

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМЕНИ К.Г. РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ  
УНИВЕРСИТЕТ)»  
(ФГБОУ ВО «МГУТУ ИМ. К.Г. РАЗУМОВСКОГО (ПКУ)»)  
УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

**КУРСОВОЙ ПРОЕКТ**

по междисциплинарному курсу: МДК.01.02. Технология разработки  
программного обеспечения  
на тему: Разработка обучающей программы по теме «Криптография данных»  
студента группы 090203-9о-20/1  
специальности 09.02.03 Программирование в компьютерных системах  
Смирнова Константина Вадимовича

|  |       |                |
|--|-------|----------------|
| Студент  | _____ | К.В Смирнов    |
| Руководитель курсового проекта   | _____ | С.Ю. Кузьменко |
| Председатель ПЦК специальности<br>09.02.03 Программирование в<br>компьютерных системах | _____ | А.И. Глускер   |

Дата защиты «\_\_\_\_» \_\_\_\_\_ 2023 г.  
Оценка:

Москва  
2023

## СОДЕРЖАНИЕ

|  |    |
|--|----|
| ВВЕДЕНИЕ.....  | 3  |
| 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....                                   | 5  |
| 1.1 Исследование предметной области .....                    | 5  |
| 1.2 Обзор существующих аналогов приложения .....             | 6  |
| 1.3 Обзор и выбор системы управления баз данных .....        | 12 |
| 1.5 Выбор и характеристика среды разработки приложения ..... | 17 |
| 2 ПРАКТИЧЕСКАЯ ЧАСТЬ .....                                   | 19 |
| 2.1 Разработка приложения .....                              | 19 |
| 2.2 Логическая модель базы данных в ER Assistant.....        | 21 |
| 2.3 Тестирование приложения .....                            | 22 |
| 2.4 Требования к техническим средствам.....                  | 23 |
| 2.5 Требования к программным средствам.....                  | 23 |
| 2.6 Настройка информационной системы.....                    | 23 |
| 2.7 Демонстрация готового продукта .....                     | 24 |
| ЗАКЛЮЧЕНИЕ .....   | 27 |
| СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ .....                         | 28 |
| ПРИЛОЖЕНИЕ А. Код программного продукта .....                | 30 |
| ПРИЛОЖЕНИЕ Б. Описание таблиц базы данных .....              | 32 |

## **ВВЕДЕНИЕ**

Современность сложно представить без использования компьютеров и интернета. Информационные технологии развивались с середины прошлого века вплоть до сегодняшних дней. Распространение персональных компьютеров и смартфонов достигло того, что люди используют устройства не только во время работы, но и в обычной жизни для отдыха, развлечений, общения на расстоянии и обучения.

В следствие взаимодействия людей как с компьютерами, так и между собой по средству соцсетей, данные пользователей, которые используются приложениями, сохраняются в базах данных для дальнейшего использования. Но с ростом объема передаваемой информации возрастает угроза утечки или взлома.

В мире, где информация является одним из самых ценных ресурсов, шифрование данных становится все более актуальной темой. Криптография – это наука, которая занимается защитой информации путем шифрования и дешифрования. К методам криптографии относятся: обеспечение конфиденциальности, целостности данных, аутентификации, шифрования. Криптография используется в сферах начиная от финансовых транзакций до военных коммуникаций.

Основная цель проекта - Разработка обучающей программы по теме «Криптография данных», представляющей из себя ознакомление с основными принципами криптографии данных, методами шифрования и дешифрования информации и приложение на основе переносной шифровальной машины «Энигма» (от нем. Änigma — загадка) для реализации данного материала.

В этапы разработки проекта входят:

- Анализ предметной области;
- Выбор средств разработки;
- Создание и заполнение базы данных;
- Разработка приложения;
- Разработка документации.

В ходе анализа предметной области были изучены основные алгоритмы шифрования, такие как симметричное и асимметричное шифрование. Рассмотрены основные аспекты шифрования. Проведен обзор существующих программ по данной теме для реализации собственного приложения.

# **1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **1.1 Исследование предметной области**

Программы по криптографии данных представляют из себя приложения, в которых реализованы способы шифрования/дешифрования данных. Но чтобы создавать такие продукты, программистам нужны соответствующие теоретические и практические знания.

Обучающие приложения варьируются по уровню сложности и объёму материала, могут включать в себя основы криптографии, методы шифрования и практические навыки. Программы такого уровня полезны компаниям, которые планируют начать обучение сотрудников/клиентов методам шифрования/дешифрования.

В связи со сложностью, программы взаимодействуют с базами данных, которые содержат данные о ключах, методах шифрования/дешифрования алфавита. Благодаря этому повышается эффективность продукта.

Актуальность создания программы для обучения криптографии данных состоит в систематизации знаний, необходимых при разработке приложения с шифрованием/дешифрование данных, для удобства обучения, а также в виде конкретного примера программы по шифрованию/дешифрования информации на основе шифровальной машины «Эни́гма». Это поспособствует увеличению количества специалистов в сфере криптографии.

Стоит отметить, что в основном программисты находят теоретическую информацию об алгоритмах шифрования и реализацию приложений по шифрованию данных отдельно. Поэтому не всегда имеется возможность найти материал, который полностью может способствовать обучению алгоритмам шифрования информации. В следствие этого, разрабатываемый продукт состоит из глав теории, в которых будет рассказано о методах шифрования данных. По результату прохождения теории по средству опроса, пользователь сможет открыть доступ практической части в виде эмулятора

шифровальной машины «Эни́гма». В теоретической части описано, как реализовать данный эмулятор, используя язык программирования.

В итоге получается обучающая программа по криптографии данных в программирование, материал которой может быть использован программистами в своих будущих наработках, что также положительно сказывается на актуальности данной работы

## 1.2 Обзор существующих аналогов приложения

Возможности приложения определялись исходя из рассмотрения уже существующих вариантов.

Рассмотренные приложения:

Приложение VeraCrypt



Рисунок 1 — Логотип приложения VeraCrypt (<https://veracrypt.ru/>)

VeraCrypt – десктопное приложение для шифрования и дешифрования данных. Является продолжением разработки TrueCrypt, предоставляя улучшенные функции и безопасность.

VeraCrypt позволяет создавать зашифрованные тома и контейнеры, а также шифровать целые диски, включая системные диски. Вы можете использовать различные алгоритмы шифрования, включая AES, Serpent и Twofish.

В приложение есть инструкции по шифрованию/дешифрованию данных, что помогает при попытках разобраться с реализацией поставленной задачи.

Приложение GPG (GNU Privacy Guard)



Рисунок 2 — Логотип приложения GPG (<https://www.gnupg.org/>)  
GPG – приложение.

GPG – Инструментом для шифрования и подписи данных. Это свободное и открытое программное обеспечение, предоставляющее реализацию стандарта OpenPGP (Pretty Good Privacy).

GPG обеспечивает шифрование данных и создание цифровых подписей, что делает его полезным инструментом для обеспечения конфиденциальности и аутентичности электронных сообщений.

С помощью GPG можно:

- Создавать и управлять парами ключей: позволяет вам генерировать ключи шифрования и подписи, а также управлять ими;
- Шифровать и дешифровать данные: можно использовать GPG для защиты конфиденциальных данных, шифруя их с помощью открытого ключа получателя;
- Создавать и проверять цифровые подписи: GPG позволяет создавать цифровые подписи для файлов и сообщений, а также проверять их подлинность;
- Обмен сообщениями с конфиденциальностью: можно использовать GPG для обмена зашифрованными сообщениями с другими пользователями GPG;

- Устанавливать доверие к открытым ключам: GPG предоставляет механизмы доверия, которые позволяют оценить, насколько можно доверять определенному открытому ключу.

GPG поддерживается на различных операционных системах, включая Linux, Windows и macOS. Отсутствует теоретическая часть, поэтому информацию о реализации заданий нужно искать в различных источниках.

### Приложение Cryptomator



Рисунок 3 — Логотип приложения Cryptomator

Cryptomator - десктопное приложение, предназначенное для обеспечения безопасности и конфиденциальности данных в облачном хранилище. Оно работает с различными облачными сервисами, такими как Dropbox, Google Drive, OneDrive и другими.

Cryptomator создает зашифрованный контейнер в облачном хранилище, который шифрует и дешифрует данные локально на компьютере перед отправкой их в облако. Это предотвращает доступ к вашим данным облаком или сторонним лицам без правильного ключа шифрования.

### Приложение AES Crypt



Рисунок 4 — Логотип приложения AES Crypt (<https://www.aescrypt.com/>)

AES Crypt — это простое в использовании десктопное приложение, предназначенное для шифрования файлов с использованием алгоритма



шифрования AES (Advanced Encryption Standard). AES является одним из наиболее распространенных и надежных алгоритмов шифрования, и его использование гарантирует высокий уровень безопасности ваших данных.

AES Crypt позволяет:

- Зашифровать файлы: можно выбрать файлы, которые нужно зашифровать, и использовать AES Crypt, чтобы преобразовать их в зашифрованный формат. Требуется пароль для доступа к данным в будущем;
- Расшифровать файлы: для расшифровки файлов требуется AES Crypt и правильный пароль. После ввода пароля приложение расшифрует файлы и сделает их доступными;
- Кроссплатформенность: AES Crypt поддерживается на различных операционных системах, включая Windows, macOS и Linux, что делает его удобным для использования на разных платформах;
- Открытое программное обеспечение: AES Crypt является открытым программным обеспечением с открытым исходным кодом, что позволяет проверить его безопасность и надежность;
- Интеграция с проводником (Explorer): на Windows AES Crypt может интегрироваться с Проводником, что облегчает шифрование и дешифрование файлов из контекстного меню.

Из-за открытого программного обеспечения можно просмотреть реализацию методов шифрования/дешифрования, что помогает разобраться в работе программы

Приложение DiskCryptor



Рисунок 5 — Логотип приложения DiskCryptor (<https://diskcryptor.org/>)

DiskCryptor — это десктопное приложение, предназначенное для шифрования дисков и дисковых разделов на компьютере. Это позволяет защитить целый диск, включая операционную систему, от несанкционированного доступа.

Приложение шифрует данные непосредственно системы, что способствует автоматической конфиденциальности

Сайт Crypto 101



Рисунок 6 — Логотип сайта Crypto 101 (<https://www.crypto101.io/>)

Crypto101 — Это онлайн-книга, предоставляющая введение в основные понятия криптографии. Она охватывает как базовые принципы, так и более сложные темы, представленные в доступной форме для начинающих.

Основные особенности Crypto101:

- Бесплатный доступ: Ресурс полностью бесплатен для использования. Это позволяет всем заинтересованным лицам получить базовое понимание криптографии без каких-либо затрат.
- Интерактивность: Crypto101 представляет информацию в интерактивной форме. Он объясняет ключевые концепции криптографии через примеры, задачи и практические сценарии, что делает процесс изучения более увлекательным и понятным.
- Широкий охват тем: Ресурс охватывает различные аспекты криптографии, включая базовые алгоритмы шифрования, цифровые подписи, протоколы аутентификации и другие ключевые концепции.
- Понятный язык: Crypto101 старается избегать излишней технической терминологии, предоставляя информацию о криптографии в понятной

форме, доступной людям без глубоких знаний математики или криптографии.

- Онлайн-доступ: Ресурс доступен в Интернете, что обеспечивает удобство доступа для всех, кто заинтересован в изучении криптографии, в любое удобное время.

Crypto101 может быть полезным для тех, кто только начинает свой путь в изучении криптографии или желает получить базовое понимание основных принципов безопасности данных и шифрования.

Сайт Cryptopals

## the cryptopals crypto challenges

Рисунок 7 — Логотип сайта Cryptopals (<https://cryptopals.com/>)

Cryptopals — это набор практических задач, созданных для того, чтобы помочь людям разобраться с основами криптографии и криптоанализа путем решения реальных задач.

Основные характеристики и особенности Cryptopals:

- Набор задач: Cryptopals предоставляет последовательность из нескольких десятков задач, каждая из которых фокусируется на различных аспектах криптографии, начиная от базовых шифров и заканчивая более сложными протоколами.
- Практический подход: Все задачи предполагают реальную практику. Участники должны решать задачи, проводить анализ шифров, писать код для криптоанализа и применять различные методы атак для понимания уязвимостей в криптографических системах.
- Постепенная сложность: Задачи в Cryptopals устроены таким образом, что они начинаются с простых и постепенно становятся все более

сложными. Это помогает учащимся систематически углублять свои знания и навыки в области криптографии.

- Открытый доступ: Cryptopals доступен онлайн бесплатно для всех желающих. Решения задач и форумы обсуждений помогают участникам изучать материалы и обмениваться знаниями.
- Подходит для программистов и криптографов: Несмотря на то, что задачи могут требовать некоторых знаний в программировании, они также обеспечивают хороший старт для людей, интересующихся криптографией и безопасностью данных.

Cryptopals — отличный способ для тех, кто хочет не только изучать теоретические аспекты криптографии, но и практически применять свои знания для анализа и преодоления различных типов криптографических защит.

Анализ существующих систем показал, что приложения по криптографии данных различные задачи, необходимые для конфиденциальности данных. Основными функциями приложений в данной предметной области должен быть просмотр теоретической части, манипулирование с ключами и алфавитом в практической части, и просмотр результата. Исходя из полученных данных, приложение будет разрабатываться с учётом перечисленных функций.

### **1.3 Обзор и выбор системы управления баз данных**

В качестве СУБД приложения была выбрана MSSQL — это легковесная реляционная база данных. Она требует установки и настройки отдельного сервера баз данных. Это особенно полезно для проектов, где имеется необходимость в масштабируемости системы, а также в случаях, когда система должна быть легко переносимой между разными платформами.

MSSQL является бесплатной и с открытым исходным кодом, что уменьшает затраты на лицензии и поддержку, особенно для небольших и средних проектов.

MSSQL проста в использовании, масштабируемости и интеграции, поддерживает запросы на языке запросов SQL, что делает работу с данными и запросами удобной и интуитивно понятной. MSSQL способна обеспечить высокую производительность, а также справляться с большинством стандартных задач баз данных. Предоставляет механизмы для обеспечения целостности данных и транзакционной безопасности, что делает ее надежной базой данных для приложений, требующих сохранности данных. MSSQL поддерживает авторизацию и управление доступом к данным, что позволяет создавать приложения с уровнем безопасности, соответствующим требованиям.

MSSQL имеет различные API для взаимодействия с множеством языков программирования, такими как C#, Python, Java, и многими другими, что делает ее удобной для использования в различных экосистемах.

В случаях, когда проект ориентирован на небольшие объемы данных, такие как консольные команды для ОС Windows. MSSQL хорошо подходит, так как он не создает значительных накладных расходов на хранение и обработку данных.



Рисунок 8 — Логотип СУБД MSSQL

## 1.4 Обзор и выбор инструментов разработки

В процессе выбора инструментов для разработки, был сделан выбор в пользу языка программирования C#. Этот выбор обусловлен следующими факторами:

- Универсальность C#: C# предоставляет обширные возможности для создания разнообразных приложений, включая оконные приложения, веб-приложения, веб-сайты и видеоигры. Это позволяет создавать разносторонние проекты, адаптированные под различные платформы и задачи;
- Интеграция с базой данных: C# обладает сильными средствами для взаимодействия с базами данных, что является важным аспектом в современной разработке. Это обеспечивает возможность эффективно хранить, обрабатывать и извлекать данные в приложениях;
- Объектно-ориентированный подход: C# основан на объектно-ориентированной парадигме программирования. Этот подход позволяет создавать программы, организованные в виде взаимодействующих объектов, что способствует более четкой и модульной разработке. Это важно для создания поддерживаемых и масштабируемых проектов.

В итоге, выбор C# оправдывается его универсальностью, способностью взаимодействия с базами данных и объектно-ориентированным подходом, что делает его мощным инструментом для разработки разнообразных проектов и обеспечивает структурированный и гибкий подход к созданию программного обеспечения.



## Рисунок 9 — Логотип языка программирования c#

Для создания приложения рассматриваются технология C# WPF Entity Framework и ADO.NET.

ADO.NET (ActiveX Data Objects for .NET) — это набор технологий и библиотек, разработанных Microsoft для работы с данными в приложениях, созданных на платформе .NET. ADO.NET предоставляет средства для подключения к источникам данных, извлечения, обновления и управления данными в базах данных и других источниках данных. Он является частью .NET Framework и обеспечивает эффективное взаимодействие между .NET-приложениями и различными источниками данных, такими как базы данных SQL Server, Oracle, XML-документы и другие.

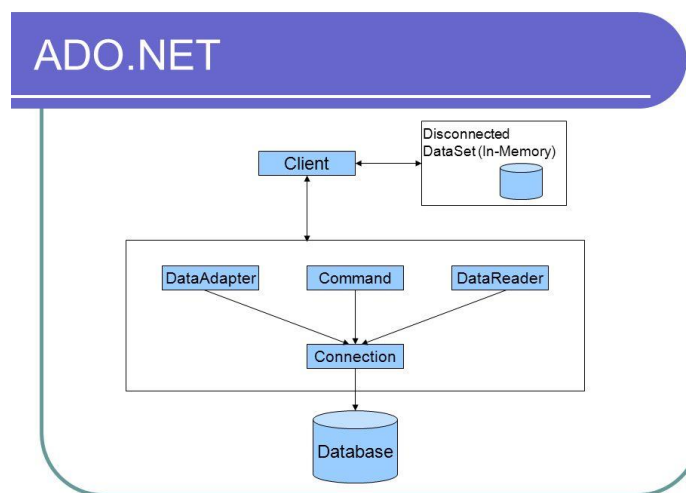


Рисунок 10 — Схема работы ADO.NET

Плюсами ADO.NET являются:

- Основной нужный функционал включен в ядро .NET Framework;
- Простота в использовании, для работы достаточно хорошо знать SQL язык;
- Стабильность и быстрота работы.

Минусами ADO.NET являются:

- Неудобство работы с объектами;
- Много повторяющихся блоков кода;

- Необходимость написания SQL-запросов.

Entity Framework - инструмент ORM (Object-Relational Mapping), позволяющий взаимодействовать с базами данных, используя объектно-ориентированный подход. EF Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL. позволяющая строить многоуровневые приложения, реализуя один из шаблонов проектирования MVC, MVP или MVVM.

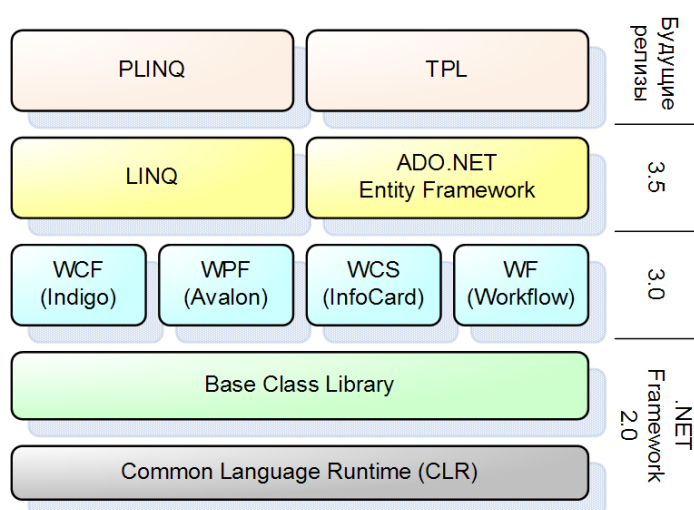


Рисунок 11 —Схема работы Entity Framework

Плюсами Entity Framework являются:

- Простота в использовании и повышение уровня абстракции;
- Генерация SQL-запросов: EF автоматически генерирует запросы на основе LINQ-запросов, что уменьшает вероятность ошибок и повышает производительность.

Минусы Entity Framework

- В некоторых случаях, при работе с большими объемами данных, Entity Framework может быть не так быстрым, как написание хранимых процедур или ручное написание SQL-запросов.;
- Возможная сложность написания LINQ запросов по сравнению с использованием хранимых процедур.



Так, как в приложение есть база данных, необходим инструмент, который позволяет эффективно работать с бд, поэтому выбран Entity Framework.

## **1.5 Выбор и характеристика среды разработки приложения**

В качестве среды разработки была выбрана Microsoft Visual Studio, так как она предоставляет множество удобств для разработчика:

- Visual Studio содержит все необходимые инструменты и функции, такие как редактор кода, отладчик, компилятор и многие другие;
- Совместимость и удобство: Visual Studio обладает превосходной интеграцией с C# - это основной язык программирования для разработки на платформе .NET. Множество инструментов и шаблонов проектов делают разработку на C# более удобной и эффективной.
- Работа с базами данных: В Visual Studio интегрирована поддержка работы с базами данных, включая подключение и управление MSSQL. Это облегчает разработку приложений, взаимодействующих с базами данных, с помощью C#.
- Entity Framework и LINQ: Visual Studio предоставляет средства для работы с Entity Framework и LINQ, что упрощает доступ и манипулирование данными в MSSQL из C#. Entity Framework позволяет создавать объектно-ориентированные модели на основе схемы базы данных, а LINQ предоставляет удобный способ запроса данных.
- Visual Studio имеет мощный отладчик, который позволяет находить и исправлять ошибки в коде. Он также поддерживает множество отладочных функций, таких как точки остановок, просмотр переменных и значения, трассировка стека вызовов и др;

- Понятный интерфейс: Интерфейс Visual Studio был разработан, чтобы сделать работу программиста простой и удобной. Все инструменты и функции легко доступны благодаря простой структуре меню и панелей инструментов.



Рисунок 12 — Логотип средства разработки Visual Studio

В следствие проведения анализа средств для создания приложения был сделан выбор в пользу средства разработки - Visual Studio в связи с тем, что данное средство представляет собой мощную и гибкую среду разработки, особенно эффективную в контексте работы с языком программирования C# и базой данных MSSQL. Её интеграция с этими технологиями обеспечивает удобство и эффективность при создании приложений, включающих в себя как бизнес-логику на C#, так и взаимодействие с данными в MSSQL.

Отладка, инструменты для работы с базами данных, поддержка языка C# и возможности расширения делают Visual Studio предпочтительным выбором для разработчиков, стремящихся создавать качественное программное обеспечение под управлением платформы .NET. Интеграция с MSSQL облегчает работу с данными и позволяет эффективно взаимодействовать с базой данных, что является важным аспектом многих приложений. Таким образом, использование Visual Studio для разработки на C# с поддержкой MSSQL дает разработчикам мощные инструменты для создания надежных и производительных приложений.

## **2. ПРАКТИЧЕСКАЯ ЧАСТЬ**

### **2.1 Разработка приложения**

Разработка функций приложения будет проходить в несколько этапов для правильного распределить времени и шагов реализации поставленной задачи.

Проект "Разработка обучающей программы по теме «Криптография данных»" представляет собой систему, построенную на трехзвенной архитектуре, разделенной на Model, View Model и View.

Паттерн MVVM (Model-View-ViewModel) позволяет отделить логику приложения от визуальной части (представления). Состоит из трех компонентов: модели (Model), модели представления (ViewModel) и представления (View).

Данный паттерн является архитектурным, то есть он задает общую архитектуру приложения. Представляет собой абстракцию данных и методов для их обработки. Модель не зависит от представления и контроллера. Представление отвечает за отображение данных пользователю и предоставляет пользовательский интерфейс. Получает данные из модели и отображает их в удобном для пользователя виде. Представление не содержит логику обработки данных. Модели представления является посредником между моделью и представлением. Он обрабатывает запросы пользователя, взаимодействует с моделью для получения или изменения данных, и обновляет представление. Модель представления содержит логику обработки пользовательских действий.

Для работы с данными приложения находится база данных, которая хранит информацию о информации теории, шифрование и пользователей. Для управления базой данных используется СУБД, которая обеспечивает надежное хранение и быстрый доступ к данным.

Используется Модель, разработанная на C#. Этот компонент отвечает за обработку запросов от базы данных, создавая объекты для работы в приложении.

В Модели представления находятся обработчики действий пользователя для манипулирования Моделью, добавляя, удаляя и редактируя доступные ему значения, предоставляя актуальные данные. Так обеспечивается

В Представление находятся элементы интерфейса, через которые пользователь взаимодействует с моделью представления для отделения логики приложения и её обработки.

На следующем этапе разрабатывается удобный, но интуитивно понятный интерфейс. Будут использованы элементы управления, чтобы пользователи могли легко осуществлять доступные операции. Приложение будет реализовывать два модуля: модуль теоретической части и модуль практической части. В меню приложения в зависимости от типа пользователя можно будет выбирать переход на нужную часть приложения. Так для обычного пользователя доступ к практической части будет только после прохождения теоретической составляющей приложения, а администратор может независимо переходить в нужные разделы программы.

После этого начнется работа над реализацией функционала приложения. Для работы с базой данных будет использована технология Entity Framework, Реализация функций по средству Entity Framework избежать SQL-инъекций и обеспечат более структурированный подход к работе с базой данных. Для написания кода будет использоваться язык C# и платформа WPF. Это обеспечит простоту разработки и поддержки кода в дальнейшем.

На последнем этапе будет проведено тестирование приложения и отладка. Это позволит выявить и исправить возможные ошибки и недочеты в функционале и интерфейсе приложения. Тестирование будет проводиться

как автоматически, с использованием специальных инструментов, так и вручную, чтобы убедиться в корректной работе всех функций приложения.

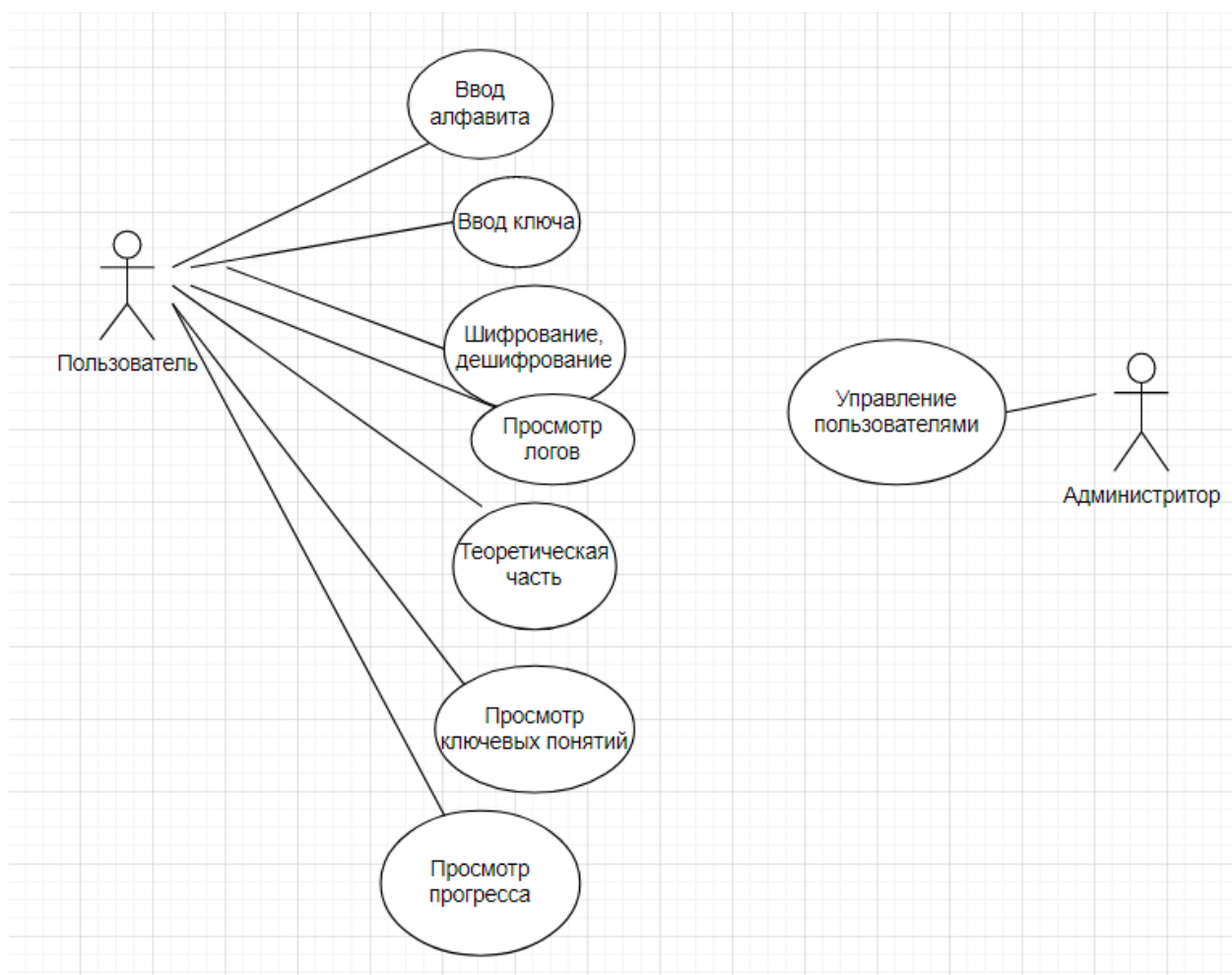


Рисунок 12. UML Диаграмма вариантов использования

## 2.2 Логическая модель базы данных в ER Assistant

Перед началом работы с устройством базы данных была разработана логическая модель. Для разработки логической модели использовалось бесплатное программное обеспечение ER Assistance. Готовая логическая схема представлена на Рисунке 13.

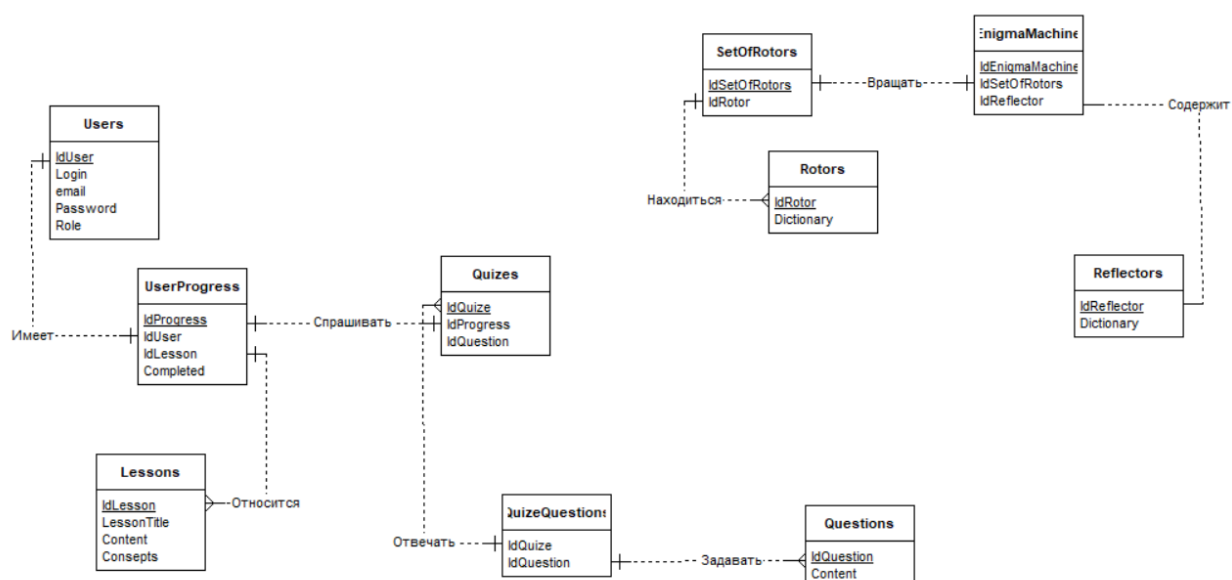


Рисунок 13. Логическая модель базы данных

### 2.3 Тестирование приложения

После разработки приложения было произведено ручное тестирование основных функций.

Таблица 3 – Метод проверки требований к приложению

| № | Действие  | Результат   | Ожидаемый результат                                       |
|---|---|---|---|
| 1 | Запуск приложения   | Приложение подключается к серверу и базе данных           | Приложение подключается к серверу и базе данных           |
| 2 | Пользователь проходит авторизацию   | Открывается окно авторизации                              | Открывается окно авторизации                              |
| 3 | Пользователь нажимает на кнопку ознакомления с теоретической частью.  | Открывается окно с информацией о теории и её разделы      | Открывается окно с информацией о теории и её разделы      |
| 4 | Пользователь нажимает кнопку «Назад»  | Пользователь переходит на предыдущее окно                 | Пользователь переходит на предыдущее окно                 |
| 7 | Пользователь нажимает кнопку «Переход к шифровальной машине»  | Открывается окно с интерфейсом шифровальной машины        | Открывается окно с интерфейсом шифровальной машины        |
| 8 | Вводится пользовательское сообщение, выбираются роторы, рефлексор, углубление и поворот для роторов. Пользователь | Пользовательское сообщение шифруется и выводится на экран | Пользовательское сообщение шифруется и выводится на экран |

|  |                                  |  |  |
|--|----------------------------------|--|--|
|  | нажимает кнопку<br>“Зашифровать” |  |  |
|--|----------------------------------|--|--|

## 2.4 Требования к техническим средствам

Для корректной работы приложения требуется соблюдать данные технические характеристики:

1. Минимальный объем оперативной памяти: 4 ГБ
2. Минимальные требования к монитору: Super VGA с разрешением 800х600 пикселей или более высоким.
3. Свободное место на диске: 150 МБ (Без установки SQL Server)
4. Доступ в Интернет
5. Тип процессора: AMD Opteron, AMD Athlon 64, Intel Xeon с поддержкой Intel EM64T, Intel Pentium IV с поддержкой EM64T.
6. Быстродействие процессора: частота 2,0 ГГц и выше

## 2.5 Требования к программным средствам

Для корректной работы приложения, программное обеспечение должно соответствовать данному списку:

1. Операционная система: Windows 10 или Windows 11
2. MS SQL Server Management Studio
3. База данных в MS SQL Server
4. .NET Framework
5. Последние драйверы для системы

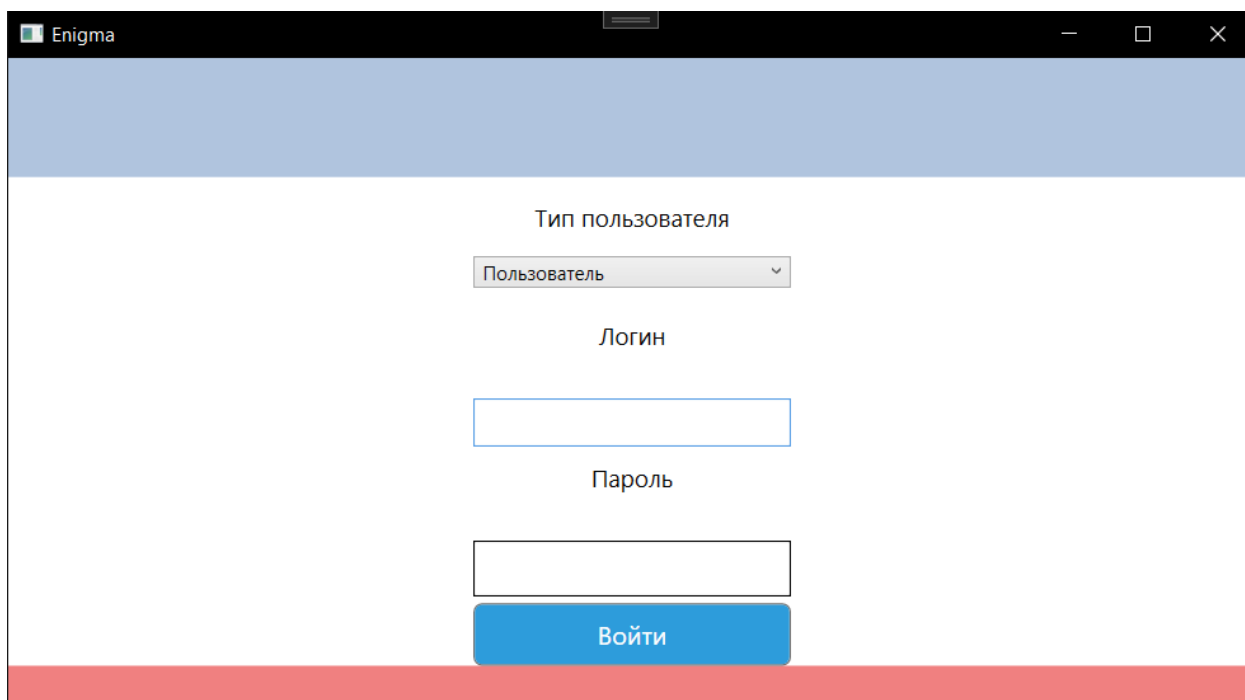
## 2.6 Настройка информационной системы

Для интеграции системы в работу понадобится:

1. Проверить, что система соответствует минимальным системным требованиям

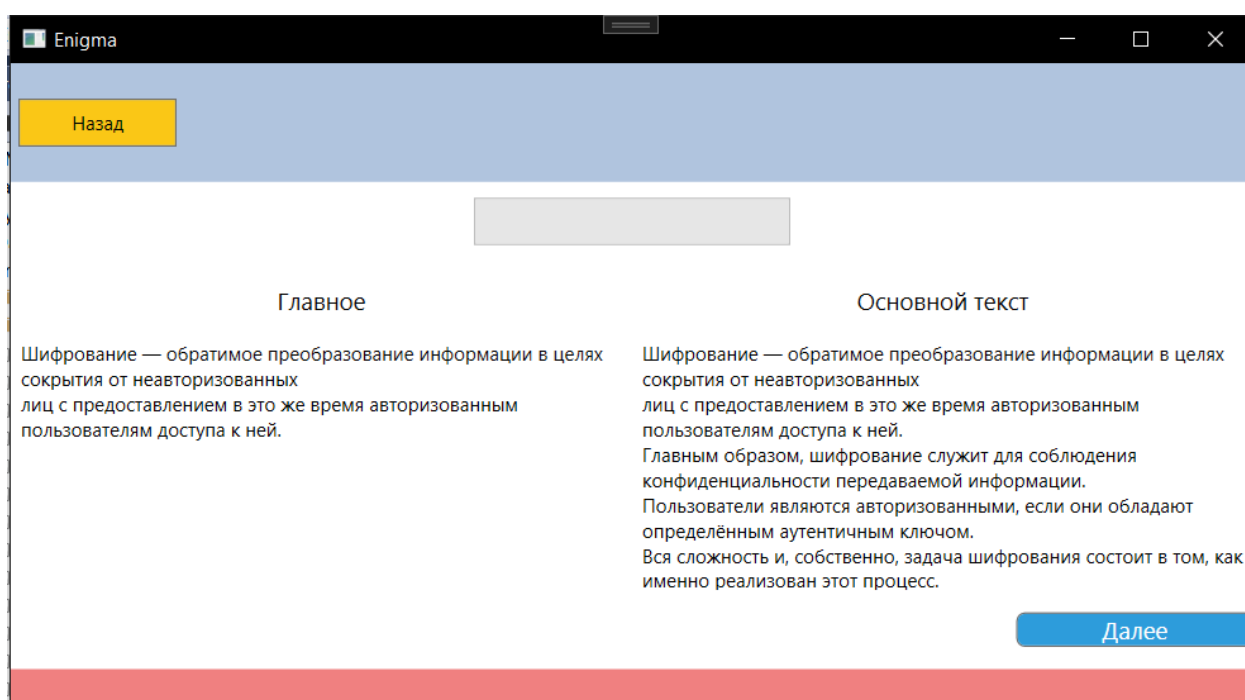
2. Установить всё необходимое программное обеспечение
3. Настроить MS SQL Server
4. Выполнить скрипты на создание базы данных и хранимых процедур
5. Заменить строку подключения в конфигурационном файле приложения

## 2.7 Демонстрация готового продукта



The screenshot shows a window titled 'Enigma' with a blue header bar. Below the header, there is a form for user authentication. It includes a dropdown menu labeled 'Тип пользователя' (User type) with 'Пользователь' (User) selected. Below this are two text input fields labeled 'Логин' (Login) and 'Пароль' (Password). At the bottom of the form is a blue button labeled 'Войти' (Login). The window has a red footer bar.

Рисунок 14. Окно «Авторизация»



The screenshot shows a window titled 'Enigma' with a blue header bar. In the top left corner of the main area is a yellow button labeled 'Назад' (Back). Below the header, there is a large grey rectangular area. Below this, the main content is divided into two columns. The left column is titled 'Главное' (Main) and contains text about encryption. The right column is titled 'Основной текст' (Main text) and contains more text about encryption. At the bottom right of the main content area is a blue button labeled 'Далее' (Next). The window has a red footer bar.



Рисунок 15. Окно «Теоретическая часть»

The screenshot shows a window titled 'Enigma' with a blue header bar. Below the header is a yellow button labeled 'Назад'. The main content area is white and contains the title 'Опрос' (Survey) in bold. Below this is the question 'Шифрование данных это -' (Data encryption is -). There are three radio button options: 'Обратимое преобразование информации в целях её сокрытия' (Reversible transformation of information for its concealment), 'необратимое сообщение' (Irreversible message), and 'Набор символов' (Set of symbols). At the bottom right is a yellow button labeled 'Ответить' (Answer). A red bar is visible at the very bottom of the window.

Рисунок 16. Окно «Опрос»

The screenshot shows a window titled 'Enigma' with a blue header bar. Below the header is a text input field labeled 'Шифруемое сообщение' (Message to be encrypted) containing the text 'THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG'. Below this are three rows of rotors, each with a label 'Ротор 1', 'Ротор 2', and 'Ротор 3'. Each rotor has a numeric input field (1, 2, 3) and a letter input field (A, B, C). To the right of the rotors are three columns of settings: 'Углубление' (Stepping) with input fields 'Y', 'M', 'D', and 'Поворот' (Turning) with input fields 'Q', 'E', 'V'. Below the rotors is a 'Рефлектор:' (Reflector) dropdown menu showing 'UKW-B'. At the bottom right are two blue buttons: 'Дешифровать' (Decrypt) and 'Зашифровать' (Encrypt). A red bar is visible at the very bottom of the window.

Рисунок 17. Окно «Практическая часть»

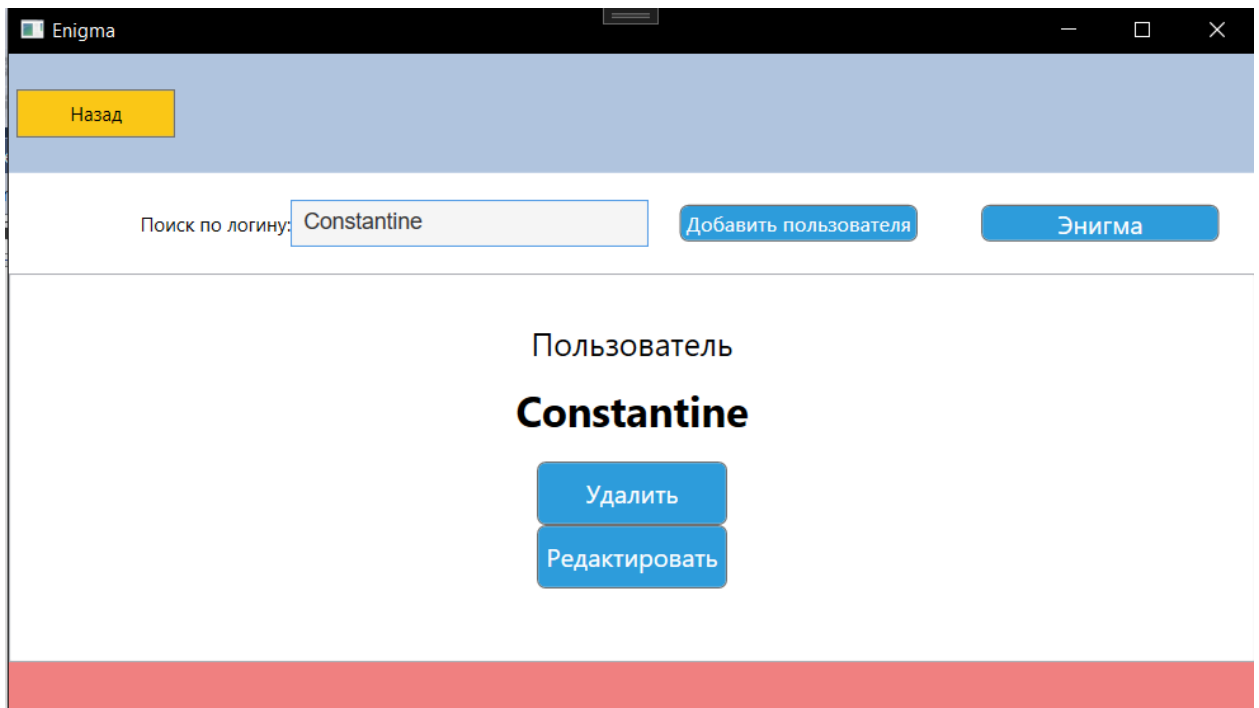


Рисунок 18. Окно «Интерфейс администратора»

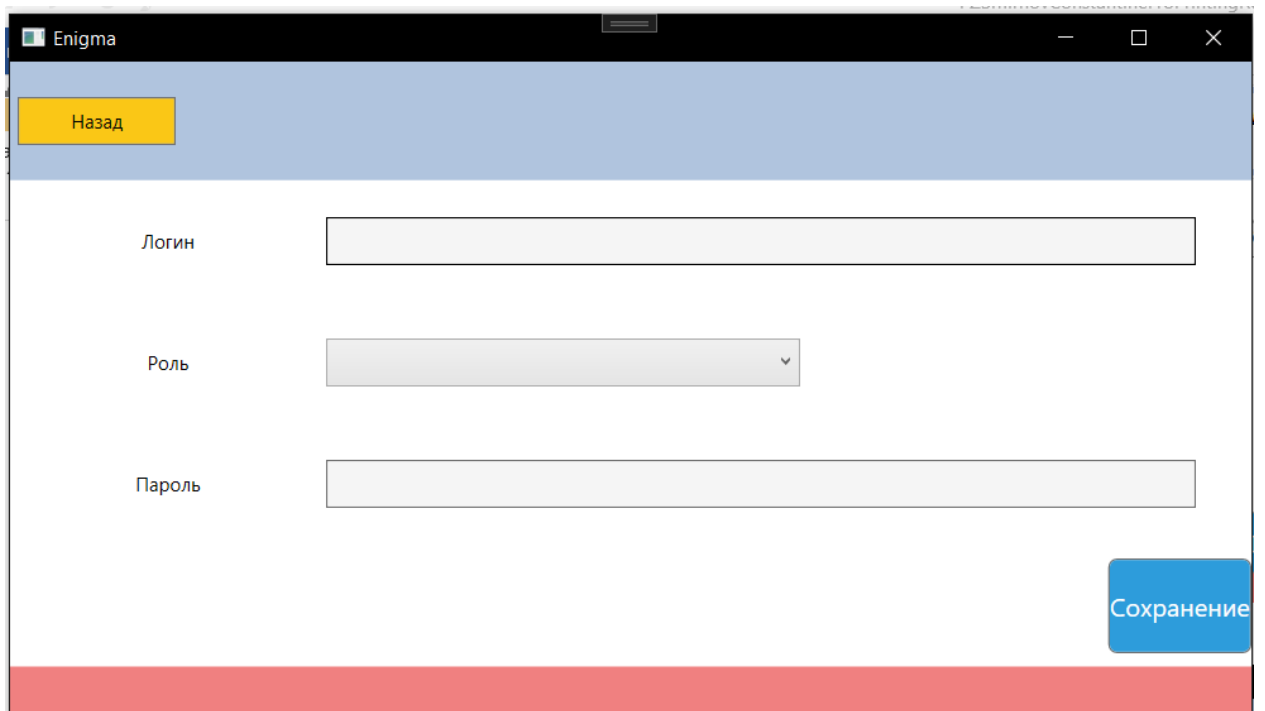


Рисунок 17. Окно «Редактирование пользователей»

## **ЗАКЛЮЧЕНИЕ**

Курсовой проект, включающий в себя обучение криптографии, и реализация шифровальной машины, был реализован. В ходе его выполнения были более подробно изучены основные принципы работы с C# WPF и SQL, а также интеграция базы данных MS SQL Server в приложение. Также были проанализированы существующие решения в данной предметной области.

Была спроектирована база данных MS SQL Server и разработано приложение с использованием C# WPF и Entity Framework, которые позволяют пользователю и администратору корректно работать с теоретической и практической частями программы.

Основной целью проекта было изучение языка C#, возможностей работы MS SQL Server и Entity Framework.

Разработанное приложение имеет потенциал для использования в обучение и может быть усовершенствовано в дальнейшем.

Дальнейшее развитие темы курсового проекта предполагает расширение функционала приложения, например, добавление новых реализаций алгоритмов шифрования данных. Также, предполагается дальнейшее изучение C# WPF и возможностей MS SQL Server.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Заполнить DataGridView – URL: <https://www.cyberforum.ru/windows-forms/thread1988785.html>
2. Beginners guide to accessing SQL Server through C# - URL: <https://www.codeproject.com/Articles/4416/Beginners-guide-to-accessing-SQL-Server-through-C>
3. WPF Calendar disable date selection - URL: <https://stackoverflow.com/questions/56188607/wpf-calendar-disable-date-selection>
4. ADO.NET. - URL: <https://ru.wikipedia.org/wiki/ADO.NET>
5. Сохранение изображения в БД C# – URL: <https://habr.com/ru/articles/700406/>
6. Сохранение и извлечение файлов из базы данных – URL: <https://metanit.com/sharp/adonetcore/2.13.php>
7. Аксёнов А. Язык программирования C#. В подлиннике. Москва: Издательство, 2022.
8. Троелсен Э. C# 9 и .NET 5. Разработка профессиональных приложений. Москва: Издательство, 2021.
9. Скит Д. C# 9.0 и .NET 5.0. Руководство для начинающих. Москва: Издательство, 2021.
10. Нейгел К. C# 9.0 и платформа .NET 5.0 для профессионалов. Москва: Издательство, 2021.
11. Албахари Д. C# 9.0 и платформа .NET 5.0. Карманный справочник. Москва: Издательство, 2021.
12. Скит Д. Программирование на C#. Шаг за шагом. Москва: Издательство, 2020.
13. Нейгел К. C# 8.0 и платформа .NET Core 3.0 для профессионалов. Москва: Издательство, 2020.
14. Албахари Д. C# 8.0 и платформа .NET Core 3.0. Карманный справочник. Москва: Издательство, 2020.

15. Троелсен Э. С# 8.0 и .NET Core 3.0. Разработка профессиональных приложений. Москва: Издательство, 2020.
16. Аксёнов А. Язык программирования С#. Учебник и справочник. Москва: Издательство, 2020.
17. Шилдт Г. С# 9.0. Полное руководство. Москва: Издательство, 2022.
18. Коньков А. С# 9.0 и .NET 5.0. Программирование для профессионалов. Москва: Издательство, 2022.
19. Петцольд Ч. Программирование на платформе .NET с использованием С#. Москва: Издательство, 2022.
20. Рихтер Д. CLR via С#. Программирование на платформе Microsoft .NET Framework 4.5 на языке С#. Москва: Издательство, 2021.

## ПРИЛОЖЕНИЕ А. Код программного продукта

В приложении далее представлены наиболее важные и интересные части кода. Полный исходный код программы представлен на носителе и в GitHub по ссылке: <https://github.com/avonavia/PokemonMicroBattler>

```
using EnigmaProject.Model;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

using EnigmaProject.Components;
using System.IO;

namespace EnigmaProject.View
{
    public partial class EnigmaAPI : Page
    {
        private int Choice;
        public EnigmaAPI()
        {
            InitializeComponent();
            Rotor1.ItemsSource = EnigmaBase.GetContext().Rotors.ToList();
            Rotor1.SelectedIndex = 0;
            Rotor2.ItemsSource = EnigmaBase.GetContext().Rotors.ToList();
            Rotor2.SelectedIndex = 1;
            Rotor3.ItemsSource = EnigmaBase.GetContext().Rotors.ToList();
            Rotor3.SelectedIndex = 2;
            Reflector.ItemsSource = EnigmaBase.GetContext().Reflectors.ToList();
            Reflector.SelectedIndex = 1;
            DataTextBox.Text = "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG";
        }

        public string Operation(string text, int choice)
        {
            var selectedRotor1 = (Rotor)Rotor1.SelectedItem;
            if (selectedRotor1 == null)
                return "Ошибка";

            var selectedRotor2 = (Rotor)Rotor2.SelectedItem;
            if (selectedRotor2 == null)
                return "Ошибка";

            var selectedRotor3 = (Rotor)Rotor3.SelectedItem;
            if (selectedRotor3 == null)
                return "Ошибка";

            var selectedReflector = (Reflector)Reflector.SelectedItem;
```

```

        if (selectedRotor3 == null)
            return "Ошибка";

        // Rotors for encryption
        //1
        MyRotor rotor1 = new MyRotor($"{selectedRotor1.Dictionary}")
        {
            //сделать окно
            Notch = NotchRotor1.Text[0], //'Y',
            Turnover = NotchRotor1.Text[0] //'Q',
        };
        //2
        MyRotor rotor2 = new MyRotor($"{selectedRotor2.Dictionary}")
        {
            Notch = NotchRotor2.Text[0], //'M',
            Turnover = NotchRotor2.Text[0] //'E',
        };
        //3
        MyRotor rotor3 = new MyRotor($"{selectedRotor3.Dictionary}")
        {
            Notch = NotchRotor3.Text[0], //'D',
            Turnover = NotchRotor3.Text[0] //'V',
        };
        //A EJMZALYXVBWFCRQUONTSPIKMGD
        //B YRUHQSLDPXNGOKMIEBFZCWJAT
        //C FVPJIAOYEDRZXWGCTKUQSBNMHL
        MyRotor ReflectorB = new MyRotor($"{selectedReflector.Dictionary}");

        Enigma e = new Enigma();

        // Plugboard
        //сделать отдельное окно
        e.Plugboard.Add('X', 'D');
        e.Plugboard.Add('A', 'V');

        e.Rotors.Add(rotor1, HeadRotor1.Text[0]); //A
        e.Rotors.Add(rotor2, HeadRotor2.Text[0]); //B
        e.Rotors.Add(rotor3, HeadRotor3.Text[0]); //C

        // Reflector
        e.Rotors.SetReflector(ReflectorB);

        string answer = "";
        if (choice == 1)
            answer = e.Encrypt(text);
        else if (choice == 2)
            answer = e.Decrypt(text);
        e.Rotors.Clear();
        return answer;
    }

    private void EncryptButton_Click(object sender, RoutedEventArgs e)
    {
        using (StreamWriter writer = new StreamWriter("Encrypt.txt", true))
        {
            //шифрование
            writer.WriteLine($"{Operation(DataTextBox.Text, 1)}");
            MessageBox.Show("Сообщение зашифровано");
        }
    }

    private void DecryptButton_Click(object sender, RoutedEventArgs e)
    {
        string text;
    }

```

```

        //дешифрование
        using (StreamReader reader = new StreamReader("Encrypt.txt"))
        {
            text = reader.ReadToEnd();
            text = Operation(text, 2);
            MessageBox.Show("Сообщение Дешифровано");
        }
        //запись ответа
        using (StreamWriter writer = new StreamWriter("Decrypt.txt", true))
        {
            writer.WriteLine($"{text}");
            MessageBox.Show("Ответ записан");
        }
    }
}

```

Код Модели представления для обработки вопросов:

```

public class LessonVM : INotifyPropertyChanged
{
    private Lesson selectedLesson;

    public ObservableCollection<Lesson> Lessons { get; set; }
    public Lesson SelectedLesson
    {
        get { return selectedLesson; }
        set
        {
            selectedLesson = value;
            OnPropertyChanged("SelectedLesson");
        }
    }

    public LessonVM()
    {
        // Инициализация Lessons из контекста данных
        Lessons = new
        ObservableCollection<Lesson>(EnigmaBase.GetContext().Lessons);
    }

    public event PropertyChangedEventHandler PropertyChanged;
    public void OnPropertyChanged([CallerMemberName] string prop = "")
    {
        PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(prop));
    }
}

```

## ПРИЛОЖЕНИЕ Б. Описание таблиц базы данных

Таблица 1 – UserProgress (Прогресс пользователя)

| Название столбца | Описание                | Тип           | Примечание |
|------------------|-------------------------|---------------|------------|
| IdProgress       | Идентификатор прогресса | Int           | PK         |
| IdUser           | Номер пользователя      | nvarchar(255) |            |
| IdLesson         | Номер урока             | Int           |            |



|           |                 |     |  |
|-----------|-----------------|-----|--|
| Completed | Пройден ли урок | bit |  |
|-----------|-----------------|-----|--|

Таблица 2 – Users (Сообщение об ошибке)

| Название столбца | Описание        | Тип           | Примечание |
|------------------|-----------------|---------------|------------|
| IdUser           | Номер прогресса | Int           | PK         |
| LoginOfUser      | Логин           | nvarchar(100) |            |
| RoleOfUser       | Номер урока     | nvarchar(13)  |            |
| PasswordOfUser   | Выполнено       | nvarchar(100) |            |

Таблица 3 – Lessons (Уроки)

| Название столбца | Описание         | Тип            | Примечание |
|------------------|------------------|----------------|------------|
| IdLesson         | Номер урока      | Int            | FK         |
| LessonTitle      | Название урока   | nvarchar(100)  |            |
| Content          | Содержимое урока | nvarchar(2000) |            |
| Concepts         | Понятия          | nvarchar(1000) |            |
| IsCompleted      | Завершён ли урок | bit            |            |

Таблица 4 – EnigmaMachine (Шифровальная машина Энигма)

| Название столбца | Описание                  | Тип           | Примечание |
|------------------|---------------------------|---------------|------------|
| IdEnigmaMachine  | Номер шифровальной машины | int           | PK         |
| IdRotor          | Номер ротора              | nvarchar(255) |            |
| IdSetOfRotors    | Номер набора роторов      | Int           |            |
| IdReflector      | Номер рефлектора          | Int           |            |

Таблица 5 – Rotors (Роторы, которые вращаются и выполняют шифрование.)

| Название столбца | Описание                   | Тип | Примечание |
|------------------|----------------------------|-----|------------|
| IdRotor          | Номер данных для обработки | int | PK         |

|               |  |     |  |
|---------------|--|-----|--|
| IdSetOfRotors | Номер набора роторов                                   | int |  |
| Dictionary    | Словарь для использования при шифрование/дешифрование. | int |  |

Таблица 6 – Reflectors (Рефлектор, который является не вращающимся ротором и служащий для шифрования.)

| Название столбца | Описание              | Тип           | Примечание |
|------------------|-----------------------|---------------|------------|
| IdPlugBoard      | Номер плагборда       | int           | РК         |
| Dencrypt         | Алгоритм дешифрования | nvarchar(255) |            |

Таблица 7 – SetOfRotors (Набор роторов для шифрования)

| Название столбца  | Описание                | Тип           | Примечание |
|-------------------|-------------------------|---------------|------------|
| IdSetOfRotors     | Номер набора роторов    | int           | РК         |
| NameOfSetOfRotors | Название набора роторов | nvarchar(100) |            |

Таблица 8 – QuizQuestions (вопросы для опроса) )

| Название столбца | Описание       | Тип | Примечание |
|------------------|----------------|-----|------------|
| IdQuiz           | Номер опроса   | Int | РК         |
| IdQuestion       | Номер вопроса. | Int |            |

Таблица 9 – Quizes (Опрос)

| Название столбца | Описание                     | Тип | Примечание |
|------------------|------------------------------|-----|------------|
| IdQuiz           | Номер опроса                 | Int | PK         |
| IdProgress       | Номер прогресса пользователя | Int |            |

Таблица 9 – Questions (Вопросы)

| Название столбца | Описание           | Тип            | Примечание |
|------------------|--------------------|----------------|------------|
| IdQuestion       | Номер вопроса      | Int            | PK         |
| Content          | Содержимое вопроса | nvarchar(2000) |            |

Таблица 10 – Questions (Вопросы)

| Название столбца | Описание           | Тип            | Примечание |
|------------------|--------------------|----------------|------------|
| IdQuestion       | Номер вопроса      | Int            | PK         |
| Content          | Содержимое вопроса | nvarchar(2000) |            |