Лицей НИУ ВШЭ

Разработка QR-код сканера «Secure QR scanner» с открытым кодом с функцией автопроверкой URL-адресов на факт фишинга

Алешин

Арсений Александрович

Москва 2025

СОДЕРЖАНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Фишинг — это вид мошенничества (часто с использованием электронных писем, текстовых сообщений или телефонных звонков), в ходе которого

злоумышленник обманом вынуждает жертву раскрыть учетные данные, банковские реквизиты или другую персональную информацию, а затем использует их в преступных целях.

Асинхронное программирование — концепция программирования, при которой результат выполнения функции доступен не сразу, а через некоторое время в виде асинхронного (нарушающего стандартный порядок выполнения) вызова. Используется для повышения производительности работы программы.

Профилирование — это процесс анализа работы программы с целью измерения ее производительности и оптимизации использования ресурсов устройства (диск, процессор, оперативная память).

API ключ (API key) — это уникальный идентификатор, который используется для подключения к API и взаимодействия с ним.

Фреймворк - программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

API (Application Programming Interface) - используется для взаимодействия программ или сервисов друг с другом. Запрос исходит от клиента, ответ посылает сервер. Например, сервер-метеослужба отправляет данные через API погодным приложениям, которые показывают их на телефоне.

ВВЕДЕНИЕ

QR-коды стали неотъемлемой частью нашей повседневной жизни, они очень активно используются в рекламе и маркетинге, так как за несколько секунд каждый пользователь со сканером сможет зайти на нужную страницу. Однако QR-коды могут быть небезопасны. Любой человек может быстро и просто создать QR-код с любой ссылкой. Таким образом мошенники могут создавать «небезопасные» QR-коды, которые будут перенаправлять пользователей на вредоносные или фишинговые веб-сайты, нацеленные на кражу персональных или иных данных.

1 ИССЛЕДОВАНИЕ ПРОБЛЕМЫ

1.1 ОПИСАНИЕ ПРОБЛЕМЫ

Злоумышленники могут создавать QR-коды, которые ведут на вредоносные или фишинговые сайты. Зачастую невозможно определить безопасен ли сайт исключительно по доменному имени. Большинство приложений для сканирования QR-кодов после декодирования показывают пользователю только ссылку на сайт и не выполняют никакой проверки на безопасность и никак не предупреждают пользователя.

1.2 АКТУАЛЬНОСТЬ

Сейчас проблема со взломами пользователей с использованием QR-кодов является актуальной. Данная тема поднимается в следующих статьях:

1. «Самые популярные схемы мошенничества с QR-кодами» за 2024 год, написанная Максимом Игнатовым (редактор блога компании Эльдорадо) [1];
2. «Россиян предупредили о взломах Telegram при помощи домовых чатов. Telegram могут взломать через QR-код домового чата» за 2025 год, написанная Романом Кильдюшкиным (заместитель редактора отдела «Технологии» издание Газета.ru) [2]

3. «Хакеры стали использовать QR-коды в мошеннических схемах» за 2024 год, написанная Юлией Гуреевой (редактор журнала RG.ru) [3]

1.3 АНАЛИЗ АНАЛОГОВ

1. Android приложения «QR-сканер — безопасный» от компании Trend Micro. Приложение выполняет проверку на безопасность ссылки, полученной после сканирования QR-кода, однако в этом QR-код сканере есть следующие недостатки:

    1. Приложение не показывает информацию о сайте (время создания, владелец и т.п). У пользователя нет возможности самостоятельно проанализировать информацию о сайте (возможная защита от ложных срабатываний);

    2. Нет возможности поменять внешний вид приложения (изменить оформление приложения) или язык (проблема удобства использования приложения);

    3. Нет возможности создать свой QR-код, что очень неудобно и подразумевает установку дополнительного приложения (проблема удобства использования приложения).

2. Приложение «Лаборатория Касперского» от компании Kaspersky. Приложение также предоставляет возможность выполнить проверку на безопасность ссылки, полученной после сканирование QR-кода.

    Недостатки приложения «Лаборатория Касперского»:

    1. Избыточный функционал, ненужный некоторым пользователям (В приложении есть много других функций, не связанных с проверкой QR-кодов, что может быть излишним для некоторых пользователей, например для людей, использующих другие антивирусные приложения для Android)

2. Понижение скорости работы телефона или скорости интернета на телефоне (на не очень мощных телефонах приложение от Kaspersky может вызывать понижение скорости работы своим антивирусным ПО, встроенным в приложение) [10]

3. Нагрузка на батарею телефона (приложение от Kaspersky может потреблять значительное количество энергии телефона и сокращать время его автономной работы) [9]

4. Невозможность создать свой QR-код в приложении (в приложении «Лаборатория Касперского» нет возможности создать свой QR-код, что может быть неудобно некоторым пользователям) (проблема удобства использования приложения)

2 РЕШЕНИЕ ПРОБЛЕМЫ

2.1 ЦЕЛИ И ЗАДАЧИ

Цель:

Создать Android-приложение для сканирования QR-кодов с проверкой веб-сайтов на безопасность использования (Проверка на фишинг или вредоносный код) и создания своих QR-кодов.

Задачи:

1. Изучить инструменты разработки Android-приложений
2. Выбрать язык программирования, среду разработки, необходимые библиотеки, API сервисы, подходящие для реализации проекта
3. Продумать и разработать функционал приложения
    1. Создать приложение с базовыми функциями сканирования и создания QR-кодов, хранением истории;
    2. Добавить функцию проверки QR-кодов на безопасность.
4. Разработать код для удаленного сервера

1. Рассмотреть и выбрать необходимые для разработки логики сервера инструменты (язык программирования, библиотеки и т.п);
2. Разработать функционал для обращения к API сервисам;
3. Разработать функционал для обработки входящих из приложения запросов.

5. Разработать пользовательский интерфейс приложения
    1. Рассмотреть аналоги и проанализировать достоинства и недостатки их пользовательского интерфейса;
    2. На основе полученной информации сформировать удобный пользовательский интерфейс конечного продукта.
6. Составить пользовательское руководство, документацию проекта и пользовательское соглашение

2.2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Требования к разрабатываемому решению:

1. Удобный и понятный пользовательский интерфейс
    1. Интерфейс приложение не перегружен;
    2. Интерфейс приложения интуитивно понятен и удобен в использовании.
2. Точность обнаружения фишинговых или вредоносных сайтов
    1. Приложение в 70–80% случаев корректно выявляет вредоносные/небезопасные ссылки;
    2. Приложение предоставляет пользователю актуальную и достоверную информацию о проверяемом сайте.
3. Возможность настроить приложение (светлая и темная темы, смена языка и т. п.)
    1. Приложение предоставляет возможность изменить оформление/тему приложения;

2. Приложение позволяет изменить язык интерфейса.

2.3 ВЫБОР ТЕХНОЛОГИЙ

Среда разработки: Android Studio

1. Имеет встроенный эмулятор для тестирования приложений;
2. Обладает обширной библиотекой шаблонов (позволит ускорить процесс разработки);
3. Имеет встроенный редактор интерфейса, позволяющий быстро и просто создать пользовательский интерфейс;
4. Имеет встроенные удобные инструменты для отладки и профилирования.

Язык программирования для разработки Android-приложения: Kotlin

1. Понятный и лаконичный синтаксис;
2. Официально поддерживается в Android Studio и является основным языком разработки Android-приложений на ряду с Java;
3. Поддерживает асинхронное программирование;
4. Является официальным языком для разработки приложений на Android, полная совместим с Java.

Фреймворк для разработки логики удаленного сервера: Flask

1. Гибок в работе, позволяет работать с множеством сторонних библиотек;
2. Предоставляет поддержку асинхронности;
3. Легок в масштабировании, можно комбинировать с другими фреймворками и приложениями для веб-разработки.

Основная библиотека для работы с QR-кодами: Zxing [4]

1. Поддерживает QR-коды различных форматов
2. Позволяет создавать QR-коды
3. Дает возможность настраивать QR-коды (цвет, иконка)
4. Высокая скорость работы

API для проверки безопасности сайта: AbuseIPDB API [5], Yandex Safe Browsing API [6], Lookup API [7]

1. Позволяют быстро получить вердикт по безопасности сайта
2. Надежные и стабильные в работе
3. Имеют бесплатные решения

Решение использовать несколько API для проверки ссылок было сделано, чтобы обеспечить точность и надежность проверки ссылок, так как собственные алгоритмы проверки могут работать некорректно или ненадежно. В дополнении, использование нескольких API сервисов позволит уменьшить количество ложных срабатываний при сканировании.

2.4 ПРИНЦИП РАБОТЫ

Анализ ссылки из QR кода (взаимодействие с API):

1. Пользователь сканирует QR-код с помощью приложения (для этого используется библиотека Zxing);
2. С полученной ссылкой делается запрос к удаленному серверу на Flask;
3. Сервер отправляет запрос к API с переданной ссылкой и возвращает ответ;
4. Приложение обрабатывает полученный ответ и возвращает пользователю статистику сканирования, информируя о том, является сайт опасным или нет.

Для обращения к API необходимо использовать специальный ключ (API key). Этот ключ указывается в URL запроса, поэтому обращаться напрямую с телефона к серверу небезопасно, так как злоумышленник может перехватить URL с ключом. Для избежания перехвата данных используется удаленный сервер, совершающий запрос к API с отправленной ссылкой. Информация о подобной проблема указываются в следующих статьях:

1. «Mobile Apps are leaking your API Keys: Discover how to prevent it» [], опубликованная компанией Build38, занимающейся информационной безопасностью мобильных приложений [10].

2. «API attacks: Understanding and Protecting your Infrastructure», опубликованная компанией CybelAngel, занимающейся информационной безопасностью [11].

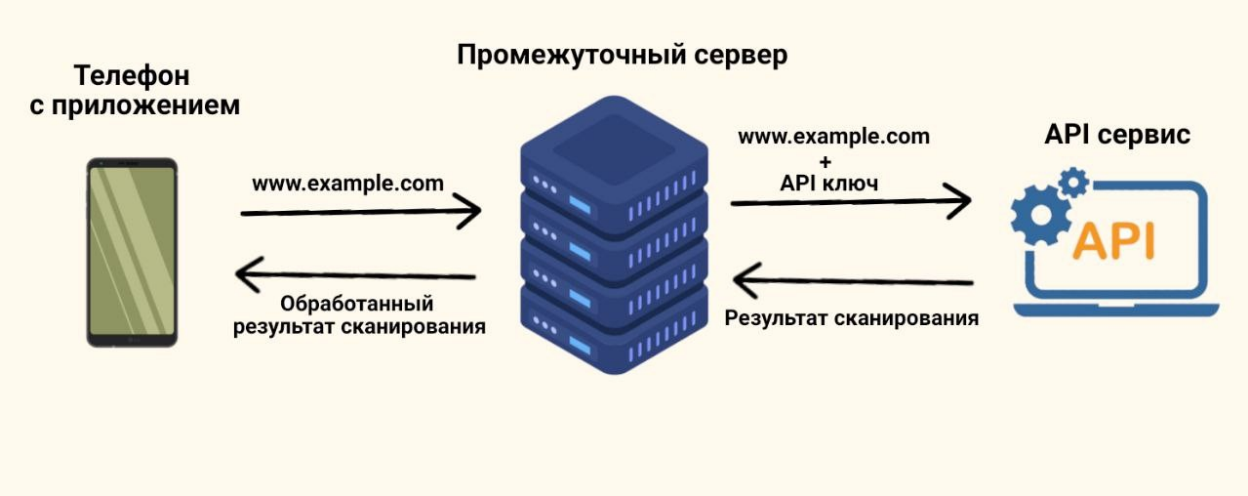

Рисунок 1, Взаимодействие с сторонними API

Генерация QR-кода:

1. С помощью библиотеки Zxing создается объект класса QrCodeWrite, который позволяет создать битовую матрицу QR-кода;

2. Пиксели изображения заполняются на определенному правилу: 1 - закрашенный пиксель, 0 - незакрашенный пиксель;

3. Элементу интерфейса приложения, отвечающему за демонстрацию QR-кода, с помощью встроенной функции setImageBitmap

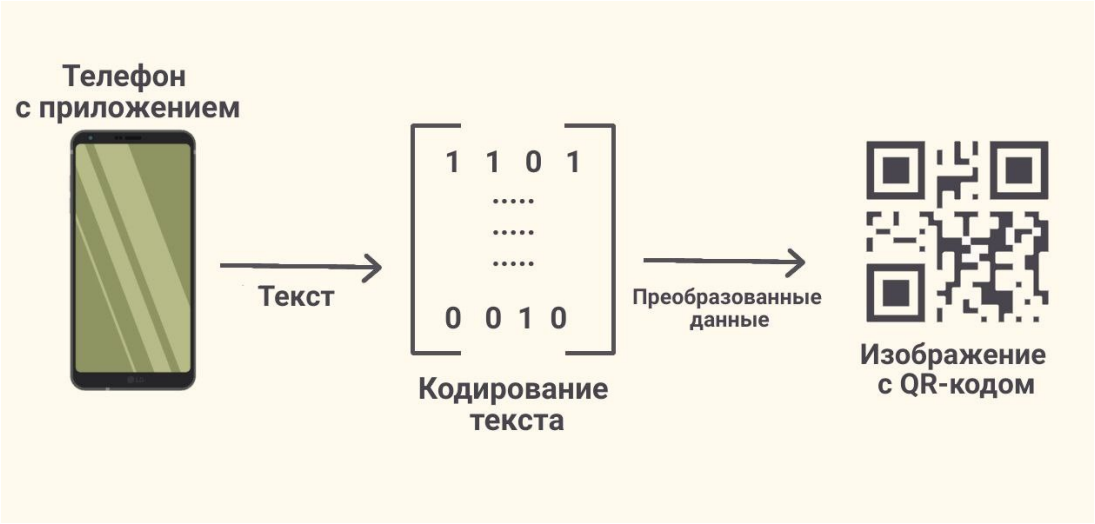

Рисунок 2: Генерация QR-кода

2.5 ТЕСТИРОВАНИЕ И ОТЛАДКА

В процессе тестирования были выявлены и устранены ошибки работы приложения. Код приложения был доработан и оптимизирован.

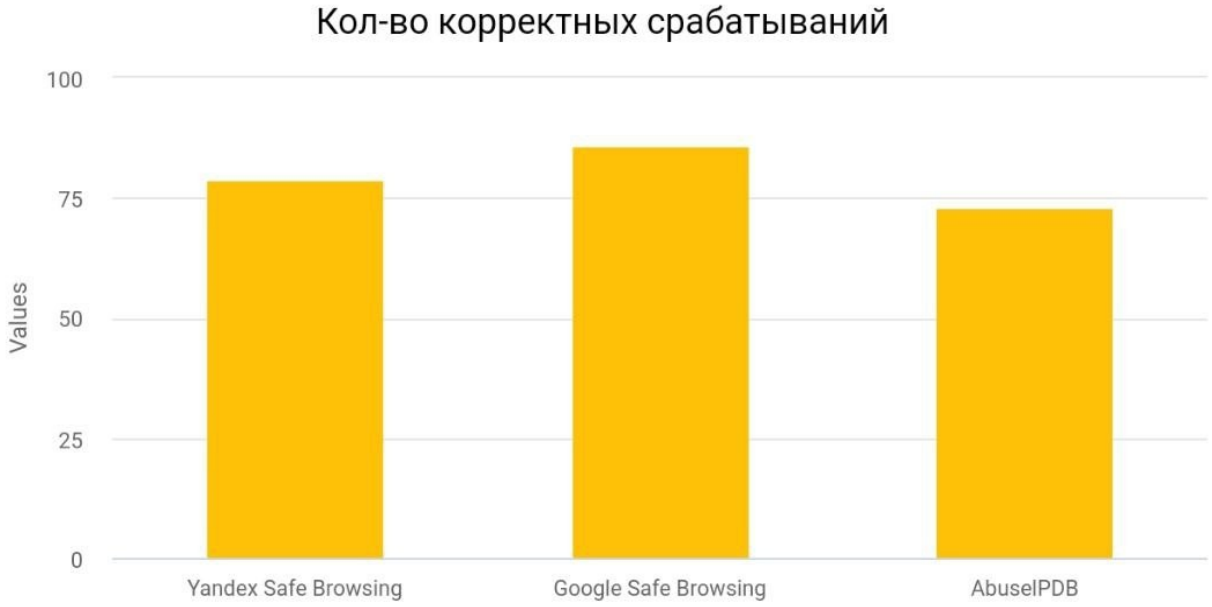

Рисунок 3: Статистика срабатываний API

2.6 СОПРОВОДИТЕЛЬНАЯ ДОКУМЕНТАЦИЯ

Для поддержания работоспособности приложения и дальнейшей разработки и развития проекта была разработана документация разработчика.

Для обеспечения удобного и комфортного использования приложения пользователями было разработано руководство пользователя, дающее полную информацию об использовании приложения

3 ОПИСАНИЕ ИТОГОВОГО РЕШЕНИЯ

Итоговое решение — приложение на Android, предоставляющее возможности сканирования, анализа и создания QR-кодов.

Основной экран приложения:

При входе в приложение пользователю предлагается выбрать одну их четырех опций:

1. «Создать»/ «Generate» – открыть меню генерации QR-кода
2. «Сканировать»/ «Scan» – начать новое сканирование
3. «История»/ «History» – просмотреть историю сканирований
4. «Настройки»/ «Settings» - настроить внешний вид приложения, просмотреть руководство пользователя

Меню сканирования:

Для начала сканирования пользователю необходимо нажать на кнопку с меню «Scan»/ «Сканировать». После нажатия у пользователя автоматически откроется камера с полем для сканирования QR-кода.

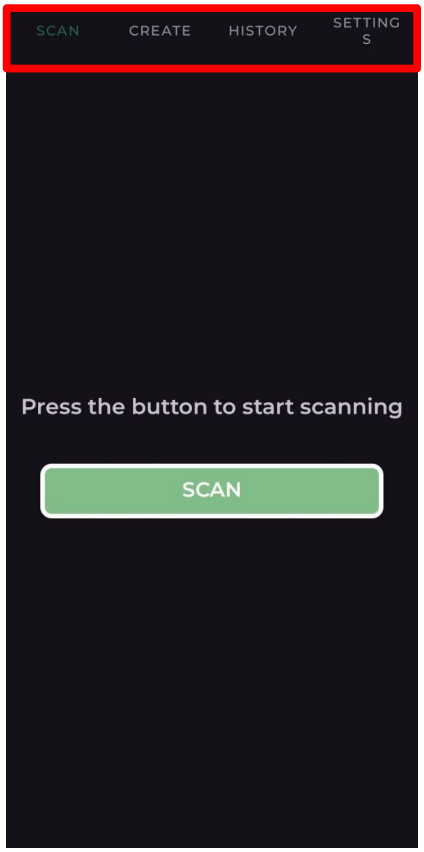

Рисунок 4: Главный экран приложения

Меню генерации QR-код:

Для генерации QR-кода пользователю необходимо вставить необходимый текст в поле ввода и нажать кнопку «Создать QR-код»/«Generate QR-Code». После этого готовый код появится над полем ввода.

Пользователь, также, может настроить «Цвет Фона»/ «Background Color» и «Цвет узора»/ «Pattern Color».

Для того, чтобы сохранить QR-код в память телефона необходимо нажать на кнопку «Сохранить»/ «Save».

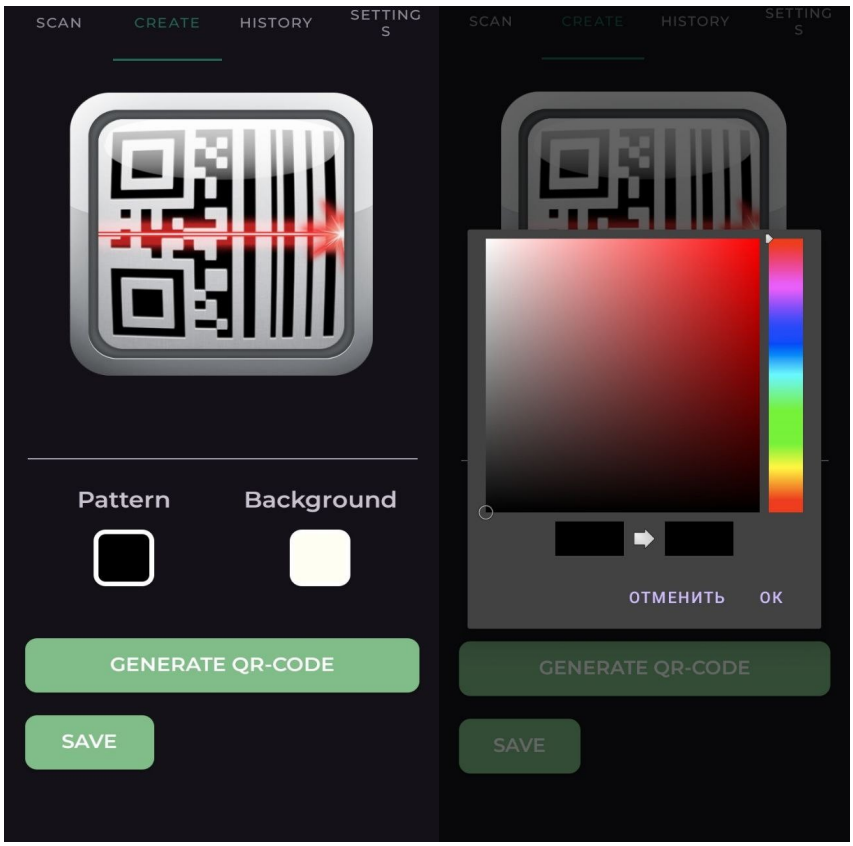

Рисунок 5: Меню создания QR-кода

Рисунок 6, Меню настройки цвета QR-кода

Меню обработки декодированной информации:

После сканирования QR-кода приложение перенаправляет пользователя в меню обработки декодированной информации. В поле «Текст»/«Text» отображается декодированный текст, который можно скопировать с помощью кнопки «Копировать»/ «Copy» или открыть в браузере, если это ссылка, с помощью кнопки «Окрыть в браузере»/ «Open in browser».

В поле «Результат проверки» /Safety Analysis отображается результат проверки ссылки с помощью нескольких API сервисов (список от 1 до 3). Напротив названия каждого из сервисов будет отображаться результат проверки («Базопасно»/ «SAFE», если ссылка не является вредоносной или фишинговой; «Опасно»/ «DANGEROUS», если ссылки является небезопасной).

Для того, чтобы вернуться к главному меня, необходимо нажать на кнопку «Вернуться в основное меню»/«Back to Main Menu».

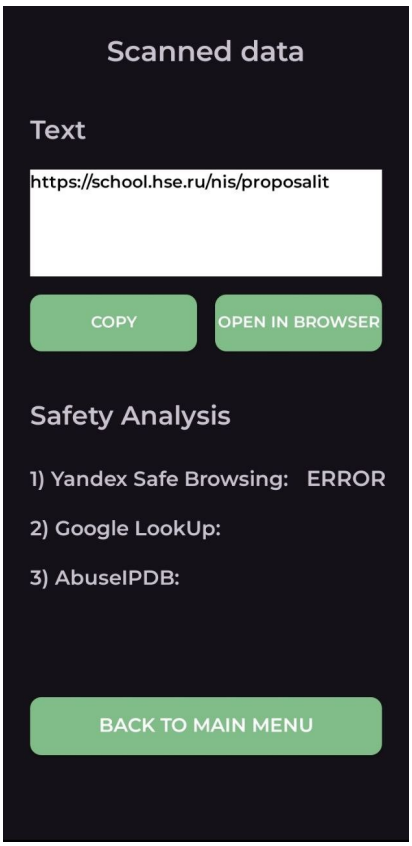

Рисунок 7, Меню анализа ссылки

Меню истории сканирований:

В меню истории сканирований отображаются все сканирования с указанием даты, времени и результата проверки ссылки.

Чтобы очистить список пользователю необходимо нажать на кнопку очистки (с значком крестика) с правом нижнем углу экрана.

Для копирования ссылки из истории сканирований пользователю необходимо нажать на элемент с соответствующей ссылкой, и она будет автоматически скопирована с буфер обмена телефона.

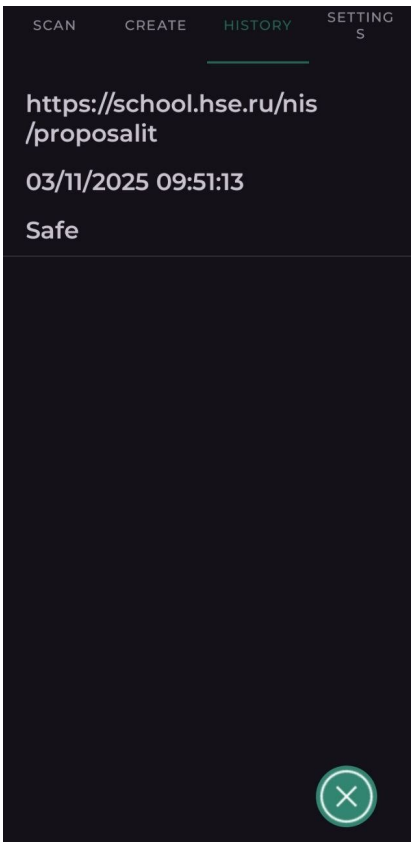

Рисунок 8, Меню истории сканирований

Меню настроек приложения:

В настройках пользователь может изменить язык приложения или поменять его оформление.

При нажатии на ползунок в пункте «Поменять тему приложения» оформление приложение будет изменяться либо на темную, либо на светлую темы (light/dark).

При нажатии на кнопку «Пользовательское руководство»/ «User guide» приложение откроет в браузере пользователя страницу с руководством по использованию приложения.

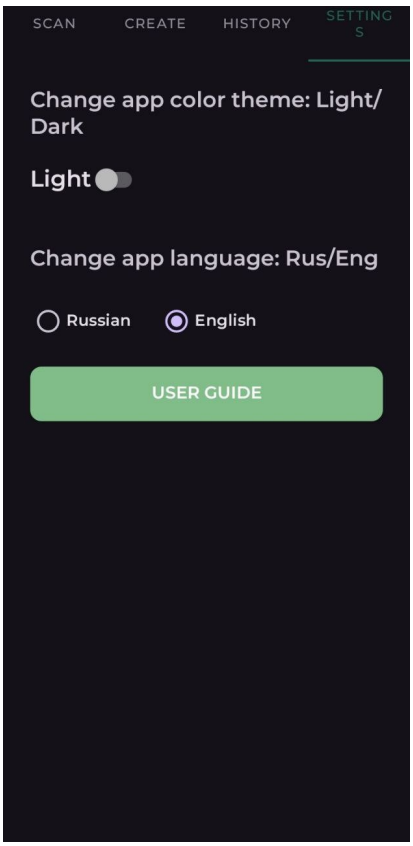

Рисунок 9, Меню настроек

3.1 ОБОСНОВАНИЕ СООТВЕТСТВИЯ

Итоговое решение соответствует поставленным целям и задачам. Результаты проведенных тестов подтверждают эффективность работы приложения. Таким образом, поставленная проблема была решена.

4 НОРМАТИВНО-ПРАВОВЫЕ АСПЕКТЫ

Приложение разработано в соответствии с ГОСТ Р 702.5.009-2024 «Мобильное приложение для смартфонов. Требования к функциональности мобильных приложений. Требования к удобству использования приложения» и удовлетворяет всем требованиям, указанным в документе.

В соответствии с федеральным законном РФ от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации. Общедоступная информация. Право на доступ к информации» сбор данных о

сайтах, проверяемых приложением, осуществляется только из открытых источников.

В соответствии с Федеральным законом от 27.07.2006 №152-ФЗ «О персональных данных. Принципы обработки персональных данных. Условия обработки персональных данных», приложение не осуществляет сбор, хранение или обработку персональных данных пользователей и не предлагает платного контента.

Использование API сервисов осуществляется только в соответствии с правилами использования, установленными разработчиками/владельцами сервисов.

4.1 РЕФЛЕКСИЯ

В процесс разработки достаточно часто возникали разного рода трудности, в основном связанные с написанием внутренней логики приложения на языке Kotlin. До этого у меня не было особого опыта разработки мобильных приложений, поэтому много приходилось делать в первый раз. Несмотря на это, я считаю, что у меня получилось сделать приложение не самого плохого качества.

Реализация проекта заинтересовала меня в мобильной разработке, и в дальнейшем я планирую делать и другие проекты под Android.

Я не собираюсь забрасывать данный проект. По моему мнению, в нем следует доработать и улучшить пользовательский интерфейс, улучшить алгоритм проверки ссылок на безопасность (возможно даже приобрести платный функционал API сервисов).

ЗАКЛЮЧЕНИЕ

QR-коды активно используются злоумышленниками для обмана пользователей и кражи их личных данных. Малый выбор приложений, которые позволяют проверять QR-кода на безопасность — это достаточно большая проблема. Приложение соответствует поставленным целям и задачам, удовлетворяет поставленному техническому заданию. На основе результатов тестирования можно сделать вывод, что приложение решает указанную проблему.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Самые популярные схемы мошенничества с QR-кодами // Эльдорадо.blog URL: https://blog.eldorado.ru/publications/samye-populyarnye-skhemy-moshennichestva-s-qr-kodami-i-sovety-kak-spasti-dengi-40345 (дата обращения: 02.03.2025).
2. Кильдюшкин Р. «Россиян предупредили о взломах Telegram при помощи домовых чатов. «Лаборатория Касперского»: Telegram могут взломать через QR-код домового чата» // Газеты.ru. - 20.01.2025 URL: https://www.gazeta.ru/tech/news/2025/01/20/24873542.shtml?updated (дата обращения: 02.03.2025).
3. Гуреева Ю. «Хакеры стали использовать QR-коды в мошеннических схемах» // Rg.ru. - 07.04.2024 URL:https://rg.ru/2024/04/07/vinovat-shtrih.html?ysclid=m8xnoemfvg463218555&utm_referrer=https%3A%2F%2Fyandex.ru%2F (дата обращения: 02.03.2025).
4. ZXing ("Zebra Crossing") barcode scanning library for Java, Android // Zxing URL: https://github.com/zxing/zxing (дата обращения: 05.03.2025).
5. API Documentation - AbuseIPDB // AbuseIPDB URL: https://www.abuseipdb.com/api.html (дата обращения: 10.03.2025).
6. Safe Browsing API // Yandex Safe Browsing URL: https://yandex.ru/dev/safebrowsing (дата обращения: 10.03.2025).
7. Safe Browsing Lookup API // Google Safe Browsing URL: https://developers.google.com/safe-browsing/v4/lookup-api (дата обращения: 10.03.2025).

8. «Kaspersky Premium быстро разряжает батарею» // Kaspersky club URL: https://forum.kasperskyclub.ru/topic/447493-kaspersky-premium-bystro-sadit-batareju/ (дата обращения: 11.03.2025).

9. «Снижение скорости интернет-соединения после установки программы Лаборатории Касперского» // Kaspersky URL: https://support.kaspersky.ru/common/error/other/9042 (дата обращения: 11.03.2025).

10. «Mobile Apps are leaking your API Keys: Discover how to prevent it» // Build38 URL: https://build38.com/blog/threats-mobile-app-device/mobile-apps-leaking-api-keys/ (дата обращения: 15.03.2025).

11. API Attacks: Understanding and Protecting Your Infrastructure // CybelAngel URL: https://cybelangel.com/api-attacks-protecting-your-infrastructure/#6-man-in-the-middle-mitm (дата обращения: 15.03.2025).

12. OpenPhish Database // OpenPhish URL: https://openphish.com/phishing_database.html (дата обращения: 28.03.2025).

13. URLhaus Database // URLhaus URL: https://urlhaus.abuse.ch/browse/ (дата обращения: 28.03.2025).