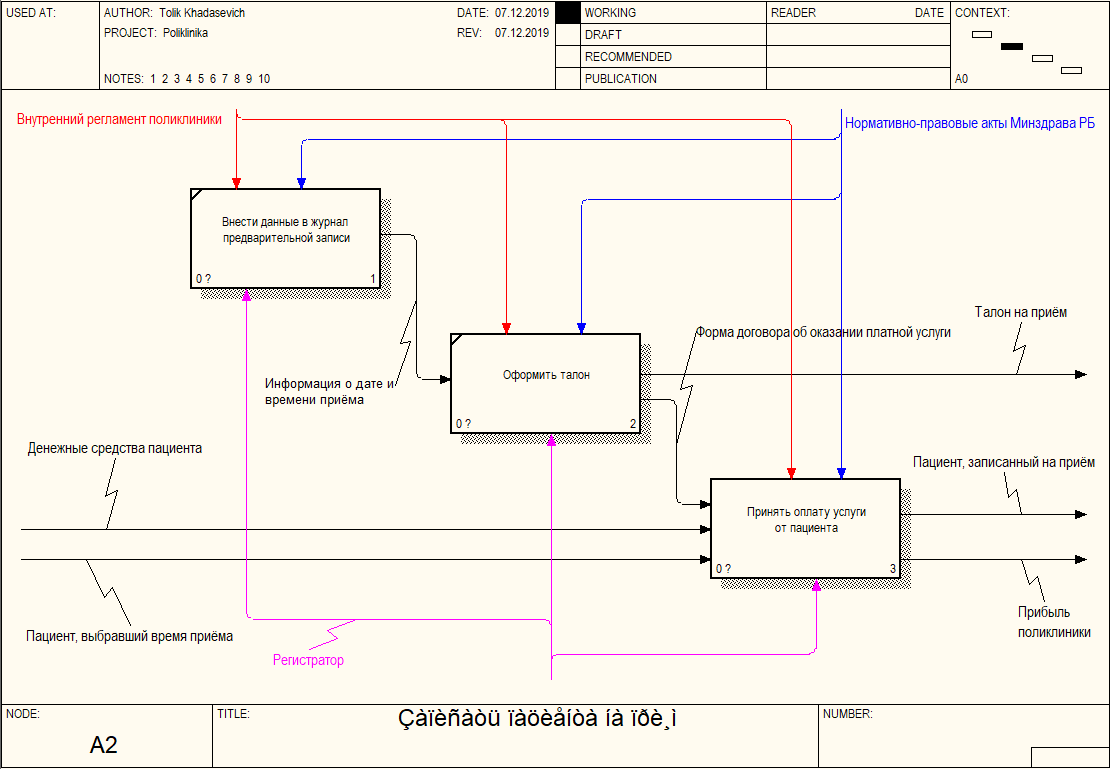


Рисунок 3.3 – Декомпозиция блока «Записать пациента на приём»

Процесс записи пациента на приём начинается с внесения его данных в журнал предварительной записи. Далее происходит оформление талона – документа, на основании которого пациент прибывает на приём в указанное время. В талоне указывается фамилия пациента, фамилия врача, ведущего приём, а также дата и время приёма. Так как в данном проекте акцент делается на оказании платных услуг, то вместе с талоном должен оформляться стандартный договор об оказании платной услуги. Форма такого договора поступает на вход в блок «Принять оплату услуги от пациента». Результатом выполнения этой процедуры является полученная поликлиникой прибыль. Механизм выполнения всех действий, связанных с записью пациента на приём, осуществляет медицинский регистратор. При этом он руководствуется внутренним регламентом поликлиники, а также нормативно-правовыми актами Министерства здравоохранения РБ.

Рассмотрим декомпозицию блока «Принять оплату услуги от пациента» (рисунок 3.4). В результате её проведения получаются три блока:

* заполнить договор об оказании платной услуги;
* оформить кассовый чек о получении денежных средств;
* внести информацию в журнал учёта платных услуг.

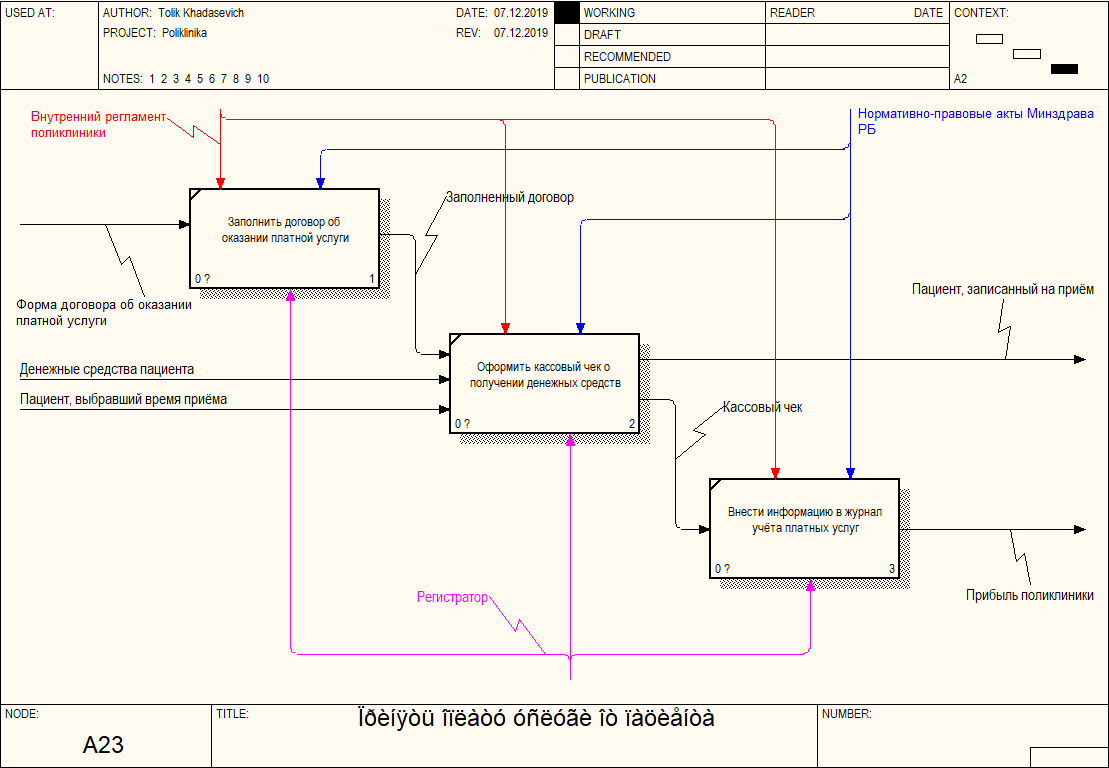


Рисунок 3.4 – Декомпозиция блока «Принять оплату услуги от пациента»

На первом этапе происходит заполнение стандартной формы договора. Заполненный договор поступает на вход в блок «Оформить кассовый чек о получении денежных средств». После принятия необходимой суммы от пациента составляется кассовый чек. Чек выдаётся пациенту, а на основании договора об оказании платной услуги вносится необходимая информация в журнал учёта платных услуг. Все эти действия осуществляются медицинским регистратором.

В результате выполнения описанных процессов систематизируется информация об оказанных платных услугах и прибыли, полученной от их оказания. Полученные данные могут быть полезными при составлении отчётов о работе и планировании дальнейшей деятельности.

# **4 ПОСТРОЕНИЕ ИНФОРМАЦИОННОЙ МОДЕЛИ СИСТЕМЫ УЧЁТА ОКАЗАНИЯ ПЛАТНЫХ УСЛУГ ПОЛИКЛИНИКОЙ**

В процессе информационного моделирования были выделены следующие сущности:

* Пользователь;
* Врач;
* Пациент;
* Талон;
* Приём;
* Процедура.

Сущность «Пользователь» служит для предоставления возможности входа в систему. Наличие такой сущности позволяет разграничить пользователей системы по типу (администратор или простой пользователь) и тем самым предоставить каждому пользователю необходимый ему функционал. Атрибутами этой сущности являются логин, пароль, тип и уникальный ID для каждого пользователя.

Из своего личного кабинета пользователь, то есть медицинский регистратор, может записывать пациентов на приём к врачу. Для этого выделена сущность «Талон». Так как талон оформляется для конкретного пациента и к конкретному врачу, эта сущность связана с сущностями «Пациент» и «Врач». Это даёт возможность выбирать врачей и пациентов из существующих списков. Регистратор выбирает необходимого врача по личному ID и оформляет талон на определённые дату и время.

Для хранения списка врачей, работающих в поликлинике, предназначена сущность «Врач», содержащая следующие атрибуты: личный ID, фамилия, имя, отчество и специализация врача. Информация в эту сущность заносится при создании базы данных, а также при необходимости может быть изменена администратором.

Сущность «Пациент» нужна для хранения информации о пациентах, которых обслуживает поликлиника. В эту сущность сведения также заносятся при создании базы данных и могут быть изменены администратором.

Врач может заносить в систему информацию о проведённом приёме пациента. С этой целью выделена сущность «Приём». Фиксируются сведения о характере пройденного обследования, диагнозе и назначениях для пациента, а также стоимости пройденного обследования. Это даёт возможность подсчитать суммарную прибыль от оказанных услуг.

Для хранения информации об услугах, предоставляемых поликлиникой, выделена сущность «Процедура». Атрибутами этой сущности являются наименование процедуры, её описание и стоимость выполнения данной процедуры. Эта сущность связана с сущностью «Приём». Такая связь позволяет выбирать процедуру, которая была проведена на приёме, с помощью уникального идентификационного номера каждой процедуры.

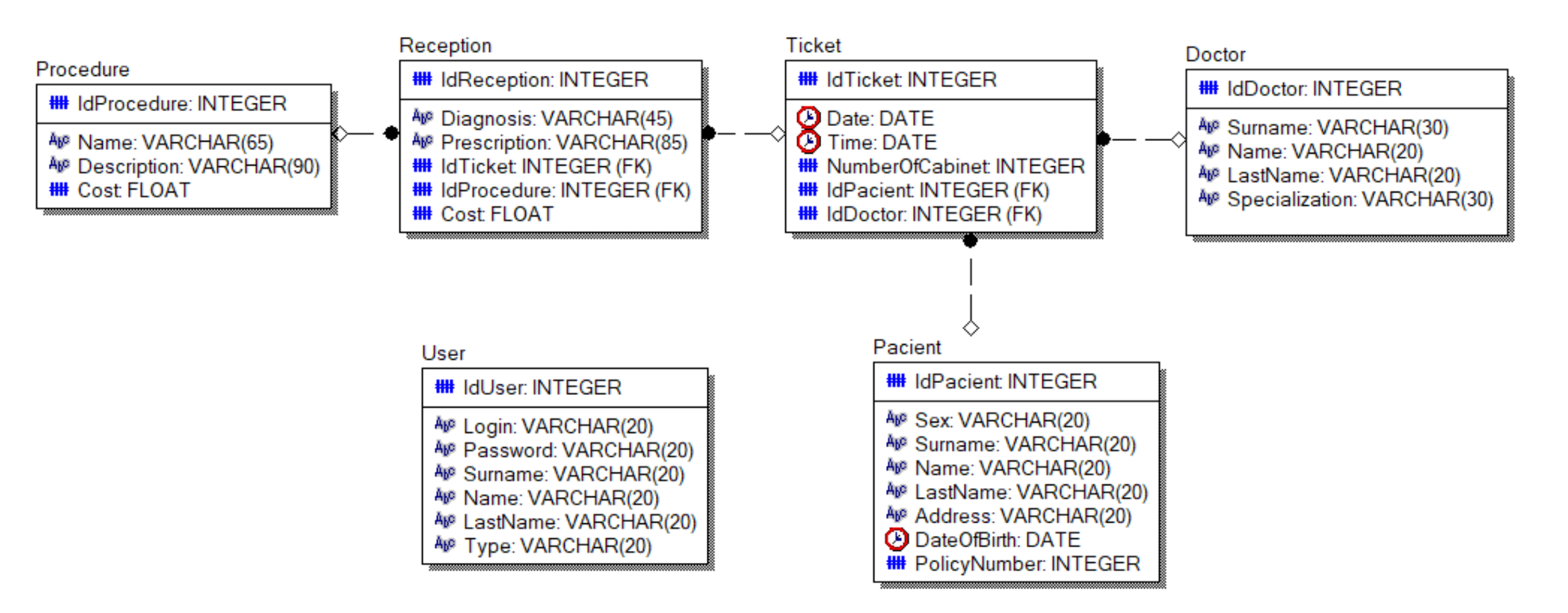
****

Рисунок 4.1 – Информационная модель базы данных

С учётом обозначенного взаимодействия сущностей смоделируем их взаимодействие в формате IDEF1.X и приведём эту модель к третьей нормальной форме. В результате последовательного приведения получается модель, соответствующая условиям третьей нормальной формы – не ключевой атрибут сущности функционально зависит только от всего первичного ключа и ни от чего другого (рисунок 4.1).

# **5 СПЕЦИФИКАЦИЯ ВАРИАНТОВ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ УЧЁТА ОКАЗАНИЯ ПЛАТНЫХ УСЛУГ ПОЛИКЛИНИКОЙ**

Суть диаграммы вариантов использования состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования. При этом актером или действующим лицом называется любая сущность, взаимодействующая с системой извне. Это может быть человек, техническое устройство, программа или любая другая система, которая может служить источником воздействия на моделируемую систему так, как определит сам разработчик. В свою очередь, вариант использования служит для описания сервисов, которые система предоставляет актеру. Диаграмма вариантов использования разрабатываемой системы представлена на рисунке 5.1.

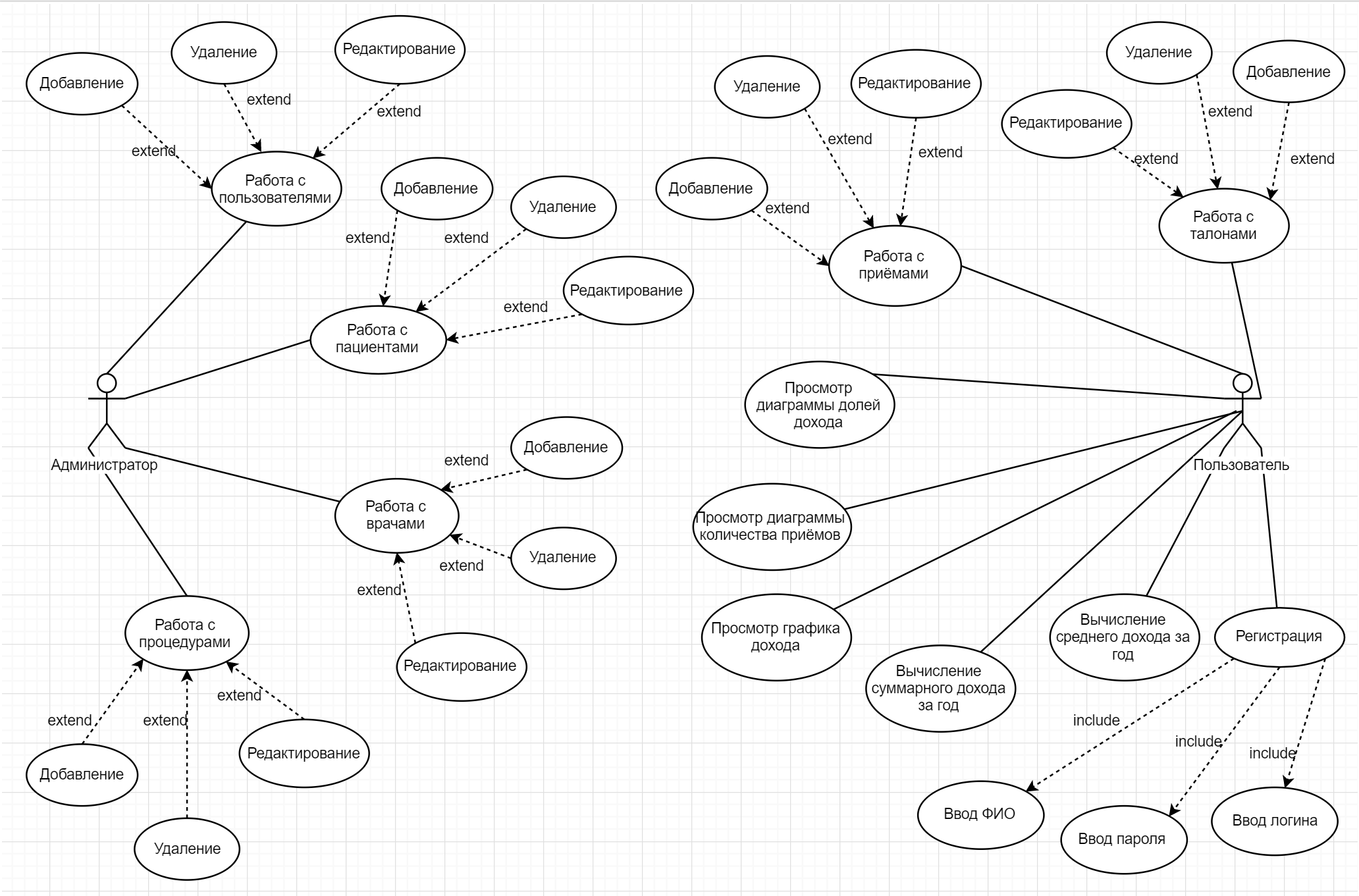


Рисунок 5.1 – Диаграмма вариантов использования

В данном проекте были выделены 2 актёра – администратор и пользователь.

Администратор обладает четырьмя вариантами использования – работа с пользователями, работа с пациентами, работа с врачами и работа с процедурами. Каждый из этих вариантов использования расширяется ещё тремя – добавлением, удалением и редактированием.

Пользователю доступны восемь вариантов использования. Два из них, работа с талонами и работа с приёмами, созданы по аналогии с вариантами использования администратора и расширяются таким же комплексом вариантов использования – добавление, удаление и редактирование. Помимо этого, пользователь может просмотреть три вида графиков – долей дохода, количества приёмов и помесячный график прибыли. Также пользователь может осуществить функцию вычисления общего дохода за текущий год и среднего размера дохода.

# **6 МОДЕЛИ ПРЕДСТАВЛЕНИЯ СИСТЕМЫ УЧЁТА ОКАЗАНИЯ ПЛАТНЫХ УСЛУГ ПОЛИКЛИНИКОЙ**

**6.1 Диаграмма состояний**

Диаграмма состояний является широко известным средством описания поведения систем. Она определяет все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате влияния некоторых событий. Данная диаграмма полезна при моделировании жизненного цикла объекта. Диаграмма состояний разрабатываемой системы представлена на рисунке А.1.

На начальном этапе работы системы происходит ожидание обращения пациента. После принятия этого обращения осуществляется выбор приемлемой даты и времени приёма. Затем можно произвести заполнение стандартной формы талона. После получения заполненной формы талона система переходит в режим ожидания оплаты. Далее происходит проверка оплаты: в случае её отсутствия система возвращается к состоянию ожидания оплаты. Если же оплата была произведена, то талон может быть распечатан. На заключительном этапе работы системы происходит передача талона пациенту. Как итог, на выходе получается оформленный талон.

**6.2 Диаграмма последовательностей**

Эта диаграмма является видом диаграмм взаимодействия, которые описывают отношения объектов в различных условиях. Условия взаимодействия задаются сценарием, полученным на этапе разработки диаграмм вариантов использования. Основываясь на диаграмме, разработанной и описанной в главе 5, была построена диаграмма последовательностей, которая показана на рисунке А.2.

Начинается работа с ввода пользователем логина и пароля. Эти данные передаются на сервер, и сервер выполняет запрос к базе данных, тем самым проверяя наличие такого пользователя. Если в базе данных такой пользователь находится, то происходит авторизация, серверу возвращается результат этой авторизации, а перед пользователем открывается пользовательское меню.

На следующем этапе пользователь вводит информацию о талоне, информация отправляется на сервер, выполняется запрос на добавление, и запись о талоне с соответствующими атрибутами добавляется в базу данных. База данных отправляет серверу результат операции добавления, а пользователь получает сообщение об успешном добавлении записи.

Для осуществления подсчёта дохода от оказания платных услуг, реализованной в приложении, пользователь совершает необходимый выбор в меню, после этого сервер, получив критерий подсчёта, осуществляет специализированный запрос к базе данных. В ответ сервер получает искомый результат, и отправляет пользователю сообщение с суммой дохода.

**6.3 Диаграммы классов**

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от уровня абстракции: классы могут представлять сущности предметной области или элементы программной системы. В связи с тем, что все основные функции системы (работа с базой данных, бизнес-функции) реализованы на серверной части, в то время как на клиентской части описана только логика работы с визуальными элементами приложения, то имеет смысл построить диаграмму классов серверной части. Она представлена на рисунке А.3.

Класс ServerMain содержит единственный метод main, который осуществляет запуск серверного приложения путём создания объекта класса Server.

В свою очередь, в классе Server создаются серверный и клиентский сокеты, а метод accept ожидает подключений клиентов. После подключения клиента создаётся объект класса ClientHandler.

Класс ClientHandler для каждого подключившегося клиента создаёт потоки ввода-вывода, а также предоставляет каждому клиенту персональный поток Thread для дальнейшей работы с приложением. В этом классе описаны вызовы всех функций, реализующих бизнес-логику системы, при поступлении соответствующих команд со стороны клиента.

В классе DatabaseOptions заданы параметры для подключения к базе данных. К этим параметрам относятся: IP-адрес, номер порта, имя пользователя базы данных, пароль для доступа, а также имя базы данных, с которой необходимо установить соединение.

От класса DatabaseOptions наследуется класс DatabaseHandler, в котором описаны все методы работы с базой данных. В начале с помощью метода getDbConnection стандартного класса java.sql.Connection устанавливается сеанс работы с базой данных. В других методах класса с помощью SQL-запросов реализован механизм выполнения CRUD-операций для всех сущностей системы. Также реализованы несколько методов, необходимых для выполнения бизнес-функций, например, подсчёт общего дохода и подсчёт среднего дохода поликлиники.

Класс DatabaseConst необходим для соотнесения идентификаторов, используемых в программе, с названиями атрибутов таблиц в базе данных.

В классе DataObject описаны геттеры и сеттеры для всех идентификаторов из класса DatabaseConst. Класс DataObject реализует интерфейс Serializable, так как создаваемые объекты этого класса должны быть сериализованы. Это необходимо для передачи таких объектов в потоки ввода-вывода.

**6.4 Диаграмма компонентов**

Эта диаграмма позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами. В качестве таких компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и так далее. Рассмотрим диаграмму компонентов для системы учёта оказания платных услуг поликлиникой, которая изображена на рисунке А.4.

Модуль ClientMain.jar отвечает за функционирование клиентской части приложения. Этот модуль содержит FXML-файлы, а также относящиеся к ним контроллеры.

Модуль ServerMain.jar реализует работу серверной части приложения. Компоненты этого модуля отвечают за установку клиент-серверного соединения, обработку запросов клиента и работу с модулем базы данных. Для взаимодействия с базой данных необходим сервер MySQL. Приложение устанавливает связь с базой данных посредством драйвера JDBC.

**6.5 Диаграмма развёртывания**

Диаграмма развертывания показывает топологию системы и распределение компонентов системы по её узлам, а также соединения – маршруты передачи информации между аппаратными узлами. На рисунке А.5 показана диаграмма развёртывания разрабатываемой системы.

Как можно заметить, основными элементами являются клиентский персональный компьютер и сервер приложения. В рамках данной системы к клиентским терминалам можно отнести компьютеры, установленные в регистратуре поликлиники, а также компьютеры, установленные в кабинетах врачей.

Сервер включает в себя исполняемый файл и систему управления базами данных MySQL. Взаимодействие между клиентскими компьютерами и серверным узлом осуществляется посредством стека протоколов TCP/IP.

# **7 ОПИСАНИЕ ПРИМЕНЕНИЯ ПАТТЕРНОВ ПРОЕКТИРОВАНИЯ**

Паттерн проектирования — это типовое решение распространённой проблемы при проектировании архитектуры программ. Паттерн представляет собой не конкретный код, а общую концепцию решения проблемы, которую необходимо подстраивать под нужды программы.

При проектировании системы был использован конструкционный шаблон MVC (Model-View-Controller). Этот шаблон разделяет работу приложения на три отдельные функциональные роли: модель данных (Model), пользовательский интерфейс (View) и управляющую логику (Controller). Таким образом, изменения, вносимые в один из компонентов, оказывают минимально возможное воздействие на другие компоненты. В данном паттерне модель не зависит от представления или управляющей логики, что делает возможным проектирование модели как независимого компонента. Схема данного шаблона представлена на рисунке 7.1.

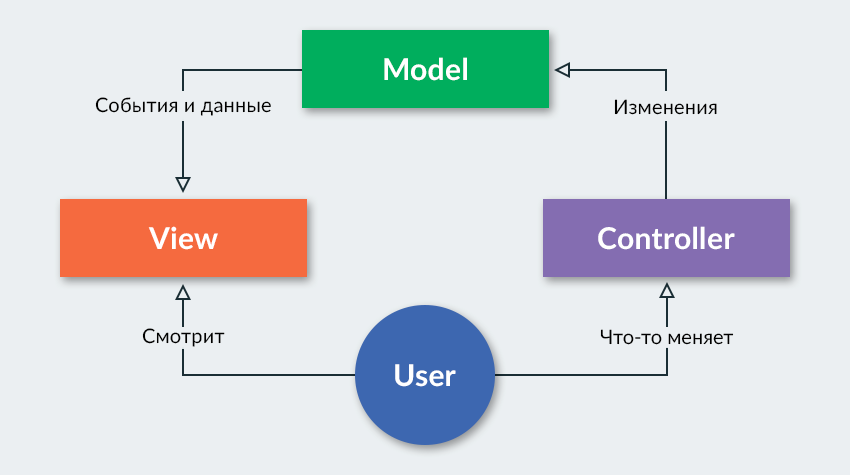


Рисунок 7.1 – Схема шаблона MVC

Первая часть содержит всю бизнес-логику приложения. Эта часть называется Модель (Model). В ней содержится код, который делает все то, для чего приложение создавалось. В данном проекте такой моделью является класс DataObject.

Вторая часть содержит все, что касается отображения данных пользователю. Она называется Вид (View). В ней содержится код, который управляет показом окон и сообщений. В данном проекте эту часть реализовывают FXML-файлы, в которых описаны все визуальные элементы приложения.

Третья часть содержит код, который занимается обработкой действий пользователя. Любые действия пользователя, направленные на изменения модели, должны обрабатываться в этой части. Такая часть называется Controller. Для каждого FXML-файла в данном проекте реализован свой контроллер.

# **8 ОПИСАНИЕ АЛГОРИТМОВ, РЕАЛИЗУЮЩИХ БИЗНЕС-ЛОГИКУ СЕРВЕРНОЙ ЧАСТИ ПРОЕКТИРУЕМОЙ СИСТЕМЫ**

**8.1 Схема алгоритма клиент-серверного соединения**

В данном курсовом проекте реализована архитектура «клиент-сервер». К преимуществам данной архитектуры, как правило, относят высокую защищённость данных, централизацию доступа к хранимым данным, быстродействие, быстроту обслуживания и обработку данных. Схема алгоритма клиент-серверного-соединения представлена на рисунке Б.1.

Программное средство разработано таким образом, что графический интерфейс реализуется на клиентской части, которая формирует и отправляет запросы серверу. Сервер же, в свою очередь, формирует запросы к базе данных. После выполнения запросов результаты возвращаются на клиентскую часть.

Работа программы начинается с установления соединения с сервером. Затем происходит проверка корректности установленного соединения. Далее, при условии успешного подключения к серверу, происходит переход к авторизации. В случае, если пользователь не имеет учётной записи в системе, то ему необходимо создать её. Это делается с помощью функции регистрации. Информация, введённая пользователем, сохраняется в базу данных. После создания учётной записи пользователь может произвести вход в систему. Сделав это, у него появляется возможность совершить какое-то действие, например просмотреть список заказанных талонов. Для этого необходимо войти в соответствующее меню. При этом будет произведено обращение к серверу, который, в свою очередь, получит необходимую выборку из базы данных и вернёт полученные результаты обратно клиенту. Если полученные результаты устраивают пользователя, то тогда он может завершить работу с системой. Однако перед этим будет разорвано соединение с сервером.

**8.2 Схема алгоритма поиска пользователя по фамилии**

Выполнение этого алгоритма начинается с ввода пользователем критерия поиска. Сервер получает это значение и создаёт SQL-запрос к базе данных. Затем происходит установка соединения с базой данных. После этого сервер отправляет созданный запрос и ждёт получения ответа. Ответ приходит в виде выборки записей из соответствующей таблицы базы данных. В случае, если в таблице записей не оказалось, то и полученная выборка будет пуста. Поэтому проводится проверка, не пустой ли пришёл список записей. Если он оказался пуст, то будет выведено соответствующее сообщение и после отправки пустого ответа работа алгоритма будет завершена.

Если же серверу удалось извлечь выборку записей, то создаётся объект ответа клиенту, к которому присоединяется извлечённый по заданному критерию список записей. Этот отправляется клиенту и на этом работа алгоритма завершается.

# **9 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ ПО РАЗВЁРТЫВАНИЮ СИСТЕМЫ УЧЁТА ОКАЗАНИЯ ПЛАТНЫХ УСЛУГ ПОЛИКЛИНИКОЙ**

Согласно требованиям к поставке, разработанная система должна обладать следующей инфраструктурой:

* в качестве базы данных используется СУБД MySQL версии 8.0;
* база данных генерируется SQL-скриптом;
* взаимодействие между серверной и клиентскими частями осуществляется с использованием сокетов и протокола TCP/IP;
* приложение поставляется в виде JAR-архива.

Для проверки работоспособности приложения необходимо создать базу данных и заполнить её тестовыми данными. Скрипт создания базы приведён в приложении Г. Этот скрипт необходимо скопировать в зону SQL-запросов среды MySQL Workbench, а затем выполнить его. На рисунке 8.1 приведён пример правильного расположения скрипта в рабочей зоне СУБД, а курсор наведён на кнопку, по нажатии которой происходит выполнение скрипта.

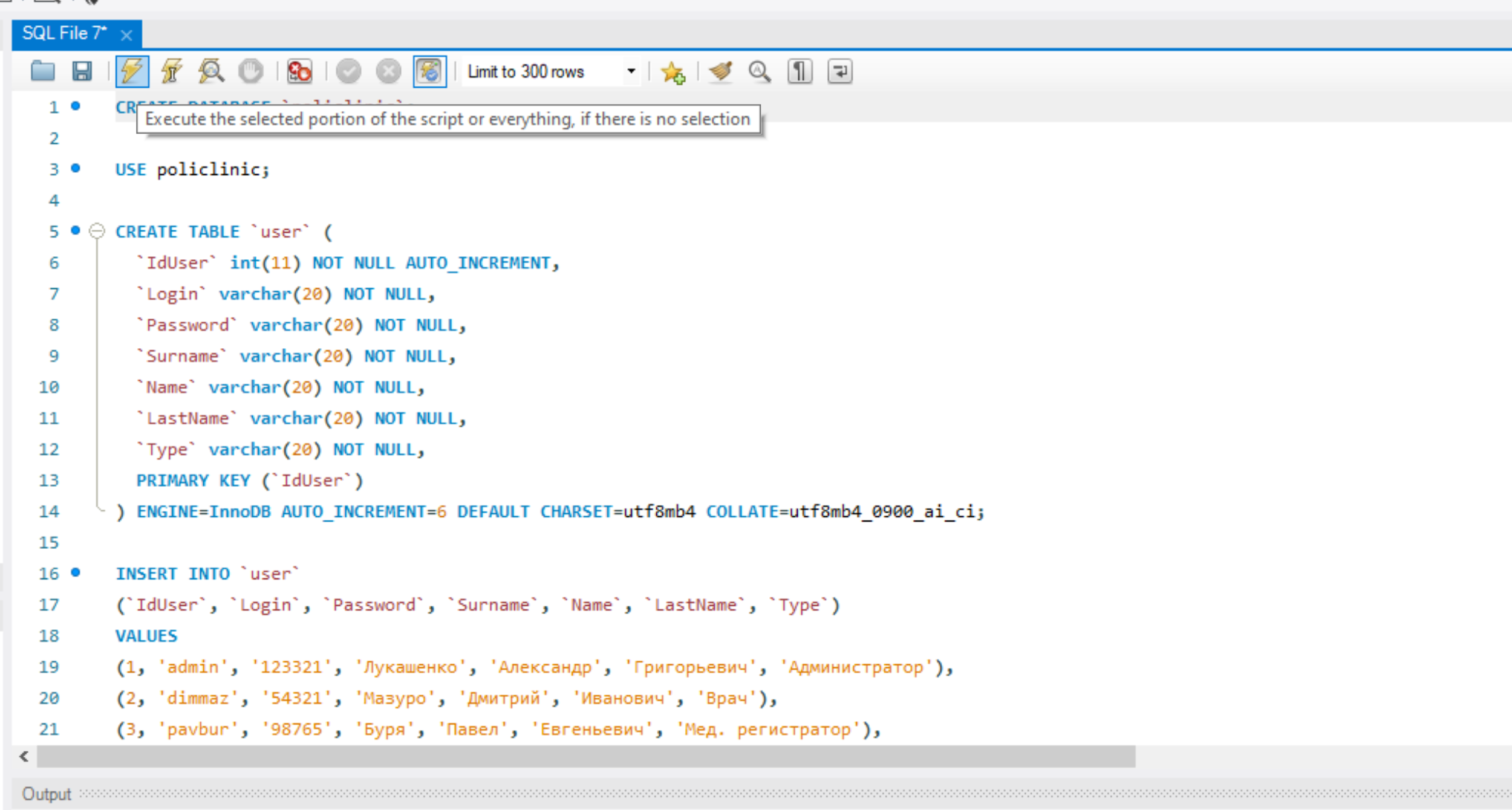


Рисунок 8.1 – Расположение скрипта в среде MySQL

Перед началом работы приложения необходимо установить соединение с сервером посредством запуска серверной части. Если оно успешно установлено, то можно совершить запуск клиентской части программного средства. Работа начинается с предложения совершить вход в систему (рисунок 8.2). Для этого пользователю необходимо ввести свои логин и пароль.

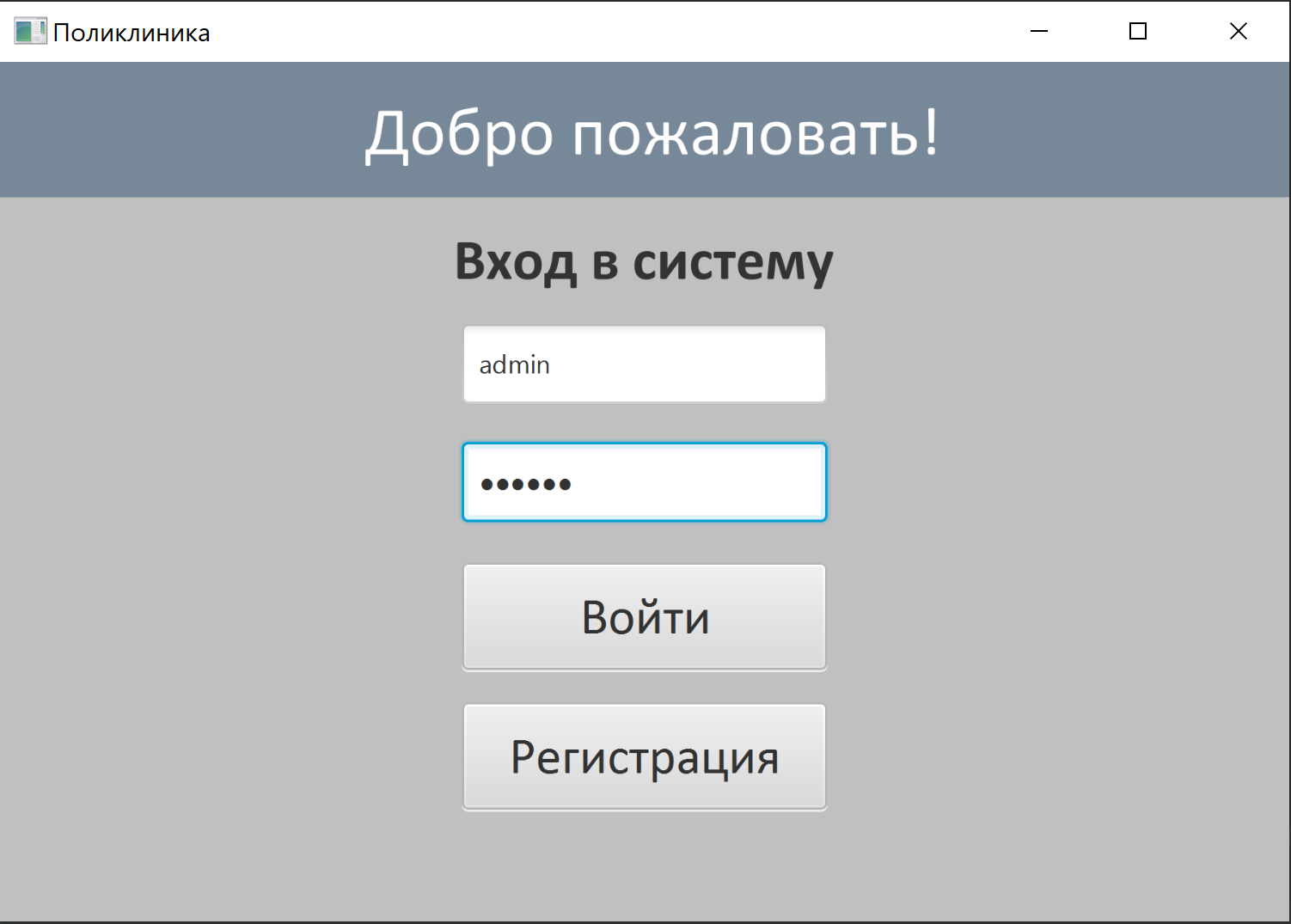


Рисунок 8.2 – Окно авторизации

В случае ввода неверного логина или пароля будет выдано сообщение об ошибке, представленное на рисунке 8.3.

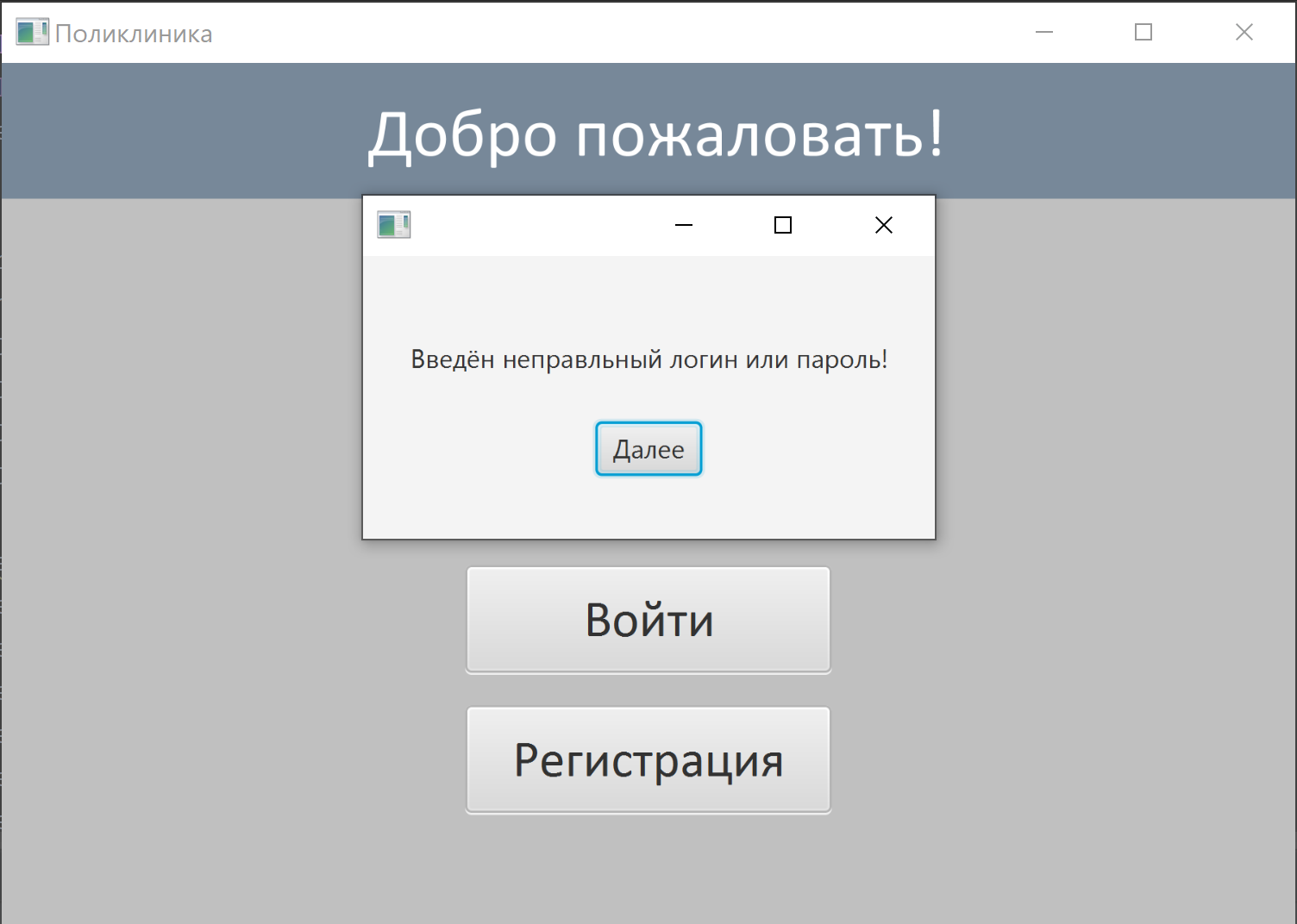


Рисунок 8.3 – Ошибка при входе в систему

Если пользователь не имеет учётной записи в системе, то ему необходимо зарегистрироваться. Это делается путём нажатия на кнопку «Регистрация» и дальнейшего ввода своих данных в окне, изображённом на рисунке 8.4.

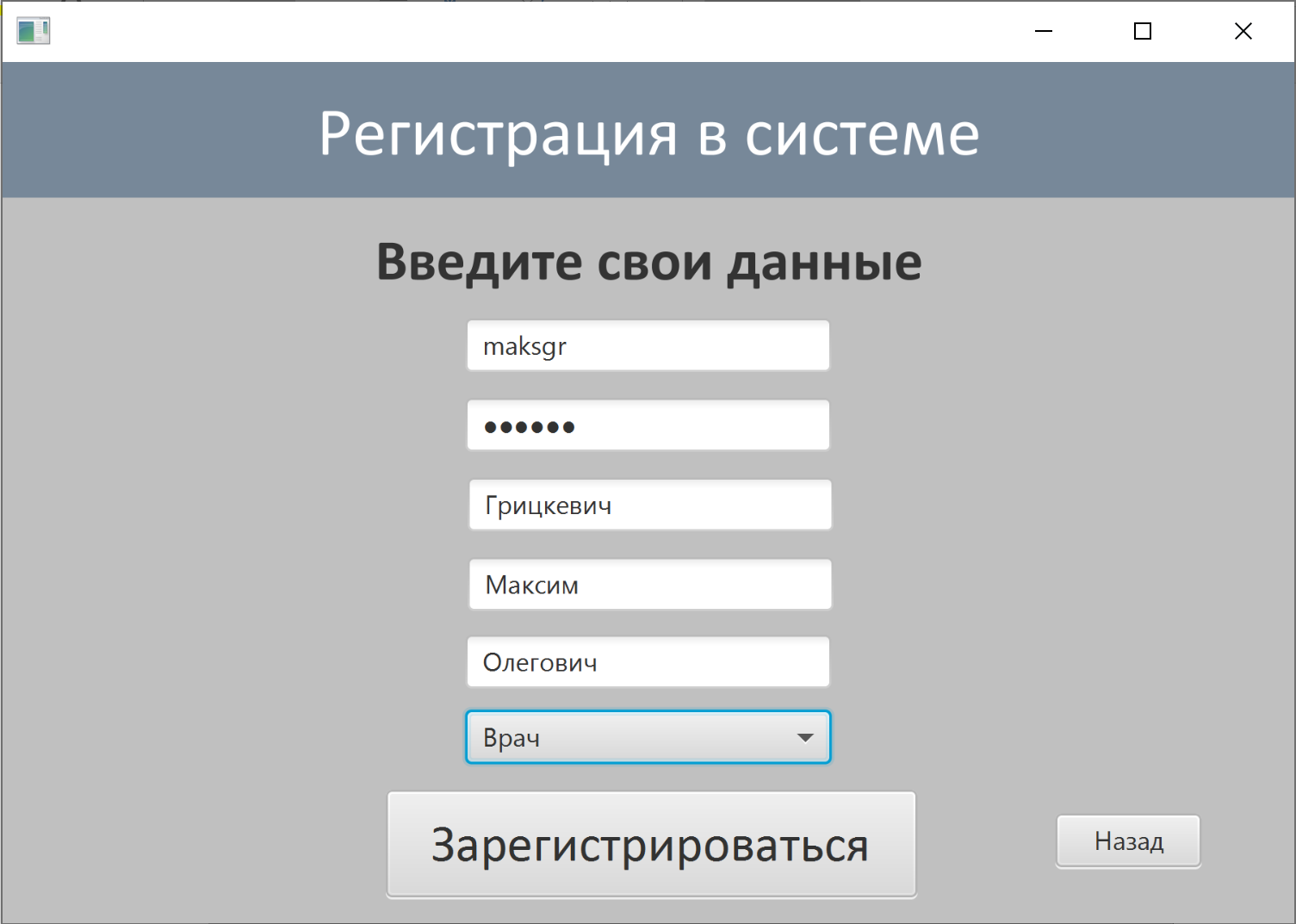


Рисунок 8.4 – Окно регистрации в системе

Для входа в качестве администратора необходимо ввести логин «admin» и соответствующий пароль. После успешного входа в качестве администратора на экран выведется окно, показанное на рисунке 8.5.

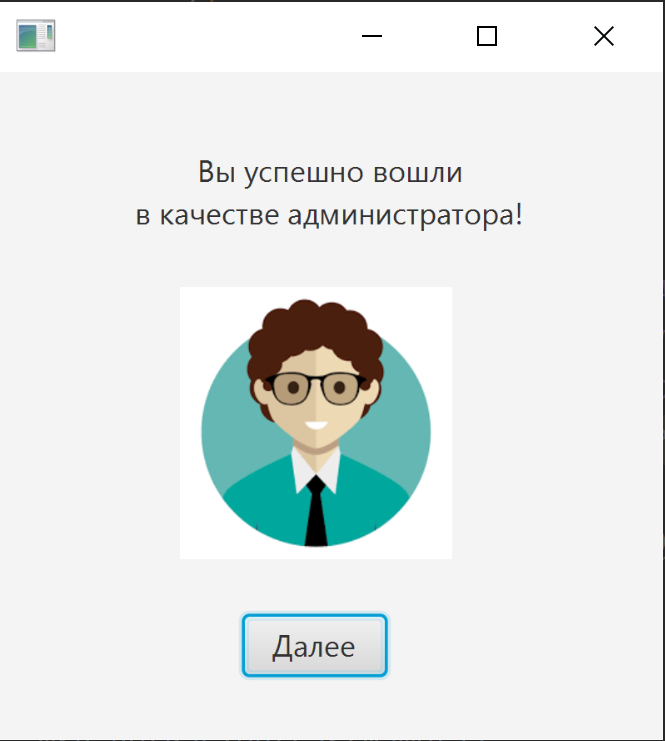


Рисунок 8.5 – Успешная авторизация администратора

По нажатию на кнопку «Далее» происходит переход в меню, которое представлено на рисунке 8.6. Администратор обладает полномочиями по работе с другими пользователями системы, со списками пациентов, врачей и процедур. В зависимости от своего выбора администратор попадёт в соответствующее окно.

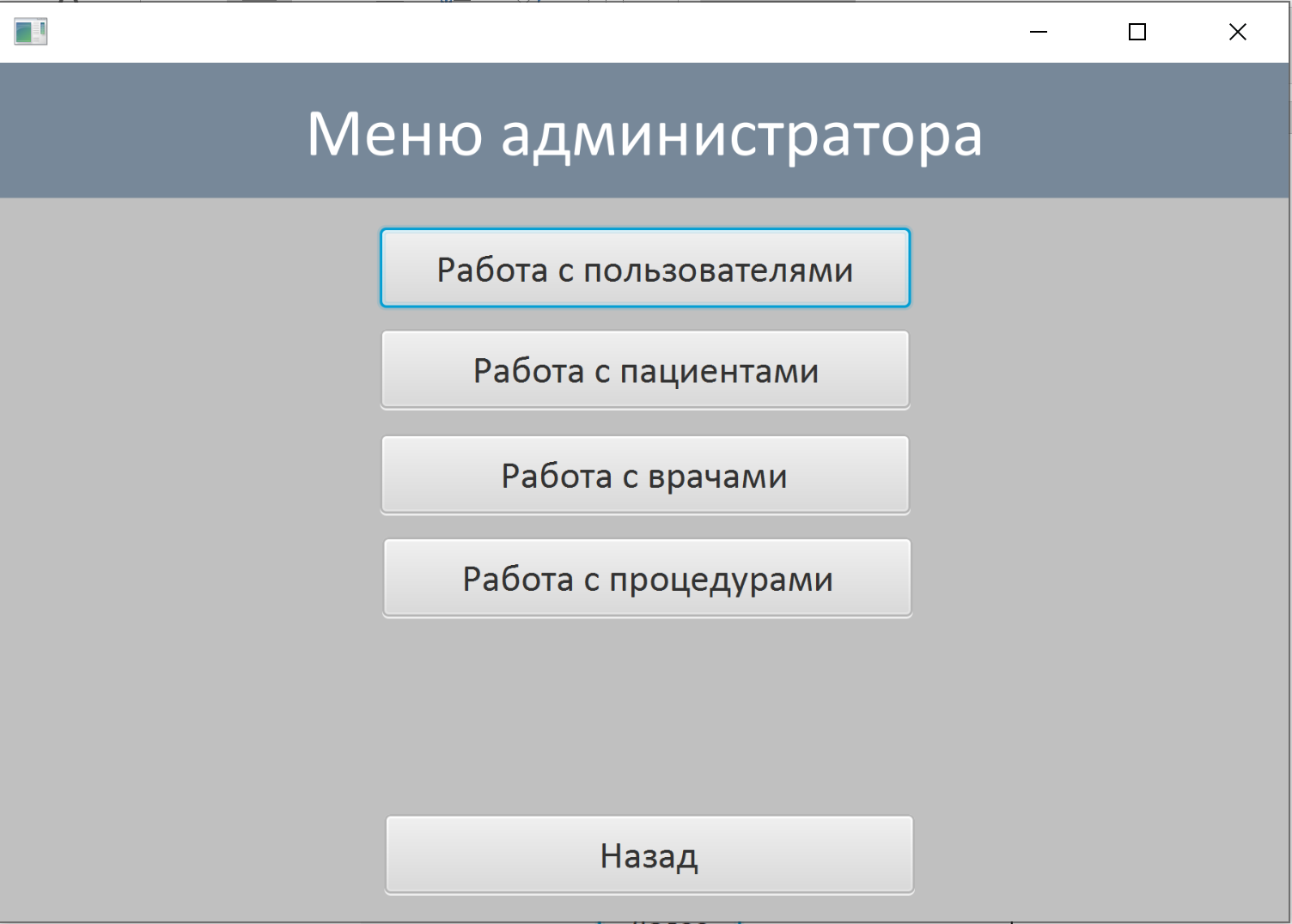


Рисунок 8.6 – Меню администратора

На рисунке 8.7 показан общий вид окна работы с пользователями. Доступны функции удаления, редактирования и поиска записей. Для удаления записи необходимо выделить её в таблице и нажать на кнопку «Удалить».

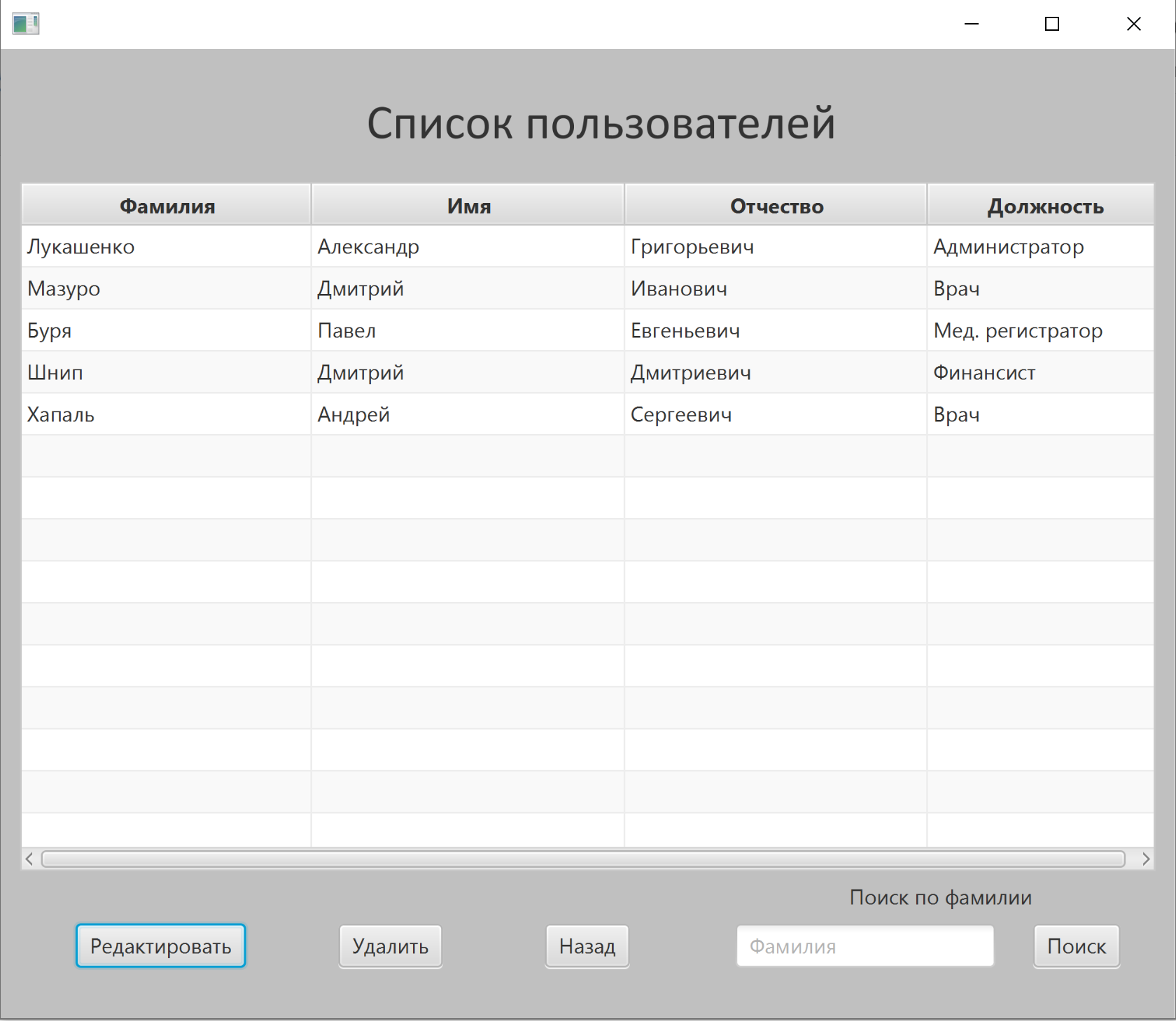


Рисунок 8.7 – Окно работы с пользователями

Окно работы с пациентами представлено на рисунке 8.8. Во всех рабочих окнах пользователю системы доступен стандартный функционал CRUD-операций. В свою очередь, на рисунке 8.9 изображено окно работы со списком врачей, которое откроется по нажатию на кнопку «Работа с врачами» в меню администратора.

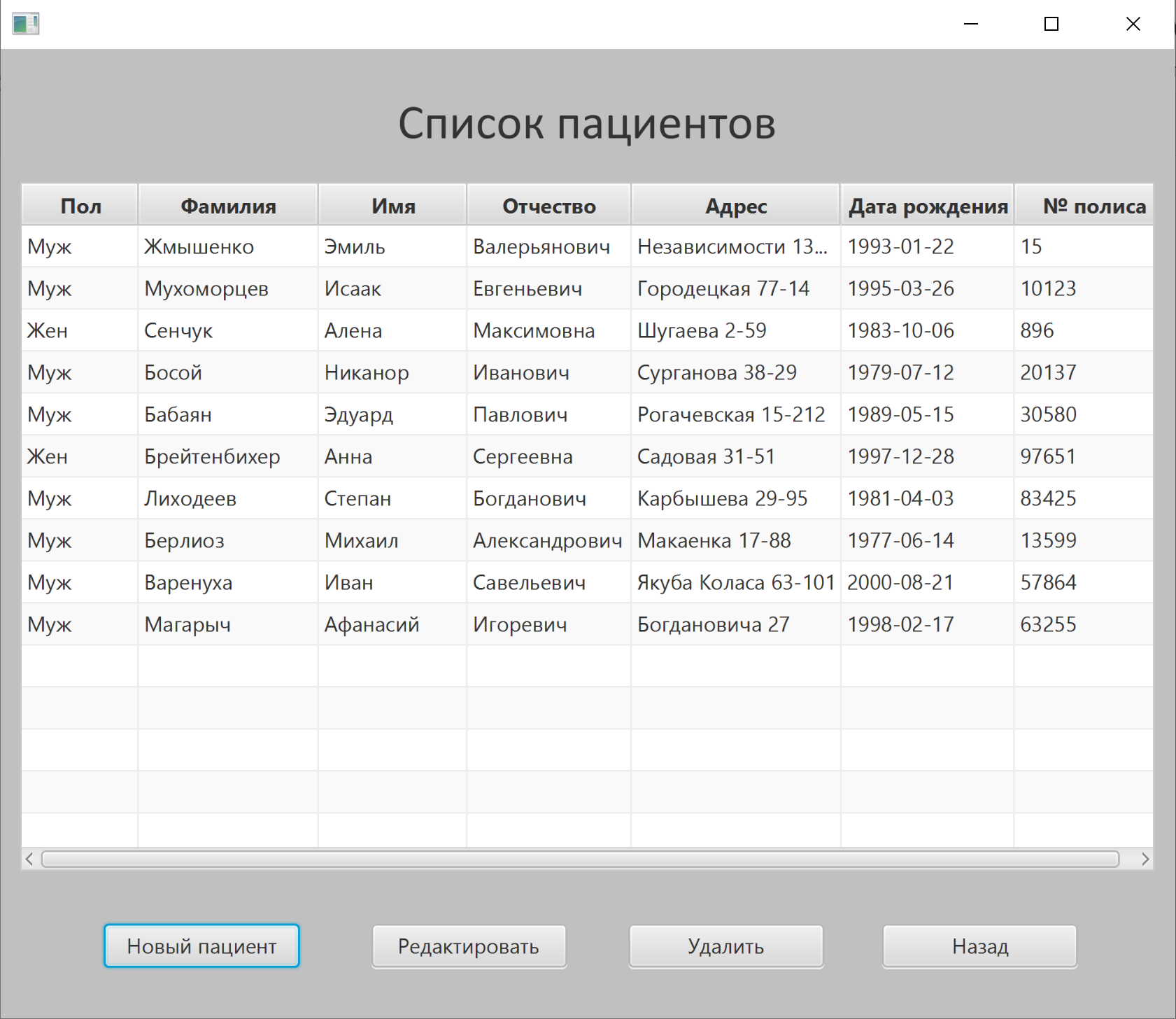


Рисунок 8.8 – Окно работы с пациентами

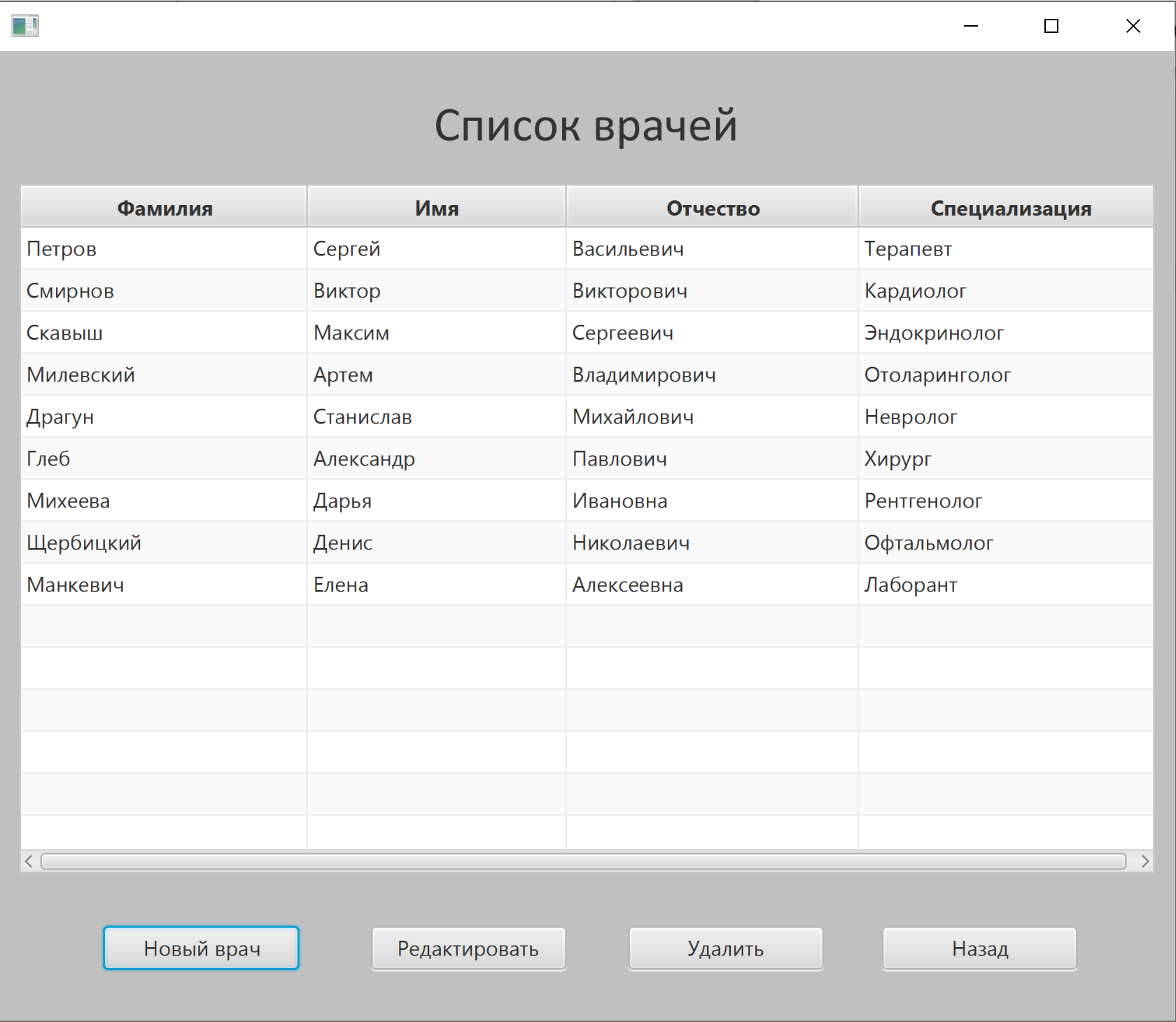


Рисунок 8.9 – Окно работы с врачами

Если же на этапе авторизации был произведён успешный вход в качестве обычного пользователя, то на экран выведется окно, показанное на рисунке 8.10.

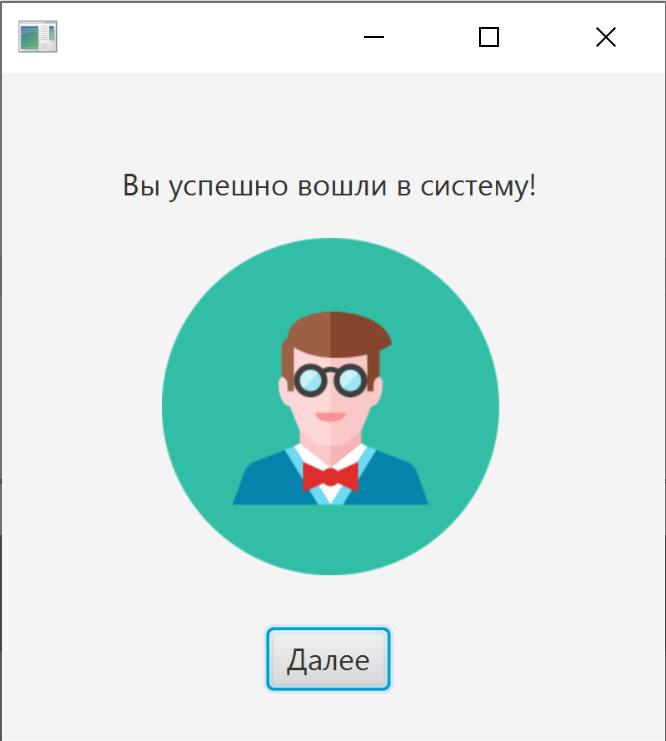


Рисунок 8.10 – Успешная авторизация пользователя

По нажатию на кнопку «Далее» происходит переход в пользовательское меню. Его изображение представлено на рисунке 8.11. Переходом в раздел работы с талонами может воспользоваться медицинский регистратор. На Нажатие кнопки «Приёмы» приведёт к открытию окна работы с приёмами пациентов. Этой функцией могут воспользоваться врачи для заполнения информации о проведённом приёме.

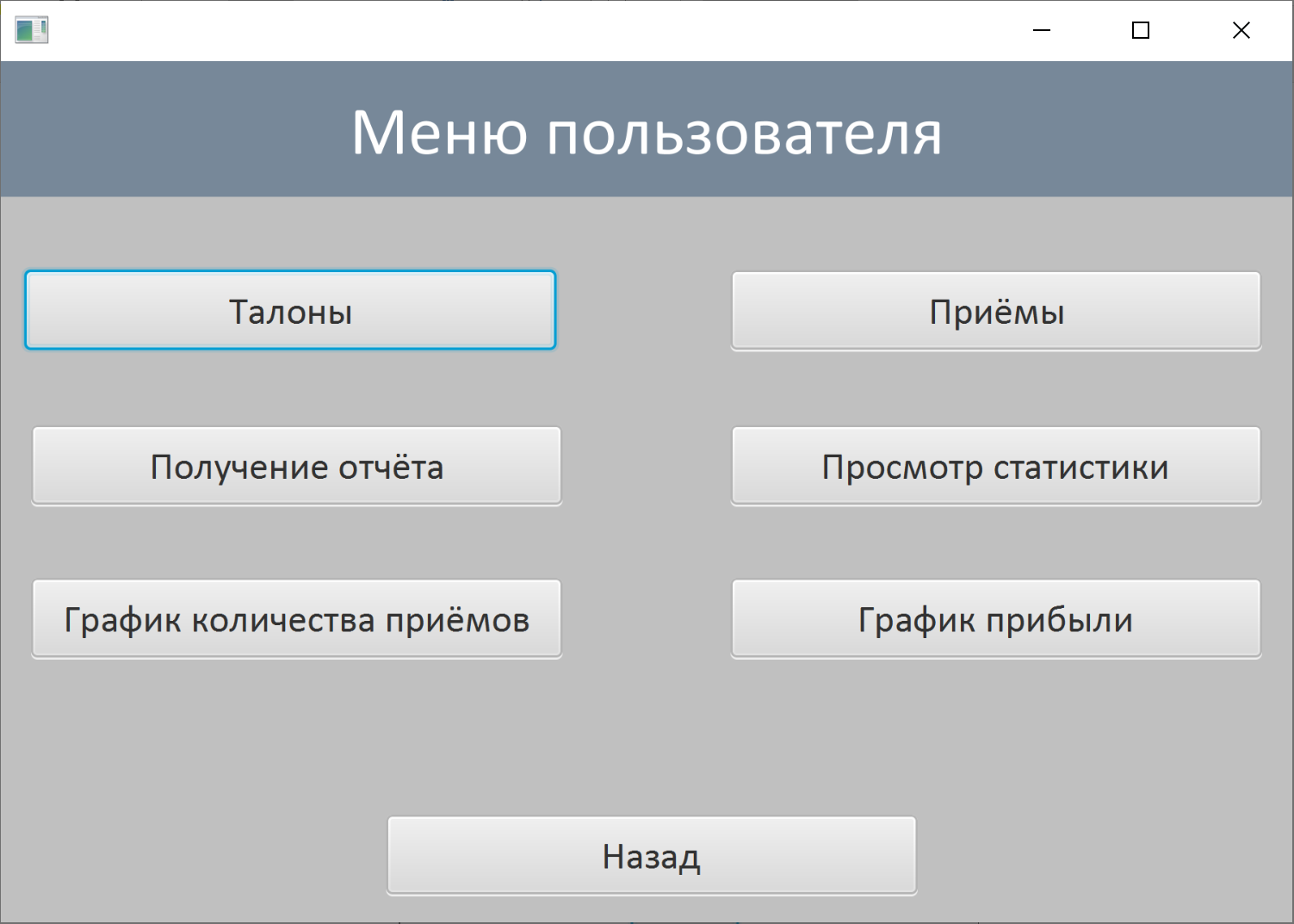


Рисунок 8.11 – Меню пользователя

# **10 РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ РАЗРАБОТАННОЙ СИСТЕМЫ УЧЁТА ОКАЗАНИЯ ПЛАТНЫХ УСЛУГ ПОЛИКЛИНИКОЙ**

В процессе работы данного приложения по разным причинам могут возникать различные ошибки. Для устойчивого функционирования необходимо предусмотреть обработку исключительных ситуаций.

Как уже упоминалось в предыдущем разделе, программное средство позволяет удалять записи, которые более не являются необходимыми. Для этого в рабочем окне нужно выделить в таблице удаляемую запись и нажать на кнопку «Удалить». При успешном удалении на экране появится сообщение, изображённое на рисунке 9.1. Это сообщение будет появляться не только при удалении записи, но также и при успешном добавлении или редактировании записи.

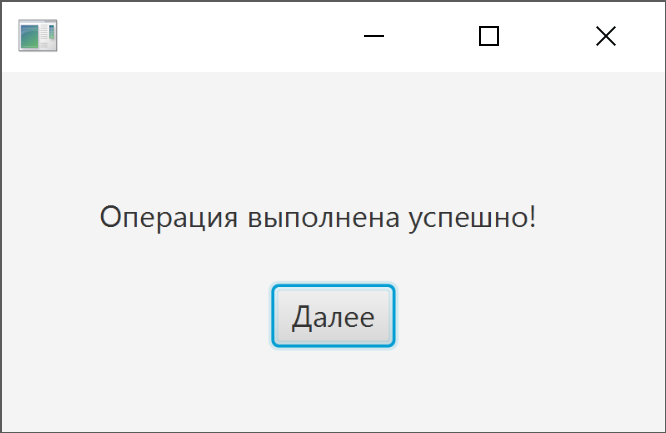


Рисунок 9.1 – Успешное действие

Если перед удалением не будет выбрана запись в таблице, то пользователь увидит сообщение об ошибке (рисунок 9.2).

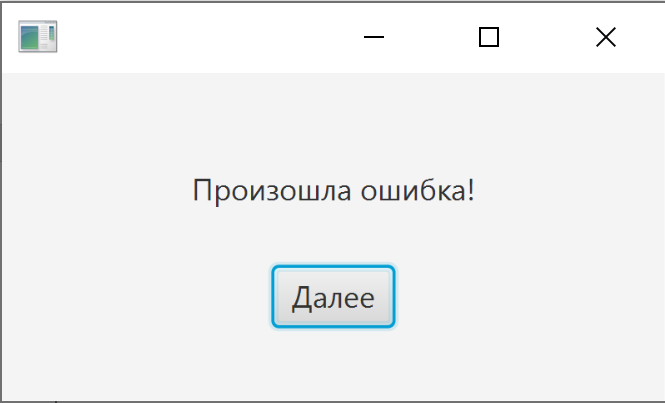


Рисунок 9.2 – Ошибка при совершении операции

Для редактирования записи необходимо, как и при удалении, сперва выбрать в таблице необходимую запись, а затем нажать на кнопку «Редактировать». На рисунке 9.3 представлен внешний вид окна редактирования записи. При открытии этого окна в текстовые поля переносятся соответствующие значения из столбцов выбранной в таблице записи.

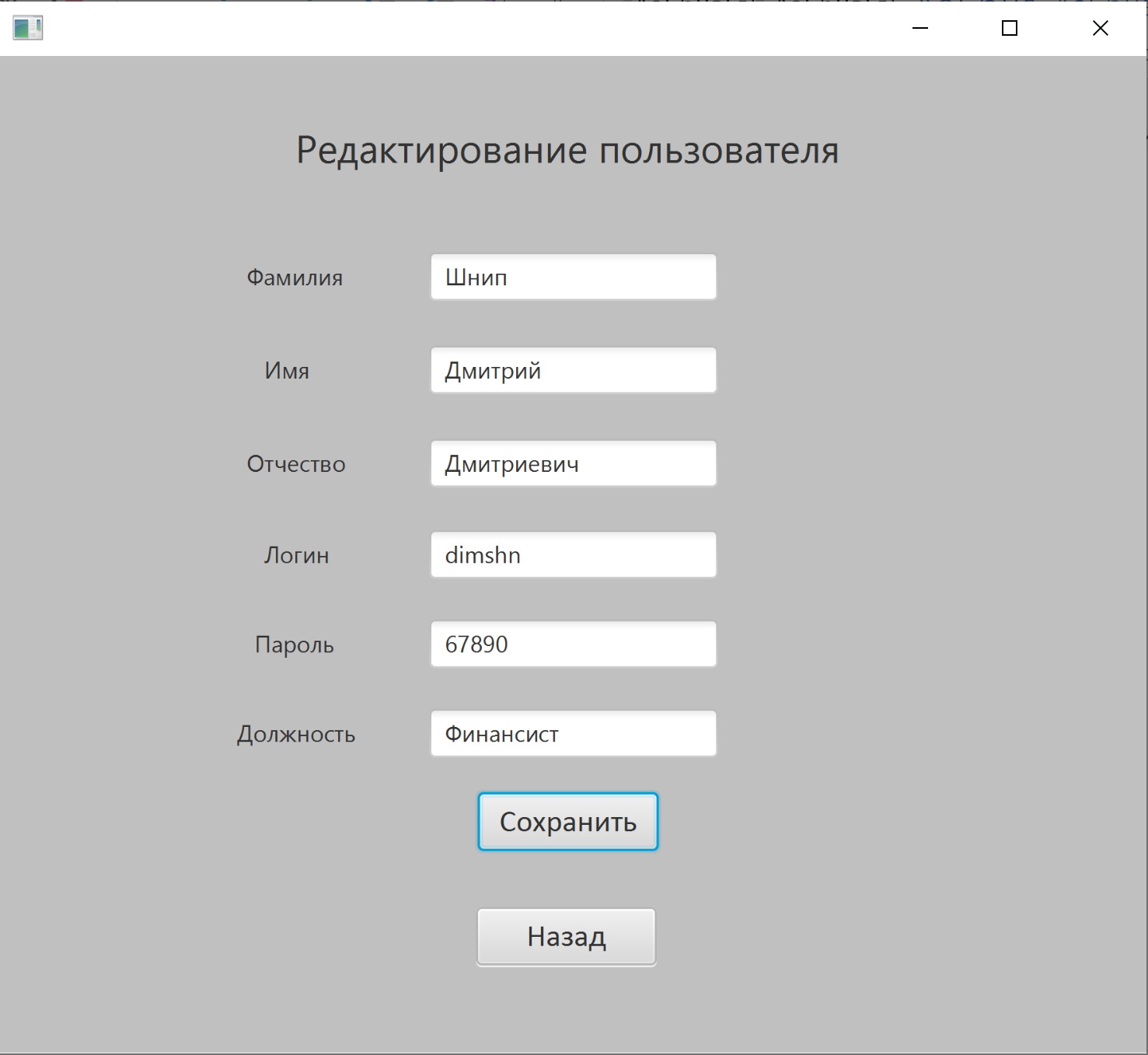


Рисунок 9.3 – Редактирование записи

Для добавления новой записи разработан стандартный внешний вид окна, показанный на рисунке 9.4. В данном случае предусмотрена обработка исключительной ситуации, когда пользователь пытается добавить запись, не заполнив все текстовые поля. В этом случае на экране будет показано соответствующее сообщение об ошибке.

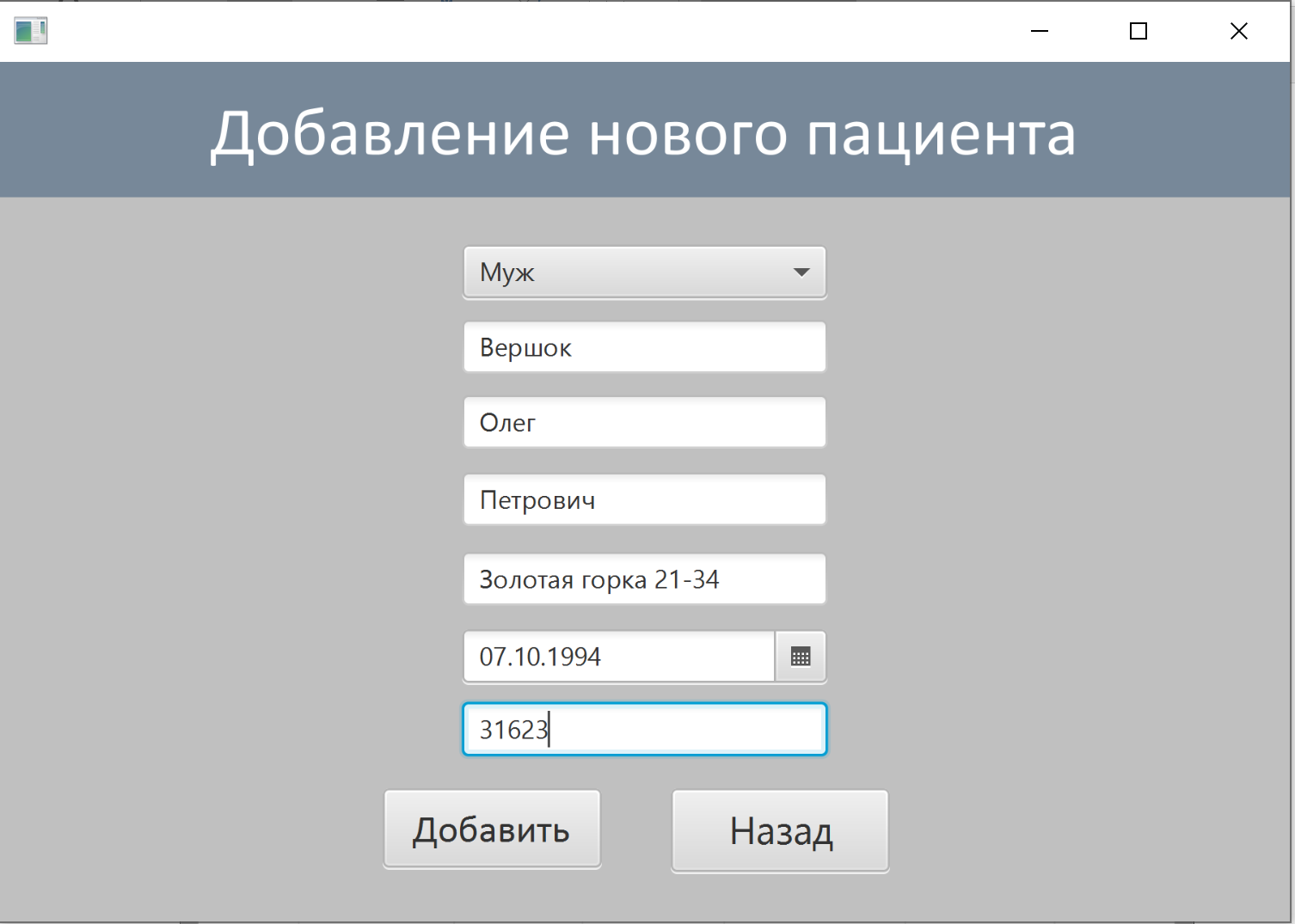


Рисунок 9.4 – Добавление новой записи

В меню пользователя предусмотрено выполнение определённого количества бизнес-функций. Их результаты представляются как в графической, так и в текстовой форме. На рисунке 9.5 представлена диаграмма, показывающая соотношение долей дохода от оказания платных приёмов и платных процедур соответственно в общей сумме дохода.

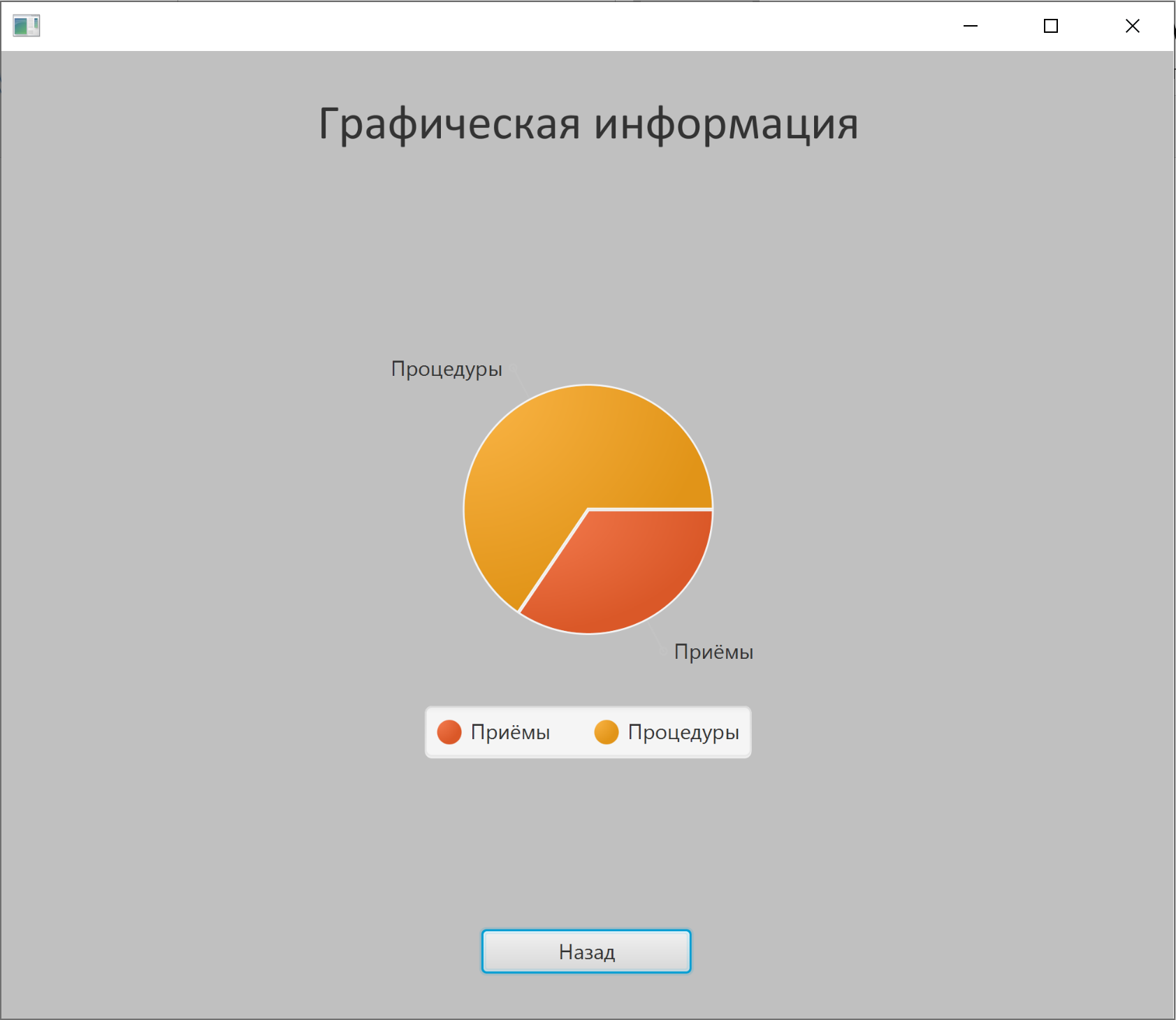


Рисунок 9.5 – Диаграмма соотношения долей в общей сумме дохода

В виде столбчатого графика представляются данные о количестве проведённых приёмов за каждый месяц текущего года. Этот график изображён на рисунке 9.6.

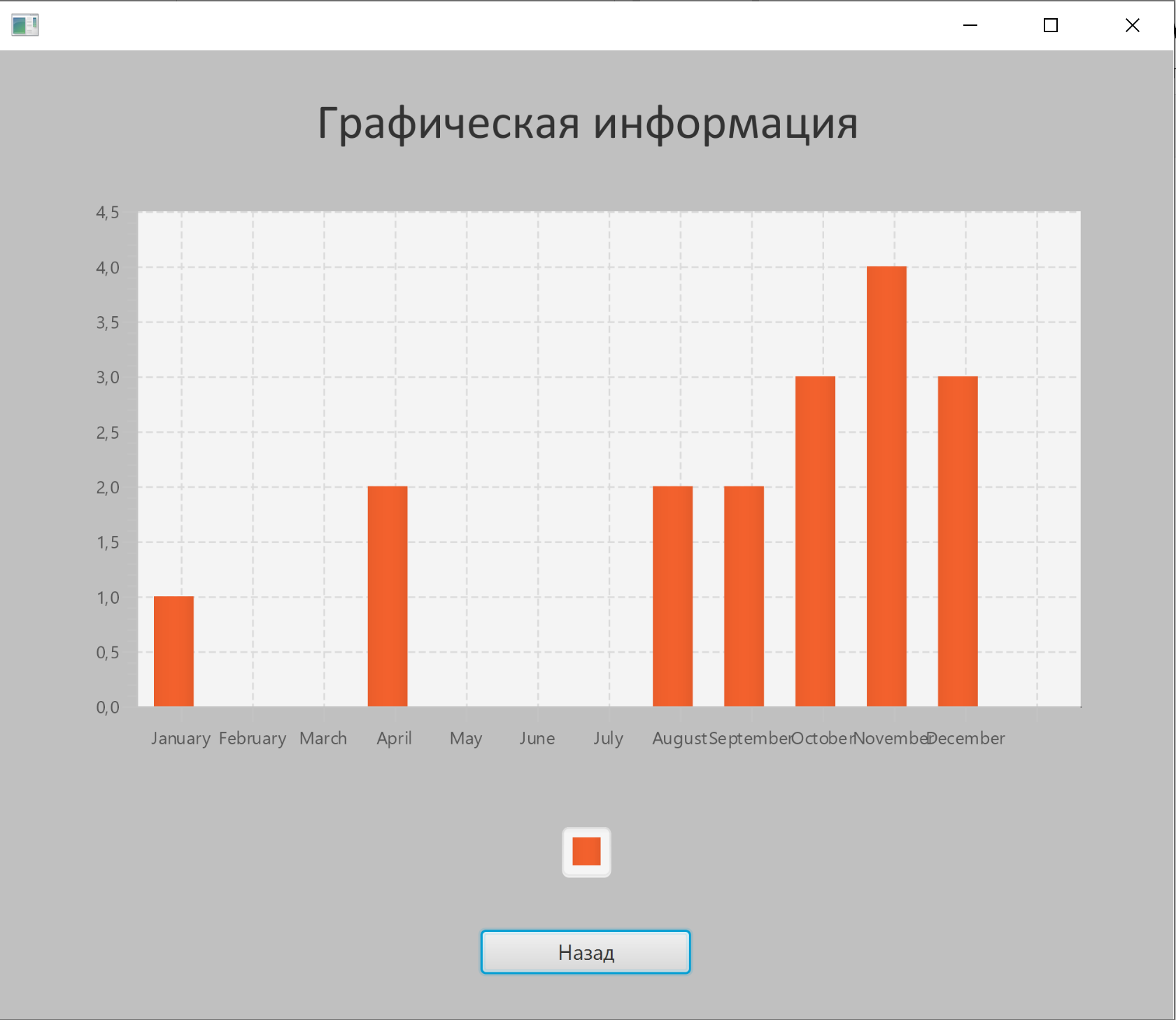


Рисунок 9.6 – График количества приёмов по месяцам

Данные о сумме дохода от оказания платных услуг за каждый месяц представляются в виде линейного графика, который представлен на рисунке 9.7.

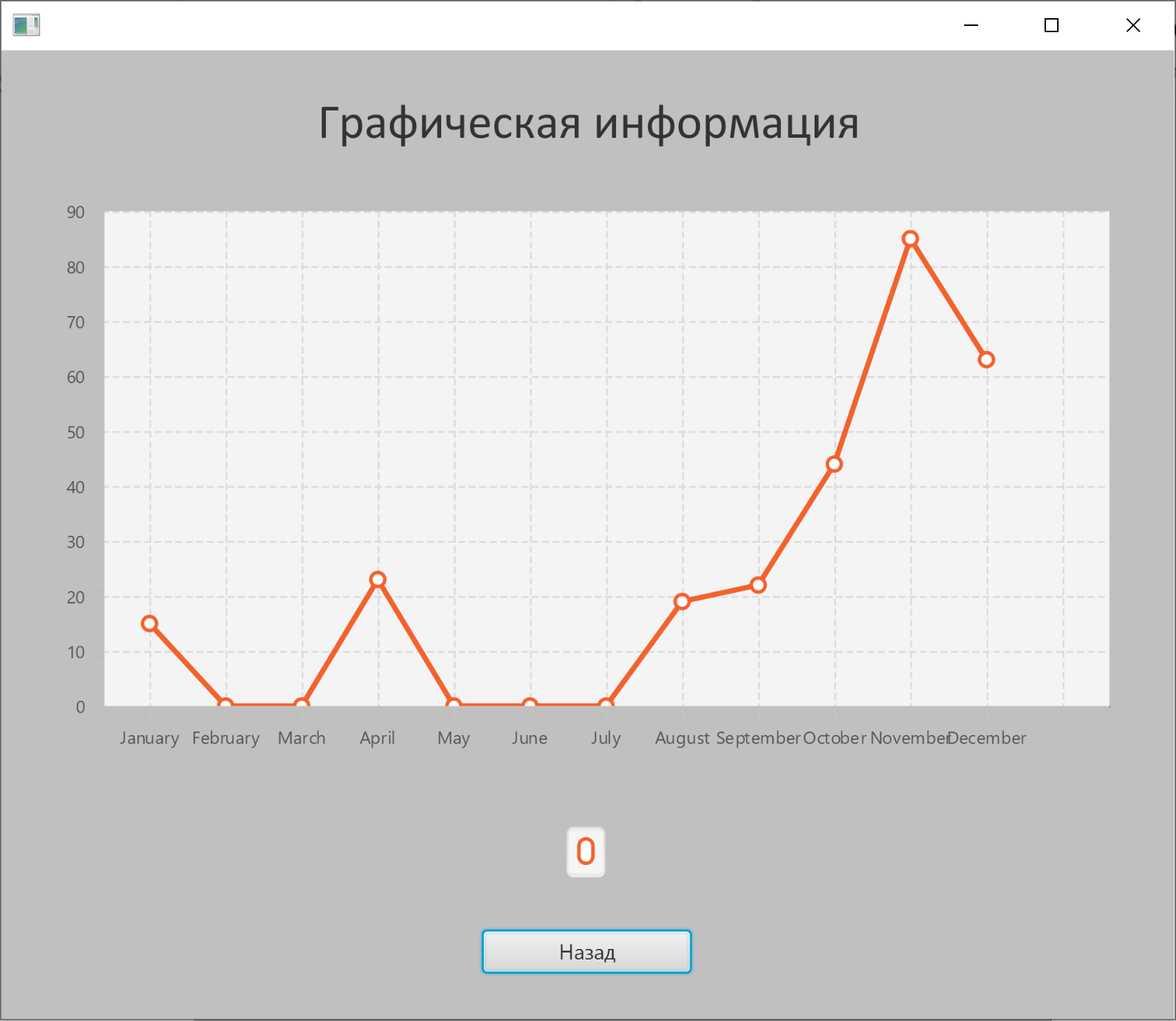


Рисунок 9.7 – График помесячного дохода за текущий год

В текстовом формате представляются некоторые статистические данные, характеризующие доходность поликлиники за текущий год. К этим показателям относятся сумма дохода от проведения платных приёмов, сумма дохода от проведения платных процедур, средняя сумма дохода от проведения платных приёмов, а также средняя сумма дохода от проведения платных процедур. Перечисленные показатели показываются в виде отчёта, изображённого на рисунке 9.8.

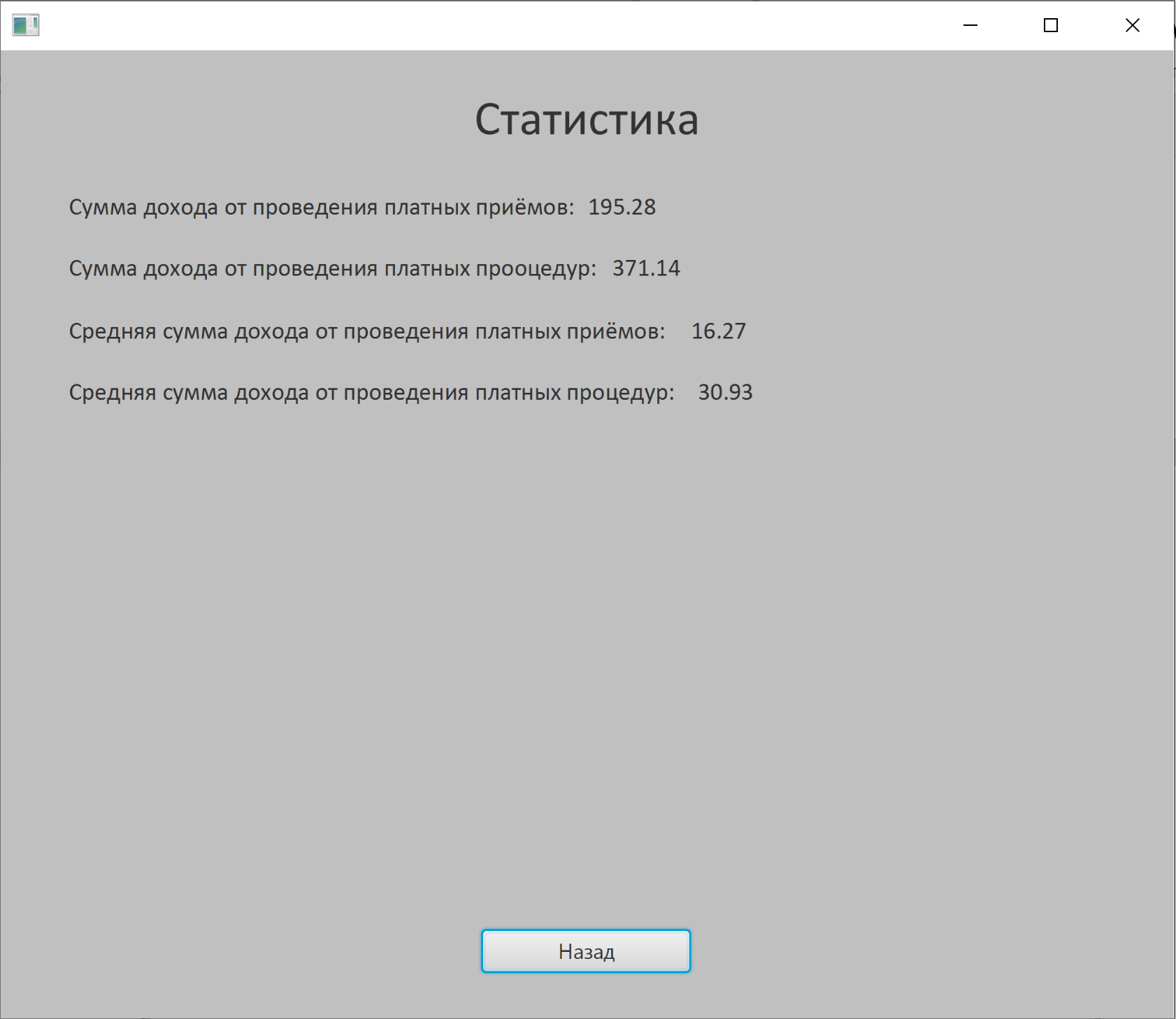


Рисунок 9.8 – Статистические данные о деятельности поликлиники за текущий год

# **ЗАКЛЮЧЕНИЕ**

В результате проделанной работы было создано программное средство, дающее возможность упростить работу сотрудников медицинских учреждений за счёт автоматизации процесса учёта платных услуг, оказываемых пациентам.

В ходе создания программного средства были подробно изучены особенности внедрения информационных технологий в сферу здравоохранения. Полученная информация позволила построить функциональную модель IDEF0, которая наглядно отображает процесс приёма пациента в поликлинике. Были показаны и описаны диаграммы UML, с помощью которых было выполнено проектирование системы. Также была рассмотрена архитектура созданного программного средства. Помимо этого, в ходе выполнения проекта было составлено руководство пользователю, где понятным и доступным языком описывается принцип работы программы. В завершение работы было проведено тестирование разработанной системы, подтвердившее работоспособность созданного программного средства.

Основные функции программного средства реализованы в соответствии с выявленными особенностями предметной области. Был разработан довольно широкий функционал для работы с информацией, которая содержится в базе данных. Стиль интерфейса программы создавался с упором на массовость потребления и использования, который позволит любому пользователю легко и просто использовать данное программное средство без лишних временных затрат.

Бизнес-логика системы даёт возможность систематизировать данные в удобной форме, а также подсчитывать прибыль от оказанных медицинских услуг за определённый период. Полученные результаты могут быть очень эффективны при подведении итогов работы учреждения, а также при определении дальнейшей стратегии развития.

Одним из отличий данного программного средства является надёжная и безопасная база данных, а также структурированная система учёта информации. Благодаря этому вероятность искажения информации минимальна.

В будущем возможно рассмотрение вопроса о расширении функционала программы или же усовершенствования имеющегося. Это обеспечит расширение спектра применения разработанного программного средства.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

[1] Академия профессионального развития [Электронный ресурс]. – Электронные данные. – Режим доступа: https://academy-prof.ru/blog/informacionnye-tehnologii-v-medicine.

[2] Информационное общество в Челябинской области [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.inf74.ru/people/it-tehnologii-na-sluzhbe-cheloveka/kak-informatsionnyie-tehnologii-primenyayutsya-v-meditsine.

[3] ALP Group [Электронный ресурс]. – Электронные данные. – Режим доступа: https://www.alp.ru/itsm/interesting/informatsionnyie-tehnologii-v-zdravoohranenii.

[4] Беловодский, А. А. Здравоохранение в России: проблемы и пути решения / А. А. Беловодский // Современные наукоёмкие технологии. – 2009. – №11. – С. 21 – 27.

[5] Бюро социальной информации [Электронный ресурс]. – Электронные данные. – Режим доступа: http://ru.belbsi.by/rights/council/theses/?tid=2837.

[6] Национальный правовой интернет-портал Республики Беларусь [Электронный ресурс]. – Электронные данные. – Режим доступа: http://pravo.by/document/?guid=3871&p0=v19302435.

[7] Министерство здравоохранения Республики Беларусь [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://minzdrav.gov.by/ru/dlya-belorusskikh-grazhdan/uchrejdenia-zdravoohranenia/polikiliniki.php>.

[8] Блог Алексея Гулынина [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://alekseygulynin.ru/chat-na-java-servernaya-chast>.

[9] YouTube [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.youtube.com/user/PlurrimiTube/featured>.

[10] JavaRush [Электронный ресурс]. – Электронные данные. – Режим доступа: https://javarush.ru.

[11] Блинов, И. Н. Java. Промышленное программирование: практическое пособие / И. Н. Блинов, В. С. Романчик. – Мн.: Универсал-Пресс, 2012.

[12] Буч, Г. UML. Классика CS / Г. Буч, Дж. Рамбо, А. Якобсон; пер. с англ. – 2-е изд. – СПб.: Питер, 2006.

# **ПРИЛОЖЕНИЕ А**

**(обязательное)**

**Диаграммы UML**

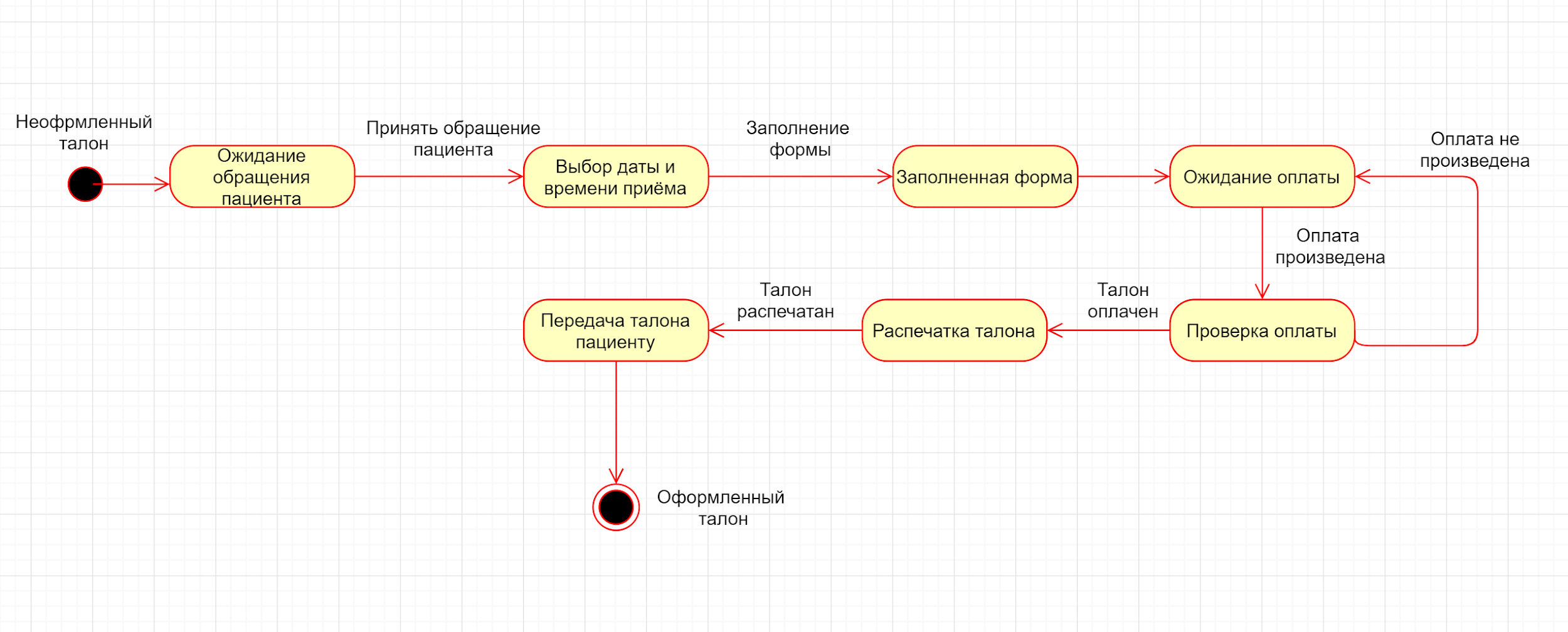


Рисунок А.1 – Диаграмма состояний

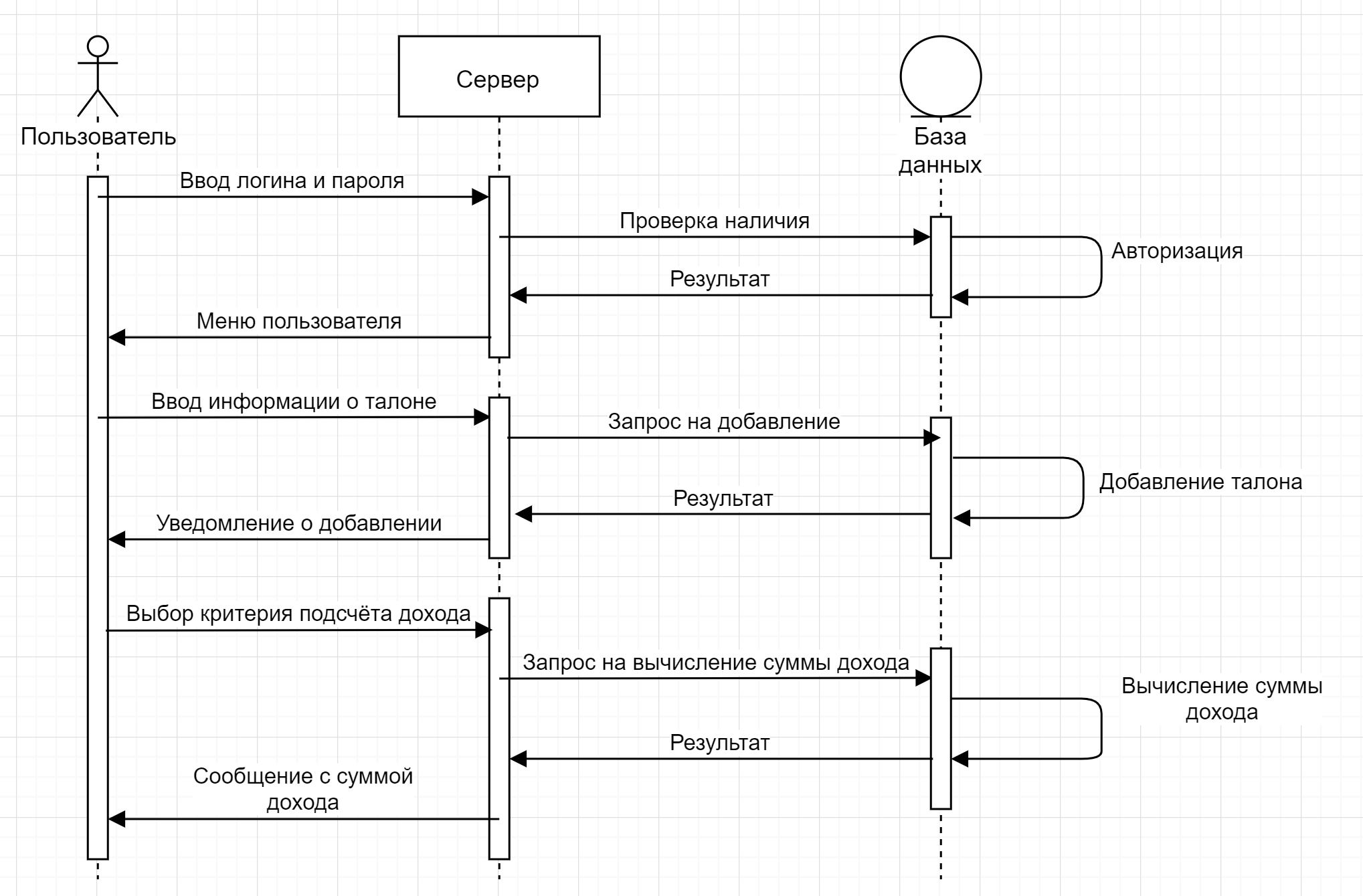


Рисунок А.2 – Диаграмма последовательностей

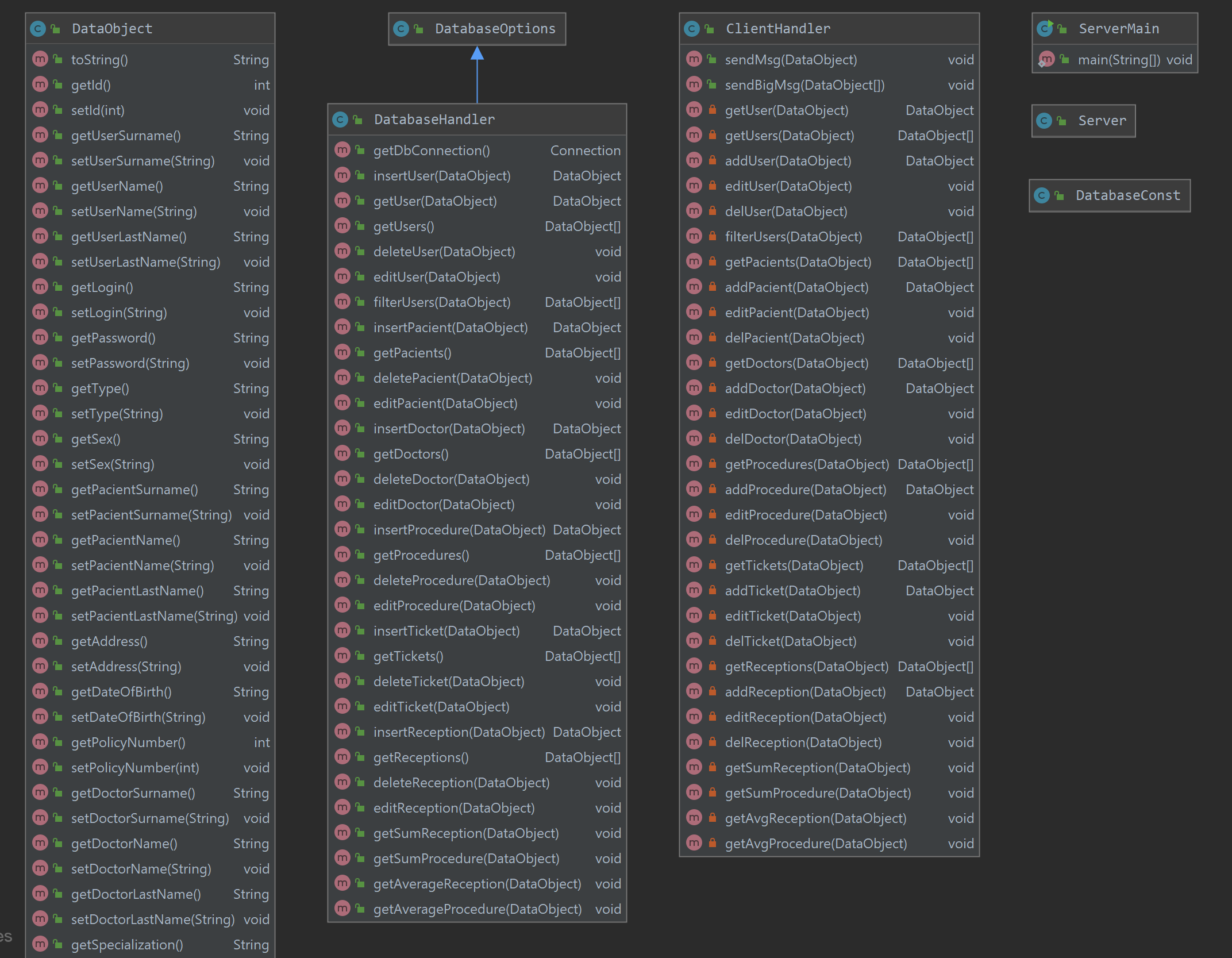


Рисунок А.3 – Диаграмма классов серверной части

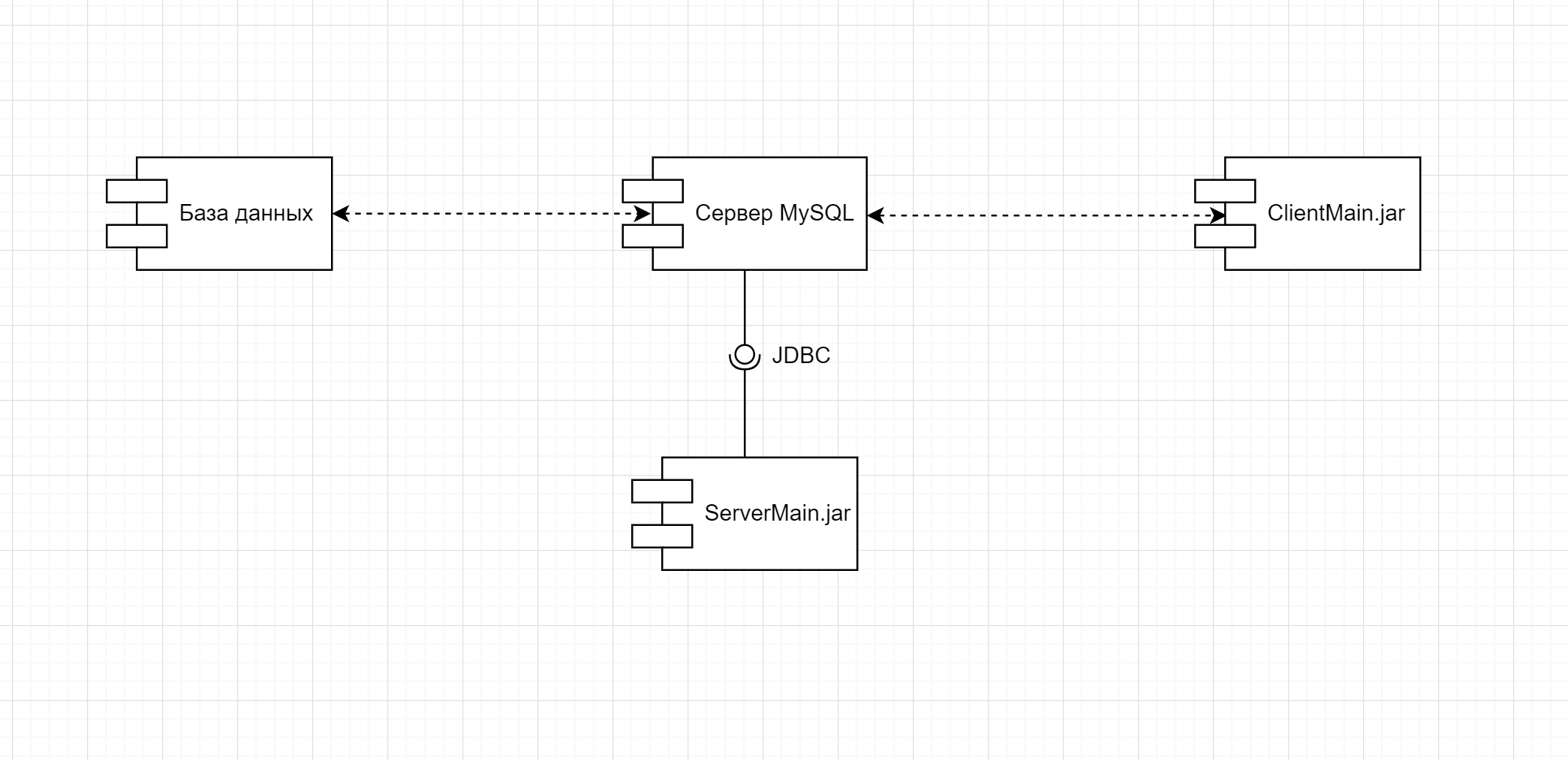


Рисунок А.4 – Диаграмма компонентов

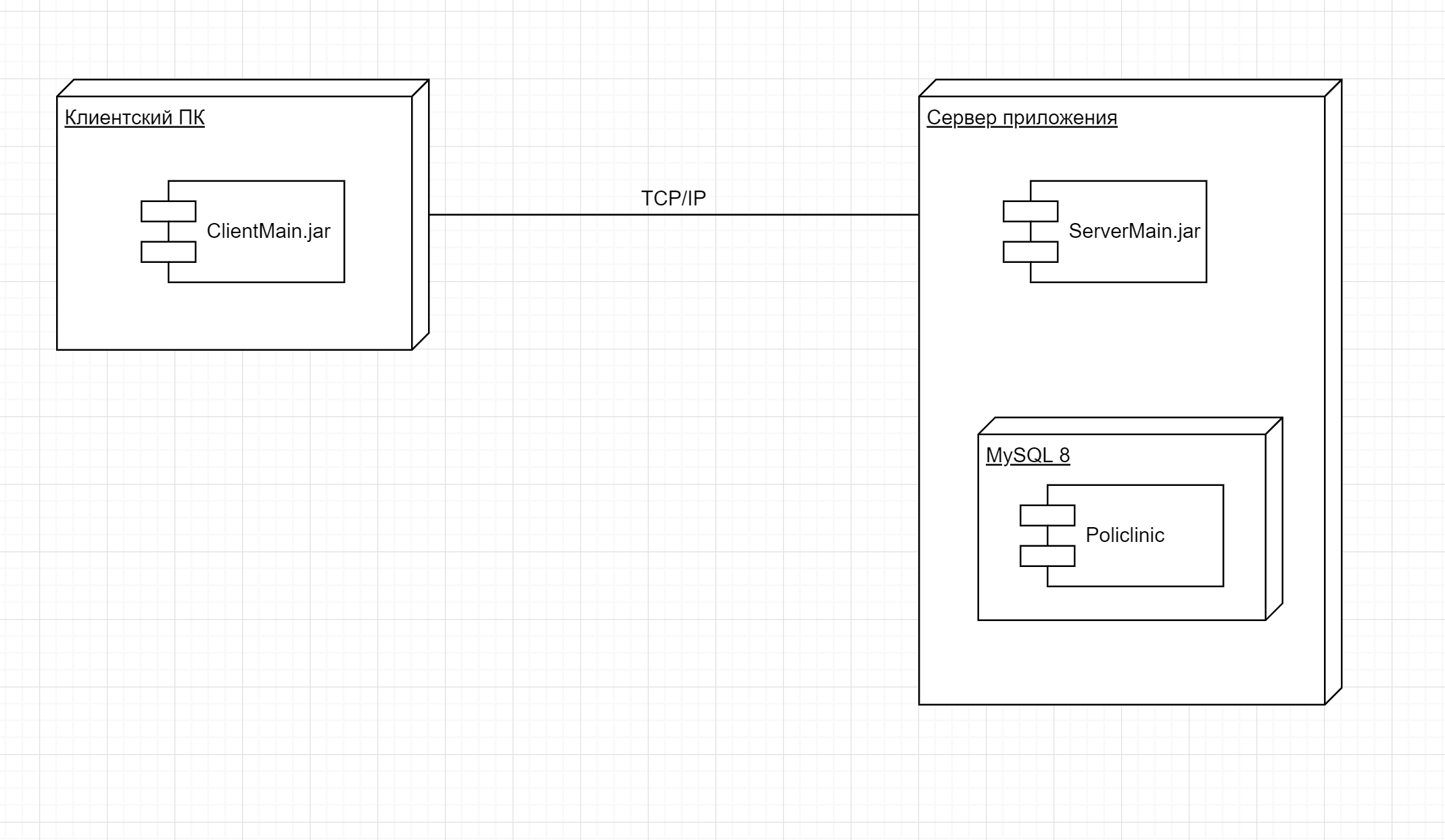


Рисунок А.5 – Диаграмма развёртывания

# **ПРИЛОЖЕНИЕ Б**

**(обязательное)**

**Схемы алгоритмов работы программы**



Рисунок Б.1 – Схема алгоритма клиент-серверного взаимодействия



Рисунок Б.2 – Схема алгоритма поиска пользователя по фамилии