**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

**Курсовая работа**

**по дисциплине «Разработка приложений для мобильных платформ»**

Тема: **Приложение для авторизации и начала сессий прокторинга**

|  |  |  |
| --- | --- | --- |
| Студент гр. 9304 |  | Матросов Д.В. |
| Студент гр. 9304 |  | Сорин А.В. |
| Студент гр. 9304 |  | Боблаков Д.С. |
| Преподаватель |  | Заславский М.М. |

Санкт-Петербург

2023

**ЗАДАНИЕ на Курсовую Работу**

|  |  |  |
| --- | --- | --- |
| Студент Матросов Д.В.  Студент Сорин А.В.  Студент Боблаков Д.С. | | |
| Группа 9304 | | |
| Тема работы: Приложение для авторизации и начала сессий прокторинга | | |
| Содержание пояснительной записки:   * Аннотация * Содержание * Введение * Сценарии использования * Пользовательский интерфейс * Разработанное приложение * Последовательность действий для осуществления сценариев использования * Выводы * Список литературы * Приложения | | |
| Предполагаемый объем пояснительной записки:  Не менее 20 страниц. | | |
| Дата выдачи задания: 13.02.2023 | | |
| Дата сдачи реферата: 15.03.2023 | | |
| Дата защиты реферата: 03.04.2023 | | |
| Студент |  | Матросов Д.В. |
| Студент |  | Сорин А.В. |
| Студент |  | Боблаков Д.С. |
| Преподаватель |  | Заславский М.М. |

**Аннотация**

В данной работе был спроектирован интерфейс мобильного приложения, реализующего функции по авторизации и началу сессий прокторинга. Приложение было написано на языке Kotlin с использованием Android Studio в качестве среды разработки.

**Summary**

In this work, the interface of a mobile application was designed that implements functions for authorization and the beginning of proctoring sessions. The application was written in Kotlin using Android Studio as a development environment.

**содержание**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Введение | | | | | 6 |
| 1. | Сценарии использования | | | | 7 |
|  | 1.1. | | Сценарий добавления дохода | | 7 |
|  | 1.2. | | Сценарий добавления расхода | | 8 |
|  | 1.3. | | Сценарий просмотра статистики | | 9 |
| 2. | Пользовательский интерфейс | | | | 10 |
|  | 2.1. | | | Макет интерфейса с графом переходов | 10 |
|  | 2.2. | | | Целевые устройства, обоснование требований и максимально подробные характеристики | 11 |
|  | 2.2.1. | | | Тип устройств | 12 |
|  | 2.2.2. | | | Аппаратная составляющая | 13 |
| 3. | Разработанное приложение | | | | 15 |
|  | 3.1. | | | Краткое описание | 15 |
|  | 3.2. | | | Схема архитектуры | 16 |
|  | 3.3. | | | Использованные технологии (внешние) | 17 |
|  | 3.4. | | | Использованные модули/системные библиотеки вашей платформы | 18 |
|  | 3.5. | | | Стратегия для обеспечения кросс-платформенности приложения | 19 |
|  | 3.6. | | | Ссылки на раздел Приложение | 20 |
| 4. | Последовательность действий для осуществления сценариев использования | | | | 21 |
|  | 4.1. | | | Измерение последовательности действий для осуществления сценариев использования | 21 |
|  | 4.2. | | | Пути для сокращения последовательности | 25 |
| 5. | Выводы | | | | 19 |
|  | 5.1. | | | Достигнутые результаты | 20 |
|  | 5.2. | | | Недостатки и пути для улучшения полученного решения | 20 |
|  | 5.3. | | | Будущее развитие решения | 20 |
| 6. | Список литературы | | | | 21 |
| 7. | Приложения | | | | 22 |
|  | | 7.1. | | Инструкция для пользователя | 22 |
|  | | 7.2. | | Снимки экрана приложения | 24 |

**введение**

Существует следующая проблема: возрастает необходимость объективно оценивать знания учащихся, находящихся вне стен учебных заведений. Это стало особенно актуально после 2020 года. Если бы у университета было бы приложение, которое позволяло студентам подтверждать самостоятельность выполнения работы, то это расширило бы возможности и географию обучающихся.

Цель работы — создать приложение, которое позволит подтверждать личность и окружение обучающегося для подтверждения объективной оценки знаний.

Причины, по которым приложение разрабатывается под мобильные устройства:

1. Пользователям придется использовать камеру для подтверждения личности и своего окружения, что гораздо удобнее делать с помощью мобильного устройства.

2. Тенденции современного мира таковы, что люди чаще начинают использовать мобильные телефоны, чем настольные компьютеры или ноутбуки.

**1. Сценарии использования**

**2. Пользовательский интерфейс**

**2.1. Макет**

Рисунок 1 — Макет

**2.2. Целевые устройства, обоснование требований и максимально подробные характеристики**

Целевые устройства — смартфоны под управлением OS Android, с наличием камеры.

**2.2.1. Тип устройств**

Смартфоны.

**2.2.2. Аппаратная составляющая**

Версия Android — не ниже 9.0 (Pie), версия sdk — не ниже 28.

**Экран**

hw.lcd.density 560

hw.lcd.height 3120

hw.lcd.width 1440

**Прочее**

avd.ini.displayname Pixel 6 Pro API 30

avd.ini.encoding UTF-8

AvdId Pixel\_6\_Pro\_API\_30

disk.dataPartition.size 2G

fastboot.chosenSnapshotFile

fastboot.forceChosenSnapshotBoot no

fastboot.forceColdBoot no

fastboot.forceFastBoot yes

hw.accelerometer yes

hw.arc false

hw.audioInput yes

hw.battery yes

hw.camera.back virtualscene

hw.camera.front emulated

hw.cpu.ncore 2

hw.device.hash2 MD5:a8abfd3536f3d35e4ba2041a7b99f40e

hw.device.manufacturer Google

hw.device.name pixel\_6\_pro

hw.dPad no

hw.gps yes

hw.gpu.enabled yes

hw.gpu.mode auto

hw.initialOrientation Portrait

hw.keyboard yes

hw.mainKeys no

hw.ramSize 1536

hw.sdCard yes

hw.sensors.orientation yes

hw.sensors.proximity yes

hw.trackBall no

image.androidVersion.api 30

image.sysdir.1 system-images/android-30/google\_apis/x86/

PlayStore.enabled false

runtime.network.latency none

runtime.network.speed full

showDeviceFrame yes

skin.dynamic yes

tag.display Google APIs

tag.id google\_apis

vm.heapSize 384

**3. Разработанное приложение**

**3.1. Краткое описание**

Приложение на данный момент представляет из себя практически полностью работоспособный UI, который обрабатывает входящие от пользователя данные и выводит результат в зависимости от пришедших данных (в данном случае QR-Code).

**3.2. Схема архитектуры**

Экран MainActivity — главный экран, который видит пользователь после запуска приложения. С данного экрана можно перейти на экран считывания QR-кода или на экран с информацией об авторах.

Экран CheckQrActivity — экран, на месте которого можно проверить верность QR-кода с информацией о сессии прокторинга.

Экран AboutActivity — экран, на котором можно видеть имена авторов приложения.

Экран BadResultActivity — экран, который отображается при ошибке ввода данных.

Экран CheckFaceActivity — Экран на котором считывается фото судента.

Экран CheckRoomActivity — экран, можно записать видео своего окружения.

Экран GoodResultActivity — экран, на который пользователь попадает при успешном входе в сессию прокторинга.

Экран ScannerActivity — экран, на месте которого можно считать QR-код.

**3.3. Использованные технологии (внешние)**

При разработке приложения не использовались внешние технологии, по причине того, что для разработки одного лишь интерфейса достаточно было того, что предоставляет Android.

**3.4. Использованные модули/системные библиотеки вашей платформы**

Список использованных модулей следующий:

1. android.content.Intent

2. android.net.Uri

3. android.os.Bundle

4. android.view.Window

5. android.widget.Button

6. androidx.activity.result.contract.ActivityResultContracts

7. androidx.appcompat.app.AppCompatActivity

8. androidx.core.content.FileProvider

9. android.widget.Toast

10. androidx.core.content.FileProvider

11. androidx.activity.result.ActivityResultLauncher

12. android.Manifest

13. android.content.pm.PackageManager

**3.5. Стратегия для обеспечения кроссплатформенности приложения**

Приложение было написано на языке Kotlin, что позволяет не думать об обеспечении кросплатформенности приложения.

**3.6. Ссылки на раздел Приложение**

Ссылка на репозиторий с приложением находится в разделе «Список литературы».

**4. Последовательность действий для осуществления сценариев использования**

**4.1. Измерение последовательности действий для осуществления сценариев использования**

В таблице №1 представлена информация о количестве действий для осуществления сценариев использования.

Таблица №1 — Последовательность действий для осуществления сценариев использования

|  |  |
| --- | --- |
|  | Работа с приложением |
| Клик | 4 |
| Двойной клик | 0 |
| Множественное касание | 0 |
| Камера | 3 |
| Микрофон | 0 |
| Ожидание | 0 |
| Физические кнопки | 0 |
| Прокрутка | 0 |
| Ввод текста | 0 |
| **Итого** | **7** |

На рисунке 2 показана последовательность действий для осуществления сценария «Работа с приложением».

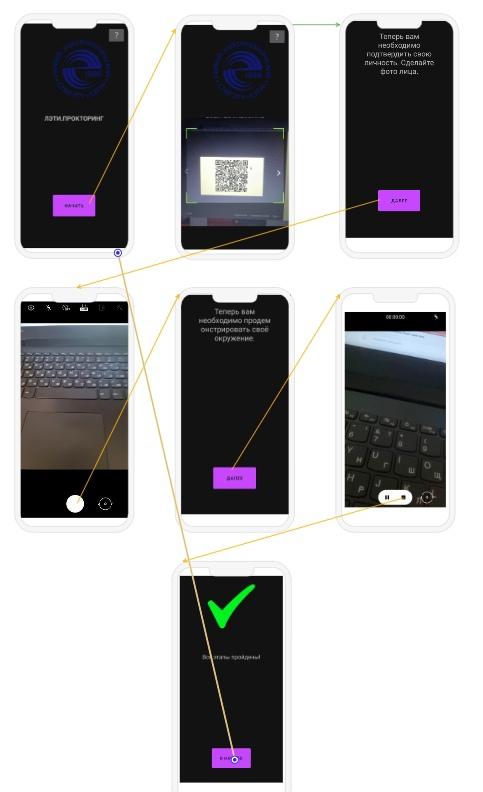


Рисунок 2 — Работа с приложением

**4.2. Пути для сокращения последовательности**

Исходя из данных, полученных при оценке интерфейса, можно сделать вывод, что интерфейс предельно прост. Старание уменьшить количество кликов выглядит не совсем оправданно. Потенциально можно уменьшить количество использовании камеры с 3 до 2 путем объединения шагов подтверждения личности пользователя и подтверждения собственного окружения.

**5. Выводы**

**5.1. Достигнутые результаты**

По итогам выполнения работы было разработано приложение, которое позволяет считывать QR-код с информацией о предстоящей сессией прокторинга, снимать портрет пользователя и записывать видео окружающей его обстановки.

**5.2. Недостатки и пути для улучшения полученного решения**

Основной недостаток приложения на данный момент — достаточно

малые возможности по определению и подтверждению личности. В настоящее время только QR-код содержит информацию о сессии и пользователе. Решить эту проблему можно путем доработки бизнес-логики, соединения клиентской части приложения с сервером.

**5.3. Будущее развитие решения**

В будущем планируется дальнейшая разработка приложения, с целью доведения его до полностью функционирующего, чтобы им можно было пользоваться непосредственно в стенах университета. После того, как клиентская часть приложения будет полноценно разработана, а серверная часть создана, оно будет опубликовано в Play Market.

**6. список Литературы**

1. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного программирования. - Спб: Питер, 2001. - 368с.: ил ISBN 5-272-00355-1
2. Экель Б., Исакова С. Atomic Kotlin. - Mindview LLC, 2021. - 636с.: ил ISBN ‏ : ‎ 978-0981872551
3. Федотенко М. Разработка мобильных приложений. Первые шаги. - Бином. Лаборатория знаний, 2019. - 338с.: ил ISBN: 978-5-00101-640-3
4. Ссылка на разработанное приложение // URL: https://github.com/moevm/adfmp1h23-proctoring

**7. ПРИЛОЖЕНИЯ**

**7.1. Инструкция для пользователя**

Для работы с приложением необходимо его скачать и открыть.

Попадая на главный экран, можно нажать на знак вопроса, что переместит пользователя на страницу с информацией.

Также на главном экране есть кнопка «Начать», нажав на которую, откроется QR-сканнер, которым необходимо отсканировать нужный QR-код с информацией о сессии прокторинга. Далее пользователю предложат сфотографировать свое лицо для подтверждения личности. После этого пользователю предложат снять видео обстановки вокруг себя.

Если все шаги будут выполнены верно, пользователь увидит зеленую галку – знак, что все отработало корректно. Если пользователь увидел красный крест, значит произошла ошибка. Из обоих этих положениях можно вернутся в начало с помощью кнопки «Вернуться».

**7.2. Снимки экрана приложения**



Рисунок 3 — Главный экран

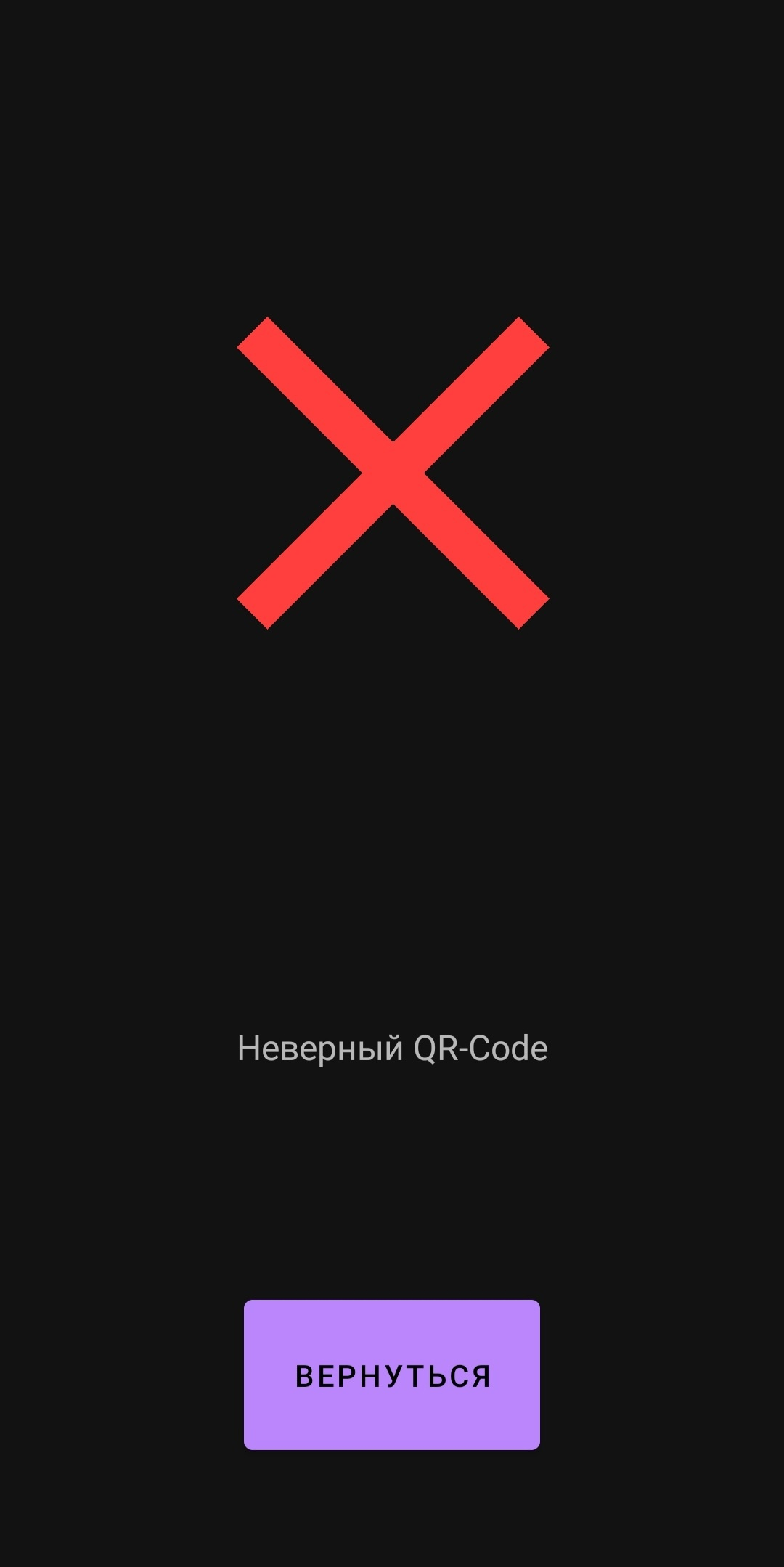


Рисунок 4 — Экран ошибки

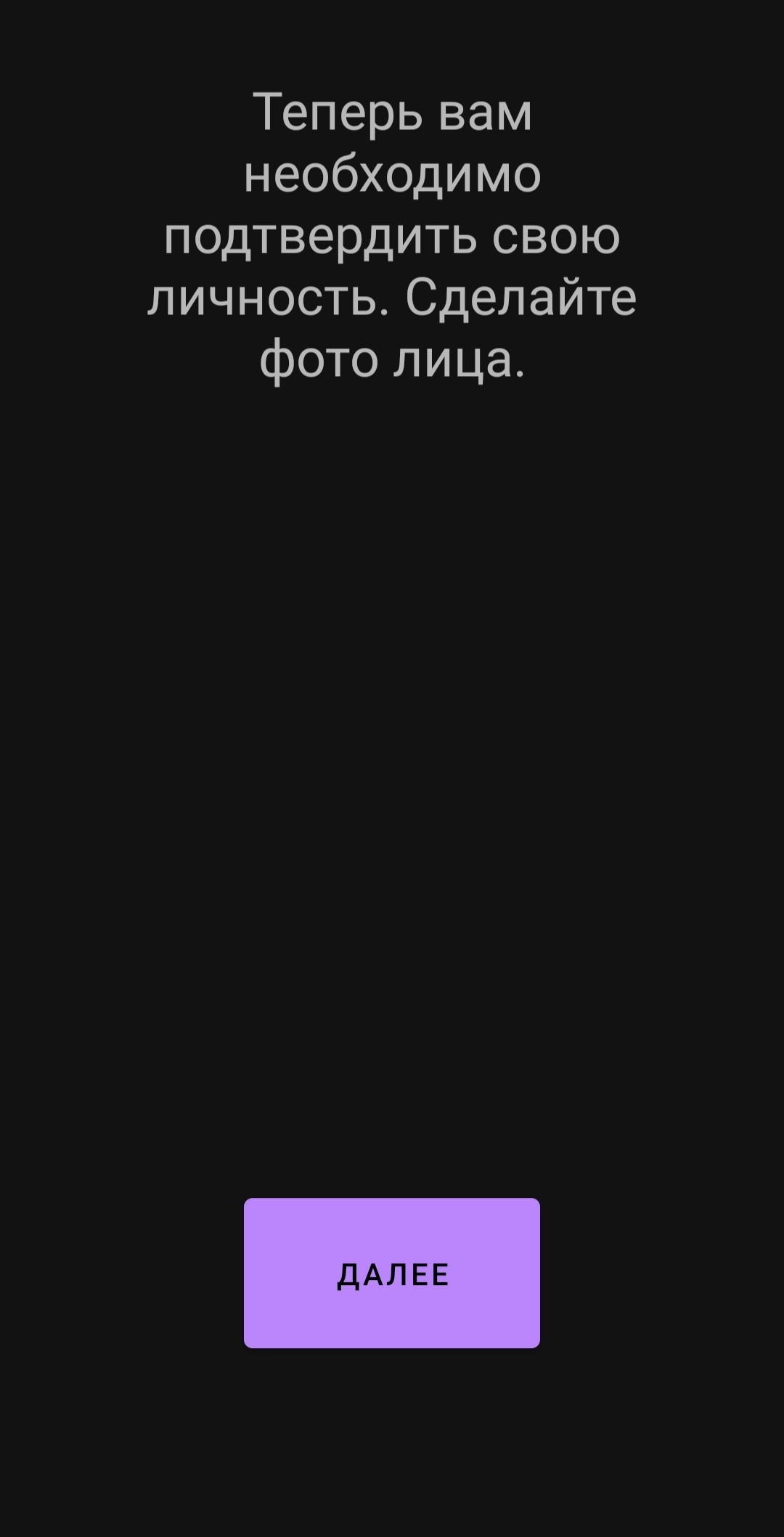


Рисунок 5 — Экран предложения подтверждения личности

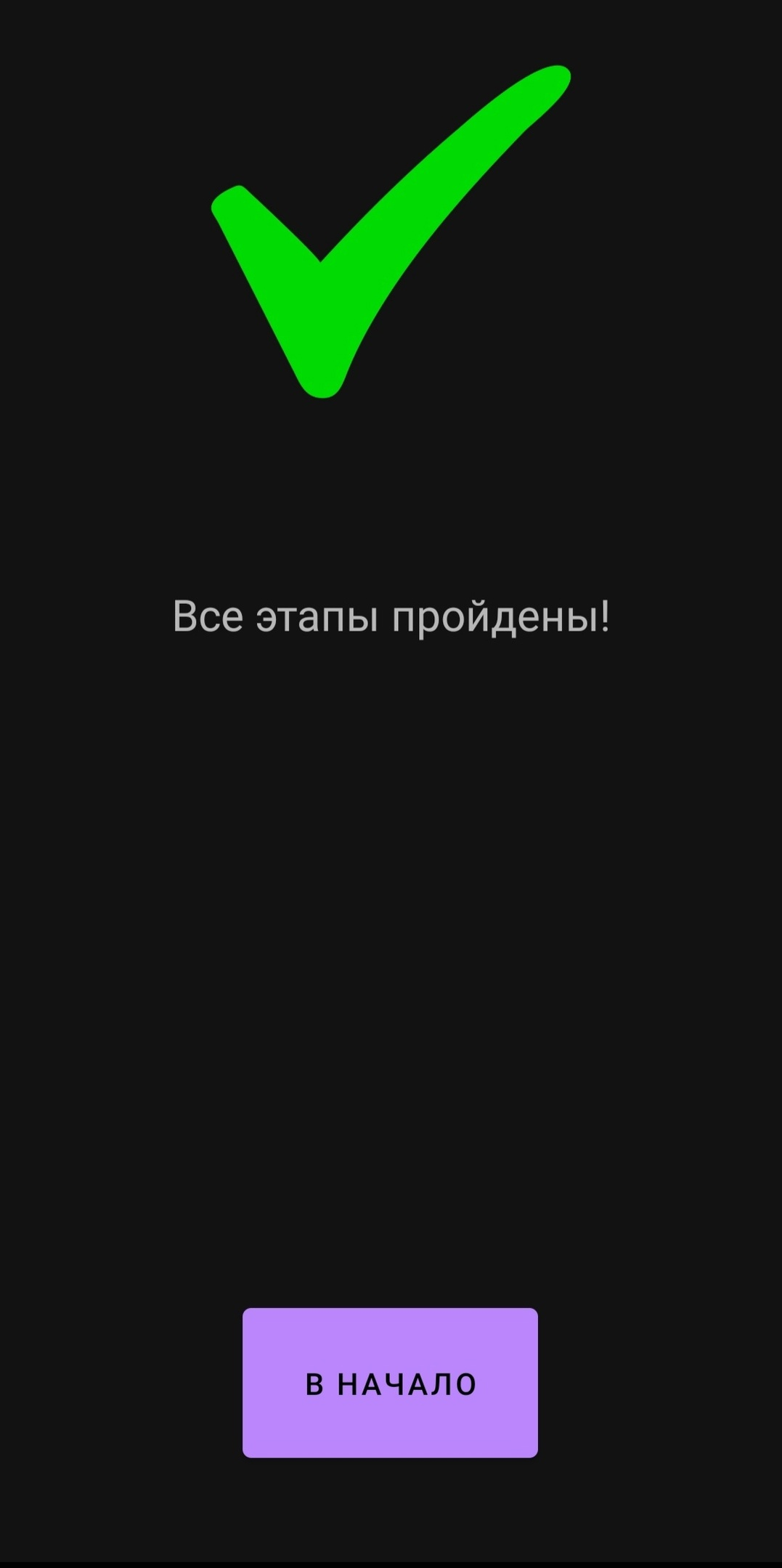


Рисунок 6 — Экран положительного исхода

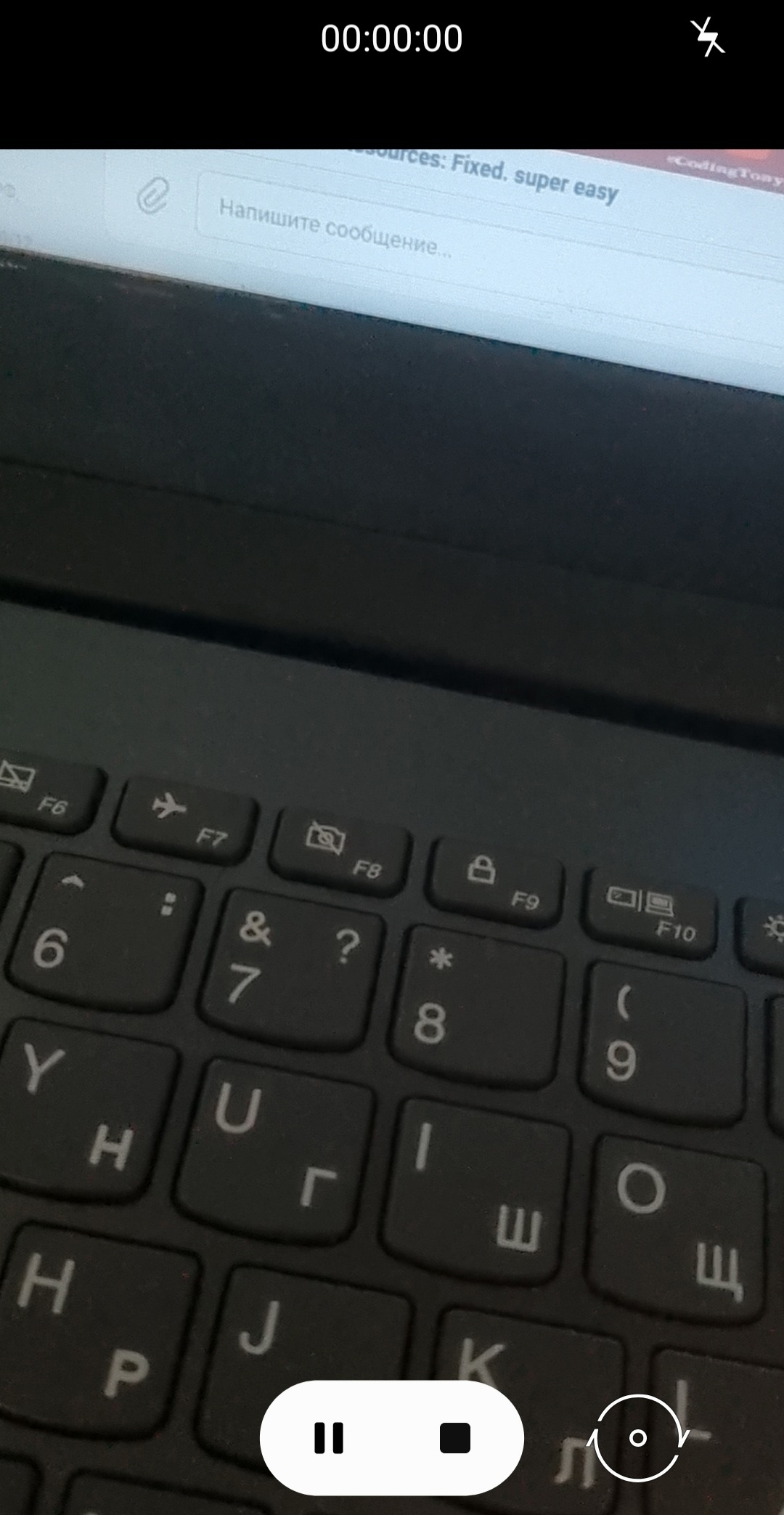


Рисунок 7 — Экран окружения(камера)



Рисунок 8 — Экран предложения проверки окружения

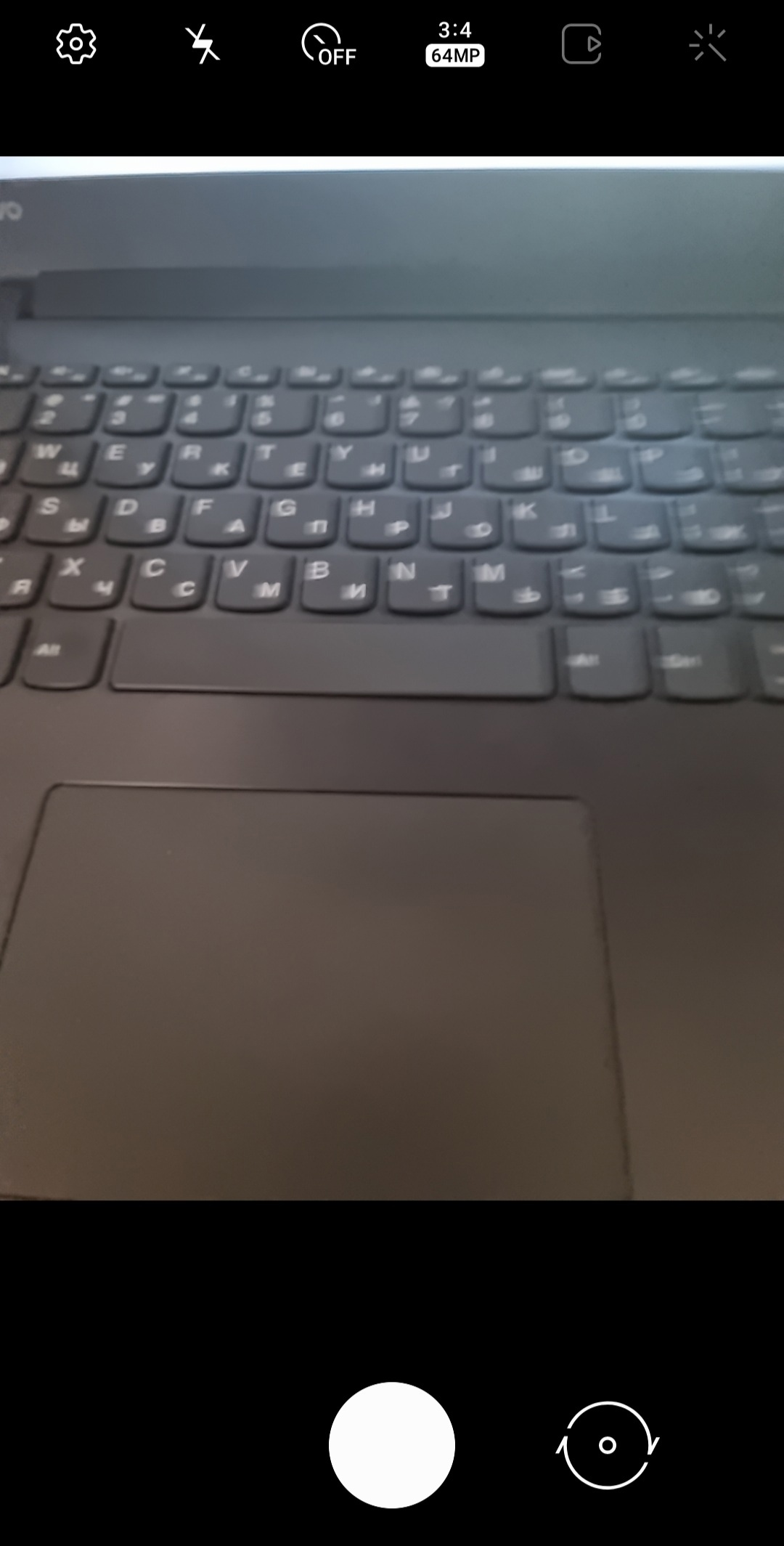


Рисунок 9 — Экран предложения проверки личности (камера)

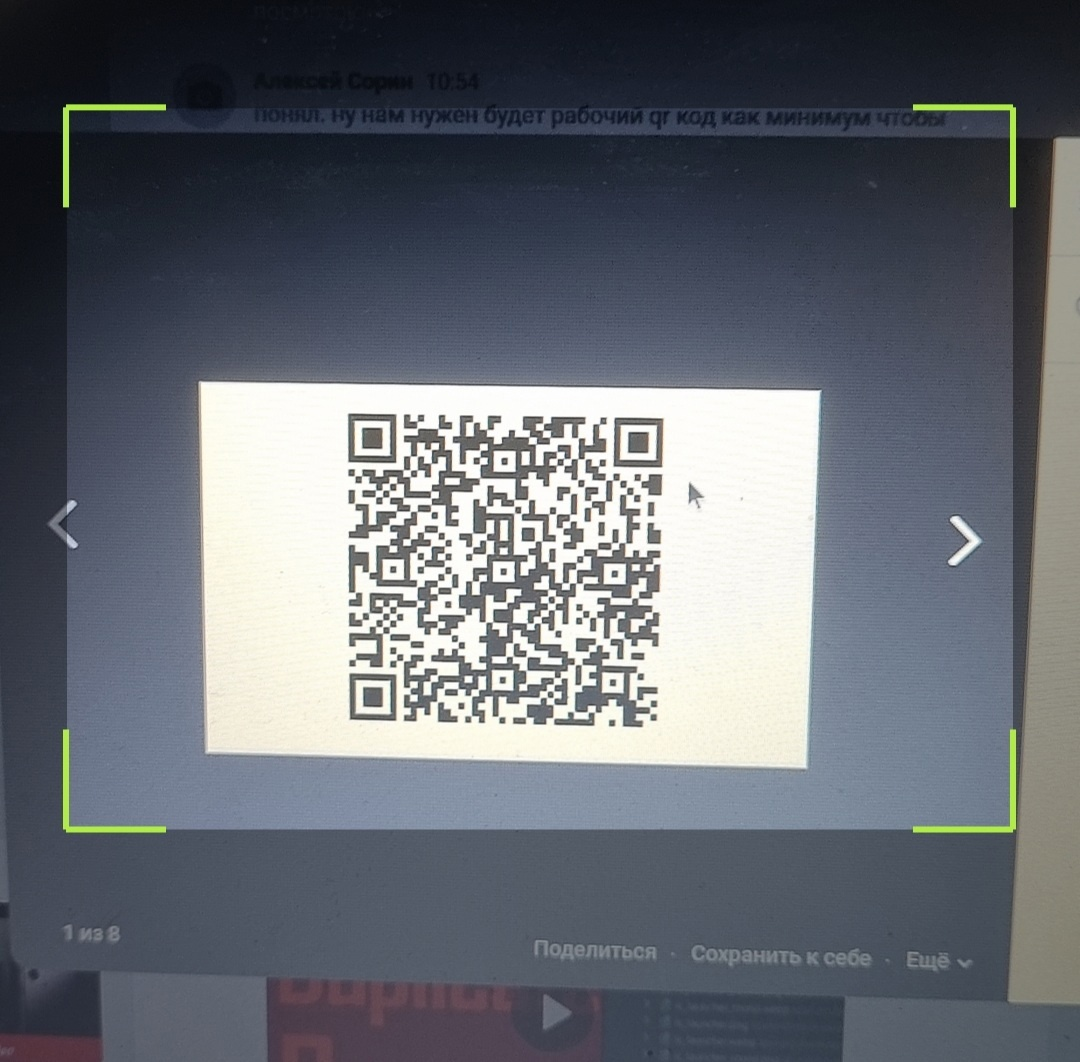


Рисунок 10 — Экран считываний QR-кода(камера)



Рисунок 11 — Экран about