Департамент образования и науки Костромской области

Областное государственное бюджетное профессиональное образовательное учреждение «Костромской энергетический техникум имени Ф.В. Чижова»

(ОГБПОУ «КЭТ имени Ф.В. Чижова»)

КУРСОВАЯ РАБОТА

Разработка информационной системы для учета лекарственных средств в аптеке

ПМ.06 Сопровождение информационных систем

МДК.06.03 Устройство и функционирование информационной системы

**специальность:** 09.02.07 «Информационные системы и программирование»

Студент **Михайлова Анастасия Валерьевна** \_\_.\_\_.2022 г.

Оценка выполнения и защиты курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_

Руководитель Бессараб С.К. **\_\_\_\_\_\_\_\_\_\_\_\_** \_\_.\_\_.20\_\_ г.

*Подпись*

Кострома 2022

**ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ КОСТРОМСКОЙ ОБЛАСТИ**

**ОБЛАСТНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**«КОСТРОМСКОЙ ЭНЕРГЕТИЧЕСКИЙ ТЕХНИКУМ ИМ. Ф.В. ЧИЖОВА»**

**специальность 09.02.07 «Информационные системы и программирование»**

|  |  |  |
| --- | --- | --- |
|  | СОГЛАСОВАНО | **УТВЕРЖДАЮ:** |
|  | И.о.заместителя руководителя  Верхне-Волжского Управления Ростехнадзора  (Костря область) | Зам. директора по учебной работе  \_\_\_\_\_\_\_\_\_\_\_(Е.И.Тюрина)  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_20 г. |

**КУРСОВОЙ ПРОЕКТ**

**ЗАДАНИЕ**

**Студента 1 группы 3 курса**

**Михайловой Анастасии Валерьевной**

**2022 год**

Тема курсового проекта:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Содержание пояснительной записки

ТИТУЛЬНЫЙ ЛИСТ

ОГЛАВЛЕНИЕ

ГЛОССАРИЙ

ВВЕДЕНИЕ

ГЛАВА 1. ОБОСНОВАНИЕ ОБЪЕКТА АВТОМАТИЗАЦИИ

Краткие сведения об информационной системе

Общие требования к информационной системе

Разработка технического задания на разработку информационной системы

Обоснование необходимости внедрения и использования информационной системы

ГЛАВА 2. РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1. Физическая структура базы данных

ГЛАВА 3. РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1. Разработка кода информационной системы

ГЛАВА 4. ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

4.1. План тестирования

4.2. Написание тест-кейсов

4.3. Юнит-тестирование

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

СПИСОК ИЛЛЮСТРАЦИЙ

ПРИЛОЖЕНИЯ

Приложение №1 Инструкции для использования системы (руководство пользователя);

Приложение №2 Техническое задание;

Приложение №3 ER-ДИАГРАММА;

Приложение №4 ЛИСТИНГ КОДА;

Приложение №5 Ссылка на облачно хранилище и репозиторий с курсовой работой.

1. Графическая часть
2. ER-ДИАГРАММА;
3. ЛИСТИНГ КОДА;
4. Электронная часть

Содержимое папки с курсовым проектом в облачном хранилище (на репозитории) и на электронном носителе:

1. Папка - Практическая часть (информационная система, исходный код заархивированный; скрипт базы данных);

2. Файл - Текст курсовой работы (в формате .docx) и распечатать. Объем 30 страниц без приложения;

3. Файл - Приложения (техническое задание, диаграммы, скриншоты основных окон ИС, листинг кода);

4. Папка - Презентация и доклад (время на выступление максимум 5 -7 мин, 15 слайдов максимум);

5. Файл - Видео (с демонстрацией работы основного функционала не больше 1 мин (видео в формате mp4 объем до 10 Мб);

6.Файл - readme.txt (внутри: ФИО, тема курсовой работы, ссылка на облачное хранилище, ссылка на репозиторий);

Задание получил\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи задания: \_\_\_\_\_\_\_

Срок выполнения: \_\_\_\_\_\_\_\_\_\_

Руководитель курсового проекта: преподаватель \_\_\_\_\_\_\_\_\_ (Бессараб С.К.)

Рассмотрено и утверждено на заседании ПЦК \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Председатель ПЦК \_\_\_\_\_\_\_\_\_(Ищук О.В.)

**ОГЛАВЛЕНИЕ**

[**ГЛОССАРИЙ** 6](#_heading=h.1fob9te)

[**ВВЕДЕНИЕ** 7](#_heading=h.3znysh7)

[**ГЛАВА 1.**](#_heading=h.tyjcwt) **ОБОСНОВАНИЕ ОБЪЕКТА АВТОМАТИЗАЦИИ** 9

[1.1.](#_heading=h.3dy6vkm) Краткие сведения об информационной системе 9

[1.2.](#_heading=h.1t3h5sf) Общие требования к информационной системе 10

[1.3.](#_heading=h.4d34og8) Разработка технического задания на разработку информационной системы 11

[1.4.](#_heading=h.z337ya) Обоснование необходимости внедрения и использования информационной системы 13

[**ГЛАВА 2.**](#_heading=h.3j2qqm3) **РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ** 14

[2.1](#_heading=h.32hioqz) Физическая структура базы данных 17

[**ГЛАВА 3.**](#_heading=h.vx1227) **РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ** 19

[3.1.](#_heading=h.3fwokq0) Разработка кода информационной системы 19

[**ГЛАВА 4.**](#_heading=h.3tbugp1) **ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ** 23

[4.1](#_heading=h.28h4qwu) План тестирования 24

[4.2](#_heading=h.46r0co2) Написание тест-кейсов 25

[4.3](#_heading=h.2lwamvv) Юнит-тестирование 31

[**ЗАКЛЮЧЕНИЕ** 35](#_heading=h.206ipza)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 36](#_heading=h.4k668n3)

[**СПИСОК ИЛЛЮСТРАЦИЙ** 38](#_heading=h.2zbgiuw)

[**ПРИЛОЖЕНИЯ** 39](#_heading=h.1egqt2p)

[Приложение №1 39](#_heading=h.3ygebqi)

[Приложение №2 40](#_heading=h.sqyw64)

[Приложение №3 47](#_heading=h.34g0dwd)

[Приложение №4 48](#_heading=h.1jlao46)

[Приложение №5 50](#_heading=h.2iq8gzs)

**ГЛОССАРИЙ**

Аптека – особая специализированная организация системы здравоохранения, занимающаяся изготовлением, фасовкой, анализом и продажей лекарственных средств. [4]

Лекарственное средство, препараты, медикаменты – это вещество или комплекс веществ синтетического или природного происхождения в определенной лекарственной форме, применяемое для профилактики, лечения заболеваний, для диагностики и ухода. [20]

Медицина – область профессиональной практической и научной деятельности, имеющая своей целью распознавание, лечение и предупреждение болезней, сохранение и укрепление здоровья и трудоспособности, продление жизни людей. [2]

Фармацевт или аптекарь — это специалист, занимающийся реализацией, хранением и подготовкой лекарственных препаратов. [1]

**ВВЕДЕНИЕ**

Информационная система – взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели [8].

Информационная система представляет собой хранилище информации, снабженное процедурами ввода, поиска, размещения и выдачи информации. Наличие таких процедур – главная особенность информационных систем, отличающих их от простых скоплений информационных материалов. [6]

Современное понимание информационной системы предполагает использование в качестве основного технического средства переработки информации персонального компьютера. Кроме того, техническое воплощение информационной системы само по себе ничего не будет значить, если не учтена роль человека, для которого предназначена производимая информация и без которого невозможно ее получение и представление.

Процесс создания ИС делится на ряд этапов, ограниченных некоторыми временными рамками и заканчивающихся выпуском конкретного продукта (моделей, программных продуктов, документации и пр.). [8]

**Объектом** исследования является деятельность аптеки.

**Предметом** исследования является автоматизация процессов учета лекарств.

**Цель** в данной работе заключается в разработке настольного приложения для работы с лекарствами в аптеке.

**Задачи:**

1. Исследовать предметную область
2. Составить техническое задание
3. Спроектировать базу данных
4. Разработать систему
5. Протестировать систему

Таким образом была разработана система для аптеки.

1. **ОБОСНОВАНИЕ ОБЪЕКТА АВТОМАТИЗАЦИИ**

## Краткие сведения об информационной системе

В данной курсовой работе мы разработали информационную систему под названием «Quick help», связанную с медициной. Она представляет работу с лекарствами в аптеке, а именно списание и добавление. Так же все лекарства в аптеке изображены в виде таблицы 1, на основании которой можно составлять отчёты, просматривать другие отчёты и изменения, внесенные другими пользователями. Информационная система предназначена для учёта лекарственных средств в аптеке.

Таблица 1

Краткие сведения об информационной системе

|  |  |
| --- | --- |
| **Название** | **Описание** |
| Название компании (организации, проекта) | Quick help |
| Цели создания приложения | Автоматизировать учёт лекарств в аптеке |
| Целевая аудитория  Пол, возраст  Образование  Чем увлекаются  Уровень дохода  Какая услуга в приложении поможет удовлетворить потребности | Мужчины, женщины, от 20 до 50 лет  Среднее профессиональное образование или высшее медицинское  Учёт лекарств в аптеке |
| Информация об услугах | Отчёт об имеющихся препаратов  Списание медикаментов  Отчёт о закупках  Направления:  Работа с лекарственными средствами |
| Конкуренты в сфере деятельности (ссылки) | <https://planetazdorovo.ru/>  <https://www.eapteka.ru/kostroma/> |

## Общие требования к информационной системе

Информационная система должна отвечать следующим требованиям:

1. Гибкость. Способность к адаптации и дальнейшему развитию подразумевают возможность приспособления информационной системы к новым условиям, новым потребностям предприятия.
2. Надежность. Требование надёжности обеспечивается созданием резервных копий хранимой информации, выполнения операций протоколирования, поддержанием качества каналов связи и физических носителей информации, использованием современных программных и аппаратных средств.
3. Эффективность. Система является эффективной, если с учётом выделенных ей ресурсов она позволяет решать возложенные на неё задачи в минимальные сроки. Эффективность системы обеспечивается оптимизацией данных и методов их обработки, применением оригинальных разработок, идей, методов проектирования.
4. Безопасность. Под безопасностью, прежде всего, подразумевается свойство системы, в силу которого посторонние лица не имеют доступа к информационным ресурсам организации, кроме тех, которые для них предназначены. Требование безопасности обеспечивается современными средствами разработки информационных систем, современной аппаратурой, методами защиты информации, применением паролей и протоколированием, постоянным мониторингом состояния безопасности операционных систем и средств их защиты. [18]

Функциональные требования к системе:

1. Внести изменения в лекарства
2. Списание лекарств
3. Добавление лекарств

## Разработка технического задания на разработку информационной системы

Техническое задание, или ТЗ — это документ, в котором фиксируются требования к проекту. Условно ТЗ можно назвать любое поручение исполнителю, главное, чтобы в нем были ясно прописаны характеристики итогового продукта.

Несмотря на всю свою важность, содержание ТЗ мало регламентировано нормативными документами. Требования к содержанию и порядку построения документа зависят от направления разработки и содержатся в следующих документах:

ГОСТ 15.201-2000. Система разработки и постановки продукции на производство (СРПП). Продукция производственно-технического назначения. Порядок разработки и постановки продукции на производство (приведены общие требования и краткие рекомендации по разработке).

ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению (кратко изложено содержание ТЗ);

ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы (достаточно подробно изложены состав и содержание ТЗ);

ГОСТ 25123-82. Машины вычислительные и системы обработки данных. Техническое задание. Порядок построения, изложения и оформления (приведен порядок построения ТЗ).

Обобщая требования этих стандартов, порядок построения, изложения и оформления ТЗ можно свести к последовательности, представленной в таблице ниже. [17]

**Таблица 2**

**Этапы работы технического задания**

|  |  |
| --- | --- |
| **Этап работы** | **Содержание работы** |
| Обоснование необходимости разработки программы | 1. Постановка задачи 2. Сбор исходных материалов 3. Выбор и обоснование критериев эффективности и качества разрабатываемой программы. 4. Обоснование необходимости проведения научно-исследовательских работ. |
| Научно-исследовательские работы | Определение структуры входных и выходных данных.  Предварительный выбор методов решения задач.  Обоснование целесообразности применения ранее разработанных программ.  Определение требований к техническим средствам.  Обоснование принципиальной возможности решения поставленной задачи |
| Разработка и утверждение технического задания | Определение требований к программе.  Разработка технико-экономического обоснования разработки программы.  Определение стадий, этапов и сроков разработки программы и документации на неё.  Выбор языков программирования.  Определение необходимости проведения научно-исследовательских работ на последующих стадиях.  Согласование и утверждение технического задания. |

На основании представленного анализа мы разработали техническое задание, которое представлено в приложении №2

## Обоснование необходимости внедрения и использования информационной системы

Система учёта лекарственных средств помогает в работе сотрудников, а именно:

1. Ведение учета по торговым наименованиям по каждому лекарственному средству отдельно, с указанием его названия, формы выпуска (таблетки, ампулы и т.д.), информации об упаковке (коробки, флаконы, тубы и т.п.), названия производителя лекарственного средства, количества, цены;
2. Контроль срока годности медикаментов, предоставляет возможность планирования закупок и контроля их рационального использования;
3. Возможность определения количества конкретного препарата в аптеке;
4. Возможность списания любого количества лекарственного средства.

Эффективность системы учета заключается в возможности планирования закупок медикаментов на любой период (1 год или 2 мес. и т.п.) на основании анализа среднесуточного использования препаратов и их остатков.

На основании технического задания принято решение разработать настолько приложение, так как система локальная.

1. **РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных (СУБД). Данные вместе с СУБД, а также приложения, которые с ними связаны, называются системой баз данных, или, для краткости, просто базой данных.

Данные в наиболее распространенных типах современных баз данных обычно хранятся в виде строк и столбцов формирующих таблицу. Этими данными можно легко управлять, изменять, обновлять, контролировать и упорядочивать. В большинстве баз данных для записи и запросов данных используется язык структурированных запросов (SQL). SQL — это язык программирования, используемый в большинстве реляционных баз данных для запросов, обработки и определения данных, а также контроля доступа. SQL был разработан в IBM в 1970-х годах. Со временем у стандарта SQL ANSI появились многочисленные расширения, разработанные такими компаниями как IBM, Oracle и Microsoft. Хотя в настоящее время SQL все еще широко используется, начали появляться новые языки программирования запросов.

Существует множество различных типов баз данных. Выбор наилучшей базы данных для конкретной компании зависит от того, как она намеревается использовать данные.

1. Реляционные базы данных

[Реляционные базы данных](https://www.oracle.com/cis/database/what-is-a-relational-database/) стали преобладать в 1980-х годах. Данные в реляционной базе организованы в виде таблиц, состоящих из столбцов и строк. Реляционная СУБД обеспечивает быстрый и эффективный доступ к структурированной информации.

1. Объектно-ориентированные базы данных

Информация в объектно-ориентированной базе данных представлена в форме объекта, как в объектно-ориентированном программировании.

1. Распределенные базы данных

Распределенная база данных состоит из двух или более частей, расположенных на разных серверах. Такая база данных может храниться на нескольких компьютерах.

Будучи централизованным репозиторием для данных, хранилище данных представляет собой тип базы данных, специально предназначенной для быстрого выполнения запросов и анализа.

База данных [NoSQL](https://www.oracle.com/cis/database/nosql-cloud.html), или нереляционная база данных, дает возможность хранить и обрабатывать неструктурированные или слабоструктурированные данные (в отличие от реляционной базы данных, задающей структуру содержащихся в ней данных). Популярность баз данных NoSQL растет по мере распространения и усложнения веб-приложений.

1. Графовые базы данных

Графовая база данных хранит данные в контексте сущностей и связей между сущностями.

База данных OLTP — это база данных предназначенная для выполнения бизнес-транзакций, выполняемых множеством пользователей.

Это лишь некоторые из десятков типов баз данных, используемых в настоящее время. Другие, менее распространенные базы данных, предназначены для очень специфических научных, финансовых и иных задач. Помимо появления новых типов, базы данных развиваются в абсолютно новых направлениях — изменяются подходы к разработке технологий, происходят значительные сдвиги, такие как внедрение облачных технологий и автоматизации. В частности, в последнее время появились следующие базы данных.

Базы данных с открытым исходным кодомимеют открытый исходный код и могут управляться средствами как SQL, так и NoSQL.

[Облачная база данных](https://www.oracle.com/cis/database/what-is-a-cloud-database/) представляет собой набор структурированных или неструктурированных данных, размещенный на частной, публичной или гибридной платформе облачных вычислений. Существует два типа моделей облачных баз данных: традиционная база данных и база данных как услуга (DBaaS). В модели DBaaS административные задачи и обслуживание выполняются поставщиком облачных услуг.

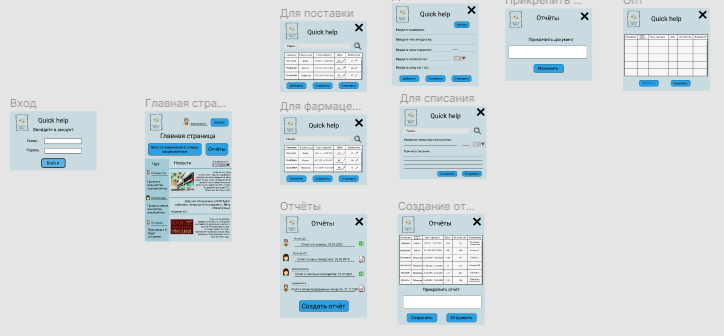
1. Многомодельная база данных объединяет разные типы моделей баз данных в единую интегрированную серверную СУБД. Это означает, что она может содержать различные типы данных.

Базы данных документов/JSON предназначены для хранения, извлечения и обработки документоориентированной информации и предоставляют современный способ хранения данных в формате JSON, а не в виде строк и столбцов.

1. Автономные базы данных

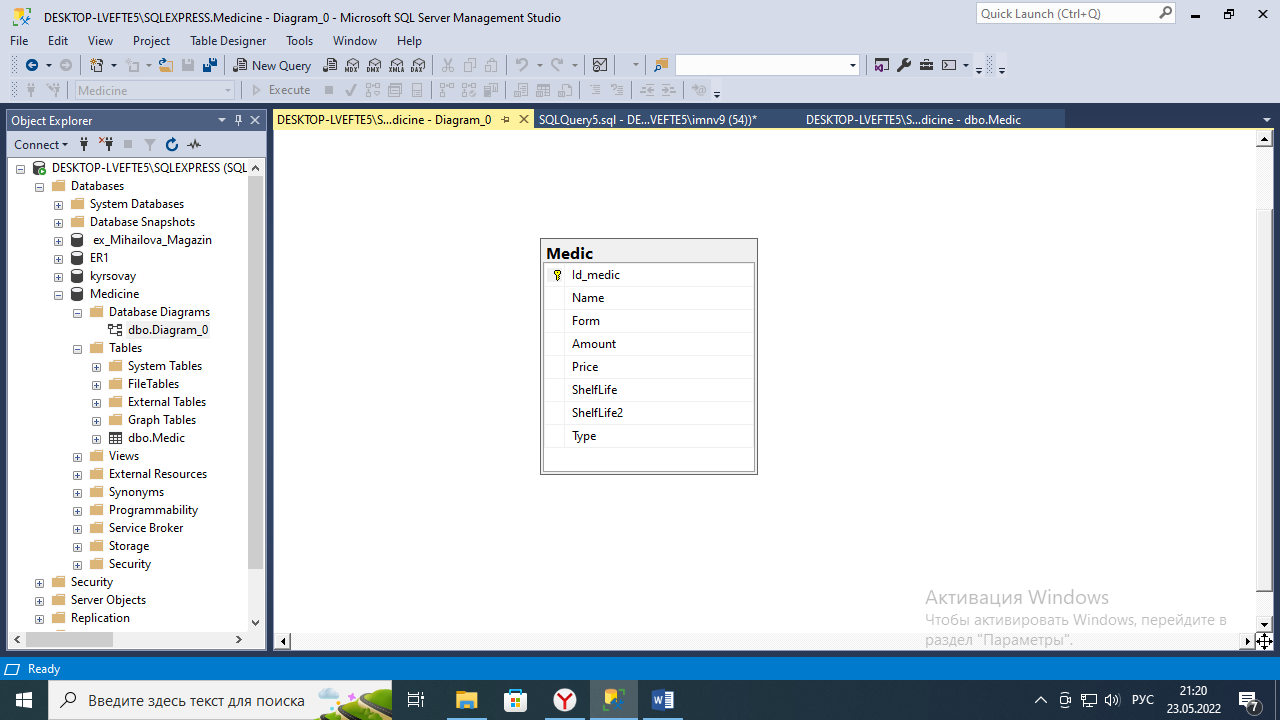
Самоуправляемые базы данных (также называемые автономными) — это новейшие и самые революционные облачные базы данных, которые используют машинное обучение для автоматизации настройки, защиты, резервного копирования, обновления и других стандартных задач обслуживания, обычно выполняемых администраторами баз данных. [20]

Перед тем как разрабатывать приложение был разработан прототип на рисунке 1.



**Рисунок 1. Прототип приложения**

В связи с невозможностью реализовать все функции, принято решение остановиться на данном варианте ERD модели базы данных ИС представленной на рисунке 2. База данных ИС «Quick help» содержит 1 таблицу.



**Рисунок 2. ER-диаграмма**

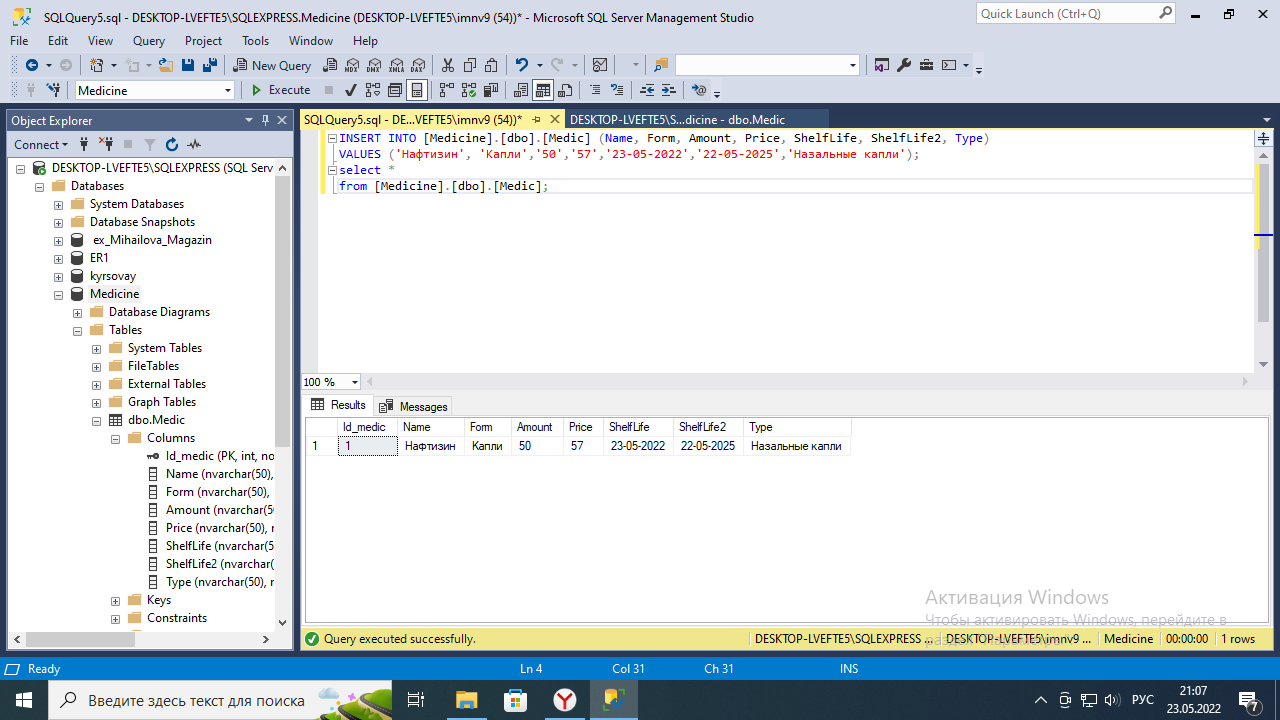
## Физическая структура базы данных

Физическая модель базы данных — это модель данных, которая определяет, каким образом представляются данные, и содержит все детали, необходимые СУБД для создания базы данных.

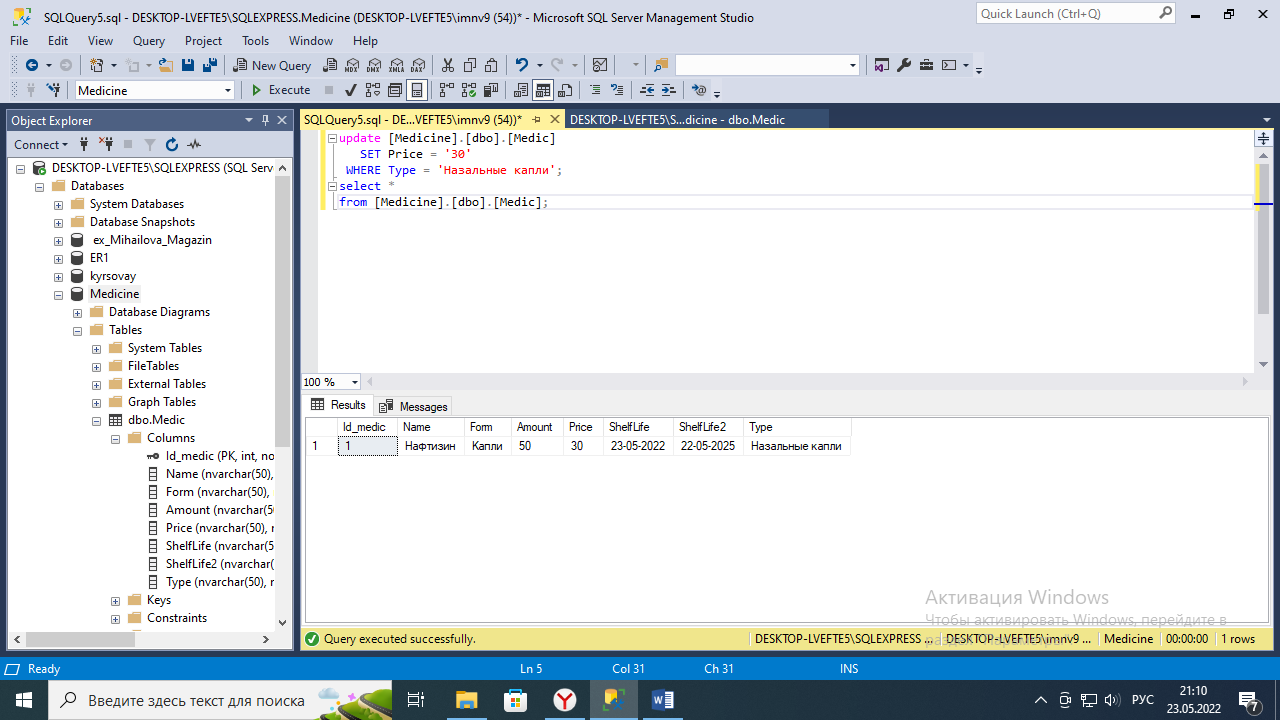
Физическая модель строится на основе логической модели, путем создания отдельной таблицы для каждой сущности. Связи между таблицами создаются так же, как на логическом уровне.

На физическом уровне объекты базы данных (таблицы, колонки и т.д.) должны называться, как этого требуют ограничения выбранной системы управления базой данных (СУБД). Физическая модель зависит от конкретной СУБД, поэтому одной и той же логической модели может соответствовать несколько физических моделей. [19]

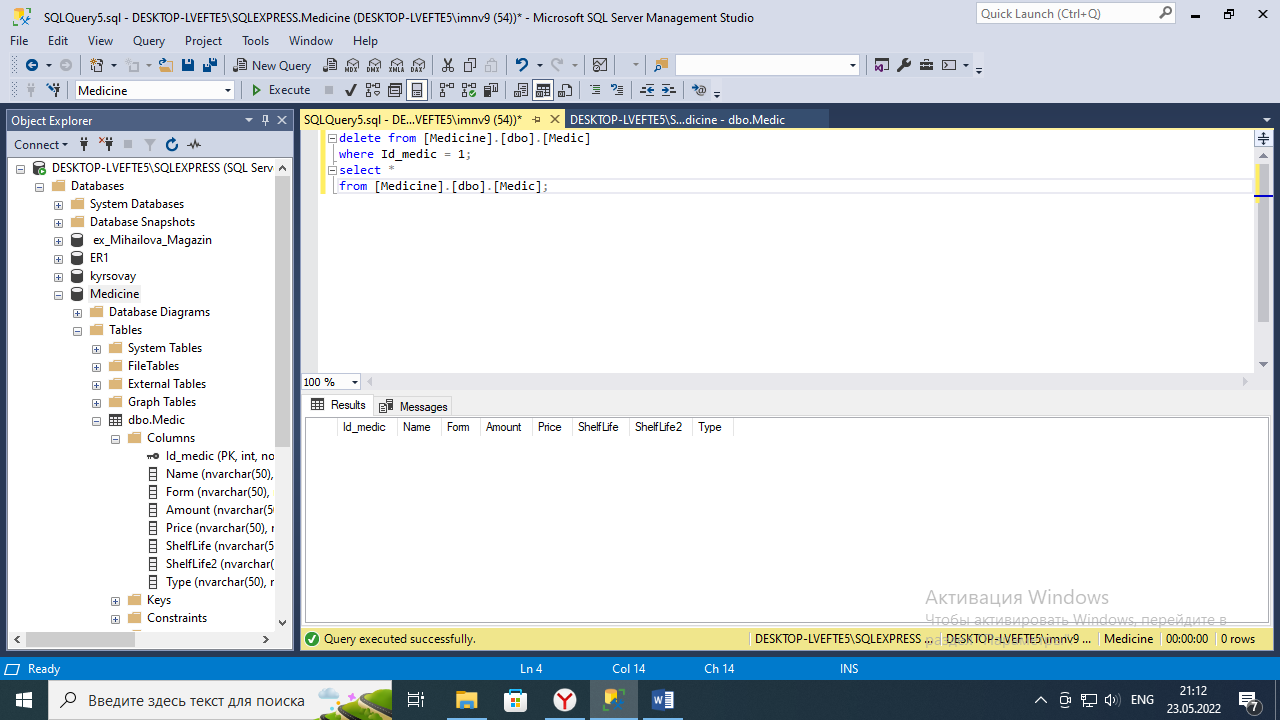
Для проверки работоспособности базы данных были проведены запросы по добавлению нового лекарства (рисунок 3), изменению лекарства (рисунок 4), удаление лекарства (рисунок 5).



**Рисунок 3. Добавление лекарства**



**Рисунок 4. Обновление характеристики лекарства**



**Рисунок 5. Удаление лекарства**

1. **РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ**

Разработка программного обеспечения — деятельность по созданию нового программного обеспечения. Разработка программного обеспечения как инженерная дисциплина является составной частью программной инженерии, наряду с дисциплинами, отвечающими за функционирование и сопровождение программных продуктов. [11]

Процесс разработки состоит из множества подпроцессов, или дисциплин, некоторые из которых перечислены ниже. Процесс — совокупность взаимосвязанных или взаимодействующих видов деятельности, преобразующих входы в выходы.

* Анализ требований → Спецификация программного обеспечения
* Проектирование программного обеспечения
* Программирование
* Тестирование программного обеспечения
* Системная интеграция (System integration)
* Внедрение программного обеспечения (или Установка программного обеспечения)
* Сопровождение программного обеспечения [10]

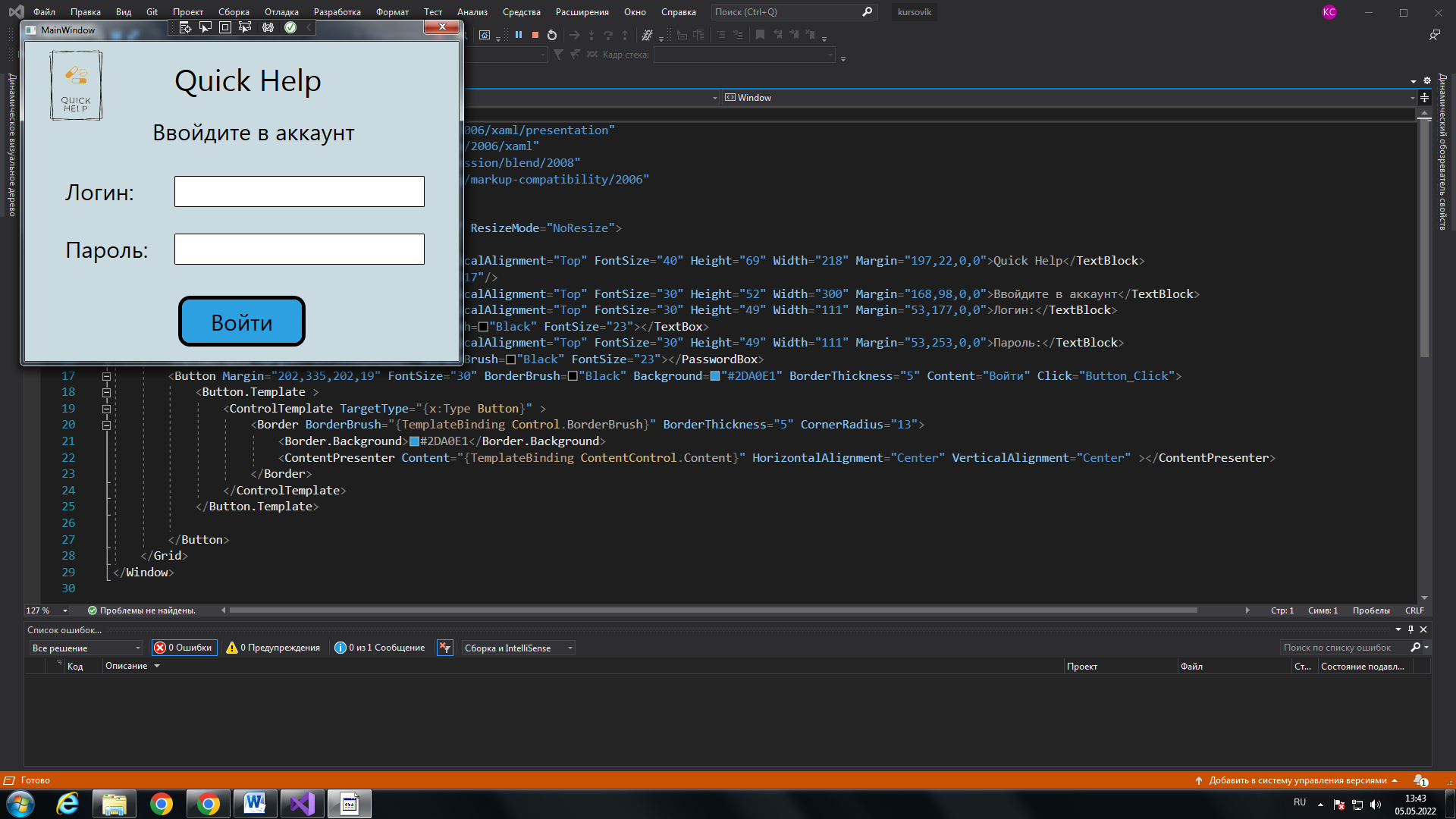
Для разработки была использована программа Visual Studio. Код написан на языке С#.

## Разработка кода информационной системы

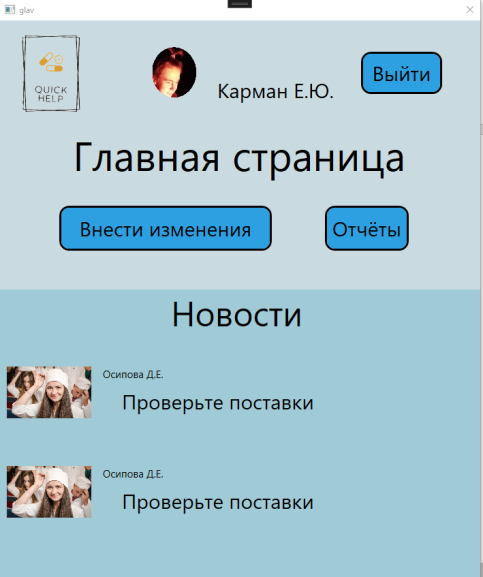
Программа разработана на языке С#.

C# — современный объектно-ориентированный, который позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО [1].

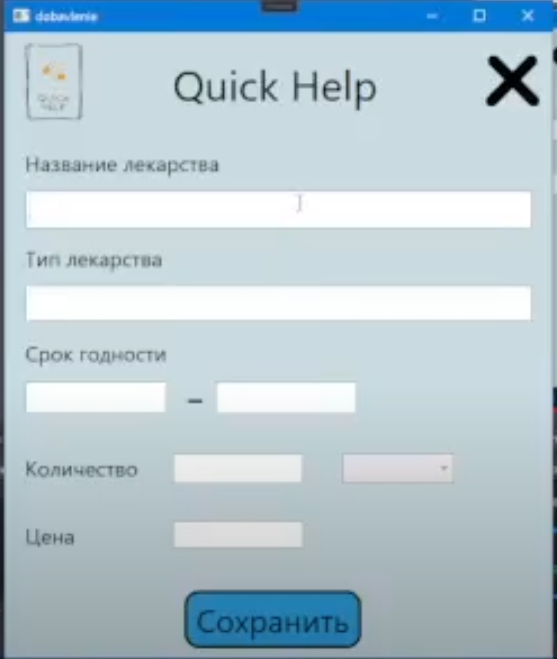
В данной главе представлен процесс реализации ИС. Продемонстрированы скриншоты окон приложения на рисунках 6, 7, 8, 9 и часть кода.



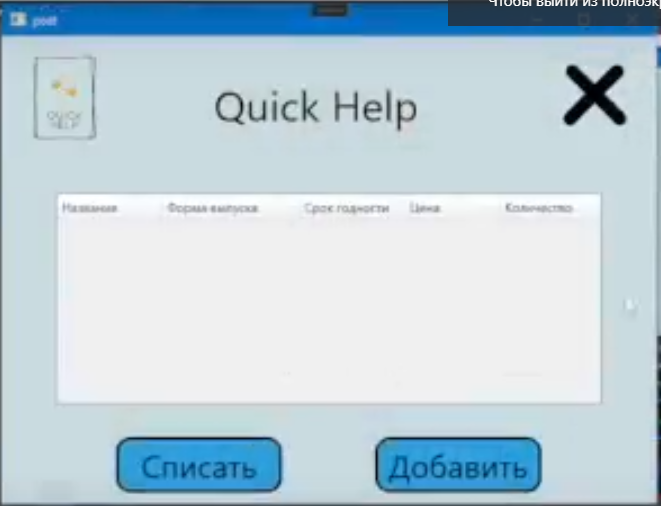
**Рисунок 6. Скриншот окна входа в приложение**



**Рисунок 7. Скриншот личной страницы**



**Рисунок 8. Скриншот окна добавления лекарства**



**Рисунок 9. Скриншот окна с лекарствами**

<Grid>

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Button Width="150" Height="70"FontSize="30"HorizontalAlignment="Right"VerticalAlignment="Top" Margin="0,50,50,0"BorderBrush="Black" Background="#2DA0E1"BorderThickness="5" Content="Выйти" Click="Button\_Click">

<Button.Template>

<ControlTemplateTargetType="{x:Type Button}" >

<BorderBorderBrush="{TemplateBindingControl.BorderBrush}"BorderThickness="3"CornerRadius="13">

<Border.Background>#2DA0E1</Border.Background>

<ContentPresenter Content="{TemplateBindingContentControl.Content}"HorizontalAlignment="Center"VerticalAlignment="Center" ></ContentPresenter>

</Border>

</ControlTemplate>

</Button.Template>

</Button>

<Image Source="logo.png" Margin="30,30,0,0" Height="100" Width="100"HorizontalAlignment="Left"VerticalAlignment="Top" />

<TextBlock x:Name="FIO" Text=""FontSize="30"VerticalAlignment="Top"HorizontalAlignment="Center" Margin="0,80,0,0"></TextBlock>

<Button Margin="83,0,0,116"VerticalAlignment="Bottom"HorizontalAlignment="Left" Width="300" Height="80"FontSize="30"BorderBrush="Black" Background="#2DA0E1"BorderThickness="5" Content="Внестиизменения" Click="Button\_Click\_1">

<Button.Template>

<ControlTemplateTargetType="{x:Type Button}" >

<BorderBorderBrush="{TemplateBindingBorderBrush}"BorderThickness="3"CornerRadius="13">

<Border.Background>

<Brush>#2DA0E1</Brush>

</Border.Background>

<ContentPresenter Content="{TemplateBinding Content}"HorizontalAlignment="Center"VerticalAlignment="Center" />

</Border>

</ControlTemplate>

</Button.Template>

</Button>

<TextBlock Text="ГЛАВНАЯСТРАНИЦА"FontSize="40"FontWeight="Black"HorizontalAlignment="Left"VerticalAlignment="Top" Margin="134,201,0,0"></TextBlock>

<Button Margin="483,0,0,116"VerticalAlignment="Bottom"HorizontalAlignment="Left" Width="150" Height="80"FontSize="30"BorderBrush="Black" Background="#2DA0E1"BorderThickness="5" Content="Склад" Click="Button\_Click\_2">

<Button.Template>

<ControlTemplateTargetType="{x:Type Button}" >

<BorderBorderBrush="{TemplateBindingBorderBrush}"BorderThickness="3"CornerRadius="13">

<Border.Background>

<Brush>#2DA0E1</Brush>

</Border.Background>

<ContentPresenter Content="{TemplateBindingContent}"HorizontalAlignment="Center"VerticalAlignment="Center" />

</Border>

</ControlTemplate>

</Button.Template>

</Button>

</Grid>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

if (Login.Text.Length < 4 || Password.Text.Length < 4)

{

MessageBox.Show("Вы неверено ввели данные");

}

else

{

glav win1 = new glav(Login.Text, Password.Text);

win1.Show();

Close();

}

}

}

В итоге была полностью разработана программа, удовлетворяющая требованиям заказчика.

1. **ТЕСТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

Тестирование – это проверка созданного программного продукта на соответствие заданным требованиям, и на отсутствие дефектов. Тестирование подразумевает запуск программы (или ее компонентов) при помощи ручных (или автоматизированных) программных средств тестирования; тестироваться могут и отдельные компоненты (или отдельные свойства) программы.

Тестирование программного обеспечения — процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом. Тестирование является неотъемлемой частью жизненного цикла программного обеспечения. Само по себе тестирование – длительный процесс проверок на соответствие ожидаемого результата. Нельзя выделить какой-то один этап как важный, каждый из них имеет одинаковый вес. При создании продукта тестировщик не просто играет важную роль, а участвует на каждом этапе разработки от концепции до выхода продукта в свет.

Как уже было отмечено, тестирование – неотъемлемая часть жизненного цикла продукта, однако необходимо понимать, с чего стоит начинать тестирование, и когда оно заканчивается. Например, неопытные специалисты ошибочно думают, что после релиза они могут переключаться на другой проект.

Всего принято выделять 7 этапов тестирования:

* Работа с требованиями. Знакомство с требованиями заказчика, что должен из себя представлять итоговый продукт, обсуждение.
* Разработка стратегии тестирования. Оценка сроков тестирования, выявление среды тестирования, объединение всей информации, полученной при работе с требованиями.
* Создание тестовой документации. Написание сценариев, которые позволят проверить функционал.
* Тестирование прототипа. Тестирование основного функционала продукта, корректировка целей, добавление фичей.
* Основное тестирование. Выполнение общей проверки продукта.
* Стабилизация. На данном этапе происходит работа над устранением багов.
* Эксплуатация. Проводится регресс-тестирование, устранение ошибок, которые нашел конечный пользователь. [15]

## План тестирования

Для отдельных полей предполагается провести:

* Позитивное тестирование приложения (корректные шаги, корректные данные).
* Негативное тестирование (подразумевает введение некорректных данных).

Для всей системы предполагается провести:

* Функциональное тестирование;
* Кросс-браузерное тестирование;
* Юзабилити тестирование;
* Тестирование пользовательского интерфейса.

Тестируемый функционал сайта:

* Будет протестирована следующая функциональность сайта:
* каталог товаров – приоритет высокий;
* пользовательская корзина- приоритет высокий;
* регистрационная форма – приоритет высокий;
* логин;
* форма отправки заказа – приоритет высокий;
* профайл пользователя – приоритет низкий.

Критерии начала и окончания тестирования:

Тестирование может быть начато, если выполнены следующие условия:

1. Готова и утверждена необходимая документация;
2. Тестируемый функционал окончен и готов для передачи в тестирование.

Тестирование окончено, если выполнены следующие условия:

1. Все найденные дефекты задокументированы.

Риски и ограничения:

* ввиду ограниченности ресурсов, гарантировать корректное отображение сайта можем только на устройствах, которые указаны в таблице «Конфигурации». [16]

## Написание тест-кейсов

Тест-кейс — это профессиональная документация тестировщика, последовательность действий, направленная на проверку какого-либо функционала, описывающая как прийти к фактическому результату. [7]

Правила написания тест-кейсов

1. Заголовок:

* должен быть чётким, кратким, понятным и однозначно характеризующим суть тест-кейса;
* не может содержать выполняемые шаги и ожидаемый результат.

1. Предусловие:

* может содержать полную информацию о состоянии системы или объекта, необходимом для начала выполнения шагов тест-кейса;
* может содержать ссылки на информационные источники, которые необходимо изучить перед прохождением тест-кейса (инструкции, описание систем…);
* не может содержать ссылки на тестируемый ресурс, если у информационной системы более одной среды, данная информация должна быть вынесена в инструкцию, и ссылка приложена в предусловии;
* не может содержать данные для авторизации, данная информация должна быть вынесена в инструкцию, и ссылка приложена в предусловии;
* не может содержать выполняемые шаги и ожидаемый результат, если нам нужно, чтобы до выполнения шагов проверки у нас была открыта главная страница, то мы в предусловии указываем «открыта главная страница сайта»;
* не может содержать ожидаемый результат.

1. Шаги проверки:

* должны быть чёткими, понятными и последовательными;
* следует избегать излишней детализации шагов. Правильно: «ввести в поле число 12».  
  Неправильно: «нажать на клавиатуре на цифру ‘1’, следующим шагом нажать на клавиатуре на цифру ‘2’»;
* должны использоваться безличные глаголы.  
  Правильно: нажать, ввести, перейти.  
  Неправильно: нажмите, введите, идите;
* не должно быть комментариев и пояснений, если есть необходимость привести мини-инструкцию, то оформляем инструкции в базе-знаний и в предусловии ссылаемся на неё;
* не должно быть жёстко прописанных статических данных (логины, пароли, имена файлов) и примеров, для исключения эффекта пестицида.

1. Ожидаемый результат:

* должен быть у каждого шага проверки;
* должно быть кратко и понятно описано состояние системы или объекта, наступающее после выполнения соответствующего шага;
* не должно быть избыточного описания.

1. Общие требования к тест-кейсам:

* язык описания тест-кейсов должен быть понятен широкому кругу пользователей, а не узкой группе лиц;
* тест-кейс должен быть максимально независим от других тест-кейсов и не ссылаться на другие тест-кейсы (лучшая практика, когда зависимостей нет вообще);
* тест-кейсы группируются в функциональные блоки по их назначению;
* в тест-кейсах, проверяющих работу функционала скриншотов быть не должно, иначе вы будете посвящать сотни часов на изменение всех скриншотов в тысячах тест-кейсах при изменении интерфейса тестируемой программы. Скриншоты могут быть добавлены только в тест-кейсы, проверяющие отображение страниц и форм. [14]

Тест кейс №1: вход в систему

|  |  |  |
| --- | --- | --- |
| **Шаги** | **Ожидаемый результат** | **Результат** |
| Открыть приложение | Приложение должно запуститься |  |
| Ввести логин | Вводится имя и фамилия |  |
| Нажать кнопку «Войти» | Переход на другую страницу |  |

Тест кейс №2: редактирование лекарств

|  |  |  |
| --- | --- | --- |
| **Шаги** | **Ожидаемый результат** | **Результат** |
| Открыть приложение | Приложение должно запуститься |  |
| Зайти в систему | Вводится имя и фамилия |  |
| Нажать на копку «Редактировать» | Открывается окно для редактирования |  |

Тест кейс №3: списание лекарств

|  |  |  |
| --- | --- | --- |
| **Шаги** | **Ожидаемый результат** | **Результат** |
| Зайти в приложение | Ввести имя и фамилия |  |
| Нажать кнопку «внесение изменение» | Открывается новое окно |  |
| Выбрать лекарство и нажать кнопку «списать» | Открывается окно с подтверждением удаления |  |

Тест кейс №4: добавление лекарств

|  |  |  |
| --- | --- | --- |
| **Шаги** | **Ожидаемый результат** | **Результат** |
| Зайти в приложение | Ввести имя и фамилия |  |
| Нажать кнопку «внесение изменение» | Открывается новое окно |  |
| Нажать кнопку «добавить» | Открывается окно для добавления лекарств |  |

Тест кейс №5: выход из системы

|  |  |  |
| --- | --- | --- |
| **Шаги** | **Ожидаемый результат** | **Результат** |
| Зайти в приложение | Ввести имя и фамилия |  |
| На главной странице нажать кнопу «выйти» | Открывается главное окно для входа в систему |  |

## Юнит-тестирование

Модульное тестирование, или юнит-тестирование (англ. unittesting) — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы.  
Идея состоит в том, чтобы писать тесты для каждой нетривиальной функции или метода. Это позволяет достаточно быстро проверить, не привело ли очередное изменение кода к регрессии, то есть к появлению ошибок в уже оттестированных местах программы, а также облегчает обнаружение и устранение таких ошибок.

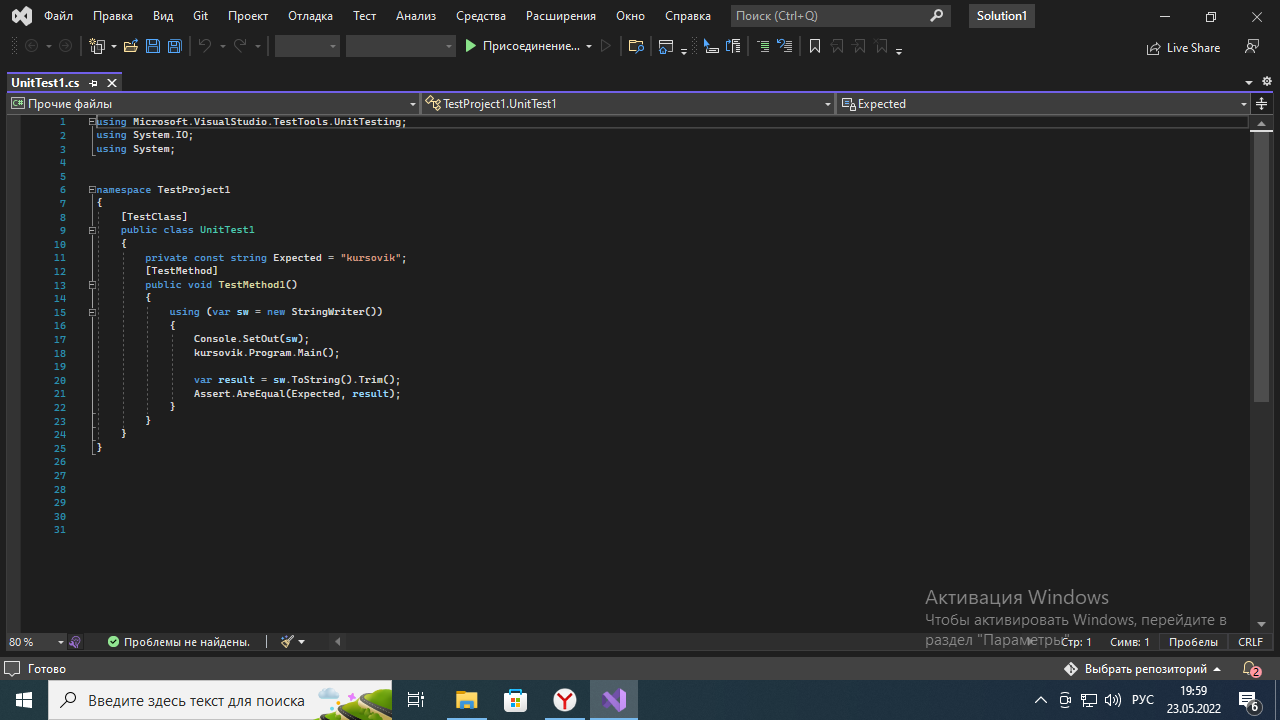
Модульное тестирование делится на ручное и автоматизированное. И хотя программная инженерия не выделяет превосходство одного над другим, чаще, все-таки, используется автоматизированное.  
При ручном модульном тестировании, как правило используется пошаговая инструкция.[22]  
Алгоритм же автоматизированного заключается в следующем:

* Разработчик записывает в приложение единицу кода, чтобы протестировать ее. После: они комментируют и, наконец, удаляют тестовый код при развертывании приложения.
* Разработчик может изолировать единицу кода для более качественного тестирования. Эта практика подразумевает копирование кода в собственную среду тестирования. Изоляция кода помогает выявить ненужные зависимости между тестируемым кодом и другими модулями или пространствами данных в продукте.
* Кодер обычно использует UnitTestFramework для разработки автоматизированных тестовых случаев. Используя инфраструктуру автоматизации, разработчик задает критерии теста для проверки корректного выполнения кода, и в процессе выполнения тестовых случаев регистрирует неудачные. Многие фреймворки автоматически отмечают и сообщают, о неудачных тестах и могут остановить последующее тестирование, опираясь на серьезность сбоя.
* Алгоритм модульного тестирования:
  + Создание тестовых случаев
  + Просмотр / переработка
  + Базовая линия
  + Выполнение тестовых случаев.

Чтобы модульное тестирование было максимально эффективным, тесты должны:

* соответствовать конкретному модулю — нельзя применять один и тот же тест для тестирования разных по назначению и реализации программных компонентов;
* быть автоматизированными — тест лучше вписать в сам код, тогда он будет запускаться автоматически и сильно упростит жизнь разработчику;
* быть своевременными — если тест нельзя написать до разработки самого кода, его лучше создавать параллельно, что сэкономит много времени в дальнейшем;
* отвечать основным задачам — при написании теста не нужно стараться учесть все возможные сценарии, лучше сосредоточиться сначала на основных, а остальные дополнять по мере необходимости;
* иметь хорошее название — описывающее, что именно тестируется, в каких условиях и с каким желаемым результатом. [23]

Для проверки работоспособности программы был написан юнит-тест, представленный на рисунке 7.



**Рисунок 10. Юнит-тест**

В ходе работы над юнит-тестом возникла ошибка в том, что база данных является локальной.

**ЗАКЛЮЧЕНИЕ**

Проводя работы по созданию информационной системы «Quick help» с целью улучшения деятельности аптеки, было сделано настольное приложение, в котором реализованы функции для учёта лекарственных средств. В процессе работы мы закрепили знания по программированию и созданию настольного приложения с графическим интерфейсом. Самостоятельно решали возникающие проблемы и научились создавать юнит-тесты. Навыки работы с базой данных значительно улучшились, а также с приложениями Visual Studio, Microsoft SQL Server Management Studio.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Краткий обзор языка С# <https://docs.microsoft.com/ru-ru/dotnet/csharp/tour-of-csharp/>
2. Кто такой фармацевт? [https://synergy.ru/about/education\_articles/](https://synergy.ru/about/education_articles/speczialnosti/kto_takoj_farmaczevt)
3. Медицина [https://bigenc.ru/medicine/text/5388472#:~:text=МЕДИЦИ́НА%20(лат.%20medicina%2C%20от%20medicus,и%20трудоспособности%2C%20продление%20жизни%20людей](https://bigenc.ru/medicine/text/5388472#:~:text=%D0%9C%D0%95%D0%94%D0%98%D0%A6%D0%98%CC%81%D0%9D%D0%90%20(%D0%BB%D0%B0%D1%82.%20medicina%2C%20%D0%BE%D1%82%20medicus,%D0%B8%20%D1%82%D1%80%D1%83%D0%B4%D0%BE%D1%81%D0%BF%D0%BE%D1%81%D0%BE%D0%B1%D0%BD%D0%BE%D1%81%D1%82%D0%B8%2C%20%D0%BF%D1%80%D0%BE%)
4. Модульное тестирование: что это? Типы, инструменты. <https://logrocon.ru/news/unit_testing>
5. Определение аптеки [https://kartaslov.ru/карта-знаний/Аптека](https://kartaslov.ru/%D0%BA%D0%B0%D1%80%D1%82%D0%B0-%D0%B7%D0%BD%D0%B0%D0%BD%D0%B8%D0%B9/%D0%90%D0%BF%D1%82%D0%B5%D0%BA%D0%B0)
6. Определение информационной системы. Задачи и функции ИС <https://studopedia.ru/4_153879_opredelenie-informatsionnoy-sistemi-zadachi-i-funktsii-is>
7. Пишем максимально эффективный тест-кейс <https://habr.com/ru/post/246463/>
8. Понятие информационной системы [https://tsput.ru/res/informat/sist\_seti\_fmo/lekcii/lekciy1.html#:~:text=Информационная](https://tsput.ru/res/informat/sist_seti_fmo/lekcii/lekciy1.html#:~:text=%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F)
9. Правильно пишем тест-кейсы. Памятка начинающему специалисту по тестированию <https://victorz.ru/202001101079>
10. Процесс разработки программного обеспечения [https://wiki2.org/ru/Процесс\_разработки\_программного\_обеспечения](https://wiki2.org/ru/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D1%81%D1%81_%D1%80%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B8_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F)
11. Разработка программного обеспечения [https://ru.wikipedia.org/wiki/Разработка\_программного\_обеспечения#:~:text=Разрабо́тка](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D1%8F#:~:text=%D0%A0%D0%B0%D0%B7%D1%80%D0%B0%D0%B1%D0%BE%CC%81%D1%82%D0%BA%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%CC%81%D0%BC%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%CC%81%D1%87%D0%B5%D0%BD%D0%B8%D1%8F%20(%D0%B0%D0%BD%D0%B3%D0)
12. РД 50-34.698-90 Руководство пользователя (пример оформления)<https://www.prj-exp.ru/patterns/pattern_user_guide.php>
13. Создание физической модели [https://studwood.net/](https://studwood.net/1764401/informatika/razrabotka_fizicheskoy_modeli_dannyh_pryamoe_proektirovanie)
14. Тестовые сценарии (Testcase), тестовые варианты. Оформление результатов тестирования <https://github.com/kolei/PiRIS/blob/master/articles/5_3_1_4_testcase.md>
15. Тестирование. Что это такое, описание, виды тестирования <https://testengineer.ru/testirovanie-chto-ehto-takoe/>
16. Тест план для интернет-магазина [https://lms.kgeu.ru/pluginfile.php?file=%2F198821%2Fmod\_resource%2Fcontent%2F1%2FПлан%20тестирования.pdf](https://lms.kgeu.ru/pluginfile.php?file=%2F198821%2Fmod_resource%2Fcontent%2F1%2F%D0%9F%D0%BB%D0%B0%D0%BD%20%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F.pdf)
17. Техническое задание <https://docs.cntd.ru/document/1200144624>
18. Требования, предъявляемые к КИС <https://www.sites.google.com/site/korpinfsis/home/trebovania-predavlaemye-k-kis>
19. Физическая модель [https://neerc.ifmo.ru/wiki/](https://neerc.ifmo.ru/wiki/index.php?title=%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85#:~:text=%D0%9E%D0%BF%D1%80%D0%B5%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%3A%20%D0%A4%D0%B8%D0%B7%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F%20%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C%20%D0%B1%D0%B0%D0%B7%D1%8B%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85,%D0%A1%D0)
20. Что такое база данных? <https://www.oracle.com/cis/database/what-is-database/>
21. Что такое лекарственное средство? <https://tabletki.ua/content/drugs/>
22. Юнит-тестирование для чайников <https://habr.com/ru/post/169381/>
23. Unit-тестирование <https://blog.skillfactory.ru/glossary/unit-testirovanie/>

**СПИСОК ИЛЛЮСТРАЦИЙ**

[Рисунок 1. Прототип приложения 16](#_heading=h.23ckvvd)

[Рисунок 2. ER-диаграмма 17](#_heading=h.ihv636)

[Рисунок 3. Добавление лекарства 18](#_heading=h.1hmsyys)

[Рисунок 4. Обновление характеристики лекарства 18](#_heading=h.41mghml)

[Рисунок 5. Удаление лекарства 18](#_heading=h.2grqrue)

[Рисунок 6. Скриншот окна входа в приложение 21](#_heading=h.1v1yuxt)

[Рисунок 7. Скриншот личной страницы 21](#_heading=h.4f1mdlm)

[Рисунок 8. Скриншот окна добавления лекарства 21](#_heading=h.2u6wntf)

[Рисунок 9. Скриншот окна с лекарствами 22](#_heading=h.19c6y18)

[Рисунок 10. Юнит-тест 35](#_heading=h.111kx3o)

[Рисунок 11. Вход в систему 49](#_heading=h.43ky6rz)

**ПРИЛОЖЕНИЯ**

## Приложение №1

Инструкции для использования системы (руководство пользователя)

* + - 1. Введение

Программа применяется в медицине, а именно в работе аптеки.

В программе реализованы такие функции как просмотр лекарств на складе, редактирование лекарств, а также их списание и добавление.

Для использования данной программы пользователь должны быть сотрудниками аптеки и умение обращаться с компьютером и программой, а также составление отчётов при имеющихся данных.

Пользователю нужно изучить такие документы как руководство пользователя, общие документы использования программы.

* + - 1. Назначение и условия применения «Quick help»

Программа предназначена для учёта лекарственных средств. Использовать её необходимо, если нужно списать, добавить или редактировать медикаменты.

* + - 1. Подготовка к работе

Для работы с «Quick help» необходимо следующее программное обеспечение:

Локальная сеть для обеспечения работы компьютера, программы

Перед началом работы на рабочем месте пользователя необходимо выполнить следующие действия:

Необходимо открыть программу и войти в неё.

* + - 1. Рекомендации по освоению

Программа предназначена для быстрого освоения, интерфейс достаточно понятен. Чтобы разобраться с работой программы, можно у сотрудников спросить, которые работают с ней или позвонить по номеру, указанный в качестве горячей линии.

## Приложение №2

Техническое задание

**1. Общие сведения о проекте**

1.1. Полное наименование системы и ее условное обозначение

Информационная система «Quick help».

1.2. Наименование предприятий разработчика и заказчика (пользователя) системы и их реквизиты:

Заказчик: ОГБОУ «Костромской энергетический техникум имени Ф.В. Чижова»

Разработчик: Михайлова Анастасия Валерьевна

1.3. Перечень документов, на основании которых создается система, кем и когда утверждены эти документы:

ISO/IES 12207:1995-08-01 «Информационная технология. Процессы ЖЦ программного обеспечения»

ГОСТ 34.601-90 «Стадии создания АС»

ГОСТ 34.602-89 «Техническое задание на создание АС»

ГОСТ 34.603-92 «Виды испытаний АС»

РД 50-34.698-90 «Требование к содержанию документов»

1.4. Плановые сроки начала и окончания работы по созданию системы:

В соответствии с календарным графиком на курсовую работу

1.5. Порядок оформления и предъявления заказчику результатов работ по созданию системы (ее частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических (программно-методических) комплексов системы:

* при выполнении каждого этапа раз в неделю разработчик предъявляет заказчику результаты своей работы в виде графиков, диаграмм, отчетов, таблиц, рисунков.
* после окончания каждого этапа разработчик предоставляет заказчику документацию, соответствующую этому этапу.

**2. Назначения и цели создания системы**

2.1. Назначение системы

Информационная система «Quick help» разработана для автоматизации ведения учета лекарственных средств в аптеке.

2.2. Цели создания системы:

* учет лекарств;
* добавление лекарств;
* списание лекарств;

**3. Характеристики объекта автоматизации**

3.1. Краткие сведения об объекте автоматизации

Объектом автоматизации является деятельность аптеки.

3.2. Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.

При проектировании ИС аптеки должны быть автоматизировано рабочее место сотрудника аптеки.

**4. Требования к системе**

4.1 Требования к системе в целом

4.1.1. Требования к структуре и функционированию системы:

ИС представляет собой единую систему без разграничения прав.

4.1.2. Требования к численности, квалификации персонала

Данная область предполагает работу за компьютером сотрудника аптеки. Поэтому необходимо для неё провести обучение для пользования ИС и основам работы на компьютере.

4.1.3. Требования к надежности:

Должна быть обеспечена надежность хранения ИС, которая заключается в возможности восстановления ИС после любого рода аппаратных и программных сбоев. Для этого необходимо обеспечить его устойчивую и бесперебойную работу, что достигается наличием источника бесперебойного питания. Периодически надо производить резервное копирование ИС.

4.1.4. Требования по эргономике и технической эстетике

Пользовательские интерфейсы должны быть просты и понятны для неподготовленных пользователей и выполнены в спокойных тонах. При этом должен быть учтен следующий стандарт:

ГОСТ Р 50948-96 «Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности»

Комфортность условий работы персонала должна обеспечиваться в соответствии со стандартами:

ГОСТ 12.2.032-78 «Рабочее место при выполнении работ сидя. Общие эргономические требования»;

ГОСТ Р 50923-96 «Дисплеи. Рабочее место оператора. Общие эргономические требования к производственной среде. Методы измерения».

Оценка эргономических параметров и параметров безопасности должна проводиться в соответствии со стандартом:

- ГОСТ Р 50949-96 «Средства отображения информации индивидуального пользования. Методы измерений и оценки эргономических параметров и параметров безопасности».

4.1.5. Требования по стандартизации и унификации:

Необходимо применять стандартное общее и базовое программное обеспечение, включающее лицензионные ОС, СУБД, использовать утвержденные формы входных и выходных документов, применять унифицированные интерфейсы.

4.2 Требования к функциям (задачам), выполняемым системой (функциональные блоки нижнего уровня иерархии дерева узлов):

* Удаление лекарств;
* Добавление лекарств;
* Редактирование лекарств;

4.3 Требования к видам обеспечения

4.3.1. Требования к информационному и программному обеспечению

Данные в системе должны быть реализованы в виде реляционной БД на основе MySQL. БД располагается на локальном компьютере под управлением операционной системы Windows 7.

Требования к техническому обеспечению

Локальный компьютер должен быть оснащен процессором не ниже Intel Xeon 2500 МГц, оперативной памятью 512 MB и RAID массивом для обеспечения целостности базы данных. Клиентские места должны быть оснащены процессором не ниже Pentium-800 МГц. Для распечатки документов и отчетов необходимо установить принтеры.

**5. Состав и содержание работ по созданию (развитию) системы**

Работы по созданию системы выполняются последовательно и включают следующие этапы:

5.1. Формирование требований к ИС

Характеристика объекта и результатов его функционирования:

Разработчик – изучает работу сотрудника аптеки, анализирует полученную информацию, делает вывод о необходимости ИС, выявляет лица, для которых будет предназначена данная система, выделяет проблемы, которые можно решить с помощью автоматизации. Заказчик – предоставляет разработчику необходимую информацию в виде документов, отчетов, выписок, журналов и прочее. Результатом этапа является описание предметной области.

5.2. Разработка концепций ИС

Изучение объекта:

Разработчик – проводит детальное изучение работы сотрудника аптеки. Результатом на данном этапе является функциональной модели IDIF0. Заказчик – предоставляет разработчику все необходимую для этого информацию.

Разработка технического задания (ТЗ)

Проводится разработка, оформление, согласование и утверждение ТЗ в соответствии с ГОСТ 34.602-83).

На этом этапе разработчик может сам создать первоначальный профиль стандартов на основании стандартов: ISO/IES 1220761995-08-01; ГОСТ 34.602-83; ГОСТ 34.601-90; ГОСТ 34.20-89; РД 50-34.689-90; ГОСТ 34.603-92

5.4. Технический проект

Разработка проектных решений по системе: разработчик - создает общие решения по системе и ее частям, по функционально-алгоритмической структуре системы, по функциям персонала и организационной структуре. Определяет структуру технических средств, алгоритмы решения и применяемые языки. Разрабатывает решения по организации и ведению информационной базы, по системе классификации и кодированию информации, по программному обеспечению.

Разработка документации на ИС: разработчик - по итогам работ составляет (РД 50-34.698-90):

* пояснительная записка к техническому проекту;
* спецификации требований и алгоритмы на функциональные группы программ, программные и информационные компоненты;
* описание организации информационной базы.

5.5. Рабочая документация

Разработка рабочей документации на ИС: разработчик – создает руководство пользователя на основании стандартов РД 50-34.698-90 и ЕСПД, которое должно быть представлено в пояснительной записке.

5.6. Ввод в действие

Организационная подготовка объекта информатизации к вводу ИС: разработчик и заказчик - проводят организационную подготовку объекта к вводу ИС.

Подготовка персонала: разработчик - обучает персонал, проверяет их способность обеспечить функционирование ИС.

Строительно-монтажные работы: производится установка и наладка технических средств.

Пуско-наладочные работы: проводятся автономная наладка аппаратных и программных средств, загрузка информации в ИС, комплексная наладка всех средств системы.

Испытания системы: проводятся предварительные испытания, опытная эксплуатация и приемочные испытания. Более подробно испытания системы представлены в пункте "Порядок контроля и приемки системы" настоящего документа.

5.7. Сопровождение ИС

Разработчик - анализирует функционирование системы, выявляет отклонения эксплуатационных характеристик от проектных значений, устраняет причины этих отклонений.

**6. Порядок контроля и приемки системы**

При проведении испытаний выявляются ошибки, ИС проверяется на соответствие функциональным и эксплуатационным требованиям, требованиям нормативных документов и ТЗ, проверяется надежность системы. При необходимости система дорабатывается и готовится к следующему этапу испытаний.

Информационная система проходит обычно три этапа испытаний:

* предварительные испытания: по усмотрению разработчика создается программа и методика автономных или комплексных испытаний (ГОСТ 34.603-92, РД 50-34.689-90, ЕСПД).
* опытная эксплуатация: разработчик – создает программу и методики испытаний (ГОСТ 34.603-92, РД 50-34.689-90, ЕСПД). В соответствии с этой программой проводят опытную эксплуатацию.
* приемочные испытания: разработчик - создает программу и методики испытаний (в соответствии со стандартом ГОСТ 34.603-92, РД 50-34.689-90, ЕСПД). В соответствии с этой программой проводят приемочные испытания.

**7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу в действие**

7.1. Преобразование входной информации виду, пригодному для обработки с помощью ПК.

Все виды документации, используемые внутри аптеки и не выходит за пределы.

7.2. Изменения, которые необходимо осуществить в объекте автоматизации

Проведение подобных работ оговаривается отдельно.

7.3. Создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ

Заказчик обязуется к моменту запуска ИС подготовить необходимые помещения для серверной, оплатить разработку проекта локальной сети и ее реализацию.

7.4. Сроки и порядок комплектования штатов и обучения персонала

Перед началом работы с ИС сотрудники аптеки должны пройти начальный курс работы с ПК и курс обучения работы с данной ИС.

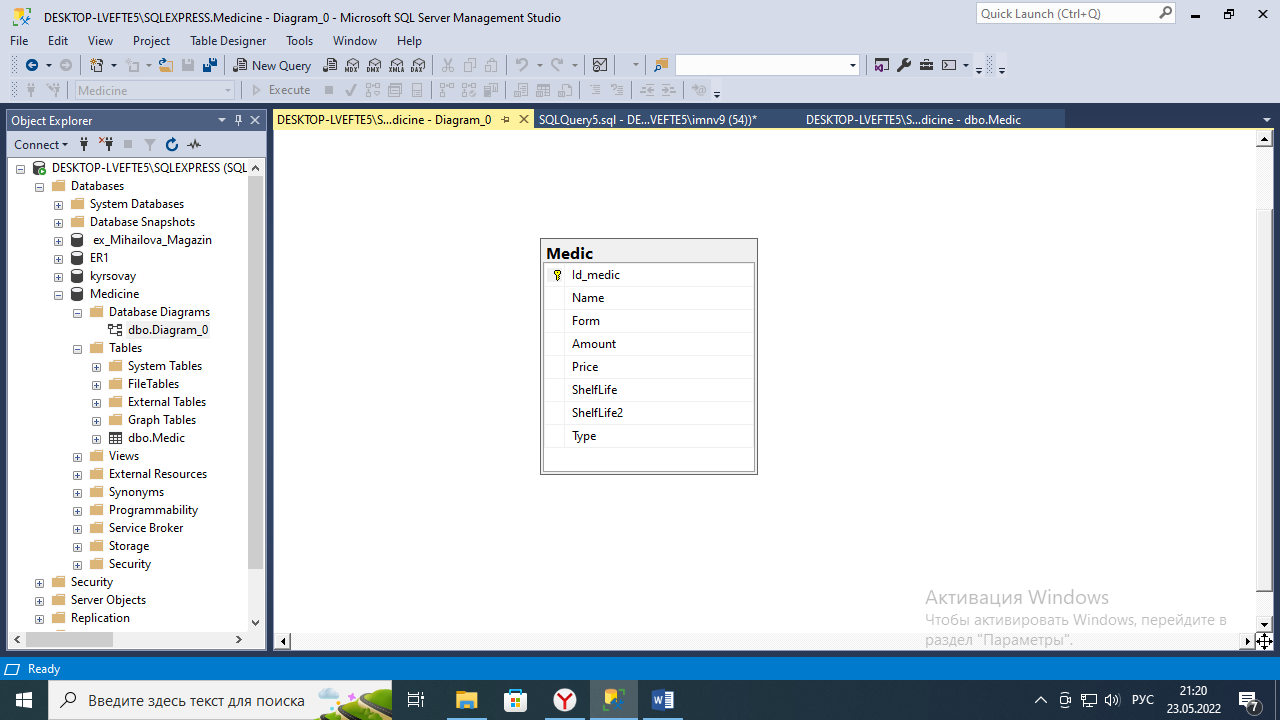
**8. Требования к документированию**

Перечень подлежащих разработке документов:

* технический проект;
* руководство пользователя;
* пояснительная записка;
* приложение.

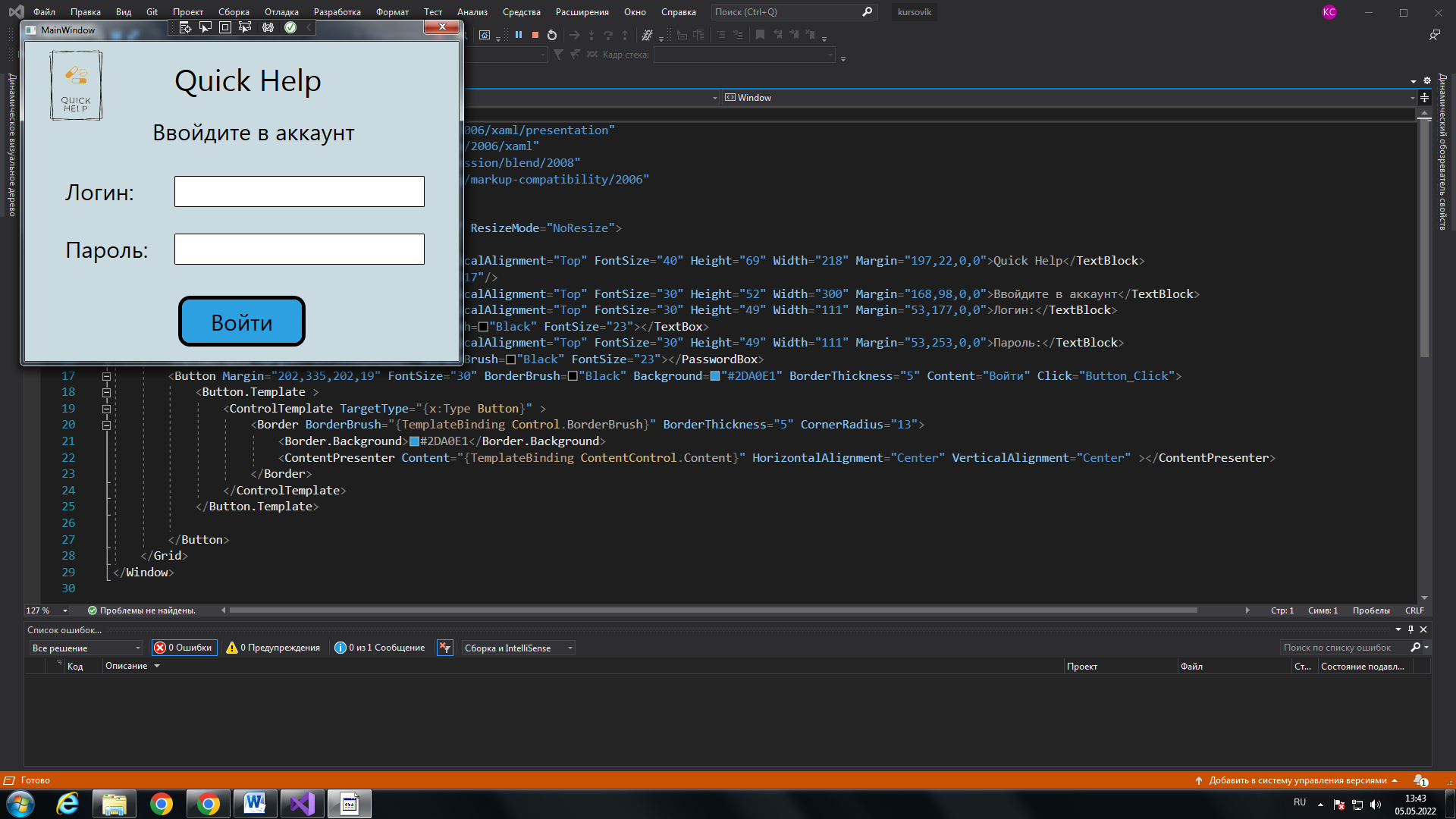
## Приложение №3

ER-ДИАГРАММА



## Приложение №4

ЛИСТИНГ КОДА



**Рисунок 11. Вход в систему**

<Grid>

<Grid.RowDefinitions>

<RowDefinition></RowDefinition>

</Grid.RowDefinitions>

<Button Width="150" Height="70"FontSize="30"HorizontalAlignment="Right"VerticalAlignment="Top" Margin="0,50,50,0"BorderBrush="Black" Background="#2DA0E1"BorderThickness="5" Content="Выйти" Click="Button\_Click">

<Button.Template>

<ControlTemplateTargetType="{x:Type Button}" >

<BorderBorderBrush="{TemplateBindingControl.BorderBrush}"BorderThickness="3"CornerRadius="13">

<Border.Background>#2DA0E1</Border.Background>

<ContentPresenter Content="{TemplateBindingContentControl.Content}"HorizontalAlignment="Center"VerticalAlignment="Center" ></ContentPresenter>

</Border>

</ControlTemplate>

</Button.Template>

</Button>

<Image Source="logo.png" Margin="30,30,0,0" Height="100" Width="100"HorizontalAlignment="Left"VerticalAlignment="Top" />

<TextBlock x:Name="FIO" Text=""FontSize="30"VerticalAlignment="Top"HorizontalAlignment="Center" Margin="0,80,0,0"></TextBlock>

<Button Margin="83,0,0,116"VerticalAlignment="Bottom"HorizontalAlignment="Left" Width="300" Height="80"FontSize="30"BorderBrush="Black" Background="#2DA0E1"BorderThickness="5" Content="Внестиизменения" Click="Button\_Click\_1">

<Button.Template>

<ControlTemplateTargetType="{x:Type Button}" >

<BorderBorderBrush="{TemplateBindingBorderBrush}"BorderThickness="3"CornerRadius="13">

<Border.Background>

<Brush>#2DA0E1</Brush>

</Border.Background>

<ContentPresenter Content="{TemplateBindingContent}"HorizontalAlignment="Center"VerticalAlignment="Center" />

</Border>

</ControlTemplate>

</Button.Template>

</Button>

<TextBlock Text="ГЛАВНАЯСТРАНИЦА"FontSize="40"FontWeight="Black"HorizontalAlignment="Left"VerticalAlignment="Top" Margin="134,201,0,0"></TextBlock>

<Button Margin="483,0,0,116"VerticalAlignment="Bottom"HorizontalAlignment="Left" Width="150" Height="80"FontSize="30"BorderBrush="Black" Background="#2DA0E1"BorderThickness="5" Content="Склад" Click="Button\_Click\_2">

<Button.Template>

<ControlTemplateTargetType="{x:Type Button}" >

<BorderBorderBrush="{TemplateBindingBorderBrush}"BorderThickness="3"CornerRadius="13">

<Border.Background>

<Brush>#2DA0E1</Brush>

</Border.Background>

<ContentPresenter Content="{TemplateBindingContent}"HorizontalAlignment="Center"VerticalAlignment="Center" />

</Border>

</ControlTemplate>

</Button.Template>

</Button>

</Grid>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

if (Login.Text.Length < 4 || Password.Text.Length < 4)

{

MessageBox.Show("Вы неверено ввели данные");

}

else

{

glav win1 = new glav(Login.Text, Password.Text);

win1.Show();

Close();

}

}

}

## Приложение №5

Ссылка на облачно хранилище и репозиторий с курсовой работой

<https://drive.google.com/drive/u/0/folders/1yetadANUgx6jdWZu0Kvu45C8Kii0ao8o>

<https://github.com/NAsty34/Mihailova-Anastasiy_3-1-2021>