

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра _____ «Прикладная математика и информатика» _____
(наименование)

09.03.03 Прикладная информатика

(код и наименование направления подготовки / специальности)

Разработка социальных и экономических информационных систем

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Разработка системы онлайн записи на прием в городскую больницу»

Обучающийся

Е.В. Козлов

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд.пед.наук, доцент, О.М. Гущина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

И.Ю. Усатова

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема: «Разработка системы онлайн записи на прием в больницу».

В современном мире информационные технологии все активнее проникают во все сферы жизни, в том числе и в систему здравоохранения. Создание системы онлайн-записи на прием в больницу- важная задача, которая позволит оптимизировать работу медицинских учреждений, повысить качество обслуживания пациентов и сделать доступ к медицинским услугам более удобным.

Актуальность работы обусловлена растущим спросом на медицинские услуги и необходимостью их оптимизации с помощью современных информационных технологий. Разработанная система позволит сделать процесс записи на прием более удобным и быстрым как для пациентов, так и для медицинского персонала, тем самым повышая общую эффективность работы.

Целью данного исследования является разработка системы онлайн записи на прием к врачу, которая будет соответствовать современным требованиям и отвечать потребностям как пациентов, так и медицинских учреждений.

Объём работы: страниц: 64, рисунков: 30, таблиц: 9. При написании работы использовалось источников: 26.

Ключевые слова: веб, онлайн, HTML, CSS, база данных, MySQL, приложение.

Объект исследования – система онлайн записи на прием в больницу.

В работу входит введение, три раздела, заключение, список литературы и приложение.

Abstract

The title of the graduation work is "The development of an online appointment system in a hospital".

The graduation work consists of an introduction, three parts, a conclusion, 9 tables, 25 sources.

The key issue of the work is the importance and the effectiveness of the use of information system of an online appointment system in a hospital.

The aim of the graduation work is to develop an online appointment system in a hospital, which will contribute to optimising the work of medical institutions and improve the quality of patient care.

The graduation work may be divided into several logically connected parts which are: analysis of the subject area of a manufacturing company; logical design of an information system for online appointment; physical design of the information system.

Finally, we create a website, which will fulfill all the formed requirements, such as: providing the possibility of online appointment, editing the data of formed orders, providing data in a convenient form.

In conclusion we would like to stress that this work is relevant because in the present time there is a need to switch hospital appointments to an online format, which makes this study important in the context of modern information technology requirements. The practical significance of the bachelor's work lies in the development of an online hospital appointment system.

Содержание

Введение.....	6
1 Функциональное моделирование предметной области	8
1.1 Техничко-экономическая характеристика предметной области.....	8
1.2 Разработка и анализ модели бизнес-процесса «как есть».....	9
1.3 Разработка модели бизнес-процесса «как должно быть»	11
1.4 Анализ существующих разработок на предмет соответствия сформулированным требованиям.....	13
1.5 Постановка задачи на разработку проекта создания/внедрения АИС ..	17
2 Проектирование системы онлайн записи на прием в городскую больницу	19
2.1 Логическое моделирование ИС и ее архитектура	19
2.1.1 Логическая модель информационной системы и её описание.....	20
2.1.2 Диаграмма взаимосвязи программных модулей и информационных файлов.....	24
2.2 Описание архитектуры АИС.....	25
2.3 Выбор технологии разработки программного обеспечения АИС	28
2.4 Проектирование базы данных АИС	32
2.5 Выбор СУБД АИС	36
2.6 Разработка физической модели данных АИС	39
3 Разработка системы онлайн записи на прием в городскую больницу и оценка ее эффективности	44
3.1 Разработка программного обеспечения АИС	44
3.1.1 Схема взаимосвязи модулей приложения АИС.....	44
3.1.2 Описание модулей приложения АИС с примерами программного кода.....	45
3.2 Описание функциональности АИС.....	46
3.3 Оценка и обоснование экономической эффективности разработки АИС	53
3.3.1 Выбор методики расчета экономической эффективности	53

3.3.2 Расчет показателей экономической эффективности проекта.....	55
Заключение	61
Список используемой литературы	63

Введение

В современном мире, где темпы жизни постоянно ускоряются, важность доступа к быстрым и эффективным медицинским услугам неоспорима. С развитием информационных технологий и цифровизацией общества появилась возможность существенно улучшить качество и доступность медицинских услуг. Одним из направлений такого улучшения является разработка системы онлайн записи на прием к врачу в больницах, что позволяет значительно сократить время ожидания для пациентов и повысить эффективность работы медицинского персонала.

Целью данной выпускной квалификационной работы является разработка системы онлайн записи на прием в больницу, которая будет способствовать оптимизации работы медицинских учреждений и улучшению качества обслуживания пациентов. В рамках работы будут рассмотрены существующие решения в данной области, проведен анализ их недостатков и преимуществ, а также разработана собственная система, учитывающая специфику работы медицинских учреждений Российской Федерации.

Объектом исследования является система онлайн записи на прием в больницу.

Предметом исследования является разработка, тестирование и внедрение системы в медицинское учреждение.

Для достижения поставленной цели предусмотрено решение следующих задач:

- Исследование существующих решений систем онлайн записи на прием в медицинские учреждения и анализ их эффективности.
- Определение требований к разрабатываемой системе, учитывая потребности как медицинских учреждений, так и их пациентов.
- Разработка архитектуры и дизайна предлагаемой системы.
- Программная реализация системы, включая разработку интерфейса пользователя и обеспечение безопасности данных.

- Тестирование системы и анализ её эффективности на примере конкретного медицинского учреждения.

Актуальность работы обусловлена растущим спросом на медицинские услуги и необходимостью их оптимизации с помощью современных информационных технологий. Разработанная система позволит сделать процесс записи на прием к врачу более удобным и быстрым как для пациентов, так и для медицинского персонала, тем самым повышая общую эффективность работы медицинских учреждений.

Результатом данной работы будет разработанная программа онлайн записи на прием в больницу.

Данная работа состоит из введения, трех разделов, заключения и списка используемой литературы.

Первый раздел посвящен анализу предметной области и постановке целей и задач для разработки информационной системы, а также концептуальному моделированию предметной области.

Второй раздел посвящен логическому проектированию информационной системы.

Третий раздел посвящен разработке и тестированию готовой информационной системы.

В заключении содержатся результаты разработки информационной системы.

Работа включает: 64 страницы, 30 рисунков, 9 таблиц и 26 источников.

1 Функциональное моделирование предметной области

1.1 Технико-экономическая характеристика предметной области

Городская клиническая больница №5 — крупное медицинское учреждение, оказывающее широкий спектр медицинских услуг жителям крупного мегаполиса. В структуру больницы входят различные отделения: терапевтическое, хирургическое, кардиологическое, неврологическое и др. Ежедневно в больнице регистрируется около 1000 обращений пациентов, что ставит перед учреждением задачи по оптимизации работы регистратуры и сокращению времени ожидания приема.

Технико-экономическое обоснование внедрения системы онлайн записи.

Существующая система записи на прием, основанная на телефонных звонках и личном обращении пациентов в регистратуру, не способна обеспечить быстрое и удобное обслуживание. Внедрение системы онлайн записи предполагает создание удобного веб-интерфейса и мобильного приложения, через которые пациенты могут записаться на прием к нужному специалисту в удобное время, а также отменить или перенести запись при необходимости. Это позволит сократить нагрузку на персонал регистратуры и повысить удовлетворенность пациентов качеством обслуживания.

Экономические выгоды внедрения системы онлайн записи многоаспектны:

- Сокращение времени ожидания приема позволит обслуживать больше пациентов без увеличения числа персонала.
- Оптимизация работы медперсонала за счет автоматизации процессов записи на прием и управления расписанием врачей.
- Снижение затрат на поддержку телефонной линии регистратуры и сокращение объема бумажной документации.

- Привлечение новых пациентов за счет повышения уровня сервиса и доступности услуг.

Внедрение системы онлайн записи в Городской клинической больнице №5 позволит достичь следующих результатов:

- Сокращение среднего времени ожидания приема на 30%.
- Увеличение количества обращений пациентов на 15% благодаря улучшению сервиса.
- Повышение удовлетворенности пациентов услугами больницы.
- Оптимизация работы медицинского и административного персонала.

Таким образом, внедрение системы онлайн записи на прием в больницу является экономически оправданным и технически реализуемым проектом, который позволит повысить эффективность работы медицинского учреждения и качество обслуживания пациентов.

1.2 Разработка и анализ модели бизнес-процесса «как есть»

Разработка и анализ модели бизнес-процесса "как есть" для системы онлайн записи на прием в больницу включает в себя изучение и документирование существующих процессов записи на прием к врачу, начиная от момента появления у пациента потребности в медицинской услуге и заканчивая фактическим посещением врача. Этот анализ позволяет выявить проблемы и неэффективные места в текущей системе, определить потребности пользователей и сформировать требования к автоматизированному варианту решения.

Шаги разработки модели бизнес-процесса «как есть».

Идентификация ключевых участников процесса: определение ролей, участвующих в процессе записи на прием, включая пациентов, врачей, медицинских администраторов и другой персонал больницы.

Сбор информации: изучение текущих методов записи на прием, включая телефонные звонки, личные обращения в регистратуру и использование электронной почты или других неформализованных методов.

Анализ действующих процедур: оценка эффективности существующих процедур записи, выявление задержек, избыточных действий и других проблемных моментов. [4]

Моделирование процесса: создание детальной модели процесса «как есть» с помощью диаграмм, например, с использованием нотации BPMN. Диаграмма должна включать все шаги процесса, от момента выявления потребности в приеме до завершения приема.

Анализ проблемных областей: идентификация узких мест, неэффективных взаимодействий и других аспектов, требующих улучшения.

Примерные области для улучшения:

- Время ожидания при записи по телефону может быть сокращено за счет автоматизации процесса.
- Ошибки при ручном вводе данных могут быть минимизированы с помощью электронной системы записи.
- Неудобство в использовании текущих методов записи для некоторых категорий пациентов, особенно для людей с ограниченными возможностями или для тех, кто предпочитает цифровые каналы связи.
- Недостаточная информация о доступных временных слотах и услугах врача, доступная пациентам при текущем методе записи.

Анализ модели бизнес-процесса «как есть» позволяет выявить ключевые проблемы и области для улучшения, которые должны быть учтены при разработке автоматизированной системы онлайн записи на прием в больницу. Основываясь на этом анализе, можно разработать требования к новой системе, которая будет эффективно адресовать выявленные проблемы, улучшая качество обслуживания пациентов и оптимизируя работу медицинского персонала. [7]

На рисунке 1 отображена модель бизнес-процесса «как есть».

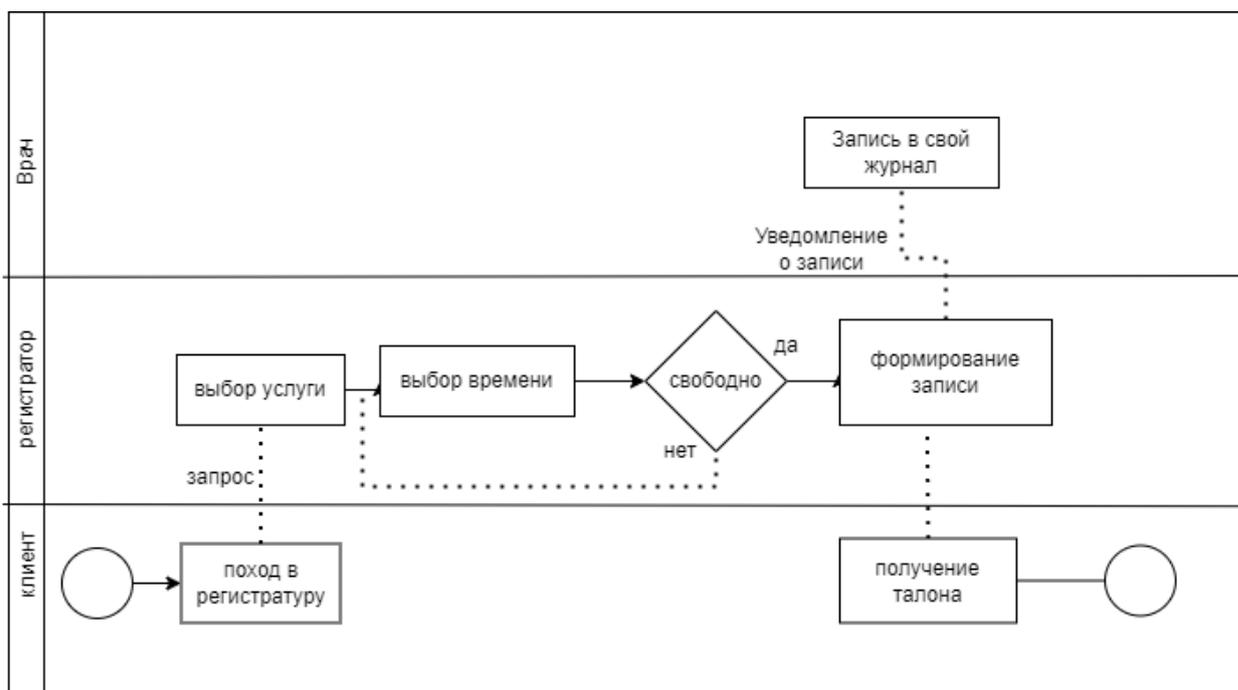


Рисунок 1 — Модель «как есть» в нотации BPMN

Анализ модели бизнес-процесса «как есть» для системы онлайн записи на прием в больницу выявляет основные неэффективности и ограничения существующих процедур записи, которые могут включать длительное ожидание на телефонной линии, ограниченный доступ к информации о доступности врачей и услуг, а также риск ошибок при вводе данных. Эти и другие проблемы подчеркивают необходимость перехода к автоматизированному варианту решения.

1.3 Разработка модели бизнес-процесса «как должно быть»

Разработка модели бизнес-процесса «как должно быть» для системы онлайн записи на прием в больницу предполагает создание оптимизированного, автоматизированного процесса, ориентированного на максимальное удобство для пациентов и эффективность работы медицинского

персонала. Целью является минимизация времени ожидания для пациентов и сокращение нагрузки на административный персонал, а также обеспечение бесперебойного доступа к медицинским услугам.

На рисунке 2 отображена модель бизнес-процесса «как должно быть».

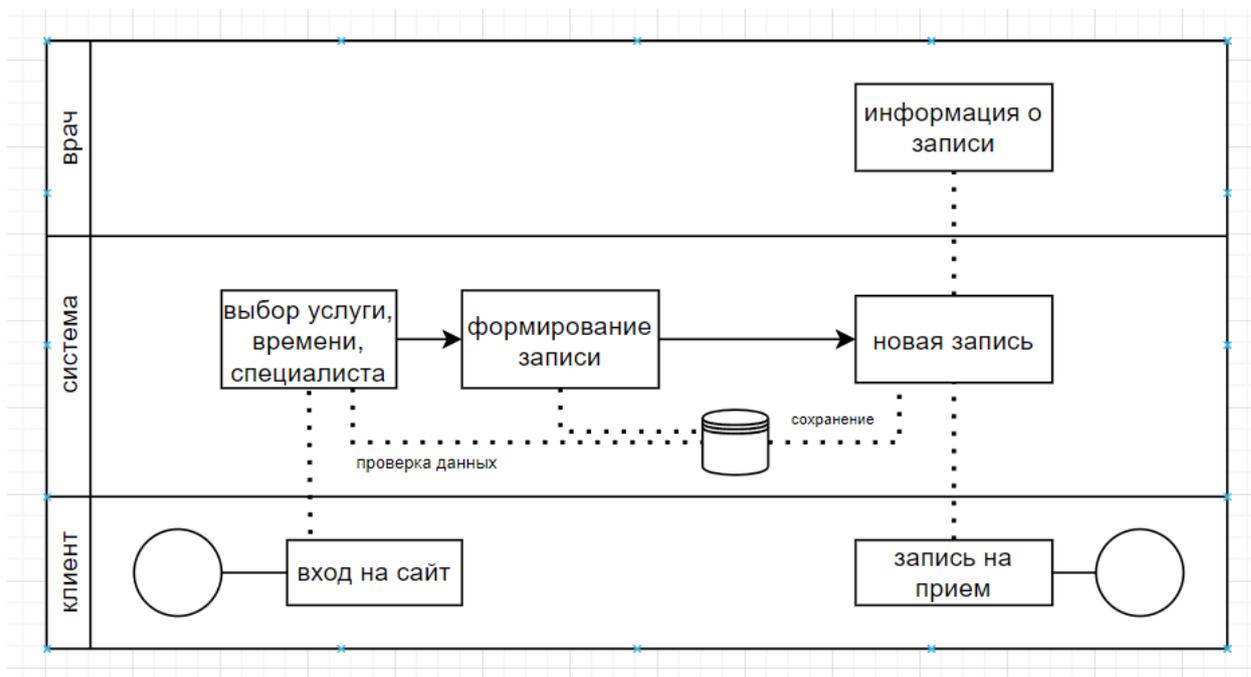


Рисунок 2 — Модель «как должно быть» в нотации BPMN

Модель «Как должно быть»:

- Предварительная регистрация пациента: пациенты могут зарегистрироваться в системе заранее через веб-сайт или мобильное приложение, указав свои персональные данные.
- Автоматическое напоминание записи на прием.
- Онлайн запись на прием: пациенты могут выбирать специалиста, услугу и удобное время, используя интерактивный календарь доступности врачей.
- Автоматическое подтверждение и напоминание: после записи на прием система автоматически отправляет подтверждение и напоминания о предстоящем визите.

- Онлайн отмена или перенос записи: пациенты имеют возможность самостоятельно отменять или переносить запись на прием.
- Электронная очередь в день приема: для минимизации времени ожидания в больнице внедряется система электронной очереди.
- Обратная связь от пациентов: после визита к врачу пациенты могут оставлять отзывы о качестве обслуживания, что позволяет учреждению совершенствовать сервис.

Исходя из проведенного анализа, можно сделать вывод что за счет внедрения автоматизированной системы можно исключить ряд проблем таких как долгое время ожидания, большую нагрузку на персонал, вероятность ошибок в записях и расписании.

1.4 Анализ существующих разработок на предмет соответствия сформулированным требованиям

В современном мире с быстро развивающимися технологиями, в больницах стало появляться все больше и больше разнообразных информационных систем, позволяющих сильно упростить многие рабочие процессы.

«Развитие информационных технологий и современных коммуникаций, появление в клиниках большого количества автоматизированных медицинских приборов, следящих систем и отдельных компьютеров привели к новому витку интереса и к значительному росту числа медицинских информационных систем.» [22]

Как утверждают «В России довольно интенсивно развиваются локальные медицинские информационные системы и сети. В настоящее время широко применяются в практике медицины компьютеризированные истории болезни и системы классификации терминов. При этом важную роль играет язык общения между базами данных и терминология.» [2]

Автоматизированные информационные системы (АИС) для больниц — это комплексные программные решения, направленные на автоматизацию медицинских и административных процессов в медицинских учреждениях. Они включают в себя управление медицинскими записями пациентов, учет медицинских услуг, планирование работы персонала, управление запасами медицинских препаратов и материалов, а также обеспечение взаимодействия между различными отделениями больницы. [5]

При анализе программ для больниц важно установить четкие критерии, чтобы обеспечить объективную и всестороннюю оценку. Критерии могут включать технические характеристики, функциональные возможности, удобство использования, интеграцию с другими системами, безопасность данных и стоимость.

Примеры программных продуктов:

- 1С:Здравоохранение;
- МИС Парацельс-А.

1С:Здравоохранение является ведущей программной платформой в России, разработанной для обеспечения комплексного управления всем спектром процессов в медицинских учреждениях различных форм собственности и профилей деятельности. Эта система представляет собой мощное решение, способное автоматизировать основные аспекты работы медицинских организаций, включая учет пациентов, планирование работы врачей, ведение медицинских карт, управление финансами и многое другое.

Главной целью 1С:Здравоохранение является повышение эффективности медицинского и административного персонала, а также улучшение качества предоставляемых услуг за счет оптимизации внутренних процессов и ресурсов.

Программное обеспечение обладает высокой степенью адаптивности и может быть настроено в соответствии с уникальными требованиями и спецификой каждого конкретного медицинского учреждения. Это достигается благодаря модульной структуре системы, позволяющей интегрировать новые

функциональные блоки без необходимости полной перестройки уже используемых процессов. Такая гибкость обеспечивает не только эффективное внедрение системы в любые медицинские организации, но и позволяет быстро адаптироваться к изменяющимся нормативным и законодательным требованиям в сфере здравоохранения.

Безопасность и конфиденциальность данных являются приоритетными направлениями в разработке и эксплуатации 1С:Здравоохранение. Система обеспечивает надежное хранение и защиту персональной информации пациентов, а также врачебной тайны, соответствуя текущим стандартам и требованиям по безопасности информации. Это достигается за счет использования современных технологий шифрования и аутентификации, а также предоставления механизмов контроля доступа и аудита действий пользователей системы.

Важной особенностью 1С:Здравоохранение является ее способность интегрироваться с другими медицинскими и информационными системами, что позволяет создавать единую информационную среду внутри медицинского учреждения и обеспечивать бесперебойный обмен данными между различными отделениями и службами. Такая интеграция способствует улучшению координации работы медицинского персонала и повышает общую эффективность лечебного и диагностического процессов.

За годы успешного применения на практике 1С:Здравоохранение зарекомендовало себя как надежное и эффективное решение, способное удовлетворить потребности самых требовательных медицинских организаций, стремящихся к внедрению передовых информационных технологий в сфере здравоохранения.

МИС Парацельс-А представляет собой продвинутую медицинскую информационную систему, специально разработанную для обеспечения полной автоматизации рабочих процессов в медицинских учреждениях разнообразного профиля и масштаба. Эта система охватывает широкий спектр функциональных возможностей, начиная от управления электронными

медицинскими записями пациентов и заканчивая финансовым учетом и аналитикой деятельности учреждения.

Парацельс-А предназначена для того, чтобы сделать процессы управления медицинскими данными более эффективными, повысить качество обслуживания пациентов и оптимизировать работу медицинского персонала. Основываясь на современных информационных технологиях, МИС Парацельс-А обладает гибкой архитектурой, позволяющей настраивать систему с учетом уникальных потребностей каждого конкретного медицинского учреждения. Внедрение этой системы позволяет значительно ускорить доступ к медицинским данным, обеспечить их централизованное хранение и защиту, а также автоматизировать рутинные задачи, связанные с документооборотом и учетом.

Одним из ключевых преимуществ Парацельс-А является ее способность к интеграции с различными медицинскими информационными системами и оборудованием, что обеспечивает обмен данными в реальном времени и повышает эффективность диагностических и лечебных процессов. Система поддерживает стандарты медицинской отчетности и обмена данными, что делает возможным взаимодействие с другими медицинскими и государственными информационными системами. Безопасность пациентских данных в Парацельс-А находится на высоком уровне благодаря применению современных методов шифрования и аутентификации пользователей.

Парацельс-А также предлагает разнообразные аналитические инструменты для мониторинга и анализа деятельности медицинского учреждения, включая отчеты о качестве оказанных услуг, финансовые отчеты, анализ загруженности персонала и многое другое. Эти инструменты позволяют руководству медицинских организаций принимать обоснованные управленческие решения, направленные на оптимизацию работы учреждения и повышение уровня медицинского обслуживания.

Сравнение этих программных продуктов показано в таблице 1.

Таблица 1 – Сравнение программных продуктов

Критерий	1С:Здравоохранение	МИС Парацельс	Мой проект
Функциональность и возможности	Высокая	Высокая	Среднее
Удобство использования	Среднее	Низкое	Высокое
Интеграция	Хорошая	Средняя	Высокая
Масштабируемость	Хорошая	Хорошая	Высокая
Техническая поддержка и обслуживание	Отличная	Хорошая	Хорошая
Стоимость	Высокая	Высокая	Низкая
Скорость разработки и внедрения	Низкая	Низкая	Высокая

Выбор конкретной АИС для больницы должен базироваться на потребностях учреждения, его размере, специализации, а также на финансовых и технических возможностях. Важным аспектом является возможность интеграции новой системы с уже используемым программным обеспечением, так как на данный момент большое количество медицинских заведений и так уже имеет некоторые свои наработки в сфере автоматизации процессов предоставления услуг или любых других различных сферах информационных технологий, а также соответствие требованиям законодательства в области защиты данных и медицинского обслуживания.

Из приведенного в таблице 1 сравнения можно сделать вывод о том, что новая система имеет ряд существенных преимуществ перед ее конкурентами. Немаловажным фактом в пользу выбора данной системы является высокая скорость разработки и внедрения, а также низкая цена.

1.5 Постановка задачи на разработку проекта создания/внедрения АИС

В ходе анализа деятельности медицинского учреждения были выявлены существенные проблемы в организации работы с записью клиентов на прием. Основными проблемами являются:

- Время ожидания. При текущих условиях записи на прием время ожидания клиента непростительно большое. При автоматизации данного процесса записи может быть сокращено время ожидания.
- Ошибки при ручном вводе данных могут быть минимизированы с помощью электронной системы записи, т.к. сотрудникам не придется заниматься ручным вводом данных в журналы.
- Неудобство в использовании текущих методов записи для некоторых категорий пациентов, особенно для людей с ОВЗ или для тех, кто предпочитает цифровые каналы связи.
- Недостаточная информация о доступных временных слотах и услугах врача, доступная пациентам при текущем методе записи.

После анализа проблем, связанных с записью на прием в больницу, и установления цели создания новой информационной системы, были определены конкретные задачи, реализация которых может значительно улучшить процессы медицинского учреждения.

Выводы по разделу 1

Анализ существующей системы записи на прием в медицинские учреждения выявил ряд проблем, включая длительное ожидание на линии при телефонной записи, ограниченный доступ к информации о расписании врачей и сложности в управлении личным календарем приемов. Разработка модели бизнес-процесса «как должно быть» направлена на решение этих проблем за счет автоматизации и оптимизации данных процессов.

Новая модель предполагает использование информационных технологий для упрощения и ускорения процесса записи на прием, повышения уровня удовлетворенности пациентов и эффективности работы медицинского персонала. Внедрение предложенной системы онлайн записи позволит медицинским учреждениям существенно улучшить качество обслуживания и доступность медицинских услуг, а также обеспечить лучшее управление ресурсами и процессами внутри организации.

2 Проектирование системы онлайн записи на прием в городскую больницу

2.1 Логическое моделирование ИС и ее архитектура

Логическая модель АИС представляет собой абстрактное описание системы, которое фокусируется на функциональности, данных и взаимодействиях между компонентами системы, не привязываясь к конкретным технологиям реализации.

Логическая модель позволяет анализировать и документировать требования к системе, определять основные процессы, объекты и их взаимосвязи, что является основой для дальнейшего физического проектирования и разработки АИС. [14]

Основные элементы логической модели АИС:

- Сущности (объекты): представляют основные данные, с которыми работает система (например, пациенты, врачи, медицинские услуги).
- Атрибуты: характеристики или свойства сущностей (например, ФИО пациента, специализация врача).
- Функции (процессы): описывают действия или операции, которые выполняются в системе (например, запись на прием, выдача направлений на анализы).
- Взаимосвязи: определяют логические связи между сущностями (например, пациент имеет запись на прием к врачу).

Логическая модель АИС включает:

- Модель данных: описывает структуру данных в системе, включая сущности и их атрибуты, а также связи между этими сущностями. Например, в медицинской информационной системе важно четко определить структуру данных о пациентах, медицинских услугах, расписании врачей и медицинских записях.

- Модель процессов: детализирует ключевые бизнес-процессы системы, их последовательность и взаимодействие. В контексте АИС это может быть процесс записи на прием к врачу, процесс ведения медицинской карты пациента или процесс выставления счетов за медицинские услуги.
- Модель взаимодействий: описывает, как пользователи и другие системы взаимодействуют с АИС. Это включает в себя интерфейсы пользователя, интеграцию с другими информационными системами и обмен данными.

Создание логической модели является критическим этапом в процессе разработки АИС, поскольку она помогает выявить и уточнить требования к системе до начала ее физической реализации, минимизируя риски ошибок и недопонимания на последующих этапах проекта. [21]

2.1.1 Логическая модель информационной системы и её описание

Для построения логической модели воспользуемся UML схемой. UML – это способ графического описания объектных моделей.

«Диаграмма вариантов использования в UML — отражает отношения между актерами и прецедентами и является частью модели прецедентов, позволяющей описать систему на концептуальном уровне». [15]

На рисунке 3 представлена диаграмма вариантов использования.

Документирование элементов (Рисунок 3):

- Клиент – Пациент медицинского учреждения.
- Врач – Сотрудник больницы, занимающийся лечением пациентов.
- Администратор – Сотрудник учреждения, занимающийся управлением системой. Это может быть, как отдельный сотрудник, так и работник регистратуры.

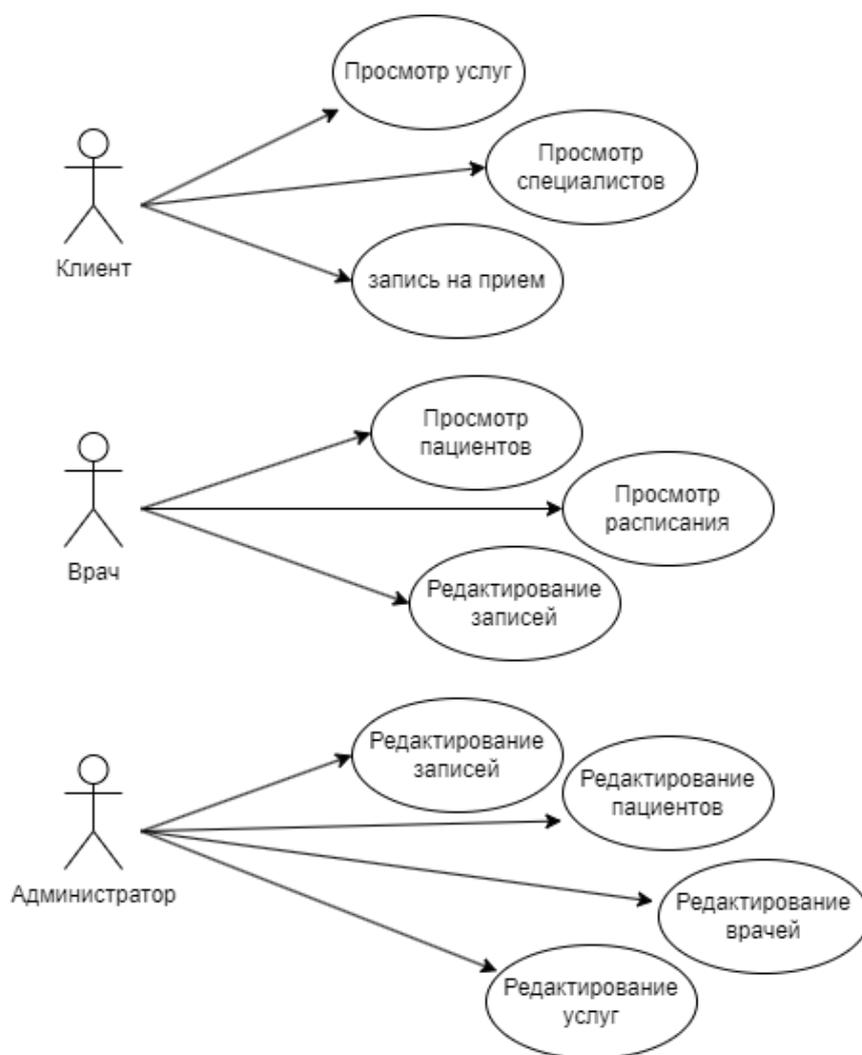


Рисунок 3 – Диаграмма вариантов использования

Выделены следующие варианты использования:

- Просмотр услуг - прецедент, где пользователь системы просматривает перечень услуг больницы.
- Просмотр специалистов - прецедент, где пользователь просматривает перечень врачей.
- Запись на прием - прецедент, где пациент записывается на посещение больницы к необходимому специалисту.
- Просмотр пациентов - прецедент, где врач или администратор просматривает список пациентов.

- Просмотр расписания - прецедент, где врач просматривает список записанных на прием пациентов.
- Редактирование записей- прецедент, где врач или администратор изменяет данные каких-либо записей в системе.
- Редактирование пациентов - прецедент, где администратор изменяет данные пациентов в системе.
- Редактирование врачей - прецедент, где администратор изменяет данные сотрудников.
- Редактирование услуг - прецедент, где администратор вносит изменения в список предоставляемых услуг.

«Диаграмма классов - это структурная диаграмма языка моделирования UML, которая показывает общую структуру иерархии классов системы, их взаимодействия, атрибуты (поля), методы, интерфейсы и взаимосвязи (отношения) между ними. Она широко используется не только для документирования и визуализации, но и для построения с помощью прямого или обратного инжиниринга» [11]

На рисунке 4 представлена диаграмма классов.

Документирование диаграммы классов (рисунок 4):

- Класс клиент – содержит информацию о посетители больницы. Он имеет уникальный идентификатор, ФИО, уровень доступа. А также набор методов авторизации, регистрации, выбора услуги, специалиста, времени.
- Класс Врач – содержит информацию о сотруднике учреждения. В него входит уникальный идентификатор, ФИО, уровень доступа, и методы просмотра пациентов, просмотра расписания, редактирование пациентов и расписания, просмотр перечня услуг.
- Класс Запись – хранит в себе информацию о записях на прием и содержит в себе уникальный идентификатор, id врача, id клиента, время записи, id услуги, а также методы создания, удаления и редактирования.

- Класс услуга – содержит уникальный идентификатор, наименование услуги, цену этой услуги и ее описание, так же содержит функции создания, удаления, изменения.
- Класс Администратор – включает в себя уникальный идентификатор, ФИО, уровень доступа, методы просмотра пациентов, врачей, услуг, методы редактирования врачей, услуг, пациентов.

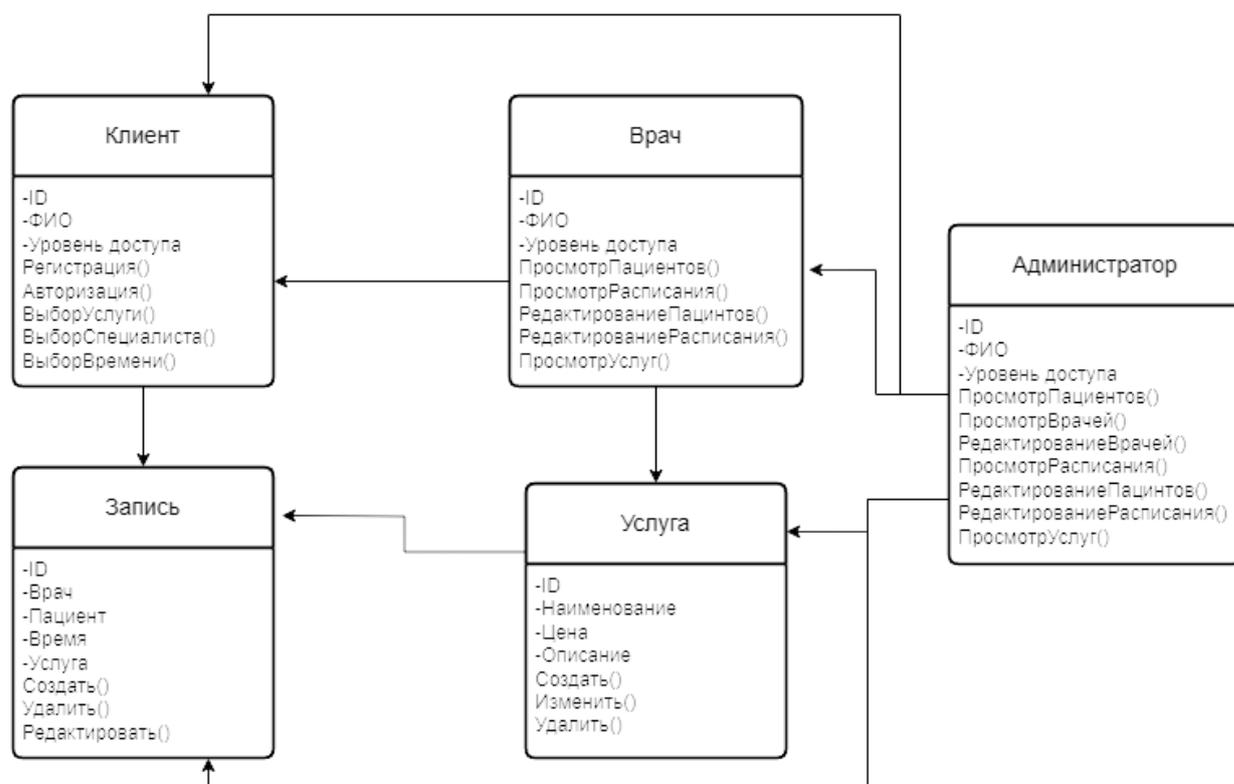


Рисунок 4 – Диаграмма классов

В ходе работы над информационной системой, основанной на анализе процессов медицинского учреждения, были выявлены ключевые характеристики и особенности функционирования. Анализ процессов позволил выделить различные аспекты, определяющие потенциальные варианты решений.

2.1.2 Диаграмма взаимосвязи программных модулей и информационных файлов

На рисунке 5 представлена структурная схема информационной системы. В первую очередь представленная схема служит наглядным примером для понимания основных взаимосвязей между частями системы, а также для понимания логики взаимодействия между различными узлами. По сути она описывает общую структуру приложения. В дальнейшем мы сможем опираться на нее при создании ИС.

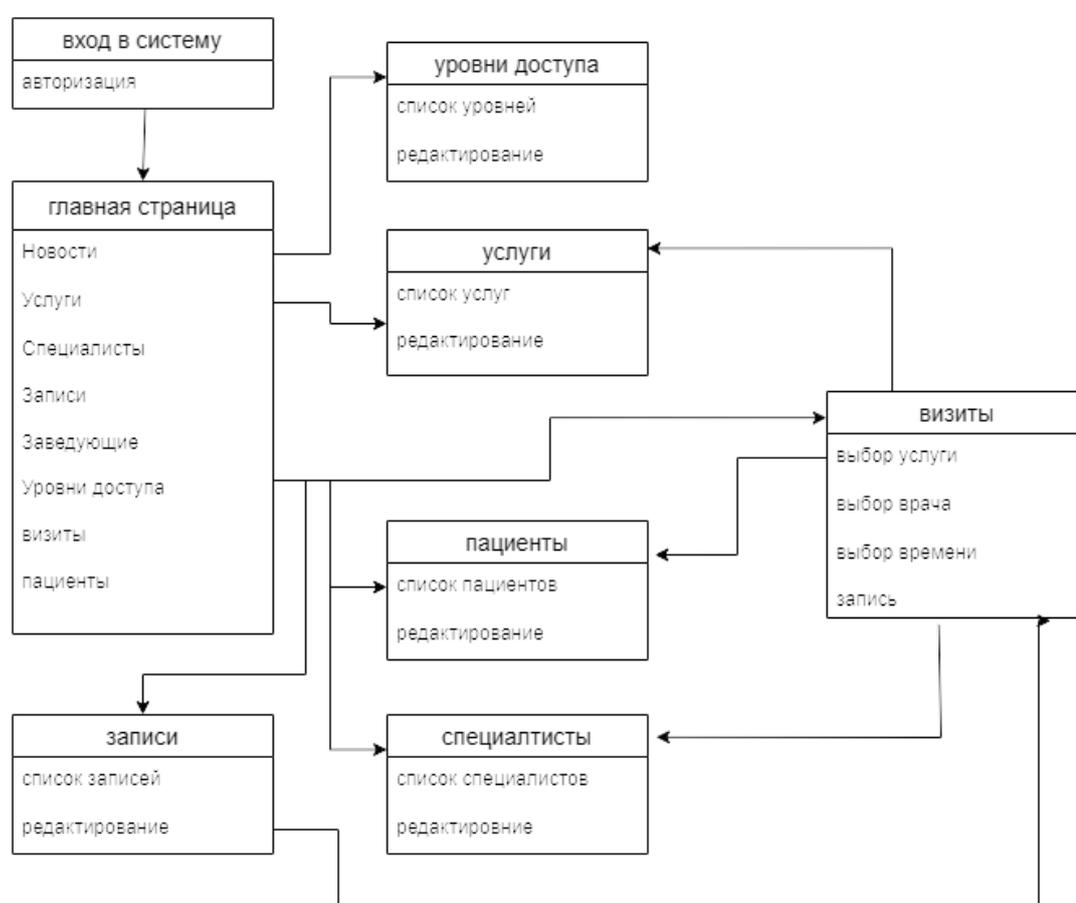


Рисунок 5 – Структурная схема информационной системы

«Диаграмма последовательности - это UML-диаграмма, которая показывает жизненный цикл объекта (создание-действие-уничтожение некоторой сущности) и взаимодействие актеров (действующих лиц)

информационной системы в рамках прецедента для некоторого набора объектов на одной оси времени». [6]

На рисунке 6 изображена диаграмма последовательности.

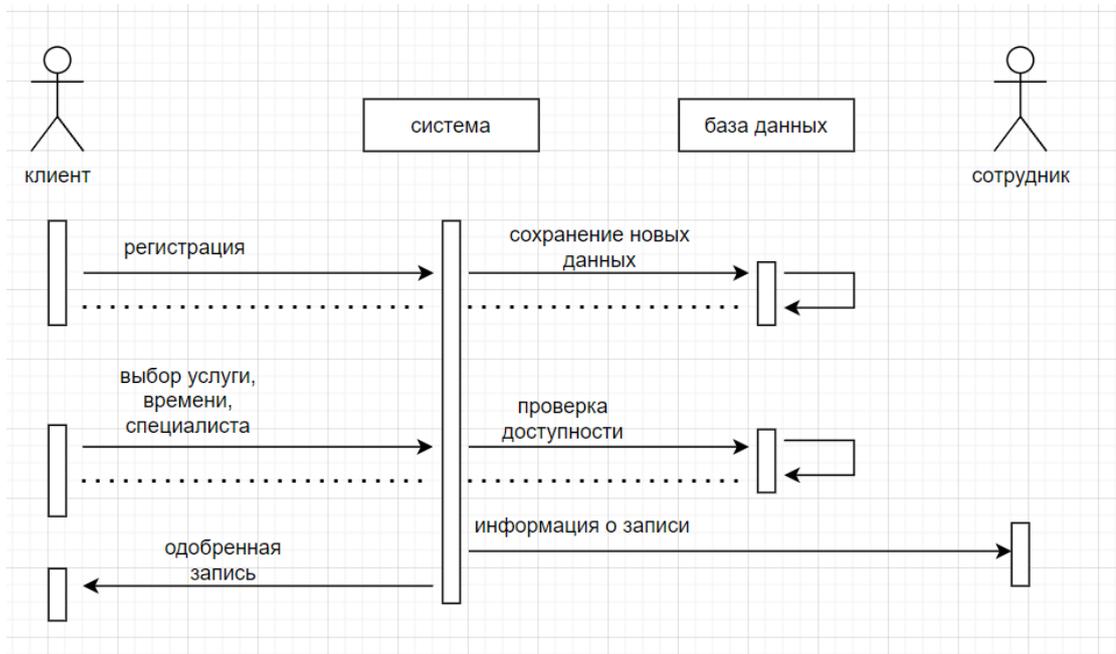


Рисунок 6 – Диаграмма последовательности

Представленная структурная схема (рисунок 6) необходима для понимания логики функционирования информационной системы, а также эта схема является основой для её создания. Данная диаграмма подробно и наглядно показывает, как выполняются операции в хронологической последовательности, то есть в одной временной линии.

2.2 Описание архитектуры АИС

По сути любой Веб-сайт – это определенный набор страниц. Эти html страницы подчиняясь определенной логике, заданной на этапе разработки, начинают взаимодействовать между собой в рамках нашего сайта.

«Сайт – это информационная единица в интернете, ресурс из веб-страниц (документов), которые объединены общей темой и связаны между друг с другом с помощью ссылок. Он зарегистрирован на одно юридическое или физическое лицо и обязательно привязан к конкретному домену, являющемуся его адресом. Сайт может состоять как из одной страницы, так и огромного количества. Каждая страница ресурса – это текстовый файл или их набор, написанный на специальном языке разметки или программирования (HTML, PHP, CSS и пр.).» [1]

Веб-сайт в свою очередь может запускаться сервере, для чего ему скорее всего потребуется доменное имя, или же просто в локальной сети.

Пользователи сайта разграничиваются между собой разным набором возможностей взаимодействия с системой. Это разграничение строится на системе уровней доступа. В нашем случае в БД есть следующие уровни:

- Администратор (он может редактировать, удалять, создавать абсолютно любые компоненты сайта, то есть обладает максимально широким спектром возможностей, максимальным функционалом и соответственно это самый высокий уровень доступа);
- Клиент – самый низкий уровень в возможности которого входит просмотр страниц и возможность взаимодействия только со своими личными данными (это минимально необходимый функционал для пользования сайтом);
- Менеджер – средний уровень доступа (обладает средним функционалом, который включает в себя управление некоторыми компонентами).

Состав веб-приложения приведен на рисунке 7.

Функции компонентов программного обеспечения могут варьироваться в зависимости от конкретных компонентов и задач, которые они выполняют.

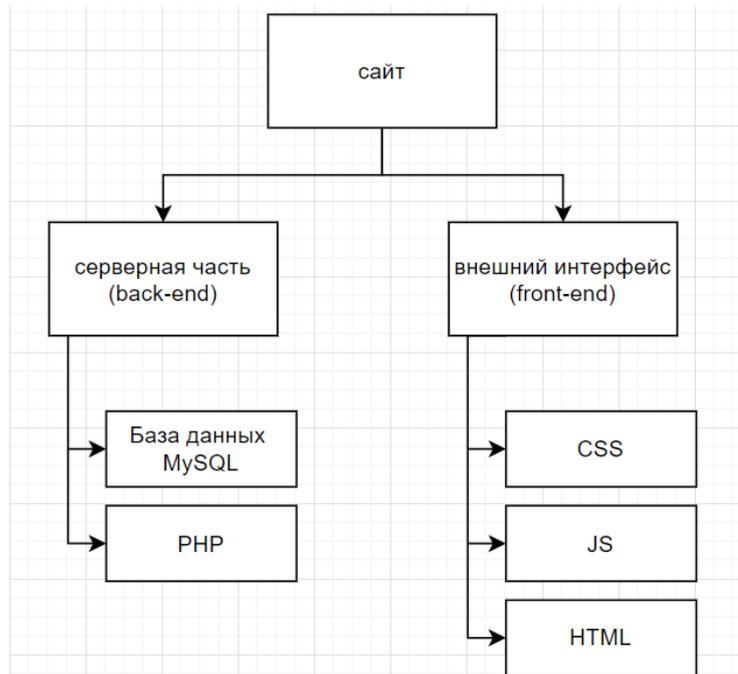


Рисунок 7 – Взаимодействие компонентов системы

«База данных - это набор связанных (не всегда) между собой таблиц. С технической точки зрения, базой данных может быть обычный каталог на диске, в котором размещены файлы таблиц, или же один большой двоичный файл, содержащий всю информацию, - тут все зависит от реализации формата базы данных. Например, в MySQL базой данных является каталог с файлами таблиц.» [9]

База данных используется для:

- хранение и управление данными;
- обеспечение целостности и безопасности данных;
- предоставление возможности поиска и манипулирования данными.

Перед началом разработки любого ПО желательно тщательно продумать структуру и организацию данных, которые будут храниться в базе, и разработать соответствующую схему базы данных, что позволит оптимизировать работу с большим объемом данных.

Серверное ПО отвечает за:

- определение протоколов взаимодействия и маршрутизацию запросов;
- управление динамическими и статическими контентом;
- обработка пользовательских запросов и генерация ответов сервера.

Серверная часть включает в себя код, который обрабатывает запросы от клиентской части и генерирует HTML-страницы, которые затем отправляются обратно на клиентскую сторону для отображения в браузере. В целом, можно сказать, что серверная часть сайта выполняет собой роль поставщика контента для клиентской части и обеспечивает более сложную логику и функциональность, чем-то, что может быть выполнено на стороне клиента.

[19]

Фронтенд (пользовательский интерфейс и связанные с ним компоненты)

[10]:

- взаимодействие с конечным пользователем, например, с посетителем сайта, с помощью отображения интерфейса в каком-то браузере;
- обработка различных действия пользователя в браузере;
- отправка различных запросов на сервер.

В нашем случае для фронтенд части разработки будут использованы: CSS, JS, HTML.

2.3 Выбор технологии разработки программного обеспечения АИС

Для создания веб-страниц используются три основных компонента: HTML, CSS и JavaScript. HTML используется для определения структуры страницы, CSS позволяет задавать ее внешний вид, а JavaScript отвечает за программные возможности, такие как анимация и взаимодействие с пользователем.

JavaScript – это изначально скриптовый язык программирования. При создании этого языка, большинство браузеров не требовало особого доступа к

памяти, поэтому он и сейчас не предоставляет низкоуровневый доступ к памяти.

«Возможности JS очень сильно зависят от окружения, в котором он применяется.» [20] Фреймворки и библиотеки, такие как jQuery, React и Angular, упрощают процесс написания кода и обеспечивают более быструю разработку приложений [13]. Кроме того, необходимо применять серверный язык программирования. Он является дополнительным компонентом, позволяя создавать динамические сайты.

На данный момент Python - один из самых популярных языков для создания веб-сайтов [3]. К основным преимуществам данного языка можно отнести:

- Простота и легкость в использовании делают этот язык идеальным решением для начинающих программистов.
- Множество библиотек, фреймворков и библиотек позволяют создавать мощные приложения для веб-сайтов.
- Универсальность: Python может быть использован как для создания простых веб-сайтов, так и сложных, и его можно использовать в других областях.
- Сообщество разработчиков имеет большое количество программистов, а также множество сайтов, которые предоставляют помощь и поддержку.

Однако Python имеет и некоторые недостатки:

- Скорость может быть ниже, чем у его конкурентов, а это в свою очередь может привести к задержке в работе веб-сайта.
- Для достижения максимального уровня производительности кода требуется оптимизация, что достаточно часто является трудной задачей для программистов.

Некоторые платформы могут ограничивать использование Python, что в итоге может привести к ограничению возможностей веб-разработок.

Для веб-разработчиков Python является неплохим выбором благодаря перечисленным выше достоинствам, но наличие некоторых недостатков могут стать веской причиной для выбора иных языков разработки.

PHP еще один популярный язык программирования, отличительной особенностью которого является его синтаксис и зачастую очень тесное вплетение кода в язык разметки HTML.

«Язык программирования PHP – это серверный язык, при помощи которого можно создавать веб-сайты, причем как небольшие лендинги, состоящие из одной страницы, так и гигантские системы, использующие сотни и тысячи серверов. Электронная энциклопедия Wikipedia, социальные сети Facebook, ВКонтакте, созданы с использованием PHP. Будучи одним из самых первых языков программирования, ориентированных на веб-разработку, PHP прошел длительный путь практически с самого начала зарождения Web. Поэтому в мире он остается одним из самых популярных и востребованных языков». [12]

Основные достоинства языка PHP:

- Большое количество разнообразных расширений языка: PHP поддерживает огромное множество всевозможных расширений, которые в свою очередь добавляют различные дополнительные функции для разрабатываемых веб-приложений.
- Широкое распространение: на данный момент PHP — практически один из самых популярных языков серверного программирования для веб-разработки. Это и обеспечивает ему доступ к колоссальному количеству разнообразных онлайн-ресурсов и большому сообществу опытных программистов и разработчиков.
- Множество фреймворков: этот язык имеет внушительное множество всевозможных фреймворков, которые по мимо того, что предоставляют полезные инструменты, так еще и сильно упрощают процесс написания кода для разработчика.

- Кроссплатформенность: это еще один весомый аргумент в пользу PHP так как поддержка всех основных операционных систем является большим конкурентным плюсом. Благодаря этому, программисты могут писать сайты на Windows или на MacOS или же запускать их на Linux-серверах. Сложности переноса при этом будут минимальными.
- Простота: Синтаксис PHP прост и легок в использовании, что зачастую снижает порог входа в коддинг и упрощает начало программирования. Это будет особенно полезно для новичков в этой сфере.

К недостаткам языка можно отнести следующие пункты:

- Безопасность: к сожалению, PHP иногда может быть уязвимым для различного рода атак. В данном случае недостаток вытекает из достоинства, а именно: из-за своей простоты PHP может принести большие проблемы, связанные с угрозой атак на сайт.
- Необходимость оптимизации: как и любой другой язык PHP требует оптимизации кода для достижения приемлемой для поставленной задачи производительности. И здесь снова минус вытекает из плюса: из-за своих особенностей PHP зачастую очень сильно затрудняет процесс оптимизации и отладки кода.

В целом можно сделать вывод о том, что PHP является более чем достойным выбором для разработки веб-приложений любой сложности. У него много плюсов которые перекрывают наличие определенных недостатков и обуславливают выбор в его сторону.

Для работы выбирается PHP, т.к. он позволяет создавать приложения, которые можно охарактеризовать такими понятиями: поддержка различных платформ; функциональность; высокая скорость разработки; наличие библиотек для работы с базами данных.

2.4 Проектирование базы данных АИС

Проектирование базы данных для АИС является очень важным этапом в разработке системы, поскольку именно от этого шага зависит, как данные будут использоваться, обрабатываться и предоставляться пользователям. «Эффективно разработанная база данных позволяет повысить производительность, надёжность и безопасность системы», а в дальнейшем упростить программистам возможную необходимость оптимизировать и адаптировать систему под новые бизнес-процессы заказчиков с минимальными затратами. [25]

Далее можно определить следующие этапы разработки БД:

- определение требований – по сути это анализ требований к системе для понимания, какие данные будут собираться, обрабатываться и храниться. Включает определение потребностей всех заинтересованных сторон, в том числе конечных пользователей и администраторов системы;
- концептуальное проектирование – создание высокоуровневой модели данных, которая описывает ключевые сущности, их атрибуты и взаимосвязи, без привязки к конкретной СУБД. Результатом этого этапа является ER-диаграмма (Entity-Relationship diagram), представляющая структуру будущей базы данных; [17]
- логическое проектирование – трансформация концептуальной модели в логическую модель данных, уже с учетом выбранной системы управления базами данных (СУБД). На этом этапе определяются типы данных для атрибутов, устанавливаются ограничения и правила целостности данных;
- физическое проектирование – разработка внутренней структуры базы данных, включая определение способов хранения данных (таблицы), индексацию для ускорения поиска данных и настройку

параметров СУБД для оптимизации производительности и безопасности;

- реализация и тестирование – создание базы данных на основе физической модели, ввод начальных данных и проведение тестирования для проверки работоспособности, производительности и безопасности;
- оптимизация и мониторинг: после внедрения базы данных в эксплуатацию проводится мониторинг ее работы для выявления и устранения узких мест, а также регулярная оптимизация для повышения производительности.

Проектирование базы данных проводят, начиная с логической схемы (рисунок 8).

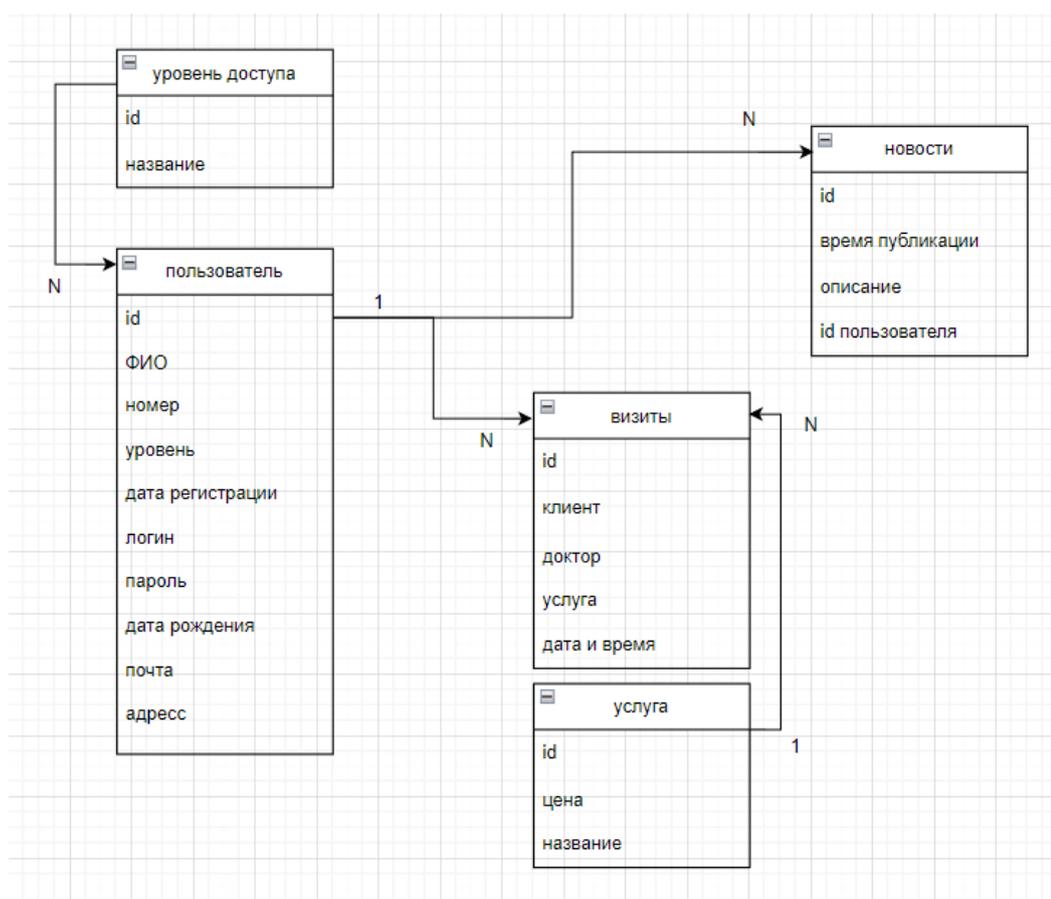


Рисунок 8 – Логическая схема БД

На рисунке видно, что БД состоит из пяти основных сущностей:

- визиты;
- пользователи;
- новости;
- услуги;
- Уровни доступа.

Если переименовать названия таблиц и полей, а также указать тип данных, то получается физическая модель БД (рисунок 9).

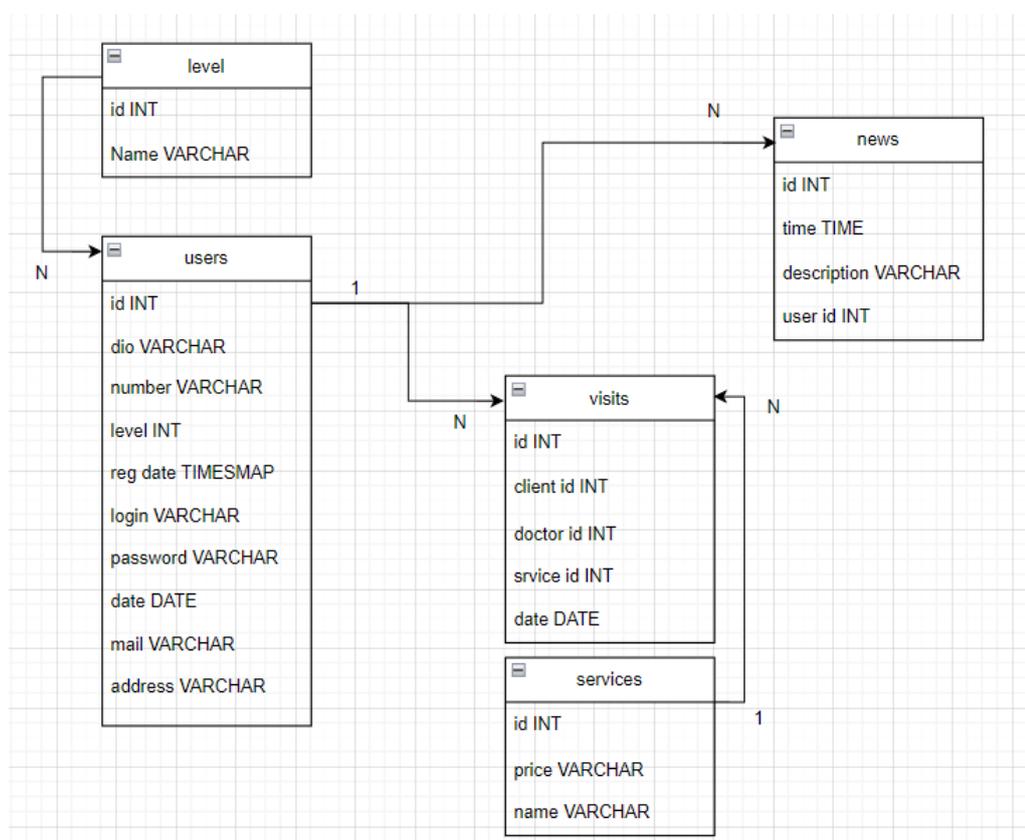


Рисунок 9 – Физическая модель БД

Надписями 1 и N обозначаются соотношение между таблицами. 1:N – это связь между таблицами под названием «Один-ко-многим», т.е. одной записи из первой таблицы могут соответствовать одна или множество записей из второй таблицы.

На рисунке 10 показана схема базы данных.

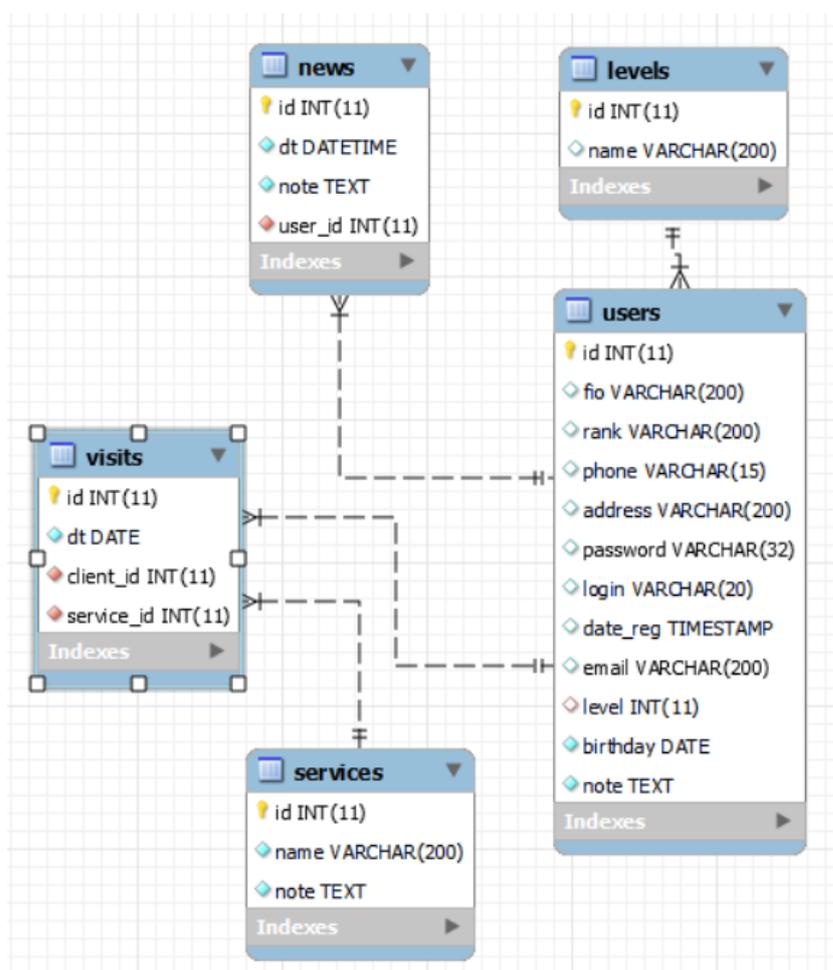


Рисунок 10 – Схема базы данных

Разработанная база (название базы данных – hospital).

База данных MySQL (список таблиц БД) приведена на рисунке 11 и включает:

- news – для хранения информации с новостями.
- levels – для хранения кодов и уровней доступа.
- users – для хранения информации о пользователях.
- visits – для хранения информации о визитах.
- services – для хранения информации об услугах.

Таблица	Действие	Строки	Тип	Сравнение
levels	★ Обзор Структура Поиск Вставить Очистить Удалить	4	InnoDB	utf8_general_ci
news	★ Обзор Структура Поиск Вставить Очистить Удалить	5	MyISAM	utf8_general_ci
services	★ Обзор Структура Поиск Вставить Очистить Удалить	6	InnoDB	utf8_general_ci
users	★ Обзор Структура Поиск Вставить Очистить Удалить	112	InnoDB	utf8_general_ci
visits	★ Обзор Структура Поиск Вставить Очистить Удалить	5	InnoDB	utf8_general_ci
5 таблиц	Всего	132	MyISAM	utf8_general_ci

Рисунок 11 – Список таблиц

На основе анализа логической схемы и физической модели базы данных, мы можем начинать проектирование настоящей реляционной базы данных.

2.5 Выбор СУБД АИС

Для создания и управления базами данных в магазине необходимо использовать систему управления базами данных (СУБД). Некоторые из популярных СУБД включают MySQL, PostgreSQL, Microsoft SQL Server, Oracle и MongoDB. «Выбор конкретной СУБД зависит от требований вашего проекта, таких как объем данных, типы запросов и доступность функций». [23]

«MySQL - это одна из самых популярных систем управления базами данных (СУБД) с открытым исходным кодом. Она широко используется для разработки веб-приложений и хранения данных». По моему мнению основными отличительными особенностями MySQL будут: [9]

- Открытый исходный код: MySQL распространяется под лицензией GPL, что позволяет бесплатно использовать, модифицировать и распространять ее.
- Производительность: это, возможно, одна из ключевых особенностей данной СУБД. Она обладает достаточно высокой производительностью и способна весьма эффективно обрабатывать большие объемы данных и большое количество запросов.

- Разнообразие хранилищ данных: эта СУБД поддерживает различные типы хранилищ, например таблицы InnoDB, MyISAM, MEMORY и другие, что позволяет выбирать наиболее подходящий тип хранилища в зависимости от требований текущего проекта.
- сообщество: у программистов, использующих MySQL, существует большое активное сообщество пользователей, что обеспечивает доступ к обширной документации, форумам поддержки и сторонним ресурсам, что в свою очередь сильно облегчает процесс разработки.
- Кроссплатформенность: MySQL доступна на различных операционных системах, включая Linux, Windows, macOS.

MySQL является мощной и гибкой СУБД, которая широко применяется в веб-разработке, электронной коммерции, аналитике данных и других областях.

PostgreSQL - это еще одна популярная и надежная система управления базами данных с открытым исходным кодом. Приведем некоторые ключевые характеристики PostgreSQL:

- PostgreSQL точно так же имеет открытый исходный код, но распространяется уже под лицензией PostgreSQL, это так же открывает большие возможности для пользователей.
- Расширяемость: PostgreSQL предлагает множество встроенных расширений, а также возможность создания пользовательских расширений, что позволяет разработчикам добавлять новую функциональность и адаптировать СУБД под свои потребности.
- Масштабируемость: PostgreSQL поддерживает горизонтальное и вертикальное масштабирование, что позволяет ей работать с большими объемами данных и справляться с высокими нагрузками на систему.
- Расширенные возможности: эта СУБД предлагает богатый набор функций, таких как транзакции, хранимые процедуры, триггеры,

представления, полнотекстовый поиск, геопространственные запросы и т.д.

В целом, PostgreSQL представляет собой хорошую СУБД с широким спектром возможностей, которая достаточно часто используется для разработки веб-приложений, управления данными и аналитики [18].

Microsoft SQL Server - это не особо популярная система управления базами данных (СУБД), разработанная компанией Microsoft. Некоторые ключевые характеристики Microsoft SQL Server:

- Интеграция с другими продуктами Microsoft: SQL Server интегрируется с другими продуктами и технологиями Microsoft, такими как Windows Server, Azure, .NET Framework, Visual Studio.
- Аналитические возможности: СУБД дает разработчикам возможности для аналитики данных, включая службы анализа данных (SSAS), службы интеграции данных (SSIS) и службы отчетов (SSRS).
- поддерживает различные языки программирования, такие как T-SQL, C#, Python, R и другие, что обеспечивает гибкость при разработке приложений и аналитических решений.
- Облачные возможности: здесь нам предоставляют возможность для интеграции с облачной платформой Microsoft Azure, которая позволяет использовать облачные вычисления [8].

SQL Server широко используется для создания и управления базами данных в корпоративных средах, веб-приложениях, системах бизнес-аналитики и других областях информационных технологий.

Oracle – это одна из наиболее часто используемых систем, которая широко используется в корпоративных средах и больших проектах. К ее особенностям можно отнести следующие пункты:

- Высокая доступность и надежность: Oracle предлагает механизмы высокой доступности, отказоустойчивости и резервирования

данных, что обеспечивает стабильную работу даже при критических ситуациях.

- Резервное копирование и восстановление: «Oracle Database идеально подходит для создания надёжных резервных копий базы данных и восстановления данных. Можно легко восстановить базу данных к нужному моменту времени. Для этого понадобится дополнительное место для хранения и архивационные механизмы» [6].
- Присутствие на рынке. «Oracle – это крупнейший поставщик СУБД, который инвестирует в исследования и разработку больше, чем его конкуренты. Что касается доли рынка СУБД, Oracle имеет наибольшую долю в сегментах серверов VMS, UNIX и OS/2. Это говорит о том, что шансы остаться без поддержки Oracle минимальны, ведь поддерживаются множество сторонних интерфейсов» [16].
- Безопасность: Oracle обеспечивает высокий уровень безопасности данных, включая механизмы аутентификации, авторизации, шифрования и аудита, что делает его привлекательным выбором для хранения чувствительной информации.

В работе выбирается MySQL, как наиболее универсальная и простая в использовании СУБД. Она может обрабатывать большие объемы данных и имеет множество инструментов для управления.

2.6 Разработка физической модели данных АИС

Структура таблицы news приведена на рисунке 12.

Таблица «Новости» содержит четыре поля:

- идентификатор;
- дата создания новости;
- текст новости;

- кем добавлена новость (по умолчанию добавляется всегда администратором).

Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
id	int(11)			Нет	Нет		AUTO_INCREMENT
dt	datetime		on update CURRENT_TIMESTAMP	Нет	CURRENT_TIMESTAMP	дата новости	ON UPDATE CURRENT_TIMESTAMP
note	text	utf8_general_ci		Нет	Нет	текст новости	
user_id	int(11)			Нет	1	кем добавлено	

Рисунок 12 – Структура таблицы news

Структура таблицы levels приведена на рисунке 13. Таблица «Уровни доступа» содержит два поля: свой уникальный идентификатор; наименование уровня.

Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
id	int(11)			Нет	Нет
name	varchar(200)	utf8_general_ci		Да	NULL

Рисунок 13 – Структура таблицы levels

Структура таблицы users приведена на рисунке 14. Таблица «Пользователи» содержит следующие поля:

- Идентификатор: Уникальный идентификатор пользователя;
- ФИО: Полное имя пользователя;
- Логин: логин для входа в систему;
- Пароль: пароль пользователя;

- Дата регистрации: Дата регистрации в системе;
- Уровень доступа;
- Электронная почта;
- Телефон: Номер телефона пользователя;
- Должность;
- Адрес: Регистрационный адрес пользователя;
- Дата рождения;
- Примечание: Дополнительная информация о пользователе.

Сервер: 127.0.0.1 » База данных: salon » Таблица: users

Обзор Структура SQL Поиск Вставить Экспорт Импорт Привилегии

Структура таблицы Связи

	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(11)			Нет	Нет		AUTO_INCREMENT
2	 fio	varchar(200)	utf8_general_ci		Да	NULL	фιο	
3	 rank	varchar(200)	utf8_general_ci		Да	NULL	должность	
4	 phone	varchar(15)	utf8_general_ci		Да	NULL		
5	 address	varchar(200)	utf8_general_ci		Да	NULL		
6	 password	varchar(32)	utf8_general_ci		Да	NULL		
7	 login	varchar(20)	utf8_general_ci		Да	NULL		
8	 date_reg	timestamp			Да	CURRENT_TIMESTAMP		
9	 email	varchar(200)	utf8_general_ci		Да	NULL		
0	 level	int(11)			Да	NULL		
1	 birthday	date			Нет	Нет		
2	 note	text	utf8_general_ci		Нет	Нет	примечание	

Рисунок 14 – Структура таблицы users

Структура таблицы visits приведена на рисунке 15. Таблица «Визиты» содержит четыре поля:

- идентификатор;
- дата и время;
- код клиента;
- код услуги.

Сервер: 127.0.0.1 » База данных: salon » Таблица: visits

Виды: Обзор Структура SQL Поиск Вставить Экспорт Импорт

Структура таблицы Связи

Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
id	int(11)			Нет	Нет		AUTO_INCREMENT
dt	date			Нет	Нет	дата и время	
client_id	int(11)			Нет	Нет	код клиента	
service_id	int(11)			Нет	Нет	код услуги	

Рисунок 15 – Структура таблицы visits

Структура таблицы services приведена на рисунке 16. Таблица «Услуги» содержит три поля: идентификатор; наименование услуги; описание услуги.

Сервер: 127.0.0.1 » База данных: salon » Таблица: services "услуги"

Виды: Обзор Структура SQL Поиск Вставить Экспорт Импорт Привлечь

Структура таблицы Связи

Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
id	int(11)			Нет	Нет		AUTO_INCREMENT
name	varchar(200)	utf8_general_ci		Нет	Нет	наименование услуги	
note	text	utf8_general_ci		Нет	Нет	описание услуги	

Рисунок 16 – Структура таблицы services

Данные услуги могут быть, например, осмотром врача либо любым другим действием.

Выводы по разделу 2

В разделе акцентируется внимание на важности тщательного выбора технологии логического моделирования, так как она является фундаментом для создания структурированной и гибкой АИС, способной адаптироваться к изменяющимся требованиям и условиям эксплуатации. Выбор между реляционной и нереляционной моделями баз данных обосновывается исходя

из специфики предметной области, объемов обрабатываемых данных и предпочтений разработчиков.

Далее описано информационное обеспечение АИС, включая использование стандартизированных классификаторов и систем кодирования, которые играют важную роль в обеспечении совместимости и обмена данными между различными подсистемами и внешними информационными ресурсами. Подробно рассматриваются типы информации, обрабатываемые в системе: нормативно-справочная, входная оперативная и выходная, каждый из которых имеет свою специфику и требования к обработке.

Ключевой частью раздела является обсуждение проектирования базы данных АИС, начиная с выбора технологии проектирования и заканчивая разработкой логической модели данных. Особое внимание уделяется обоснованию выбора реляционной модели в качестве основы для базы данных, что позволяет обеспечить высокий уровень целостности данных и удобство их обработки с использованием SQL-запросов.

В заключительной части раздела формулируются требования к аппаратно-программному обеспечению АИС, что включает необходимые характеристики серверного оборудования, требования к СУБД, к программному обеспечению для разработки, тестирования и эксплуатации системы, а также к сетевой инфраструктуре.

Таким образом, можно сказать, что успешное логическое проектирование АИС требует комплексного подхода, учитывающего как технические аспекты реализации системы, так и требования пользователей к функциональности и удобству использования. Выбор оптимальных технологий и инструментов для разработки базы данных и других компонентов АИС, а также внимание к деталям информационного обеспечения и требованиям к аппаратно-программному комплексу, лежит в основе создания эффективных и надежных информационных систем, способных служить надежным инструментом в деятельности организаций различных сфер.

3 Разработка системы онлайн записи на прием в городскую больницу и оценка ее эффективности

3.1 Разработка программного обеспечения АИС

3.1.1 Схема взаимосвязи модулей приложения АИС

Программное обеспечение представляет собой веб-приложение, которое будет выполняться на веб-сервере или в локальной сети.

Из диаграммы развертывания (рисунок 17) видно, что система состоит из трех компонентов:

- сервер;
- клиент;
- БД.

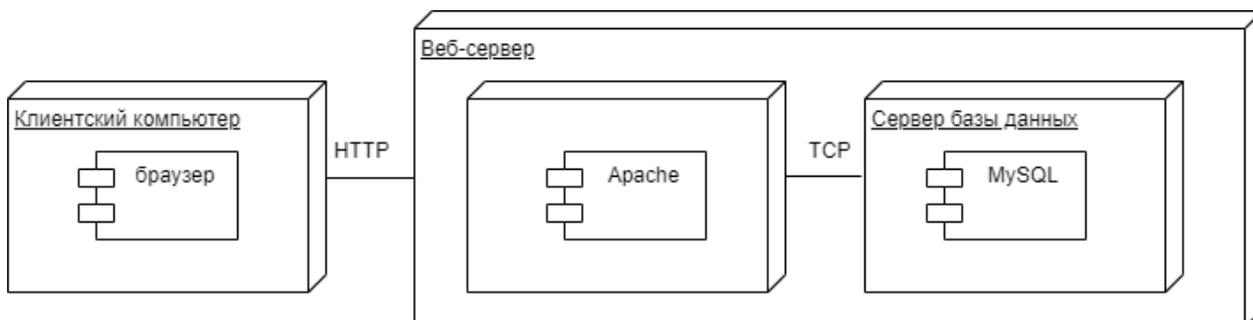


Рисунок 17 – Диаграмма развертывания

Каждая страница состоит из четырех частей:

- заголовок;
- меню;
- контент (зависит от страницы);
- подвал.

Создание новой страницы производится при процессе разработки сайта вручную: «либо на основании пустого шаблона, либо на основании

аналогичной страницы.» Например, следующие страницы имеют аналогичный код: новости; уровни доступа.

Меню рендерится в зависимости от уровня доступа пользователей сайта.

3.1.2 Описание модулей приложения АИС с примерами программного кода

Страницы, доступные администратору и являющиеся редакторами информации, построены по схожему принципу [26].

Пояснения по назначению папок и файлов веб-приложения:

- css – папка со стилями оформления;
- js – папка с javascript-файлами (библиотека jquery и некоторые дополнительные компоненты: jquery.tablesorter.js – для сортировки таблиц по столбцам; select2.min.js – удобный компонент (замена select));
- gen – папка со скриптами для заполнения БД тестовыми данными;
- images – папка с изображениями;
- .htaccess – файл с настройками веб-сервера;
- auth.php – скрипт проверяет наличие авторизации;
- backup.bat – делает бэкап БД;
- database.php – настройки БД;
- db.sql – дамп БД (сюда складывается бэкап БД);
- func.php – набор полезных используемых функций;
- index.php – главная страница;
- install_db.bat – запуск восстановления БД из бэкапа;
- levels.php – скрипт управления правами пользователей и группами пользователей;
- menu.php – скрипт формирует меню слева на каждой странице;
- scripts.php – при необходимости подключает на страницах нужные js-скрипты;
- style.css – стилевое оформление страниц;

– users.php – скрипт-редактор пользователей

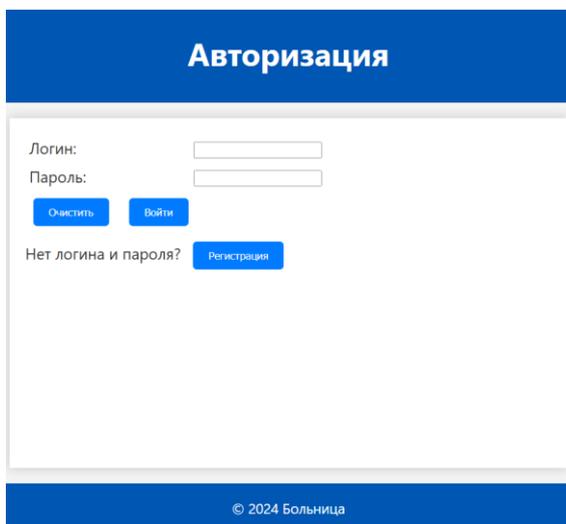
Авторизация происходит после заполнения логина и пароля на странице login.php. При авторизации в сессии (переменная \$_SESSION – тип массив) сохраняются логин, код уровня доступа (группа пользователей) и id пользователя.

3.2 Описание функциональности АИС

Функциональное тестирование направлено на проверку того, все ли возможности и функции сайта работают как было задумано. Тестирование включает в себя несколько параметров, таких как проверка интерфейса, API, базы данных, тестирование сервера. [24]

Функциональное тестирование заключалось в тестировании работоспособности веб-страниц данного веб-приложения. Результаты можно увидеть ниже на скриншотах экрана.

Страница авторизации приведена на рисунке 18.



The image shows a web form titled "Авторизация" (Authorization) on a blue background. The form contains two input fields: "Логин:" (Login) and "Пароль:" (Password). Below the password field are two buttons: "Очистить" (Clear) and "Войти" (Login). Below these buttons is a link "Нет логина и пароля?" (No login and password?) with a "Регистрация" (Registration) button next to it. At the bottom of the page, there is a footer with the text "© 2024 Больница" (© 2024 Hospital).

Рисунок 18 – Страница авторизации

Страница регистрации приведена на рисунке 19.

Регистрация

ФИО: Иванов Иван Иванович

Логин: login

Пароль: password

Email: example@domain.com

Очистить поля Зарегистрироваться Авторизация

© 2024 Больница

Рисунок 19 – Страница регистрации

Регистрация простая. Пользователь должен внести минимум информации о себе:

- ФИО;
- логин;
- пароль;
- электронная почта.

Для регистрации нет необходимости в получении подтверждающего кода на электронную почту или СМС на телефон. С целью улучшения надежности, конечно, необходимо рассматривать данные функции как необходимость. Однако в рамках учебной работы отправка СМС затруднена, а отправка кода с локального веб-сервера на внешние электронные адреса невозможна.

Для авторизации с правами администратора можно использовать логин admin и пароль pass10. Логины заведующих для тестов: manager1, manager2, manager3. Пароль заведующих: pass5; пароль клиента: pass1. Логины врачей для тестов: doctor1, doctor2. Пароли врачей: pass2. Логины пациентов для тестов: patient1, patient2. Пароли пациентов: pass1. Логины и пароли администраторов, заведующих, врачей, пациентов можно увидеть в базе

данных. Пароли хранятся в открытом виде для удобства тестирования. В дальнейшем при использовании программы необходимо скрывать пароли и хранить их в виде хэшей.

На рисунке 20 представлено главное меню программы.

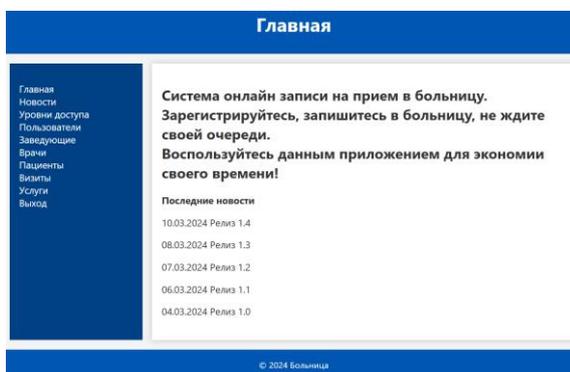


Рисунок 20 – Главное меню

На странице Пользователи (рисунок 21) можно управлять пользователями (удалять, добавлять, редактировать).

Пользователи									
ФИО	Дата рождения	Электронная почта	Уровень доступа	Дата регистрации	Логин	Пароль	Удл.	Ред.	
Осенний Павлид Архипович	1943-01-02	osenniy.palladiy8902@microsoft.com	Администраторы	2024-03-10 17:25:44	admin	pass10	✗	✎	
Бараненский Велizar Прохорович	1956-12-30	baranenskiyvelizar5771@list.ru	Заведующие	2024-03-10 17:27:32	manager1	pass5	✗	✎	
Чиженок Евгений Вольфович	1959-11-22	chizhenokevmeniy9284@bk.ru	Заведующие	2024-03-10 17:27:32	manager2	pass5	✗	✎	
Флавицкий Леопольд Никандрович	1974-12-19	flavickiyleopold6812@qip.ru	Заведующие	2024-03-10 17:27:32	manager3	pass5	✗	✎	
Анцидинов Адриан Фролович	1942-10-30	ankidinovadrian850@ukr.net	Заведующие	2024-03-10 17:27:32	ankidinovadrian850	cvEQLeA_39C	✗	✎	
Варнаков Кандидий Марьяч	1948-07-31	varnakovkandidiy269@live.com	Заведующие	2024-03-10 17:27:32	varnakovkandidiy269	2DL_lgDmWx	✗	✎	
Серeda Аверкий Федорович	1989-10-27	seredaaverkiy4693@live.com	Заведующие	2024-03-10 17:27:32	seredaaverkiy4693	Fk3D_b	✗	✎	
Федосенко Ермолай Анатольевич	1956-08-19	fedosenkoermolay2193@microsoft.com	Врачи	2024-03-10 17:27:53	doctor1	pass2	✗	✎	
Коробейщиков Валентин Ефимович	1997-10-24	korobeyschikovvalentin6010@yahoo.com	Врачи	2024-03-10 17:27:53	doctor2	pass2	✗	✎	
Камынин Якуб Вольфович	1944-09-02	kamyninyakub1744@ukr.net	Врачи	2024-03-10 17:27:53	kamyninyakub1744	pass2	✗	✎	

Рисунок 21 – Страница «Пользователи»

Редактор пользователей позволяет редактировать пользователей. В данном редакторе администратор может создать или изменить существующего пользователя. Для этого у него имеется набор полей таких как ФИО, почта и другие. При выборе конкретного пользователя, его текущие данные автоматически заполнятся в этих полях, что позволит избежать ручного ввода. (рисунок 22).

Редактор

ФИО

Дата рождения

Электронная почта

Уровень доступа

Дата регистрации

Логин

Пароль

Рисунок 22 – Редактор пользователей

Редактор групп пользователей (рисунок 23) позволяет редактировать группы пользователей. Здесь пользователя самого высокого уровня доступа дается возможность редактировать уровни.

Редактор уровней доступа

Название уровня доступа	Удал.	Ред.
Пациенты	✗	✎
Врачи	✗	✎
Заведующие	✗	✎
Администраторы	✗	✎

Первая страница -- -- Последняя страница

Редактор

Название уровня доступа

Рисунок 23 – Редактор групп пользователей

На рисунке 24 приведена страница «Новости». Редактор заведующих позволяет редактировать данные о заведующих. Данная страница показана на рисунке 25.

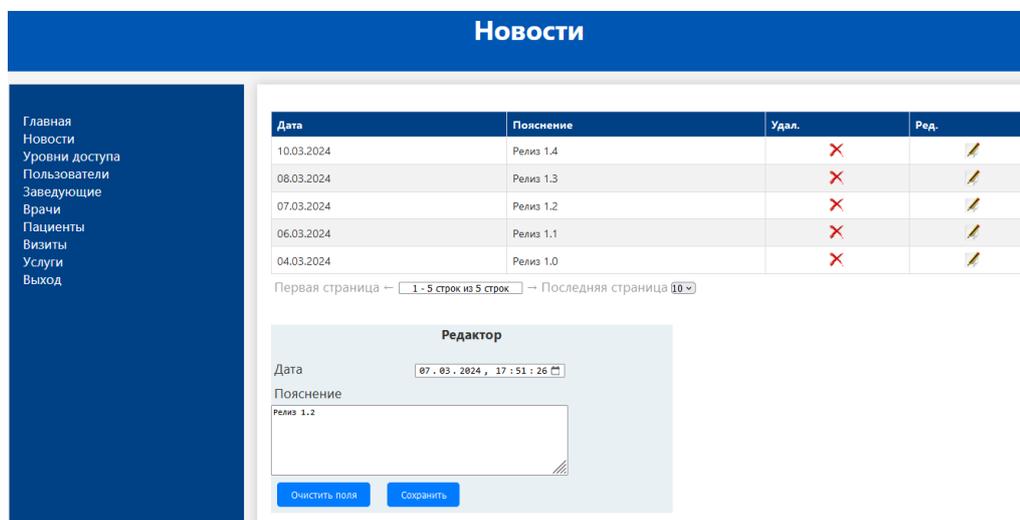


Рисунок 24 – Редактор новостей

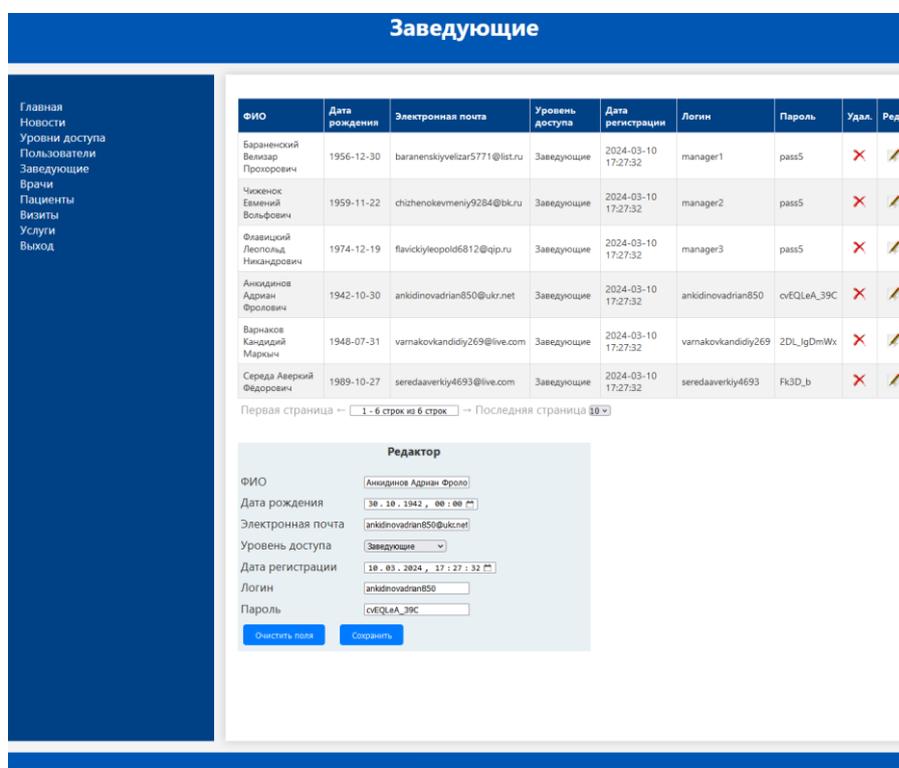


Рисунок 25 – Страница «Заведующие»

Отметим, что данная страница по своей сути аналогична такой же странице «Пользователи». Разница заключается в том, что на странице «Заведующие» мы получаем ограниченный набор пользователей. Ограничение строится на принципе разграничения прав доступа, иными словами просто фильтруются те пользователи которые подпадают под права заведующих.

Далее перейдем к редактору врачей, который позволяет редактировать данные о врачах. Данная страница показана на рисунке 26.

ФИО	Дата рождения	Электронная почта	Уровень доступа	Дата регистрации	Логин	Пароль	Удал.	Ред.
Федосенко Ермолай Анатольевич	1956-08-19	fedosenkoermolay2193@microsoft.com	Врачи	2024-03-10 17:27:53	doctor1	pass2	✗	✎
Коробейшиков Валентин Ефимович	1997-10-24	korobeyschikovalentin6010@yahoo.com	Врачи	2024-03-10 17:27:53	doctor2	pass2	✗	✎
Камынин Якуб Вольфович	1944-09-02	kamyninyakub1744@ukr.net	Врачи	2024-03-10 17:27:53	kamyninyakub1744	pass2	✗	✎
Витазев Авенир Геннадиевич	1994-01-02	vityazevavenir5155@list.ru	Врачи	2024-03-10 17:27:53	vityazevavenir5155	pass2	✗	✎
Бударов Август Жанович	1980-10-13	budarovavgust160@list.ru	Врачи	2024-03-10 17:27:53	budarovavgust160	JlvwPbvKzWZgQjhtn2iN	✗	✎
Петяшин Савелий Харлампиевич	1955-05-04	petyashinsaveliy5897@ukr.net	Врачи	2024-03-10 17:27:53	petyashinsaveliy5897	64k1s4tO	✗	✎
Свирыкин Андрон Еремеевич	1943-03-02	sviryakinandron8722@bk.ru	Врачи	2024-03-10 17:27:53	sviryakinandron8722	Odwtot7Bfek2iTWmriTKMo	✗	✎
Черномордик Филлимон Елисеевич	1956-09-30	chernomordikfilimon5960@bk.ru	Врачи	2024-03-10 17:27:53	chernomordikfilimon5960	EqQv.JAwEx.5Mx_6SbgXD	✗	✎
Кастулов Аполлос Дмитриевич	1976-05-20	kastulovapollos8276@bk.ru	Врачи	2024-03-10 17:27:53	kastulovapollos8276	dr5c2WoeUba	✗	✎
Верховцев Порфир Каллиникович	1965-12-27	verhovcevporfir3969@microsoft.com	Врачи	2024-03-10 17:27:53	verhovcevporfir3969	QLXVOR3a1SYzibPPQuwPJNL	✗	✎

Первая страница — 1 - 10 строк из 10 строк — Последняя страница 10

Редактор

ФИО:

Дата рождения:

Электронная почта:

Уровень доступа:

Дата регистрации:

Логин:

Пароль:

Рисунок 26 – Страница «Врачи»

По своей сути данная страница аналогична странице «Пользователи» кроме той особенности, что на данной странице уже отображаются врачи.

Редактор пациентов позволяет редактировать данные о клиентах. Редактор пациентов показан на рисунке 27.

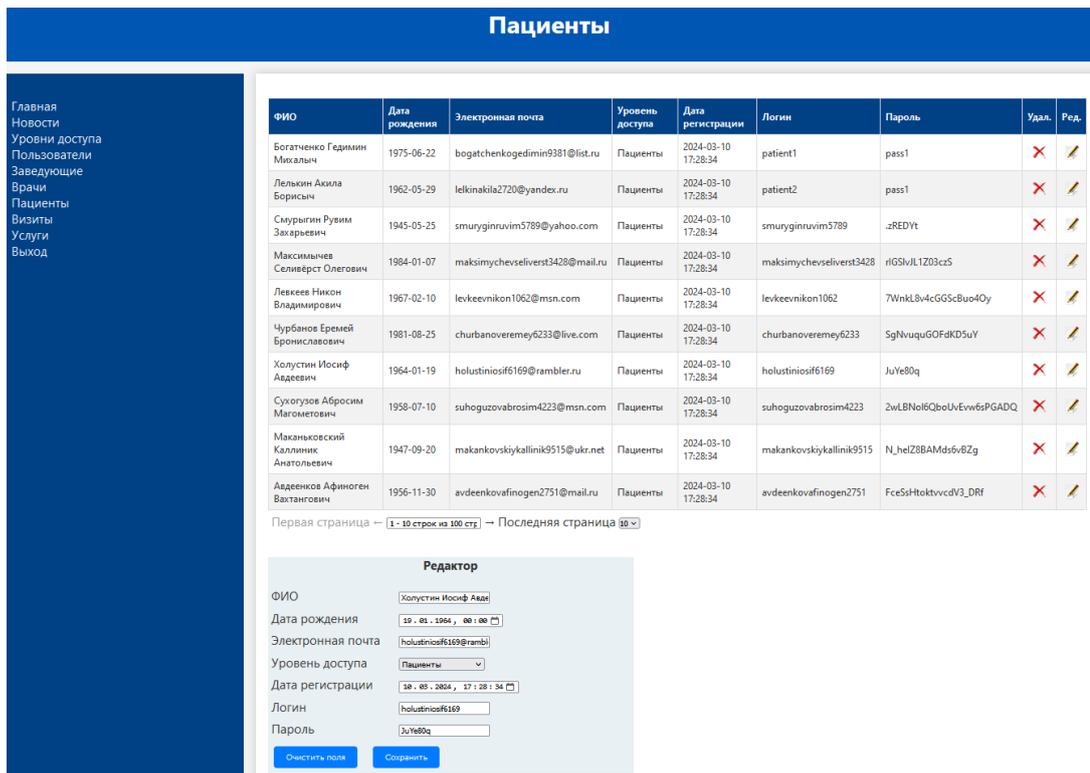


Рисунок 27 – Страница «Пациенты»

Редактор услуг представлен на рисунке 28.

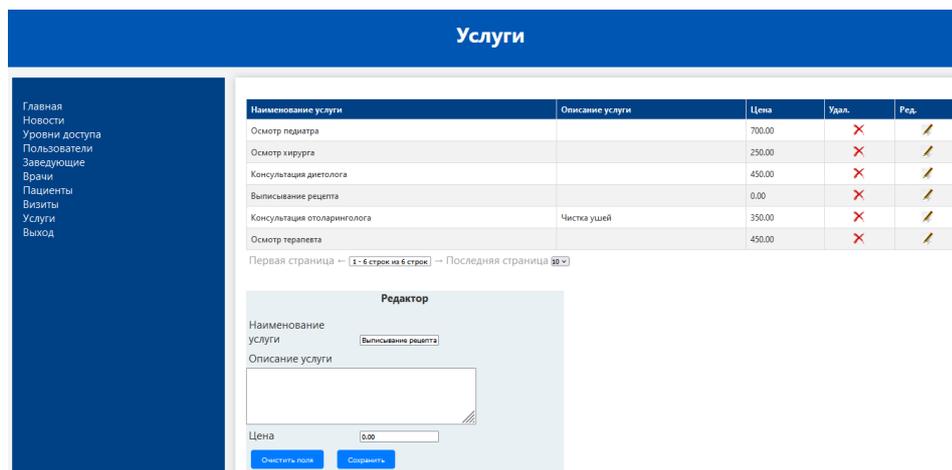


Рисунок 28 – Страница «Услуги»

Редактор визитов позволяет редактировать данные о визитах пациентов. Данная страница показана на рисунке 29.

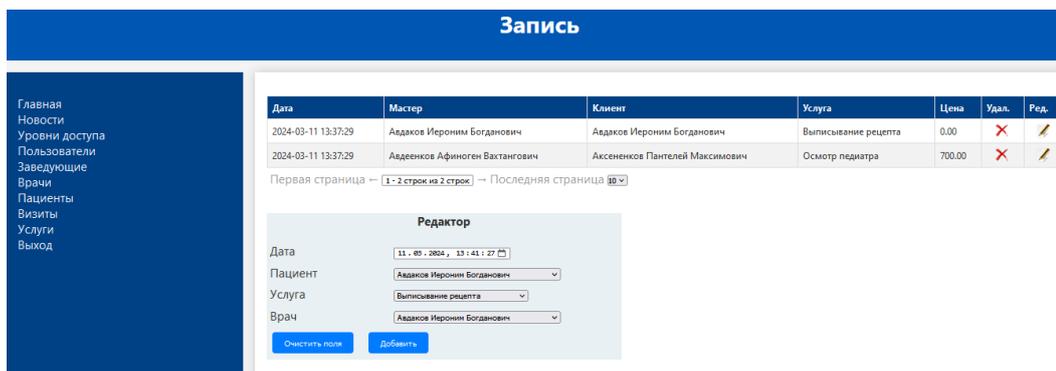


Рисунок 29 – Страница «Визиты»

Администраторы или Заведующие могут просматривать и редактировать пациентов, так же при желании они за счет своих привилегий в уровне доступа внутри системы могут удалить необходимую запись из таблицы пациентов. Пациенты могут просматривать и редактировать только свои визиты. Администраторы и заведующие могут контролировать загруженность врачей и управлять остальными аспектами в системе. В свою очередь врачи могут в удобном формате посмотреть список пациентов и записей на прием чтобы грамотно спланировать свое рабочее время и не отвлекаться на лишние действия.

Результатом применения данного веб-приложения является автоматизация работы больницы: пациенты могут удаленно просмотреть список врачей, список предоставляемых услуг, а также сделать запись на удобную для себя дату и время.

3.3 Оценка и обоснование экономической эффективности разработки АИС

3.3.1 Выбор методики расчета экономической эффективности

В соответствии с ГОСТ 19.102-94 «Стадии разработки» имеются пять стадий разработки программного обеспечения:

- составление технического задания;

- эскизное проектирование будущего проекта;
- техническое и рабочее проектирование;
- внедрение.

Для правильного следования всем этапам разработки в первую очередь необходимо указать трудоемкость каждого этапа. Трудоемкость этапов можно указывать в днях или в часах, но поскольку удобнее всего дальнейшие расчеты проводить именно в часах, мы так же будем использовать эту единицу измерения.

Таблица 2 включает наименования этапов и их трудоемкость. На ней мы можем наглядно посмотреть каждую стадию разработки, где каждая стадия разбита на определенные этапы со своей трудоемкостью в часах.

Таблица 2 – Трудоемкость этапов

Стадия	Этапы	Трудоемкость, ч
Техническое задание	Необходимость	13
	Исследовательские работы	23
	Создание и утверждение ТЗ	5
Эскизный проект	Разработка	41
	Утверждение	10
Технический проект	Разработка	50
	Утверждение	11
Рабочий проект	Разработка ПО	93
	Разработка документации	22
	Тестирование	18
Внедрение	Включение в рабочий контур	14
Итого		300

Далее мы вычислим время которое потребуется для реализации проекта по простой формуле (формула 1):

$$T_{\text{рп}} = \frac{T_{\text{час}}}{t_{\text{рд}}}, \quad (1)$$

где

$T_{\text{час}}$ – общее время,

$t_{\text{рд}}$ – число рабочих часов.

Таким образом мы получим, что: $T_{рп} = \frac{T_{час}}{t_{рд}} = \frac{300}{8} = 38$ дней.

Исходя из данного расчета, можно сделать вывод, что на реализацию всех стадий разработки уйдет около 38 дней.

Для планирования воспользуемся диаграммой Ганта. [17] Для разработки программного обеспечения привлекаются два человека:

- начальник отдела разработки;
- full-stack разработчик (программист).

Диаграмма Ганта показана на рисунке 30.

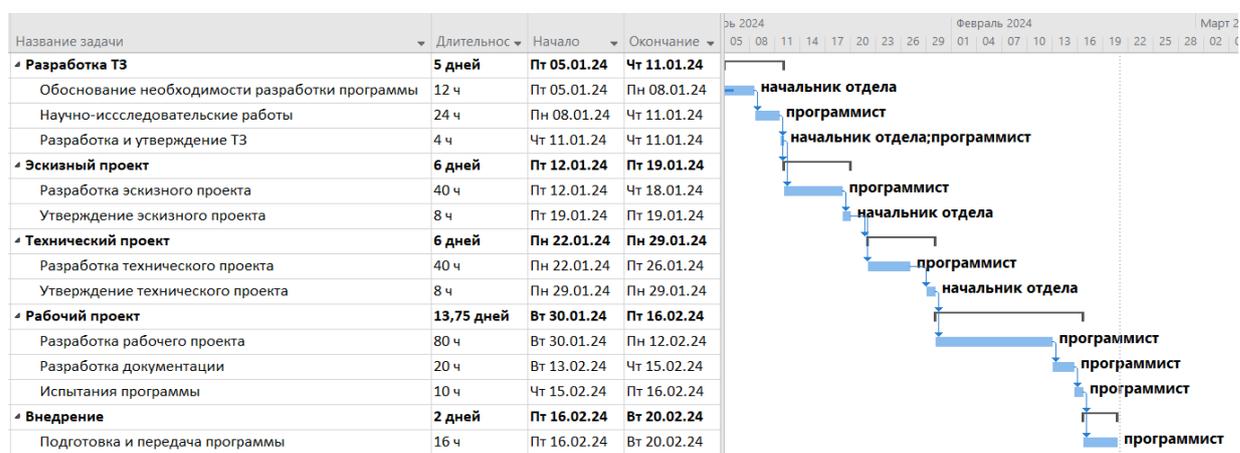


Рисунок 30 – Диаграмма Ганта

Из этой диаграммы можно наблюдать, что начальник задействован в выполнении проекта всего 4 дня, а программист в свою очередь 28 дней, это связано с тем что именно он непосредственно занимается основными этапами разработки такими как написание кода, тестирование и внедрение.

3.3.2 Расчет показателей экономической эффективности проекта

Обычная смета затрат на разработку ПО, в том числе и АИС, включает следующие основные статьи:

- Зарботную плату;
- Эксплуатационные расходы;
- Расходы на приобретение средств вычислительной техники;

– Накладные расходы.

Таблица 3 содержит в себе всю информацию об оплате труда сотрудников, задействованных в разработке. Здесь представлен список специалистов, задействованных в проекте, далее идет их месячных оклад в рублях, и в заключении колонка с оплатой полного восьмичасового рабочего дня в рублях.

Таблица 3 – Оплата труда

Сотрудник	Оклад, руб.	Оплата за 8 часовой день, руб.
Начальник	100000	5000
Разработчик	60000	3000

В таблице 4 рассчитаны суммарные затраты на оплату труда. Здесь для каждого сотрудника мы выписываем их почасовую оплату за полный день, количество дней необходимых на закрытие нашего проекта в соответствии с предыдущими расчетами, включая диаграмму Ганта, и в последнем столбце мы вычисляем сумму затрат на каждого сотрудника необходимую для полного цикла разработки ПО.

Таблица 4 – Суммарные затраты на оплату труда

Специалист	Оплата за 8 часовой день	Кол-во рабочих дней	Затрата на оплату труда
Начальник отдела	5000	4	20000
Программист	3000	28	84000
Итого			104000

Стоит отметить, что дополнительно мы можем рассчитать страховые взносы, которые зачастую составляют треть от затрат на основную заработную плату. Не стоит упускать из внимания данные расчеты так как страховые взносы - это обязательные платежи для любого легального работодателя на территории РФ.

Взносы рассчитаем по формуле 2.

$$CB = CЗП \cdot CO \quad (2)$$

На следующем шаге нам необходимо учитывать, что для успешного выполнения работ будут нужны некоторые расходные материалы, например, бумага для составления бумажных отчетов, принтер для печати этих самых отчетов по проекту. Их стоимость показана в таблице 5.

Таблица 5 – Затраты на расходные материалы

Наименование	Единица измерения	Количество	Цена	Итого
Бумага	Пачка	1	200	200
Картридж для принтера	Штука	1	500	500
Флеш-накопитель	Штука	2	300	600
Переходники	Штука	3	120	360
Итого				1660

Здесь наглядно показан весь список номенклатуры, которую мы будем учитывать в расходных материалах. Для каждого наименования указана единица измерения, а также количество этих единиц. Далее указан столбец с ценами на каждую единицу и итоговая сумма всех расходов.

Эксплуатационные расходы показаны в таблице 6.

Таблица 6 – Эксплуатационные расходы

Компьютер	Мощность, кВт	Кол-во дней	Кол-во часов	Кол-во кВт*ч	Стоимость кВт*ч	Цена
Ноутбук	0,1	4	32	3,2	6,17	197
ПК	0,1	28	224	22,4	6,17	1382
Принтер	0,1	30	256	25,6	6,17	1580
Итого						3159

В таблице 7 показана калькуляция накладных расходов на основании утвержденных процентных ставок.

Таблица 7 – Калькуляция накладных расходов

Статья затрат	Сумма	%	Сумма расходов
Заработная плата	104000	5	5200
Расходные материалы	1840	2	44
Эксплуатационные расходы	3159	2	63
Итого			5308

Далее проведем расчет планируемой себестоимости разрабатываемого ПО. Эти расчеты приведены в таблице 8.

Таблица 8 – Структура планируемой себестоимости

Статья затрат	Сумма
Заработная плата сотрудников	104000
Страховые взносы	31616
Расходные материалы	2220
Эксплуатационные расходы	3159
Накладные расходы	5308
Итого	146303

Таким образом, можно сделать вывод, что трудозатраты специалиста при окладе 60000 рублей: $ЗП = 60000 + 60000 \cdot 30,4\% = 78240$ руб.

Стоимость часа работы составит: $ЗП_{1ч} = \frac{ЗП}{160} = \frac{78240}{160} = 489$ руб.

Общая экономия за месяц составит: $Э_{мес} = ЗП_{1ч} \cdot 32 = 489 \cdot 32 = 15648$ руб.

Экономический эффект от внедрения (формула 3):

$$ЭЭ = Э_{год} - E_H \cdot K_C, \quad (3)$$

где

$Э_{год}$ – экономия от использования продукта, руб.;

Еп – нормативный коэффициент экономической эффективности;

Кс – единовременные затраты, руб.

Экономический эффект составит $ЭЭ = 187776 - 0,15 \cdot 146303 = 165831$ руб.

Эффективность от внедрения (формула 4):

$$ЭФ = \frac{ЭЭ}{Кс}, \quad (4)$$

Срок окупаемости (формула 5):

$$T_{co} = \frac{Кс}{ЭЭ}, \quad (5)$$

Тогда срок окупаемости составит $T_{co} = \frac{146303}{165831} \approx 0,88$ года \approx 11 месяцев.

Таким образом, можно сделать вывод, что разработка данного программного обеспечения экономически целесообразна.

Выводы по разделу 3

Раздел охватывает всесторонний анализ и практическое применение ключевых компонентов для создания эффективной и надежной автоматизированной информационной системы. Раздел начинается с важного этапа выбора архитектуры системы, что лежит в основе дальнейшего проектирования и определяет основные принципы построения АИС, влияя на её масштабируемость, надежность и удобство поддержки.

Следующий шаг включает в себя выбор технологий разработки программного обеспечения, где принимается решение о применении конкретных инструментов и языков программирования, что напрямую воздействует на скорость разработки, возможности интеграции и гибкость

системы. Критическим моментом является также выбор системы управления базами данных, который опирается на анализ требований к структуре, хранению и обработке данных. Это решение важно для обеспечения эффективного управления данными и поддержки бизнес-процессов организации. Разработка физической модели данных становится следующим шагом, который является ключевым фактором в выборе определенной структуры и способа хранения будущих данных в выбранной нами системе управления данными, что в конечном счете будет влиять не только на производительность системы, но и на целостность содержащихся в ней данных.

Создание программного обеспечения включает проектирование модулей системы и разработку кода, где особое внимание уделяется обеспечению модульности, расширяемости и взаимосвязи компонентов системы, что способствует адаптации системы к изменяющимся условиям эксплуатации и требованиям пользователей.

Описание функциональности системы дает полное представление о возможностях АИС, подчеркивая её основные и дополнительные функции для улучшения эффективности работы пользователей. Особое внимание уделено оценке экономической эффективности разработки АИС, включая выбор методики расчета и анализ показателей экономической эффективности. Этот этап позволяет оценить ожидаемую отдачу от проекта и его влияние на финансовые показатели организации, подтверждая целесообразность инвестиций в разработку и внедрение системы.

Таким образом, необходимо подчеркнуть необходимость комплексного подхода к физическому проектированию АИС, начиная от выбора архитектуры и заканчивая оценкой экономической эффективности проекта. Этот процесс требует тщательного планирования и координации всех этапов разработки для создания системы, которая не только отвечает текущим требованиям, но и способна адаптироваться к будущим вызовам.

Заключение

Работа над проектом демонстрирует комплексный подход к решению задач автоматизации в предметной области, начиная от анализа текущего состояния и потребностей, и заканчивая практической реализацией и оценкой эффективности внедренной системы.

Были исследованы и описаны функциональные процессы предметной области, что позволило выявить ключевые направления для автоматизации и оптимизации работы. Выбор технологии концептуального и логического моделирования АИС основывался на глубоком анализе требований к системе и позволил создать надежную и масштабируемую архитектуру, способную адаптироваться к изменяющимся условиям эксплуатации.

В процессе проектирования базы данных был сделан акцент на обеспечении целостности, безопасности и эффективности обработки данных. Выбор реляционной модели базы данных оправдал себя, предоставив мощные средства для выполнения сложных запросов и аналитической работы с данными.

Разработка программного обеспечения АИС была выполнена с учетом современных методик и лучших практик программирования, что обеспечило высокую надежность и производительность системы. Особое внимание было уделено пользовательскому интерфейсу и удобству использования системы, чтобы максимально упростить процесс адаптации персонала к новой системе.

Оценка экономической эффективности разработки АИС подтвердила ее высокую стоимостную эффективность и положительное влияние на операционную деятельность организации. Результаты внедрения показали значительное улучшение ключевых показателей работы, сокращение времени на выполнение рутинных операций и повышение уровня удовлетворенности как сотрудников, так и клиентов организации.

В заключение, разработка и внедрение АИС стали значительным шагом в цифровизации и автоматизации процессов в предметной области, открывая

новые возможности для дальнейшего развития и совершенствования деятельности организации. Работа над проектом подчеркнула важность тесного взаимодействия между разработчиками, пользователями и управлением организации, что является ключом к успешному внедрению инновационных решений в современном мире.

Список используемой литературы

1. Ботрос, Д. Тинли С. MySQL по максимуму. – СПб.: Питер, 2023. – 432 с.
2. Васильев А. Программирование на PHP в примерах и задачах. – М.: Эксмо, 2021. – 352 с.
3. Веб-сайт определение и виды [Электронный ресурс] URL: <https://www.seonews.ru/glossary/veb-sayt/>
4. Гвоздева В.А. Информатика, автоматизированные информационные технологии и системы. – М.: ИНФРА-М, 2018. – 544 с.
5. Евгеньев Г.Б. Основы автоматизации технологических процессов и производств / Г.Б. Евгеньев, С.С. Гаврюшин, А.В. Грошев, М.В. Овсянников, П.С. Шильников. – Москва: Изд-во МГТУУ им. Н.Э. Баумана, 2018. – 441 с.
6. Информационные технологии в сфере здравоохранения [Электронный ресурс] URL: <https://edu.tatar.ru/upload/storage/org6153/files/применение%20информационных%20технологий%20в%20профессиональной%20деятельности.pdf>
7. Как начать моделировать бизнес-процессы в BPMN [Электронный ресурс] URL: <https://habr.com/ru/articles/697326/>
8. Колисниченко Д. PHP 5/6 и MySQL 6 Разработка Web приложений. 3-е изд. – СПб.: BHV, 2011. – 528 с.
9. Колисниченко Д. PHP и MySQL. Разработка Web-приложений. – СПб.: BHV, 2017. – 640 с.
10. Коржов В. Многоуровневые системы клиент-сервер [Электронный ресурс] URL: <https://www.osp.ru/nets/1997/06/142618>
11. Котеров И. PHP 8. Наиболее полное руководство. – СПб.: BHV, 2023. – 992 с.
12. Кузнецов М., Симдянов И.В. Самоучитель PHP7. – Санкт-Петербург: БХВ-Петербург, 2018. – 320 с.
13. МакГрат М. JavaScript для начинающих. – М.: Эксмо, 2022. – 232 с.

14. Мезенцев, К.Н. Автоматизированные информационные системы. - М.: ИЦ Академия, 2018. - 176 с.
15. Мокеев В.В. Бизнес-информатика / В.В. Мокеев, Е.В Бунова, О.С. Буслаева. – Челябинск: издательский центр ЮУрГУ, 2018. – 67 с.
16. Одинцов Б.Е. Информационные системы управления эффективностью бизнеса. - Люберцы: Юрайт, 2018. - 206 с.
17. Плюсы и минусы Oracle [Электронный ресурс] URL: https://blog.uniwex.io/plyusy-i-minusy-oracle-database/#Достоинства_Oracle_Database
18. Попов А. А. Эргономика пользовательских интерфейсов в информационных системах. – М.: Кнорус, 2023. – 304 с.
19. Скляр Д. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов. – М.: Вильямс, 2020. – 464 с.
20. Современный учебник JavaScript [Электронный ресурс] URL: <https://learn.javascript.ru/intro>
21. Ammann Paul, Offutt Jeff. Introduction to Software Testing - Cambridge University Press, 2016. – 346 p. – ISBN-10 0521880386; ISBN978-0521880381.
22. Integration DEfinition for information modeling [Электронный ресурс] URL: <https://ru.wikipedia.org/wiki/IDEF1X>
23. Martin Fowler. Patterns of Enterprise Application Architecture. [Электронный ресурс] URL: https://bizoutlook.by/pdf/Patterns_of_Enterprise_Application_Architecture.pdf
24. Matt Z. PHP 8: Objects, patterns, and programming techniques [Электронный ресурс] URL: <https://sgp1.vultrobjects.com/books/Codeing/PHP8ObjectsPatternsandPractice.pdf>
25. Robin Nixon. Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites, 2014. – 700 p.
26. Simpson K. You Don't Know JS Yet: Get Started [Электронный ресурс] URL: <https://www.booksfree.org/wp-content/uploads/2021/09/you-do-not-know.pdf>