International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I1P113

Volume 72 Issue 1, 130-140, January 2024

Original Article

Implementation of an Authentication System based on Facial Recognition, Artificial Intelligence and Blockchain

Grimaldo Sthiward Casaño Rivera[1], Marko Jair Arteaga Polo[2], Laberiano Andrade-Arenas[3]

[1,2,3]*Faculty of Sciences and Engineering, University of Sciences and Humanities, Lima-Peru.*

[3]*Corresponding Author: landrade@uch.edu.pe*

Received: 03 August 2023 Revised: 20 October 2023 Accepted: 16 December 2023 Published: 07 January 2024

Abstract *- This study proposes the implementation of an innovative login system that uses facial recognition based on artificial intelligence and blockchain technology to guarantee the privacy and security of the data of customers of the company Dialyma. The aim is to replace manual access with traditional credentials with a more secure and efficient facial biometric method. For this, Python is used in the backend and libraries such as OpenCV and TensorFlow for facial recognition. The Reniec service, called "Face Biometric Web Service", is implemented for facial validation. Blockchain technology is used to store and protect personal data securely, ensuring their integrity and preventing possible attacks or manipulations. The implementation of this system offers greater security data privacy and improves the user experience, reducing the risks associated with passwords. The results support its effectiveness in protecting the personal data of users of Dialyma customers. This innovative solution contributes to the protection of sensitive data, improves user experience and strengthens confidence in the management of personal information in the business environment.*

Keywords *- Authentication, System, Facial recognition, Blockchain, Python.*

1. Introduction

To begin with, the company Dialyma is an organization that develops software for companies that are in the field of healthcare, whether clinical or hospital etc. The problem raised is how to improve the process of privacy and security of personal data in the login system for the staff of the clients of the health sector of the company Dialyma since, at present, access to the system is carried out through a manual process, where users must enter their username and password. Although this practice has been common for years, it is not free of security risks. In today's increasingly digitized and cyber-attack-prone global context, facial recognition [1] has become an essential tool for ensuring the security and integrity of computer systems. Its ability to use algorithms and image processing techniques,[2] facial recognition can compare a person's captured image with a database of previously recorded images to determine if there is a match to identify and authenticate users accurately and securely makes it a preferred option in authentic, offering strong protection against threats and security risks. In the case of facial recognition, [3] a person's face information is used to verify their identity. However, it is important to note that facial recognition may have limitations in terms of accuracy and reliability, especially in environments with variable lighting or complex backgrounds.

In addition, facial recognition can be used in different environments, such as mobile devices, security systems and access control, making it a versatile option for authentication. Dialyma has been dedicated for years to providing technological tools that enable healthcare institutions to optimize their operations and provide quality patient care. Recognizing the importance of security in managing patient confidential information, the company has been constantly innovating in new solutions that address the challenges and risks associated with access to systems. The combination of facial recognition based on Artificial Intelligence (AI) and blockchain technology has led Dialyma to develop a secure and reliable authentication system that overcomes the limitations of traditional passwords and delivers an enhanced user experience. The implementation of facial recognition as an authentication method in the software that the company Dialyma makes for its healthcare clients is justified by various reasons of great relevance in the computer security component. First, traditional passwords have proven to be highly vulnerable to various security threats. By implementing [4] facial recognition, password dependency is eliminated, and the possibility of hackers accessing the system in an unauthorized way is significantly reduced. Secondly, facial recognition based on artificial AI offers an additional level of security.

The advanced algorithms used [5] in this technology are able to identify and authenticate users accurately and reliably by analyzing the unique features of their faces. In addition, facial recognition can detect and prevent fraud attempts, such as the use of photographs or masks, making it a highly secure method of authentication. It is necessary to consider the impact of this technology [6] on the fundamental rights of citizens and to ensure that the principles of proportionality and necessity in its use are respected [7].

Thirdly, the adoption of blockchain technology guarantees the privacy and integrity of users' personal data. By storing [8] the identification information in an encrypted blockchain log, manipulation and unauthorized access to such data is prevented. This gives users the confidence that their personal information is protected and complies with privacy and data protection standards [9].

The research aims to implement a facial recognition login system based on AI and blockchain technology that guarantees privacy and security for the staff and clients of the health sector of the company Dialyma.

2. Literature Review

In this section, the literature will be reviewed to explore recent studies and developments in the field of Face Recognition, AI and Blockchain Technology, with the aim of analyzing their impact on the authentication and security of computer systems. Facial recognition [10] is a type of biometric authentication that has advantages and limitations, like all biometric subsystems. According to [11], it describes how AI can be used to improve security in access control to an entity in pandemic times, using facial detection technology to identify people who wear facial masks in real-time and thus enabling more efficient and secure control of access to facilities.

According to [12], it mentions that to implement a system of access control by facial recognition, it is necessary to use Raspberry P 3B+, and the OpenCV library will be used along with XML extensions of classifiers, which will allow the device to recognize previously trained and configured faces with the hard classifier cascade. Once a face with a certain level of reliability is detected, it will be activated and will allow access to the environment.

According to [13], it proposes a methodology that combines the use of convolutional neural networks, deep learning algorithms, and real-time image capture technology for facial expression recognition and classification for real-time facial trait identification and extraction. The system used uses a real-time camera to capture facial images, which are processed by the deep learning algorithm, and the system was developed using the Python programming language and the Tensor Flow deep learning library.

According to [14], it mentions that blockchain technology has generated a revolution in recent years. Beyond the new cryptocurrencies that have emerged around different blockchain implementations, numerous contributions offer interesting services, such as a data structure linked through a decentralized network that guarantees a high level of security. Existing blockchains do not fully meet the needs in many respects, which has led to limited or superficial adoption of this technology. To overcome these limitations, Kriper, a blockchain designed for the corporate world, is introduced. Kriper is open and community-based, allowing segregation and privacy according to specific needs. It also [6] offers a distributed storage system with permissions and light message services. According to [15], they present and analyze two facial recognition strategies: Haar Outpouring and Nearby Double Example. It seeks to evaluate and compare both strategies for their implementation. The results show that the accuracy of the Haar cascade strategy is higher than that of the local binary pattern, although the Haar waterfall execution time is longer than the local Binary Pattern.

According to [5], it mentions that facial recognition has been used in identity authentication in a variety of contexts, including access to mobile devices and identification of passengers at airports. However, it is also noted that the use of this technology in authentication presents important challenges, such as the need to guarantee the accuracy and reliability of data, the protection of privacy and security of personal data, and the necessity to ensure equality and non-discrimination in access to services.

According to [10], it mentions that facial recognition is a form of biometric authentication that is more secure than conventional passwords since biometrical traits cannot be falsified. Furthermore, facial recognition enables passive authentication, which means that users can prove their identity simply by being in the room without having to interact with the system at all. It also points out that biometric identification systems, including [15] face recognition, are susceptible to supplanting attacks, in which an attacker could submit false biometrical information to gain credibility. To counter these attacks, he proposes an image encryption scheme that is integrated into the facial recognition process. According to [17], it tells that the facial recognition system to control the entry and exit of employees at BICU is an automated algorithm process that verifies and recognizes the identity of the person according to their facial characteristics. This system is responsible for recording the entry and exit of university employees by recognizing their faces. The proposal promises more accuracy in identification than the current control systems. Moreover, mentions that facial recognition technologies are increasingly being used as a form of authentication in different areas, including transport and education, in this latter area of education, including addressing issues such as campus security, automated registration and student emotion detection.

According to [18], it mentions that for carrying out assistance control with facial recognition and OpenCV in authentication, the API designed with face recognition and OpenCV that focuses on the registration and control of student assistance can be used. The implementation of the API in authentication involves the use of facial recognition [13] to verify the identity of a person by comparing the facial patterns captured with the stored data.

This technology is used as an effective method for verifying identity in various systems. Authentication is carried out through facial pattern recognition, and it uses OpenCV [20], which is a computer vision library that is used to detect and recognize faces using specialized algorithms and techniques.

This tool allows you to identify facial features and make comparisons for the identification of individuals in images and videos. With its use, it is possible to authenticate a person's identity through his/her face.

According to [21], it mentions that the proposed system for facial recognition in login authentication uses tools such as OpenCV and Raspberry Pi. The system captures facial images, compares them to a database, and allows access if there is a match. This eliminates the need for standard passwords and pins, providing greater security when logging in.

Facial recognition technology compares facial features with templates in a database. It is crucial for security research, such as criminal detection, forensic investigation, facial tracking, and airport security.

OpenCV [20] is a computer vision library widely used in facial recognition, object detection, and image processing, with many algorithms and functions.The Raspberry Pi [22] is a credit card-sized computer card that is used to perform various tasks like a typical computer.

It has two models, Model A and Model B, which differ in USB port and power consumption. It is also used in the project to facilitate work and drive innovation. It is portable and economical and features a 5MP camera to capture HD images and videos.

According to [23], it mentions that blockchain technology ensures decentralized and secure access, authorizing IoT resources and avoiding unauthorized access. Eliminates intermediaries and offers immutability, decentralization, anonymity and confidentiality.

Although it presents challenges, it is promising in access control. Technological tools for authentication vary per platform. Biometry, such as facial recognition iris scanning or fingerprints, is a common tool for verifying identity [24-27].

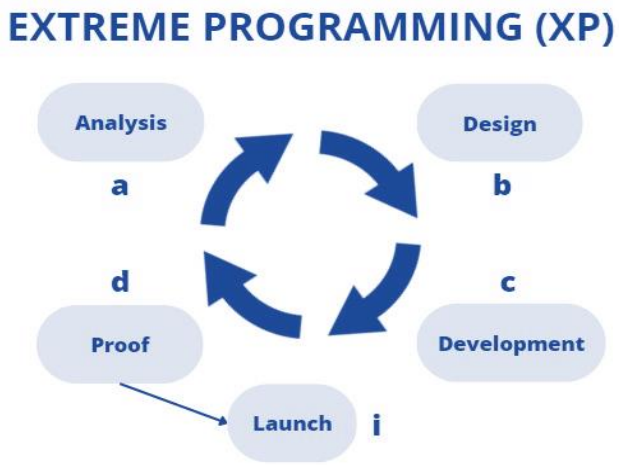

Fig. 1 Methodology XP: The four stages. (a) Analysis. (b) Design. (c) Development. (d) Proof (i) Launch.

3. Methodology

The XP methodology (Figure 1), also known as Extreme Programming, was used as a methodology [28]. This agile software development methodology focuses on the fast and adaptable delivery of high-quality software. XP [29] is based on core values and principles, which include close and constant communication between team members, simplicity in software design and deployment, continuous feedback and the ability to adapt as new requirements arise. The XP process is divided into four stages.

3.1. Analysis

At this stage, the functions of the system [30] and their linkage to each of the requirements found with the user histories are identified, and a class model should be produced, if necessary, accompanied by the refinement of the use case model.

3.2. Design

This phase will seek to develop a simple code, addressing only what is necessary [31] for the software to work. The aim is to obtain a functional prototype. Also, for the design of the object-oriented software, CRC cards (Class-Responsibility-Collaboration) will be used. In addition, CRC cards will be used as a design tool [32] to define software classes, responsibilities and how they collaborate with each other.

These cards help to visualize and organize the structure and behavior of the software, facilitating communication and decision-making in the development team.

3.3. Development

At this stage, before each user story is developed, [33] the client must specify in detail what it will do and must also be present during the tests to ensure that the implemented history matches the functionality that has been developed.

3.4. Proof

At this stage the focus of XP is the use of tests [34] to verify that the code implementation works. Applications that will run tests must be built in a testing-specific development environment. At this stage, evidence should be created that does not depend on the code to be evaluated in the future. It is recommended that you create [35] summary tests from the code in the future, so make sure that the test is independent of the code that is evaluated in each iteration

4. Results

4.1. Analysis

- This part collects the requirements that are required to implement the Dialyma system.
- User History I - Iteration I: The requirement is a generic window for users of no importance in the areas in which they are and a window with 2 functionalities for the administrator.
- User History I - Iteration II: The requirement is that in any window in which you log in, it is through facial recognition.
- User History III - Iteration III: The Dialyma manager reports that all images (faces) of users are required to be in the database with blockchain encryption.
- User History III - Iteration IV: The Dialyma manager reports that a key and an IV are required for the encryption and decryption of the images, either at the time of recording them or obtaining them from the database.

4.2. Design

In this part, they implement some of the diagrams needed to implement the development of facial recognition authentication software. In addition, asymmetrical encryption is used for encrypting the image (Figure 2). The diagram to record (Figure 3) contains a few steps A. The administrator enters the authentication system and chooses the Option to Register User B. The user who will be registered approaches and presses the register button, and focuses his face. C. When capturing the image, it is sent to generate the key and the encryption IV D. The key and IV are registered in the encryption data table. Then, the key and the encryption IV are encrypted in AES in CBC mode with blockchain technology, just as the image captured in the form of bytes is encrypted, and E. is saved in the "user" table in MySQL. The authentication diagram (Figure 4) contains a few steps A. The user enters the authentication system B. Press the login button C. The camera captures the image and the username, with the username checked in the "user" table. Suppose the name exists with the user's id. In that case, it is filtered with the related table "encryption_data" D. In the table "encryption _data", the key and the IV of the user are obtained to decrypt the image on the "username" table E. When obtaining the unencrypted image, it is compared to the image captured by the camera. If compatibility is greater than 0.98, the user gets access to its respective dashboard.

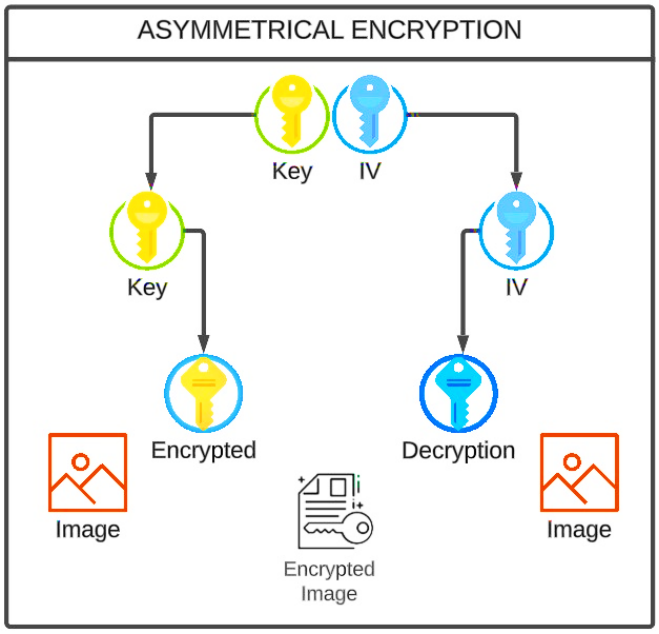

Fig. 2 Asymmetrical encryption

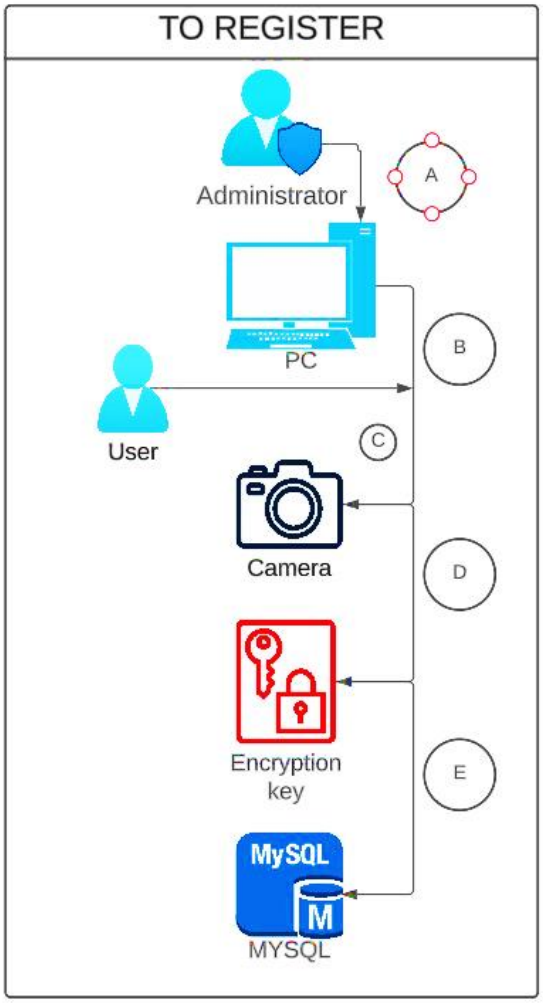

Fig. 3 The diagram for registering users with 5 steps A, B, C, D, E

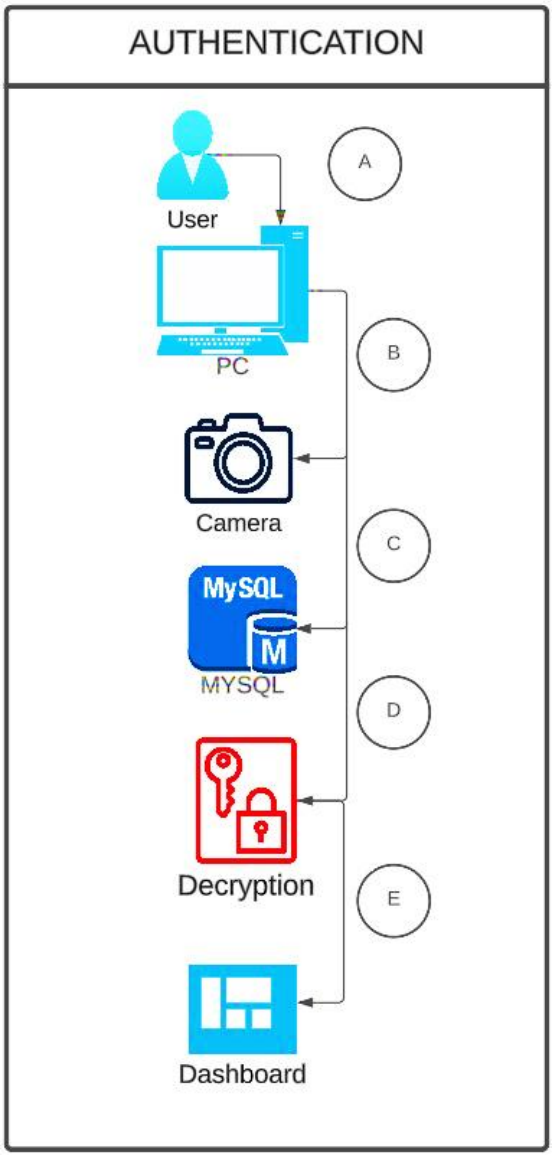

Fig. 4 Authentication diagram with 5 steps A, B, C, D, E

4.3. Development

For the development of this authentication system, first, the imports of the Python libraries and, at the same time, the connector with MySQL [17] are carried out. The project was carried out in 2 main Python files, one in database.py (Figure 5), where the connection is made with MySQL and the import of the library [36] Crypto dome for encryption with AES encrypting [16] and Json import as it will use a Json file where are the parameters for the connection with the database manager.

The other file is facial_recognition.py (Figure 6), which contains libraries such as tkinter, which is a Python library that is used to create and develop desktop applications, also the MTCNN library, [19] is a neural network that has the ability to identify faces and facial reference points in images, and also the Pyplot library which is a Matplotlib module [10] that offers a series of simple functions to add elements such as lines, images or text to the axes of a graph and finally the OpenCV library [18], [20] to process the input image and analyze the composition of primary colors, such as red, green and blue.

Creates the function of writing a file (Figure 7) to write into the file the unencrypted image of the database at the time you want to get the user to login. To obtain path information (Figure 8), only place the location of the folder in a global variable.

```
import mysql.connector as db
import json
import os
from Cryptodome.Cipher import AES
from Cryptodome.Util.Padding import pad, unpad
from Cryptodome.Random import get_random_bytes
```

Fig. 5 Importing libraries in the database.py file.

```
from tkinter import *
from tkinter import Tk, Button
from tkinter import messagebox as msg
import os
import cv2
from matplotlib import pyplot as plt
from mtcnn.mtcnn import MTCNN
import database as db
import json
```

Fig. 6 Importing libraries in the facial_recognition.py file

```
def write_file(data, path):
    # Convert binary data to proper format and write it on your computer
    with open(path, 'wb') as file:
        file.write(data)
```

Fig. 7 Write file function

```
#put the location of the folder and at the end put /
path = "D:/Proyect/"
```

Fig. 8 Location of the folder

The function of converting to binary data (Figure 9) is created to convert digital data, such as images, into binary format when you want to register a user.

The compatibility function (Figure 10) is created to compare the image that is decrypted from the "user" table of the "company" database with the image captured at the time of authentication for login.

```python
def convertToBinaryData(filename):
    # Convert digital data to binary format
    try:
        with open(filename, 'rb') as file:
            binaryData = file.read()
        return binaryData
    except:
        return 0
```

Fig. 9 Convert to binary data function

```python
def compatibility(img1, img2):
    orb = cv2.ORB_create()

    kpa, dac1 = orb.detectAndCompute(img1, None)
    kpa, dac2 = orb.detectAndCompute(img2, None)

    comp = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)

    matches = comp.match(dac1, dac2)

    similar = [x for x in matches if x.distance < 70]
    if len(matches) == 0:
        return 0
    return len(similar)/len(matches)
```

Fig. 10 Compatibility function

It is created by the user registration function (Figure 11) to register the username and photo in the database; within the function, there is a try block (Figure 12) where the generation of key and IV encryption keys is carried out.

Subsequently being registered in the table "encryption_data" having registered their keys by obtaining the id of the row inserted in "encrypting_data", it proceeds to the configuration of the AES encryption in CBC mode with the technology of blockchain at the same time the captured image is converted into bytes in the function of convertToBinaryData with the parameter that is an image when generating the encrypt and the binary conversion of the image, at that time the encryption is carried out of the bytes of the picture and then is registered into the "user" table of the "company" database.

```python
def registerUser(name, photo):
    id = 0
    inserted = 0

    try: ...
    except db.Error as e:
        print(f"Failed inserting image: {e}")
    finally:
        if con.is_connected():
            cursor.close()
            con.close()
    return {"id": id, "affected":inserted}
```

Fig. 11 Register user function

```python
try:
    con = db.connect(host=keys["host"], user=keys["user"],
        password=keys["password"], database=keys["database"])
    cursor = con.cursor()
    datacursor = con.cursor()

    # Generate the encryption key
    key = get_random_bytes(32) # Generate a 32-byte (256-bit) key
    iv = get_random_bytes(16)  # Generate a 16 byte (128 bit) IV

    # Insert data into table [encryption_data]
    insert_encryption_data = "INSERT INTO encryption_data (`key`, `iv`) VALUES (%s, %s)"
    datacursor.execute(insert_encryption_data, (key, iv))
    con.commit()

    # Get the generated ID for the row inserted into [encryption_data]
    encryption_data_id = datacursor.lastrowid

    # Configure AES encryption in CBC mode, BLOCKCHAIN Technology
    cipher = AES.new(key,AES.MODE_CBC,iv)

    # Read the image in the form of bytes, in a function
    image_bytes = convertToBinaryData(photo)

    # Encrypt the image bytesn
    padded_image_bytes = pad(image_bytes, 16)
    encrypted_image = cipher.encrypt(padded_image_bytes)

    sql = "INSERT INTO user (name, photo, encryption_data_id) VALUES (%s,%s,%s)"
    if encrypted_image:
        cursor.execute(sql, (name, encrypted_image, encryption_data_id))
        con.commit()
        inserted = cursor.rowcount
        id = cursor.lastrowid
except db.Error as e: ...
finally: ...
return {"id": id, "affected":inserted}
```

Fig. 12 Try block

The function get user (Figure 13) was created to obtain the user and image. Inside the function, there is a try block (Figure 14) where the user will be verified, and then the decrypted image will be obtained.

Inside the try block, there is a SQL query with the user name that is sent to the function to verify if it exists or not in the "user" table, and if it is affirmative, the user id will be sent as a parameter in another SQL query in the "encryption data" table to obtain the key and the IV unique keys generated for the decryption of the image and when it is decrypted the image is sent in bytes in the Write file function to then go through a (Figure 15) if where it uses the compatibility function to give access or deny.

```python
def getUser(name, path):
    id = 0
    rows = 0

    try: ...

    except db.Error as e:
        print(f"Failed to read image: {e}")
    finally:
        if con.is_connected():
            cursor.close()
            con.close()
    return {"id": id, "affected": rows}
```

Fig. 13 Function get user

```
try:
    con = db.connect(host=keys["host"], user=keys["user"],
        password=keys["password"], database=keys["database"])
    cursor = con.cursor()
    datacursor = con.cursor()

    sql = "SELECT * FROM `user` WHERE name = %s"
    cursor.execute(sql, (name,))
    records = cursor.fetchall()

    # Check if the data exists
    for row in records:
        id = row[0]
        encrypted_image = row[2]
        encryption_id = row[3]

        if id is not None:
            view_encryption_data = "SELECT * FROM `encryption_data` WHERE id = %s"
            datacursor.execute(view_encryption_data,(encryption_id,))
            security = datacursor.fetchall()

            for data in security:
                key = data[1]
                iv = data[2]

            #decrypt stored image
            decipher = AES.new(key, AES.MODE_CBC, iv)
            decrypted_image_bytes = unpad(decipher.decrypt(encrypted_image), 16)

            write_file(decrypted_image_bytes, path)
            rows = len(records)
        else:
            print(f"User not found in database")

except db.Error as e:
    print(f"Failed to read image: {e}")
```

Fig. 14 Try block in the get user function

```
res_db = db.getUser(user_login, path + img_user)
if(res_db["affected"]):
    my_files = os.listdir()
    if img_user in my_files:
        face_reg = cv2.imread(img_user, 0)
        face_log = cv2.imread(img, 0)

        comp = compatibility(face_reg, face_log)

        if comp >= 0.98:
            print("{}Compatibilidad del {:.1%}{}".
            format(color_success, float(comp), color_normal))

            printAndShow(ventana, f"Bienvenido, {user_login}", 1)
        else:
            print("{}Compatibilidad del {:.1%}{}".
            format(color_error, float(comp), color_normal))

            printAndShow(ventana,"¡Error! "
            "Incopatibilidad de datos, enfocar su rostro correctamente.", 0)
        os.remove(img_user)

    else:
        printAndShow(ventana, "¡Error! Usuario no encontrado", 0)
else:
    printAndShow(ventana, "¡Error! Usuario no encontrado", 0)
os.remove(img)
```

Fig. 15 If in get user function

4.4. Proof

This section shows the tests of the authentication system, with different roles, either user or administrator.

The first test is the visualization of the two windows; one is for all users (Figure 16), and the other one is only for the administrator (Figure 17).

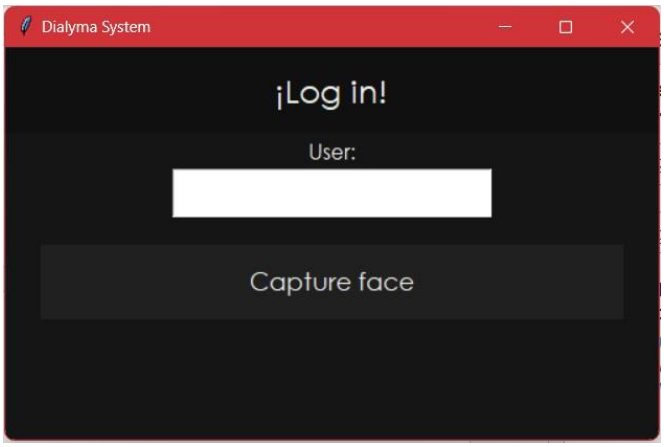

Fig. 16 User main window

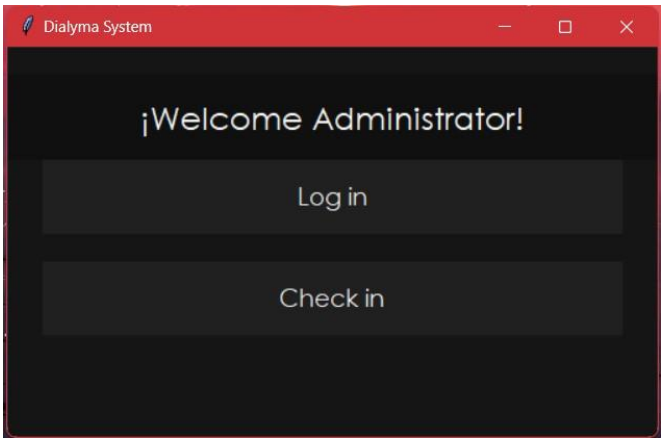

Fig 17 Main manager window

To register a user, enter the user's name (Figure 18) and then capture the user's image (Figure 19). When the registration is successful, a success message is displayed (Figure 20). At the same time, the terminal displays a message of successful registration in the "user" table of the database (Figure 21).

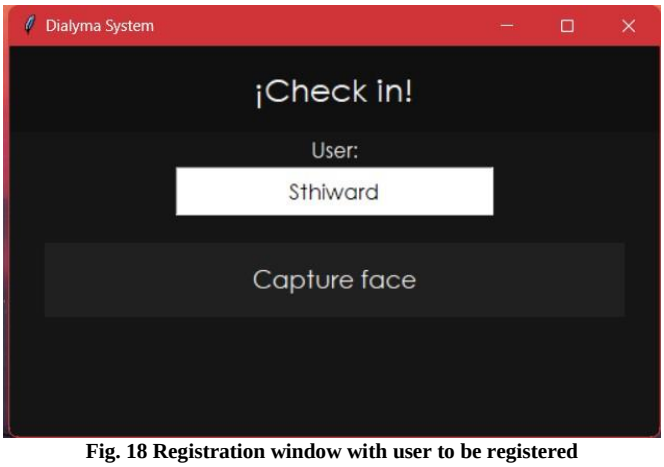

Fig. 18 Registration window with user to be registered

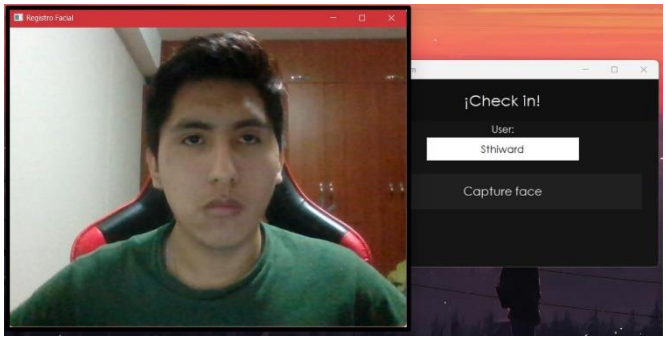

Fig. 19 The capture of the user's image for registration

Fig. 23 User or Administrator image capture

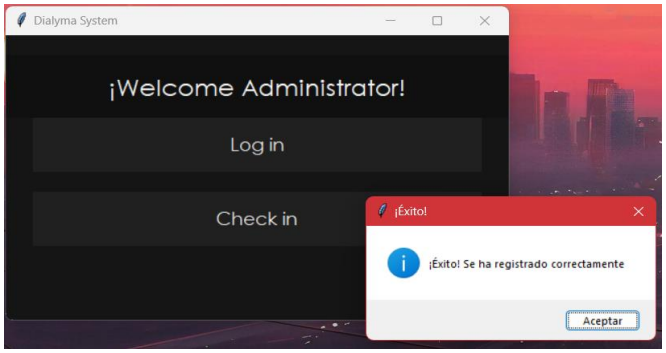

Fig. 20 Registration success message

Fig. 24 Error message when logging in

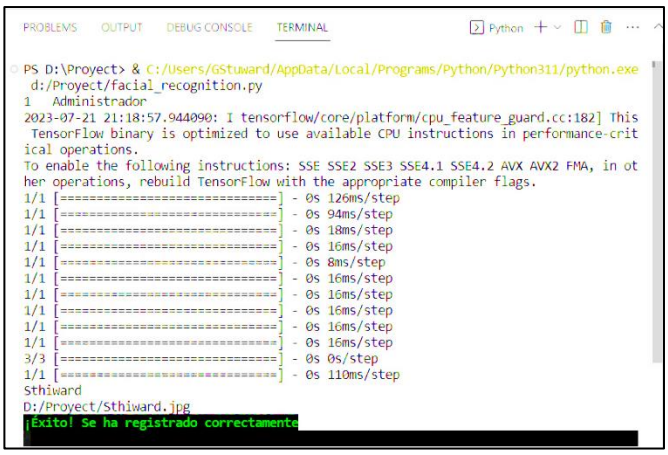

Fig. 21 Message on terminal

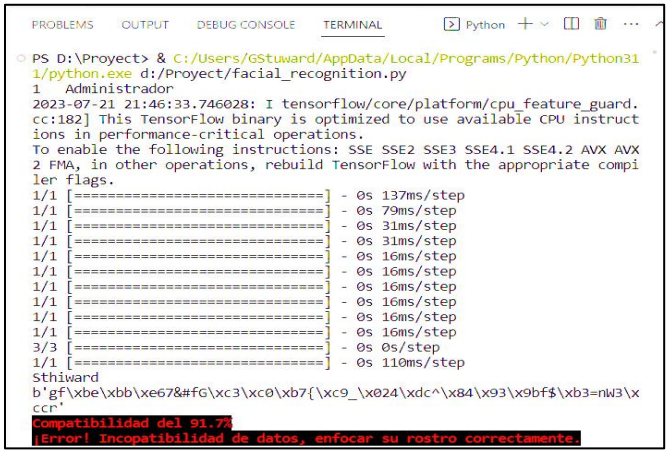

Fig. 25 error message in the terminal when logging in

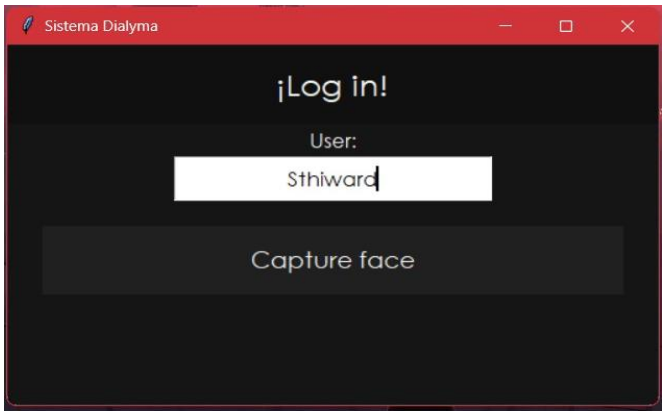

Fig. 22 Window to log in with the user

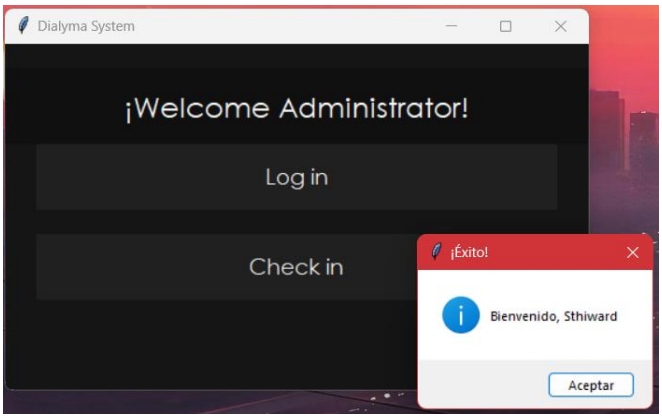

Fig. 26 Login message

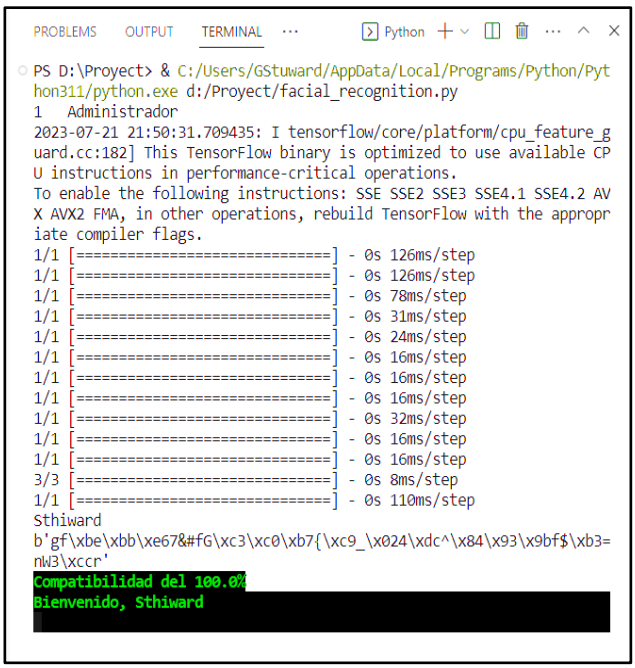

Fig. 27 Login message in the terminal with the compatibility percentage

For user or Administrator authentication for system login, the user's name is entered (Figure 22), and then the user or Administrator image is captured for credential verification (Figure 23). The credentials were verified, and if it is not compatible, it will show text inside the window (Figure 24). At the same time, a message mentioning the incompatibility will be shown in the terminal (Figure 25). If the verification is correct and the compatibility is the same, it will be shown in a floating message (Figure 26). At the same time, the terminal will show a message with the percentage of compatibility (Figure 27).

5. Conclusion

The implementation of an authentication system based on facial recognition, AI and blockchain technology to protect the privacy and security of personal data is an innovative and effective solution to improve the security of Dialyma's customers. By eliminating the need for passwords and reducing the possibility of unauthorized access by hackers, this facial biometric method can improve the security of personal data compared to traditional authentication methods. In addition, encryption of images and data stored in the database using blockchain technology provides an additional layer of security and privacy.

Overall, the implementation of this authentication system seems to be a promising solution for improving the security of personal data in healthcare and other sectors.

References

[1] Ashish Basare, Darshan Bhojak, and Ramesh Solanki, "Biometric Authentication System," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 6, pp. 3232-3238, 2023. [CrossRef] [Publisher Link]

[2] Lixiang Li et al., "A Review of Face Recognition Technology," *IEEE Access*, vol. 8, pp. 139110–139120, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Sifatnur Rahman, Mahabur Rahman, and Mijanur Rahman, "Automated Student Attendance System Using Fingerprint Recognition," *Edelweiss Applied Science and Technology*, vol. 2, no. 1, pp. 90–94, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] Lara-Jacho Steven Bryan, Albarracín-Zambrano Luis Orlando, and Ponce-Ruiz Dionisio Vitalio, "Facial Recognition Prototype to Improve Student Attendance Control at UNIANDES, Quevedo," *Revista Arbitrada Interdisciplinaria Koinonía*, vol. 5, no. 2, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] José L. Aznarte, Mariano Melendo Pardo, and Juan Manuel Lacruz López, "On the Use of Facial Recognition Technologies in University: The Uned Case," *RIED-Revista Iberoamericana de Educacion a Distancia*, vol. 25, no. 1, pp. 261–277, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Antón Fructuoso Freire Montero, "Facial Recognition as a Crime Investigation and Prevention Instrument," *Anuario da Facultade de Dereito da Universidade da Coruña*, vol. 26, pp. 64–88, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Sukesha Subhash Patil, and Y.L. Puranik, "Blockchain Technology," *International Journal of Trend in Scientific Research and Development*, vol. 3, no. 4, pp. 573–574, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[8] Om Pal et al., "Key Management for Blockchain Technology," *ICT Express*, vol. 7, no. 1, pp. 76–80, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Thien Huynh-The et al., "Blockchain for the Metaverse: A Review," *Future Generation Computer Systems*, vol. 143, pp. 401–419, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[10] Eimad Abusham et al., "Facial Image Encryption for Secure Face Recognition System," *Electronics (Switzerland)*, vol. 12, no. 3, pp. 1-26, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Enrique Lee Huamaní, and Lilian Ocares-Cunyarachi, "Use of Artificial Intelligence for Face Detection with Face Mask in Real Time to Control the Entrance to an Entity," *International Journal of Emerging Technology and Advanced Engineering*, vol. 11, no. 11, pp. 68-75, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Moreano Rojas, and Maldonado Lezama, "*Access Control Using Facial Recognition*," Thesis, Universidad Ricardo Palma Facultad de Ingeniería, pp. 1-18, 2019. [Publisher Link]

[13] Jose Carlos Bustamante, Ciro Rodriguez, and Doris Esenarro, "Real Time Facial Expression Recognition System Based on Deep Learning," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2S11, pp. 4047–4051, 2019. [CrossRef] [Publisher Link]

[14] María Isabel Rojo-Rivas et al., "Kriper: A Blockchain Network with Permissioned Storage," *Future Generation Computer Systems*, vol. 138, pp. 160-171, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Anirudha B. Shetty et al., "Facial Recognition Using Haar Cascade and LBP classifiers," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 330–335, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Taufik Hidayat, and Rahutomo Mahardiko, "A Systematic Literature Review Method on AES Algorithm for Data Sharing Encryption on Cloud Computing," *International Journal of Artificial Intelligence Research*, vol. 4, no. 1, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Yelsy Jiany Allen Ramirez, Lorvick Kayton Tucker Knight, and Dexon Mckensy Sambola, "Facial Recognition System for Employee Entry and Exit Control," *Wani*, no. 76, pp. 3-11, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Ingrid Angélica García Torres et al., "API for Attendance Control with Facial Recognition Using OpenCv.JS," *Revista Tecnológica Ciencia y Educación Edwards Deming*, vol. 5, no. 1, 2021. [Google Scholar] [Publisher Link]

[19] Qinying Yuan, "Research on Classroom Emotion Recognition Algorithm Based on Visual Emotion Classification," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1-10, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] P. Raguraman et al., "Color Detection of RGB Images Using Python and OpenCv," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 7, no. 1, pp. 109–112, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] R. Sandesh et al., "Smart Door Lock/Unlock Using Raspberry Pi," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 6, no. 3, pp. 543–548, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Faishal Rahaman et al., "Design and Implementation of a Face Recognition Based Door Access Security System Using Raspberry Pi," *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 11, pp. 1705-1709, 2021. [Google Scholar] [Publisher Link]

[23] Sarra Namane, and Imed Ben Dhaou, "Blockchain-Based Access Control Techniques for IoT Applications," *Electronics (Switzerland)*, vol. 11, no. 14, pp. 1-29, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] Xu Xu, Zhou Ruan, and Lei Yang, "Facial Expression Recognition Based on Graph Neural Network," *2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC)*, Beijing, China, pp. 211–214. 2020. [CrossRef] [Google Scholar] [Publisher Link]

[25] Alejandro Boza-Chua et al., "Development of a Security System Based on Facial Recognition Oriented to the Management and Diversion of Criminal Attacks," *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 2, pp. 48–54, 2022. [CrossRef] [Publisher Link]

[26] Juan Carlos Suárez Macedo, Nestor Asbel Cayllahua Aquino, Pedro Huamaní Navarrete, "Practical Approach of Application the Deep Learning Toolbox by Matlab in the Facial Recognition of Students," *Proceedings of the 18th Latin American and Caribbean Consortium of Engineering Institutions International Multi-Conference for Engineering, Education and Technology*, pp. 1-9, 2020. [CrossRef] [Publisher Link]

[27] R. Sivapriyan, N. Pavan Kumar, and H.L. Suresh, "Analysis of Facial Recognition Techniques," *Materials Today Proceedings*, vol. 57, pp. 2350–2354, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Asma Akhtar, Birra Bakhtawar, and Samia Akhtar, "Extreme Programming Vs Scrum: A Comparison of Agile Models," *International Journal of Technology, Innovation and Management (IJTIM)*, vol. 2, no. 2, pp. 80-96, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Khusbhu Sahendrasingh Yadav, Maleeha Arif Yasvi, and Shubhika, "Review On Extreme Programming-XP," *International Conference on Robotics, Smart Technology Communication and Electronics Engineering*, at Delhi, 2019. [Google Scholar] [Publisher Link]

[30] Freddy Adrian Moreira Pinargote et al., "Methodological Proposal for Software Development in Degree Projects in the Specialty of Computer Systems Engineering," *IJERI: International Journal of Educational Research and Innovation*, no. 12, pp. 76–89, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[31] Edwin Bautista Villegas, "Agile Methodologies XP and Scrum, Used for the Development of Web Pages, under MVC, with PHP Language and Laravel Framework," vol. 1, no. 1, pp. 1-7, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[32] Alicia Raeburn, "Extreme Programming (XP) Produces Results, but is it the Right Methodology for you? *Asana, Agile Methodology*, 2022. [Google Scholar] [Publisher Link]

[33] Alejandro Boza-Chua et al., "Development of a Security System Based on Facial Recognition Oriented to the Management and Diversion of Criminal Attacks," *International Journal of Emerging Technology and Advanced Engineering*, vol. 12, no. 2, pp. 48–54, 2022. [CrossRef] [Publisher Link]

[34] Jean Carlos Albuquerque Souza, and Marcus Rogério Oliveira, "Agile Methodologies: A Comparison between Extreme Programming (XP) and Scrum," *Ciência & Tecnologia*, vol. 13, no. 1, pp. 133–141, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[35] Diego León Ramírez-Bedoya, John Willian Branch-Bedoya, and Jovani Alberto Jiménez-Builes, "Methodology of Software Development for Robotic Educational Platforms Using ROS-XP," *Revista Politécnica*, vol. 15, no. 30, pp. 55–69, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[36] Holman A. Montiel, Fredy S. Martínez, and Edwar G. Jacinto, "Implementation of Password Hashing on Embedded Systems with Cryptographic Acceleration Unit," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 2, 2022. [CrossRef] [Google Scholar] [Publisher Link]