**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ3

1. **ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ**6
   1. Введение в предметную область6
   2. Анализ готовых решений10
2. **ПРОЕКТИРОВАНИЕ СИСТЕМЫ**12
   1. Общее назначение системы12
   2. Требования к функциям, выполняемым системой13
   3. План тестирования17
3. **РЕАЛИЗАЦИЯ ПРОЕКТА СИСТЕМЫ**19
   1. Описание среды разработки19
      1. Выбор и описание программных инструментов19
      2. Обоснование выбора инструментария по разработке19
   2. Разработка программного модуля20
      1. Реализация пользовательского интерфейса программы20
      2. Описание кодом функциональных узлов модуля21
      3. Результат работы и тестирования22

**ЗАКЛЮЧЕНИЕ**24

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**25

**ПРИЛОЖЕНИЕ 1**27

**ПРИЛОЖЕНИЕ 2**30

**ПРИЛОЖЕНИЕ 3**40

**ВВЕДЕНИЕ**

Данный курсовой проект посвящен разработке программного модуля "Система учета пациентов в стоматологической клинике". В современных условиях эффективное управление медицинским учреждением невозможно без использования специализированного программного обеспечения. Особенно это актуально для стоматологических клиник, где важно вести точный учет пациентов, их истории болезни, назначенного лечения и оказанных услуг.

Целью курсового проекта является создание программного модуля «Система учёта пациентов в стоматологической клинике», который позволит автоматизировать процесс учета пациентов в стоматологической клинике, повысить эффективность работы персонала и улучшить качество обслуживания клиентов.

Для достижения поставленной цели необходимо решить следующие задачи:

1) Проанализировать предметную область, изучить специфику работы стоматологических клиник и требования к системам учета пациентов.

2) Проанализировать готовые решения, представленные на рынке, выявить их достоинства и недостатки.

3) Подготовить техническое задание на разработку программного модуля, определив функциональные и нефункциональные требования к системе.

4) Подготовить план тестирования разрабатываемого программного модуля для обеспечения его качества и надежности.

5) Обосновать выбор инструментов и средств разработки, учитывая требования к производительности, безопасности и удобству использования системы.

6) Описать реализацию технических задач, включая проектирование архитектуры системы, разработку базы данных и пользовательского интерфейса.

7) Выполнить тестирование разработанного программного модуля в соответствии с подготовленным планом, выявить и устранить возможные ошибки.

Объект исследования является бизнес-процесс учета пациентов в стоматологической клинике, включающий все этапы взаимодействия с пациентами — от записи на прием до ведения истории болезни и оформления оказанных услуг.

Предмет исследования: Цикл управления данными о пациентах в стоматологической клинике:

Создание:

* Процессы регистрации и ввода информации о пациентах.
* Формирование истории болезни и документации лечения.

Хранение:

* Организация базы данных для надежного и безопасного хранения информации о пациентах.
* Обеспечение конфиденциальности и защиты данных.

Обработка:

* Автоматизированные процессы управления информацией о пациентах (обновление записей, назначение лечения, управление услугами).
* Интеграция с другими системами для комплексного управления клиникой.

Анализ:

* Разработка инструментов для анализа данных о пациентах (статистика посещений, эффективность лечения, демографические данные).
* Получение оперативной отчетности для улучшения качества обслуживания.

Актуальность данного исследования обусловлена не только необходимостью, но и настоятельной потребностью современных стоматологических клиник внедрять информационные технологии для повышения своей конкурентоспособности на рынке медицинских услуг. В условиях постоянно растущих требований пациентов к качеству и доступности медицинской помощи, а также усиления конкуренции, автоматизация учета пациентов становится ключевым фактором успеха. Это позволяет не только оптимизировать внутренние процессы, но и улучшить взаимодействие с клиентами, предоставить более персонализированный подход, повысить уровень их удовлетворенности и, соответственно, увеличить рентабельность и прибыльность учреждения.

Таким образом, разработка программного модуля "Система учета пациентов в стоматологической клинике" является важным и своевременным шагом к модернизации работы медицинского учреждения. Внедрение данной системы позволит автоматизировать ключевые бизнес-процессы, минимизировать риски ошибок, связанных с ручным вводом данных, повысить точность и оперативность учета информации о пациентах, а также улучшить качество предоставляемых услуг. Ожидается, что система приведет к увеличению эффективности работы персонала, освободив их от рутинных задач, снижению операционных издержек и повышению удовлетворенности клиентов. В перспективе разработанный модуль станет основой для дальнейших улучшений и расширения функционала, способствуя устойчивому развитию стоматологической клиники в условиях растущей конкуренции на рынке медицинских услуг и обеспечивая ей долгосрочные конкурентные преимущества.

**1. Описание предметной области**

1. **Введение в предметную область**

Предметная область "Система учета в стоматологической клинике" охватывает комплекс взаимосвязанных процессов, направленных на сбор, обработку, хранение и анализ информации, необходимой для эффективного функционирования клиники и предоставления качественных медицинских услуг пациентам.

Традиционно учет в стоматологических клиниках велся с помощью бумажной документации, что создавало ряд проблем: громоздкость, дублирование информации, сложности с поиском и анализом данных, риск ошибок и потери данных.

Внедрение информационных технологий (ИТ) трансформирует учет в стоматологии, предлагая автоматизированные решения для оптимизации бизнес-процессов и повышения качества обслуживания.

Современные теоретические подходы к системам учета в стоматологических клиниках предполагают комплексное использование методов информационных технологий в управленческой и маркетинговой деятельности. Такие системы обеспечивают логическую и физическую интеграцию данных, создавая единую информационную среду. Это требует анализа и проектирования не только самого программного обеспечения (ПО), но и организационных процессов компании.

Теоретические основы предметной области включают медицинскую документацию (стандарты ведения электронных медицинских карт, требования к хранению и защите медицинской информации), управление клиникой (планирование ресурсов, финансовый учет, маркетинг и взаимодействие с пациентами) и информационные технологии (базы данных, программная инженерия, безопасность информационных систем).

Современные требования к ИС в этой области включают функциональность (ведение электронных медицинских карт, учет материалов, финансовый учет, запись на прием, аналитика и отчетность), удобство использования (интуитивно понятный интерфейс, простота в освоении, мобильность), безопасность (защита конфиденциальных данных пациентов, соответствие законодательству) и интеграцию (возможность обмена данными с другими системами, используемыми в клинике).

В современном обществе, которое активно использует информационные технологии и данные, законодательные аспекты играют ключевую роль в регулировании различных сфер общественной жизни. В российском законодательстве существует ряд нормативных актов и стандартов, которые обеспечивают правовые и организационные рамки для работы с информацией, ее защитой и использованием в различных областях, включая образование, информационные технологии и управление документацией.

1. Федеральный закон "О персональных данных" (от 27.07.2006 N 152-ФЗ) [1] регулирует порядок обработки персональных данных. Он определяет права субъектов персональных данных и обязанности операторов, что является критически важным в условиях цифровизации и массового сбора данных.
2. Федеральный закон "Об информации, информационных технологиях и о защите информации" (от 27.07.2006 N 149-ФЗ.) [2] закладывает правовые основы оборота информации, использования информационных технологий и обеспечения ее защиты. Это способствует созданию безопасной информационной среды и поддерживает развитие цифровой экономики.
3. Федеральный закон "О безопасности критической информационной инфраструктуры Российской Федерации" ( от 26.07.2017 N 187-ФЗ) [4]. Фокусируется на вопросах обеспечения безопасности критически важной информационной инфраструктуры, включая защиту от кибератак. Это особенно актуально в условиях возрастающих угроз в киберпространстве.

Кроме законов, существенную роль играют и государственные стандарты (ГОСТы), которые регулируют различные аспекты документооборота и информационной деятельности:

1. ГОСТ 19.701–90 [5] Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения. .
2. ГОСТ Р ИСО/МЭК 12207–2010 [3] Системная и программная инженерия. Процессы жизненного цикла программных средств.

Эти нормативные акты и стандарты формируют основу для эффективного управления информацией и обеспечения ее безопасности, а также для поддержания высокого уровня организации в образовательной и информационной сферах.

В мире разработки программного обеспечения (ПО) существует множество различных подходов и методов, каждый из которых имеет свои преимущества и недостатки. Однако, все они объединены одной общей целью - создать эффективную и качественную систему, которая удовлетворяет потребностям пользователей.

Методология разработки ПО — это комплексный подход к созданию программного обеспечения, который включает в себя набор принципов, методов и инструментов, направленных на достижение этой цели. Она определяет, каким образом будет проходить процесс разработки, какие этапы будут включены, и как будут взаимодействовать участники проекта.

Существует множество различных методологий разработки ПО, каждая из которых имеет свои особенности и преимущества. Некоторые из наиболее популярных методологий включают:

* Водопадная модель - традиционная модель разработки, в которой проект разбивается на последовательные фазы, каждая из которых должна быть завершена до начала следующей.
* V-Model - модель, которая сочетает в себе элементы водопадной модели и итеративной разработки.
* RAD (Быстрая разработка приложений) - модель, которая предполагает быструю разработку системы с использованием готовых компонентов и инструментов.
* Agile (Гибкая методология) - модель, которая предполагает разработку системы в несколько итераций, каждая из которых добавляет новые функции к системе, с постоянным взаимодействием с заинтересованными сторонами и итеративным улучшением.

Для проекта по разработке информационной системы для стоматологической клиники, Agile (гибкая методология) будет наиболее подходящей. Эта методология позволяет адаптироваться к изменениям в требованиях, взаимодействовать с заинтересованными сторонами, итеративно улучшать систему, управлять рисками, фокусироваться на пользователе и быстро доставлять ценность.

Agile предполагает разработку системы в несколько итераций, каждая из которых добавляет новые функции к системе. Каждая итерация (спринт) длится несколько недель или месяцев и включает в себя следующие этапы:

* Планирование - определение целей и задач для спринта.
* Разработка - разработка функций и компонентов системы.
* Тестирование - тестирование разработанных функций и компонентов.
* Демонстрация - демонстрация результатов спринта заинтересованным сторонам.
* Ретроспектива - анализ результатов спринта и определение областей для улучшения.

Agile требует активного и постоянного участия клиента, что может быть сложно обеспечить в некоторых случаях. Однако, это позволяет быстро получать обратную связь и вносить необходимые изменения в систему.

В целом Agile — это гибкая и эффективная методология разработки ПО, которая позволяет создавать высококачественные системы, удовлетворяющие потребностям пользователей.

**1.2 Анализ готовых решений по системе управления пациентами стоматологической клиники**

Выбор системы управления пациентами для стоматологической клиники – задача, требующая тщательного анализа. Рынок предлагает широкий спектр решений, подходящих для различных типов медицинских учреждений, от небольших частных клиник до крупных сетей и специализированных центров.

Рассмотрим, например, частные стоматологические клиники. Для них важна эффективность управления и высокое качество обслуживания пациентов. Здесь оптимальным решением может стать система с интуитивно понятным интерфейсом и доступной ценой, такая как "1C: Медицина. Стоматологическая клиника". Её преимущество – русскоязычная поддержка и лёгкость интеграции с другими продуктами 1С, что особенно ценно для уже использующих эту платформу. Однако, функциональность может быть ограниченнее по сравнению с западными аналогами, требуя регулярных обновлений.

В отличие от частных клиник, сетевые стоматологические клиники нуждаются в централизованном управлении данными пациентов и координации работы между филиалами. Для них важна масштабируемость и надёжность системы. Более сложные системы, такие как Dentrix, хотя и стоят дороже и требуют значительных вложений в обучение персонала, предлагают широкий функционал и высокую степень интеграции с различными медицинскими устройствами, что оправдывает затраты в долгосрочной перспективе. Dentrix охватывает все аспекты работы клиники, от управления расписанием до финансового учёта и интеграции с рентгенологическими системами.

Специализированные центры, такие как ортодонтические клиники, могут извлечь выгоду из использования систем, ориентированных на их специфику. OrthoTrac, например, предлагает специализированные функции для отслеживания прогресса лечения и интеграции с 3D-сканерами. Однако, его ограниченная применимость для общих стоматологических клиник делает его менее универсальным решением.

Облачные системы, подобные Curve Dental, обеспечивают доступ к данным из любого места, что удобно для мобильных сотрудников и позволяет оперативно реагировать на запросы пациентов. Но подобное решение не лишено недостатков: зависимость от стабильного интернет-соединения и потенциальные вопросы безопасности данных требуют внимательного рассмотрения.

В итоге выбор системы управления пациентами зависит от конкретных потребностей клиники. Необходимо учитывать размер клиники, её специализацию, бюджет, а также желаемый уровень функциональности и интеграции с существующими системами. Важно взвесить преимущества и недостатки каждого решения, чтобы выбрать оптимальную систему, которая обеспечит эффективное управление и повысит качество обслуживания пациентов.

Таким образом, система учета в стоматологической клинике автоматизирует ключевые процессы, такие как регистрация пациентов, ведение медицинской документации, управление расписанием приема и формирование отчетности. Она интегрируется с медицинскими устройствами, что позволяет врачам получать точные данные в режиме реального времени. Это минимизирует ошибки, связанные с человеческим фактором, повышает эффективность работы клиники и обеспечивает соблюдение нормативных требований, включая защиту персональных данных пациентов. Внедрение такой системы улучшает качество обслуживания, повышает удовлетворенность пациентов и упрощает работу административного и медицинского персонала, что в конечном итоге способствует укреплению репутации клиники и ее успешному развитию.

**2. Проектирование системы**

**2.1. Общее назначение системы**

Система учета пациентов в стоматологической клинике предназначена для автоматизации процессов сбора, хранения и обработки данных о пациентах и предоставляемых им медицинских услугах. Основная цель системы — улучшить управление медицинскими процессами, сократить трудозатраты со стороны медицинского и административного персонала, а также обеспечить удобный доступ к информации для всех участников рабочего процесса.

Предполагаемый результат:

* Полная автоматизация учета пациентов, их истории болезни и назначенных процедур.
* Сокращение времени на ручную обработку данных и оформление документов.
* Оперативное формирование отчетов для различных уровней управления (врачи, администраторы, руководство клиники).
* Повышение точности и достоверности данных за счет исключения человеческого фактора.
* Интеграция с другими системами клиники для создания единой информационной среды.

Эффект от внедрения:

* Повышение прозрачности медицинского процесса: пациенты и персонал смогут в реальном времени отслеживать информацию о записях на прием, результатах лечения и оплатах.
* Ускорение процесса принятия решений за счет быстрого доступа к актуальной информации.
* Снижение нагрузки на сотрудников клиники, так как рутинные операции учета и анализа выполняются автоматически.
* Улучшение качества обслуживания пациентов за счет своевременного выявления и решения проблем.
* Соответствие законодательным требованиям по защите персональных данных, что минимизирует риски нарушения конфиденциальности.

**2.2 Требования к функциям, выполняемым системой**

Система учета пациентов в стоматологической клинике должна выполнять ряд ключевых функций, обеспечивающих автоматизацию процессов сбора, обработки и анализа данных о пациентах и предоставляемых медицинских услугах. Исходя из анализа предметной области, можно выделить следующие функциональные задачи программного модуля.

Функциональные задачи программного модуля:

1. Регистрация и управление данными пациентов:
   * Ввод и хранение персональных данных пациентов.
   * Обновление и корректировка информации о пациентах.
   * Поиск и фильтрация пациентов по различным критериям.
2. Ведение медицинской документации:
   * Создание и хранение истории болезни каждого пациента.
   * Запись диагнозов, результатов обследований и назначенного лечения.
3. Управление расписанием и запись на прием:
   * Запись пациентов на прием к врачам.
   * Управление расписанием работы врачей.
4. Учет услуг и расчет стоимости:
   * Ведение списка медицинских услуг и их стоимости.
   * Управление оплатами и учетом задолженностей.
5. Управление персоналом и доступом:
   * Регистрация сотрудников клиники и распределение ролей.
   * Разграничение прав доступа к функциям системы.
   * Обеспечение безопасности и конфиденциальности данных.
6. Формирование отчетности и аналитика:
   * Анализ эффективности работы клиники и специалистов.
   * Экспорт данных для дальнейшей обработки.
7. Управление лечением и назначениями:
   * Фиксация проводимого лечения и назначений.
   * Управление планами лечения пациентов.

Основные объекты предметной области

Основными объектами предметной области для системы учета пациентов в стоматологической клинике являются объекты, представленные в таблице 1.

Таблица 1 - Объекты предметной области

| № п/п | Объект | Действия |
| --- | --- | --- |
| 1 | Пациент | Регистрация, просмотр истории лечения |
| 2 | Врач | Проведение консультаций, назначение лечения, ведение истории |
| 3 | Администратор | Управление пользователями, настройка системы, обслуживание |
| 4 | Услуга | Предоставление, расписание, стоимость |
| 5 | Приём | Запись, проведение, статус |
| 6 | Лечение | Назначение, ведение, завершение |
| 7 | Медикамент | Использование, дозировка, производитель |
| 8 | Связь Лечение-Медикамент | Присвоение медикаментов лечению |
| 9 | Связь Приём-Услуга | Предоставление услуг во время приёма |

Для выполнения задач, описанных в Таблице 1 "Объекты предметной области", система учета пациентов в стоматологической клинике опирается на данные, которые хранятся в базе данных.

Ниже представлены таблицы, которые детально описывают каждую сущность, её атрибуты, типы данных этих атрибутов, а также краткое описание каждого из них.

Таблица 2 - Атрибуты сущности «Пациент»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| patient\_id | INT | Уникальный идентификатор пациента (авто инкремент) |
| first\_name | VARCHAR (50) | Имя пациента |
| last\_name | VARCHAR (50) | Фамилия пациента |
| middle\_name | VARCHAR (50) | Отчество пациента |
| birth\_date | DATE | Дата рождения пациента |
| gender | CHAR (1) | Пол пациента (М/Ж) |
| phone | VARCHAR (20) | Контактный телефон пациента |
| email | VARCHAR (100) | Электронная почта пациента |
| address | VARCHAR (200) | Адрес проживания пациента |
| passport\_number | VARCHAR (20) | Номер паспорта пациента |
| insurance\_number | VARCHAR (20) | Номер страхового полиса пациента |

Таблица 3 - Атрибуты сущности «Врач»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| doctor\_id | INT | Уникальный идентификатор врача (авто инкремент) |
| first\_name | VARCHAR (50) | Имя врача |
| last\_name | VARCHAR (50) | Фамилия врача |
| middle\_name | VARCHAR (50) | Отчество врача |
| specialization | VARCHAR (100) | Специализация врача |
| phone | VARCHAR (20) | Контактный телефон врача |
| email | VARCHAR (100) | Электронная почта врача |

Таблица 4 - Атрибуты сущности «Администратор»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| admin\_id | INT | Уникальный идентификатор администратора (авто инкремент) |
| username | VARCHAR (50) | Имя пользователя для входа |
| password | VARCHAR (255) | Пароль администратора |
| role | VARCHAR (20) | Роль администратора в системе |

Таблица 5 - Атрибуты сущности «Услуга»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| service\_id | INT | Уникальный идентификатор услуги (авто инкремент) |
| service\_name | VARCHAR (100) | Название услуги |
| description | TEXT | Описание услуги |
| price | DECIMAL (10,2) | Стоимость услуги |

Таблица 6 - Атрибуты сущности «Приём»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| appointment\_id | INT | Уникальный идентификатор приёма (авто инкремент) |
| patient\_id | INT | Идентификатор пациента (внешний ключ) |
| doctor\_id | INT | Идентификатор врача (внешний ключ) |
| appointment\_date | DATETIME | Дата и время приёма |
| status | VARCHAR (20) | Статус приёма (например, "Запланирован", "Проведён") |
| notes | TEXT | Заметки о приёме |

Таблица 7 - Атрибуты сущности «Лечение»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| treatment\_id | INT | Уникальный идентификатор лечения (авто инкремент) |
| patient\_id | INT | Идентификатор пациента (внешний ключ) |
| doctor\_id | INT | Идентификатор врача (внешний ключ) |
| start\_date | DATE | Дата начала лечения |
| end\_date | DATE | Дата окончания лечения |
| diagnosis | TEXT | Диагноз |
| treatment\_plan | TEXT | План лечения |

Таблица 8 - Атрибуты сущности «Медикамент»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| medication\_id | INT | Уникальный идентификатор медикамента (авто инкремент) |
| name | VARCHAR (100) | Название медикамента |
| description | TEXT | Описание медикамента |
| dosage | VARCHAR (50) | Дозировка медикамента |
| manufacturer | VARCHAR (100) | Производитель медикамента |

Таблица 9 - Атрибуты сущности «Связь Лечение-Медикамент»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| treatment\_id | INT | Идентификатор лечения (внешний ключ) |
| medication\_id | INT | Идентификатор медикамента (внешний ключ) |

Таблица 10 - Атрибуты сущности «Связь Приём-Услуга»

| Атрибут | Тип данных | Описание |
| --- | --- | --- |
| appointment\_id | INT | Идентификатор приёма (внешний ключ) |
| service\_id | INT | Идентификатор услуги (внешний ключ) |

* 1. **План тестирования**

Тестирование данного модуля направлено на обеспечение надежной и эффективной работы системы учета пациентов в стоматологической клинике. Основная цель тестирования — убедиться, что система соответствует заявленным требованиям и функциональным возможностям.

Задачи тестирования:

1. Функциональная: Проверка правильности выполнения всех функций системы, включая регистрацию пациентов, управление записями на прием, ведение историй болезней

2. Производительности и стабильности: Оценка времени отклика системы при различных нагрузках: Тестирование системы на стабильность при длительной работе и большом количестве пользователей.

3. Удобство и юзабилити (графический интерфейс): Оценка удобства и интуитивности пользовательского интерфейса. Проверка корректности отображения всех элементов интерфейса на различных устройствах и разрешениях экранов.

4. Обеспечение безопасности данных: Проверка защиты данных от несанкционированного доступа. Тестирование механизмов аутентификации и авторизации пользователей.

5. Совместимость с различными ПО и аппаратными средствами (Python, PyQt6, SQlite3): Проверка совместимости системы с различными версиями Python и PyQt6.

Таким образом, система учета в стоматологической клинике автоматизирует работу с данными о пациентах и услугах, что значительно упрощает и ускоряет процессы управления. Она позволяет эффективно регистрировать пациентов, вести медицинскую документацию, управлять расписанием приема и формировать отчеты. Это помогает повысить качество обслуживания, улучшить взаимодействие между врачами и административным персоналом, а также обеспечить более точный учет всех медицинских процедур и финансовых операций. Система также способствует улучшению контроля за работой клиники, что в конечном итоге приводит к повышению удовлетворенности пациентов и эффективности работы всего медицинского учреждения.

**3. Реализация проекта системы**

**3.1 Описание среды разработки**

Для реализации проекта системы учета пациентов в стоматологической клинике потребуется использование языка программирования и среды разработки, обеспечивающих эффективную разработку и управление кодом.

* + 1. **Выбор и описание программных инструментов**

Для разработки системы стоматологической клиники были выбраны следующие программные инструменты:

Python [7], [9], [10], [11], — это высокоуровневый язык программирования, который широко используется для разработки приложений, веб-сервисов, научных вычислений и автоматизации задач.

SQLite [15] — это легковесная встраиваемая реляционная база данных, идеально подходящая для небольших приложений, где требуется хранение данных в структурированном виде.

Bcrypt [14]— это криптографическая библиотека, обеспечивающая безопасное хранение паролей путем хеширования с использованием соли.

PyQt6 [7], [9], [10], [11], — это набор инструментов для создания графического интерфейса пользователя (GUI) на Python, предоставляющий богатый набор виджетов и функциональных возможностей.

Openpyxl [11]— это библиотека для работы с файлами Excel, позволяющая читать, записывать и изменять данные, а также применять стили и форматирование.

Модуль re [11] в Python используется для работы с регулярными выражениями, позволяя выполнять поиск, замену и проверку строк на соответствие определенным шаблонам.

**3.1.2 Обоснование выбора инструментов**

Python выбран из-за его простоты, универсальности и богатой экосистемы библиотек. Он позволяет быстро разрабатывать приложения, а его синтаксис делает код легко читаемым и поддерживаемым.

SQLite используется для хранения структурированных данных без необходимости установки отдельного сервера БД, что удобно для небольших приложений.

Bcrypt обеспечивает надежную защиту паролей от атак перебора и подбора благодаря хешированию с солью.

PyQt6 позволяет создавать красивые и функциональные графические интерфейсы, адаптируемые под разные платформы.

Openpyxl предоставляет удобные инструменты для работы с файлами Excel: чтение, запись и форматирование данных.

Модуль re используется для эффективной обработки текста, например валидации пользовательского ввода или поиска шаблонов.

Выбранный стек обеспечивает все необходимое для создания надежного, безопасного и функционального приложения с удобным интерфейсом.

**3.2 Разработка программного модуля**

Для успешного взаимодействия пользователей с системой учета пациентов в стоматологической клинике был разработан функциональный пользовательский интерфейс. В его реализации предусмотрено размещение кнопок, отображение информации и обработка действий пользователя. Ключевые аспекты создания программного продукта включают четкое представление о логике и структуре разработки программного модуля.

Создание программного продукта для системы учета пациентов в стоматологической клинике включает в себя несколько ключевых аспектов: создание базы данных, разработку пользовательского интерфейса, реализацию механизмов авторизации и аутентификации, создание форм для регистрации новых пациентов и врачей, разработку функций для управления приемами и графиком работы врачей, а также реализацию механизмов безопасности.

**3.2.1 Реализация пользовательского интерфейса программы**

При запуске приложения пользователь видит окно авторизации (ПРИЛОЖЕНИЕ 3, рисунок 4). После ввода логина и пароля определяется роль пользователя.

После авторизации пользователи могут иметь следующие роли:

1. Администратор: управление пользователями (добавление, удаление, редактирование пациентов и врачей). Управление приемами, подтверждение записей на прием. Экспорт медицинских карт в формат Excel.
2. Врач: просмотр и редактирование записей на прием. Добавление и редактирование медицинских карт пациентов.
3. Пациент: просмотр своих записей на прием. Просмотр информации о врачах. Просмотр своей медицинской карты. Запись на прием.

Если пациент успешно войдет в свой аккаунт, то он увидит: свои фамилию, имя, отчество, таблицу с записями на прием, таблицу с информацией о врачах, таблицу со своей медицинской картой, кнопку для записи на прием (ПРИЛОЖЕНИЕ 3, рисунок 5–7).

Если врач успешно войдет в свой аккаунт, то он увидит: свои фамилию, имя, отчество, таблицу с записями на прием, таблицу с медицинскими картами пациентов, кнопки для редактирования записей на прием и добавления медицинских карт (ПРИЛОЖЕНИЕ 3, рисунок 8–12).

Если администратор успешно войдет в свой аккаунт, то он увидит: вкладки для управления пользователями, приемами, расписание врачей, и экспорта медицинских карт (ПРИЛОЖЕНИЕ 3, рисунок 13–21).

**3.2.2 Описание кодом функциональных узлов модуля**

После включения приложения пользователь увидит окно авторизации, введя свои данные и войдя в аккаунт программа сама определит его роль. Также в окне авторизации пациенту нужно будет пройти регистрацию создание окна регистрацию, и функция регистрации (def login) находится в классе class PatientRegisterWindow (ПРИЛОЖЕНИЕ 3, рисунок 22–28)

* Создание окна авторизации и функция авторизации (def login) находится в классе LoginWindow (ПРИЛОЖЕНИЕ 3, рисунок 29–31).

Если пациент войдет в свой аккаунт, то для него откроется окно пациента, в котором будет находиться:

Таблица с записями на прием (def update\_appointments\_table) (ПРИЛОЖЕНИЕ 3, рисунок 32). Таблица с информацией о врачах (def update\_doctors\_table) (ПРИЛОЖЕНИЕ 3, рисунок 33). Таблица со своей медицинской картой (def update\_medical\_record\_table) (ПРИЛОЖЕНИЕ 3, рисунок 34). Код для записи на прием (def show\_appointment\_window). (ПРИЛОЖЕНИЕ 3, рисунок 35)

Если врач войдет в свой аккаунт, то для него откроется окно врача, в котором будут находиться:

Таблица с записями на прием (def load\_appointments). Таблица с медицинскими картами пациентов (def load\_patient\_medical\_records). Функция для редактирования записей на прием (def edit\_appointment) и добавления медицинских карт (def add\_treatment) (ПРИЛОЖЕНИЕ 3, рисунок 36–39).

Если администратор войдет в свой аккаунт, то он увидит вкладку добавления пациента. Также в этом окне находятся еще вкладки для управления пользователями, приемами и экспорта медицинских карт.

* Отображение окна администратора, отображение виджетов во вкладке создания пациента и функция для добавления пациента (def add\_patient) (ПРИЛОЖЕНИЕ 3, рисунок 40) находятся в классе MainWindow.

Таким образом, разработанный модуль системы учета пациентов в стоматологической клинике обеспечивает авторизацию пользователей, отображение записей на прием, информации о врачах и медицинских карт пациентов, а также административные инструменты для управления данными. Подробно о каждой функции и детали приложения можете узнать на сайте GitHub [6], [12].

**3.2.3 Результат работы и тестирования**

Проведение тестирования окна авторизации (ПРИЛОЖЕНИЕ 3, рисунок 41, 44).

* Проверка входа с правильными и неправильными данными.
* Проверка отображения соответствующего окна в зависимости от роли пользователя.

Проверка на создание пациента в окне авторизации (ПРИЛОЖЕНИЕ 3, рисунок 45)

* Проверка добавления нового пациента через окно администратора. (ПРИЛОЖЕНИЕ 3, рисунок 46)
* Проверка отображения нового пациента в списке пациентов (ПРИЛОЖЕНИЕ 3, рисунок 47)

Теперь надо проверить, появился ли новый пациент в списке его пациентов при запросе на приём

* Проверка на возможность пациента запросить заявку на приём у администратора (ПРИЛОЖЕНИЕ 3, рисунок 48).
* Проверка отображения пациента в базе данных у администратора (ПРИЛОЖЕНИЕ 3, рисунок 49).
* Проверка отображения нового пациента в списке пациентов врача (ПРИЛОЖЕНИЕ 3, рисунок 50).
* Проверка возможности врача добавлять и редактировать медицинские карты пациента (ПРИЛОЖЕНИЕ 3, рисунок 51).

Проверка отображения в окне пациента записей в медицинской карте (ПРИЛОЖЕНИЕ 3, рисунок 52).

Остальные позитивные и негативные тесты, в которых были проверены основные функции приложения, были проведены в (ПРИЛОЖЕНИЕ 2, таблица 11).

Таким образом, в результате тестирования окна авторизации, добавления пациента, добавления записей на прием и медицинских карт, было подтверждено корректное функционирование основных функций системы учета пациентов в стоматологической клинике.

**ЗАКЛЮЧЕНИЕ**

В ходе курсового проекта была разработана система учета пациентов в стоматологической клинике, направленная на автоматизацию процессов сбора, хранения и обработки данных о пациентах и предоставляемых услугах. Основная цель проекта заключалась в улучшении управления медицинскими процессами, сокращении трудозатрат персонала и обеспечении удобного доступа к информации для всех участников рабочего процесса.

Основные результаты проекта включают анализ специфики работы стоматологических клиник и требований к системам учета пациентов, определение требований к функциям системы, включая регистрацию пациентов, ведение медицинской документации, управление расписанием и оплатой, а также формирование отчетности. Выбраны и обоснованы инструменты разработки (Python, SQLite, PyQt6, bcrypt, openpyxl, модуль re ). Разработан пользовательский интерфейс и реализованы основные функциональные узлы модуля. Проведено тестирование основных функций системы, подтвердившее ее корректное функционирование.

Ожидаемые эффекты от внедрения системы включают повышение прозрачности медицинского процесса, ускорение процесса принятия решений, снижение нагрузки на сотрудников клиники, улучшение качества обслуживания пациентов и соответствие законодательным требованиям по защите персональных данных.

Разработанная система учета пациентов в стоматологической клинике является важным шагом к модернизации работы медицинского учреждения. Внедрение данной системы позволит автоматизировать ключевые бизнес-процессы, повысить точность и оперативность учета информации о пациентах, а также улучшить качество предоставляемых услуг.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

Нормативно-правовые источники

1. Федеральный закон "О персональных данных" (от 27.07.2006 N 152-ФЗ) регулирует порядок обработки персональных данных. Он определяет права субъектов персональных данных и обязанности операторов, что является критически важным в условиях цифровизации и массового сбора данных.
2. Федеральный закон "Об информации, информационных технологиях и о защите информации" (N 149-ФЗ) закладывает правовые основы оборота информации, использования информационных технологий и обеспечения ее защиты. Это способствует созданию безопасной информационной среды и поддерживает развитие цифровой экономики.
3. ГОСТ Р ИСО/МЭК 12207–2010 Системная и программная инженерия. Процессы жизненного цикла программных средств.
4. Федеральный закон "О безопасности критической информационной инфраструктуры Российской Федерации" (от 26.07.2017 N 187-ФЗ) фокусируется на вопросах обеспечения безопасности критически важной информационной инфраструктуры, включая защиту от кибератак. Это особенно актуально в условиях возрастающих угроз в киберпространстве.
5. ГОСТ 19.701–90 Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения.

Монографии

1. Грекович С. Н. Информационная система «Системы учёта пациентов в стоматологической клинике», ссылка на репозиторий GitHub:   
   https://github.com/keyvvest/Dantistikss.git

Онлайн-курсы

1. Python-разработчик //https://skillbox.ru/course/profession-python/
2. Интерактивный тренажер по SQL //https://sql-academy.org/ru
3. Поколение Python": курс для продвинутых //https://stepik.org/course/67/promo
4. Основы программирования на Python //https://stepik.org/course/58852/promo

Интернет-источники

1. PyCharm Documentation. (Официальная документация PyCharm) //https://www.jetbrains.com/pycharm/documentation
2. Python Documentation (Официальная документация Python). Руководство по языку программирования Python, включая стандартную библиотеку, учебные пособия и справочные материалы //https://www.python.org
3. GitHub Documentation: GitHub Guides, GitHub Docs, GitHub Learning Lab. Полное руководство по использованию платформы для хостинга IT-проектов и их совместной разработки //https://github.com
4. SQL Tutorial and Reference Guide. Интерактивный учебник по SQL с практическими примерами и упражнениями для освоения языка структурированных запросов //https://sqlzoo.net/wiki/
5. Bcrypt Documentation and Implementation Guide. Техническая документация по использованию библиотеки Bcrypt для хеширования паролей в Python-приложениях //https://zetcode.com/python/bcrypt
6. SQLite Documentation (Официальная документация SQLite) //https://www.sqlite.org/docs.html

**ПРИЛОЖЕНИЕ 1**

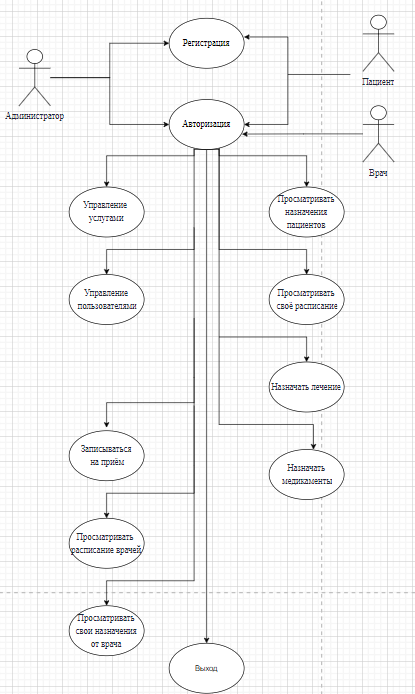


Рисунок 1 - Диаграмма прецедентов

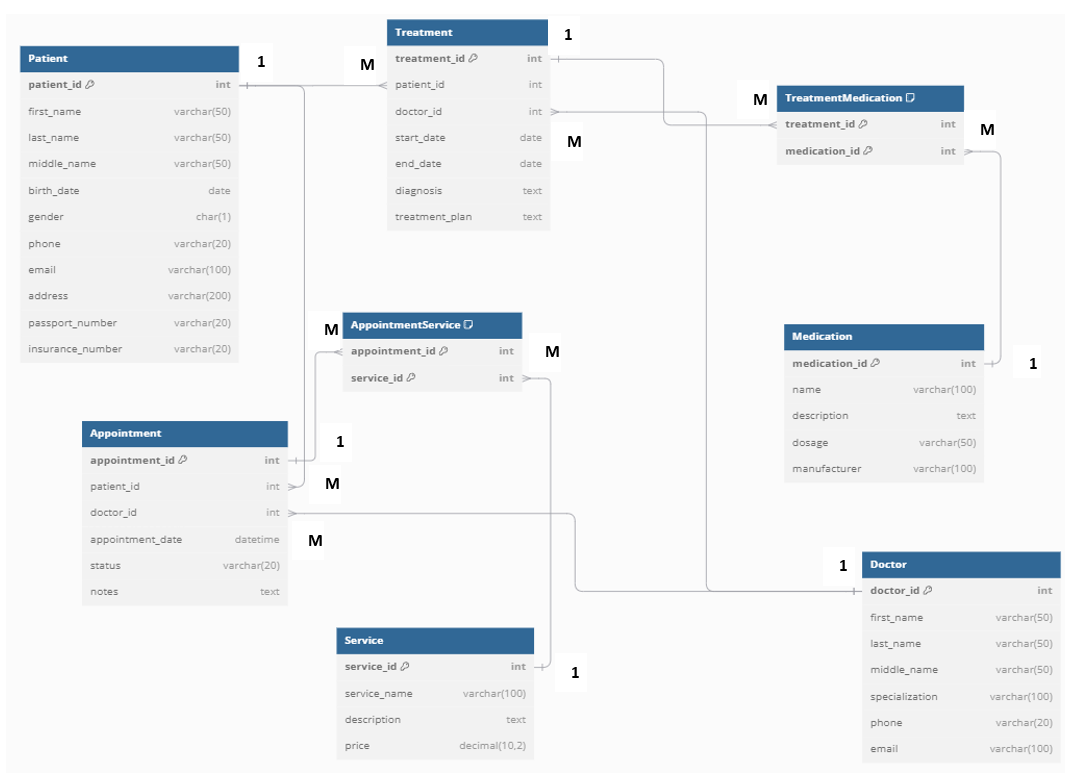


Рисунок 2 - Диаграмма IDEF1X

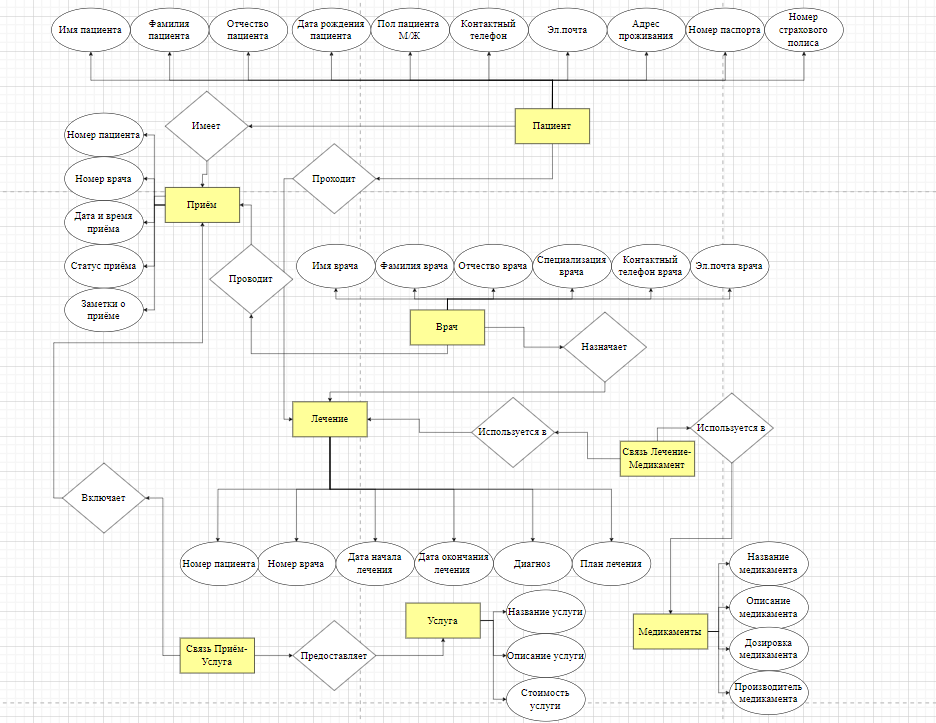


Рисунок 3 - ER диаграмма.

**ПРИЛОЖЕНИЕ 2**

Таблица 11 - План тестирования

| Порядковый номер | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
| --- | --- | --- | --- | --- | --- |
| 1 | Авторизация пользователя | Логин, пароль | Вводим корректные данные: Логин: admin Пароль: password123 | Успешный вход | Успешный вход |
|  |  |  |  |  | |
|  |  |  | Вводим некорректные данные: Логин: admin  Пароль: password | Ошибка: неверное имя пользователя или пароль | Ошибка: неверное имя пользователя или пароль |
|  |  |  |  |  | |
|  |  |  | Оставляем поле логина пустым | Ошибка: неверное имя пользователя или пароль | Ошибка: неверное имя пользователя или пароль |

Продолжение таблицы 11

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Порядковый номер | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | | Результат тестирования |
|  |  |  |  |  | | |
|  |  |  | Оставляем поле пароля пустым | Ошибка: неверное имя пользователя или пароль | Ошибка: неверное имя пользователя или пароль | |
|  |  |  |  |  | | |
| 2 | Регистрация пациента | Имя пользователя, пароль, имя, фамилия, отчество, пол, телефон, дата рождения, серия и номер паспорта, номер полиса | Вводим корректные данные | Регистрация прошла успешно | Регистрация прошла успешно | |

Продолжение таблицы 11

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Порядковый номер | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | | Результат тестирования |
|  |  |  | Введенные данные:  Имя пользователя: svetlana  Пароль: 040406cc  Подтверждение пароля: 040406cc  Имя: Светлана  Фамилия: Грекович  Отчество: Николаевна  Дата рождения: 2006-04-04  Пол: Ж  Телефон: +79321458996  Email: lana.@mail.ru  Адрес: Москва  Серия паспорта:6565  Номер паспорта:58963  Номер страховки:7854589653265869 |  | | |
|  |  |  | Поля: Имя, Фамилия содержат цифры | Ошибка: поля Имя, Фамилия, Отчество должны содержать только русский текст и начинаться с заглавной буквы | Ошибка: поля Имя, Фамилия, Отчество должны содержать только русский текст и начинаться с заглавной буквы | |

Продолжение таблицы 11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Порядковый номер | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  | 1) Введено на месте поля «Имя» число 99666 |  | |
|  |  |  | 2) Введено на месте поля «Фамилия» 5555 |  | |
|  |  |  | 3) Введено на месте «Отчество» 2006 |  | |

Продолжение таблицы 11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Порядковый номер | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  | Поле телефон: некорректный формат 89653462518 | Ошибка: телефон должен быть в формате +7XXXXXXXXXX | Ошибка: телефон должен быть в формате +7XXXXXXXXXX |
|  |  |  |  |  | |
|  |  |  | Поле дата рождения: некорректный формат 4 августа | Ошибка: дата должна быть в формате ГГГГ-ММ-ДД | Ошибка: дата должна быть в формате ГГГГ-ММ-ДД |
|  |  |  |  |  | |
|  |  |  | Поле серия паспорта: некорректный формат | Ошибка: серия паспорта должна содержать только 4 цифры, номер паспорта (6 цифр), номер страховки 16) | Ошибка: серия паспорта должна содержать только 4 цифры, номер паспорта ( 6 цифр), номер страховки 16) |
|  |  |  | 1) Серия паспорта; 6500 |  | |
|  |  |  | 2) Номер Паспорта: 0 |  | |

Продолжение таблицы 11

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Порядковый номер | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования | |
|  |  |  | 3) Номер страховки:2 |  | | |
| 3 | Просмотр пациентов | Таблица пациентов | Заходим в окно админа под паролем: passwrd123  имя пользователя: admin.  При авторизации нажимаем на вкладку "Пациенты" | Отображается список всех пациентов | | Отображается список всех пациентов |
|  |  |  |  |  | | |
|  |  |  | Удаляем пациента из таблицы. Нажимаем на ненужного пациента будет автоматически удален. | Пациент успешно удален | | Пациент успешно удален |
|  |  |  |  |  | | |

Продолжение таблицы 11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Порядковый номер | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
| 4 | Просмотр врачей | Таблица врачей | Открываем вкладку "Врачи" в окне администратора. | Отображается список всех врачей | Отображается список всех врачей |
|  |  |  |  |  | |
|  |  |  | Добавляем врача. | Регистрация прошла успешно | Регистрация прошла успешно |
|  |  |  | Ввод:  Имя пользователя: irina  Пароль:040608iii  Имя: Ирина  Фамилия: Ивирсова  Отчество: Викторовна  Специализация: Стоматолог-хирург  Телефон:+79365485236  Email: irina@dent.com |  | |
|  |  |  | Удаляем врача из таблицы | Врач успешно удален | Врач успешно удален |
|  |  |  |  |  | |

Продолжение таблицы 11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Порядковый номер | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  | Редактирование врача. | Данные врача успешно обновлены | Данные врача успешно обновлены |
|  |  |  | Изменился номер телефона на +79321244589 |  | |
| 5 | Подтверждение приема | Приёмы | Нажимаем на поле с заявкой на приём, чтобы запись подтвердилась. | Запись на приём подтверждена | Запись на приём подтверждена |
|  |  |  |  |  | |

Продолжение таблицы 11

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Порядковый номер | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | | Результат тестирования |
|  |  |  | Редактируем данные о приёме. | Данные о приёме успешно обновлены. | | Данные о приёме успешно обновлены |
|  |  |  | Добавлено время 18:00:00 |  | | |
| 6 | Записываться на приём | Пациент, врач, дата приема | Пациент запрашивает запись на день и времякоторое ему удобно, а потом с администратором запись подтверждают либо изменяют. | Запись на приём к врач успешно создана | Запись на приём к врачу успешно создана | |
|  |  |  | Пациент зарегистрировался к Анне Смирновой на 2024-12-12 и ждёт подтверждения от администратора |  | | |
|  | Редактирование приёма | Таблица врачей | Изменять статус, назначать план лечения, добавлять заметки, изменять медицинскую запись | Медицинская запись успешно обновлена | Медицинская запись успешно обновлена | |

Продолжение таблицы 11

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Порядковый номер | Наименование функциональности | Наименование поля | Тестовый набор | Ожидаемый результат | Результат тестирования |
|  |  |  |  |  | |

**ПРИЛОЖЕНИЕ 3**

Скриншоты графического интерфейса

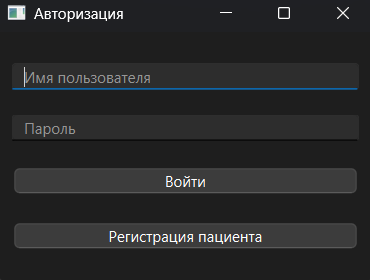


Рисунок 4 – Окно авторизации.

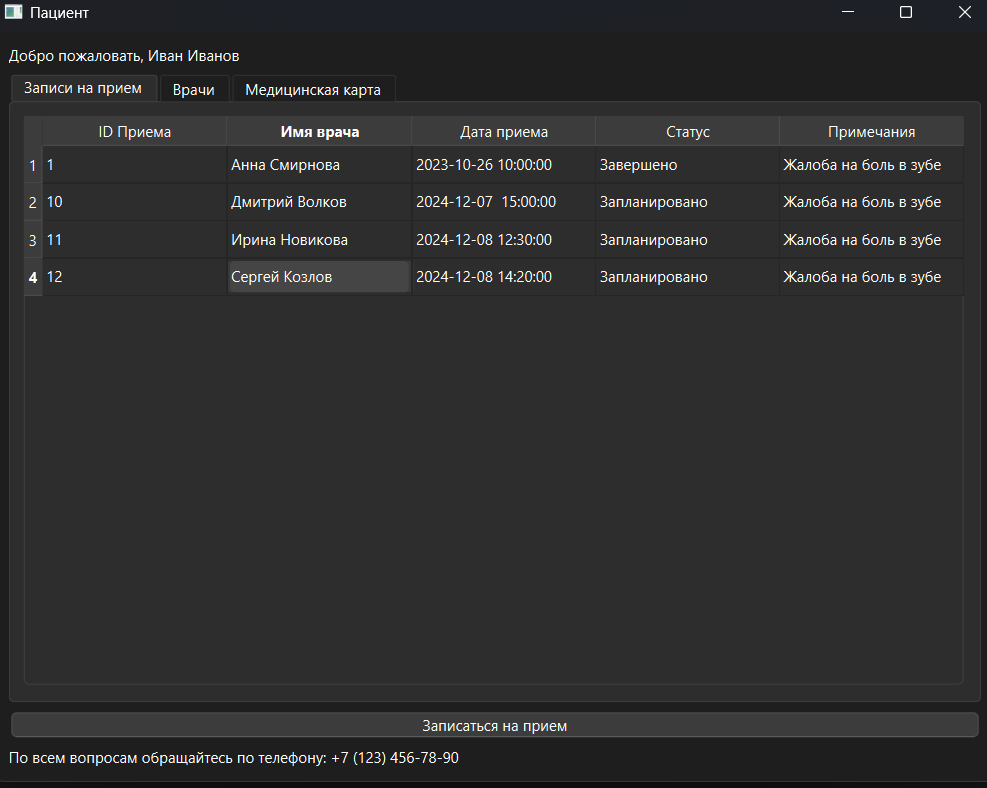
****

Рисунок 5 – Окно пациента.

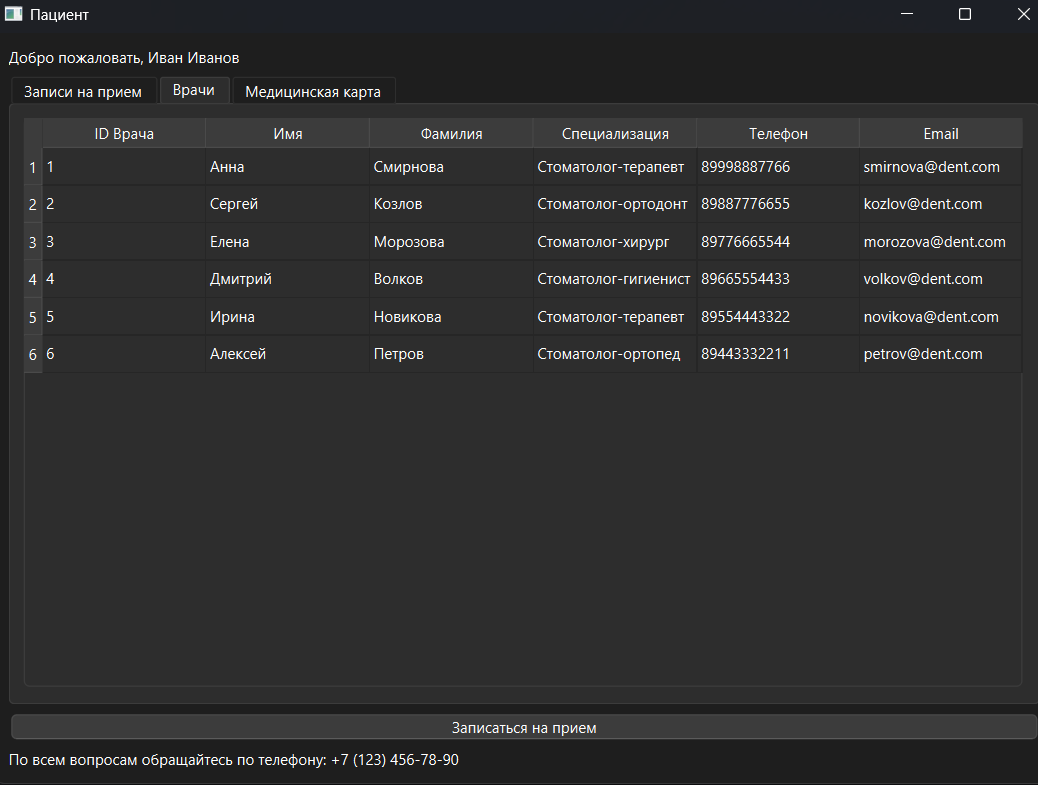


Рисунок 6 – Окно пациента. Вкладка «Врачи».

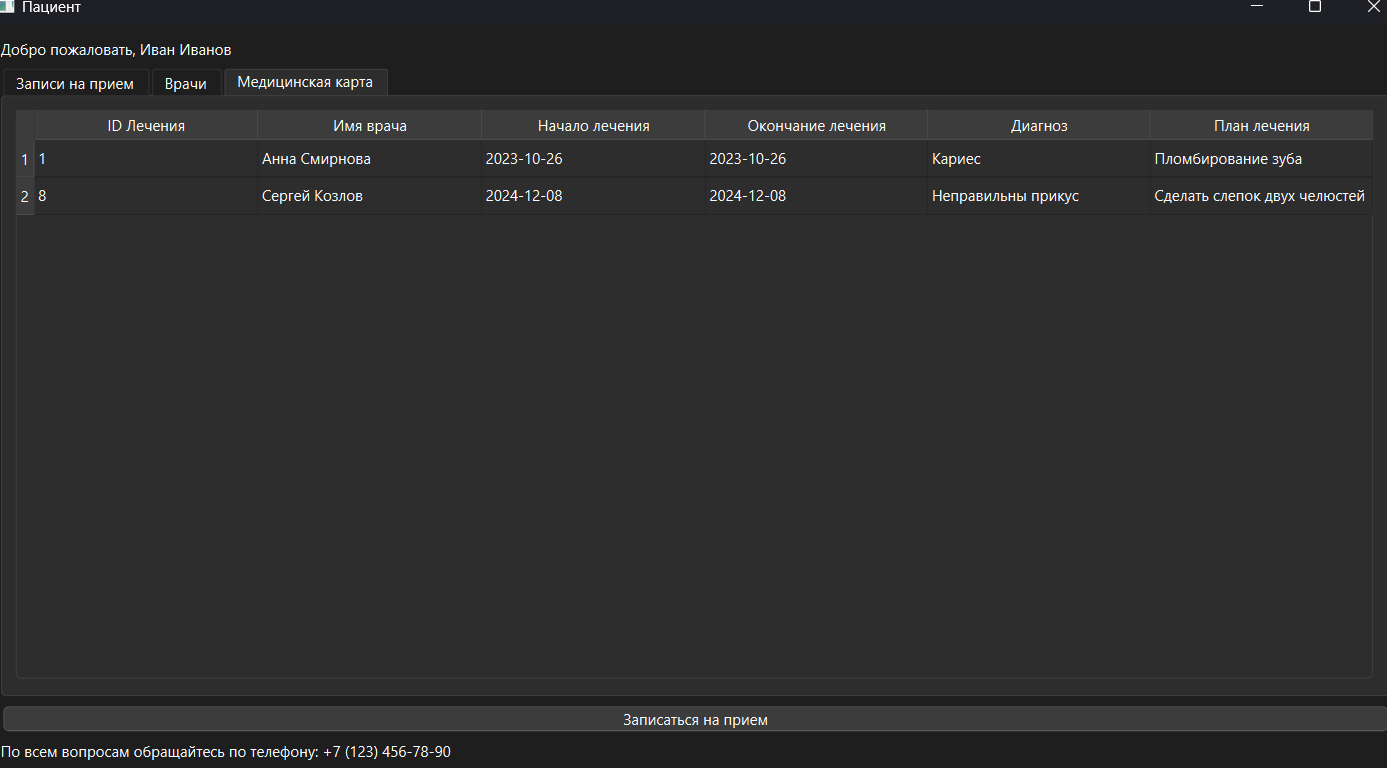
****

Рисунок 6 – Окно пациента при просмотре своей медицинской карты.

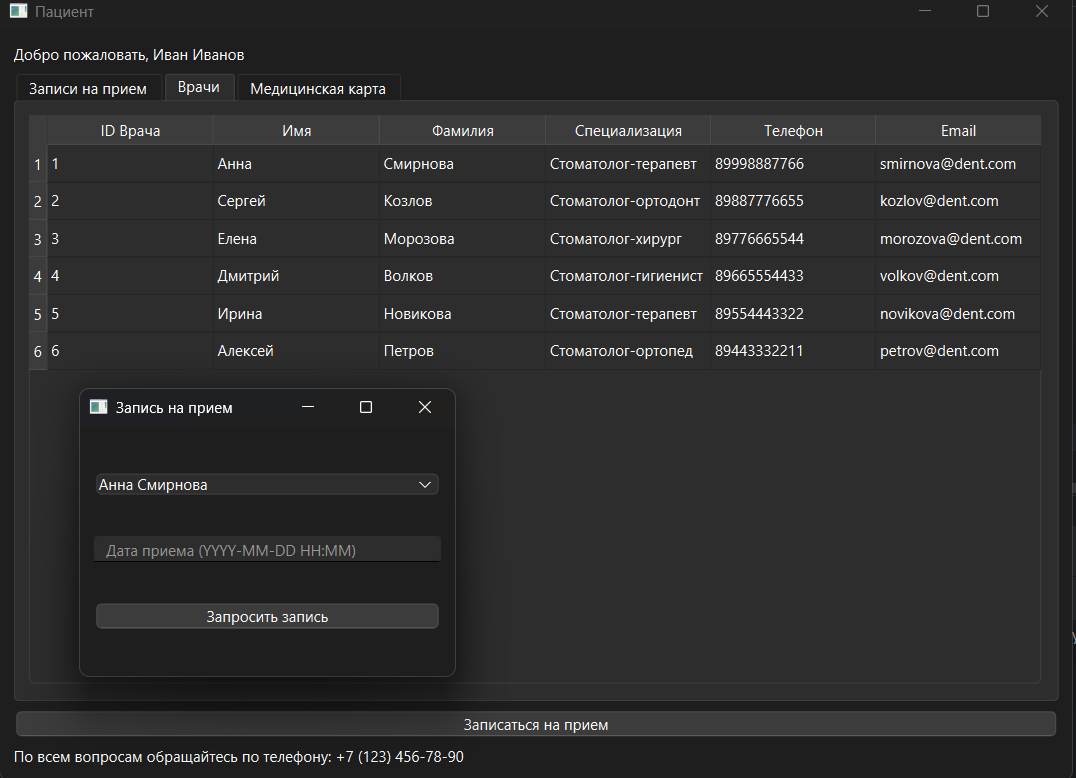


Рисунок 7 – Окно пациента после нажатия на кнопку «Записаться на приём».

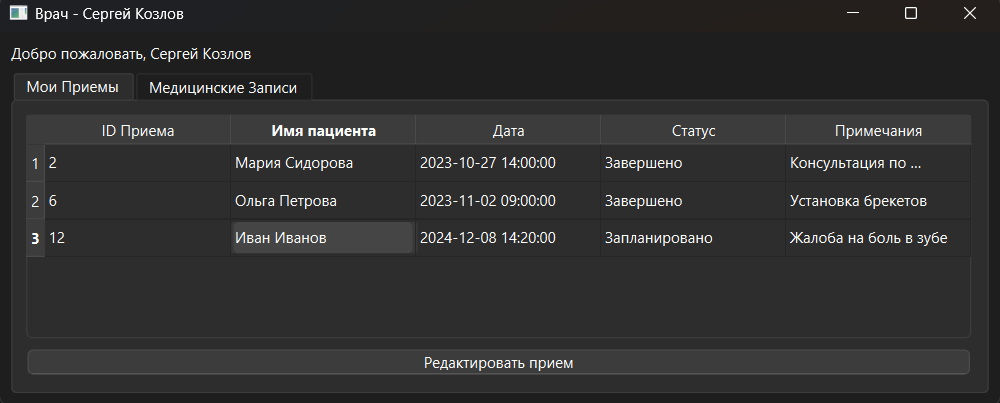


Рисунок 8 – Окно врача. Вкладка «Мои Приёмы». Нажимаем на пациента и переходим во вкладку «Медицинские Записки».

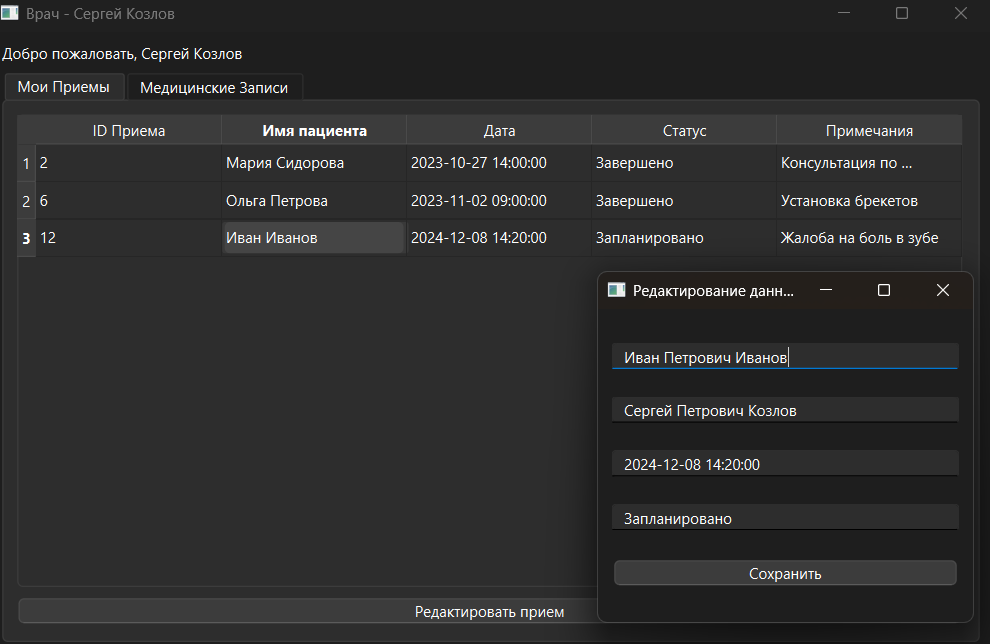


Рисунок 9 – Окно врача. Вкладка «Мои приёмы». После нажатия на кнопку «Редактировать приём» высвечивается окно, показанное на рисунке 9.

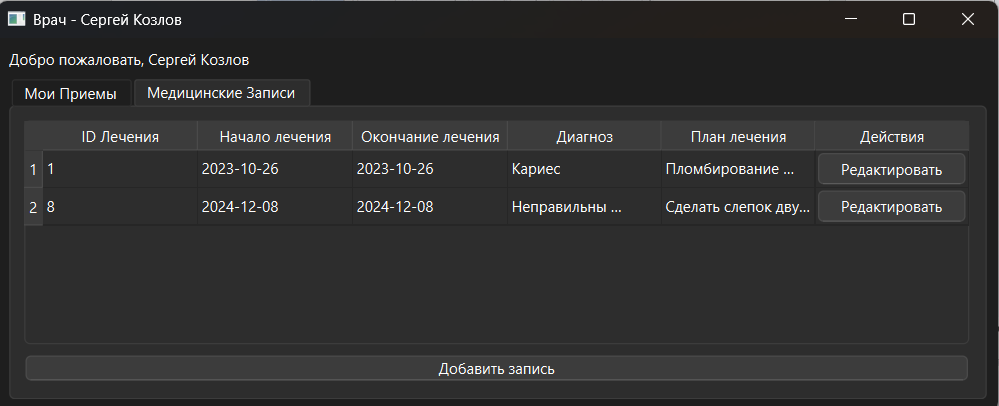


Рисунок 10 – Окно врача. Вкладка «Медицинские Записи».

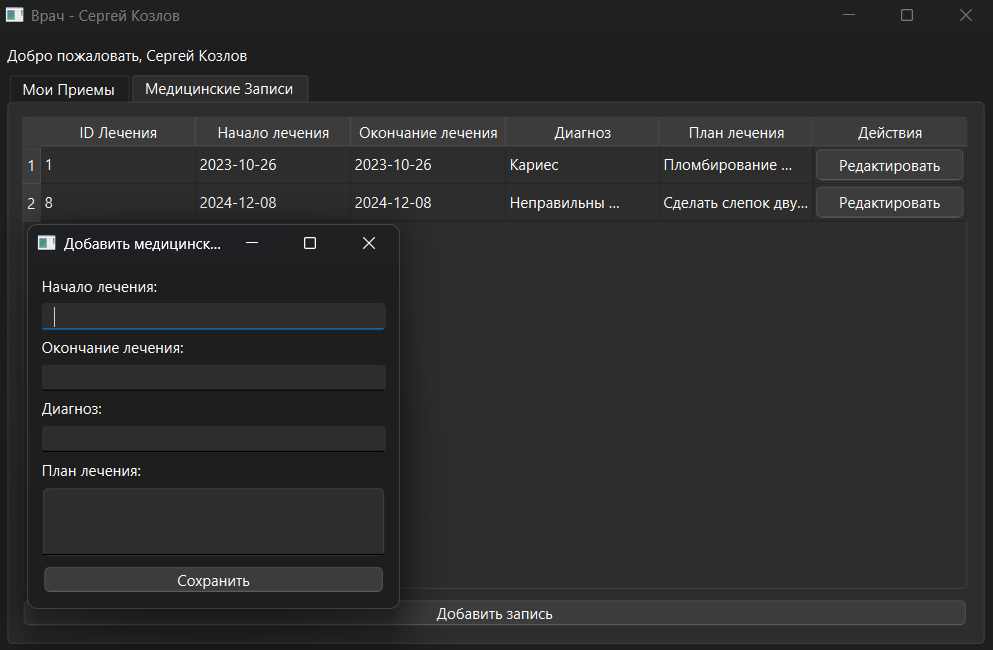


Рисунок 11– Окно врача. Вкладка «Медицинские Записки». После нажатия на кнопку «Добавить запись» высвечивается окно, показанное на рисунке 11.

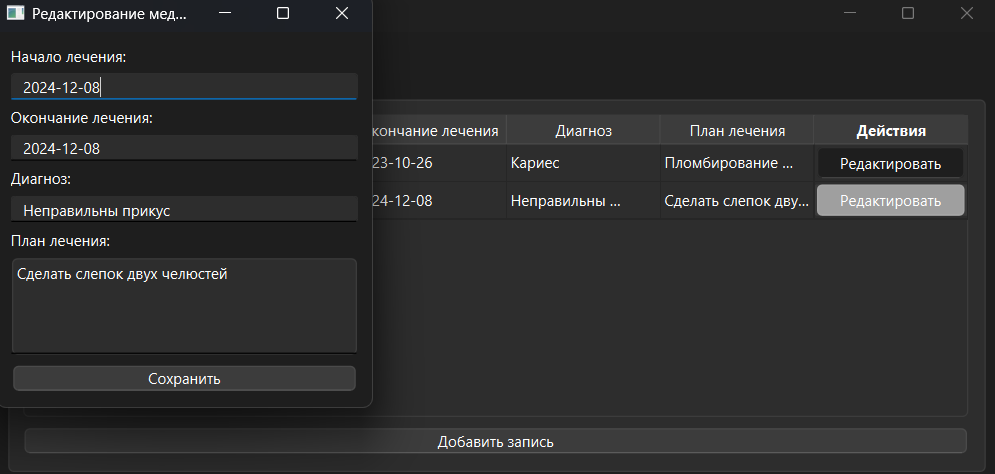


Рисунок 12– Окно врача. Вкладка «Медицинские Записки». После нажатия на кнопку «Редактировать» высвечивается окно, показанное на рисунке 12.

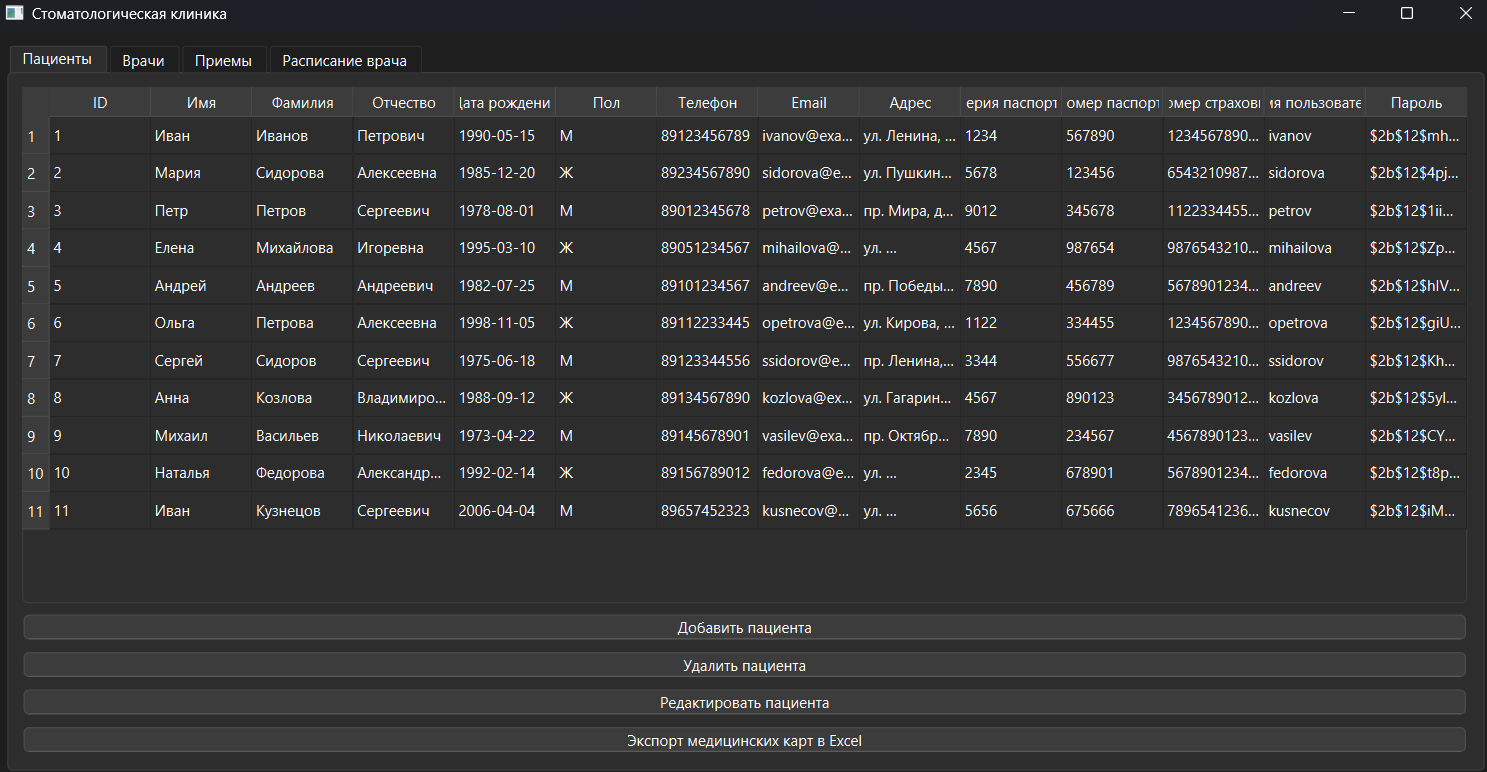


Рисунок 13 – Окно администратора, вкладка: «Пациенты».

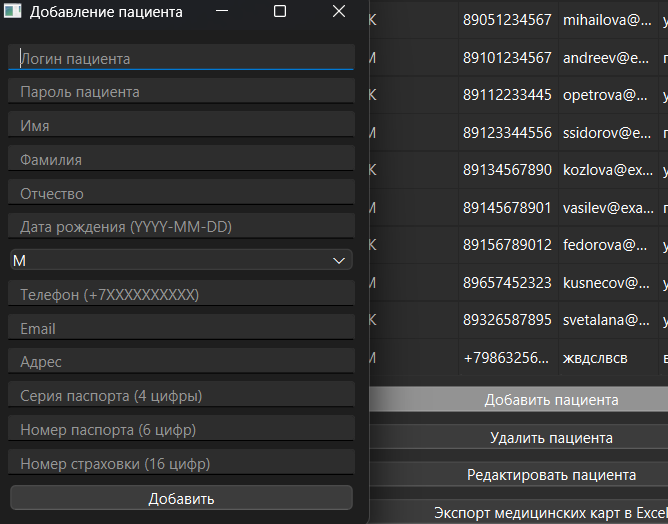


Рисунок 14 – Окно администратора, вкладка: «Пациенты». Вывод окна после нажатия «Добавить пациента».

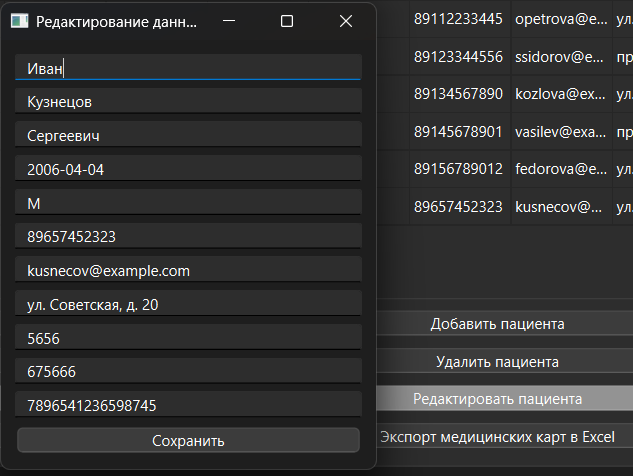


Рисунок 15 – Окно администратора, вкладка: «Пациенты». Вывод окна после нажатия «Редактировать пациента».

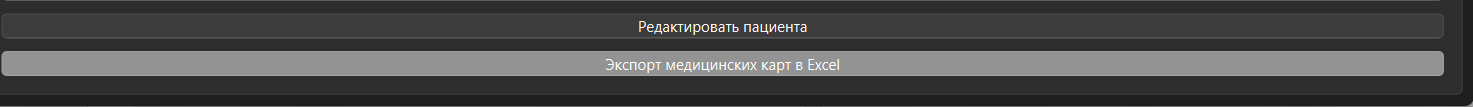
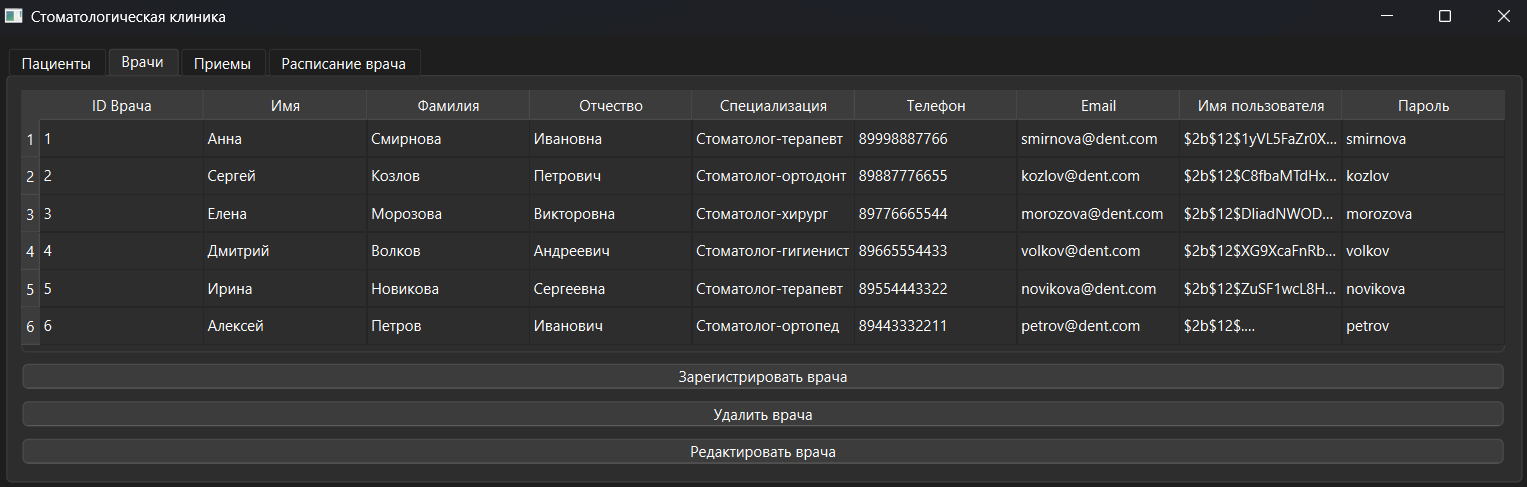
 



Рисунок 16– Окно администратора, вкладка: «Пациенты».

Вывод данных после нажатии кнопки «Экспорт медицинских карт в Excel».



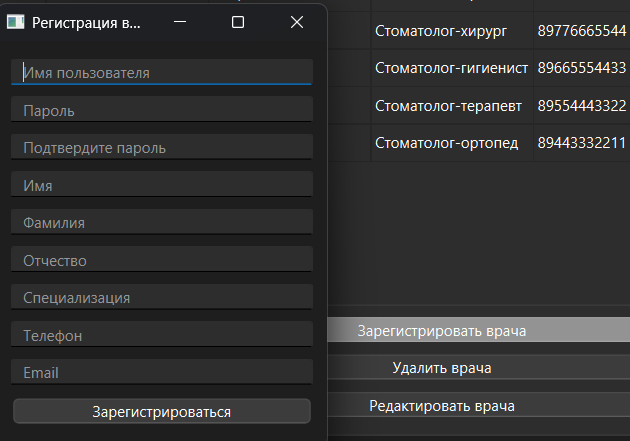


Рисунок 17 – Окно администратора, вкладка: «Врачи». Вывод окна после нажатия кнопки «Редактировать врача».

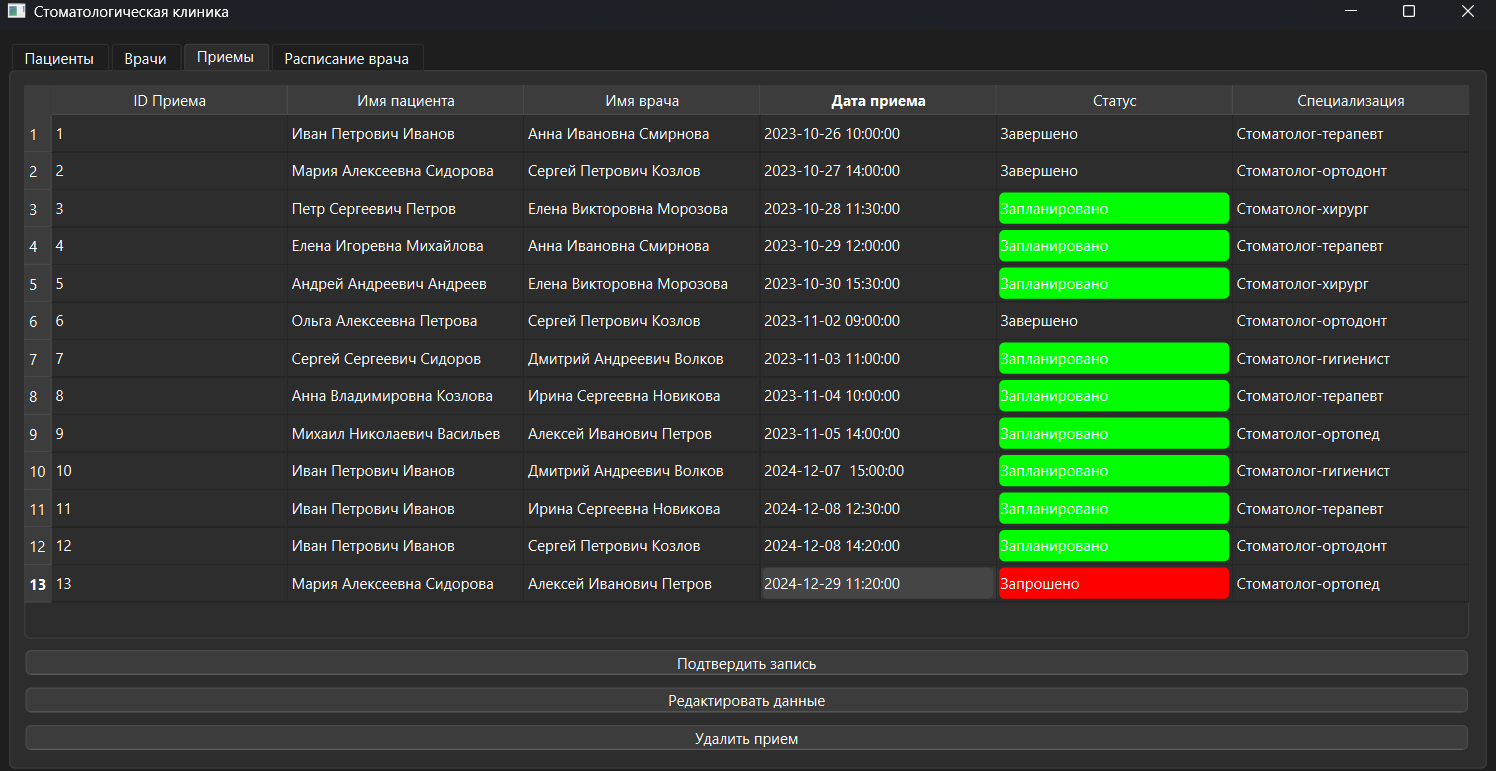


Рисунок 18 – Окно администратора, вкладка: «Приёмы».

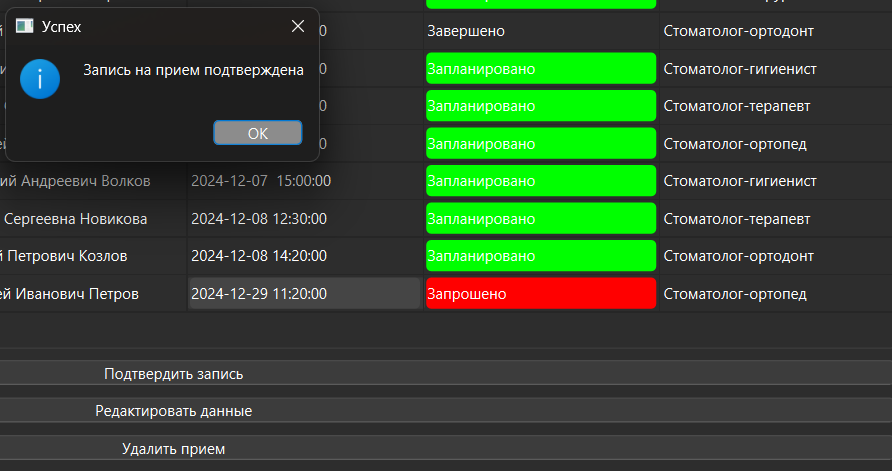


Рисунок 19 – Окно администратора, вкладка: «Приёмы». Данные о статусе приёма меняются после нажатия на кнопку «Подтвердить запись».

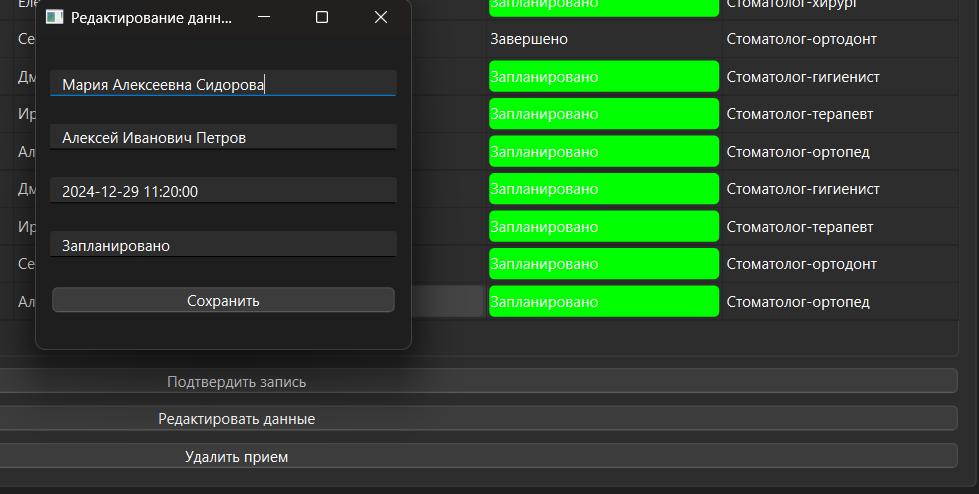


Рисунок 20– Окно администратора, вкладка: «Приёмы». Вывод окна после нажатия на кнопку «Подтвердить запись».

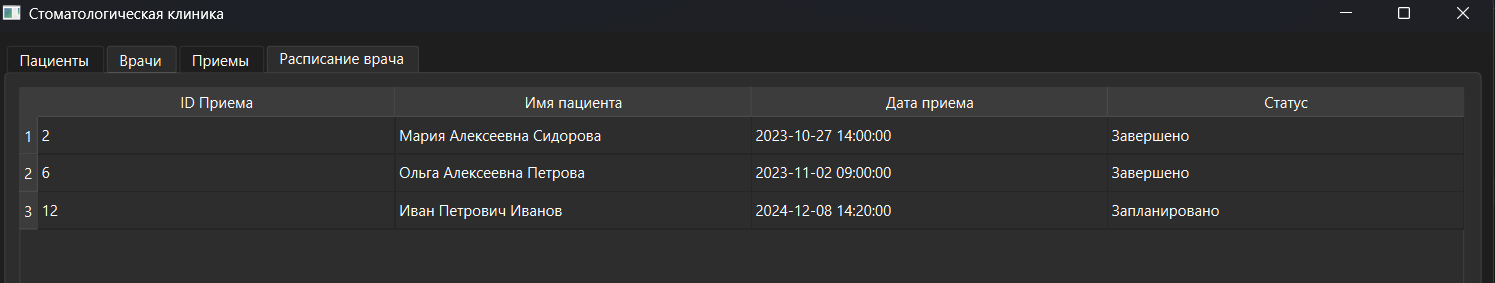


Рисунок 21 – Окно администратора. Во вкладке «Врачи» нажимаем на врача и после открывается вкладка «Расписание врача» с приёмами данного врача на которого нажали.



Рисунок 22 – Код для создания окна регистрации и функции регистрации.



Рисунок 23 – Продолжения кода для создания окна регистрации и функции регистрации.

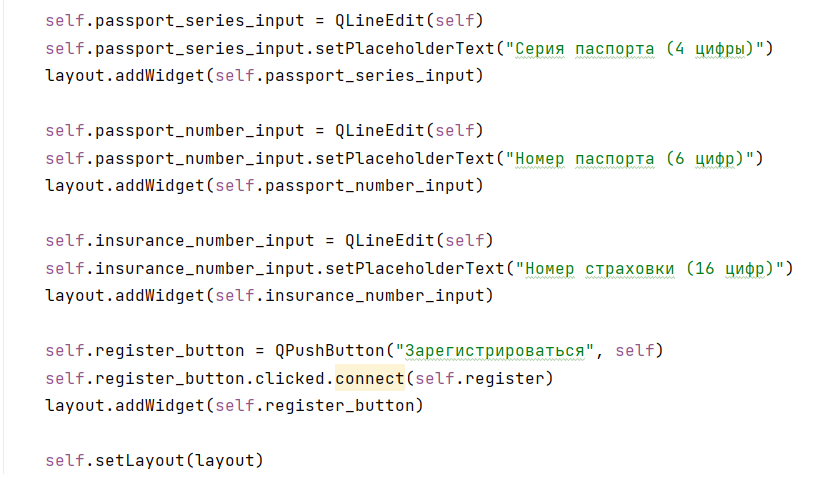


Рисунок 24 – Продолжения кода для создания окна регистрации и функции регистрации.

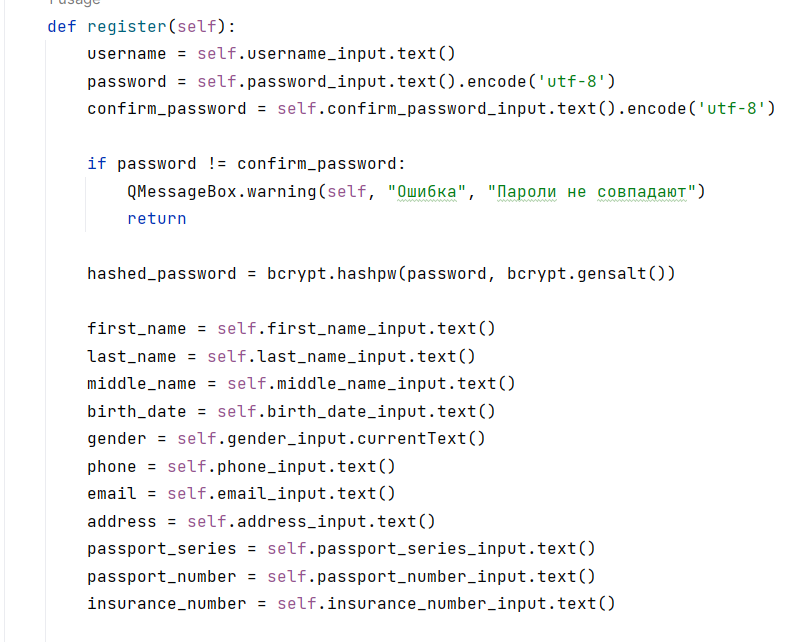


Рисунок 25 – Продолжения кода для создания окна регистрации и функции регистрации.

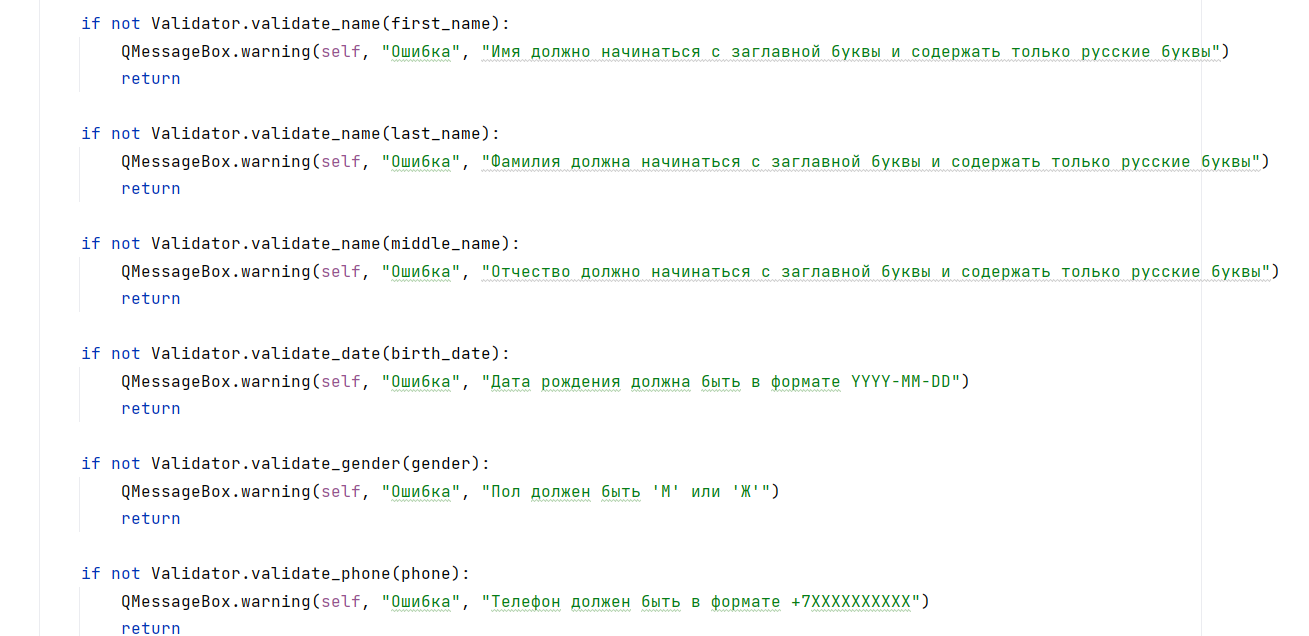


Рисунок 26 – Продолжения кода для создания окна регистрации и функции регистрации.

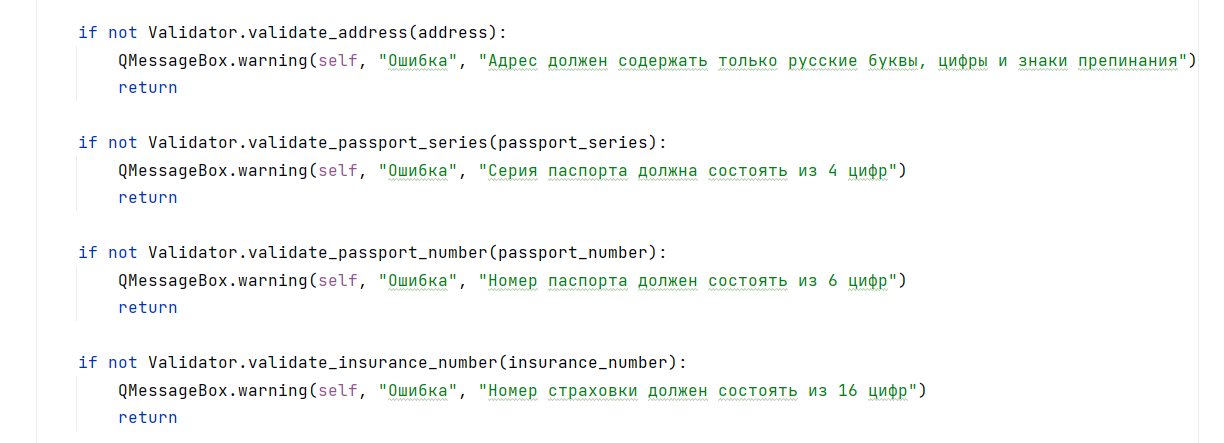


Рисунок 27 – Продолжения кода для создания окна регистрации и функции регистрации.

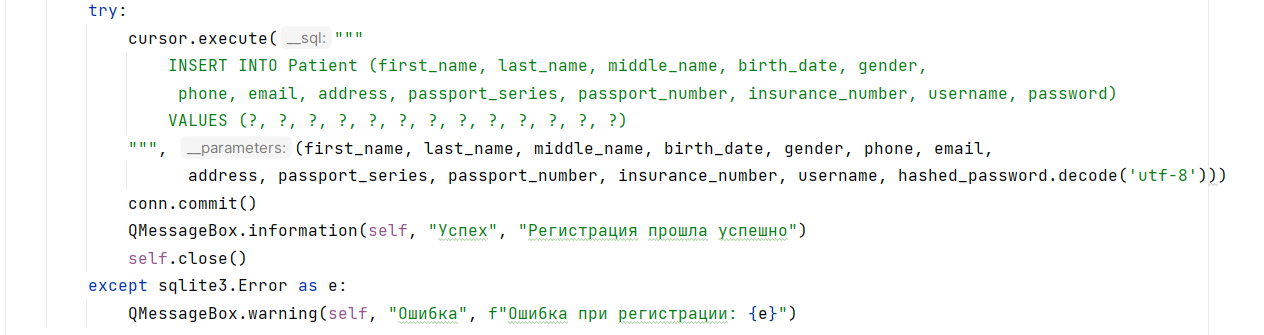


Рисунок 28 – Продолжения кода для создания окна регистрации и функции регистрации.



Рисунок 29– Код для создания окна авторизации и функция авторизации.



Рисунок 30 – Продолжения кода для создания окна авторизации и функция авторизации.

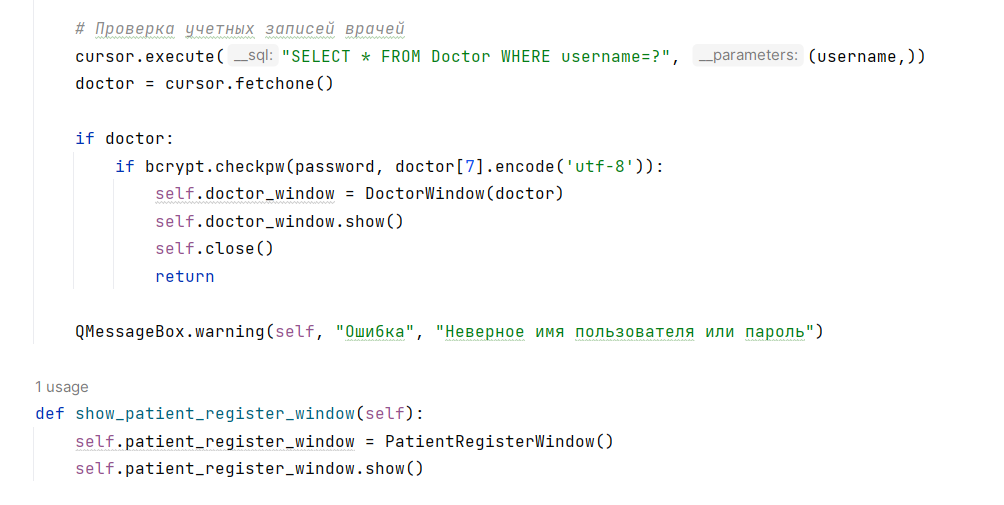


Рисунок 31 – Продолжения кода для создания окна авторизации и функция авторизации.

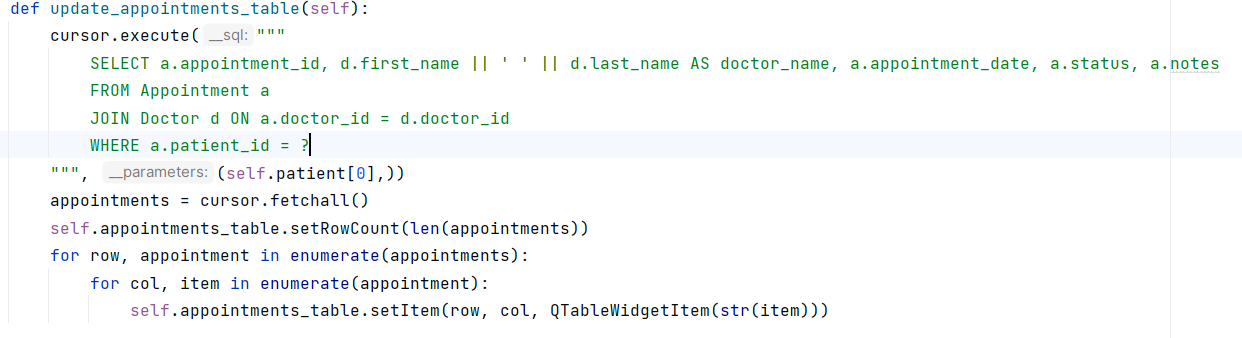


Рисунок 32 – Таблица с записями на прием.

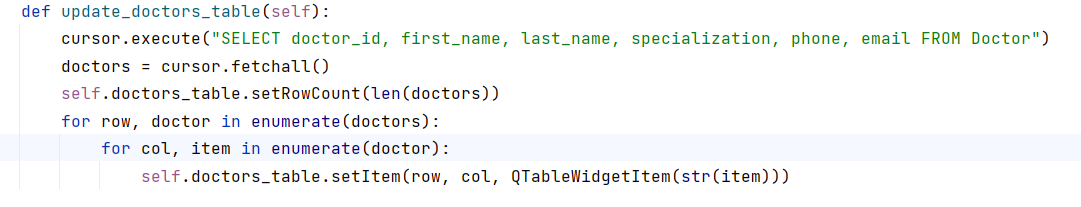


Рисунок 33 – Таблица с информацией о врачах.

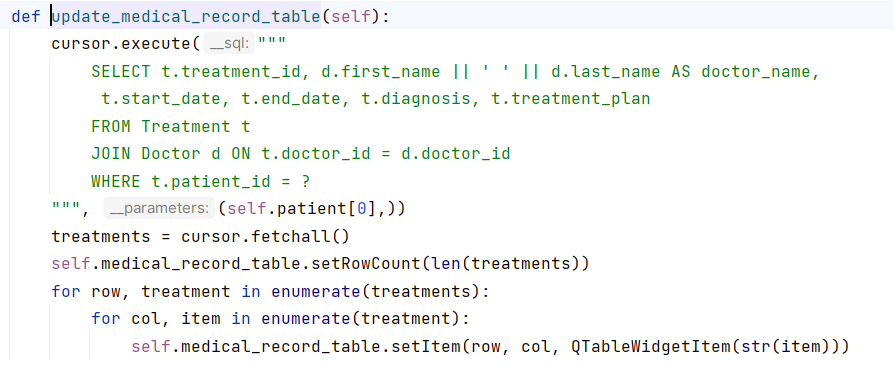


Рисунок 34 – Таблица со своей медицинской картой.

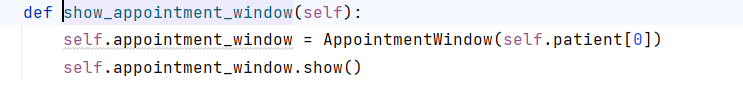


Рисунок 35 – Код для записи на приём

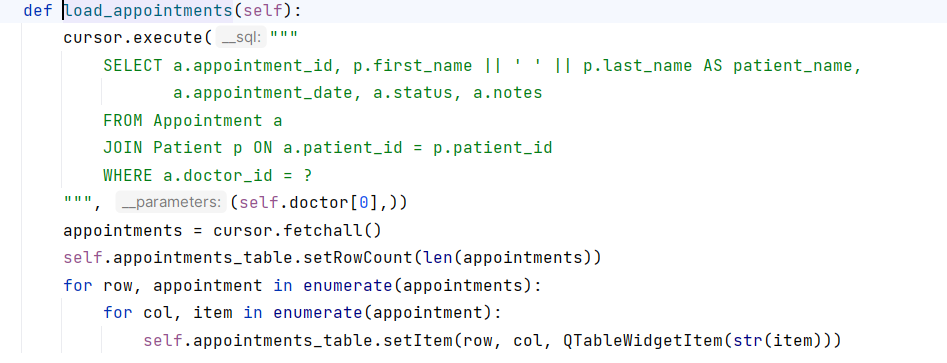


Рисунок 36 – Таблица с записями на приём

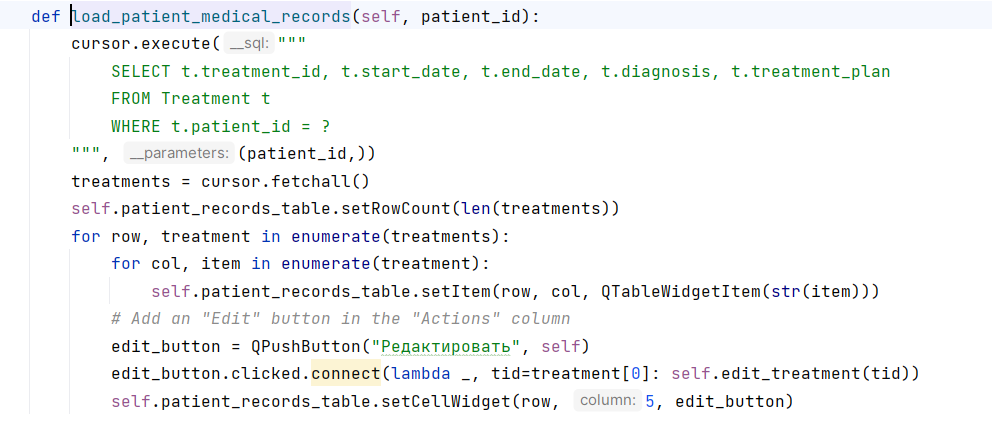


Рисунок 37– Таблица с медицинскими картами пациентов

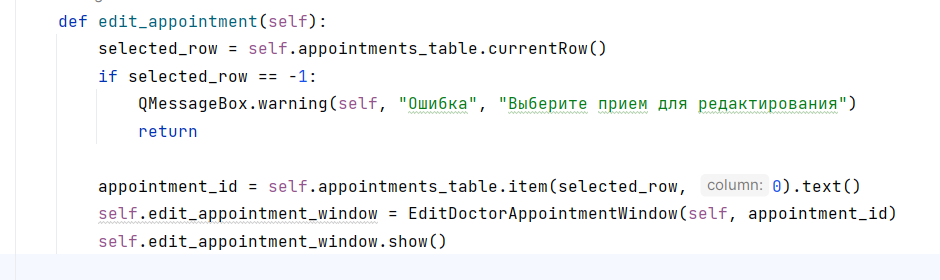


Рисунок 38 – Функция для редактирования записей на прием

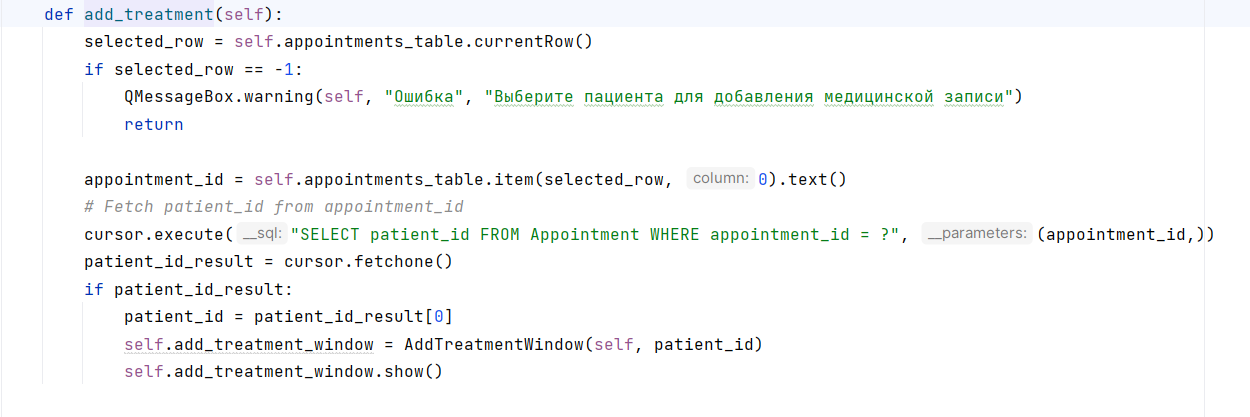


Рисунок 39 – Функция добавления медицинских карт

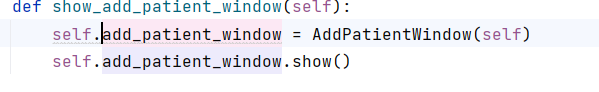


Рисунок 40 – Функция для добавления пациента и отображения виджетов

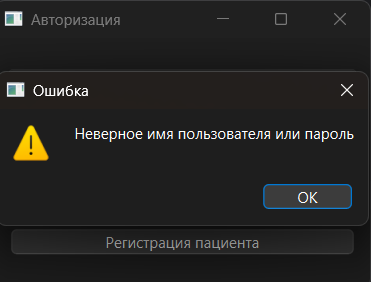


Рисунок 41 – Проверка входа с неправильными данными

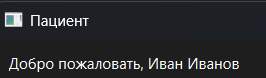


Рисунок 42 – Проверка входа с правильными данными

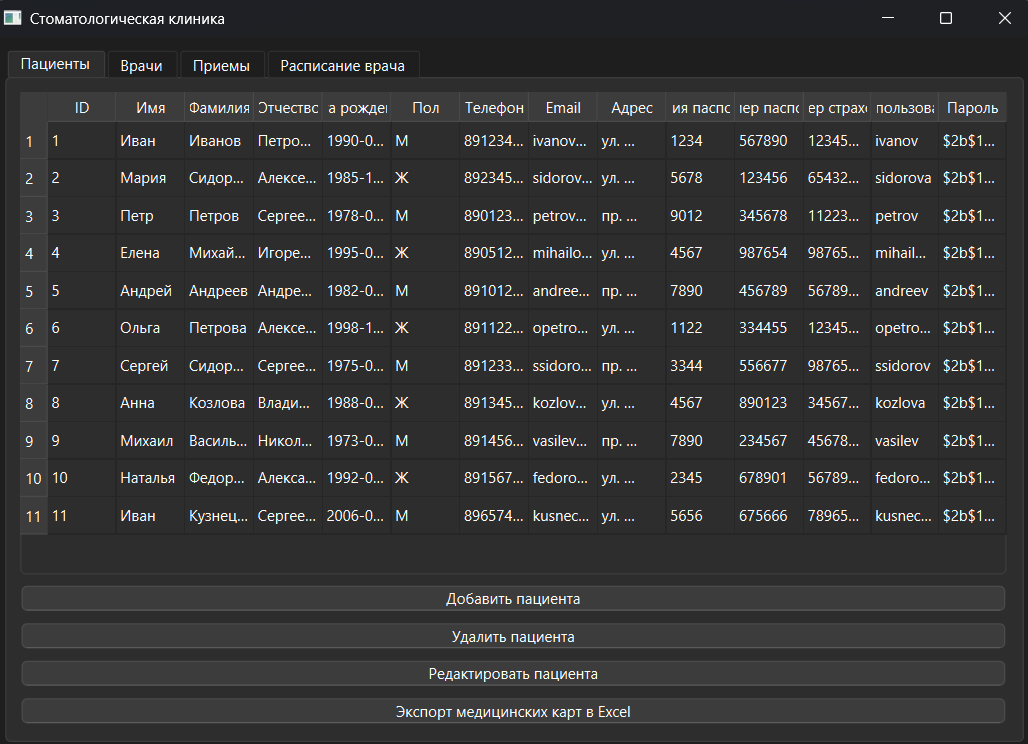


Рисунок 43 – Проверка отображения окна в зависимости от роли пользователя (администратор)

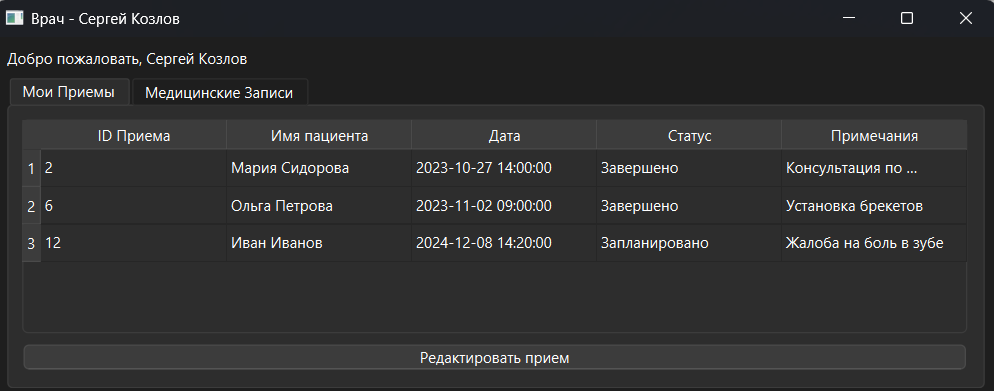


Рисунок 44 – Проверка отображения окна в зависимости от роли пользователя (врач)

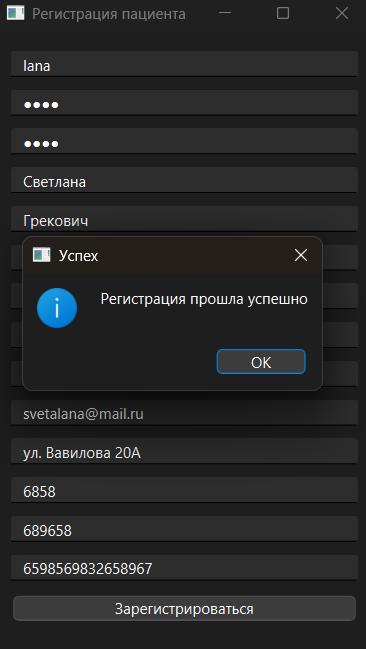


Рисунок 45 – Создание пациента в окне авторизации

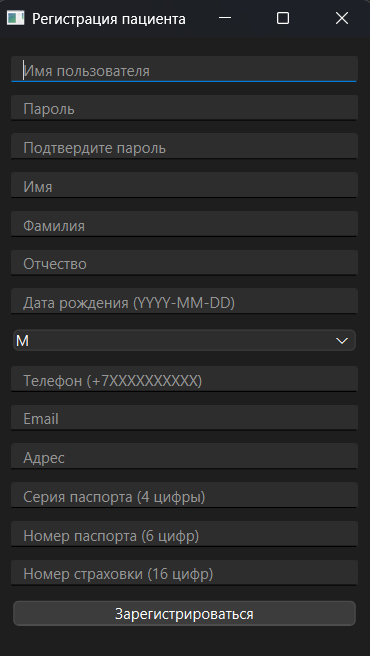


Рисунок 46 – Проверка добавления нового пациента через окно администратора.

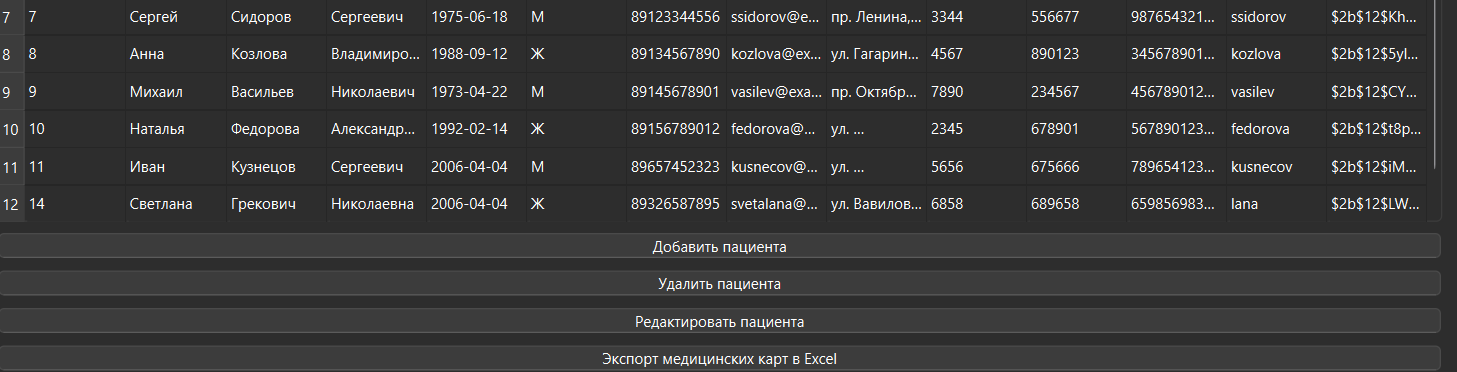


Рисунок 47– Проверка отображения нового пациента в списке пациентов. Пациент добавлен под ID 14!

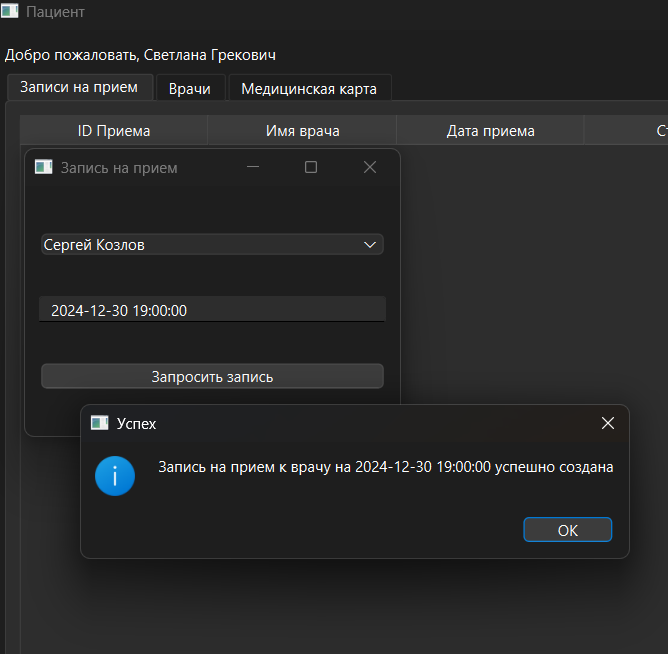


Рисунок 48– Новый пациент смог зайти к себе в профиль и запроситься на приём к врачу

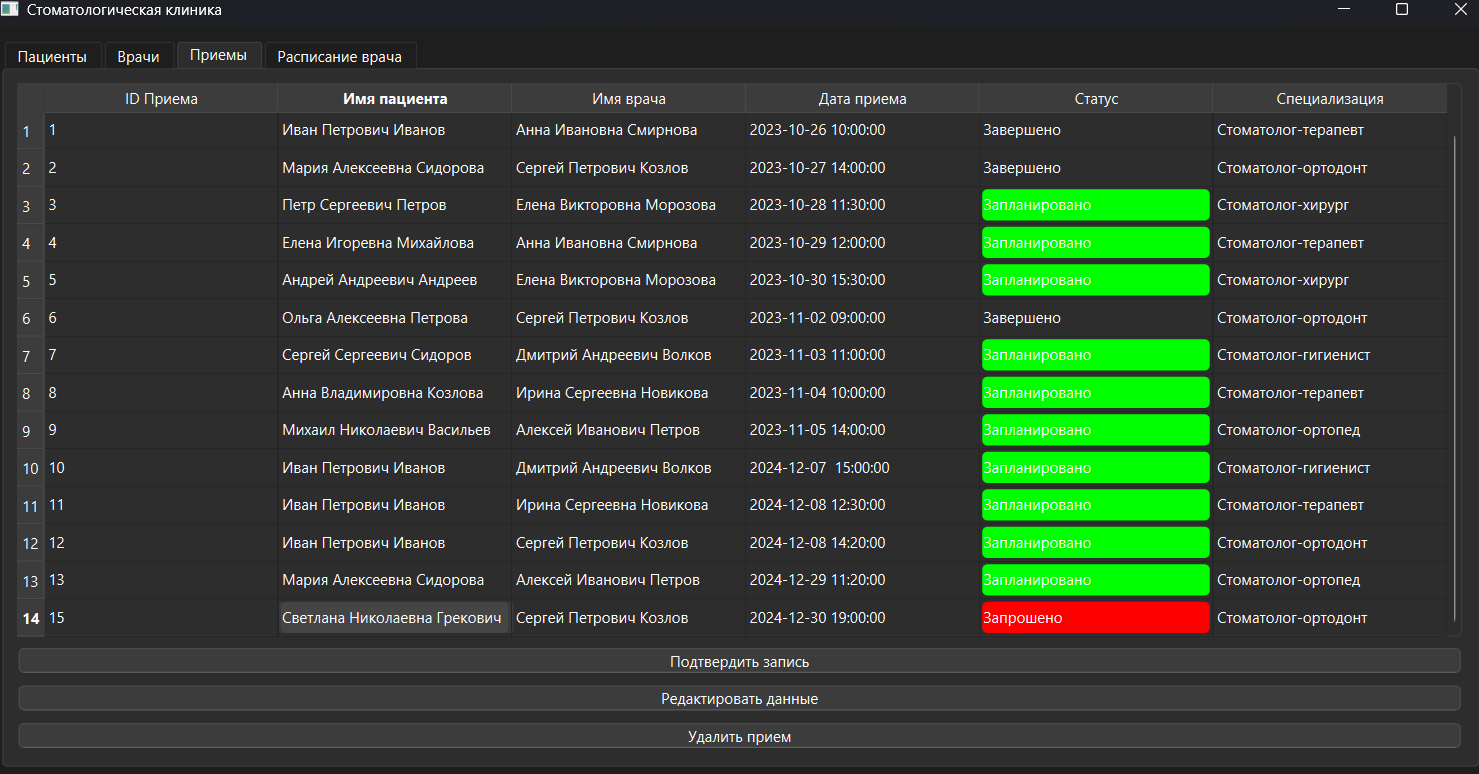


Рисунок 49–Проверка на отправление заявки на приём. Всё выполнилось, заявка ждет подтверждения от администратора!

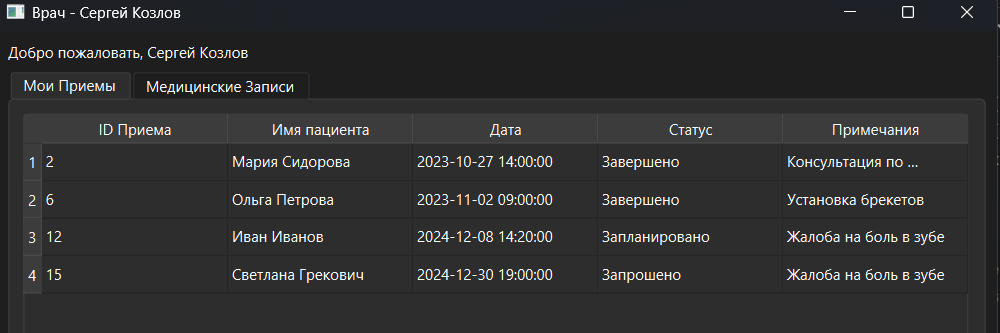


Рисунок 50– Проверка отображения нового пациента в списке пациентов врача. Добавился!

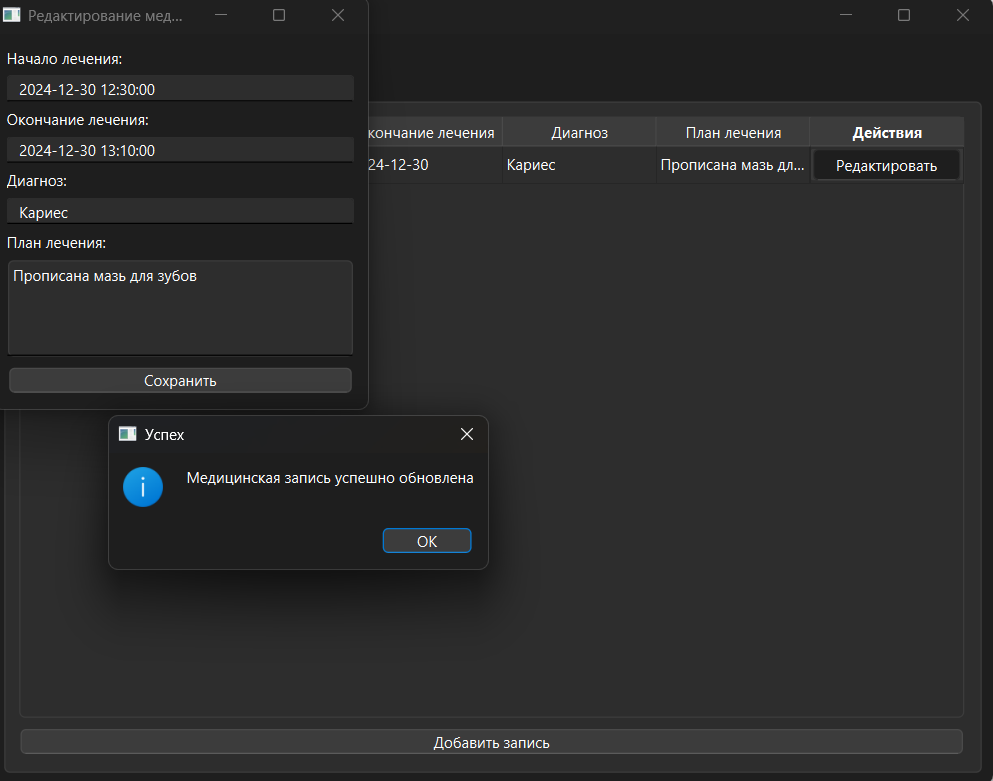
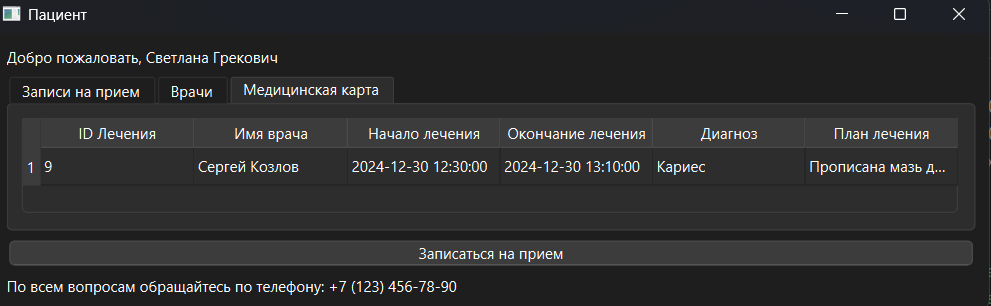


Рисунок 51– Проверка возможности врача добавлять и редактировать медицинские карты пациента

Рисунок 52– Окно пациента. Проверка отображения записей в медицинской карте