АРХИТЕКТУРА И КОМПОНЕНТЫ РАЗРАБАТЫВАЕМОЙ СИСТЕМЫ БИОМЕТРИЧЕСКОЙ АУТЕНТИФИКАЦИИ В ОФИСНОЙ СРЕДЕ

Д.С. Калининский, *магистрант*
Московский технический университет связи и информатики
(Россия, г. Москва)

DOI:10.24412/2500-1000-2023-7-1-192-196

Аннотация. *В данной статье проводится детальный обзор архитектурных и технологических аспектов, которые лежат в основе процесса разработки программной системы биометрической аутентификации. В статье освещается ключевые элементы такой системы, включая модули аутентификации, контроля доступа и мониторинга, при этом акцентируя внимание на их важности для обеспечения повышенного уровня безопасности в офисной среде. Представлен подробный анализ каждой из этих компонентов, рассмотрены их функциональные обязанности и взаимодействие между собой, чтобы предоставить читателям глубокое понимание принципов работы системы. Кроме того, детально рассматриваются выбранные для разработки системы технологии и инструменты, включая языки программирования C# и Python, библиотеки OpenCV, dlib, TensorFlow и Flask.*

Ключевые слова: *биометрическая аутентификация, архитектура ПО, система контроля доступа, мониторинг, технологический стек, Python, C#, OpenCV, dlib, TensorFlow, Flask.*

Безопасность в современном офисном пространстве становится все более актуальной темой с учетом угроз, возникающих из-за быстрого развития технологий и появления новых каналов кибератак [1-3]. Эффективные системы защиты и управления доступом играют ключевую роль в обеспечении безопасной рабочей среды. Программные системы биометрической аутентификации представляют собой один из передовых подходов к обеспечению защиты и контроля доступа [4].

Целью данной статьи является представление подробного анализа архитектуры и выбранных технологий в разрабатываемой программной системе биометрической аутентификации. Работа сосредоточена на исследовании различных компонентов такой системы, включая модули аутентификации, контроля доступа и мониторинга, с акцентом на их роль в обеспечении безопасности в офисной среде [5-6].

Разрабатываемая система стремится создать надежную и удобную в использовании среду, в которой биометрические данные используются для подтверждения идентичности пользователя, обеспечивая таким образом эффективный контроль доступа и высокий уровень защиты [7].

Описание архитектуры и компонентов программного решения

Архитектура программного решения для системы аутентификации построена на основе клиентской архитектуры. В такой архитектуре клиентское приложение обрабатывает запросы на аутентификацию и хранит информацию о пользователях и их биометрических данных. Далее на рисунке 20 представлена блок схема работы программы.

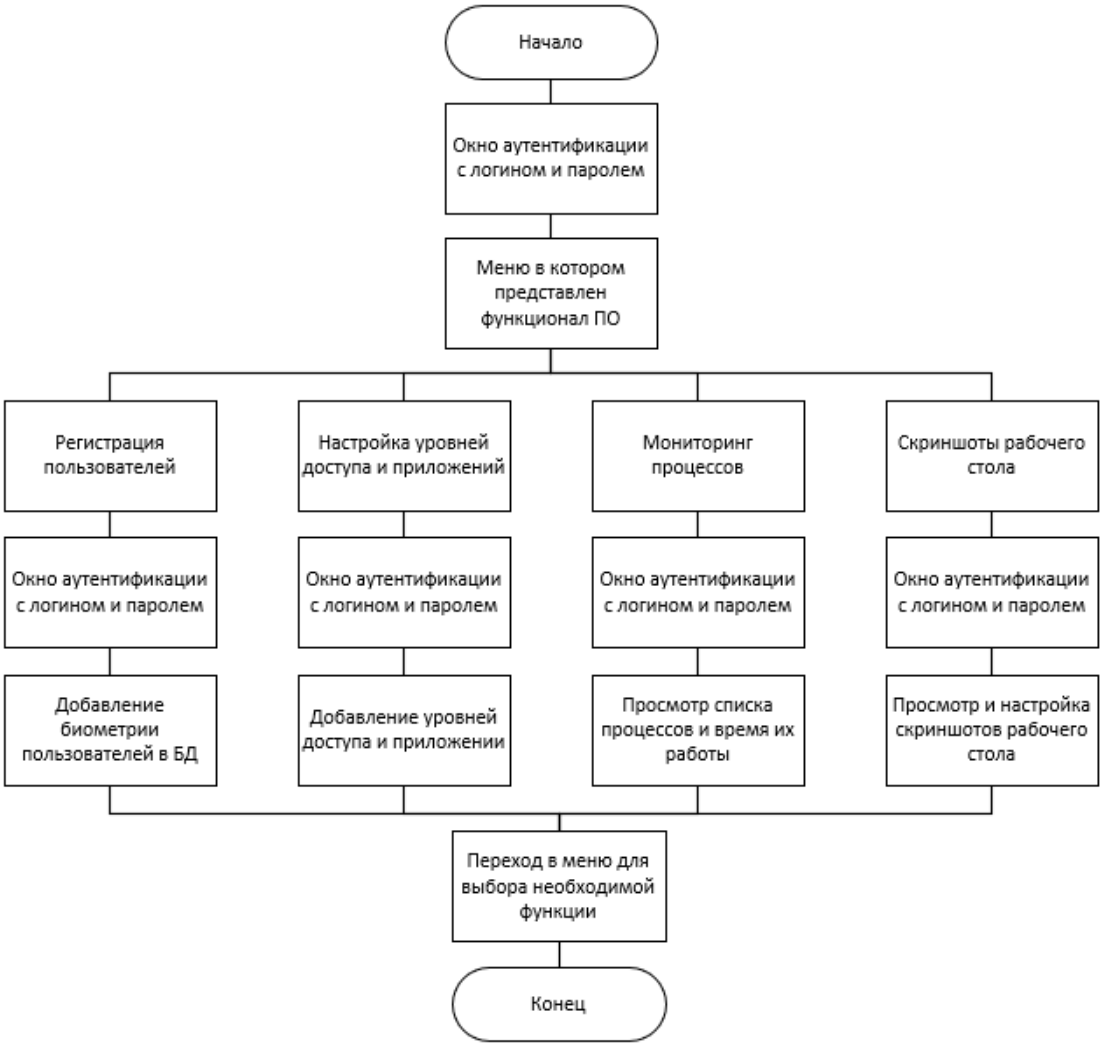

Рис. 1. Архитектура программного обеспечения

Ниже представлено описание основных компонентов программного решения:

1. Клиентское приложение (Windows Forms):

- Форма аутентификации: Визуальный интерфейс для ввода биометрических данных пользователя (например, с помощью камеры для распознавания лица).

- Форма доступа к приложениям: Визуальный интерфейс для выбора приложения и предоставления доступа на основе уровней доступа.

- Форма мониторинга процессов на рабочем месте: Визуальный интерфейс для отображения процессов приложении и время их работы.

- Форма мониторинга скриншотов экрана: Визуальный интерфейс для просмотра скриншотов и настройки их работы.

2. Биометрический модуль:

Модуль захвата биометрических данных: Отвечает за захват и обработку биометрических данных, таких как изображение лица, с использованием соответствующих биометрических сенсоров или камер.

3. База данных (или файл):

Хранит информацию о зарегистрированных пользователях, их биометрических данных и уровнях доступа к приложениям.

Программа будет использовать клиентское приложение для аутентификации пользователей на основе биометрических данных и предоставления доступа к приложениям в соответствии с заданными уровнями доступа. Приложение будет взаимодействовать с биометрическим модулем для получения и обработки биометрических данных, а также с базой данных (или файлом) для хранения информации о пользователях и их уровнях доступа.

Для реализации биометрии был написан Python-скрипт (face_recognition()), который использует каскадный классификатор Haar для обнаружения лиц на видеопотоке с веб-камеры и отображает прямоугольники вокруг обнаруженных лиц. Можно настроить параметры классификатора и веб-камеры в соответствии с вашими требованиями. В этом ПО функция authenticate() получает два аргумента: face_roi (область интереса лица) и known_faces (список известных лиц). Функция проходит по каждому известному лицу и сравнивает его с переданной областью лица. Если есть совпадение, функция возвращает True, иначе возвращает False. Данная функция для выполнения кода имеет предобученные модели, такое как shape_predictor_68_face_landmarks.dat и dlib_face_recognition_resnet_model_v1.dat. Они загружены с официального сайта dlib. Также для работы алгоритма, установилены необходимые зависимости, такие как dlib и scipy.

Чтобы работал алгоритм биометрии ему необходима база данных, которая хранит информацию о пользователях и их биометрию для того, чтобы добавлять пользователей используется второй Python-скрипт (known_faces()), он представляет биометрию работников компании в виде кода (рис. 2).

```
known_faces = [

    # Лицевые особенности для первого известного лица

    [0.123, 0.456, 0.789, ...],

    # Лицевые особенности для второго известного лица

    [0.987, 0.654, 0.321, ...],

    # и так далее...

]
```

Рис. 2. Пример биометрической базы данных алгоритма

Применяемые технологические решения и инструментарий

Следующий набор технологий и инструментов был применён в процессе создания программной системы для биометрической идентификации [8-10]:

1. Программирование на C# – этот язык был выбран в качестве основного для создания функций и интерфейса программного продукта.

2. Программирование на Python – этот язык применялся для реализации алгоритмов биометрической идентификации и взаимодействия с биометрическими устройствами.

3. Библиотека OpenCV – используется для манипуляций с изображениями и видеоматериалами, включая обработку и анализ биометрической информации. OpenCV предлагает множество функций и алгоритмов для работы с изображениями, таких как определение контуров, выделение объектов, фильтрация и сегментация.

4. Библиотека dlib – используется для обработки и анализа изображений и видео, включая применение моделей машинного обучения. Dlib предоставляет сильные инструменты для распознавания лиц и работы с разными атрибутами лица.

5. Библиотека TensorFlow – применяется для обучения моделей машинного обучения на основе биометрических данных. TensorFlow предлагает мощные инструменты для обучения и оптимизации глубоких нейронных сетей, а также для обработки и анализа больших объемов данных.

6. Библиотека Flask – применяется для создания веб-приложения биометрической

идентификации. Flask позволяет создавать простые и масштабируемые веб-приложения, способные работать с различными типами биометрических данных.

7. Среда разработки PyCharm – используется для создания, отладки и тестирования программного решения для биометрической идентификации. PyCharm предлагает широкий набор инструментов для разработки приложений на Python, включая автодополнение, отладчик, тестовые инструменты и многое другое.

8. Биометрическое оборудование – использовалось для сбора биометрических данных в ходе тестирования программного продукта. Использовались камеры для съёмки лиц и сканеры отпечатков пальцев.

Заключение

Была описана структура разрабатываемой программной системы и ее элементы. Основные модули и функциональные части, необходимые для поддержания безопасности в рабочем офисе, были уточнены. Это включает модули для аутентификации, управления доступом, мониторинга и обеспечения безопасности.

Кроме того, были перечислены технологии и инструменты, применённые в процессе разработки программного решения. Это включает языки программирования, фреймворки, системы управления базами данных и другие компоненты, необходимые для функционирования системы.

Важно подчеркнуть, что выбор и использование этих технологий были основаны на их способности решить конкретные задачи, их надежности, а также на лёгкости интеграции с другими системами [11, 12].Этот подход обеспечивает гибкость и возможность адаптации системы к изменяющимся потребностям и требованиям среды.

Библиографический список

1. Федеральная служба безопасности (ФСБ). – [Электронный ресурс]. – Режим доступа: https://www.fsb.ru/.
2. SANS Institute. – [Электронный ресурс]. – Режим доступа: https://www.sans.org/.
3. National Institute of Standards and Technology (NIST). – [Электронный ресурс]. – Режим доступа: https://www.nist.gov/.
4. OWASP (Open Web Application Security Project). – [Электронный ресурс]. – Режим доступа: https://owasp.org/.
5. CISA (Cybersecurity and Infrastructure Security Agency). – [Электронный ресурс]. – Режим доступа: https://www.cisa.gov/.
6. Microsoft Security. – [Электронный ресурс]. – Режим доступа: https://www.microsoft.com/security.
7. IBM Security. – [Электронный ресурс]. – Режим доступа: https://www.ibm.com/security.
8. Cisco Security. – [Электронный ресурс]. – Режим доступа: https://www.cisco.com/c/en/us/products/security/index.html.
9. Kaspersky Lab. – [Электронный ресурс]. – Режим доступа: https://www.kaspersky.com/.
10. Symantec. – [Электронный ресурс]. – Режим доступа: https://www.symantec.com/.
11. McAfee. – [Электронный ресурс]. – Режим доступа: https://www.mcafee.com/.
12. Ross, D.A. Introduction to Cybersecurity: Understanding and Applying Principles and Practices. Wiley, 2018.

ARCHITECTURE AND COMPONENTS OF THE DEVELOPED BIOMETRIC AUTHENTICATION SYSTEM IN THE OFFICE ENVIRONMENT

D.S. Kalininskiy, *Graduate Student*
Moscow Technical University of Communications and Informatics
(Russia, Moscow)

Abstract. *This article provides a detailed overview of the architectural and technological aspects that underlie the process of developing a biometric authentication software system. The article highlights the key elements of such a system, including authentication, access control and monitoring modules, while emphasizing their importance for providing an increased level of security in the office environment. A detailed analysis of each of these components is presented, their functional responsibilities and interactions are considered in order to provide readers with a deep understanding of the principles of the system. In addition, the technologies and tools chosen for the development of the system are considered in detail, including the C # and Python programming languages, the OpenCV, dlib, TensorFlow and Flask libraries.*

Keywords: *biometric authentication, software architecture, access control system, monitoring, technology stack, Python, C#, OpenCV, dlib, TensorFlow, Flask.*