**1 слайд**

Темой работы является:

«Менеджер паролей»

**2 слайд**

С каждым днем число пользователей, использующих интернет-сервисы, увеличивается, и вместе с ним растут их требования к качеству и безопасности обработки и хранения персональных данных.

Одним из наиболее важных аспектов безопасности является сохранение конфиденциальности паролей пользователей.

**3 слайд**

Зачем это нужно?

Каждый пользователь компьютера рано или поздно сталкивается с проблемой хранения паролей, и каждый пользователь решает её по-разному.

Одни используют простые и запоминаемые пароли, лишая себя бремени самого хранения, другие записывают пароли на бумаге, третьи хранят свои пароли в текстовых файлах.

На помощь пользователю приходит менеджер паролей.

Они предоставляют надежное и удобное решение, позволяя хранить пароли в защищенной форме, генерировать сложные пароли и автоматически заполнять их при необходимости, обеспечивая безопасность и удобство для пользователей.

**4 слайд**

Аналоги

**5 слайд**

KeePass - это свободное и бесплатное приложение для управления паролями, которое позволяет сохранять и защищать пароли пользователей в зашифрованном виде.

Он бесплатный, имеет открытый исходный код, сохраняет пароли локально в зашифрованном файле. Но если попробовать создать новое хранилище паролей, то пользователя топят огромным количеством настроек, сложности ключа, выбора алгоритмов шифрования и так далее.

**6 слайд**

Pass – стандартное хранилище паролей для Unix –подобных систем. Приложение не имеет облачного хранения, что уменьшает риски взлома и утечки данных.

приложение не предоставляет удобный графический интерфейс, что может быть неудобно для пользователей, предпочитающих визуальную навигацию. Также его нет на windows, хотя по данным NetmarketShare — сайта, собирающего статистику об используемых ОС и браузерах, по состоянию на сентябрь 2022 года платформе Windows отдало предпочтение 87,88% пользователей, а ОС Linux — только 2,31% владельцев настольных компьютеров и ноутбуков.

**7 слайд**

Dashlane – менеджер паролей, который обладает интуитивно понятным интерйфейсом, а также имеет функционал автозаполнения форм паролей. К сожалению, он не представляет возможности хранить свои пароли на компьютере. Также он является платным, а следовательно, имеет закрытый исходный код.

**8 слайд**.

В основном я использовал стандартные инструменты, предоставленные в Visual Studio.

С# и его стандатные библиотеки предоставляют огромный инструментарий, позволяющий шифровать данные.

Windows forms позволяет создавать простые приложения с привычным пользователю интерфейсом.

Newtonsoft.Json использовался для работы с файлами.

**9 слайд.**

В начале разработки передо мной встала первостепенная задача: как хранить пароли?

Необходимо было придумать безопасный способ хранения пароля.

Для этого был реализован класс Password.

при создании экземпляра класса с помощью алгоритма AES и ключа пользователя шифруется пароль. Ключ пользователя преобразуется с помощью Rfc 2898 DeriveBytes. Это стандарт формирования ключа на основе пароля. В нашем случае – новый ключ генерируется на основе старого ,чтобы пользователь не беспокоился, что заданный им ключ при создании базы паролей был слабым. При запущенном приложении, поля обьекта защищены с помощью технологии ProtectedMemory, импортируемая стандартной системной библиотекой System.Security.Cryptography. Она шифрует блоки памяти вызовом метода Protect(). К сожалению, она шифрует блоки, кратные 16 байтам, поэтому к каждому полю класса идёт дополнительное поле, определяющее его длину. В открытом виде пароль появляется лишь на миг, в отдельном потоке, когда поступает команда скопировать его в буфер обмена на 10 секунд.

**10 слайд.**

Один пароль для пользователя ничего не значит. К нему необходима дополнительная инфомация, такая как: Название аккаунта, логин, телефон, секретный вопрос и т.д.

Для решения этой задачи мною был написан класс AccountEntry. В нем есть поля Login, Password, AccountName, и список дополнительных полей в Dictonary.

AccountEntry тем временем группируются в GroupEntry.

GroupEntry – это класс, который представляет собой древовидную структуру данных, в которой содержится List таких же GroupEntry, и List AccountEntry.

**11 слайд.**

Данное решение позволило удобно экспортировать данные в файл, так как Newtonsoft.Json позоволяет гиибко парсить обьекты в строку.

Пароль вместе с другими данными аккаунта безопасно хранятся в оперативной памяти. Теперь необходимо всё сохранить в файле.

Для работы с файлами были реализованы 2 класса: FileManager и FileEncryptor.

Эти 2 класса являются статическими, что облегчает процесс создания кода. Нет необходимости создавать экземпляры классов, чтобы использовать только один метод.

В FileManager передается корневая запись GroupEntry, которая парсится с помощью Newtonsoft.Json в строку. Далее эта строка шифруется AES, и записывается в файл.

В итоге в файле дважды зашифрованный разными ключами пароль и единожды зашифрованные остальные данные.

**12 слайд.**

Также был разработан графический интерфейс, который с легкостью может быть освоен рядовым пользователем компьютера, не имея углубленных знаний в сфере ИТ.

Привычный вид окна приложения не пугает пользователя излишней загруженностью функций предлагает легкое использование мною разработанного инструмента в полную силу.

Можно заметить сходство между интерфейсом аналога KeePass и моим менеджером, но KeePass крайне перегружен, что затрудняет навигацию по приложению.

В моем приложении первым делом можно заметить акцент на том, что пользователь будет делать чаще всего: добавлять или удалять записи. Всё просто – слева группы – справа аккаунты. Для каждой стороны свои кнопки. Рядом с «Файл» кнопка «Сохранить». Она, очевидно, сохраняет базу паролей в файл.

**13 слайд.**

Также интуитивно понятное заполнение аккаунта, простой инструмент генерации пароля и форма входа. Всё это постигаемо без сложностей, интуитивно понятно и не требует знаний самих алгоритмов для полного использования всего функционала приложения.

**14 слайд**

В ходе выполнения выпускной квалификационной работы был разработан инструмент хранения, редактирования и генерирования паролей, с использованием языка C#, как основного языка программирования.

Приложение предоставляет пользователю возможность импортировать и хранить информацию о пользовательских учетных записях, организовывать пароли по категориям и выполнять поиск и фильтрацию для удобного доступа.

Данная работа открывает перспективы для импортирования разработанной технологии менеджера паролей на мобильные устройства, такие как смартфоны и планшеты. Это позволит пользователям удобно и безопасно управлять своими паролями в любое время и в любом месте. Приложение для мобильных устройств может предоставлять аналогичные функции хранения, генерации и управления паролями, а также синхронизировать данные с версией для настольных компьютеров.