Министерство образования и науки Краснодарского края

Государственное автономное профессиональное

образовательное учреждение Краснодарского края

«Лабинский аграрный техникум»

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заместитель директора по  учебной работе    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.А. Мезенцева  «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г. |

**Дипломный проект**

по специальности 09.02.07 «Информационные системы и программирование»

Разработка информационной системы по учету работы отдела АСУ в ГБУЗ "Лабинская центральная районная больница"

Руководитель\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.Д. Петров

подпись дата должность, ученая степень

Выпускник \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_ А. Ю. Федоренко

подпись дата

|  |  |
| --- | --- |
| ДОПУЩЕН К ЗАЩИТЕ  Председатель УМО преподавателей компьютерных специальностей  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.П. Ефентьева  «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. | НОРМОКОНТРОЛЬ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_С.С. Сурков  «\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г. |

Дата защиты «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г. Оценка \_\_\_\_ «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

Протокол ГЭК № \_\_\_\_\_\_\_\_ от «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc407240252)

[Глава 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА ТРЕБОВАНИЙ 6](#_Toc604626703)

[1.1 Общая характеристика ГБУЗ «Лабинская центральная районная больница» 6](#_Toc853309712)

[1.2 Описание функциональной структуры отдела АСУ 7](#_Toc1005478255)

[1.3 Анализ проблемных ситуаций и информационных потоков 9](#_Toc1855109112)

[1.4 Постановка задачи автоматизации учета работы отдела АСУ 11](#_Toc1443566339)

[1.5 Формулирование требований к информационной системе 12](#_Toc1609166619)

[Вывод по главе 14](#_Toc997322860)

[Глава 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ 16](#_Toc1765196641)

[2.1 Обоснование выбора методов и технологий разработки 16](#_Toc87142904)

[2.2 Проектирование функциональной структуры системы 18](#_Toc897088657)

[2.3 Разработка концептуальной и логической моделей базы данных 21](#_Toc2004774841)

[2.4 Выбор программного и технического обеспечения 26](#_Toc1382451027)

[Вывод по главе 30](#_Toc1138741129)

**ВВЕДЕНИЕ**

В современной цифровой экономике эффективное управление информационными ресурсами становится ключевым фактором успешной деятельности любой организации, особенно в такой социально значимой сфере как здравоохранение. ГБУЗ "Лабинская центральная районная больница", являясь крупным медицинским учреждением Краснодарского края, ежедневно сталкивается с необходимостью обработки значительных объемов данных, что требует надежной и эффективной работы отдела автоматизированных систем управления.

В настоящее время процессы учета рабочего времени сотрудников отдела АСУ, планирования задач и контроля их выполнения осуществляются преимущественно вручную с использованием электронных таблиц и бумажных носителей. Такой подход приводит к ряду существенных проблем:

* Значительным временным затратам на составление отчетности;
* Высокой вероятности ошибок при ручном вводе данных;
* Отсутствию единой системы учета выполненных работ;
* Сложностям в анализе эффективности работы подразделения;
* Затруднениям при планировании ресурсов.

Разработка специализированной информационной системы для автоматизации этих процессов позволит не только устранить указанные недостатки, но и обеспечит:

* Оперативный доступ к актуальной информации о состоянии работ;
* Автоматизацию процессов документооборота;
* Возможность анализа показателей эффективности;
* Соблюдение требований нормативных документов в сфере здравоохранения.

Цель исследования - разработка и внедрение информационной системы автоматизированного учета работы отдела АСУ в ГБУЗ "Лабинская центральная районная больница", обеспечивающей повышение эффективности управления ресурсами подразделения.

Задачи исследования:

1. Провести комплексный анализ организационной структуры и бизнес-процессов отдела АСУ;
2. Определить функциональные и нефункциональные требования к системе;
3. Разработать концептуальную, логическую и физическую модели базы данных;
4. Выбрать оптимальные технологии реализации системы;
5. Реализовать программный продукт с учетом требований информационной безопасности;
6. Провести тестирование системы на соответствие требованиям;
7. Разработать комплект эксплуатационной документации.

Объект исследования – процессы управления и учета деятельности отдела автоматизированных систем управления в медицинском учреждении.

Предмет исследования – методы и средства автоматизации учета рабочего времени, планирования задач и контроля выполнения работ в отделе АСУ медицинского учреждения.

Методы исследования:

* Системный анализ предметной области;
* Методы проектирования информационных систем;
* Современные технологии разработки программного обеспечения;
* Методы тестирования и верификации ПО;
* Методы оценки эффективности внедрения.

Реализация проекта позволит повысить эффективность управления отделом АСУ, сократить временные затраты на составление отчетности и уменьшить количество ошибок в учетных данных. Кроме того, он обеспечит прозрачность процессов планирования и контроля, а также повысит качество принимаемых управленческих решений.

**Глава 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И РАЗРАБОТКА ТРЕБОВАНИЙ**

1. **Общая характеристика ГБУЗ «Лабинская центральная районная больница»**

Государственное бюджетное учреждение здравоохранения «Лабинская центральная районная больница» (ГБУЗ «Лабинская ЦРБ» МЗ КК) является ключевым медицинским учреждением Лабинского района Краснодарского края. Больница была основана в 1989 году в результате реорганизации сети учреждений здравоохранения города Лабинска и района. С момента создания учреждение прошло несколько этапов преобразований, включая изменение организационно-правовой формы и наименования, и в 2018 году было переведено в государственную собственность Краснодарского края.

ГБУЗ «Лабинская ЦРБ» располагается по адресу: 352501, Краснодарский край, г. Лабинск, ул. Пирогова, д. 1. Учреждение является некоммерческой организацией, находящейся в ведении министерства здравоохранения Краснодарского края. Основной целью деятельности больницы является оказание квалифицированной медицинской помощи населению в соответствии с территориальной программой государственных гарантий.

Основные направления деятельности:

* Оказание стационарной и амбулаторной медицинской помощи;
* Проведение диагностических и лечебных процедур;
* Фармацевтическая деятельность;
* Использование источников ионизирующего излучения для медицинских целей.

Структура учреждения включает в себя отделения различного профиля, включая терапевтическое, хирургическое, педиатрическое, а также специализированные службы, такие как отделение анестезиологии и реанимации, клинико-диагностическая лаборатория и другие. Важным структурным подразделением является отдел автоматизированных систем управления (АСУ), который отвечает за внедрение, сопровождение и развитие информационных технологий в учреждении.

Техническое оснащение:

Больница оснащена современным медицинским оборудованием и информационными системами, которые позволяют эффективно осуществлять учет пациентов, управлять медицинскими данными и оптимизировать административные процессы. Однако, как показал анализ, существующие системы учета работы отдела АСУ требуют модернизации для повышения прозрачности, скорости обработки данных и минимизации ручного труда.

Финансирование и управление:

Деятельность учреждения финансируется за счет средств краевого бюджета и обязательного медицинского страхования. Управление осуществляется главным врачом, который назначается министерством здравоохранения Краснодарского края.

Таким образом, ГБУЗ «Лабинская ЦРБ» представляет собой многопрофильное медицинское учреждение, деятельность которого требует эффективного управления информационными ресурсами, что делает актуальной разработку специализированной информационной системы для отдела АСУ.

1. **Описание функциональной структуры отдела АСУ**

Отдел автоматизированных систем управления играет важную роль в обеспечении эффективной работы медицинского учреждения, отвечая за внедрение, сопровождение и постоянное совершенствование информационных технологий. Основной целью деятельности отдела является автоматизация ключевых процессов больницы, включая медицинские услуги, административное управление и документооборот.

В сфере разработки и внедрения ИТ-решений специалисты отдела занимаются созданием и адаптацией программного обеспечения, предназначенного для учета пациентов, управления медицинскими услугами, кадрового делопроизводства и финансовой отчетности. Важной частью работы является интеграция внутренних систем с внешними платформами, такими как ЕГИСЗ и системы электронного документооборота, что позволяет обеспечить согласованность данных и упростить обмен информацией.

Техническая поддержка и обслуживание включают в себя обеспечение бесперебойной работы серверного оборудования, рабочих станций и периферийных устройств. Сотрудники отдела настраивают и администрируют локальную сеть, следят за информационной безопасностью и оперативно устраняют возникающие неполадки в программном обеспечении и технике.

Управление базами данных является еще одним важным направлением. Отдел занимается разработкой, оптимизацией и сопровождением баз данных медицинской информационной системы, а также обеспечивает регулярное резервное копирование и восстановление информации для предотвращения потери данных.

Автоматизация документооборота позволяет ускорить обработку внутренней и внешней корреспонденции за счет внедрения специализированных систем электронного документооборота. Дополнительно отдел разрабатывает отчетные формы для контролирующих органов, что упрощает процесс предоставления необходимой документации.

Важной составляющей работы отдела является обучение и консультации персонала. Сотрудники проводят инструктажи по работе с новыми программными продуктами, а также разрабатывают методические материалы и инструкции для удобства пользователей.

Несмотря на активное развитие информационных технологий, в работе отдела сохраняются некоторые сложности. Одной из основных проблем является ручной учет задач, что приводит к дублированию информации и потере времени. Отсутствие единой системы учета работ снижает эффективность управления процессами.

Еще одной трудностью является децентрализованное хранение документов. Часть отчетности до сих пор ведется в Excel, что усложняет контроль, анализ и поиск необходимых данных.

Кроме того, многие процессы остаются недостаточно автоматизированными, что вынуждает сотрудников вручную вводить значительные объемы информации. Это увеличивает вероятность ошибок и замедляет выполнение задач.

Для повышения эффективности работы отдела необходимо дальнейшее внедрение современных ИТ-решений, которые позволят оптимизировать учет задач, централизовать хранение документов и сократить долю ручного труда в повседневных операциях.

1. **Анализ проблемных ситуаций и информационных потоков**

Проведенное исследование деятельности отдела автоматизированных систем управления выявило ряд системных проблем, существенно снижающих эффективность работы подразделения. Сложившаяся ситуация требует детального рассмотрения и скорейшего принятия мер по оптимизации ключевых процессов.

Основные проблемные области:

* В сфере учета выполняемых работ наблюдается серьезный организационный пробел. Отсутствие единой платформы для фиксации заявок и контроля выполнения задач приводит к дублированию информации в разрозненных системах - электронных таблицах Excel и бумажных журналах. Такая практика не только увеличивает временные затраты сотрудников, но и создает сложности при оценке реальной загрузки специалистов и планировании работ;
* Особую озабоченность вызывает состояние документооборота в отделе. Несмотря на частичную автоматизацию некоторых процессов, система согласования документов остается несовершенной. Сотрудники регулярно сталкиваются с проблемами поиска актуальных версий документов, а отсутствие четкого механизма контроля исполнения поручений приводит к сбоям в рабочих процессах;
* Серьезным риском для организации является децентрализованный подход к хранению данных. Критически важная информация распределена по локальным компьютерам сотрудников без создания единого защищенного репозитория. Такая практика не только затрудняет доступ к необходимым данным, но и создает реальную угрозу их потери при технических сбоях;
* Процесс формирования отчетности также требует существенной доработки. Необходимость ручного сбора и консолидации данных из различных источников приводит к значительным временным затратам и повышает вероятность ошибок при подготовке регулярных отчетов.

В работе отдела можно выделить три основных направления информационных потоков. Входящие потоки включают запросы на техническую поддержку, требования по доработке систем, нормативные документы и данные оборудования. Внутренние процессы охватывают распределение задач, обмен технической документацией, мониторинг выполнения работ и учет возникающих инцидентов. Исходящие потоки представляют отчетную документацию, технические инструкции, аналитические материалы и предложения по развитию ИТ-инфраструктуры.

Основные проблемы в организации информационных потоков связаны с преобладанием неформальных каналов коммуникации, отсутствием единых регламентов работы с данными, необходимостью многократного ввода одинаковой информации в разные системы и задержками при передаче сведений между подразделениями.

Выявленные проблемы имеют комплексный негативный эффект. На организационном уровне это проявляется в низкой прозрачности процессов, сложностях управления нагрузкой персонала и задержках выполнения важных задач. Экономические последствия включают потери рабочего времени, необходимость содержания избыточного штата и финансовые риски из-за ошибок учета. Технологические ограничения выражаются в трудностях получения актуальной информации, масштабирования решений и интеграции новых систем.

Проведенный анализ однозначно свидетельствует о необходимости внедрения специализированной информационной системы. Такое решение позволит формализовать ключевые бизнес-процессы, централизовать управление информационными потоками, повысить эффективность работы сотрудников и улучшить качество управленческих решений. Реализация этого проекта должна стать приоритетным направлением в развитии ИТ-инфраструктуры учреждения.

1. **Постановка задачи автоматизации учета работы отдела АСУ**

На основании анализа существующих проблем и информационных потоков отдела АСУ ГБУЗ "Лабинская ЦРБ" сформулирована задача разработки специализированной информационной системы для автоматизации учета работы подразделения.

Цели автоматизации:

1. Создание единого информационного пространства для управления задачами отдела АСУ;
2. Оптимизация процессов обработки заявок и документооборота;
3. Повышение прозрачности и контролируемости рабочих процессов;
4. Снижение временных затрат на рутинные операции на 30-40%.

Функциональные требования к автоматизированной системе управления:

1. Модуль обработки заявок:

Система должна обеспечивать комплексное управление заявками, включая их регистрацию с обязательной классификацией по приоритетам и статусам выполнения. Для каждой заявки необходимо предусмотреть назначение ответственного исполнителя, контроль сроков выполнения (от момента создания до закрытия), а также возможность ведения обсуждений через систему комментариев.

1. Модуль управления пользователями

Требуется реализовать гибкую систему разграничения прав доступа на основе ролевой модели. Система должна позволять управлять учетными записями сотрудников и их компетенциями, обеспечивая дифференцированный доступ к функционалу в зависимости от должностных обязанностей.

1. Отчетный модуль

Автоматизированный механизм формирования отчетности должен включать генерацию отчетов с возможностью их экспорта в форматы офисных документов и вывода на печать.

В основе системы должны лежать следующие основные сущности:

* Пользователи и их роли;
* Заявки с атрибутами (статусы, приоритеты);
* Организационная структура подразделений;
* Механизмы комментирования и отчетности.

1. **Формулирование требований к информационной системе**

На основании проведенного анализа предметной области и поставленных задач автоматизации сформулирован комплекс требований к разрабатываемой системе:

1. Функциональные требования:

Система должна предоставлять инструменты для управления заявками и задачами, включая регистрацию новых заявок с обязательным указанием названия, описания, статуса и приоритета. Должна быть реализована возможность назначения исполнителей, контроля сроков выполнения, а также изменения статусов заявок (например, "в работе", "выполнено" и другие). Для удобства пользователей система должна поддерживать поиск заявок и ведение истории обсуждений через комментарии.

Управление пользователями и правами доступа предполагает разграничение ролей: администратор, сотрудник отдела АСУ и обычный пользователь. Каждый зарегистрированный пользователь получает личный профиль, в котором отображается история связанных с ним заявок.

Для анализа работы отдела система должна автоматически формировать отчеты и предоставлять возможность их экспорта в формат docx.

1. Технические требования:

Система должна обеспечивать стабильную работу при одновременном использовании 50 и более пользователей. Время отклика на действия пользователей не должно превышать 2 секунд, а ежедневная нагрузка — до 100 новых заявок. В целях безопасности обязательна процедура аутентификации и авторизации пользователей.

1. Требования к интерфейсу:

Интерфейс системы должен быть интуитивно понятным и полностью локализованным на русский язык. Основные разделы включают:

* Список заявок с возможностью фильтрации и сортировки;
* Форму создания и редактирования заявок;
* Личный кабинет пользователя с историей активности;
* Раздел отчетности и аналитики;
* Административную панель для управления системой.

1. Требования к надежности:

Система должна быть отказоустойчивой: в случае сбоев необходимо автоматическое сохранение данных. Также требуется обеспечить защиту конфиденциальной информации за счет разграничения прав доступа.

1. Требования к сопровождению:

Для удобства администрирования система должна включать панель управления с возможностью настройки прав доступа. Дополнительно необходимо разработать руководство пользователя, содержащее описание функциональных возможностей и инструкции по работе с системой.

**Вывод по главе**

Проведенный анализ предметной области и деятельности отдела автоматизированных систем управления (АСУ) ГБУЗ «Лабинская ЦРБ» позволил выявить ключевые проблемы, препятствующие эффективной работе подразделения. Основные сложности связаны с отсутствием единой системы учета задач, децентрализованным хранением данных, ручным вводом информации и недостаточной автоматизацией документооборота. Эти факторы приводят к снижению прозрачности процессов, увеличению временных затрат и повышению риска ошибок.

На основании проведенного исследования сформулированы цели автоматизации, включая создание единого информационного пространства, оптимизацию обработки заявок, повышение контроля за рабочими процессами и сокращение рутинных операций. Определены функциональные, технические и интерфейсные требования к разрабатываемой системе, а также требования к надежности и сопровождению.

Реализация предложенной информационной системы позволит:

* Централизовать управление задачами и заявками;
* Минимизировать ручной ввод данных;
* Повысить скорость обработки информации;
* Улучшить контроль исполнения задач;
* Обеспечить безопасное хранение и разграничение доступа к данным.

Таким образом, внедрение автоматизированной системы учета работы отдела АСУ является необходимым шагом для повышения эффективности управления ИТ-процессами в учреждении. Разработанные требования послужат основой для проектирования и реализации системы, которая оптимизирует рабочие процессы и повысит качество обслуживания пользователей.

**Глава 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

1. **Выбор и обоснование технологий и программного обеспечения**

При разработке информационной системы учета работы отдела АСУ для медицинского учреждения ключевое значение имеет правильный выбор технологического стека. Этот выбор определяет не только текущую функциональность системы, но и возможности ее дальнейшего развития, простоту поддержки и надежность работы. В условиях медицинского учреждения особенно важны стабильность работы, безопасность данных и простота использования, так как система будет эксплуатироваться персоналом без глубоких технических знаний.

Технологический стек должен обеспечивать выполнение всех функциональных требований, предъявляемых к системе, включая управление заявками, формирование отчетов и разграничение прав доступа. При этом важно учитывать существующую ИТ-инфраструктуру учреждения, которая в большинстве медицинских организаций построена на решениях Microsoft. Дополнительными факторами выбора являются доступность специалистов для поддержки системы и соответствие бюджетным ограничениям проекта.

Основой для клиентской части системы был выбран язык программирования C# в сочетании с Windows Forms. C# является современным объектно-ориентированным языком с сильной типизацией, что способствует созданию надежного и поддерживаемого кода. Широкая распространенность этого языка обеспечивает доступность квалифицированных разработчиков для поддержки и развития системы. Интеграция с платформой .NET Framework предоставляет богатый набор библиотек для реализации различных функций системы.

Windows Forms была выбрана в качестве технологии для пользовательского интерфейса благодаря своей простоте разработки и естественной интеграции с операционной системой Windows. Это особенно важно в условиях медицинского учреждения, где персонал привык работать с Windows-приложениями. Windows Forms позволяет создавать интуитивно понятные интерфейсы с минимальным временем освоения для конечных пользователей.

В качестве основной среды разработки выбрана Microsoft Visual Studio Community 2022 (рис. 2.3) - мощная интегрированная среда разработки, предоставляющая все необходимые инструменты для создания Windows-приложений. Ее преимущества для данного проекта включают:

* Полноценную поддержку разработки на C# и Windows Forms
* Встроенные инструменты для работы с базами данных
* Интегрированную систему контроля версий Git
* Богатый набор средств отладки и профилирования
* Возможность использования расширений через NuGet
* Бесплатную лицензию для образовательных и некоммерческих проектов

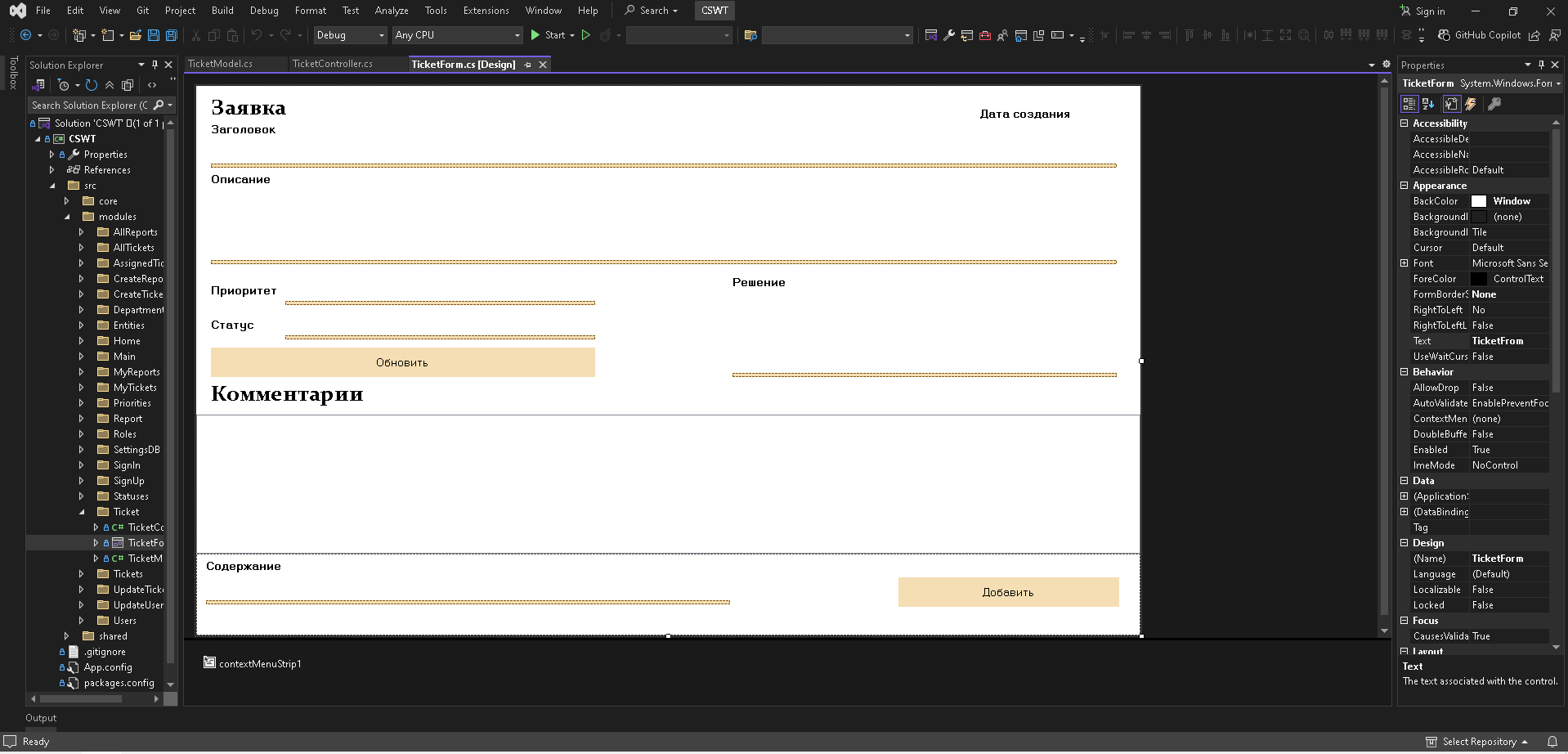


Рис. 2.1 - Интерфейс Visual Studio

Для хранения данных была выбрана система управления базами данных PostgreSQL 15. PostgreSQL — это мощная реляционная база данных с открытым исходным кодом, которая обладает расширенным набором функций для обработки и хранения информации. Эта СУБД сочетает в себе надежность коммерческих решений с открытостью исходного кода. PostgreSQL полностью поддерживает ACID-транзакции, что критически важно для системы учета заявок, где необходимо гарантировать целостность данных. Встроенные механизмы репликации обеспечивают отказоустойчивость системы, а поддержка JSON позволяет гибко хранить структурированные данные отчетов. Для работы с СУБД PostgreSQL используется pgAdmin4 инструмент с графическим интерфейсом для управления базами данных PostgreSQL (рис. 2.2).

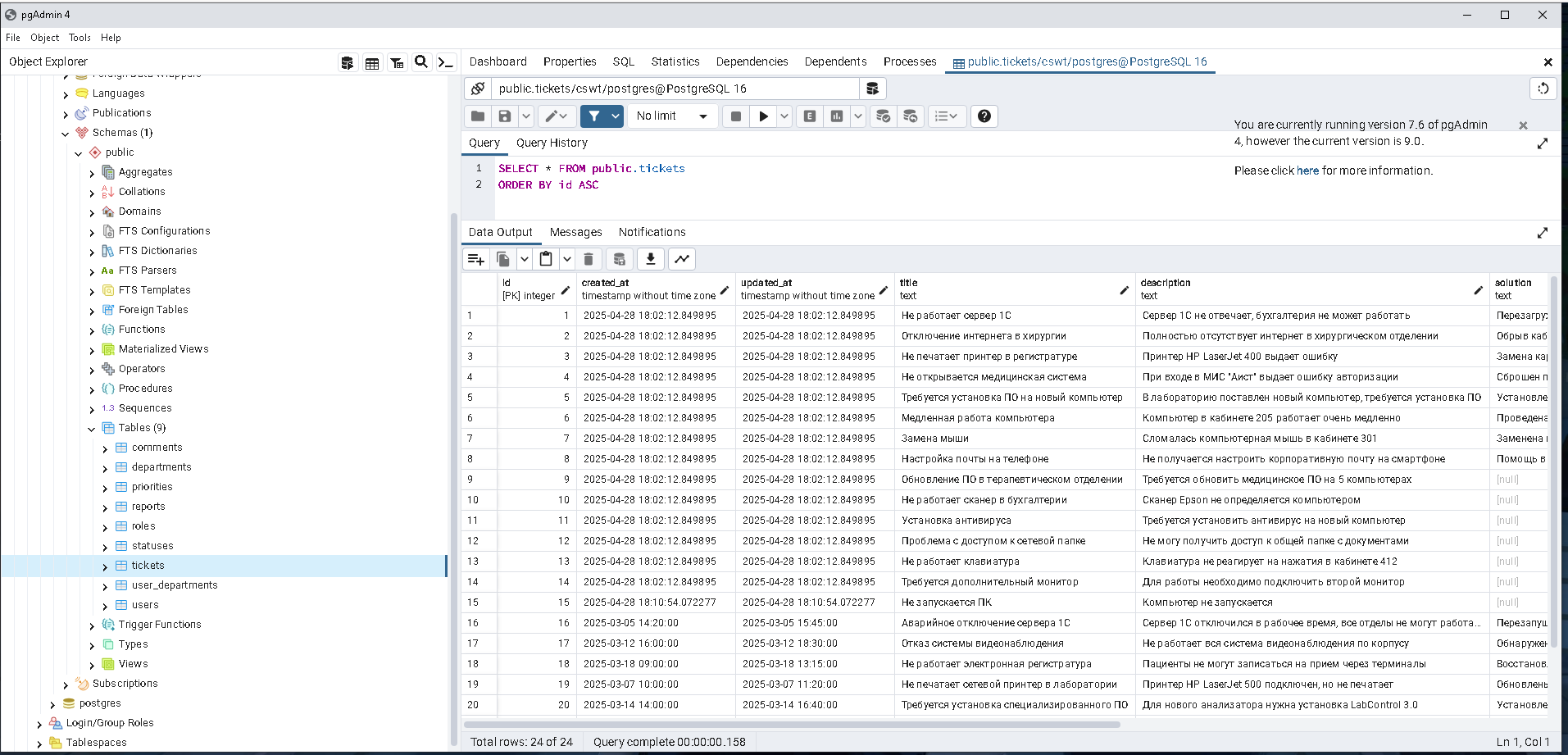


Рис. 2.2 - Интерфейс pgAdmin4

Взаимодействие между клиентским приложением и базой данных осуществляется через драйвер Npgsql, который является стандартным средством работы с PostgreSQL из .NET-приложений. Этот драйвер обеспечивает высокую производительность и поддерживает все возможности PostgreSQL, включая асинхронные операции, что важно для обеспечения отзывчивости интерфейса при работе с большими объемами данных.

Система построена по классической трехуровневой архитектуре, что позволяет четко разделить ответственность между компонентами (рис. 2.3). Уровень представления реализован на Windows Forms и отвечает за взаимодействие с пользователем. Бизнес-логика системы вынесена в отдельный слой, что упрощает ее тестирование и модификацию. Доступ к данным осуществляется через слой репозиториев, который абстрагирует работу с базой данных.

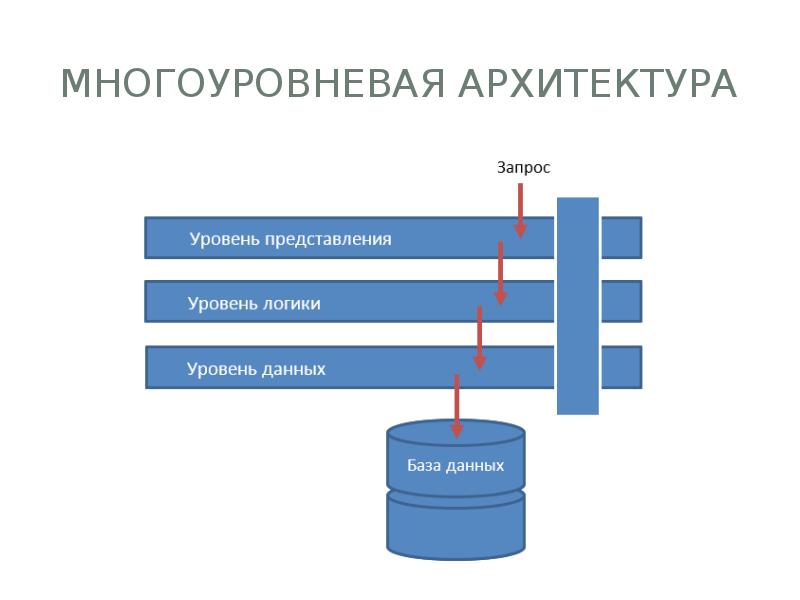


Рис. 2.3 - Многоуровневая архитектура

Такой подход обеспечивает гибкость архитектуры и позволяет в будущем модифицировать отдельные компоненты системы без необходимости ее полной переработки. Например, при необходимости добавления веб-интерфейса можно будет использовать существующий слой бизнес-логики, заменив только уровень представления.

Серверная часть системы развертывается на выделенном оборудовании с учетом требований к производительности и надежности. Минимальная конфигурация сервера включает многоядерный процессор, 8 ГБ оперативной памяти и SSD-накопитель для системы. Данные рекомендуется хранить на отдельном жестком диске с регулярным резервным копированием.

В качестве операционной системы сервера может использоваться как Windows Server, так и Linux, так как PostgreSQL поддерживает обе платформы. Для медицинских учреждений, где преобладает Windows-инфраструктура, часто предпочтительнее первый вариант из-за простоты администрирования.

Клиентские рабочие станции должны соответствовать современным требованиям для комфортной работы с приложением. Особое внимание уделяется стабильности сетевого соединения между клиентами и сервером базы данных, так как это напрямую влияет на скорость работы системы.

Выбранный технологический стек позволяет легко расширять функциональность системы в будущем. Одним из направлений развития может стать добавление веб-интерфейса на базе Blazor, что позволит предоставить доступ к системе через браузер. Другим перспективным направлением является разработка мобильного приложения для оперативного управления заявками с использованием Xamarin или .NET MAUI.

Выбранные технологии обеспечивают баланс между надежностью, производительностью и стоимостью владения, что особенно важно для бюджетных медицинских учреждений. Гибкость архитектуры позволяет адаптировать систему к изменяющимся требованиям без необходимости полной замены технологического стека.

1. **Проектирование функциональной структуры системы**

На основании проведенного анализа требований к системе учета работы отдела АСУ была разработана комплексная функциональная архитектура, построенная по трехуровневому принципу. Такая архитектура обеспечивает четкое разделение ответственности между компонентами системы и упрощает ее дальнейшее развитие и поддержку.

Уровень представления реализован в виде Windows Forms приложения, которое обеспечивает взаимодействие с конечными пользователями системы. Этот уровень отвечает за отображение информации, сбор пользовательского ввода и обеспечение удобного интерфейса для работы с системой. Выбор Windows Forms обусловлен необходимостью создания привычного для медицинского персонала интерфейса с минимальным временем освоения.

Уровень бизнес-логики разработан на языке C и содержит все основные алгоритмы обработки данных. Этот уровень включает классы, отвечающие за обработку заявок, управление пользователями, генерацию отчетов и другие ключевые функции системы. Бизнес-логика полностью отделена от уровня представления, что позволяет изменять интерфейсные решения без переработки основной функциональности.

Уровень данных построен на базе PostgreSQL 15 - современной реляционной СУБД с открытым исходным кодом. Этот уровень отвечает за надежное хранение и эффективный поиск информации. PostgreSQL была выбрана благодаря ее стабильности, производительности и соответствию требованиям проекта по обработке транзакций.

Система включает четыре ключевых функциональных модуля, каждый из которых решает определенный круг задач:

1. Модуль авторизации и аутентификации обеспечивает безопасный доступ к системе. Он реализует ролевую модель управления правами, где каждому пользователю назначается определенная роль (администратор, исполнитель, обычный пользователь). Модуль управляет пользовательскими сессиями и предоставляет механизмы восстановления доступа.
2. Модуль управления заявками является центральным компонентом системы. Он позволяет создавать новые заявки, назначать их конкретным исполнителям, изменять статусы выполнения и добавлять комментарии по ходу работы. Каждая заявка содержит информацию о приоритете выполнения, сроке обработки и истории изменений. На рисунке 2.4 представлена схема жизненного цикла заявки, начиная от момента ее создания до окончательного закрытия.

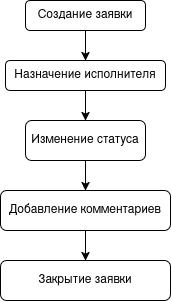


Рис. 2.4 - Жизненный цикл заявки

1. Модуль отчетности предоставляет инструменты для анализа работы отдела АСУ. Он позволяет генерировать стандартные отчеты. Особенностью модуля является возможность экспорта отчетов в формат DOCX для их дальнейшего использования в официальной документации. Отчеты формируются на основе актуальных данных и могут быть настроены в соответствии с потребностями руководства.
2. Административный модуль предназначен для управления системой. Он позволяет регистрировать новых пользователей, настраивать их права доступа, управлять структурными подразделениями и настраивать системные параметры. Администратор может изменять перечень возможных статусов заявок, настраивать приоритеты и категории задач.

Функционал работы с заявками включает полный цикл управления задачами:

* Создание новых заявок с указанием всех необходимых параметров;
* Редактирование существующих заявок;
* Закрытие выполненных заявок с указанием решения;
* Назначение приоритетов (низкий, средний, высокий);
* Изменение статусов выполнения (новая, в работе, на проверке, выполнена);

Функционал управления пользователями обеспечивает:

* Регистрацию новых пользователей с обязательной проверкой уникальности логина
* Назначение ролей и прав доступа в соответствии с должностными обязанностями
* Привязку пользователей к структурным подразделениям
* Возможность сброса паролей администратором системы

Пользовательский интерфейс системы включает несколько основных экранов, каждый из которых решает конкретные задачи:

1. Экран авторизации - точка входа в систему, где пользователь вводит свои учетные данные. Интерфейс минималистичен и содержит только необходимые элементы для входа.
2. Главное меню - центральный узел навигации, откуда пользователь может перейти к любому функционалу в соответствии со своими правами доступа. Меню адаптируется под роль пользователя, скрывая недоступные функции.
3. Список заявок - основной рабочий экран системы, где отображается перечень задач с возможностью поиска. Интерфейс предоставляет быстрый доступ к основным операциям с заявками.
4. Форма создания/редактирования заявки содержит все необходимые поля для ввода информации о задаче. Поля динамически проверяют вводимые данные и предупреждают об ошибках.
5. Профиль пользователя позволяет просматривать и редактировать личные данные, изменять пароль и настраивать некоторые параметры интерфейса.
6. Административная панель предоставляет расширенные инструменты управления системой, доступные только пользователям с соответствующими правами.

При проектировании интерфейса были применены следующие принципы UX/UI:

* Единый стиль оформления всех элементов интерфейса
* Интуитивно понятная навигация с минимальным количеством переходов
* Оптимизация рабочих процессов для сокращения количества действий
* Четкая визуальная иерархия информации
* Контекстные подсказки и сообщения об ошибках

Работа с базой данных организована с использованием драйвера Npgsql, который обеспечивает эффективное взаимодействие между клиентским приложением и PostgreSQL. Все запросы к базе данных оптимизированы для минимизации времени отклика. Критически важные операции, такие как обновление статусов заявок или изменение прав пользователей, выполняются в рамках транзакций для обеспечения целостности данных.

Бизнес-логика системы включает многоуровневую валидацию вводимых данных. Каждый уровень (клиентский, серверный и уровень базы данных) выполняет свою часть проверок. Обработка ошибок реализована через централизованную систему исключений, которая гарантирует, что ни одна ошибка не останется без внимания, а пользователь получит понятное сообщение о проблеме.

Для поддержания стабильной и бесперебойной работы системы внедрен комплексный подход к обеспечению надежности, охватывающий как обработку возможных сбоев, так и оптимизацию производительности.

В области обработки ошибок система оснащена многоуровневым механизмом защиты. Центральным элементом является глобальный обработчик исключений, который перехватывает и классифицирует все непредвиденные ситуации в процессе работы. Пользователи получают понятные сообщения об ошибках на естественном языке, что позволяет быстро понять суть проблемы без технических подробностей. Все критические события и ошибки фиксируются в системном журнале с детализацией времени возникновения, условий и контекста выполнения.

Вопросы производительности решаются комплексно на нескольких уровнях. Все SQL-запросы проходят тщательный анализ и оптимизацию для обеспечения максимальной скорости выполнения. В базе данных реализована продуманная система индексов, значительно ускоряющая поиск и выборку информации. Для операций, требующих значительного времени обработки, применяются асинхронные методы выполнения, что предотвращает блокировку интерфейса.

На рисунке 2.5 представлена схема базы данных, отражающая все таблицы и связи между ними. Схема разработана с соблюдением принципов нормализации до третьей нормальной формы, что обеспечивает эффективное хранение данных без избыточности.

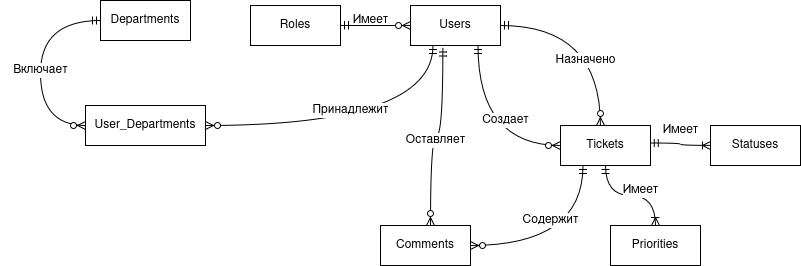


Рис. 2.5 - Схема базы данных

1. **Разработка концептуальной и логической моделей базы данных**

На этапе проектирования была разработана детальная концептуальная модель данных, представленная в виде ER-диаграммы (рис. 2.6). Эта модель отражает ключевые сущности системы и их взаимосвязи, формируя целостное представление о предметной области. Модель включает такие основные сущности как пользователи, роли, заявки, статусы, приоритеты, подразделения и комментарии, каждая из которых обладает уникальным набором атрибутов.

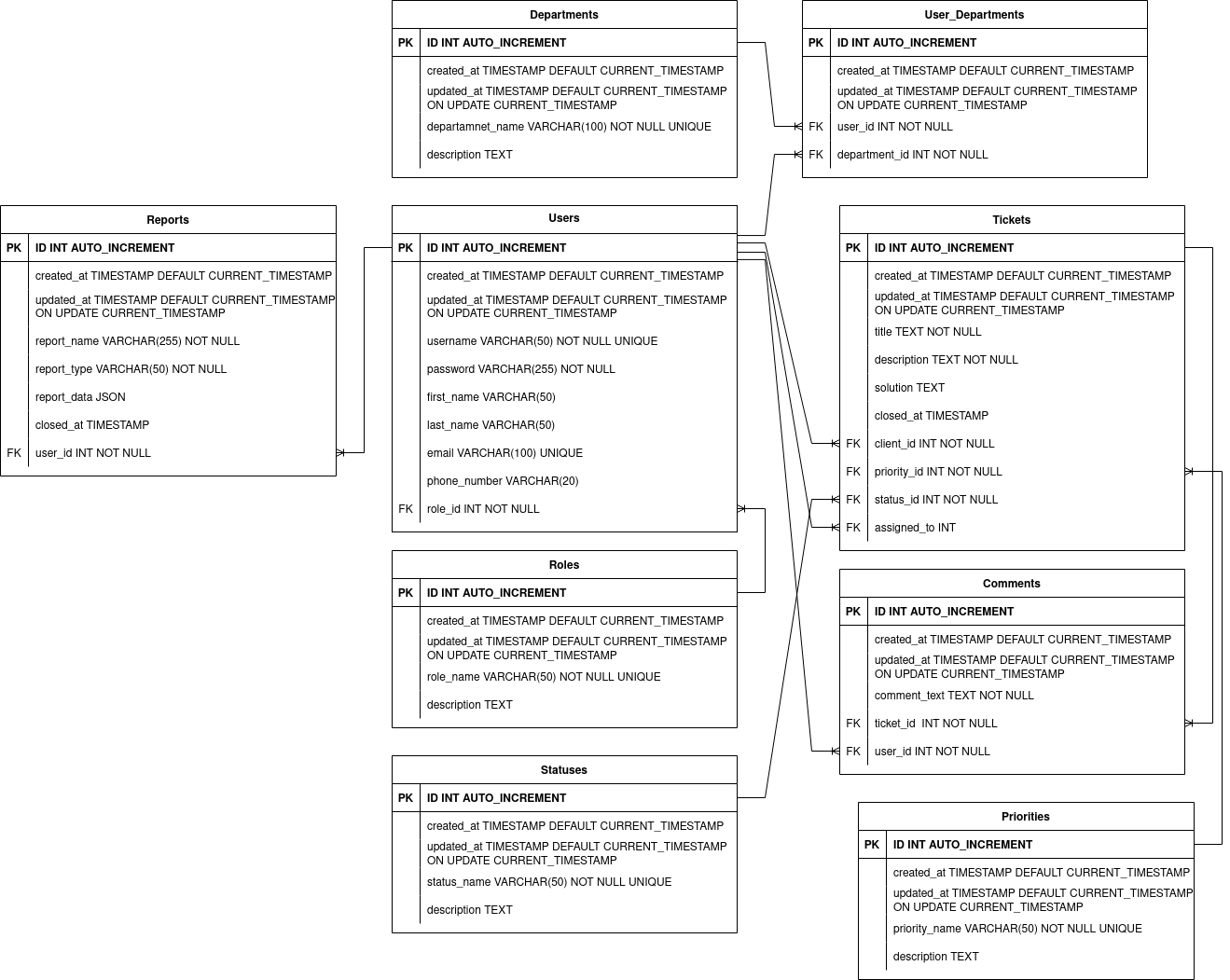


Рис. 2.6 - ER-диаграмма

На основе концептуальной модели была разработана логическая структура базы данных, реализованная в PostgreSQL. Рассмотрим подробно каждую таблицу и ее назначение.

1. Таблица Roles хранит информацию о ролевой модели системы:

CREATE TABLE Roles (

ID SERIAL PRIMARY KEY,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

role\_name VARCHAR(50) NOT NULL UNIQUE,

description TEXT

);

Эта таблица определяет различные роли пользователей (администратор, исполнитель, пользователь) и их описание. Автоматическое заполнение временных меток позволяет отслеживать историю изменений.

1. Таблица Users содержит учетные данные и персональную информацию пользователей:

CREATE TABLE Users (

ID SERIAL PRIMARY KEY,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

username VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100) UNIQUE,

phone\_number VARCHAR(20),

role\_id INT,

FOREIGN KEY (role\_id) REFERENCES Roles(ID)

);

В таблице хранятся все данные о пользователях, а уникальные ограничения на логин и email предотвращают дублирование учетных записей.

1. Таблица Statuses определяет жизненный цикл заявок:

CREATE TABLE Statuses (

ID SERIAL PRIMARY KEY,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

status\_name VARCHAR(50) NOT NULL UNIQUE,

description TEXT

);

Каждый статус (открыта, в работе, выполнена) имеет четкое описание, что упрощает работу с системой для новых пользователей.

1. Таблица Priorities устанавливает уровни важности задач:

CREATE TABLE Priorities (

ID SERIAL PRIMARY KEY,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

priority\_name VARCHAR(50) NOT NULL UNIQUE,

description TEXT

);

Система поддерживает стандартные приоритеты (низкий, средний, высокий), но структура таблицы позволяет легко добавлять новые уровни при необходимости.

1. Таблица Departments отражает организационную структуру учреждения:

CREATE TABLE Departments (

ID SERIAL PRIMARY KEY,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

department\_name VARCHAR(100) NOT NULL UNIQUE,

description TEXT

);

1. Таблица User\_Departments реализует связь между пользователями и подразделениями:

CREATE TABLE User\_Departments (

ID SERIAL PRIMARY KEY,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

user\_id INT,

department\_id INT,

FOREIGN KEY (user\_id) REFERENCES Users(ID),

FOREIGN KEY (department\_id) REFERENCES Departments(ID),

UNIQUE (user\_id, department\_id)

);

Уникальный индекс на паре user\_id и department\_id гарантирует, что один пользователь не будет привязан к одному подразделению несколько раз.

1. Таблица Tickets - центральная сущность системы:

CREATE TABLE Tickets (

ID SERIAL PRIMARY KEY,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

title TEXT NOT NULL,

description TEXT NOT NULL,

solution TEXT,

closed\_at TIMESTAMP,

client\_id INT,

priority\_id INT,

status\_id INT,

assigned\_to INT,

FOREIGN KEY (client\_id) REFERENCES Users(ID),

FOREIGN KEY (priority\_id) REFERENCES Priorities(ID),

FOREIGN KEY (status\_id) REFERENCES Statuses(ID),

FOREIGN KEY (assigned\_to) REFERENCES Users(ID)

);

Каждая заявка содержит полную информацию о проблеме, ее решении и участниках процесса. Внешние ключи обеспечивают целостность данных.

1. Таблица Comments хранит историю обсуждений:

CREATE TABLE Comments (

ID SERIAL PRIMARY KEY,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

comment\_text TEXT NOT NULL,

ticket\_id INT,

user\_id INT,

FOREIGN KEY (ticket\_id) REFERENCES Tickets(ID),

FOREIGN KEY (user\_id) REFERENCES Users(ID)

);

1. Таблица Reports предназначена для хранения отчетов:

CREATE TABLE Reports (

ID SERIAL PRIMARY KEY,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

report\_name VARCHAR(255) NOT NULL,

report\_type VARCHAR(50) NOT NULL,

report\_data JSONB,

user\_id INT,

FOREIGN KEY (user\_id) REFERENCES Users(ID)

);

Использование типа JSONB для report\_data позволяет гибко хранить различные форматы отчетов.

Разработанная структура базы данных соответствует принципам нормализации до третьей нормальной формы (3NF). Это означает, что все атрибуты не содержат повторяющихся групп и отсутствуют частичные зависимости от составного первичного ключа.

Для обеспечения целостности данных реализованы:

* Автоматическое обновление полей created\_at и updated\_at
* Внешние ключи для всех связей между таблицами
* Уникальные ограничения на критически важные поля
* Проверочные ограничения на допустимые значения

Оптимизация производительности достигается за счет использования транзакций для критических операций и оптимизированных SQL-запросов

Разработанная модель обеспечивает эффективное хранение данных, быстрый доступ к информации и простоту дальнейшего расширения функциональности системы. Гибкость структуры позволяет адаптировать базу данных к изменяющимся требованиям без кардинальных переработок.

**Вывод по главе**

В ходе проектирования информационной системы учета работы отдела АСУ для медицинского учреждения была проделана комплексная работа по формированию архитектуры, выбору технологий и разработке структуры базы данных. Основой системы стал технологический стек, включающий C# и Windows Forms для клиентской части, PostgreSQL в качестве СУБД и трехуровневую архитектуру, обеспечивающую четкое разделение ответственности между компонентами. Такой подход гарантирует надежность, безопасность и простоту эксплуатации системы, что особенно важно в условиях медицинского учреждения.

Ключевое внимание было уделено проектированию функциональной структуры, включающей модули авторизации, управления заявками, отчетности и администрирования. Каждый модуль разработан с учетом специфики работы медицинского персонала, обеспечивая интуитивно понятный интерфейс и оптимизированные рабочие процессы.

Разработанная концептуальная и логическая модель базы данных отражает все сущности предметной области и их взаимосвязи. Структура базы данных соответствует принципам нормализации до третьей нормальной формы, что минимизирует избыточность данных и обеспечивает целостность информации. Использование внешних ключей, уникальных ограничений и транзакций гарантирует корректность работы системы даже при высокой нагрузке.

Гибкость выбранных решений позволяет легко адаптировать систему к изменяющимся требованиям, добавлять новые функции и масштабировать ее в будущем. В целом, проведенное проектирование создает прочную основу для разработки информационной системы, отвечающей всем поставленным задачам и требованиям заказчика.

**Глава 3. РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

1. **Реализация базы данных**

Процесс реализации базы данных для информационной системы учета работы отдела АСУ включал несколько ключевых этапов: установку и настройку СУБД, проектирование структуры, наполнение тестовыми данными и обеспечение безопасности. В качестве системы управления базами данных была выбрана PostgreSQL 15, установленная на операционную систему Windows 10. Выбор данной СУБД обусловлен ее надежностью, производительностью и соответствием требованиям проекта.

Установка PostgreSQL выполнялась через официальный инсталлятор EnterpriseDB с использованием графического интерфейса. В процессе установки были заданы следующие параметры:

* Пароль для суперпользователя postgres
* Порт подключения по умолчанию: 5432
* Локализация: Russian\_Russia.1251
* Компоненты: сервер, pgAdmin, командные строки

После установки была выполнена первоначальная настройка через утилиту pgAdmin 4 (рис. 3.1), которая предоставляет удобный графический интерфейс для администрирования PostgreSQL.

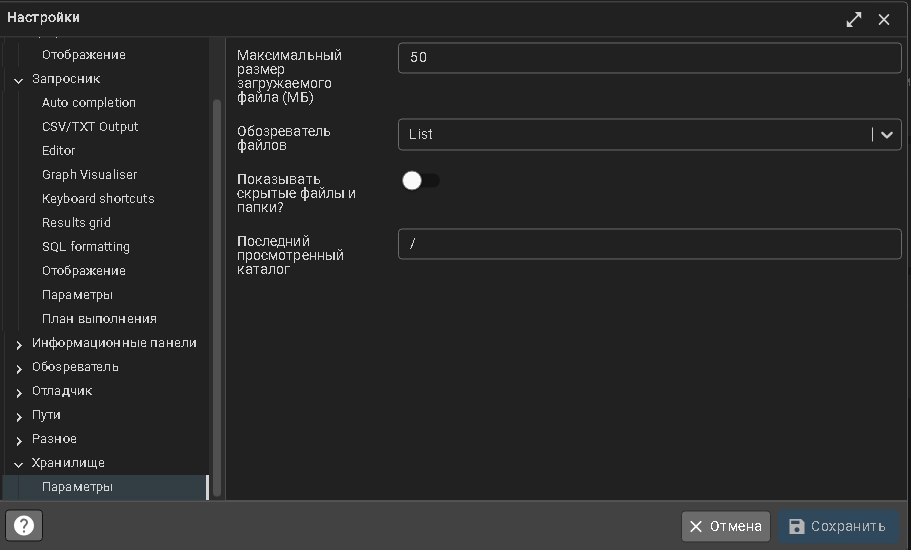


Рис. 3.1 - Настройка pgAdmin 4

Структура базы данных была реализована в соответствии с логической моделью, разработанной на этапе проектирования. Для создания таблиц использовались SQL-запросы, выполненные через интерфейс pgAdmin. Пример создания основных таблиц:

CREATE TABLE roles (

id SERIAL PRIMARY KEY,

role\_name VARCHAR(50) NOT NULL UNIQUE,

description TEXT,

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

CREATE TABLE users (

id SERIAL PRIMARY KEY,

username VARCHAR(50) NOT NULL UNIQUE,

password VARCHAR(255) NOT NULL,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100),

role\_id INTEGER REFERENCES roles(id),

created\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

updated\_at TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

Полная схема базы данных, отображающая все таблицы и связи между ними, представлена на рисунке 2.6.

Для тестирования функциональности системы база данных была заполнена тестовыми данными. Пример добавления тестовых записей:

-- Добавление ролей

INSERT INTO Roles (role\_name, description) VALUES

('Admin', 'Полный доступ ко всем функциям системы'),

('ASU\_staff', 'Доступ к заявкам и отчетам'),

('Client', 'Может создавать заявки и просматривать свои заявки');

INSERT INTO users (username, password, first\_name, last\_name, email, role\_id) VALUES

('admin',’ admin\_password', 'Иван', 'Иванов', 'admin@example.com', 1),

(‘staff'', ‘staff\_password', 'Петр', 'Петров', staff@example.com', 2),

('client;', ‘client\_password', 'Сергей', 'Сергеев', client@example.com', 3);

Пример тестовых данных для таблицы заявок:

INSERT INTO tickets (title, description, client\_id, priority\_id, status\_id, assigned\_to) VALUES

('Не работает принтер', 'В кабинете 205 не печатает принтер HP LaserJet', 3, 2, 1, 2),

('Ошибка в программе', 'При попытке открыть историю болезни выдает ошибку', 3, 1, 2, 2);

Для защиты данных были реализованы следующие меры безопасности:

1. Настройка ролевого доступа в PostgreSQL
2. Регулярное создание резервных копий с помощью pg\_dump

Пример резервного копирования:

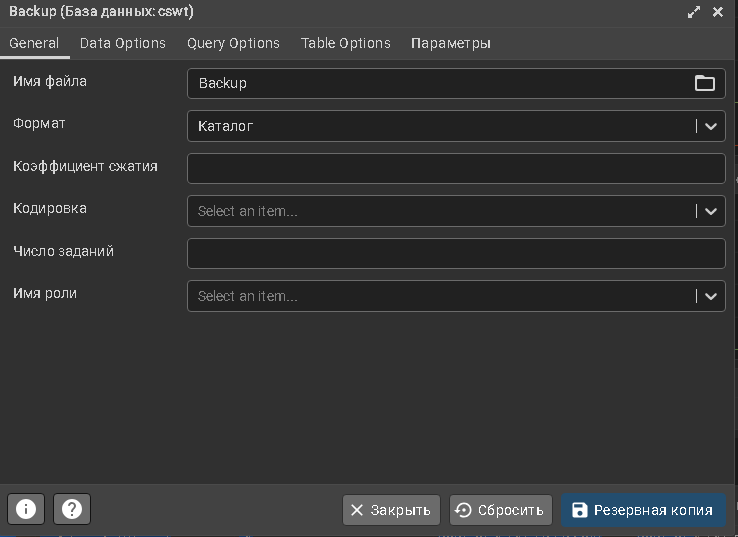


Рис. 3.2 - Резервная копия

Реализованная база данных полностью соответствует требованиям проекта и обеспечивает:

* Надежное хранение данных
* Быстрый доступ к информации
* Целостность данных через систему ограничений
* Простоту масштабирования и расширения

Для интеграции базы данных с клиентским приложением использовался драйвер Npgsql. Пример подключения:

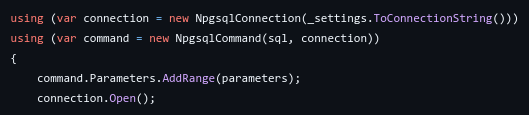


Рис. 3.3 - Подключение к БД

На рисунке 3.3 из \_settings получаем строку подключения и открываем подключение

1. **Реализация клиентского приложения**

Клиентское приложение информационной системы учета работы отдела АСУ было разработано с использованием языка программирования C#, Windows Forms и платформы .NET Framework 4.7.2. В качестве среды разработки применялась Microsoft Visual Studio Community 2022, как было сказано в главе 2.1.

Архитектура приложения следует принципам трехуровневой архитектуры с четким разделением ответственности между компонентами, что обеспечивает высокую сопровождаемость и возможность дальнейшего расширения системы.

Архитектура состоит из 3 слоев:

1. Уровень представления представляет собой формы Windows Forms которые составляют основу пользовательского интерфейса. Каждая форма отвечает за отображение данных и сбор пользовательского ввода, делегируя обработку бизнес-логике (рис. 3.4). Основные формы системы: вход, регистрация, создание заявки, управление заявками, редактирование заявок в зависимости от роли пользователя и отчеты.

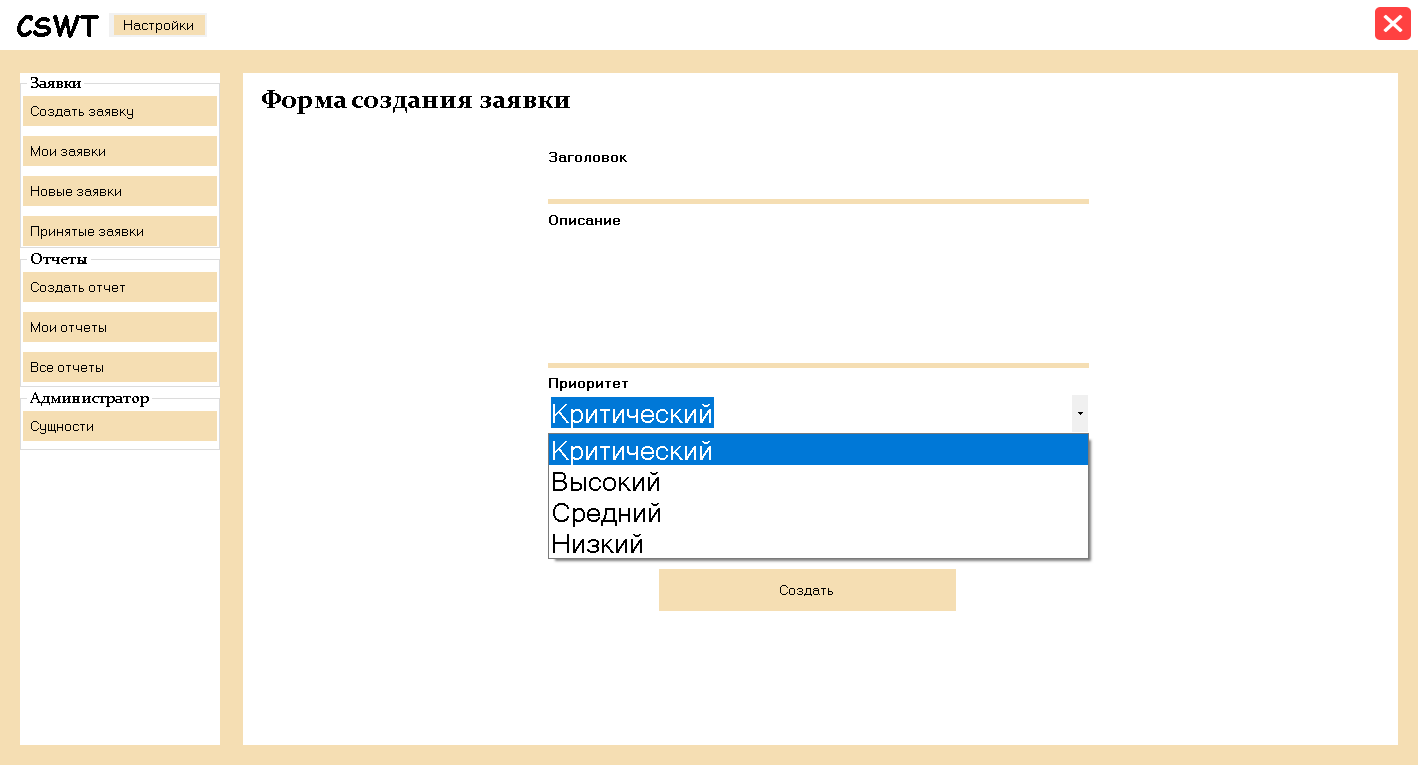


Рис. 3.4 - Уровень представления

1. Уровень бизнес-логики реализован через систему контроллеров, отвечающих за обработку и передачи данных в модели (рис. 3.5). У каждой формы есть свой сервис.



Рис. 3.5 - Уровень бизнес-логики

1. Уровень доступа к данным включает репозиторий, отвечающий за управлением соединения, отправку запросов и вывода ошибок. Также туда входят модели и сервисы, которые создают и отправляют запросы в репозиторий.

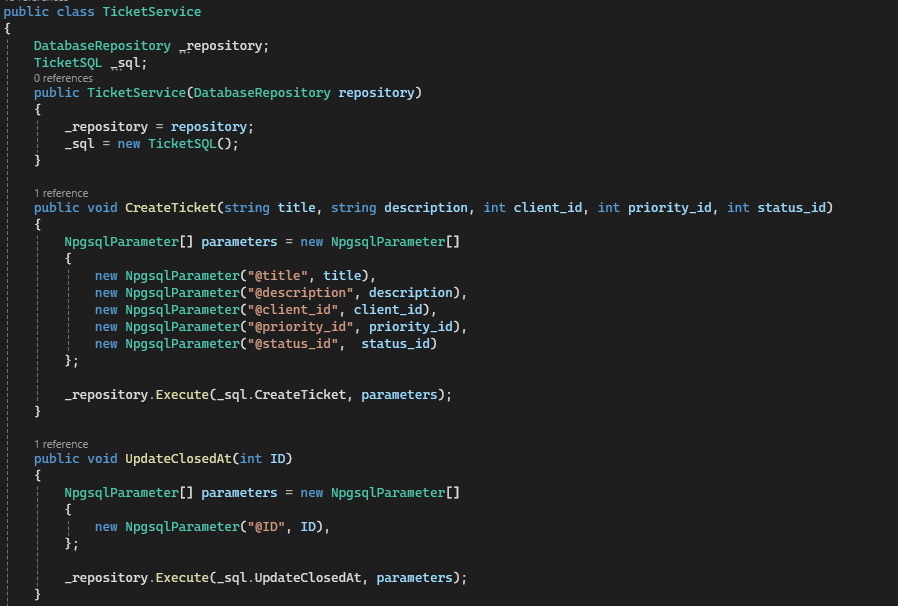


Рис. 3.6 - Уровень данных

**Авторизации состоит из 2 модулей: вход и регистрация** . При входе в систему пользователь проходит многоэтапную проверку: сначала проверяется наличие учетной записи с указанным логином, затем корректность введенного пароля. При регистрации проводится проверка на наличия пользователя с таким же паролем и почтой, и в случае существования такого пользователя будет ошибка. После успешной регистрации всплывет соответствующее сообщение (рис. 3.7). После успешной аутентификации система устанавливает соответствующие права доступа в зависимости от роли пользователя.

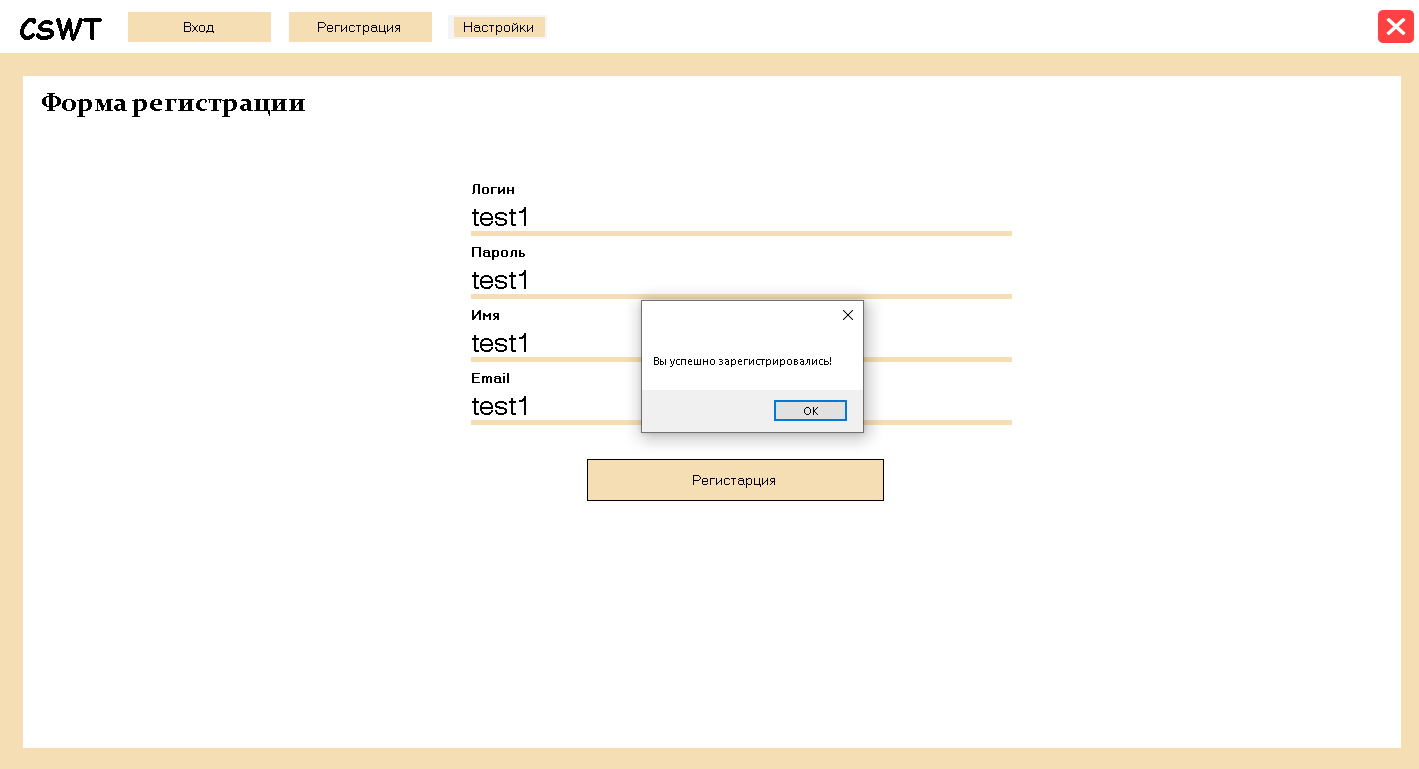


Рис. 3.7 - Модуль регистрации

Модули управления заявками представляет собой центральные компоненты системы, обеспечивающие полный жизненный цикл обработки заявок - от создания до закрытия. При создании новой заявки система автоматически присваивает ей уникальный идентификатор, фиксирует дату и время создания, а также связывает заявку с учетной записью создателя. Для каждой заявки предусмотрены обязательные поля: название проблемы, описание проблемы, статус и приоритет. Система позволяет назначать исполнителей, изменять статусы выполнения, добавлять комментарии.

Пользователь с уровнем доступа клиента может создавать заявку, просмотреть заявки, удалить, редактировать и писать комментарии к своим заявкам.

Пользователь с доступом сотрудника АСУ имеет доступ к новым заявкам с возможность принятия ее, к своим принятым заявкам, к комментариям, к редактированию и выполнению заявки (рис. 3.8)

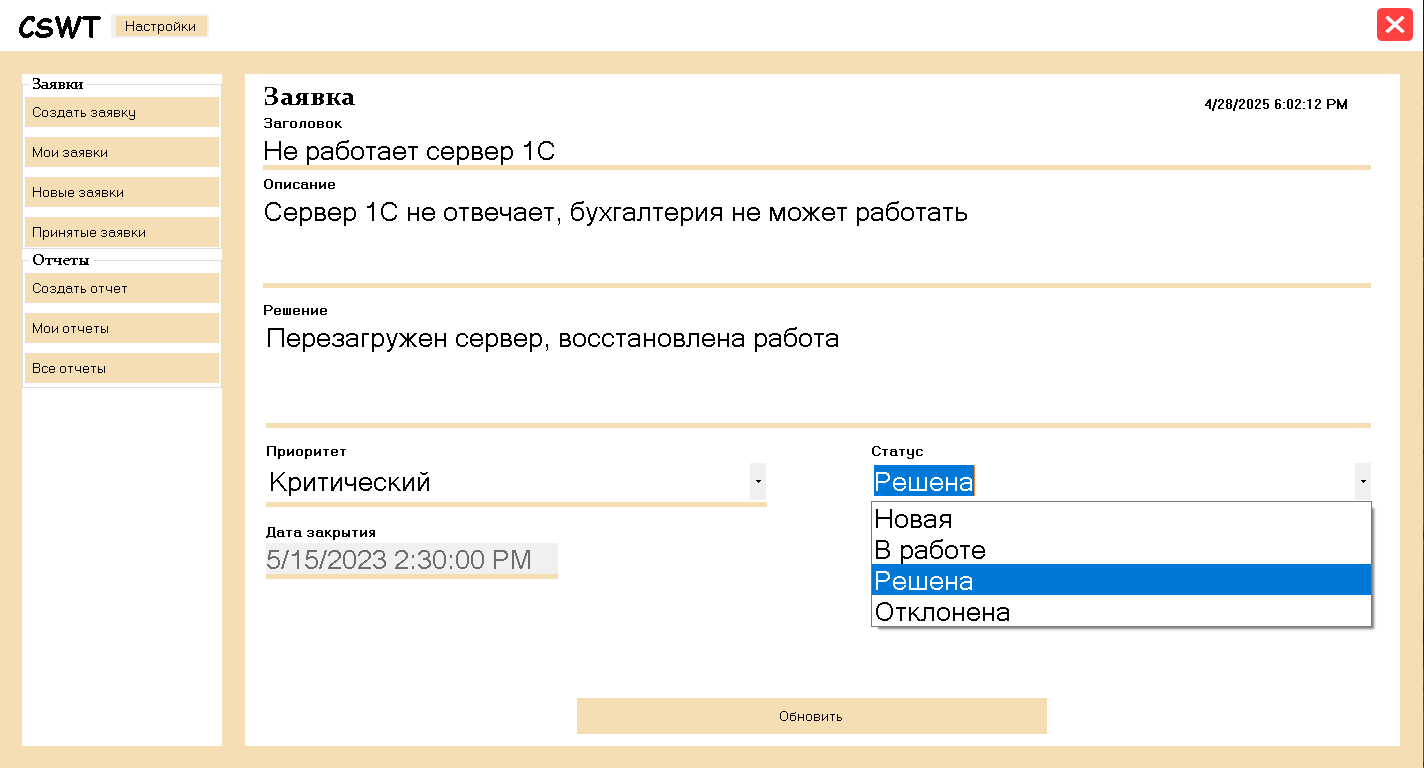


Рис. 3.8 - Модуль заявки

Модуль отчетности предоставляет широкие возможности для анализа работы отдела АСУ. Система поддерживает генерацию отчетов различных типов: статистические отчеты по всем заявкам, отчеты заявкам за последний месяц, отчеты по сотрудникам. Для каждого отчета предусмотрен различные формат вывода - DOCX или печать. Модуль отчетности состоит из генерации отчета, просмотра своих и всех отчетов и формы просмотра отчета.

Отчет состоит из названия, типа и данных (Рис. 3.9). Названия и тип генерируются автоматически, а данные собираются из БД.

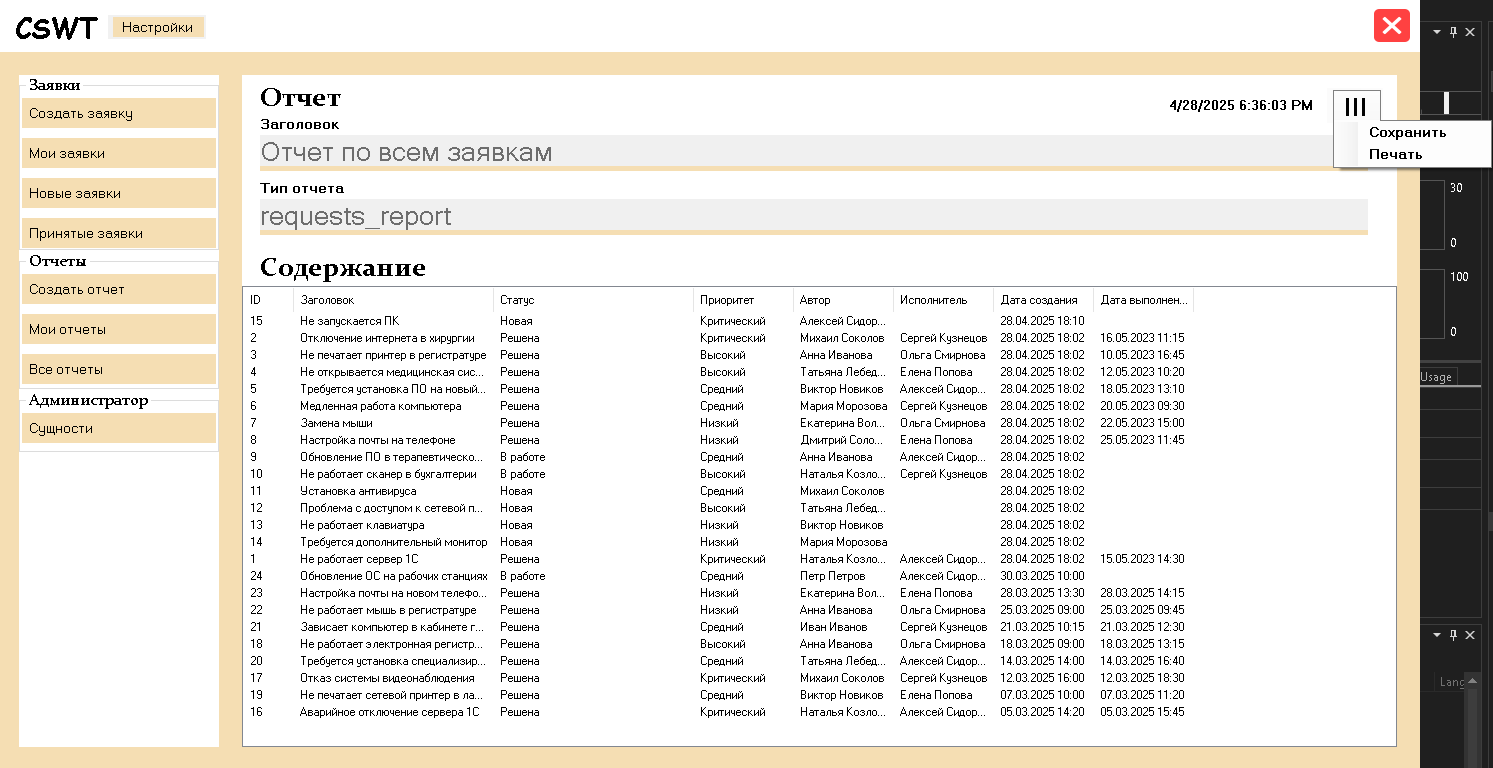


Рис. 3.9 - Модуль отчета

Административный модуль предназначен для управления системными параметрами и настройками. Администраторы могут управлять всеми данными системы - статусами, приоритетами, структурными подразделениями, заявками, пользователями, ролями (Рис. 3.10). Для каждого справочника предусмотрены функции добавления и удаления записей с обязательной проверкой на использование в существующих заявках.

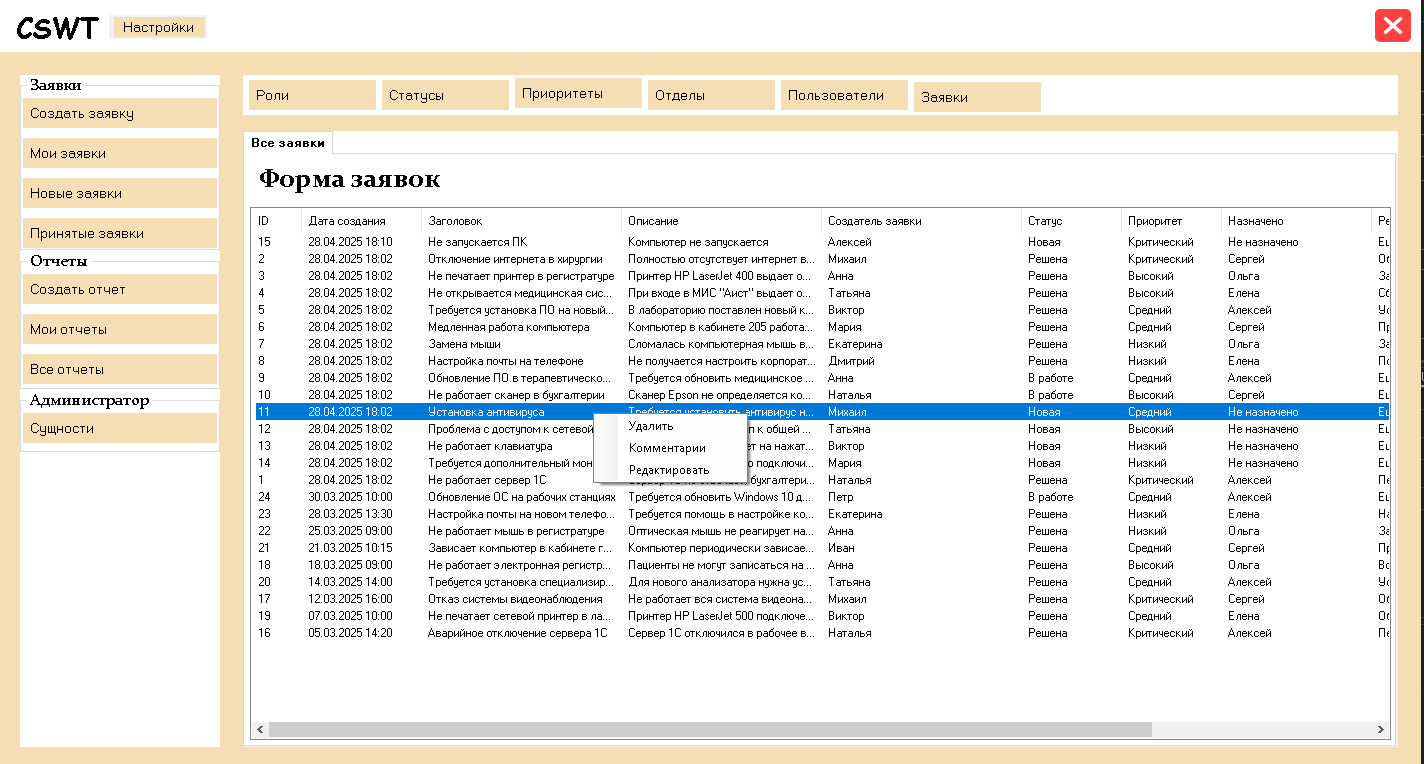


Рис. 3.10 - Управление заявками администратором

Также был разработан модель отвечающий за работы с подключение к БД.

Он содержит данные о БД, нужные для подключения. Данные хранятся в отдельном файле и при запросе достаются из этого файла. Сохранить настройки можно при помощи формы настроек БД (Рис. 3.11)

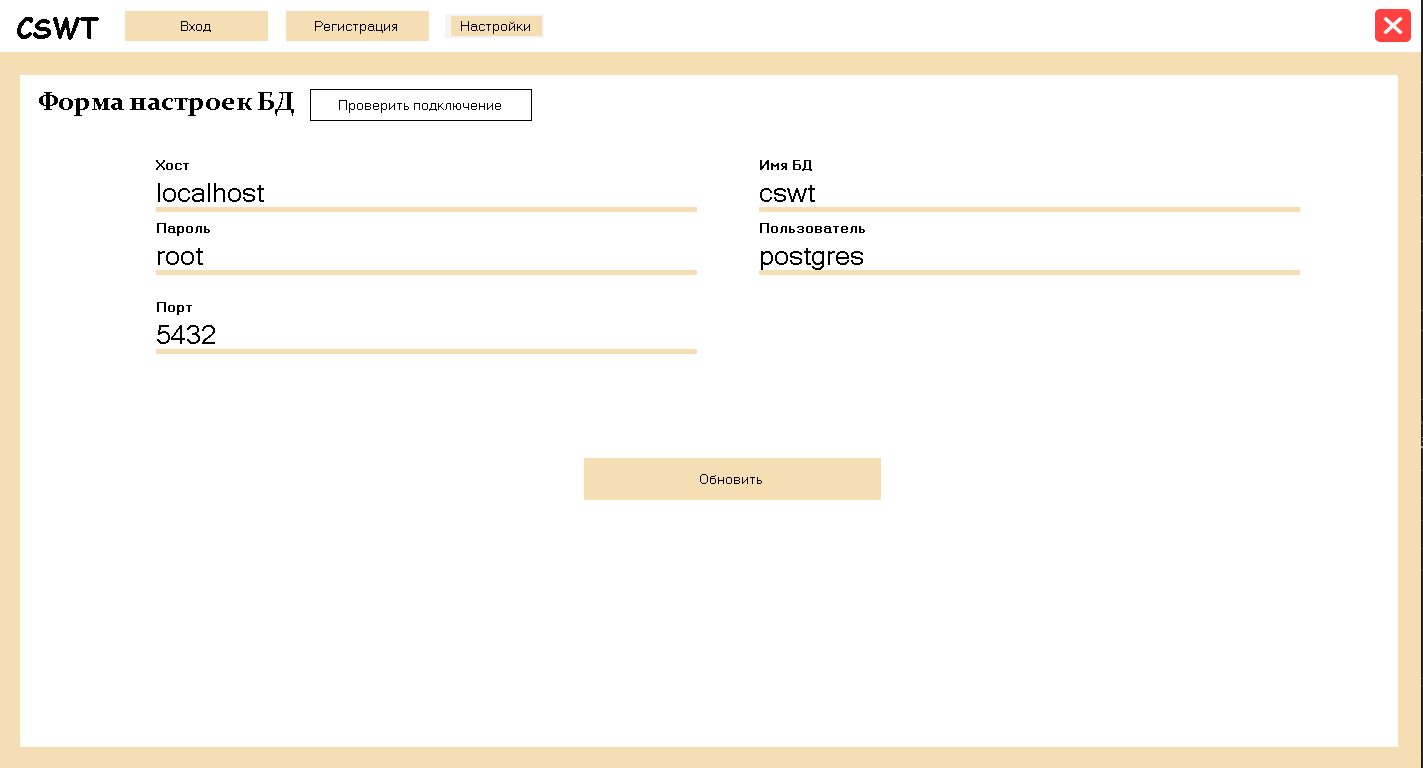


Рис. 3.11 - Подключение к БД

Модуль профиля позволяет обновить свой профиль (Рис. 3.12). Можно изменить все данные кроме роли, она доступны только администраторам.



Рис. 3.12 - Профиль

Каждый модуль системы был реализован с учетом требований надежности и отказоустойчивости. Все критические операции выполняются в рамках транзакций, что гарантирует целостность данных даже в случае сбоев.

Интерфейсы всех модулей были разработаны с учетом эргономических требований и особенностей работы медицинского персонала. Особое внимание уделено простоте освоения - даже новые пользователи могут быстро начать работу с системой благодаря интуитивно понятному интерфейсу и встроенным подсказкам. Все элементы интерфейса локализованы на русский язык и соответствуют принятой в организации терминологии.

Техническая реализация модулей основана на современных технологиях и подходах к разработке программного обеспечения. Код системы хорошо структурирован, содержит подробные комментарии и соответствует принятым стандартам именования. Это существенно упрощает дальнейшее развитие и сопровождение системы, позволяя новым разработчикам быстро вникать в особенности реализации.

1. **Руководство пользователя**

Руководство предназначено для всех категорий пользователей информационной системы учета работы отдела автоматизированных систем управления ГБУЗ "Лабинская центральная районная больница". Разработанная система представляет собой современное программное решение, позволяющее оптимизировать процессы обработки заявок, контроля выполнения задач и формирования аналитической отчетности.

Руководство содержит подробные инструкции по работе с системой, начиная от процедуры установки и заканчивая решением типовых проблем. Особое внимание уделено описанию функциональных возможностей системы для различных категорий пользователей - обычных сотрудников, специалистов отдела АСУ и администраторов системы.

Перед первым запуском системы необходимо выполнить ее установку на рабочую станцию. Для корректной работы приложения требуется компьютер с операционной системой Windows 10 или более новой версией. Обязательным условием является наличие установленной платформы .NET Framework версии 4.7.2 или выше. В случае работы в сетевом режиме необходимо предварительно настроить подключение к серверу базы данных PostgreSQL.

Процедура входа в систему начинается с запуска клиентского приложения. На экране авторизации пользователю необходимо ввести свои учетные данные. После проверки введенных данных система предоставляет доступ к функционалу в соответствии с назначенной ролью пользователя. В случае неверного ввода данных система отображает соответствующее сообщение об ошибке.

Для восстановления доступа при утрате пароля необходимо обратится к администратору.

Работа с заявками является ключевой функцией системы. Для создания новой заявки пользователю необходимо перейти в соответствующий раздел меню и заполнить обязательные поля формы: название проблемы, ее описание и уровень приоритета. После сохранения заявка поступает в обработку и становится доступной для просмотра и редактирования.

В разделе "Мои заявки" пользователь может просматривать список всех созданных им обращений. Для каждой заявки предусмотрена возможность редактирования (изменения описания и названия) и удаления. Особое внимание уделено механизму комментирования - пользователи могут добавлять пояснения и уточнения по ходу обработки заявки, что значительно улучшает коммуникацию между сотрудниками.

Для специалистов отдела АСУ система предоставляет расширенный функционал управления задачами. В разделе "Новые заявки" отображаются все поступившие обращения, которые можно принять в работу. После принятия заявки сотрудник может изменять ее статус в соответствии с этапами обработки: "В работе", "Отклонено" или "Выполнено". Это позволяет четко отслеживать прогресс решения каждой проблемы.

Формирование отчетов - важная аналитическая функция системы. Пользователи могут генерировать различные типы отчетов: статистику по заявкам за выбранный период, анализ выполненных задач за месяц или отчеп по пользователям. Сформированные отчеты можно экспортировать в формат DOCX для дальнейшего редактирования или сразу выводить на печать.

Администраторам системы доступен расширенный функционал управления. В их обязанности входит регистрация новых пользователей, редактирование их учетных данных, удаление учетных записей при необходимости. Также администраторы могут настраивать системные данные: добавлять новые статусы заявок, виды приоритетов, структурные подразделения организации.

В процессе работы с системой пользователи могут столкнуться с некоторыми трудностями. Наиболее распространенной проблемой является невозможность входа в систему. В этом случае рекомендуется проверить правильность ввода логина и пароля, обратив внимание на состояние клавиши Caps Lock. Если проблема сохраняется, следует обратиться к администратору системы.

Другая частая ситуация - невозможность сохранения созданной заявки. В этом случае необходимо проверить заполнение всех обязательных полей формы (название и описание проблемы), а также убедиться в наличии активного подключения к серверу базы данных.

Для изменения персональных данных пользователю необходимо перейти в раздел "Профиль" и выбрать пункт меню "Редактировать данные". Здесь можно изменить контактную информацию и другие параметры учетной записи (за исключением роли, которые изменяются только администратором).

В случае возникновения технических проблем, не описанных в данном руководстве, пользователи могут обратиться в службу технической поддержки. Контактная информация для обращения: электронная почта [support@labinsk-crb.ru](https://mailto:support@labinsk-crb.ru" \t "/home/Shau/Documents\\x/_blank). Специалисты поддержки готовы помочь в решении любых вопросов, связанных с работой системы.

Список использованных источников:

1. ГОСТ Р 52636-2006 "Электронные медицинские карты"
2. ГОСТ Р 55079-2012 "Информационные технологии в здравоохранении"
3. ГОСТ 34.601-90 "Автоматизированные системы. Стадии создания"
4. ГОСТ Р ИСО/МЭК 12207-2010 "Процессы жизненного цикла программных средств"
5. Приказ Минздрава РФ от 28.02.2019 N 124н "О порядке ведения медицинской документации"
6. Хомоненко А.Д. Базы данных. - СПб.: Корона-Век, 2020. - 736 с.
7. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. - М.: Вильямс, 2021. - 896 с.
8. Фаулер М. Архитектура корпоративных программных приложений. - М.: Вильямс, 2019. - 544 с.
9. Устав ГБУЗ "Лабинская ЦРБ" URL: Федеральный закон от 29.07.2018 № 250-ФЗ «О цифровых технологиях в здравоохранении».
10. Приказ Минздрава РФ от 28.02.2019 № 124н «О порядке ведения медицинской документации»
11. Регламент работы отдела АСУ ГБУЗ "Лабинская ЦРБ" (внутренний документ, 2023 г.)
12. Методические рекомендации по подготовке и оформлению дипломных проектов для специальности 09.02.07 "Информационные системы и программирование". ГАПОУ КК "Лабинский аграрный техникум", 2025 г.
13. Федеральный закон № 323-ФЗ "Об основах охраны здоровья граждан в Российской Федерации"
14. Приказ Минздрава России № 786н "Об утверждении Правил организации системы документооборота в сфере охраны здоровья"
15. ГОСТ Р 52653-2006 "Информационно-коммуникационные технологии в здравоохранении"
16. Материалы внутреннего аудита информационных систем ГБУЗ "Лабинская ЦРБ" за 2024 год
17. Аналитический отчет о состоянии ИТ-инфраструктуры медицинских учреждений Краснодарского края за 2024 год
18. Агальцов В.П. Базы данных. — М.: ИНФРА-М, 2019.
19. Алексеев Ю. Е., Ваулин А. С., Куро А. В. Практикум по программированию. Обработка числовых данных. — М.: Изд. МГТУ им. Н. Э. Баумана, 2019.
20. Гохберг Г.С., Зафиевский А.В., Короткин А.В. Информационные технологии. — М.: Издательский центр «Академия», 2020. — 208 с.
21. Гребенюк Е.И., Гребенюк Н.А. Технические средства информатизации. — М.: Издательский центр «Академия», 2020. — 272 с.
22. Михеева Е.В. Информационные технологии в профессиональной деятельности. — М.: Издательский центр «Академия», 2021. — 384 с.
23. Михеева Е.В. Практикум по информационным технологиям в профессиональной деятельности. — М.: Издательский центр «Академия», 2021. — 256 с.
24. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных. — Спб.: Корона, 2019. — 736 с.