**Содержание**

Введение 1

1 Общая часть 1

1.1 Структура предприятия, место и функции информационного отдела 1

1.2 Анализ бизнес-процессов и постановка задачи 1

1.3 Анализ средств разработки 11

2 Специальная часть 11

2.1 Проектирование архитектуры решения 11

2.2 Проектирование базы данных 11

2.3 Проектирование программных модулей решения 11

2.4 Реализация технического решения 11

2.4.1 Разработка пользовательского интерфейса решения 11

2.4.2 Разработка программных модулей решения 11

2.5 Тестирование разработанного решения 11

2.3.2 Разработка интерфейса программы 11

3 Организационно-экономическая часть 11

3.1 Расчет сметы затрат на разработку программного решения 11

3.2 Расчет экономической эффективности внедрения программного решения 11

4 Мероприятия по технике безопасности и охране окружающей среды 11

4.1 Организация безопасности труда на предприятии 11

4.2 Техника безопасности при работе с вычислительной техникой 11

Заключение 11

Список использованных источников 11

# **Введение**

Автоматизация учета данных регистратуры поликлиники — это внедрение информационной системы, которая позволяет автоматизировать процессы записи на прием к врачу, учета посещений пациентов, выписки направлений и других медицинских документов, а также управления расписанием работы врачей.

Актуальность применения автоматизации учета данных регистратуры поликлиники заключается в том, что современные медицинские учреждения все чаще стремятся автоматизировать свои процессы, что позволяет значительно упростить и ускорить работу медицинского персонала, а также повысить качество и безопасность предоставляемых медицинских услуг.

Практическая значимость подобных приложений заключается в следующем:

1. Удобство хранения и доступа к медицинским данным пациентов. Автоматизация учета данных регистратуры поликлиники позволяет хранить все данные пациента в одном месте, что облегчает их поиск и доступность для врачей и медицинского персонала.
2. Улучшение качества предоставляемых медицинских услуг. Благодаря автоматизации учета данных регистратуры поликлиники врачи имеют доступ к полной и актуальной информации о пациенте, что помогает им принимать обоснованные решения и предоставлять более качественное лечение.
3. Сокращение времени на обработку и анализ медицинских данных. Автоматизация процессов сбора, хранения и анализа данных позволяет сократить время, затрачиваемое на рутинные операции, и увеличить эффективность работы медицинского персонала.
4. Улучшение безопасности данных. Автоматизация учета данных регистратуры поликлиники позволяет установить ограничения доступа к конфиденциальной информации, что повышает безопасность медицинских данных пациентов.

Основой любой электронной системы является база данных.

База данных (БД) — это структурированная коллекция данных, которая используется для хранения информации о пациентах и их медицинской истории. Любая база данных создается при помощи систем управления базами данных.

Система управления базами данных (СУБД) — это специализированное программное обеспечение, предназначенное для управления и организации данных в базах данных.

Во время прохождения производственной практики в ГБУЗ «Районная больница г. Верхнеуральск», было дано задание, которое заключалось в разработке системы, которая позволит вести учет данных пациентов в электронном виде.

В соответствии с полученным заданием на производственной практике была определена цель дипломного проекта: «Автоматизация учета данных регистратуры поликлиники».

Исходя из цели работы был описан следующий круг задач дипломного проекта:

1. Изучение требований заказчика и проведение анализа предметной области — данные пациента и их учет.
2. Проектирование модели базы данных.
3. Создание пользовательского интерфейса учета данных регистратуры поликлиники.
4. Реализация функциональных возможностей электронной системы.
5. Тестирование и отладка приложения для обеспечения его корректной работы и соответствия требованиям.
6. Оформление технической документации в виде пояснительной записки на дипломный проект.

# **1 Общая часть**

# **1.1 Структура предприятия, место и функции информационного отдела**

Районная больница города Верхнеуральск – осуществляет свою деятельность в соответствии с предметом и целями деятельности, согласно общепринятым стандартам качества оказываемой медицинской помощи определенными в соответствии с законами и иными нормативными правовыми актами Российской Федерации.

Учреждение функционирует как амбулатория, медицинский центр, поликлиника для взрослых, диагностический центр, больница и родильный дом, где ведут прием врачи различных специализаций,

Поликлиника для взрослых ГБУЗ Районная больница г. Верхнеуральск пользуется современным оборудованием. Высококвалифицированные специалисты, которые качественно обслуживают своих пациентов, а также приемлемая стоимость на услуги приносит мед центру ГБУЗ Районная больница г. Верхнеуральск высокую популярность. На web-сайте центра вам предоставляется информация о ценах на лечение и возможность записаться на прием онлайн или по номеру телефона. Онлайн-записи к врачу обрабатываются круглыми сутками. Лечение наших пациентов благодаря хорошим результатам ультрасовременной диагностики выполняется на достойном уровне. ГБУЗ Районная больница г. Верхнеуральск предоставляет индивидуальный подход к каждому пациенту, нет очередей и созданы все комфортные условия для пациентов.

Целями ГБУЗ Районная больница г. Верхнеуральск являются:

* обеспечение высокого качества и доступности оказываемой медицинской помощи;
* соблюдение требований Территориальной программы государственных гарантий оказания гражданам бесплатной медицинской помощи;
* соблюдение стандартов и порядков оказания медицинской помощи;
* повышение профессионального уровня работников;
* создание благоприятных условий работникам для выполнения своих должностных обязанностей;
* соблюдение Кодекса медицинского работника;
* эффективное расходование заработанных финансовых средств.

В больнице работают специалисты, занимающиеся информационно-технической деятельностью, а именно: инженер-программист, оператор ЭВМ, системный администратор.

Основные функции для специалистов информационно-технической деятельности являются:

* выполнение работ по разработке, тестированию, отладки, внедрению и сопровождению программного обеспечения;
* ведение технологической документации;
* организация доступа сотрудников к информационным ресурсам, представленным в электронном виде;
* обеспечение бесперебойной работы сетевого оборудования, компьютерной техники коллективного использования, оборудования передачи данных, устройств ввода и вывода данных (принтер, сканер, микрофон и так далее).

Инженер-программист — это человек, который применяет принципы разработки программного обеспечения при проектировании, разработке, обслуживании, тестировании и оценке компьютерного программного обеспечения.

Инженер-программист обязан:

1. Принимать участие в проведении работ по разработке, внедрению, опытной эксплуатации, наладке, модернизации, сопровождению.
2. На основе анализа математической модели и алгоритмов решения задач мониторинга и управления технологических процессов своевременно разрабатывать программное обеспечение, проводить его тестирование и отладку.
3. Разрабатывать технологию решения задачи по всем этапам обработки информации.
4. Осуществлять освоение программного обеспечения автоматизированных систем.
5. Разрабатывать информационное обеспечение систем управления.
6. Выполнять работу по отладке программного обеспечения, проводить корректировку разработанной программы на основе анализа выходных данных.
7. Определять объем и содержание данных контрольных примеров, обеспечивающих наиболее полную проверку соответствия программы их функциональному назначению.
8. Разрабатывать инструкции по работе с программным обеспечением, оформлять необходимую техническую документацию в соответствии со стандартами и документами.
9. Обеспечивать объективность и достоверность информации, материалов и отчетности, а также в пределах своей компетенции, конфиденциальность информации, составляющую коммерческую или служебную тайну.



Рисунок 1 – Организационная структура

Для автоматизации деятельности предприятия используются персональные компьютеры, ноутбуки и моноблоки для сотрудников, сервера для обеспечения доступа к базе данных и общим сетевым ресурсам, мультифункциональным устройствам для работы с документами (принтер, сканер, МФУ), высокоскоростной доступ в Интернет. Удаленные здания подключаются с помощью VPN, который шифрует все данные.

Виртуальная частная сеть (VPN) — технология, которая позволяет установить безопасное подключение к сети Интернет.

При подключении к Интернету через VPN, программное обеспечение создает безопасное соединение между вашим устройством и удаленным VPN-сервером, шифруя ваши данные просмотра веб-страниц и скрывая ваш IP-адрес.

Для более удобного просмотра всех комплектующих установленных в устройство можно просмотреть в таблице 1.

Таблица 1 – Технические характеристики ПК, ноутбука и моноблока.

|  |  |  |  |
| --- | --- | --- | --- |
| **Название** | **Характеристики устройства** | | |
| **ПК** | **Ноутбука** | **Моноблока** |
| Материнская плата | GIGABYTE B450M H | 15-dk2050ur | Intel® 300 серии H310 |
| Процессор | AMD Ryzen 3 | Intel Core i5 10300H | Intel Core i3, Core i5, Core i7  H310 |
| Видеокарта | Встроенная | NVIDIA GeForce GTX 1650 Ti - 4 ГБ | Intel® UHD Graphics Engine 600 серии |
| Оперативная память | DDR4 8G 16G 2666MHz | DDR4 8G 3200MHz | DDR4 8G 16G 2400/2666MHz |
| Жесткий диск | 250 - 500 ГБ | - | 1 — 6 Т |
| Тверд. Тел. накопитель | 512 ГБ | 512 ГБ | 120 — 512 ГБ |
| Блок питания | Sven 350W | Li-Ion | 150-200W |
| Монитор | Samsung SyncMaster S22A200B | - | - |

Совокупность программ и сопровождающей их документации, предназначенная для решения задач на ПК, называется программным обеспечением.

Программное обеспечение делится на:

* системное программное обеспечение;
* прикладное программное обеспечение;
* инструментальное программное обеспечение.

Системное программное обеспечение – совокупность программ и программных комплексов для обеспечения работы компьютера и сетей.

Прикладное программное обеспечение – комплекс взаимосвязанных программ для решения задач определенного класса или конкретной предметной области.

Инструментальное программное обеспечение – программное обеспечение, предназначенное для использования в ходе проектирования, разработки и сопровождения программ.

В организации используются следующее системное программного обеспечение, представленное в таблицах 2 и 3.

Таблица 2 – Используемые архиваторы.

|  |  |
| --- | --- |
| **Название** | **Описание** |
| Winrar | Архиватор файлов для 32- и 64-разрядных операционных систем Windows, позволяющий создавать, изменять и распаковывать архивы RAR и ZIP, а также работать с множеством архивов других форматов |
| 7-Zip | Свободный файловый архиватор с высокой степенью сжатия данных. Поддерживает несколько алгоритмов сжатия и множество форматов данных, включая собственный формат 7z c высокоэффективным алгоритмом сжатия LZMA |
| Ark | Программа для работы с архивами различных форматов в среде KDE. С помощью Ark архивы можно просматривать, извлекать, создавать и изменять. Она может обрабатывать различные форматы (tar, gzip, bzip2, rar, zip, lha) |

Таблица 3 – Используемые операционные системы.

|  |  |
| --- | --- |
| **Название** | **Описание** |
| OS Windows 7/8/10/11 | Операционная система для персональных компьютеров и рабочих станций, разработанная корпорацией Microsoft в рамках семейства Windows NT |
| OS AstraLinux (OREL) версии 12.2.40 и выше | Операционная система на базе ядра Linux, которая внедряется в России в качестве альтернативы Microsoft Windows. Обеспечивает степень защиты обрабатываемой информации до уровня государственной тайны «особой важности» включительно |

Прикладное программное обеспечение, используемое в организации:

* Microsoft Word - текстовый процессор, предназначенный для создания, просмотра и редактирования текстовых документов, с локальным применением простейших форм таблично-матричных алгоритмов;
* Microsoft Excel - программа для работы с электронными таблицами, созданная корпорацией Microsoft;
* LibreOffice Writer - текстовый процессор и визуальный редактор HTML, входящий в состав офисного пакета LibreOffice;
* LibreOffice Calc - табличный процессор, входящий в состав офисного пакета LibreOffice. Является ответвлением табличного процессора OpenOffice Calc;
* Adobe Acrobat — пакет программ, выпускаемый с 1993 года компанией Adobe Systems и предназначенный для создания и просмотра электронных публикаций в формате PDF;
* ABBYY FineReader - программа для оптического распознавания символов, разработанная международной компанией ABBYY. Программа позволяет переводить изображения документов в электронные редактируемые форматы;
* ANYDesk - Приложение для удаленного рабочего стола, распространяемое компанией AnyDesk Software GmbH. Проприетарное программное обеспечение обеспечивает независимый от платформы удаленный доступ к персональным компьютерам и другим устройствам, на которых запущено основное приложение.

Инструментальное программное обеспечение, используемое в организации:

* Visual Studio - линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств;
* Mircosoft SQL Server Management Studio - Утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера;
* Медицинская информационная система «БАРС.Здравоохранение»;
* Visual Studio Code - Текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений.

# **1.2 Анализ бизнес-процессов и постановка задачи**

Проблема, выявленная в результате анализа деятельности регистратуры поликлиники, заключается в неэффективном учете и обработке данных. В настоящее время процессы учета информации о посещениях пациентов, записи на прием к врачам, выписке направлений и других административных процедурах осуществляются вручную, что часто приводит к ошибкам, задержкам и неэффективному использованию ресурсов.

Для решения данной проблемы в будущем планируется внедрение системы автоматизации учета данных регистратуры поликлиники. Это позволит ускорить процессы обработки информации, снизить вероятность ошибок, улучшить качество обслуживания пациентов и оптимизировать рабочие процессы сотрудников регистратуры.

Основанием для разработки проекта автоматизации учета данных регистратуры поликлиники послужило стремление к повышению эффективности работы медицинского учреждения, улучшению качества обслуживания пациентов и сокращению временных затрат на административные процессы.

Автоматизация учета данных также позволит лучше контролировать информацию, предотвращать потерю данных и обеспечить более точный анализ статистической информации для принятия управленческих решений.

# **1.3 Анализ средств разработки**

Для разработки автоматизация учета данных регистратуры поликлиники была выбрана среда разработки Visual Studio и Radzen Blazor Studio. Выбор пал именно на эти среды, потому что имеется большое количество документации и электронных ресурсов, которые позволяют решить трудности возникшие на момент разработки. Visual Studio поддерживает универсальный и гибкий язык программирования, а также файлы совместимые с системными. Удобный интерфейс, а также проведение тестирования с использованием точек остановок. Radzen Blazor Studio предоставляет возможность создания веб-приложений на платформе Blazor, что позволяет создать современное и эффективное приложение с использованием технологии WebAssembly.

СУБД была выбрана SQLite — это компактная и легковесная СУБД, которая хранит базы данных в одном файле, что удобно для развертывания и поддержки. SQLite обладает хорошей производительностью и поддерживает стандарт SQL, что обеспечит надежное хранение данных для вашего проекта. Также она была выбрана, потому что хорошо интегрируется с проектами от Microsoft, к примеру .Net Framework.

Технология доступа к данным пала на SQLiteStudio, потому что она является гибкой и легко переносимой вместе с проектом, что позволяет быстро вносить какие-либо изменения или дополнения в проекте, а также в самой базе данных. Также она обладает высокой надежностью, удобным интерфейсом, большим набором инструментов.

Язык программирования использовал C#, потому что имеется опыт в работе с ним, а также он легок в изучение. Наличие обширной документации на сайтах, которые помогают при возникновении вопросов. Он также имеет интеграцию с платформой .NET и обладает высокой надежностью и много поточностью.

# **2 Специальная часть**

# **2.1 Проектирование архитектуры решения**

В приложении есть несколько пользователей: администратор, регистратор, доктор и пациент.

Пациент обращается к регистратору с целью пройти осмотр, регистратор осуществляет проверку наличия учетной записи пользователя в системе и осуществляет запись к нужному доктору, если учетной записи пользователя нету, то регистратор начинает добавлять данные пациента в систему, а после чего записывает к нужному доктору. Также если есть какие-то ошибки в уже имеющейся учетной записи пациента, регистратор может их изменить.

Администратор осуществляет добавление пользователей в систему в соответствие, редактирование и их удаление.

После того как запись пациента к нужному врачу состоялась, доктор проводит осмотр пациента и заполняет данные в систему. Также имеется возможность изменения данных проведенного осмотра или удаление осмотра.



Рисунок 2 – Диаграмма Use-Case

Ход реализации программы:

Определение логики работы системы.

Создание проекта в Radzen Blazor Studio, определение цвета компонентов для разработки и стиля текста.

Создание базы данных в SQLite, построение таблиц, для хранения данных об пользователях, проведенных осмотрах и так далее.

Проектирование интерфейса, продумывание визуальной части системы.

Реализация функциональных возможностей для работы с данными (добавление, редактирование, удаление, экспорт, отправка сообщений и так далее).

# **2.2 Проектирование базы данных**

Входная информация – это информация, которая подается на вход в систему, процесс или программу. Она может быть представлена в виде текста, чисел, изображений, звуковых файлов и так далее. Входная информация служит для выполнения определенных операций, вычислений или обработки данных.

Входные данные разработанного приложения: логин и пароль, с помощью которых осуществляются вход в систему, данные о пациенте (фамилия, имя, отчество, паспортные данные, полис и так далее), которые необходимы для регистрации пациента в системе, данные вводимые пользователем при осмотре пациента.

Таблица 4 – «Gender» (пол)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| Title | NVARCHAR(10) | Название |

Таблица 5 – «Diagnosis» (диагноз)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| Title | NVARCHAR(250) | Название |

Таблица 6 – «Visit» (посещение)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| DateVisit | DATE | Дата посещения |
| TimeVisit | TIME | Время посещения |
| Complaint | NVARCHAR(500) | Жалоба |
| Appointments | NVARCHAR(500) | Назначения |
| Status | NVARCHAR(20) | Статус |

Таблица 7 – «Schedule» (расписание)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| DateStart | DATE | Дата начала |
| TimeStart | TIME | Время начала |
| TimeEnd | TIME | Время окончания |
| Duration | INTEGER | Длительность приема |
| Cabinet | NVARCHAR(30) | Кабинет |

Таблица 8 – «Doctor» (доктор)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| Surname | NVARCHAR(50) | Фамилия |
| Name | NVARCHAR(50) | Имя |
| Forename | NVARCHAR(50) | Отчество |
| Phone | NVARCHAR(20) | Телефон |

Таблица 9 – «Specialization» (специализация)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| Title | NVARCHAR(100) | Название |

Таблица 10 – «AspNetUserClaims» (требования пользователя ASP NET)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| ClaimType | TEXT | Тип претензии |
| ClaimValue | TEXT | Значение претензии |

Таблица 11 – «AspNetRoles» (роли в сети)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| ConcurrencyStamp | TEXT | Штамп параллелизма |
| Name | TEXT | Имя |
| NormalizedName | TEXT | Нормализованное имя |

Таблица 12 – «AspNetRoleClaims» (сетевые требования к роли Asp)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| ClaimType | TEXT | Тип претензии |
| ClaimValue | TEXT | Значение претензии |

Таблица 13 – «AspNetUserLogins» (логины пользователей ASP NET)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| ProviderDisplayName | TEXT | Отображаемое имя поставщика |

Таблица 14 – «AspNetUserTokens» (токены пользователей Asp Net)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| Value | TEXT | Значение |

Таблица 15 – «\_\_EFMigrationsHistory» (\_\_история миграций\_EF)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| ProductVersion | TEXT | Версия продукта |

Таблица 16 – «AspNetUsers» (пользователи ASP NET)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| AccessFailedCount | INTEGER | Количество неудачных попыток доступа |
| ConcurrencyStamp | TEXT | Штамп параллелизма |
| Email | TEXT | Электронная почта |
| EmailConfirmed | INTEGER | Подтвержденный адрес электронной почты |
| LockoutEnabled | INTEGER | Включена блокировка |
| LockoutEnd | TEXT | Завершение блокировки |
| NormalizedEmail | TEXT | Нормализованная электронная почта |
| NormalizedUserName | TEXT | Нормализованное имя пользователя |
| PasswordHash | TEXT | Хэш пароля |
| PhoneNumber | TEXT | Номер телефона |
| PhoneNumberConfirmed | INTEGER | Подтвержденный номер телефона |
| SecurityStamp | TEXT | Защитный штамп |
| TwoFactorEnabled | INTEGER | Включен двухфакторный режим |
| UserName | TEXT | Имя пользователя |

Таблица 17 – «Patient» (пациент)

|  |  |  |
| --- | --- | --- |
| **Имя поля** | **Тип данных** | **Комментарий** |
| Surname | NVARCHAR(50) | Фамилия |
| Name | NVARCHAR(50) | Имя |
| Forename | NVARCHAR(50) | Отчество |
| PlaceResidence | NVARCHAR(255) | Место жительства |
| Address | NVARCHAR(100) | Адрес |
| Phone | NVARCHAR(20) | Телефон |
| Email | NVARCHAR(50) | Электронная почта |
| DateOfBirth | DATE | Дата рождения |
| NumberCard | INTENGER | Номер карты |
| MHIpolicy | NVARCHAR(19) | Полюс ОМС |
| VMIpolicy | NVARCHAR(19) | Полюс ДМС |
| PassportSeries | NVARCHAR(5) | Серия паспорта |
| PassportNumber | NVARCHAR(6) | Номер паспорта |
| DateOfIssue | DATE | Дата получения |

Выходная информация – это информация, которая предоставляется или передается системой, или устройством наружу или другой системе, или устройству. Она может быть представлена в различных формах, таких как текст, изображения, звук или видео.

Выходная информация разработанного приложения организована в виде вывода данных о пациенте.

Дополнить базу данных значениями и вставить картинку

Рисунок 4 – Вывод осмотра пациента

Модель базы данных разработана на основе анализа требований заказчика и изучения предметной области – данные пациента и их учет.

Модель базы данных – это абстрактная структура, которая определяет способ организации и хранения данных в базе данных. Она определяет структуру данных, схему таблиц, связей между таблицами и правила целостности данных. Модель базы данных помогает описать и организовать данные для их эффективного использования и обработки. В зависимости от потребностей и особенностей проекта, могут быть использованы различные модели баз данных, такие как реляционная, иерархическая, сетевая, объектно-ориентированная, графовая и др.

Предметная область – область знаний или деятельности, которая имеет определенные характеристики, термины, понятия, методы и принципы, связанные с определенной областью знания или практики. Предметная область может быть узкой (например, физика элементарных частиц) или широкой (например, медицина) и обычно определяет предмет исследования или работы в определенной области. Например, в области информационных технологий предметная область может включать такие области, как программирование, базы данных, сети, компьютерная безопасность и т. д.

В разработанной модели базы данных объектами являются созданные таблицы:



Рисунок 5 – Логическая модель базы данных

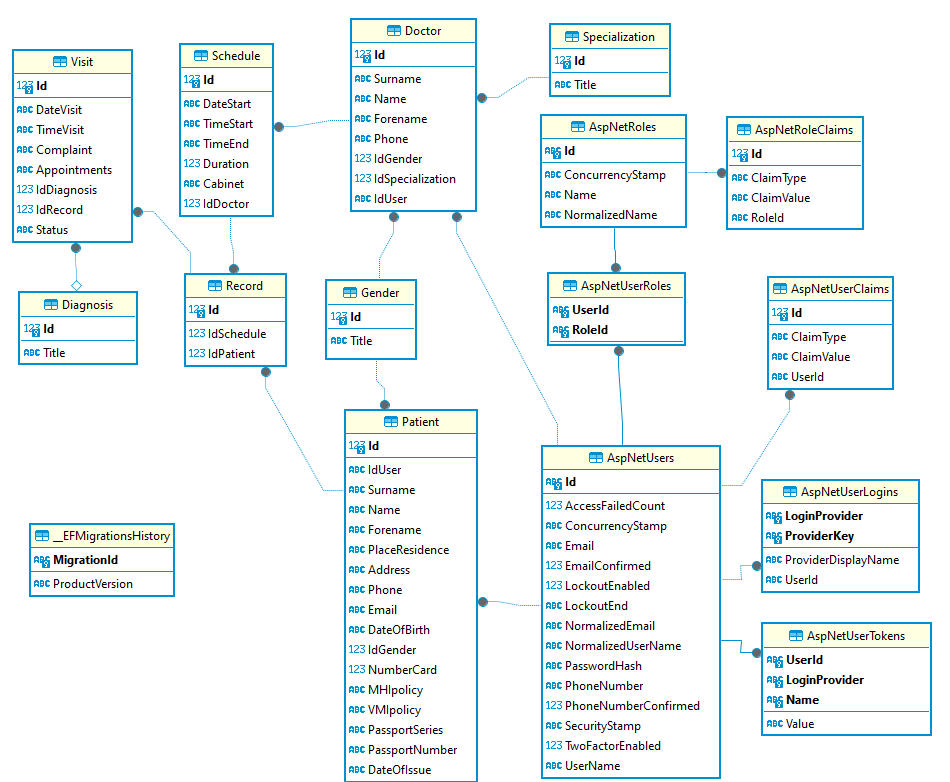


Рисунок 6 – Физическая модель базы данных

Описать входные и выходные данные.

Описать проектирование логической модели БД + предоставить ERD.

Описать сущности БД и связи между ними.

Предоставить физическую модель БД.

Описать нормализацию своей БД.

# **2.3 Проектирование программных модулей решения**

Здесь рисуем диаграммы, блок-схемы модулей, описываем человеческим языком предназначение функций и что они будут решать.

# **2.4 Реализация технического решения**

# **2.4.1 Разработка пользовательского интерфейса решения**

Описание интерфейса программного решения/веб-ресурса (необходимо пошагово описать формы).

Здесь вставляем скрины реализованного приложения (интерфейс); всплывающие окна.

Для вебов: вставляем скрины разработанного сайта, popup и подобные окна для взаимодействия с пользователем, также формы.

# **2.4.2 Разработка программных модулей решения**

Вставляем несколько важных функций, как они реализованы и что делают, приводим пример работы пользователя с функциями.

# **2.5 Тестирование разработанного решения**

Описание тестовых сценариев // описываем сценарии по типу: входные данные, ожидаемый результат, полученный результат.

Обработка исключений, проверка корректности введенных данных, обратная связь с пользователем.

# **3 Организационно-экономическая часть**

## **3.1 Расчет сметы затрат на разработку программы**

# **3.1.1 Составление проекта выполнения работ**

Для расчета сметы затрат составлен проект выполнения работ по созданию программы. Он представляет собой перечень мероприятий, которые необходимо выполнить, чтобы разработать и внедрить программу «Автоматизации учёта данных регистратуры поликлиники». Работы перечислены в требуемой последовательности с установленной продолжительностью каждого этапа, данные в таблице 1.

Таблица 1 – Проект выполнения работ по созданию программы.

|  |  |
| --- | --- |
| **Наименование этапов** | **Продолжительность, дни** |
| Получение задания на разработку программы | 1 |
| Сбор и изучение требований заказчика, анализ предметной области | 2 |
| Проектирование концептуальной модели | 2 |
| Проектирование графического интерфейса программного приложения | 4 |
| Разработка функциональных возможностей программы | 6 |
| Тестирование программы | 2 |
| Отладка программного продукта и встреча с заказчиком | 2 |
| Эксплуатация, сдача проекта заказчику | 1 |

Общие затраты времени на разработку программы определены как сумма продолжительности работ и составляют 20 дней.

# **3.1.2 Расчет материальных затрат**

В составе материальных затрат по разработке программы отражена стоимость:

* приобретаемых материалов, которые являются необходимым компонентом при проведении работ;
* покупной энергии, расходуемой на производственные и хозяйственные нужды.

Рассчитываем затраты на эксплуатационные материалы, исходные данные представлены в таблице 2, в ней же и результаты расчетов.

Таблица 2 – Затраты на эксплуатационные материалы.

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование материала** | **Количество** | **Цена, руб.** | **Сумма, руб.** |
| Диск (шт.) | 1 | 25 | 25 |
| Интернет (дней) | 20 | 23 | 460 |
| Бумага (шт.) | 20 | 0,6 | 12 |
| Ручка (шт.) | 1 | 5 | 5 |
| Итого |  |  | См = 502 руб. |

См – сумма затрат на эксплуатационные материалы.

Расчёт стоимости электроэнергии.

ЭВМ будет использоваться 20 дней по 8 часов в день, то есть 160 часов.

Исходные данные:

* потребляемая мощность – 0,30 кВт/ч;
* время работы на ЭВМ – 160 ч;
* тариф по электроэнергии – 3.9 руб. /кВт

Рассчитываем стоимость электроэнергии (Сэл.) по формуле:

Сэл. = P × tраб. × Ц, руб.,

где P – потребляемая мощность, кВт/ч;

tраб. – время работы на ЭВМ, ч;

Ц – цена за 1 кВт/ч, руб.

Сэл. = 0.30 × 160 × 3.9 = 187,2 руб.

Рассчитываем сумму материальных затрат (См.з.):

См.з. = См. + Сэл., руб.,

См.з. = 502 + 187,2 = 689,2 руб.

# **3.1.3 Расчет затрат на оплату труда**

Для оплаты труда программиста чаше всего используется повременная зарплата.

Затраты на оплату труда включают:

* заработную плату программиста;
* оплату отпуска программиста за отработанное время.

Расчет заработной платы программиста.

Исходные данные:

* время работы над программой, 20 дн.;
* должностной оклад за месяц, 20000 руб.;
* количество рабочих дней за месяц, 22 дн.;
* процент премии, 50 %;
* районный коэффициент, 1,15.

Рассчитываем заработную плату по тарифу:

где Зпл.по тар.- заработная плата по тарифу, руб;

T – время работы над программой, дн.;

ДО – должностной оклад за месяц, руб.;

tм-ц – количество дней за месяц, дн.

Рассчитываем сумму премии:

где ∑прем. – сумма премии, руб.;

%Пр. – процент премии.

Рассчитываем сумму зарплаты за отработанное время:

. руб.

Рассчитываем сумму зарплаты с районным коэффициентом:

,

где с районным коэффициентом, руб.,

РК – районный коэффициент, 1,15

Для расчета очередного отпуска определяем дневную зарплату (Дн.з.пл.):

Исходные данные:

* дневная зарплата, 1568,18 руб.;
* средняя продолжительность отпуска, 20 рабочих дн.;
* годовой фонд рабочего времени, 247 дн.;
* время работы над программой, 20 дн.

Рассчитываем сумму оплаты за отпуск:

где ∑оп.от. – сумма оплаты отпуска, руб.;

tот. – продолжительность отпусков, дн.;

Фр.в. – годовой фонд рабочего времени, дн.;

Т – время работы над программой, дн.;

Дн.з.пл. – дневная заработная плата, руб.;

Рассчитываем сумму затрат на оплату труда (Зот.):

, руб.

# **3.1.4 Расчет амортизационных отчислений**

Для разработки программы «Автоматизации учёта данных регистратуры поликлиники» использовалась вычислительная техника в виде персонального компьютера, в который входит монитор и системный блок. Вычислительная техника входит в состав основных производственных фондов.

Основные производственные фонды – это совокупность средств труда, функционирующих в сфере материального производства в неизменной натуральной форме в течение длительного времени и переносящих свою стоимость на вновь созданный продукт по частям, по мере изнашивания.

Амортизационные отчисления – это постепенное погашение стоимости основных фондов предприятия (зданий, оборудования, подвижного состава и т.п.), изнашивающихся в процессе работы и от времени.

Рассчитываем амортизационные отчисления за год, исходные данные и результаты расчётов в таблице 3.

Таблица 3 - Амортизационные отчисления за год

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование основных производственных фондов** | **Стоимость ОПФ, руб.** | **Норма амортизации, %** | **Сумма амортизации, руб.** |
| Системный блок | 100000 | 17 | 17000 |
| Монитор | 32000 | 17 | 5440 |
| Итого |  |  | Ar = 22440 |

Ar - годовая сумма амортизационных отчислений.

Годовая сумма амортизационных отчислений (Ar) рассчитывается по формуле:

где Cn – стоимость первоначальная, руб;

Ha – норма амортизации, %

Рассчитываем годовую сумма амортизационных отчислений системного блока, Aг с.б.:

Рассчитываем годовую сумму амортизационных отчислений монитора, Aг м.:

Рассчитываем сумму амортизационных отчислений за время разработки программы:

где Ar – годовая сумма амортизационных отчислений, руб.;

n – количество дней на разработку программы;

365 – количество дней в году.

# **3.1.5 Расчет стоимости разрабатываемой программы**

Себестоимость — стоимостная оценка текущих затрат на производство и реализацию продукции.

Себестоимость продукции формируется из следующих элементов

* материальные затраты;
* затраты на оплату труда;
* отчисления на социальные нужды;
* амортизация основных фондов;
* накладные расходы.

В элементе «отчисления на социальные нужды» отражаются обязательные отчисления органам государственного и негосударственного социального страхования, Пенсионного фонда, Государственного фонда занятости и медицинского страхования, в процентах к оплате труда работников.

Общий процент отчислений в фонды остается прежним по сравнению с предыдущим 2023 годом – 30%, из них:

* 22% в пенсионный фонд;
* 2,9% в фонд социального страхования;
* 5,1% в фонд медицинского страхования.

Накладные расходы — это налоги, сборы, отчисления в специальные фонды, платежи по кредитам, затраты на командировки, оплата услуг связи и другие.

Рассчитываем себестоимость разрабатываемой программы по формуле:

руб.,

где С/С – себестоимость, руб.;

См.з. – материальные затраты, руб.;

З.о.т. – затраты га оплату труда, руб.;

А – амортизационные отчисления, руб.;

От. – отчисления на социальные нужды, руб.;

НР – накладные расходы, руб.

Рассчитываем отчисления на социальные нужды.

Исходные данные:

* затраты на оплату труда -руб.;
* установленный размер отчислений - 36 %.

где %отчисл. – процент отчислений, %

От. – отчисления на социальные нужды, руб.

Рассчитываем накладные расходы (80% от основной заработной платы)

Рассчитываем себестоимость.

Исходные данные:

См.з – материальные затраты, 871,45 руб.;

З.о.т. – затраты на оплату труда, руб.;

А – амортизационные отчисления, руб.;

От. – отчисления на социальные нужды, руб.;

НР – прочие расходы (накладные расходы), руб.

Определяем структуру себестоимости разрабатываемой программы.

Структура – процентное содержание элементов затрат к общей сумме себестоимости.

Таблица 4 – Структура себестоимости

|  |  |  |
| --- | --- | --- |
| **Наименование статей затрат** | **Сумма, руб.** | **Структура, %** |
| Материальные затраты | 871,45 | 1,19 |
| Затраты на оплату труда | 33903,2 | 46,25 |
| Отчисления на социальные нужды | 10170,96 | 13,88 |
| Амортизационные отчисления | 1229,6 | 1,68 |
| Накладные расходы | 27122,56 | 37 |
| Итого: | 73297,77 | 100 |

Вывод: самую большую долю затрат на разработку программы составляет 46,25% затраты на оплату труда, а самую малую долю – 1,19% материальные затраты.

# **3.2 Расчет экономической эффективности внедрения программы**

Экономическая эффективность – это соотношение результатов с затратами.

Если разрабатываемая программа пользуется спросом, то возможна ее реализация. В этом случае эффективность определяется возможной прибылью от реализации разработанной программы.

Для этого необходимо установить цену на разработанную программу.

При установлении цены используют полную сбытовую себестоимость товарной продукции, она включает производственную себестоимость и внепроизводственные расходы.

Внепроизводственные (коммерческие) расходы включают затраты, связанные с реализацией продукции (расфасовка, упаковка, отгрузка, реклама, маркетинговые исследования, комиссионные; хранение, транспортировка), а также различного рода отчисления и платежи.

Полная себестоимость рассчитывается по формуле:

где С/Ск.– полная себестоимость единицы продукции, руб;

С/С – себестоимость единицы продукции производственная, руб;

Вн.р. – внепроизводственные расходы (7-10% от производственной себестоимости), руб.

Исходные данные:

* внепроизводственные расходы – руб. (8 % от производственной себестоимости);
* производственная стоимость – руб.

Оптовую цену предприятия можно определить расчетным путем, используя уровень рентабельности продукции.

Уровень рентабельности продукции устанавливают в зависимости от величины прибыли (20-40% от коммерческой (полной) себестоимости единицы продукции).

Рассчитываем цену разрабатываемой программы по формуле:

где Ц – цена разрабатываемой программы, руб;

С/Ск. – себестоимость коммерческая (полная) разрабатываемой программы, руб.;

Pnp. – рентабельность продукции, %.

Исходные данные:

* рентабельность продукции – 20 %;
* полная себестоимость продукции – руб.

Расчет прибыли от реализации программы.

Прибыль от реализации продукции (работ, услуг) определяется как разница между выручкой от реализации продукции (работ, услуг) в действующих ценах (без налога на добавленную стоимость, акцизного налога) и затратами на ее производство и реализацию, включаемыми в себестоимость продукции (работ, услуг).

Рассчитываем прибыль от реализации разрабатываемой программы:

где Пр. – прибыль от реализации программы, руб.

# **4 Мероприятия по технике безопасности и охране окружающей среды**

# **4.1 Организация безопасности труда на предприятии**

Описать меры безопасности труда, используемые на предприятии.

# **4.2 Техника безопасности при работе с вычислительной техникой**

Описать технику безопасности при работе с ВТ (на предприятии).

# **Заключение**

В соответствии с полученным заданием, была достигнута цель дипломного проекта: автоматизация учета данных регистратуры поликлиники.

В ходе работы над дипломным проектом были выполнены следующие задачи:

1. Изучены требования заказчика и проведен анализ предметной области — данные пациента и их учет.
2. Построена модель базы данных.
3. Создан пользовательский интерфейс электронной картотеки.
4. Реализованы функциональные возможности электронной системы.
5. Проведено тестирование и выполнена отладка приложения для обеспечения его корректной работы и в соответствии со всеми требованиями.
6. Оформлена техническая документация в виде пояснительной записки на курсовой проект.

Итогом дипломного проекта стала разработанная система автоматизация учета данных регистратуры поликлиники с помощью языка программирования C#, системы управления базами данных SQLiteStudio и среды разработки Microsoft Visual Studio и Radzen Blazor Studio.

.

# **Список использованных источников**

1. Database setup | Create a Blazor CRM application with Radzen Blazor Studio. URL: https://www.radzen.com/blazor-studio/documentation/crm/database/ (Дата обращения 20.04.2024).
2. Basic CRUD operations in Blazor using SQLite as the database. URL: https://executecommands.com/crud-in-blazor-using-sqlite-entity-framework/ (Дата обращения 20.04.2024).
3. Первое приложение на Blazor. URL: https://metanit.com/sharp/blazor/1.2.php (Дата обращения 20.04.2024).
4. Radzen Blazor Components. URL: https://blazor.radzen.com/ (Дата обращения 20.04.2024).
5. Дружинина, И.В. Информационные технологии в профессиональной деятельности средних медицинских работников: учебное пособие / И.В. Дружинина. – СПб.: Лань, 2018. – 122с.: ил. – ISBN 978-5-8114-7186-7.
6. Кислицын, Е.В. Современные технологии разработки программного обеспечения: учеб. пособие / Е. В. Кислицын, М. А. Панов. - Екатеринбург: Изд-во УрГЭУ, 2021. - 176 с.: ил. - ISBN 978-5-8111-7114-5.
7. Часовских, В.П. Управление ИТ-разработкой и внедрением: учеб. пособие / В. П. Часовских, М. П. Воронов, В. Г. Лабунец, Е. Н. Стариков. - Екатеринбург: Изд-во УрГЭУ, 2021. - 173 с.: ил. - ISBN 978-5-8458-1812-5.
8. Кэмпбелл, Л. Базы данных. Инжиниринг надежности: научная литература / Л. Кэмпбелл, Ч. Мейджорс – СПб.: Питер, 2020. – 304 с.: ил. – ISBN 978-5-4461-1310-8.
9. ASP.NET Core Blazor. URL: https://learn.microsoft.com/ru-ru/aspnet/core/blazor/?view=aspnetcore-5.0 (Дата обращения 19.04.2024).
10. Как использовать Entity Framework Core в приложении Blazor. URL: <https://www.c-sharpcorner.com/article/how-to-use-entity-framework-core-in-blazor-application/> (Дата обращения: 19.04.2024).
11. Паттерны проектирования в C#. URL: <https://refactoring.guru/ru/design-patterns/csharp> (Дата обращения: 19.04.2024).
12. C# 9.0 in a Nutshell: The Definitive Reference. URL: https://www.amazon.com/C-9-0-Nutshell-Definitive-Reference/dp/1098100964 (Дата обращения 21.04.2024).
13. Руководство по ASP.NET Core 5. URL: https://metanit.com/sharp/aspnet5/ (Дата обращения 20.04.2024).
14. Создание веб-API с помощью ASP.Net Core. URL: https://learn.microsoft.com/ru-ru/aspnet/core/web-api/?view=aspnetcore-5.0 (Дата обращения 22.04.2024).
15. Синтаксис LINQ. URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/linq/ (Дата обращения 22.04.2024).
16. Атрибуты. URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/advanced-topics/reflection-and-attributes/ (Дата обращения 22.04.2024).
17. Асинхронное программирование с async и await. URL: https://learn.microsoft.com/ru-ru/dotnet/csharp/asynchronous-programming/ (Дата обращения 20.04.2024).
18. Делегаты (Руководство по программированию на C#). URl: https://learn.microsoft.com/ru-ru/dotnet/csharp/programming-guide/delegates/ (Дата обращения 19.04.2024).
19. Blazor in Action. URL: https://www.manning.com/books/blazor-in-action (Дата обращения 19.04.2024).
20. Blazor. URL: https://www.c-sharpcorner.com/technologies/blazor (Дата обращения 19.04.2024).