



Министерство науки и высшего образования Российской  
Федерации Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Московский государственный технический  
университет имени Н.Э. Баумана  
(национальный исследовательский  
университет)» (МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА

---

***РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К КУРСОВОЙ  
РАБОТЕ НА ТЕМУ:  
Разработка базы данных для информационной  
системы распознавания лиц***

Студент ИУ9-62Б  
(Группа)

П. С. Самохвалова  
(Подпись, дата) (И. О. Фамилия)

Руководитель курсовой работы

Д. П. Посевин  
(Подпись, дата) (И. О. Фамилия)

## СОДЕРЖАНИЕ

ПОСТАНОВКА ЗАДАЧИ.....	3
1. Изучение теоретических основ создания систем распознавания лиц.....	4
1.1. Основные понятия, используемые в системах распознавания лиц.....	4
1.2. Принципы работы систем распознавания лиц.....	4
1.3. Использование баз данных для систем распознавания лиц.....	5
2. Выбор стека технологий.....	7
3. Создание реляционной модели базы данных.....	9
3.1. Создание модели «сущность – связь».....	9
3.2. Описание сущностей.....	9
3.3. Преобразование модели «сущность – связь» в реляционную модель.....	11
3.4. Описание таблиц.....	12
4. Создание приложения распознавания лиц с базой данных.....	16
4.1. Реализация графического интерфейса.....	16
4.2. Создание таблиц.....	16
4.3. Реализация регистрации и авторизации.....	22
4.4. Распознавание лиц и их идентификация.....	23
4.5. Ввод и изменение имен.....	25
4.6. Вывод информации, хранящейся в базе данных.....	26
5. Результаты работы.....	28
5.1. Регистрация и авторизация.....	28
5.2. Распознавание лиц и ввод имён.....	30
5.3. Содержимое папки dataset.....	35
5.4. Вывод информации из базы данных.....	38
ЗАКЛЮЧЕНИЕ.....	44
СПИСОК ЛИТЕРАТУРЫ.....	45

## ПОСТАНОВКА ЗАДАЧИ

Целью курсовой работы является создание базы данных для информационной системы распознавания лиц.

Для достижения поставленной цели необходимо решить следующие задачи:

- 1) выбрать стек технологий для реализации базы данных;
- 2) создать реляционную модель базы данных;
- 3) создать графический интерфейс приложения;
- 4) реализовать приложение, осуществляющее распознавание лиц и использующее базу данных для хранения информации;
- 5) реализовать вывод информации из базы данных;
- 6) провести тестирование приложения.

## **1. Изучение теоретических основ создания системы распознавания лиц с базой данных**

### **1.1. Основные понятия, используемые в системах распознавания лиц**

Распознавание лиц – это способ идентификации или подтверждения личности человека по его лицу. Систему распознавания лиц можно использовать для идентификации людей на фотографиях, видео или в режиме реального времени.

Распознавание лиц – это категория биометрических систем аутентификации. К другим видам биометрических систем аутентификации относятся такие виды как: распознавание голоса, распознавание отпечатков пальцев и распознавание сетчатки или радужной оболочки глаза. Данные технологии в основном применяются для обеспечения безопасности и соблюдения правопорядка, однако в настоящее время можно наблюдать возрастание интереса к другим сферам применения этих технологий.

### **1.2. Принципы работы систем распознавания лиц**

Технология распознавания лиц работает посредством сопоставления лиц людей, проходящих мимо специальных камер, с уже имеющимися изображениями людей. Есть разные технологии распознавания лиц, но в общем и целом они работают следующим образом:

#### **Шаг 1. Обнаружение лица**

На первом шаге камера обнаруживает и фиксирует положение изображения лица, как одного, так и в толпе. На изображении может быть человек, смотрящий в анфас или в профиль.

#### **Шаг 2. Анализ лица**

Далее, на втором шаге, выполняется снимок и проводится анализ изображения лица. Подавляющее большинство технологий распознавания лиц применяют 2D, а не 3D-изображения. Это связано с тем, что 2D-изображения легче и удобнее сопоставлять с общедоступными фотографиями или фотографиями в базе данных. Программа проводит считывание геометрии

лица. Ключевыми параметрами характеристики изображения лица являются следующие: расстояние между глазами, глубина глазниц, расстояние от лба до подбородка, форма скул и контуры губ, ушей и подбородка. Целью является определение черт, отличающих данное конкретное лицо.

#### Шаг 3. Преобразование изображения в данные

На третьем шаге, при проведении анализа аналоговая информация (лицо) преобразуется в набор цифровой информации на основе черт лица человека. По сути, анализ лица представляет собой математическую формулу. Цифровой код носит название «отпечаток лица». У всякого человека имеется свой уникальный отпечаток лица.

#### Шаг 4. Поиск совпадения

На четвертом шаге проходит сравнение отпечатка лица с данными в базе известных лиц. В случае, если отпечаток лица совпадает с изображением в базе данных для распознавания лиц, устанавливается, чье это лицо [2].

### **1.3. Использование баз данных для систем распознавания лиц**

Базы данных играют очень важную роль в системах распознавания лиц. Они необходимы для обучения и тестирования алгоритмов распознавания лиц, а также для хранения и управления информацией о пользователях, которые используют систему распознавания.

В системах распознавания лиц базы данных используются для хранения изображений лиц, которые затем применяются для обучения алгоритмов распознавания. Чтобы система могла распознавать лица пользователей, базы данных должны содержать качественные и разнообразные изображения лиц с описанием технических характеристик и биометрических данных.

Одной из ключевых задач при разработке и использовании баз данных для систем распознавания лиц является обеспечение безопасности хранения личной информации пользователей. Для этой цели базы данных обязательно должны быть защищены от несанкционированного доступа и взлома.

Помимо обучения алгоритмов и хранения информации о пользователях, базы данных также используются для анализа и оценки работоспособности

систем распознавания лиц, а также для выявления ошибок и недостатков, которые могут возникнуть в процессе использования системы.

Кроме того, базы данных для распознавания лиц могут использоваться в различных приложениях и отраслях, таких как безопасность, контроль доступа, государственная безопасность, финансы, медицина и т.д.

## 2. Выбор стека технологий

Для выполнения курсовой работы был выбран язык программирования Python, его модули и библиотеки и объектно-реляционная система управления базами данных PostgreSQL.

PostgreSQL – это объектно-реляционная система управления базами данных (ORDBMS), основанная на POSTGRES версии 4.2, разработанная на факультете компьютерных наук Калифорнийского университета в Беркли. POSTGRES стал пионером во многих концепциях, которые стали доступны в некоторых коммерческих системах баз данных гораздо позже.

PostgreSQL является потомком оригинального кода Berkeley с открытым исходным кодом. Он поддерживает большую часть стандарта SQL и предлагает множество современных функций:

- сложные запросы;
- внешние ключи;
- триггеры;
- обновляемые представления;
- целостность транзакций;
- управление многовариантным параллелизмом.

Кроме того, PostgreSQL может быть расширен пользователем многими способами, например, путем добавления новых:

- типов данных;
- функций;
- операторов;
- агрегатных функций;
- методов индексирования;
- процедурных языков.

А благодаря свободной лицензии PostgreSQL может использоваться, модифицироваться и распространяться кем угодно бесплатно для любых целей, будь то частные, коммерческие или академические [1].

Для написания программы на языке программирования Python были выбраны для использования его модули и библиотеки:

- tkinter – библиотека для разработки графического интерфейса;
- cv2 – библиотека компьютерного зрения, которая предназначена для анализа, классификации и обработки изображений;
- face\_recognition – библиотека распознавания лиц;
- speech\_recognition – библиотека распознавания речи;
- psycopg2 – модуль для подключения к PostgreSQL, выполнения SQL-запросов и других операций с базой данных;
- os – модуль для работы с операционной системой;
- time – модуль для работы со временем;
- PIL – библиотека Python Imaging Library для работы с изображениями;
- shutil – модуль для высокоуровневых операций с файлами и коллекциями файлов;
- pickle – модуль, который реализует двоичные протоколы для сериализации и десериализации структуры Python объекта;
- sys – модуль, который обеспечивает доступ к некоторым переменным и функциям, взаимодействующим с интерпретатором Python;
- re – модуль для работы с регулярными выражениями.



### 3. Создание реляционной модели базы данных

#### 3.1. Создание модели «сущность – связь»

В данной модели содержится информация об изображениях, идентифицированных людях, не идентифицированных людях, данных для доступа к системе, полной статистике системы, типах изменений.

На рисунке 3.1 представлена созданная модель «сущность – связь».

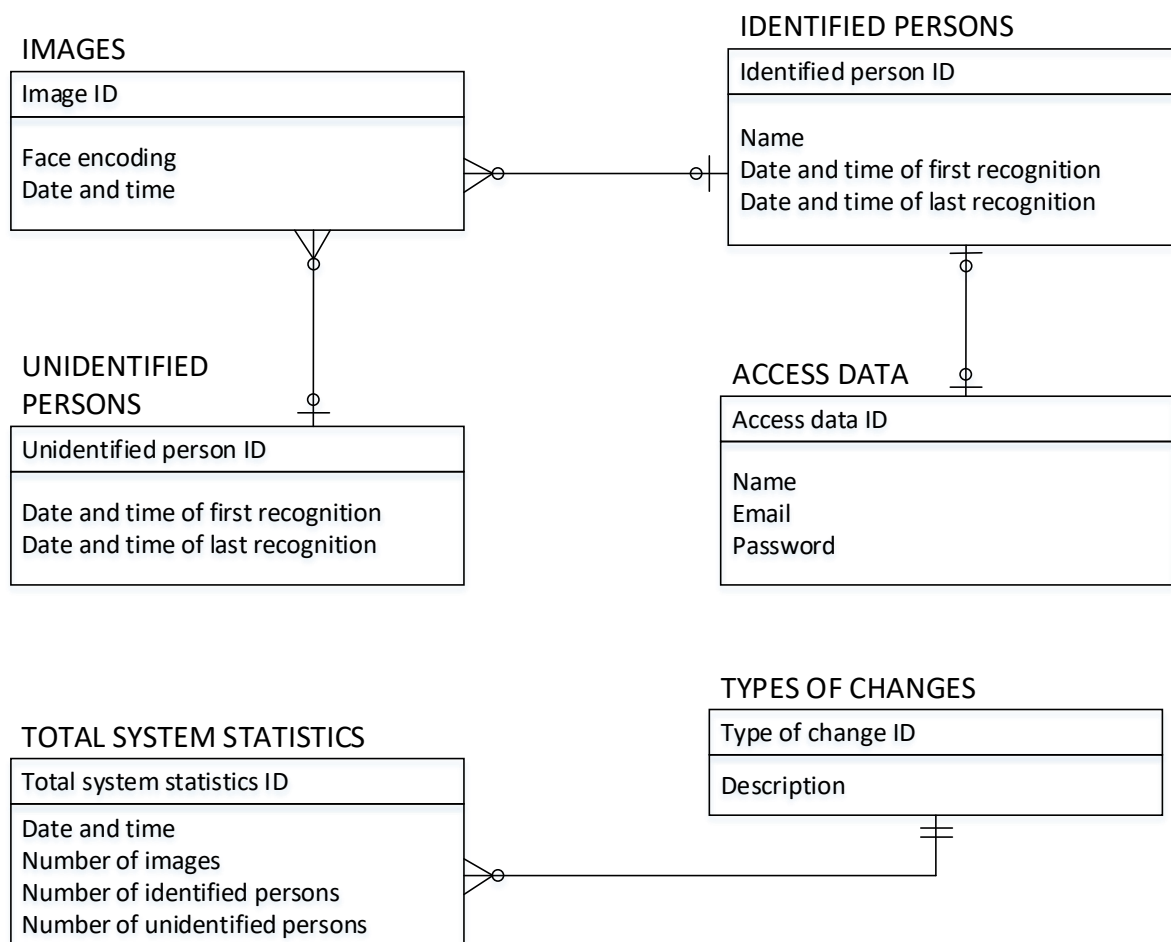


Рисунок 3.1 – Модель «сущность – связь»

#### 3.2. Описание сущностей

1) **IMAGES** – сущность, являющаяся абстракцией изображения.

Атрибуты:

- ImageId – идентификатор;
- FaceEncoding – кодировка лица;
- DateAndTime – дата и время сохранения изображения.

2) **IDENTIFIED PERSONS** – сущность, являющаяся абстракцией идентифицированного человека.

Атрибуты:

- IdentifiedPersonId – идентификатор;
- Name – имя;
- DateAndTimeOfFirstRecognition – дата и время первого распознавания камерой;
- DateAndTimeOfLastRecognition – дата и время последнего распознавания камерой.

3) UNIDENTIFIED PERSONS – сущность, являющаяся абстракцией не идентифицированного человека.

Атрибуты:

- UnidentifiedPersonId – идентификатор;
- DateAndTimeOfFirstRecognition – дата и время первого распознавания камерой;
- DateAndTimeOfLastRecognition – дата и время последнего распознавания камерой.

4) ACCESS DATA – сущность, являющаяся абстракцией данных для доступа.

Атрибуты:

- AccessDataId – идентификатор;
- Name – имя;
- Email – адрес электронной почты;
- Password – пароль.

5) TOTAL SYSTEM STATISTICS – сущность, являющаяся абстракцией полной статистики системы.

Атрибуты:

- TotalSystemStatisticsId – идентификатор;
- DateAndTime – дата и время сохранения статистики;
- NumberOfImages – число изображений;
- NumberOfIdentifiedPersons – число идентифицированных

людей;

- NumberOfUnidentifiedPersons – число не идентифицированных людей.

6) TYPES OF CHANGES – сущность, являющаяся абстракцией типов изменений.

Атрибуты:

- TypeOfChangeId – идентификатор;
- Description – описание изменения.

Рассмотрим связи между сущностями.

1) Рассмотрим связь между сущностями IMAGES и IDENTIFIED PERSONS. Изображению может соответствовать один идентифицированный человек, а может не соответствовать ни одного. У идентифицированного человека может быть 0 или более изображений.

2) Рассмотрим связь между сущностями IMAGES и UNIDENTIFIED PERSONS. Изображению может соответствовать один не идентифицированный человек, а может не соответствовать ни одного. У не идентифицированного человека может быть 0 или более изображений.

3) Рассмотрим связь между сущностями IDENTIFIED PERSONS и ACCESS DATA. У идентифицированного человека могут быть одни данные для доступа, а может их не быть вовсе. Данным для доступа соответствует один идентифицированный человек или не соответствует ни одного.

4) Рассмотрим связь между сущностями TOTAL SYSTEM STATISTICS и TYPES OF CHANGES. Каждой новой статистике может соответствовать ровно один тип изменения. Каждому типу изменения может соответствовать 0 или более статистик.

### **3.3. Преобразование модели «сущность – связь» в реляционную модель.**

Далее модель «сущность – связь» была преобразована в реляционную

модель. На рисунке 3.3 представлена реляционная модель базы данных.

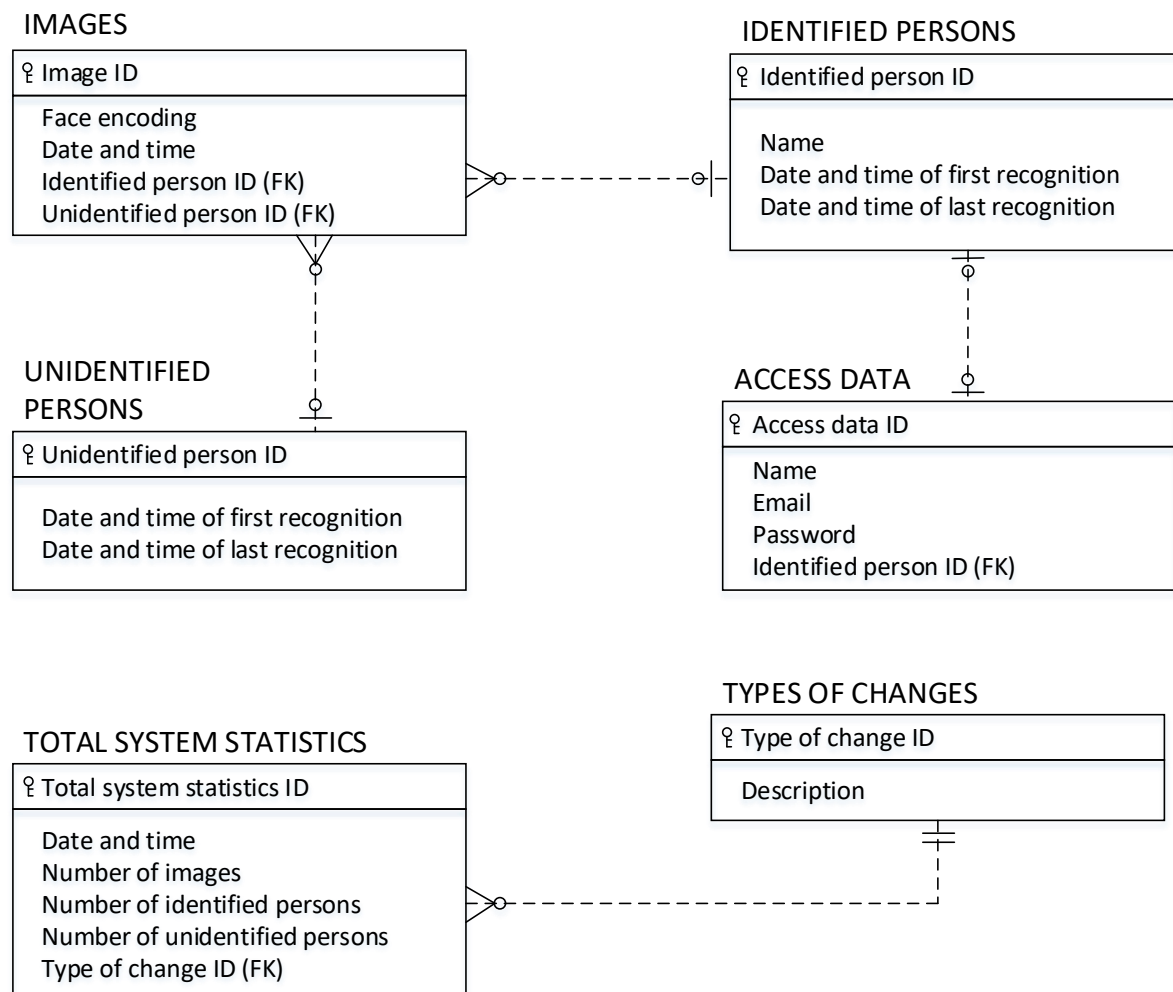


Рисунок 3.3 – Реляционная модель

### 3.4. Описание таблиц

Опишем таблицы реляционной модели.

В таблице 3.4.1 представлены описания столбцов для IMAGES.

Таблица 3.4.1 – IMAGES

Column name	Type	Key	NULL Status	Remarks
ImageId	SERIAL	PRIMARY KEY	NOT NULL	
FaceEncoding	BYTEA	No	NOT NULL	
DateAndTime	TIMESTAMP	No	NOT NULL	
IdentifiedPersonId	INTEGER	FOREIGN KEY	NULL	

UnidentifiedPersonId	INTEGER	FOREIGN KEY	NULL	
----------------------	---------	----------------	------	--

В таблице 3.4.2 представлены описания столбцов для IDENTIFIED PERSONS.

Таблица 3.4.2 – IDENTIFIED PERSONS

Column name	Type	Key	NULL Status	Remarks
IdentifiedPersonId	SERIAL	PRIMARY KEY	NOT NULL	
Name	varchar(50)	No	NOT NULL	UNIQUE
DateAndTimeOfFirstRecognition	TIMESTAMP	No	NOT NULL	
DateAndTimeOfLastRecognition	TIMESTAMP	No	NOT NULL	

В таблице 3.4.3 представлены описания столбцов для UNIDENTIFIED PERSONS.

Таблица 3.4.3 – UNIDENTIFIED PERSONS

Column name	Type	Key	NULL Status	Remarks
UnidentifiedPersonId	SERIAL	PRIMARY KEY	NOT NULL	
DateAndTimeOfFirstRecognition	TIMESTAMP	No	NOT NULL	
DateAndTimeOfLastRecognition	TIMESTAMP	No	NOT NULL	

В таблице 3.4.4 представлены описания столбцов для ACCESS DATA.

Таблица 3.4.4 – ACCESS DATA

Column name	Type	Key	NULL Status	Remarks
AccessDataId	SERIAL	PRIMARY KEY	NOT NULL	
Name	varchar(50)	No	NOT NULL	UNIQUE
Email	varchar(50)	No	NOT NULL	UNIQUE
Password	text	No	NOT NULL	
IdentifiedPersonId	INTEGER	FOREIGN KEY	NULL	

В таблице 3.4.5 представлены описания столбцов для TOTAL SYSTEM STATISTICS.

Таблица 3.4.5 – TOTAL SYSTEM STATISTICS

Column name	Type	Key	NULL Status	Remarks
TotalSystemStatisticsId	SERIAL	PRIMARY KEY	NOT NULL	
DateAndTime	TIMESTAMP	No	NOT NULL	
NumberOfImages	INTEGER	No	NOT NULL	
NumberOfIdentifiedPersons	INTEGER	No	NOT NULL	
NumberOfUnidentifiedPersons	INTEGER	No	NOT NULL	
TypeOfChangeId	INTEGER	FOREIGN KEY	NOT NULL	

В таблице 3.4.6 представлены описания столбцов для TYPES OF CHANGES.

Таблица 3.4.6 – TYPES OF CHANGES

Column name	Type	Key	NULL Status	Remarks
-------------	------	-----	-------------	---------

TypeOfChangeId	SERIAL	PRIMARY KEY	NOT NULL	
Description	varchar(150)	No	NOT NULL	UNIQUE

## **4. Создание приложения, осуществляющего распознавание лиц и использующего базу данных для хранения информации**

### **4.1. Пользовательский интерфейс приложения**

Сначала был создан пользовательский интерфейс приложения. Для его создания использовалась библиотека tkinter.

Пользовательский интерфейс состоит из трех окон – окна, в котором происходит регистрация и авторизация, окна, в котором происходит распознавание лиц и ввод имён, и окна, в котором происходит вывод информации из базы данных на экран.

При запуске приложения открывается окно регистрации и авторизации. В нём находятся поля для ввода имени, логина и пароля, кнопки «Зарегистрироваться», «Войти в систему распознавания лиц и ввода имён», «Войти в систему вывода информации», «Очистить dataset», «Выйти». Также есть строки для вывода сообщений о некорректном вводе.

При нажатии на кнопку «Войти в систему распознавания лиц и ввода имён» открывается окно распознавания лиц и ввода имён. Там можно увидеть видеопоток, получаемый с камеры компьютера. Также в этом окне находятся кнопки для голосового и текстового ввода имени человека, находящегося перед камерой и поля и кнопки для изменения имени.

Если из окна регистрации и авторизации нажать на кнопку «Войти в систему вывода информации», откроется окно, в котором происходит вывод информации из базы данных на экран. В нём находятся поля и кнопки для вывода информации об идентифицированном человеке по его имени, поле для вывода списка идентифицированных людей, которые были распознаны камерой за определённый промежуток времени, и таблица для вывода статистики по работе программы за определенный промежуток времени.

### **4.2. Создание таблиц**

Далее были написаны функции, обращающиеся к базе данных и создающие таблицы.

Была реализована функция, создающая таблицу IMAGES.



На рисунке 4.2.1 представлен код функции, создающей таблицу IMAGES.

```
def create_table_images():
    try:
        connection = psycopg2.connect(
            host=host,|
            user=user,
            password=password,
            database=db_name
        )
        connection.autocommit = True
        with connection.cursor() as cursor:
            cursor.execute(
                """CREATE TABLE if not exists IMAGES (
                    ImageId serial PRIMARY KEY,
                    FaceEncoding BYTEA NOT NULL,
                    IdentifiedPersonId integer NULL,
                    CONSTRAINT fk_image_identified_person FOREIGN KEY (IdentifiedPersonId) REFERENCES
                    IDENTIFIED_PERSONS (IdentifiedPersonId) ON DELETE CASCADE,
                    UnidentifiedPersonId integer NULL,
                    CONSTRAINT fk_image_unidentified_person FOREIGN KEY (UnidentifiedPersonId) REFERENCES
                    UNIDENTIFIED_PERSONS (UnidentifiedPersonId) ON DELETE CASCADE,
                    DateAndTime TIMESTAMP NOT NULL,
                    CONSTRAINT check_ids CHECK ((IdentifiedPersonId IS NOT NULL AND UnidentifiedPersonId IS NULL) or
                    (IdentifiedPersonId IS NULL AND UnidentifiedPersonId IS NOT NULL))
                );"""
            )
        print("CREATE TABLE if not exists IMAGES")
    except Exception as _ex:
        print(_ex)
    finally:
        if connection:
            connection.close()
```

Рисунок 4.2.1 – Код функции create\_table\_images()

В таблице IMAGES хранится информация об изображениях – идентификатор изображения, кодировка лица на изображении, время сохранения изображения. Для внешних ключей IdentifiedPersonId и UnidentifiedPersonId добавлены опции ON DELETE CASCADE, чтобы при удалении идентифицированного человека из таблицы IDENTIFIED PERSONS или при удалении неидентифицированного человека из таблицы UNIDENTIFIED PERSONS удалялись также все его фотографии. Также добавлено ограничение, проверяющее, что из внешних ключей IdentifiedPersonId и UnidentifiedPersonId один является NULL, а другой является NOT NULL. В таблице IMAGES хранится только описание изображений. Сами изображения будут храниться в папках с именами идентифицированных людей или, для не идентифицированных людей, с

именами, начинающимися на строку «person», за которой следует id не идентифицированного человека.

Далее была реализована функция, создающая таблицу IDENTIFIED PERSONS.

На рисунке 4.2.2 представлен код функции, создающей таблицу IDENTIFIED PERSONS.

```
def create_table_identified_persons():
    try:
        connection = psycopg2.connect(
            host=host,
            user=user,
            password=password,
            database=db_name
        )
        connection.autocommit = True
        with connection.cursor() as cursor:
            cursor.execute(
                """CREATE TABLE if not exists IDENTIFIED_PERSONS(
                    IdentifiedPersonId serial PRIMARY KEY,
                    Name varchar(50) UNIQUE NOT NULL,
                    DateAndTimeOfFirstRecognition TIMESTAMP NOT NULL,
                    DateAndTimeOfLastRecognition TIMESTAMP NOT NULL
                );"""
            )
            print("CREATE TABLE if not exists IDENTIFIED_PERSONS")
    except Exception as _ex:
        print(_ex)
    finally:
        if connection:
            connection.close()
```

Рисунок 4.2.2 – Код функции create\_table\_identified\_persons()

В таблице IDENTIFIED PERSONS хранится информация о идентифицированных людях, то есть людях, которых распознала камера и для которых затем было введено имя. В таблице IDENTIFIED PERSONS хранится

идентификатор человека, имя, дата и время первого и последнего распознавания – то есть первый момент, когда идентифицированный человек был распознан камерой и последний момент, когда идентифицированный человек был распознан камерой.

Также была реализована функция, создающая таблицу UNIDENTIFIED PERSONS.

На рисунке 4.2.3 представлен код функции, создающей таблицу UNIDENTIFIED PERSONS.

```
def create_table_unidentified_persons():
    try:
        connection = psycopg2.connect(
            host=host,
            user=user,
            password=password,
            database=db_name
        )
        connection.autocommit = True
        with connection.cursor() as cursor:
            cursor.execute(
                """CREATE TABLE if not exists UNIDENTIFIED_PERSONS(
                    UnidentifiedPersonId serial PRIMARY KEY,
                    DateAndTimeOfFirstRecognition TIMESTAMP NOT NULL,
                    DateAndTimeOfLastRecognition TIMESTAMP NOT NULL
                );"""
            )
            print("CREATE TABLE if not exists UNIDENTIFIED_PERSONS")
    except Exception as _ex:
        print(_ex)
    finally:
        if connection:
            connection.close()
```

Рисунок 4.2.3 – Код функции create\_table\_unidentified\_persons()

В таблице UNIDENTIFIED PERSONS хранится информация о не идентифицированных людях, то есть людях, которых распознала камера, но

которым не было присвоено имя. В таблице UNIDENTIFIED PERSONS хранится идентификатор человека, дата и время первого и последнего распознавания – то есть первый момент, когда не идентифицированный человек был распознан камерой и последний момент, когда не идентифицированный человек был распознан камерой.

Затем была реализована функция, создающая таблицу ACCESS DATA.

На рисунке 4.2.4 представлен код функции, создающей таблицу ACCESS DATA.

```
def create_table_access_data():
    try:
        connection = psycopg2.connect(
            host=host,
            user=user,
            password=password,
            database=db_name
        )
        connection.autocommit = True
        with connection.cursor() as cursor:
            cursor.execute(
                """CREATE TABLE if not exists ACCESS_DATA(
                    AccessDataId serial PRIMARY KEY,
                    Name varchar(50) UNIQUE NOT NULL,
                    Email varchar(50) UNIQUE NOT NULL,
                    Password text NOT NULL,
                    IdentifiedPersonId integer NULL,
                    CONSTRAINT fk_access_data_identified_persons FOREIGN KEY (IdentifiedPersonId)
                    REFERENCES IDENTIFIED_PERSONS (IdentifiedPersonId) ON DELETE SET NULL
                );"""
            )
            print("CREATE TABLE if not exists ACCESS_DATA")
    except Exception as _ex:
        print(_ex)
    finally:
        if connection:
            connection.close()
```

Рисунок 4.2.4 – Код функции create\_table\_access\_data()

В таблице ACCESS DATA хранится информация о данных для доступа – идентификатор данных для доступа, имя человека, которому принадлежат эти данные, адрес электронной почты и пароль. Пароли в базе данных будут зашифровываться при помощи специальной функции crypt(), предназначенной для хеширования паролей. Для внешнего ключа

IdentifiedPersonId добавлена опция ON DELETE SET NULL, чтобы при удалении идентифицированного человека из таблицы IDENTIFIED PERSONS в таблице ACCESS DATA внешний ключ IdentifiedPersonId становился равным NULL.

Далее была реализована функция, создающая таблицу TOTAL SYSTEM STATISTICS.

На рисунке 4.2.5 представлен код функции, создающей таблицу TOTAL SYSTEM STATISTICS.

```
def create_table_total_system_statistics():
    try:
        connection = psycopg2.connect(
            host=host,
            user=user,
            password=password,
            database=db_name
        )
        connection.autocommit = True
        with connection.cursor() as cursor:
            cursor.execute(
                """CREATE TABLE if not exists TOTAL_SYSTEM_STATISTICS(
                    TotalSystemStatisticsId serial PRIMARY KEY,
                    DateAndTime TIMESTAMP NOT NULL,
                    NumberOfImages integer NOT NULL,
                    NumberOfIdentifiedPersons integer NOT NULL,
                    NumberOfUnidentifiedPersons integer NOT NULL,
                    TypeOfChangeId INTEGER NOT NULL,
                    CONSTRAINT fk_total_system_statistics_type_of_changes FOREIGN KEY (TypeOfChangeId)
                    REFERENCES TYPES_OF_CHANGES (TypeOfChangeId) ON DELETE NO ACTION
                );"""
            )
            print("CREATE TABLE if not exists TOTAL_SYSTEM_STATISTICS")
    except Exception as _ex:
        print(_ex)
    finally:
        if connection:
            connection.close()
```

Рисунок 4.2.5 – Код функции create\_table\_total\_system\_statistics()

В таблице TOTAL SYSTEM STATISTICS хранится статистика, собираемая во время работы приложения. В момент, когда происходит какое-либо из изменений, хранящихся в таблице TYPES OF CHANGES, например, распознавание нового человека, ввод имени голосом или текстом, изменение имени, в таблицу TOTAL SYSTEM STATISTICS будет добавляться новая запись, содержащая дату и время изменения, число изображений, число

идентифицированных людей, число не идентифицированных людей на текущий момент и идентификатор типа изменения из таблицы TYPES OF CHANGES. Для внешнего ключа TypeOfChangeId добавлена опция ON DELETE NO ACTION, чтобы запретить удаление типов изменений, записи с которыми уже были сохранены в таблице TOTAL SYSTEM STATISTICS.

Затем была реализована функция, создающая таблицу TYPES OF CHANGES.

На рисунке 4.2.6 представлен код функции, создающей таблицу TYPES OF CHANGES.

```
def create_table_types_of_changes():
    try:
        connection = psycopg2.connect(
            host=host,
            user=user,
            password=password,
            database=db_name
        )
        connection.autocommit = True
        with connection.cursor() as cursor:
            cursor.execute(
                """CREATE TABLE if not exists TYPES_OF_CHANGES(
                    TypeOfChangeId serial PRIMARY KEY,
                    Description varchar(150) UNIQUE NOT NULL
                );"""
            )
            print("CREATE TABLE if not exists TYPES_OF_CHANGES")
    except Exception as _ex:
        print(_ex)
    finally:
        if connection:
            connection.close()
```

Рисунок 4.2.6 – Код функции create\_table\_types\_of\_changes()

В таблице TYPES OF CHANGES хранится информация о типах изменений – идентификатор типа изменения и описание изменения.

### 4.3. Регистрация и авторизация

Далее были реализованы функции регистрации и авторизации в системе. Они запускаются при нажатии кнопок «Зарегистрироваться», «Войти в систему распознавания лиц и ввода имён», «Войти в систему вывода информации». При регистрации проверяется, что поля не пустые, адрес электронной почты корректен, при этом проверка осуществляется при помощи регулярного выражения, длина пароля не менее 6 символов, и что повторно введенный пароль совпадает с начальным. Также проверяется, что в таблице ACCESS DATA ещё не присутствует такое имя или такой адрес электронной почты. Если данные для регистрации удовлетворяют всем условиям, они добавляются в таблицу ACCESS DATA. При этом пароль шифруется с помощью функции crypt().

На рисунке 4.3 представлен код функции, добавляющей новые данные для доступа в таблицу ACCESS DATA. Пароль зашифровывается.

```
def insert_into_access_data(name, email, passw):
    try:
        connection = psycopg2.connect(
            host=host,
            user=user,
            password=password,
            database=db_name
        )
        connection.autocommit = True
        with connection.cursor() as cursor:
            cursor.execute(
                """
                CREATE EXTENSION IF NOT EXISTS pgcrypto;
                INSERT INTO ACCESS_DATA(Name, Email, Password) VALUES (%s, %s, crypt(%s, gen_salt('md5')));
                """ , (name, email, passw))
            print("INSERT INTO ACCESS_DATA(Name, Email, Password) VALUES")
    except Exception as _ex:
        print(_ex)
    finally:
        if connection:
            connection.close()
```

Рисунок 4.3 – Код функции insert\_into\_access\_data(name, email, passw)

При авторизации происходит проверка адреса электронной почты и пароля с помощью обращения к таблице ACCESS DATA.

#### 4.4. Распознавание лиц и их идентификация

В окне распознавания лиц и ввода имен можно увидеть видеопоток, получаемый с камеры компьютера. Распознавание и идентификация лиц

осуществляется в функции `start()`. Сначала эта функция производит обнаружение лица в видеопотоке, получаемом с камеры компьютера. При обнаружении лица выделяется прямоугольником. Далее функция проходит по всем местоположениям и кодировкам лиц текущего кадра и сравнивает текущую кодировку лица с кодировками лиц на изображениях, сохраненных в таблицу `IMAGES`. Если кодировка лица совпадает с кодировкой лица изображения из таблицы `IMAGES`, то внизу прямоугольника пишется имя человека или, если человек не был идентифицирован, то строка «person», за которой следует `id` не идентифицированного человека. Если лиц с текущей кодировкой не было найдено в таблице `IMAGES`, то, значит, камера обнаружила нового человека. Информация о нём добавляется в таблицу `UNIDENTIFIED PERSONS`. Значениями `DateAndTimeOfFirstRecognition` и `DateAndTimeOfLastRecognition` становится текущий момент времени.

На рисунке 4.4.1 приведён код функции, добавляющей нового не идентифицированного человека.

```
def insert_into_unidentified_persons():
    try:
        connection = psycopg2.connect(
            host=host,
            user=user,
            password=password,
            database=db_name
        )
        connection.autocommit = True
        with connection.cursor() as cursor:
            cursor.execute("INSERT INTO UNIDENTIFIED_PERSONS(DateAndTimeOfFirstRecognition, DateAndTimeOfLastRecognition)"
                           " VALUES (NOW(), NOW());")
            print("INSERT INTO UNIDENTIFIED_PERSONS(DateAndTimeOfFirstRecognition, DateAndTimeOfLastRecognition) VALUES")
    except Exception as _ex:
        print(_ex)
    finally:
        if connection:
            connection.close()
```

Рисунок 4.4.1 – Код функции `insert_into_unidentified_persons()`

Каждые 3 секунды информация об изображении сохраняется в таблицу `IMAGES`, а само изображение сохраняется в папке `dataset` в папку с именем человека, которому принадлежит эта фотография.

На рисунке 4.4.2 приведена часть кода функции `start()`, в которой происходит сохранение информации об изображениях и сохранение самих



изображений, а также вызов функции добавления нового не идентифицированного человека.

```
if new_time - last_time >= 3:
    is_3_sec_passed = True
    face_img = image[top:bottom, left:right]
    face_img = cv2.cvtColor(face_img, cv2.COLOR_BGR2RGB)
    pil_img = Image.fromarray(face_img)
    if recognized_identified_person_id != 0:
        count = count_images_of_identified_person(recognized_identified_person_id) + 1
        pil_img.save("dataset/" + name + "/" + f"img_{count}.jpg")
        insert_identified_person_image_into_images(pickle.dumps(face_encoding), recognized_identified_person_id)
        update_date_and_time_of_last_recognition_in_table_identified_persons(recognized_identified_person_id)
    elif recognized_unidentified_person_id != 0:
        count = count_images_of_unidentified_person(recognized_unidentified_person_id) + 1
        pil_img.save("dataset/" + name + "/" + f"img_{count}.jpg")
        insert_unidentified_person_image_into_images(pickle.dumps(face_encoding), recognized_unidentified_person_id)
        update_date_and_time_of_last_recognition_in_table_unidentified_persons(recognized_unidentified_person_id)
    else:
        count = identified_person_number + unidentified_person_number + 1
        name = f"person_{count}"
        os.makedirs("dataset/" + name)
        pil_img.save("dataset/" + name + "/" + f"img_1.jpg")
        insert_into_unidentified_persons()
        insert_unidentified_person_image_into_images(pickle.dumps(face_encoding), count)
        insert_statistic(1)
```

Рисунок 4.4.2 – Часть кода функции start()

Также при повторном распознавании идентифицированного или не идентифицированного человека поле DateAndTimeOfLastRecognition обновляется – в него записывается текущее время. А при добавлении нового не идентифицированного человека в таблицу TOTAL SYSTEM STATISTIC добавляется новая запись, связанная с типом изменения, заключающимся в распознавании нового человека.

#### 4.5. Ввод и изменение имен

Также в окне распознавания лиц и изменения имен находятся кнопки для голосового и текстового ввода имени человека, находящегося перед камерой и поля и кнопки для изменения имени.

В глобальной переменной name\_now хранится имя человека, находящегося перед камерой в данный момент. Это имя можно поменять текстом или голосом. Перед изменением происходит проверка того, что перед камерой находится ровно один человек.

На рисунке 4.5 приведена функция, изменяющая имя человека, находящегося перед камерой, при помощи голосового ввода.

```

def listen():
    try:
        with speech_recognition.Microphone() as mic:
            sr.adjust_for_ambient_noise(source=mic, duration=0.5)
            audio = sr.listen(source=mic)
            query = sr.recognize_google(audio_data=audio, language='ru-RU').lower()
        return query
    except speech_recognition.UnknownValueError:
        return None

def change_name_from_speech_recognition():
    global name_now, number_of_persons_in_front_of_the_camera
    if number_of_persons_in_front_of_the_camera == 0:
        dialog_label.config(text='Перед камерой никого нет', width="50")
        dialog_label.update_idletasks()
        return
    if number_of_persons_in_front_of_the_camera > 1:
        dialog_label.config(text='Перед камерой находится более одного человека', width="50")
        dialog_label.update_idletasks()
        return
    dialog_label.config(text='Скажите имя', width="50")
    dialog_label.update_idletasks()
    query = listen()
    if query:
        change_name(name_now, query[0].upper() + query[1:])
        insert_statistic(4)
    else:
        dialog_label.config(text='Не получилось распознать имя', width="50")
        dialog_label.update_idletasks()

```

Рисунок 4.5 – Код функции change\_name\_from\_speech\_recognition()

Также можно изменить имя, введя текущее имя и новое имя. Если текущее имя является именем идентифицированного человека, и при этом новое имя не содержится в таблице IDENTIFIED PERSONS, то нужно просто изменить его имя на новое имя. А если текущее имя является строкой, начинающейся на «person», за которой следует id не идентифицированного человека, и при этом новое имя не содержится в таблице IDENTIFIED PERSONS, то нужно удалить из таблицы не идентифицированного человека по данному id и вставить его остальные данные и новое имя в таблицу идентифицированных людей.

#### 4.6. Вывод информации, хранящейся в базе данных

В окне вывода информации можно вывести информацию из базы данных на экран. Вверху окна находятся поля и кнопки для вывода информации об идентифицированном человеке по его имени. При вводе

имени идентифицированного человека и нажатии кнопки «Вывести информацию» выводится информация о дате и времени первого и последнего распознавания человека камерой.

Далее располагается поле для вывода списка идентифицированных людей, которые были распознаны камерой за определённый промежуток времени, и таблица для вывода статистики по работе программы за определённый промежуток времени. Для вывода этой информации, нужно ввести промежуток времени – дату и время начала или только дату начала (в этом случае время начала станет равным 00:00:00) и дату и время конца или только дату конца (в этом случае время конца станет равным 23:59:59). Далее при нажатии кнопки «Вывести список идентифицированных людей» будет выведен список идентифицированных людей, которые были распознаны камерой в данном промежутке времени. При нажатии кнопки «Вывести статистику» будут выведены записи из таблицы со статистикой за данный промежуток времени.

На рисунке 4.6 приведена функция, осуществляющая получение записей из таблицы со статистикой за данный промежуток времени.

```
def get_statistics_list_from_time_interval(start_time, end_time):
    try:
        connection = psycopg2.connect(
            host=host,
            user=user,
            password=password,
            database=db_name
        )
        connection.autocommit = True
        with connection.cursor() as cursor:
            cursor.execute(
                """
                SELECT DateAndTime, NumberOfImages, NumberOfIdentifiedPersons, NumberOfUnidentifiedPersons, TypeOfChangeId
                FROM TOTAL_SYSTEM_STATISTICS WHERE %s <= DateAndTime AND DateAndTime <= %s;
                """
                (start_time, end_time)
            )
            return cursor.fetchall()
    except Exception as _ex:
        print(_ex)
    finally:
        if connection:
            connection.close()
```

Рисунок 4.6 – Код функции  
get\_statistics\_list\_from\_time\_interval(start\_time, end\_time)

## 5. Результаты работы

### 5.1. Регистрация и авторизация

При запуске программы открывается окно регистрации и авторизации.

На рисунке 5.1.1 изображено окно регистрации и авторизации. В нём находятся поля для ввода имени, логина и пароля, кнопки «Зарегистрироваться», «Войти в систему распознавания лиц и ввода имён», «Войти в систему вывода информации», «Очистить dataset», «Выйти». Также есть строки для вывода сообщений о некорректном вводе.

The screenshot shows a window titled "Face recognition system with database". The window contains two main sections: "Регистрация" (Registration) and "Вход" (Login). The "Регистрация" section includes input fields for "Введите имя" (Enter name), "Введите адрес электронной почты" (Enter email address), "Введите пароль" (Enter password), and "Повторите пароль" (Repeat password), followed by a "Зарегистрироваться" (Register) button. The "Вход" section includes input fields for "Введите адрес электронной почты" (Enter email address) and "Введите пароль" (Enter password), followed by a "Войти в систему распознавания лиц и ввода имен" (Login to face recognition and name input system) button. Below the login section are buttons for "Войти в систему вывода информации" (Login to information output system), "Очистить базу данных" (Clear database), and "Выйти" (Exit).

Рисунок 5.1.1 – Окно регистрации и авторизации

Для регистрации нужно ввести имя, адрес электронной почты и пароль. Ввод будет проверен на корректность, и при успешной регистрации должна появиться надпись «Регистрация прошла успешно!».

На рисунке 5.1.2 изображен ввод имени, адреса электронной почты и пароля для регистрации.

The screenshot shows a web application window titled "Face recognition system with database". The main content area has a light gray background. At the top, the word "Регистрация" (Registration) is displayed in bold. Below it, the text "Введите имя" (Enter name) is followed by a text input field containing "Sam". Then, "Введите адрес электронной почты" (Enter email address) is followed by a text input field containing "sam@yandex.ru". Next, "Введите пароль" (Enter password) is followed by a password input field with "\*\*\*\*\*". Below that, "Повторите пароль" (Repeat password) is followed by another password input field with "\*\*\*\*\*". A button labeled "Зарегистрироваться" (Register) is positioned below the password fields. Underneath the button, the text "Регистрация прошла успешно!" (Registration was successful!) is displayed. The next section is titled "Вход" (Login) in bold. It contains two text input fields: the first is labeled "Введите адрес электронной почты" (Enter email address) and the second is labeled "Введите пароль" (Enter password). Below these fields are three buttons: "Войти в систему распознавания лиц и ввода имен" (Login to face recognition and name input system), "Войти в систему вывода информации" (Login to information output system), and "Очистить базу данных" (Clear database). At the very bottom, there is a button labeled "Выйти" (Logout).

Рисунок 5.1.2 – Ввод данных для регистрации

На рисунке 5.1.3 изображен ввод адреса электронной почты и пароля для авторизации.

The screenshot shows a window titled "Face recognition system with database" with standard Windows window controls. The interface is divided into two main sections: "Регистрация" (Registration) and "Вход" (Login). The registration section includes fields for "Введите имя" (Enter name), "Введите адрес электронной почты" (Enter email address), "Введите пароль" (Enter password), and "Повторите пароль" (Repeat password), followed by a "Зарегистрироваться" (Register) button. The login section includes fields for "Введите адрес электронной почты" (Enter email address) with the example "sam@yandex.ru" and "Введите пароль" (Enter password) with masked characters "\*\*\*\*\*". Below the login section are three buttons: "Войти в систему распознавания лиц и ввода имен" (Login to face recognition and name input system), "Войти в систему вывода информации" (Login to information output system), and "Очистить базу данных" (Clear database). At the bottom center is a "Выйти" (Logout) button.

**Регистрация**

Введите имя

Введите адрес электронной почты

Введите пароль

Повторите пароль

Зарегистрироваться

**Вход**

Введите адрес электронной почты

sam@yandex.ru

Введите пароль

\*\*\*\*\*

Войти в систему распознавания лиц и ввода имен

Войти в систему вывода информации

Очистить базу данных

Выйти

Рисунок 5.1.3 – Ввод данных для авторизации

## 5.2. Распознавание лиц и ввод имён

В окне распознавания лиц и ввода имён можно увидеть видеопоток, получаемый с камеры компьютера. Также в этом окне находятся кнопки для голосового и текстового ввода имени человека, находящегося перед камерой и поля и кнопки для изменения имени.

На рисунке 5.2.1 изображено окно распознавания лиц и ввода имен.

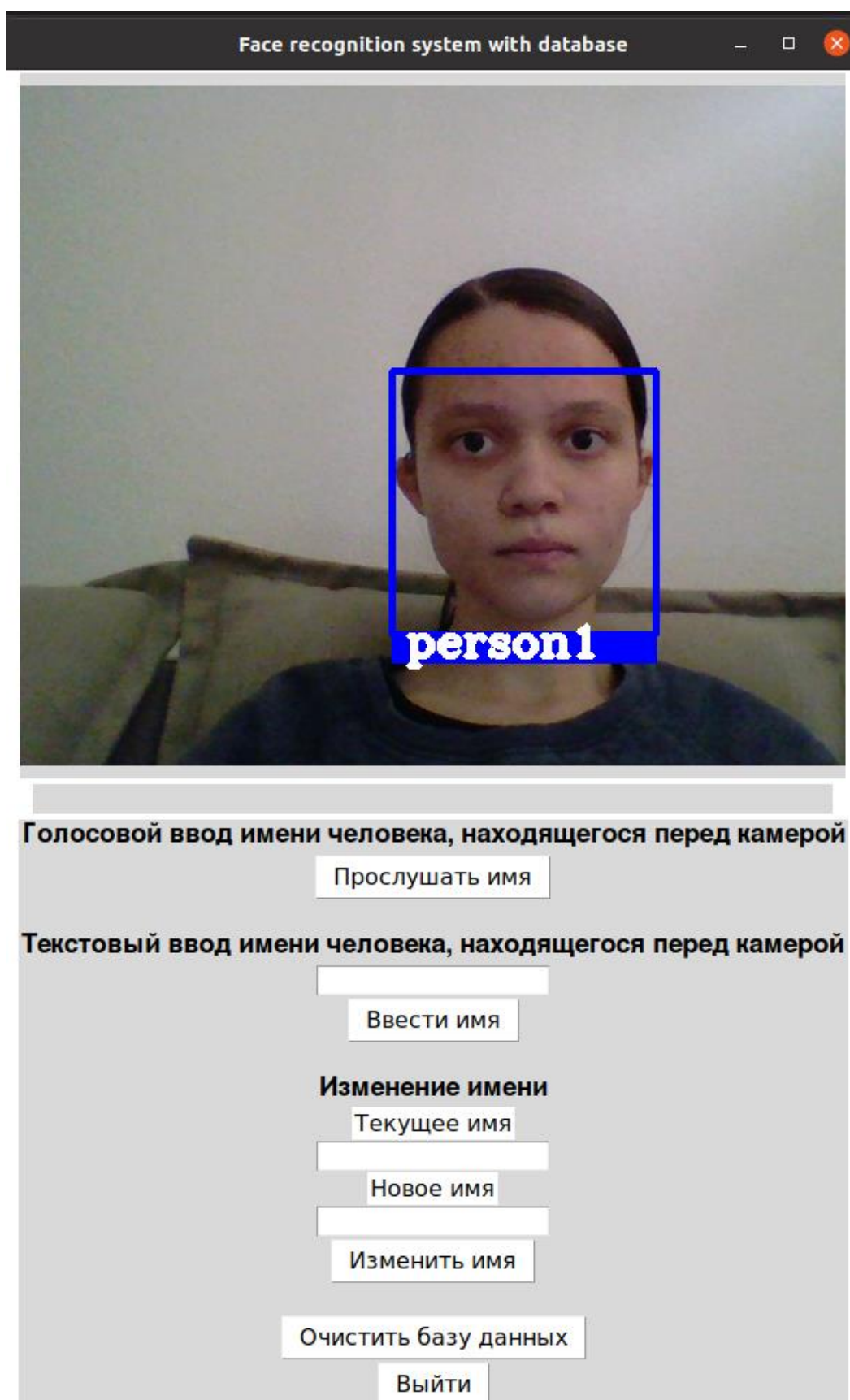


Рисунок 5.2.1 – Окно распознавания лиц и ввода имён

Для голосового ввода имени человека, находящегося перед камерой, нужно нажать кнопку «Прослушать имя» и произнести имя.



На рисунке 5.2.2 изображен ввод имени голосом.

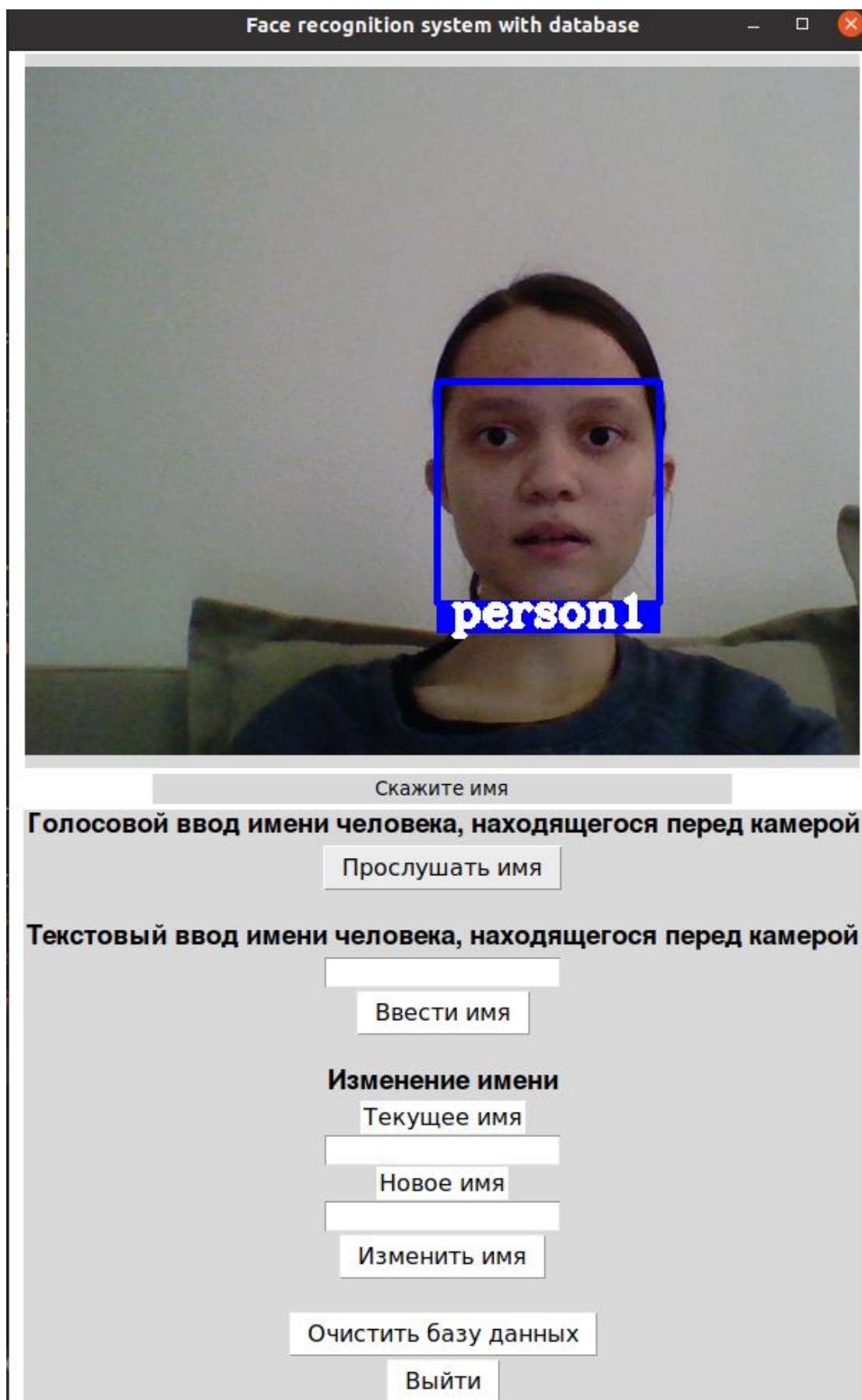


Рисунок 5.2.2 – Ввод имени голосом



На рисунке 5.2.3 изображен человек с введенным именем.

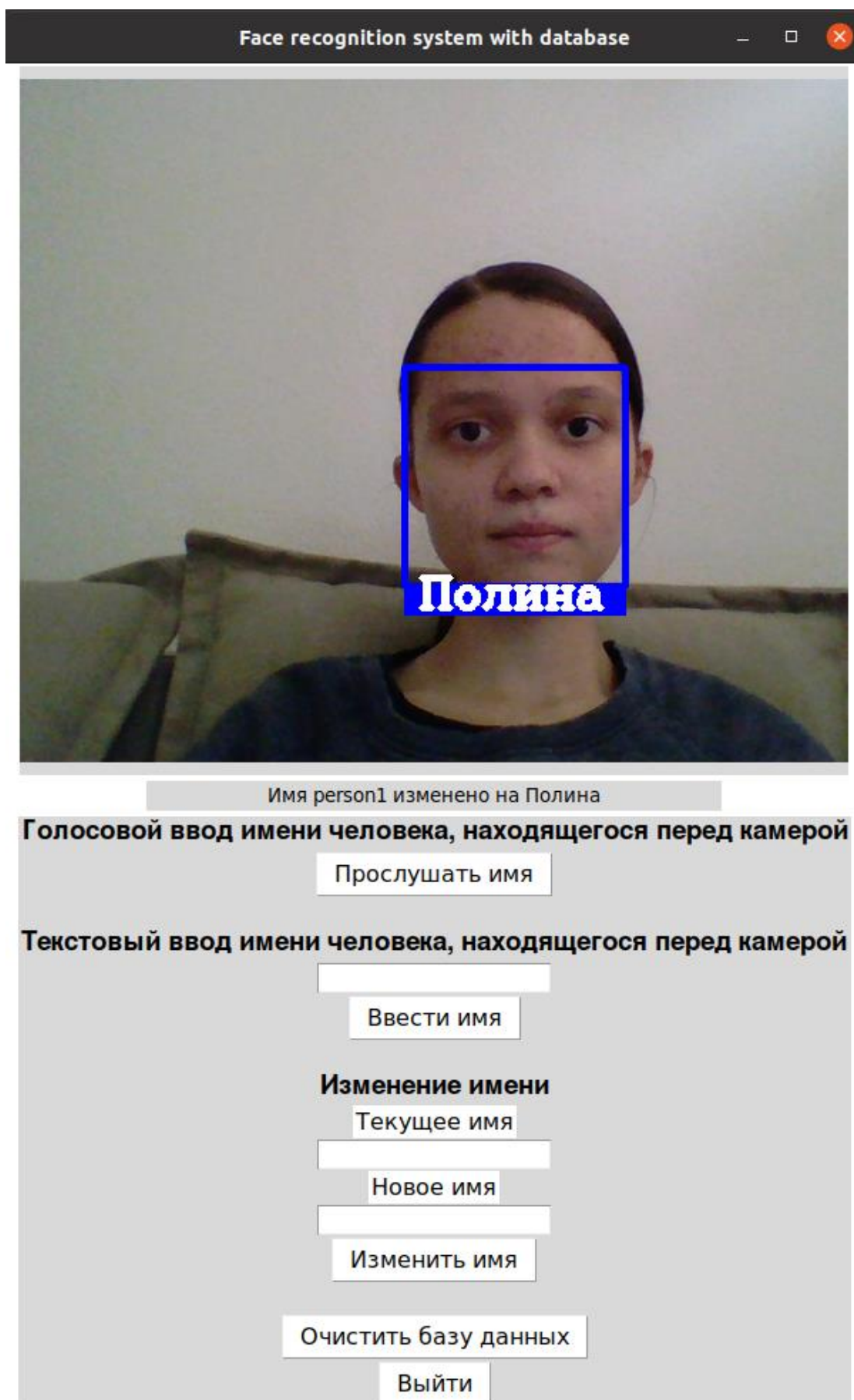


Рисунок 5.2.3 – Человек с введенным именем

На рисунке 5.2.4 изображены два не идентифицированных человека.

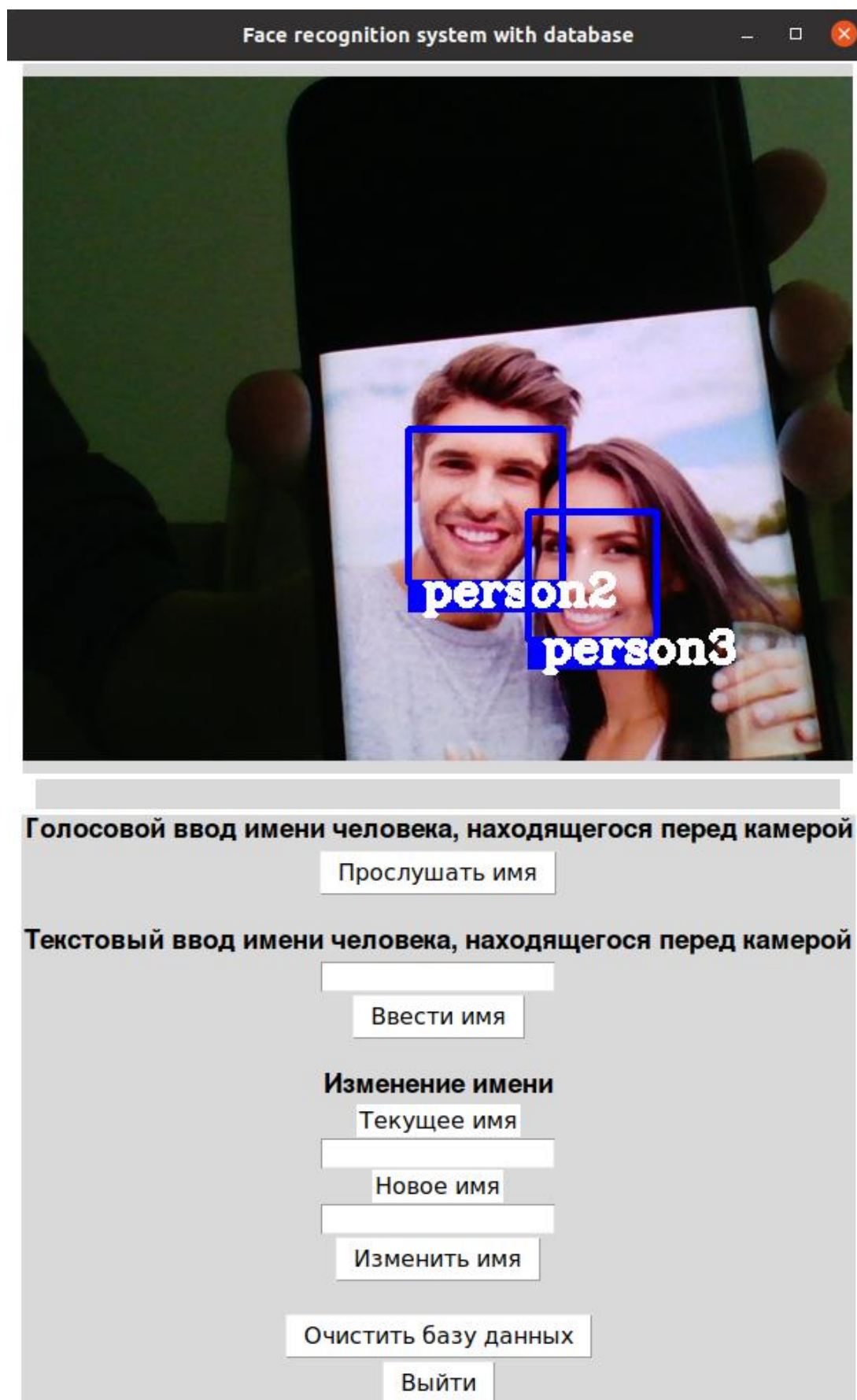


Рисунок 5.2.4 – Распознавание не идентифицированных людей

На рисунке 5.2.5 изображен не идентифицированный человек.

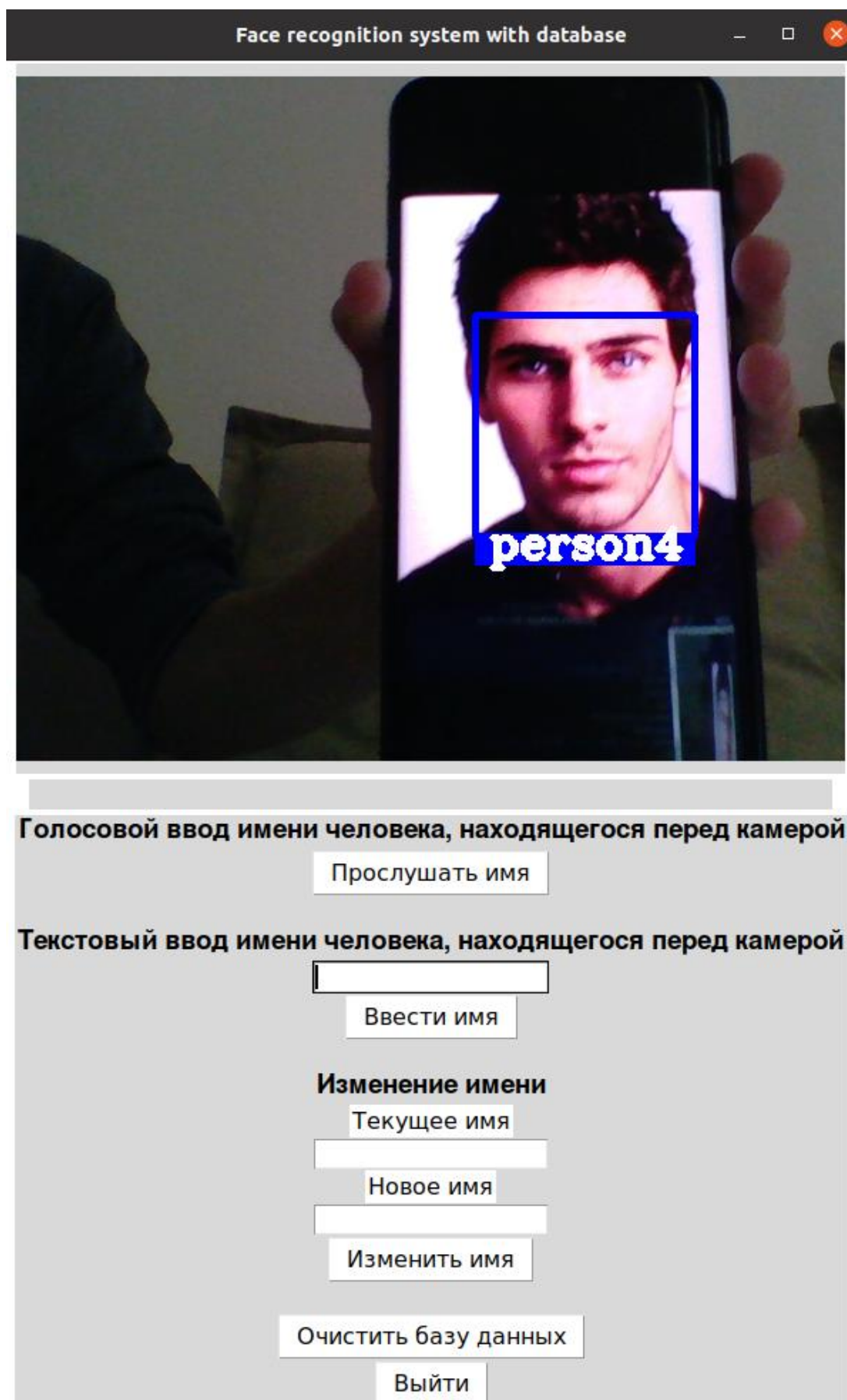


Рисунок 5.2.5 – Распознавание не идентифицированного человека

Для текстового ввода имени человека, находящегося перед камерой, нужно ввести новое имя и нажать кнопку «Ввести имя».

На рисунке 5.2.6 изображен человек с введенным текстом именем.

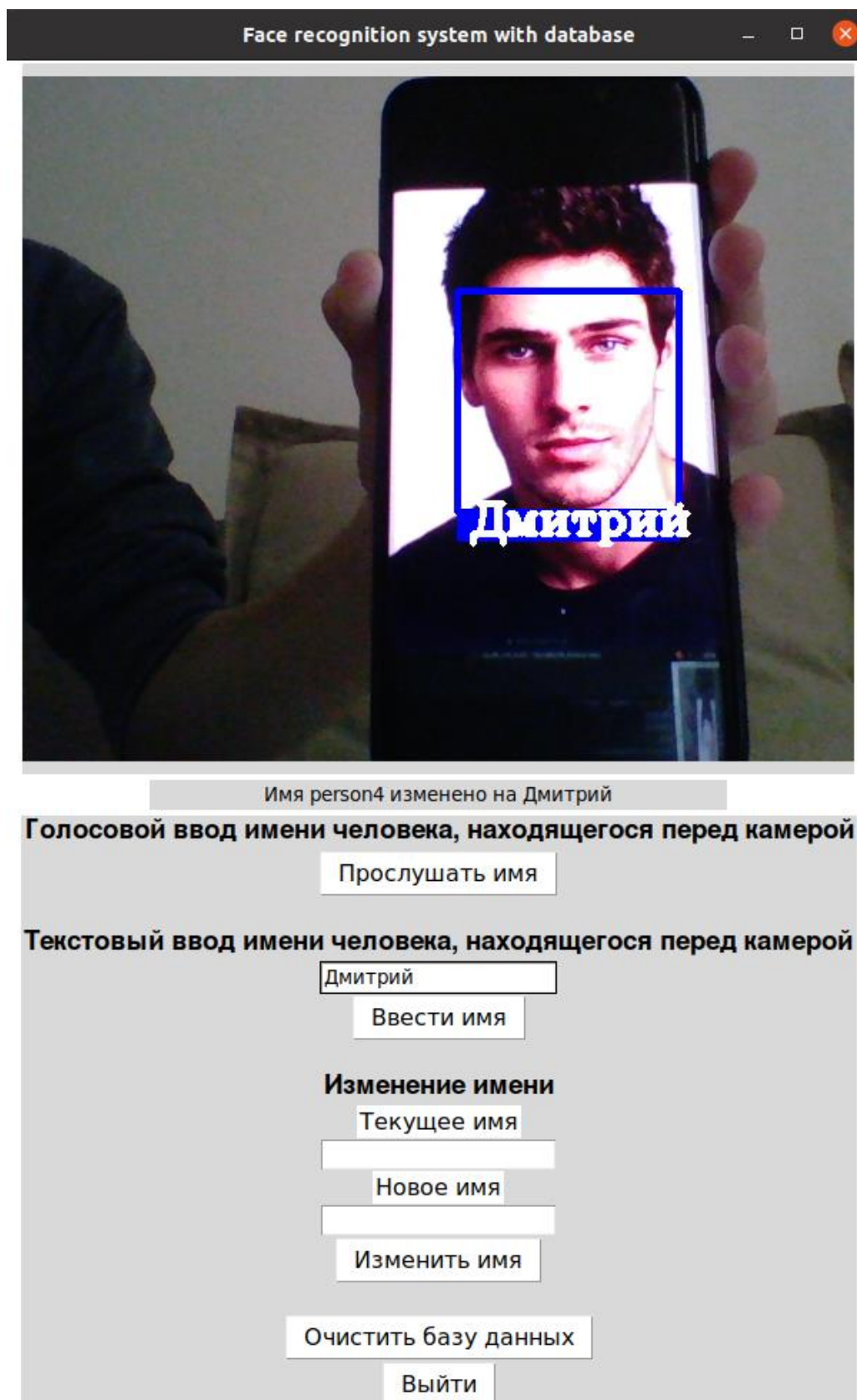


Рисунок 5.2.6 – Человек с введенным текстом именем



### 5.3. Содержимое папки dataset

На рисунке 5.3.1 изображены папки, находящиеся в папке dataset.

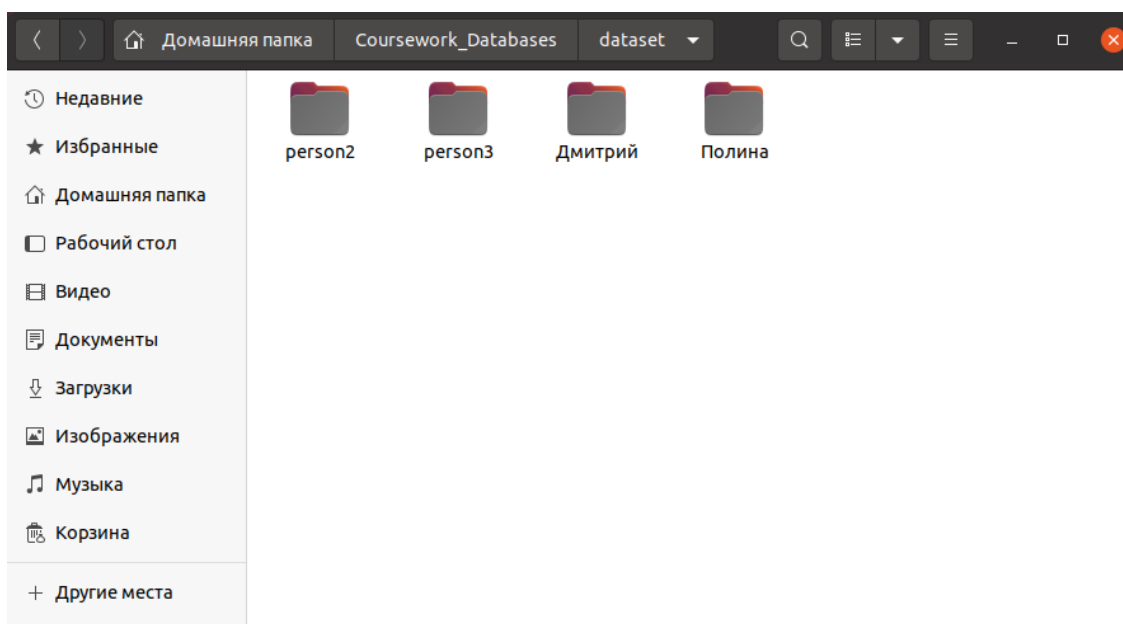


Рисунок 5.3.1 – Содержимое папки dataset

На рисунке 5.3.2 изображены сохраненные фотографии Полины.

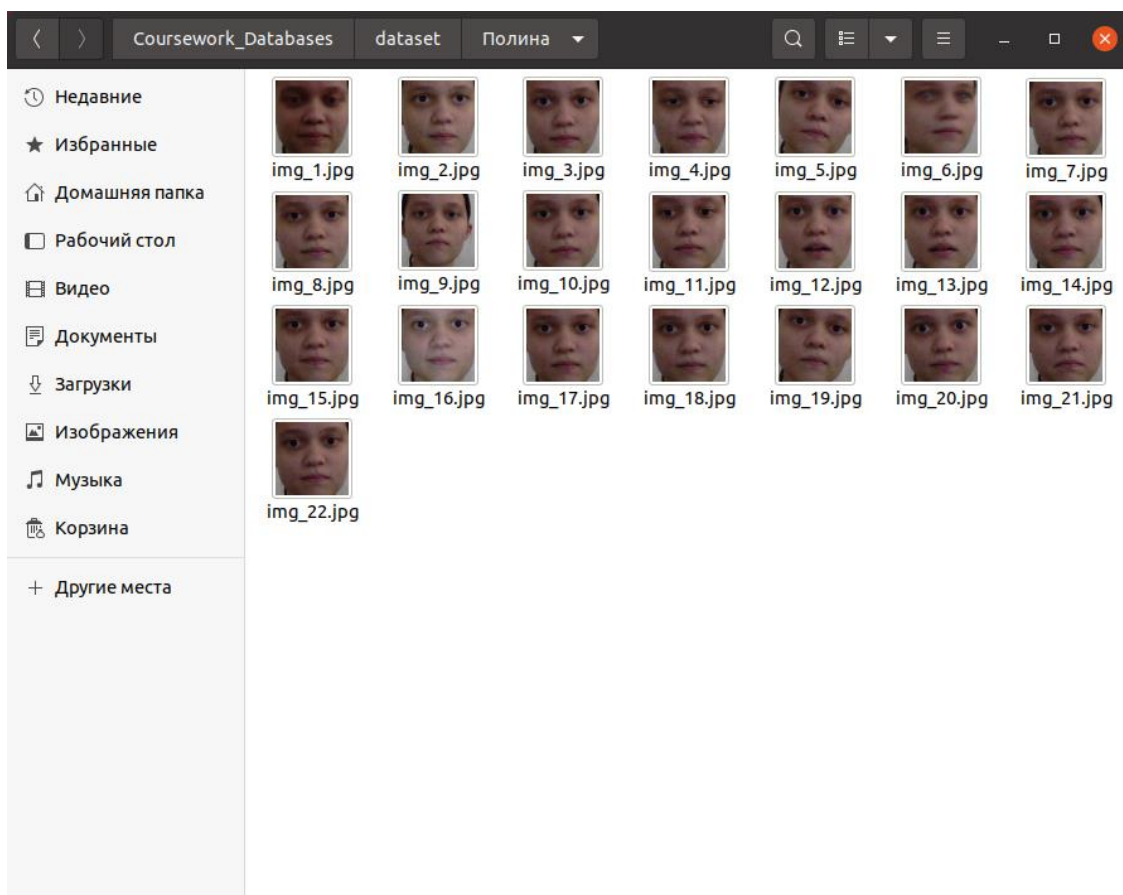


Рисунок 5.3.2 – Сохраненные фотографии Полины

На рисунке 5.3.3 изображены сохраненные фотографии Дмитрия.

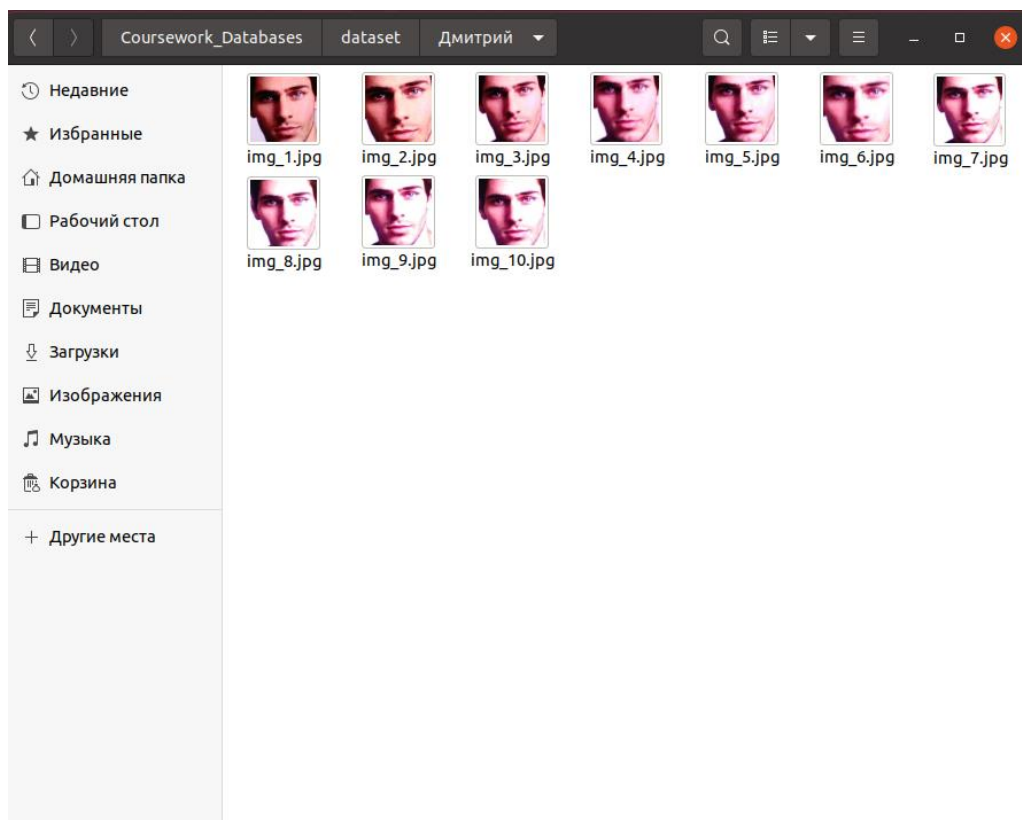


Рисунок 5.3.3 – Сохраненные фотографии Дмитрия

На рисунке 5.3.4 изображены сохраненные фотографии person2.

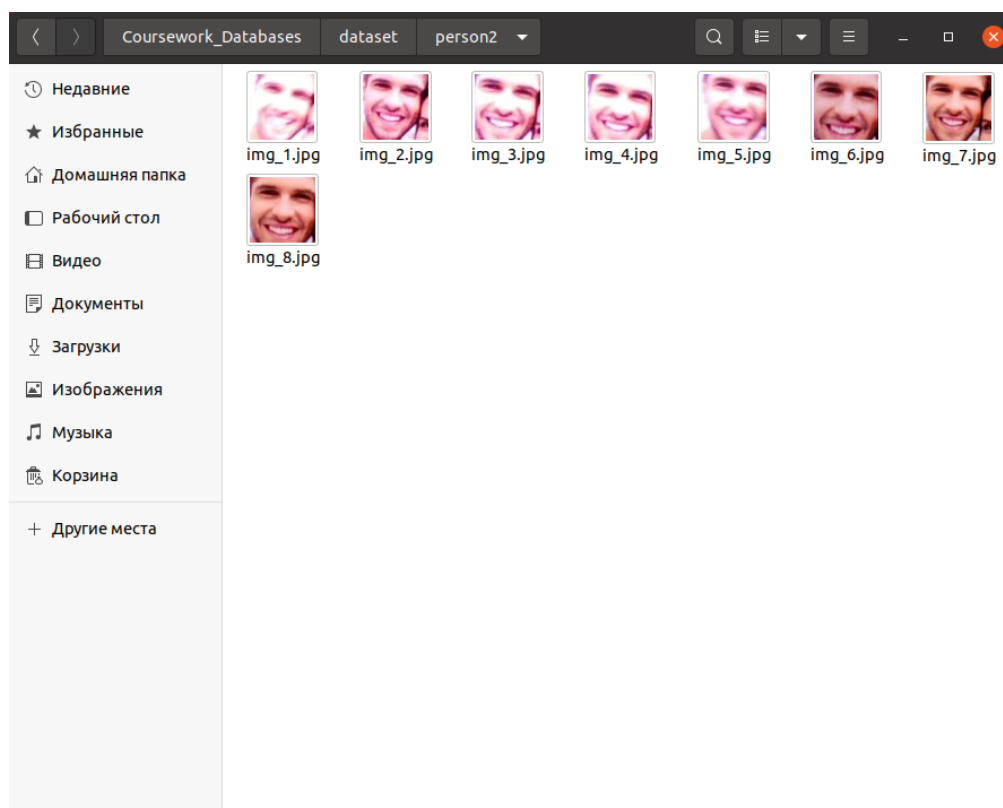


Рисунок 5.3.4 – Сохраненные фотографии person2

На рисунке 5.3.5 изображены сохраненные фотографии person3.

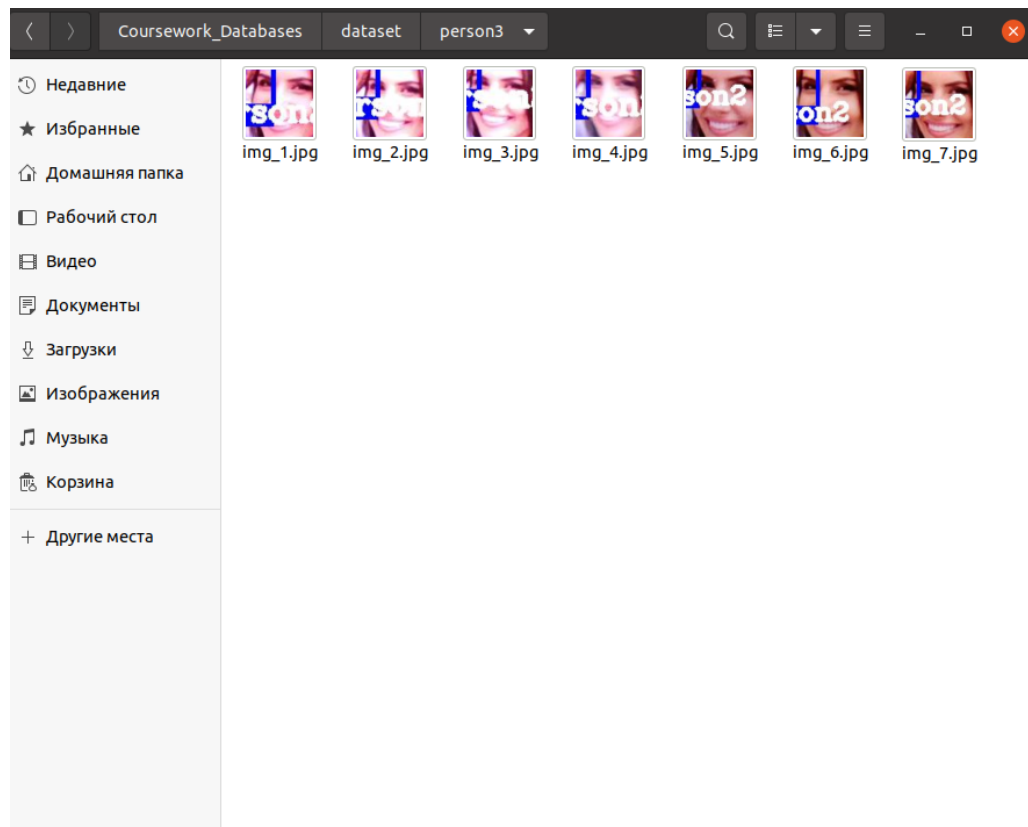


Рисунок 5.3.5 – Сохраненные фотографии person3

## 5.4. Вывод информации из базы данных

На рисунке 5.4.1 изображен ввод адреса электронной почты и пароля для авторизации.

Рисунок 5.4.1 – Ввод данных для авторизации

На рисунке 5.4.2 изображено окно вывода информации из базы данных.

The screenshot shows a window titled "Face recognition" with a dark header bar. The main content area has a light gray background. At the top, it says "Вывод информации об идентифицированном человеке" (Output information about the identified person). Below this is a text input field labeled "Введите имя" (Enter name) and a button labeled "Вывести информацию" (Output information). The next section is titled "Вывод списка идентифицированных людей, которые были распознаны камерой за определенный промежуток времени / Вывод статистики за определенный промежуток времени" (Output list of identified people who were recognized by the camera for a certain period of time / Output statistics for a certain period of time). This section contains two date and time input fields, both labeled "Введите дату и время начала в формате YYYY-MM-DD HH:MM:SS (время указывать не обязательно)" (Enter the start date and time in the format YYYY-MM-DD HH:MM:SS (time is not necessarily indicated)). Below these are two buttons: "Вывести список идентифицированных людей" (Output list of identified people) and "Вывести статистику" (Output statistics). At the bottom of the window are two more buttons: "Очистить базу данных" (Clear database) and "Выйти" (Exit).

Рисунок 5.4.2 – Окно вывода информации из базы данных

Для вывода даты и времени первого и последнего распознавания идентифицированного человека нужно ввести его имя и нажать кнопку «Вывести информацию».

На рисунке 5.4.3 изображен вывод даты и времени первого и последнего распознавания Полины.

The screenshot shows the same "Face recognition" window, but now it displays the output of the "Вывести информацию" button. The text input field "Введите имя" now contains the name "Полина". Below the input fields, the application has output the following text: "Дата и время первого распознавания: 2023-06-27 14:45:02.369266" and "Дата и время последнего распознавания: 2023-06-27 14:46:11.977576". The rest of the interface, including the other input fields and buttons, remains the same as in the previous screenshot.

Рисунок 5.4.3 – Вывод информации о Полине



На рисунке 5.4.4 изображен вывод даты и времени первого и последнего распознавания Дмитрия.

The screenshot shows a window titled "Face recognition" with a dark header. The main content area has a light gray background. At the top, it says "Вывод информации об идентифицированном человеке". Below this, there is a text input field with "Дмитрий" entered, followed by a "Вывести информацию" button. The output shows two lines of text: "Дата и время первого распознавания: 2023-06-27 14:49:29.706660" and "Дата и время последнего распознавания: 2023-06-27 14:49:58.704707". Below this, there is a section titled "Вывод списка идентифицированных людей, которые были распознаны камерой за определенный промежуток времени / Вывод статистики за определенный промежуток времени". This section contains two text input fields for start and end dates/times, both with empty values, and a "Вывести список идентифицированных людей" button. Below the input fields, there is a "Вывести статистику" button. At the bottom of the window, there are two buttons: "Очистить базу данных" and "Выйти".

Рисунок 5.4.4 – Вывод информации о Дмитрии

Для вывода списка идентифицированных людей, которые были распознаны камерой за определенный промежуток времени, нужно ввести дату и время начала и конца этого промежутка и нажать кнопку «Вывести список идентифицированных людей».

На рисунке 5.4.5 изображен вывод списка идентифицированных людей, распознанных 2023-06-27 числа.

The screenshot shows the same "Face recognition" window. The "Вывести информацию" button is still visible. Below it, the "Вывести список идентифицированных людей" button is now active, and the output shows a list of names: "Полина, Дмитрий". The "Вывести статистику" button is also visible. The "Очистить базу данных" and "Выйти" buttons are at the bottom.

Рисунок 5.4.5 – Вывод списка идентифицированных людей

Для вывода статистики за определенный промежуток времени, нужно ввести дату и время начала и конца этого промежутка и нажать кнопку «Вывести статистику».

На рисунке 5.4.6 изображена таблица статистики за 2023-06-27 число.

Вывести статистику

Дата и время	Число изображений	Число идентифицированных	Число неидентифицированных	Тип изменения
2023-06-27 14:45:02.465704	1	0	1	1
2023-06-27 14:45:42.525429	12	1	0	4
2023-06-27 14:48:18.474263	23	1	1	1
2023-06-27 14:48:22.001224	25	1	2	1
2023-06-27 14:49:29.761302	38	1	3	1
2023-06-27 14:49:46.458354	43	2	2	3

Очистить базу данных

Выйти

Рисунок 5.4.6 – Вывод таблицы статистики

На рисунке 5.4.7 изображен вывод списка людей, распознанных в промежутке времени с 2023-06-27 14:45:00 до 2023-06-27 14:49:10.

Вывести список идентифицированных людей

Полина

Вывести статистику

Очистить базу данных

Выйти

Рисунок 5.4.7 – Вывод списка идентифицированных людей

На рисунке 5.4.8 изображен вывод таблицы статистик, полученных в промежутке времени с 2023-06-27 14:45:00 до 2023-06-27 14:49:10.

Face recognition

Вывод информации об идентифицированном человеке

Введите имя

Вывести информацию

Вывод списка идентифицированных людей, которые были распознаны камерой за определенный промежуток времени / Вывод статистики за определенный промежуток времени

Введите дату и время начала в формате YYYY-MM-DD HH:MM:SS (время указывать не обязательно)

2023-06-27 14:45:00

Введите дату и время конца в формате YYYY-MM-DD HH:MM:SS (время указывать не обязательно)

2023-06-27 14:49:10

Вывести список идентифицированных людей

Вывести статистику

Дата и время	Число изображений	Число идентифицированных	Число неидентифицированных	Тип изменения
2023-06-27 14:45:02.465704	1	0	1	1
2023-06-27 14:45:42.525429	12	1	0	4
2023-06-27 14:48:18.474263	23	1	1	1
2023-06-27 14:48:22.001224	25	1	2	1

Очистить базу данных

Выйти

Рисунок 5.4.8 – Вывод таблицы статистики

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсовой работы:

- 1) была разработана реляционная модель базы данных для информационной системы распознавания лиц;
- 2) разработанное ранее приложение для распознавания лиц было доработано с использованием базы данных для хранения информации и конфигурационных параметров;
- 3) был реализован вывод информации из базы данных;
- 4) было проведено тестирование приложения.

## СПИСОК ЛИТЕРАТУРЫ

1. Документация PostgreSQL: сайт // <https://www.postgresql.org/docs/> (дата обращения 31.05.23).
2. Сайт // <https://www.kaspersky.ru/resource-center/definitions/what-is-facial-recognition> (дата обращения 31.05.23).
3. Документация библиотеки face\_recognition: сайт // <https://face-recognition.readthedocs.io/en/latest/index.html#> (дата обращения 31.05.23).
4. Описание библиотеки speech\_recognition: сайт // <https://pypi.org/project/SpeechRecognition/> (дата обращения 31.05.23).
5. Электронный ресурс // [https://e-learning.bmstu.ru/iu6/pluginfile.php/18676/mod\\_resource/content/1/PostgreSQL\\_Osnovy - Richard Stounz.pdf](https://e-learning.bmstu.ru/iu6/pluginfile.php/18676/mod_resource/content/1/PostgreSQL_Osnovy_-_Richard_Stounz.pdf) (дата обращения 31.05.23).