Творческий проект по технологии на тему:

**«Программа для защиты от мошеннических сайтов»**

**Шифр работы:**

2025 г.

# 

# Реферат

Отчёт 35 с., 4 ч., 16 рис., 3 табл., 7 источников, 4 прил.

ПРОГРАММА ДЛЯ ЗАЩИТЫ ОТ МОШЕННИЧЕСКИХ САЙТОВ

Объектом исследования являются мошеннические сайты и методы защиты от них.

Цель работы – программа для защиты от мошеннических сайтов.

В результате исследования была создана программа для защиты от мошеннических сайтов.

# Оглавление

[1. ВВЕДЕНИЕ 7](#_Toc194306015)

[1.1. Проблема и её актуальность 7](#_Toc194306016)

[1.2 Цели и задачи проекта 10](#_Toc194306017)

[2. ПОДГОТОВИТЕЛЬНЫЙ ЭТАП 11](#_Toc194306018)

[2.1 Анализ угроз фишинга 11](#_Toc194306019)

[2.2 Анализ целевой аудитории 12](#_Toc194306020)

[2.3 Интеграция 12](#_Toc194306021)

[2.4 Анализ соответствия нормативно-правовой базе 13](#_Toc194306022)

[2.5 Выбор инструментов и методов 13](#_Toc194306023)

[2.6 Оригинальность и сравнение с аналогами 15](#_Toc194306024)

[3. ТЕХНОЛОГИЧЕСКИЙ ЭТАП 17](#_Toc194306025)

[3.1 Концепция будущего продукта и его характеристики 17](#_Toc194306026)

[3.2 Технические характеристики будущего продукта 17](#_Toc194306027)

[3.3 Процесс разработки программы (ручная проверка) 17](#_Toc194306028)

[3.4 Процесс разработки программы (автоматическая защита) 21](#_Toc194306030)

[3.5 Дальнейшая оптимизация алгоритма работы 22](#_Toc194306031)

[3.6 Проведение тестов 24](#_Toc194306032)

[4. ЗАКЛЮЧЕНИЕ 25](#_Toc194306033)

[4.1 Дальнейшее развитие проекта 25](#_Toc194306034)

[4.2 Демоверсия программы 25](#_Toc194306035)

[4.3 Итог 25](#_Toc194306036)

[Список использованной литературы 27](#_Toc194306037)

[Приложение №1. Графики и диаграммы 28](#_Toc194306038)

[Приложение №2. Программа 30](#_Toc194306039)

[Приложение №3. Схемы и иллюстрации 33](#_Toc194306041)

[Приложение №4. Таблицы 34](#_Toc194306042)

# ВВЕДЕНИЕ

## Проблема и её актуальность

В настоящее время наблюдается стремительное увеличение числа киберпреступлений. Одним из самых распространённых видов киберпреступления является фишинг. В этом случае жертва, поддавшись обману, неосознанно передаёт конфиденциальную информацию, которую мошенники затем используют для нанесения материального вреда.

С каждым годом количество фишинговых атак увеличивается, а их методы модернизируются. Ключевым фактором успеха таких атак остаётся человеческий фактор: мошенники активно используют социальную инженерию, манипулируя эмоциями и доверием жертв.

Анализ актуальности проблемы основан на отчёте “Лаборатории Касперского” по спаму и фишингу за 2023 год. Основной целью фишинговых атак в последние года является почта – 96% от всех атак. Для понимания цельной картины стоит рассмотреть данные как на локальном уровне, используя опрос в школе, так и на глобальном уровне, используя отчёты и исследования больших компаний. В опросе участвовали школьники 10 классов, в сумме 26 человек. Из результатов следует, что более 50% человек сталкиваются с фишингом на постоянной основе.

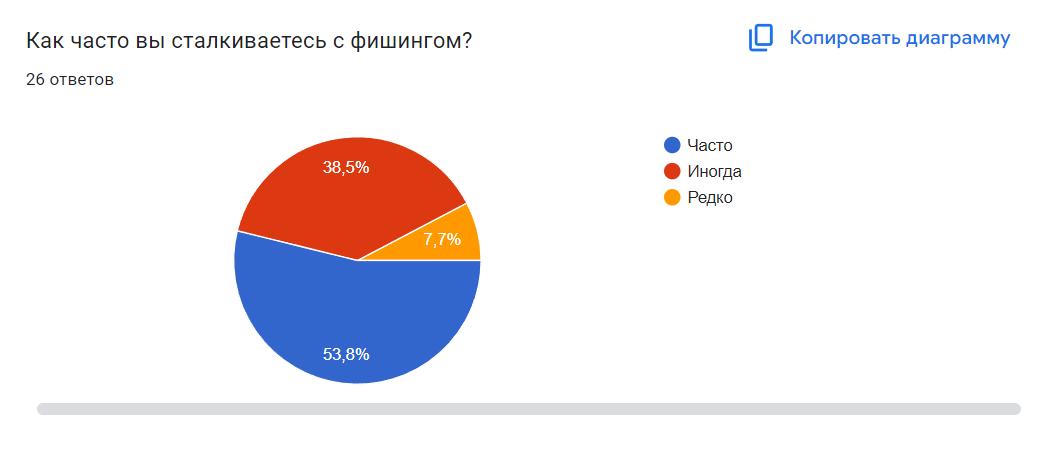


Рис. П1.1 Пункт опроса про фишинг

На Рис. П.2. мы можем видеть диаграмму, на которой показана доля спама в почтовом трафике российского интернета. В среднем на протяжении 2023 года она составляла 46,59%, что является крайне высоким показателем и указывает на то, что почти каждое второе письмо является спамом. Из исследования также можно понять, что основными инструментами фишинга на почте являются фишинговые ссылки и вредоносные файлы. Рассмотрим Рис. П.3. на котором изображена диаграмма количества срабатываний антивируса на попытки открытия вредоносных файлов во вложениях к письмам и Рис. П.4. на котором изображена диаграмма количества срабатываний антивируса на попытки перехода по вредоносным ссылкам в письмах. Всего попыток открыть вредоносные файлы было 136 миллионов, а попыток перейти по опасным ссылкам – 710 миллионов. По этим данным мы можем судить, что количество попыток открыть вредоносные ссылки преобладает более чем в 5 раз. Соответственно, проблема фишинговых сайтов актуальней и подлежит скорейшему решению.

Рис. П1.2. Доля спама в почтовом трафике российского интернета

Рис. П1.3. Количество срабатываний почтового антивируса на попытки открыть вредоносный файл

Рис. П1.4. Количество срабатываний системы “Антифишинг” на попытки перехода по мошеннической ссылке

## 1.2 Цели и задачи проекта

**Цель работы**: Создание программы по защите от вредоносных сайтов/ссылок, удобной массовому пользователю.

**Задачи:**

1. Изучить угрозы, которые несут мошеннические сайты;
2. **Провести анализ целевой аудитории;**
3. Выбрать оптимальные методы и инструменты для разработки решения проблемы;
4. Рассмотреть аналоги решения проблемы и аналоги продукта;
5. Разработать рабочий прототип программы;
6. Провести опрос про удобство рабочего прототипа;
7. Составить план дальнейшего развития проекта;

# 2. ПОДГОТОВИТЕЛЬНЫЙ ЭТАП

## 2.1 Анализ угроз фишинга

Существует много видов фишинга, связанных с почтой, но выделить можно несколько основных:

1. Email/spam – наиболее распространенный вид фишинга. Поддельное письмо с вредоносным файлом/ссылкой отправляется огромному количеству пользователей, в надежде на то, что фишинг атака закончится успехом.
2. Spear phishing – похож на спам, но отличается тем, что атака становится более целенаправленной. Злоумышленники знают кого преследуют, и заранее исследуют цель для подготовки наиболее удачной атаки. Может затрагивать как обычных пользователей, так и организацию.
3. Whaling – метод не сильно отличающийся от spear phishing, но целевая группа становится более специфичной и ограниченной. Этот метод нацелен на руководящие должности.
4. Clone phishing – метод, в котором настоящее и ранее отправленное электронное письмо, содержащее вложение или ссылку, используется для создания идентичного письма, в котором настоящие вложения/ссылки будут заменены на вредоносные.

Как мы увидели по анализу актуальности – большинство методов фишинга связаны с вредоносными ссылками, вследствие чего решать мы будем именно эту проблему. Для разработки методов борьбы с фишинговыми сайтами необходимо, прежде всего, изучить угрозы, которые они представляют. Поскольку технологии мошенничества постоянно развиваются, важно сосредоточиться на актуальных угрозах, чтобы понять общие принципы работы таких сайтов. Несмотря на разнообразие фишинговых сайтов, их механизмы обмана часто схожи. Для понимания этих механизмов достаточно рассмотреть несколько типичных примеров:

1. **Сайты для геймеров с предложением покупки игр или внутриигровой валюты**. Пользователь, надеясь получить товар по выгодной цене, вводит свои персональные данные и оплачивает покупку, но в результате остаётся без товара, а его данные попадают в руки мошенников.
2. **Сайты с предложениями о выигрышах или компенсациях**. Соблазнившись возможностью легко получить деньги, пользователь вводит свои персональные данные, включая данные банковской карты, что может привести как к краже средств, так и к краже личных данных.

Как видно, основная опасность фишинговых сайтов заключается не в их технической вредоносности, а в обмане пользователя, который добровольно передаёт свои данные или совершает платежи. Соответственно, мы можем предотвратить атаку даже после того, как пользователь перешёл по ссылке.

## Анализ целевой аудитории

Большинство пользователей интернета можно разделить на две группы – знающих о фишинговых сайтах и умеющих от них защищаться, и незнающих пользователей, которые крайне легко могут попасться на уловку мошенников. В моих интересах покрыть потребности обоих групп пользователей, соответственно наша программа будет рассчитана на массового пользователя. Для массового пользователя программа должна обладать несколькими качествами: интуитивно понятным интерфейсом, низкими требованиями к устройству, а также портативностью.

## Интеграция

Главной целью программы является защита пользователей от вредоносных сайтов в повседневной жизни. Программа позволит, не отрываясь от основных задач, быстро и легко опознать фишинговую ссылку. Основными преимуществами нашей программы по сравнению с другими будет несколько факторов: лёгкость в использовании, прозрачность работы программы, а также наличие инструментов под разные группы пользователей.

## Анализ соответствия нормативно-правовой базе

Для составления плана дальнейшего развития проекта нужно будет понимать, какие группы лиц больше всего пользуются нашей программой. Так как она будет предполагать обработку персональных данных в виде опроса пользователей при первом заходе в программу – нам нужно будет соблюсти требования Федерального закона «О персональных данных» от 27.07.2006 N 152-ФЗ.

Так как данные будут собираться для статистики, условия для сбора персональных данных должны соответствовать п. 8 ст. 6 Федерального закона «О персональных данных» от 27.07.2006 N 152-ФЗ, а именно сбор должен проходить с согласия пользователя, а данные должны быть обезличены. Также собранные данные не будут пересекаться со списком специальных данных, указанных в п. 8 ст. 6 Федерального закона «О персональных данных» от 27.07.2006 N 152-ФЗ.

## Выбор инструментов и методов

Этот этап можно считать по настоящему важным, так как от выбранных инструментов зависят многие характеристики конечного продукта. Мы разрабатываем десктопное приложение, с вектором на дальнейшую кроссплатформенность, вследствие чего основным инструментом на протяжении всей разработки будет фреймворк. Для удобного просмотра основных параметров фреймворка была сделана таблица.



Табл. 1. Сравнение аналогов

Самыми главными параметрами при выборе можно по праву считать поддерживаемые платформы и лицензию. Для начала следует разобрать вопрос лицензий. Все виды лицензий (кроме GPL) в таблице почти не накладывают ограничений на использование в коммерческих продуктах. Также, как можно видеть в таблице, Electron js и GTK не имеют поддержки мобильных устройств, что сразу делает их намного менее привлекательными для наших задач, в виду того, что в будущем программа будет портирована. В итоге нам стоит рассмотреть только 2 фреймворка – Flutter и Qt. По своим параметрам они очень похожи, но отличия возникают при их ближайшем рассмотрении: Flutter поддерживает только язык программирования Dart, в то время как на Qt можно писать почти на всех наиболее распространённых языках, что уменьшит время разработки. Также Flutter проигрывает Qt в аспектах веса конечной программы и потребления оперативной памяти. После выбора фреймворка было решено писать на Python, так как по сравнению с c++ с ним будет проще и быстрее работать.

После выбора языка перед нами появляется проблема в виде того, что библиотека PyQt для работы с Qt распространяется по лицензии GPL, что не позволит использовать её в коммерческих проектах. Но также существует и библиотека PySide, которая почти полностью идентична PyQt, но при этом распространяется по лицензии LGPL, соответственно использовать мы будем её.

Среди методов, изначально рассматриваемых для реализации в прототипе программы, можно выделить 3 основных:

1. Анализ сайта с помощью нейросетей по совокупности его характеристик. При таком методе рассматриваются абсолютно разные характеристики – от регистрационных данных сайта до характеристик его URL адреса. Проблема этого метода состоит в том, что он очень сложно алгоритмически реализуется, а также крайне сложно поддается стандартизации, вследствие чего данный вариант будет отложен на будущее.
2. Ручная проверка ссылки с использованием базы с характеристиками сайтов.
3. Использование автоматической проверки сайтов, на которые заходит пользователь с помощью DNS.

В качестве инструмента для ручной проверки сайтов был выбран API от VirusTotal, так как он предоставляет крайне обширный набор данных, а также имеет наиболее высокую репутацию среди себе подобных. Для автоматической проверки был использован Яндекс DNS.

## Оригинальность и сравнение с аналогами

Непрямыми аналогами программы можно считать сайты и приложения с функцией проверки ссылок. Не было найдено ни одного десктопного приложения с возможностью проверить ссылки или подключить DNS для защиты от них, из-за чего было решено сравнить функционал нашей программы с сайтами и мобильными приложениями. Для наглядности сравнения была также приведена таблица.



Табл. 2. Сравнение с аналогами

Несмотря на то, что функционал является решающим фактором в использовании того или иного инструмента, ценой нельзя пренебрегать, так как многие люди просто не готовы платить за антивирусы, что было также подтверждено в моём опросе.



Рис.П1.5. Пункт опроса про оплату антивируса

# ТЕХНОЛОГИЧЕСКИЙ ЭТАП

## Концепция будущего продукта и его характеристики

Программа должна будет содержать инструменты для защиты как для продвинутых, так и непродвинутых пользователей. Помимо этого, программа рассчитана на массового пользователя, вследствие чего она будет обладать интуитивно понятным интерфейсом и простотой использования.

## Технические характеристики будущего продукта

Для того, чтобы программу можно было считать легкой, она должна иметь низкое потребление оперативной памяти и быть легковесной. Для достижения этой цели будут поставлены требования в виде веса меньше 100мб и потребления оперативной памяти меньше 100мб. Ввиду этого по мере разработки необходимо выбирать инструменты, обладающие минимальным влиянием на эти два параметра. Для начальной реализации в программе было выбрано 2 метода: Ручная проверка ссылки с использованием базы с характеристиками сайтов и автоматическая защита при помощи DNS. Ручная проверка будет брать ссылку из текстового окна для ввода и впоследствии выводить анализ сайта в окнах с информацией. Автоматическая защита будет реализована через подключение устройства пользователя к DNS посредством оптимальных инструментов. Проверка успешности реализации ручной проверки будет проведена через проведение тестов с набором ссылок, обладающих разными свойствами. Для подтверждения работоспособности автоматической защиты будет достаточно проверить эту функцию на нескольких устройствах.

## Процесс разработки программы (ручная проверка)

Так как основным инструментом работы будет являться фреймворк Qt – изначально нужно было выбрать подход к разработке. Немалый процент разработчиков на этом фреймворке выбирает не использовать Qt Designer – среду для графической разработки интерфейсов, что я считаю крайне неоптимальным вариантом.

Сначала было решено сделать набросок интерфейса без дизайна, чтобы разобраться с Qt Designer и примерно понять сложность реализации тех или иных графических функций. Сперва была сделана страница с ручной проверкой сайта.

# 

Рис. П2.1. Начальный интерфейс программы без стилизации

Стилизация была выполнена с помощью языка (англ. Cascading Style Sheets – “Каскадные Таблицы Стилей”). Был выбран минималистичный стиль для максимального удобства пользователя при работе с программой.



Рис. П2.2. Начальный интерфейс программы со стилизацией

Следующим шагом было написание кода для программы. Для начала была сделана проверка ссылки на валидность во избежание ошибок. Далее была реализована проверка ссылки через библиотеку VirusTotal и получение её параметров в объект. Но ВирусТотал передает анализ сайта в основном в виде цифр, например анализ сайта происходит большим количеством разных антивирусов, вследствие чего итог анализа будет выглядеть как словарь с “голосами” разных антивирусов за степень опасности. Данный вариант может подойти для более продвинутых пользователей, но обычного человека такой вариант представления информации введёт в ступор. Соответственно оптимальным вариантом будет сделать для приложения два варианта вывода информации: упрощенный, который будет содержать только самую важную информацию, а также представлять её в упрощенном виде, и расширенный режим, при котором будет выводиться максимально большое количество полезной информации, и, кроме того, результат будет представлен в изначальном виде.

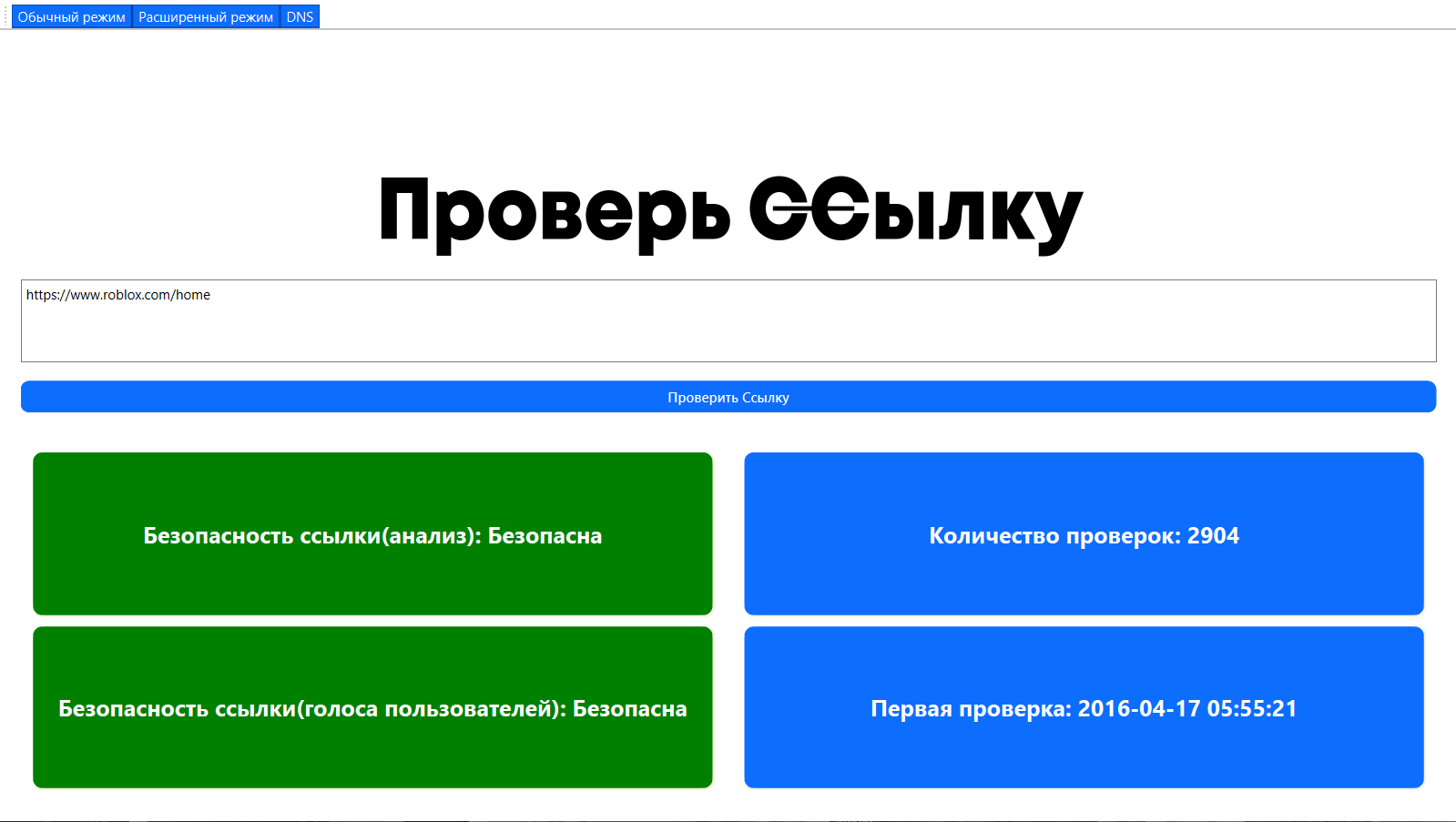


Рис. П2.3. Упрощённая проверка сайта

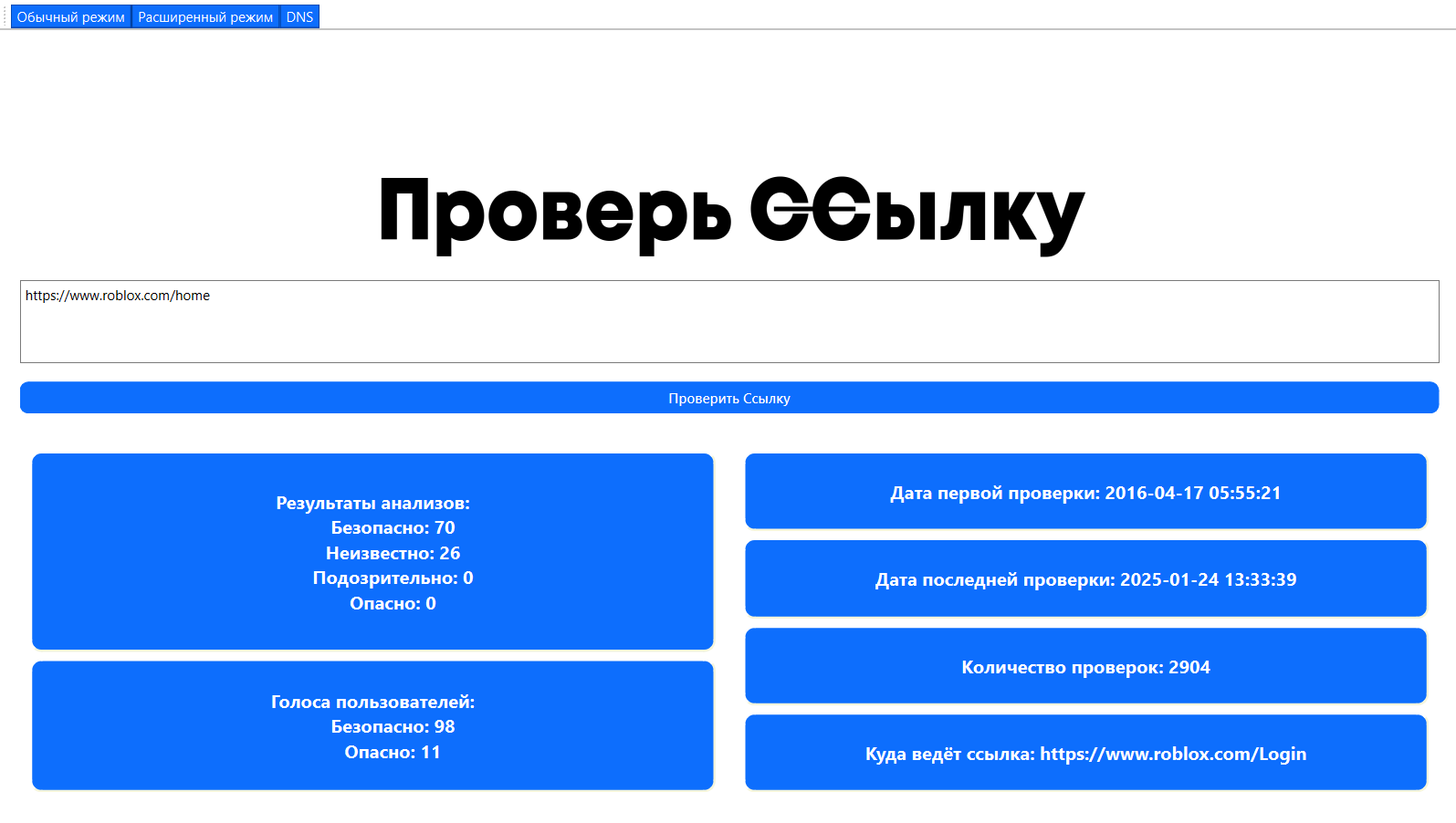


Рис. П2.4. Расширенная проверка сайта

На данный момент алгоритм работы можно кратко описать так: Пользователь нажимает на кнопку проверки, после этого программа берёт ссылку из текстового окна для ввода и отправляет запрос на VirusTotal API, а при получении “раскладывает” полученные данные по окнам с информацией.

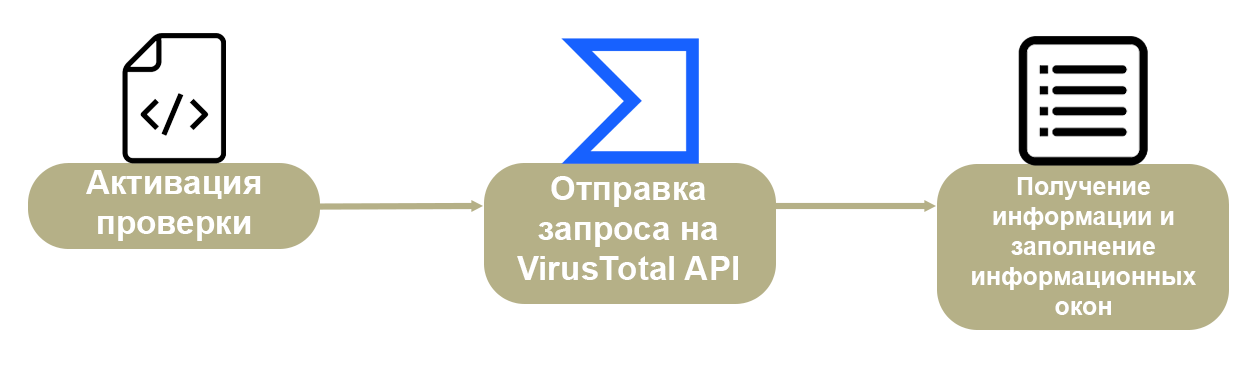


Рис. П3.1. Схема стандартной работы программы

## Процесс разработки программы (автоматическая защита)

Вторым этапом разработки программы будет являться добавление автоматической защиты при помощи DNS. Для начала следует кратко объяснить, что делает DNS сервер. DNS сервер – это сервер, на котором хранятся данные IP адресов сайтов. Пользователь отправляет на сервер ссылку на сайт, а взамен получает IP адрес сайта. Существуют DNS сервера с поддержкой защиты от вредоносных сайтов, одним из таких является Яндекс DNS, который и планируется использовать.

На стороне нашей программы останется только подключение к DNS устройства пользователя. Есть несколько методов, но было решено выбрать тот, который меньше всего влияет на характеристики программы, и им является подключение DNS проведением команд через командную строку при помощи библиотеки subprocess на python. Главным плюсом этого решения является то, что библиотека subprocess является встроенной, т. е. не будет влиять на вес конечной программы в отличие от сторонних библиотек.

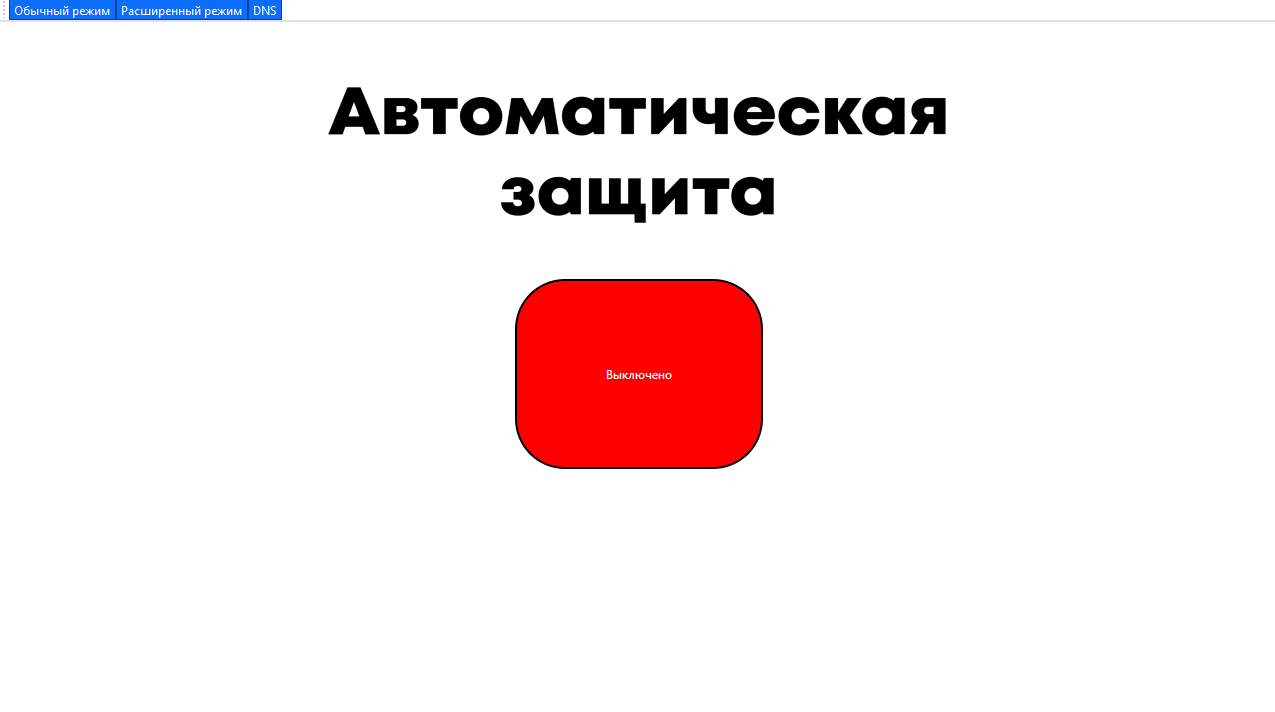


Рис. П2.5. Раздел с подключением DNS

## Дальнейшая оптимизация алгоритма работы

С первого взгляда работа нашей программы ничем не ограничена, но проблема кроется в лимитах на использование функционала ВирусТотала в рамках его бесплатного API, так как с одного аккаунта можно делать только 4 запроса в минуту и 500 запросов в неделю. У этой проблемы есть несколько решений:

1. Создание большого количества аккаунтов ВирусТотала – решение скорее временное, чем постоянное. Это позволит использовать API в количестве, пропорционально соответствующем количеству аккаунтов (5 аккаунтов – 20 запросов в минуту и 2500 запросов в день, 25 аккаунтов – 100 запросов в минуту и 12500 запросов в день и т.д.). Данный метод будет работать через перебор рабочих ключей при запросе.

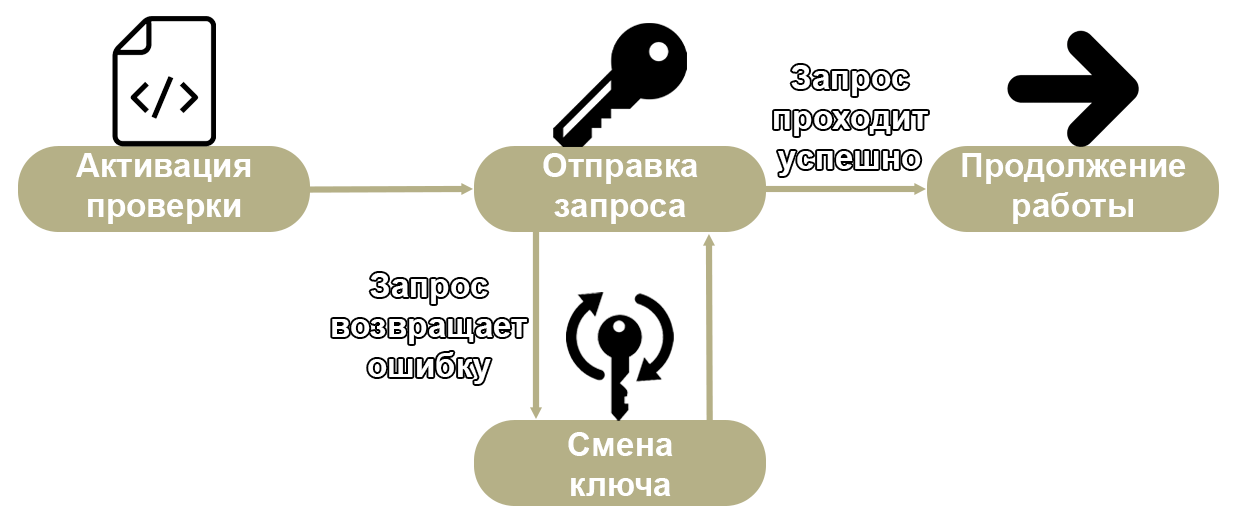


Рис. П3.2. Схема работы с базой данных

1. Реализация базы данных. Данный вариант является самым актуальным, так как многие люди проверяют одни и те же сайты, вследствие чего крайне большое количество запросов тратится впустую. База данных поможет избежать данной проблемы, так как информация по сайту будет искаться сначала в базе данных, а уже потом, при её отсутствии – будет проводиться запрос ВирусТоталу с последующим дополнением базы данных.

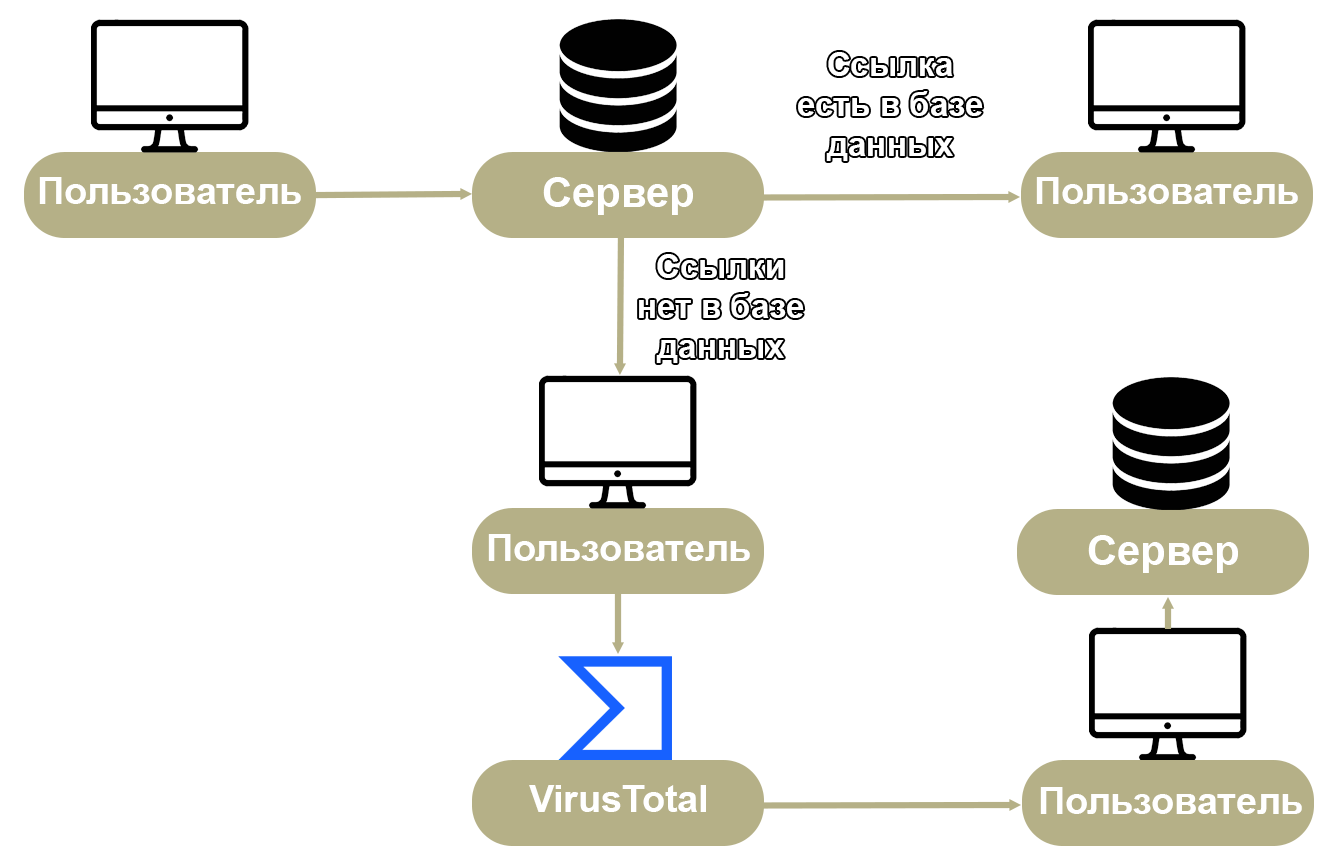


Рис. П3.3. Схема работы с базой данных

## 3.6 Проведение тестов

Для проверки работоспособности и устойчивости программы в разных ситуациях необходимо создать набор тестов. Скриншоты к тестам будут приложены в папке с проектом



Табл. 3. Тесты ручной проверки сайтов

Также, для того чтобы наши задачи к конечному продукту считались успешно выполненными – были проведены замеры потребления оперативной памяти и веса программы. Программа потребляет 43мб оперативной памяти и занимает 45мб на диске, т.е. наши задачи успешно выполнены.



Рис. П2.6. Потребление оперативной памяти



Рис. П2.7. вес программы

# ЗАКЛЮЧЕНИЕ

## 4.1 Дальнейшее развитие проекта

У программы есть крайне большое количество путей развития, но, по моему мнению, есть всего несколько вещей, на которых стоит сосредоточить основное внимание:

1. Оптимизация уже существующего инструментария. Хоть на данный момент программа и работает успешно, но ей требуется немалое количество доработок чтобы стать абсолютно самостоятельным инструментом.
2. Внедрение искусственного интеллекта для проверки сайтов. Это позволит добавить дополнительный слой проверки, а также крайне сильно отличит программу от возможных конкурентов.

## 4.2 Демоверсия программы

Для наглядной демонстрации программы была собрана в один exe файл и загружена на гитхаб. Попробовать программу можно по ссылке <https://github.com/Sayge1/Infobez-project> или QR коду:



## 4.3 Итог

Для проверки интерфейса программы на соответствие заявленным целям был проведён опрос среди учащихся 10 классов МБОУ Гимназии №1. По результатам опроса из 26 опрошенных более 88% считают интерфейс удобным и интуитивно понятным, обладает малым весом в 50мб, а по результатам тестов можно понять, что программа полностью работоспособна и устойчива, вследствие чего поставленные цели и задачи можно считать успешно выполненными.

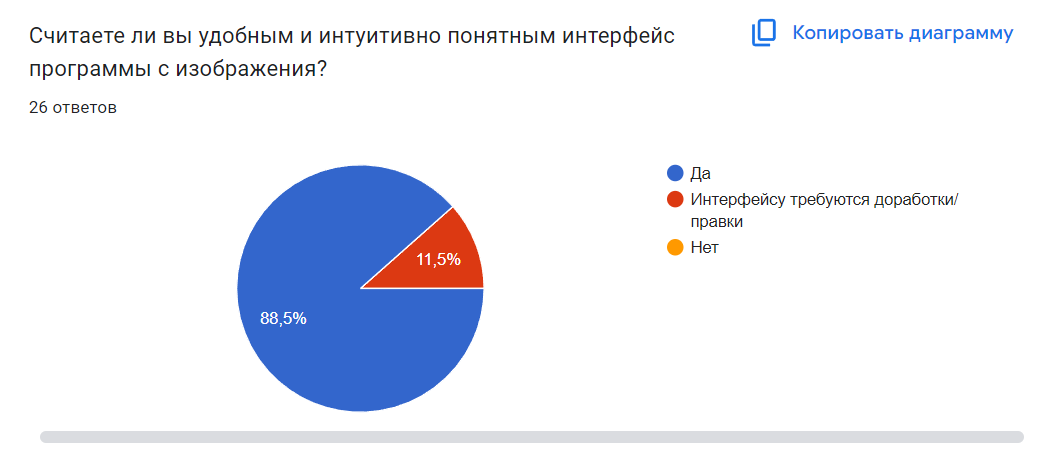


Рис.П1.6. Пункт опроса с оценкой программы

# Список использованной литературы

1. Отчёт “Лаборатории Касперского” по спаму и фишингу за 2023 год [Электронный ресурс]. URL: <https://securelist.ru/spam-phishing-report-2023/109104/>
2. APWG | Phishing Activity Trends Reports [Электронный ресурс]. URL: https://apwg.org/trendsreports/
3. Анализ угроз информационной безопасности при использовании фишинговых сайтов [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/analiz-ugroz-informatsionnoy-bezopasnosti-pri-ispolzovanii-fishingovyh-saytov/viewer>
4. Исследование техник фишинга и методов защиты от него [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/issledovanie-tehnik-fishinga-i-metodov-zaschity-ot-nego/viewer>
5. Федеральный закон "О персональных данных" от 27.07.2006 N 152-ФЗ // СПС КонсультантПлюс
6. Классификация фишинговых атак и меры противодействия им [Электронный ресурс]. URL: <https://cyberleninka.ru/article/n/klassifikatsiya-fishingovyh-atak-i-mery-protivodeystviya-im>
7. Obligations of the GPL and LGPL [Электронный ресурс]. URL: https://www.qt.io/licensing/open-source-lgpl-obligations

# Приложение №1. Графики и диаграммы

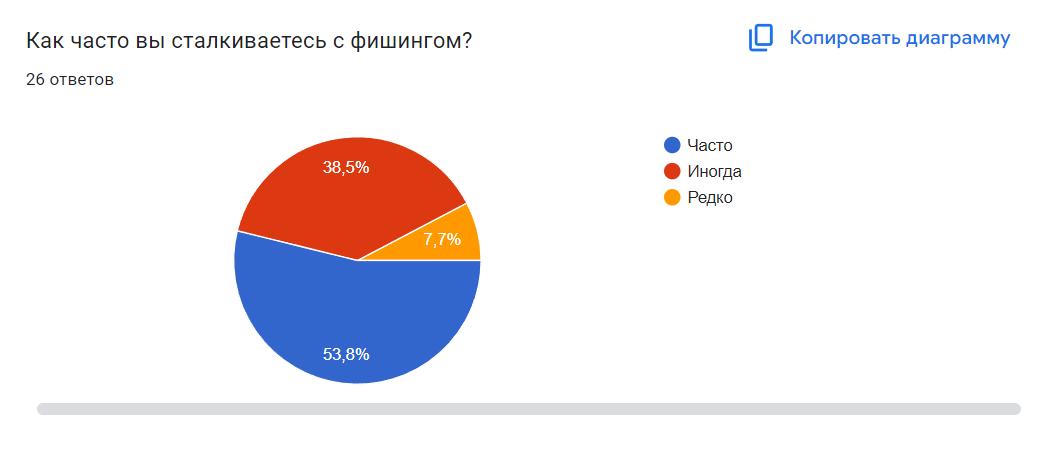


Рис. П1.1 Пункт опроса про фишинг

Рис. П1.2. Доля спама в почтовом трафике российского интернета

Рис. П1.3. Количество срабатываний почтового антивируса на попытки открыть вредоносный файл

Рис. П1.4. Количество срабатываний системы “Антифишинг” на попытки перехода по мошеннической ссылке



Рис.П1.5. Пункт опроса про оплату антивируса

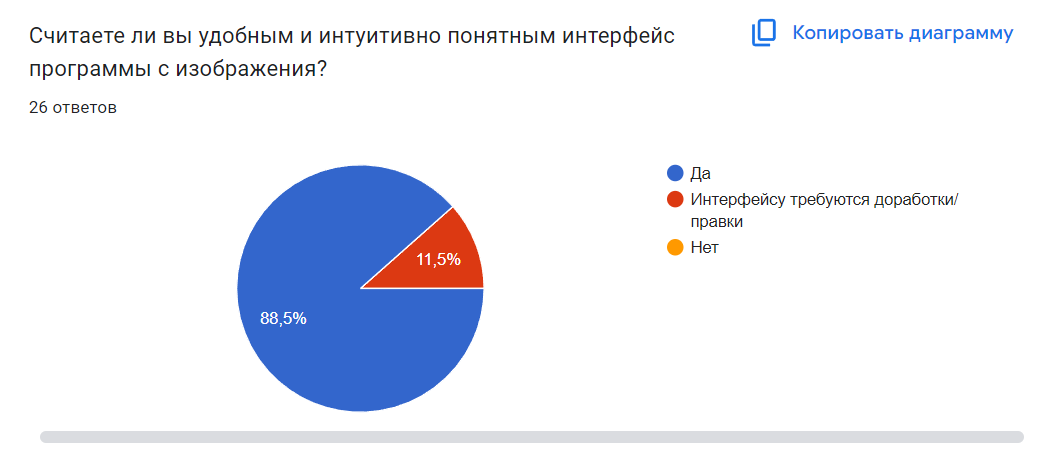


Рис.П1.6. Пункт опроса с оценкой программы

# Приложение №2. Программа

# 

Рис. П2.1. Начальный интерфейс программы без стилизации



Рис. П2.2. Начальный интерфейс программы со стилизацией

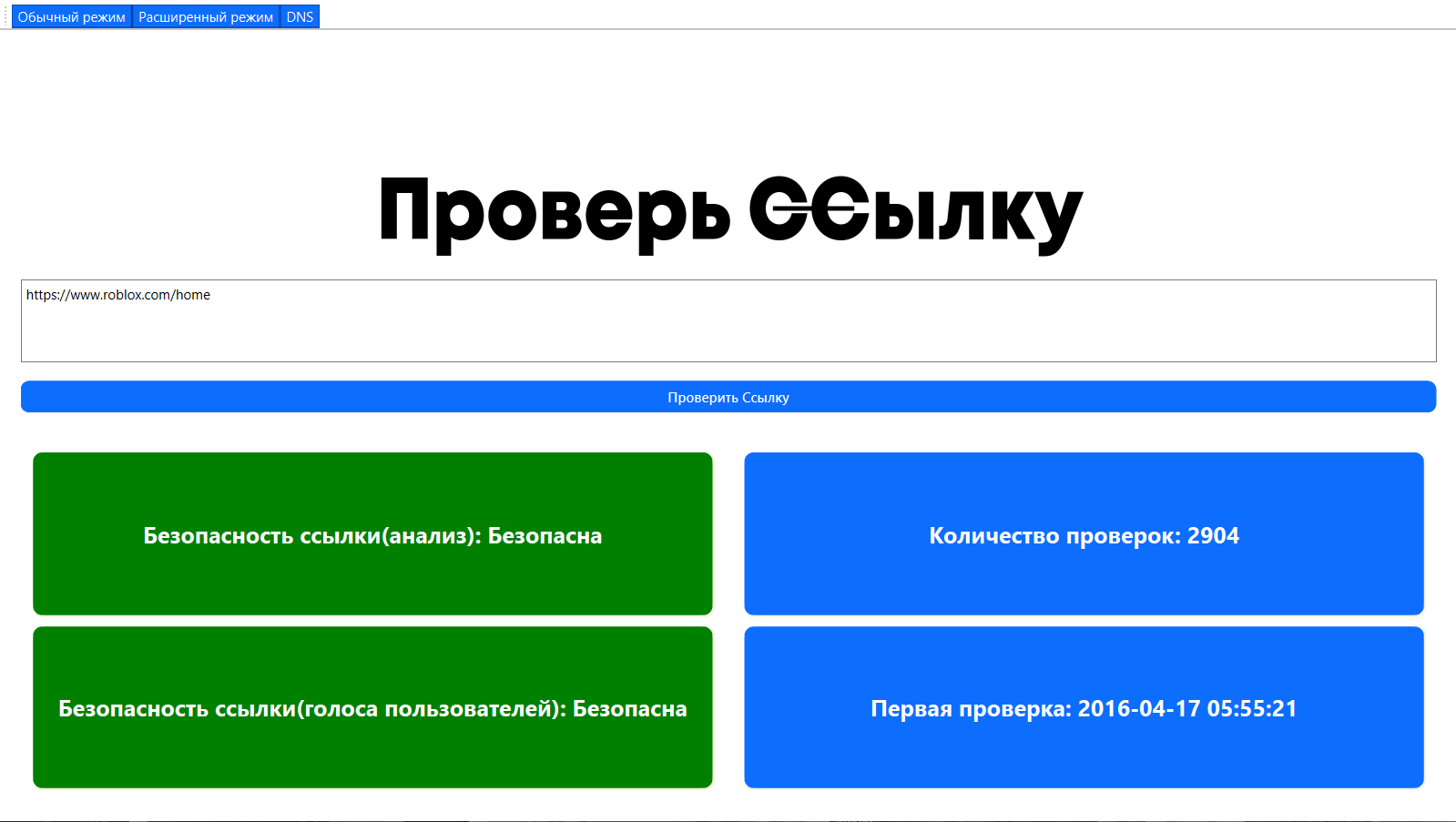


Рис. П2.3. Упрощённая проверка сайта

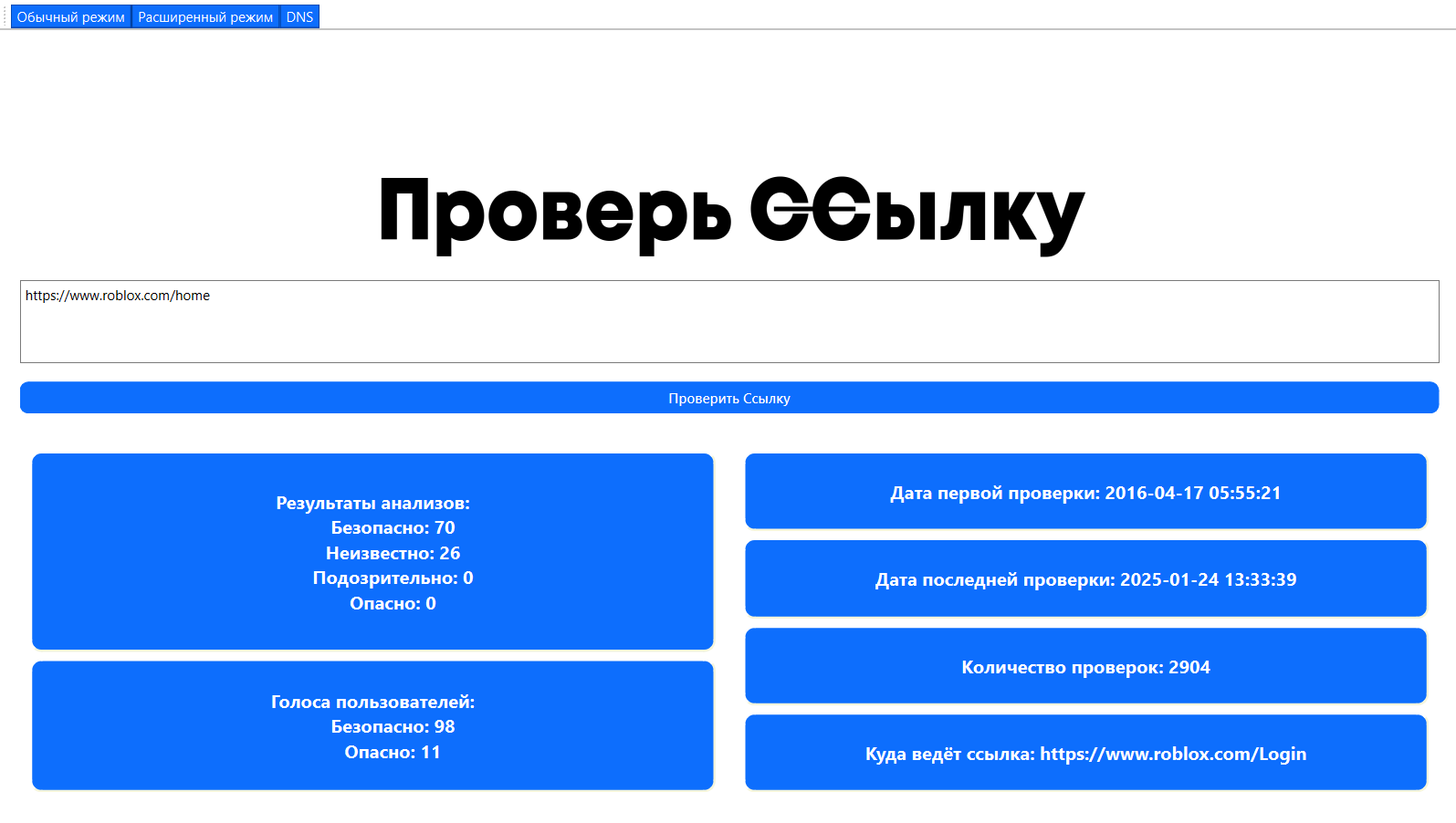


Рис. П2.4. Расширенная проверка сайта

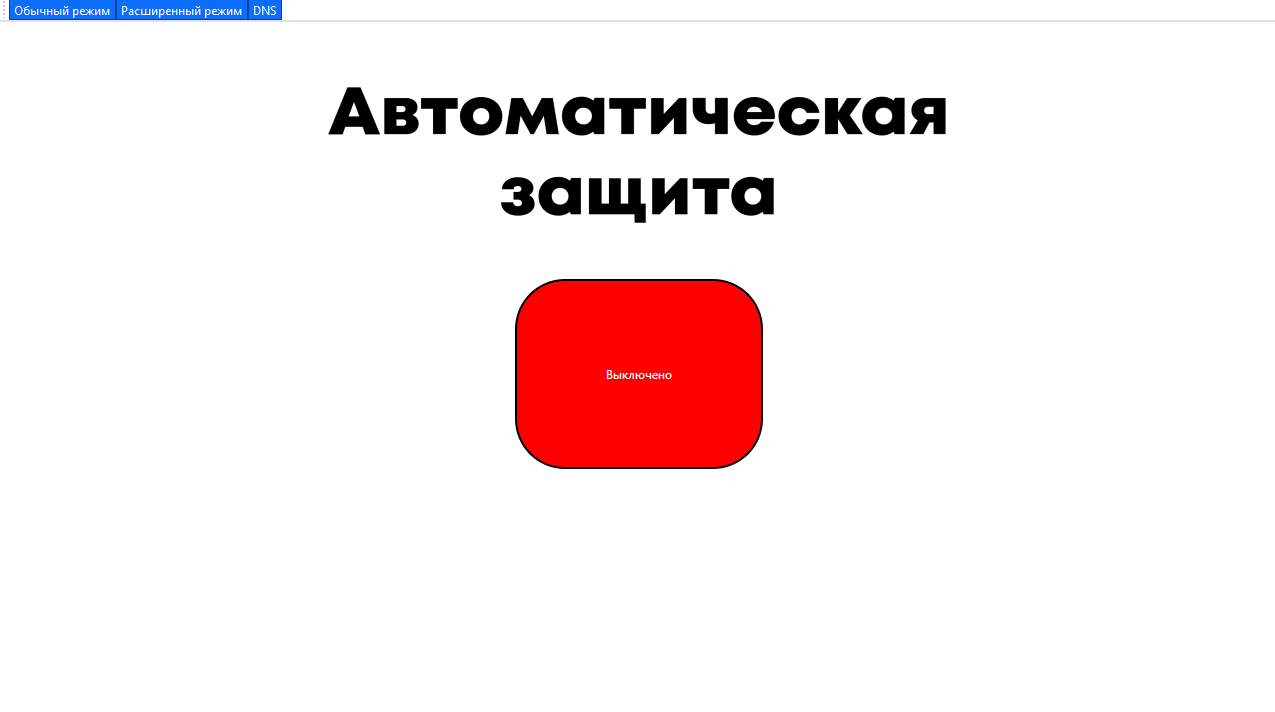


Рис. П2.5. Раздел с подключением DNS



Рис. П2.6. Потребление оперативной памяти



Рис. П2.7. вес программы

# Приложение №3. Схемы и иллюстрации

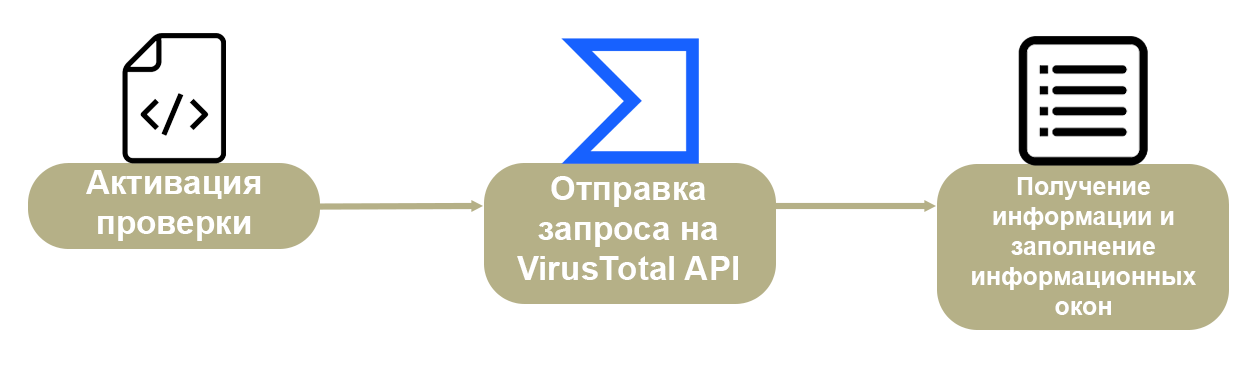


Рис. П3.1. Схема стандартной работы программы

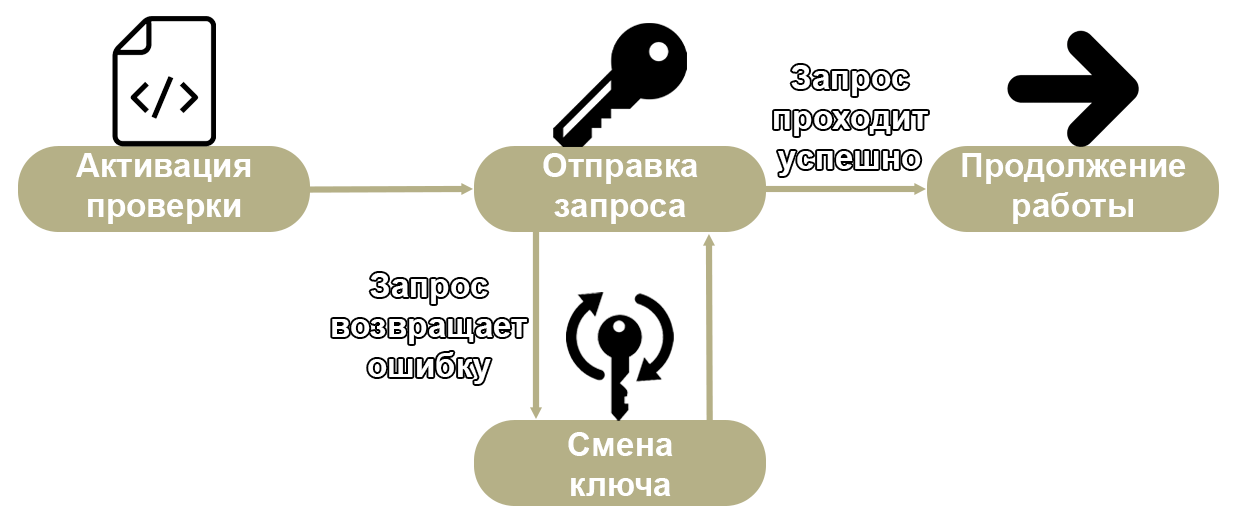


Рис. П3.2. Схема работы с базой данных

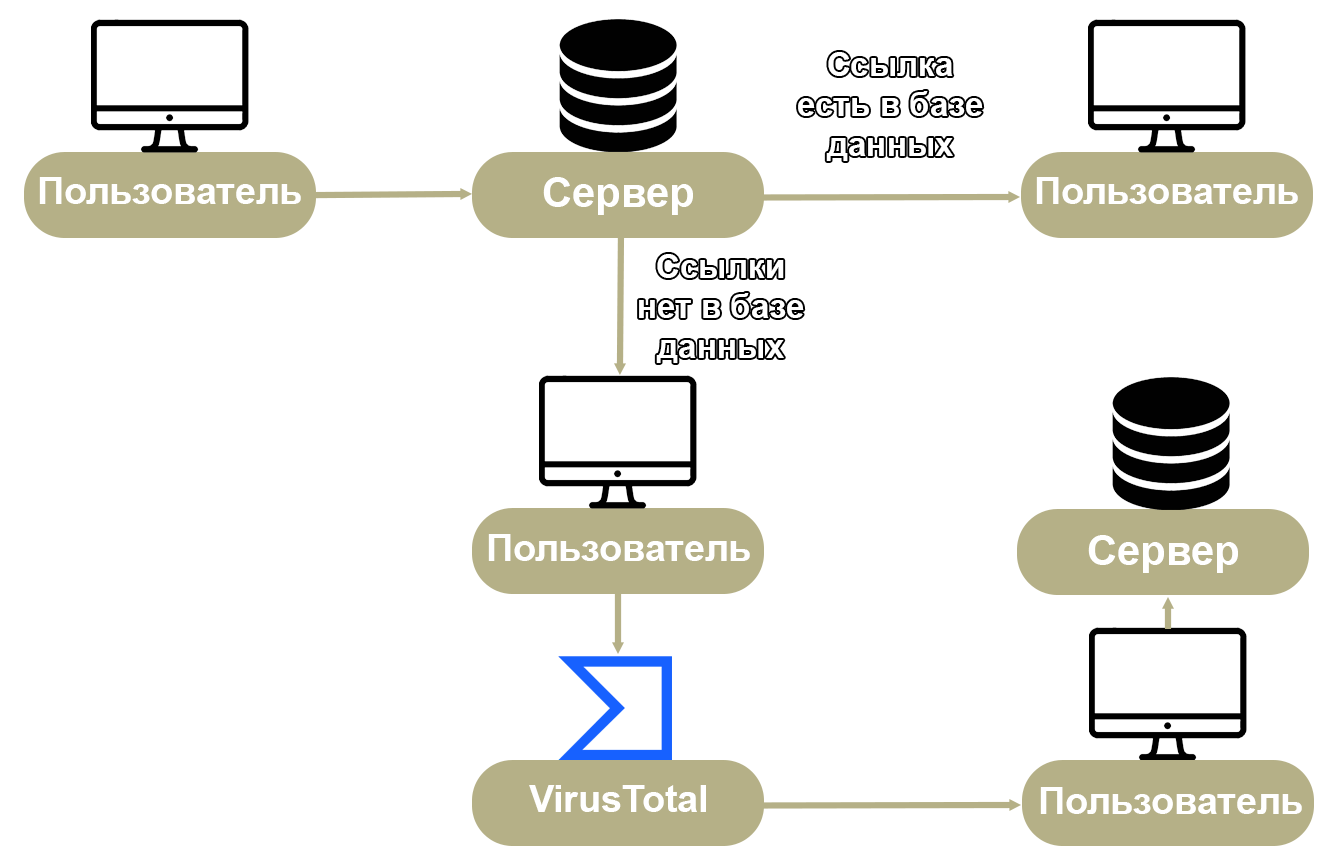


Рис. П3.3. Схема работы с базой данных

# Приложение №4. Таблицы



Табл. 1. Сравнение аналогов



Табл. 2. Сравнение с аналогами



Табл. 3. Тесты ручной проверки сайтов