Mobile Application to Alert Health Risks in Schools in Peru based on the Experience of Expert Teacher

Lucia Asencios-Trujillo[1], Carlos Jaciento La-Rosa-Longobardi[2], Djamila Gallegos-Espinoza[3]
[1,2,3] Graduate School, Universidad Nacional de Educación Enrique Guzmán y Valle, Lima-Peru

Abstract

The new COVID-19 coronavirus outbreak at the end of 2019 has posed a tremendous threat to the whole world. Driven by the need to return to classes in a face-to-face manner and to be able to prevent the spread of contagions, using Scrum methodology, we developed a prototype of a mobile application with facial recognition to detect and alert users with high temperature, suspicion related to COVID-19 and track nearby contacts. Real-time data is obtained from each user's device through the mobile application to allow measures to be taken and isolate users with suspected cases of the virus. In this way, the contagion and spread of COVID-19 in Peruvian schools can be controlled.

Keywords: COVID-19, contagions, close contacts facial recognition, suspicions.

1. Introduction

The new outbreak of coronavirus (COVID-19) at the end of 2019 originated Wuhan (China) is a contagious disease by affecting people through temperature, shortness of breath and in many cases, death is caused by sudden or gradual collapse of the respiratory and intestinal system where, the United States with more than 104 thousand deaths and Europe were the most affected and lately India[1]. Currently, it has become a global problem that affects an infinite number of sectors and human activities, one of the main ones being related to education services in almost all countries of the world, affecting all levels of education[2]. In Peru, the Ministry of Education issued Legislative Decree 1496 on May 10, 2020, in which all higher education institutions decided to cancel face-to-face classes and develop distance and virtual courses to help prevent the spread of the virus causing COVID-19[3].

Many of the European countries affected by the coronavirus strain returned to face-to-face classes and implemented mechanisms to detect symptoms early. The Institute for Image Communication and Information Processing implemented a FLIR one thermal camera connected to an Android phone to collect thermal video for real-time health screening[4]. In this system, we extract the respiratory rate and one of the four common respiratory patterns (eupnea, bradypnea, tachypnea and Apnea) according to the signals obtained from the nostrils[4]. In 2020, the Department of Electrical, Electronics and Systems Engineering, Universiti Kebangsaan Malaysia submitted a proposal for SPP-COVID-Net by integrating the SPP module into DarkCovid-Net. The strength of SPP-COVIDNet can be attributed to its ability to process multiscale features due to the integration of the SPP module. The proposed algorithm is suitable for cell phone applications[5]. On the other hand, in China, they developed a user-friendly mobile application that can help the user to identify their health status based on the inputs they typed, such as symptoms, days elapsed with symptoms. Therefore, it allows the user to avoid the worst case scenario of unprofessional self-diagnosis[6].

Taking into account the initiative of the government and the Ministry of Education to progressively return to face-to-face classes in some regions of the country. Likewise, the state forecasts that the start of face-to-face classes in 2022 will be massively 100%. Therefore, the development of the research work proposes the development of a prototype of a mobile application for the early detection of COVID-19 symptoms, preventing the spread in schools.

This document is structured as follows, in the following section II we will describe in detail the methodology used for the mobile application as well as the solution approach and the technological tools used. In section III we will show the results obtained by means of the prototypes according to the user's requirements. In section IV we will deal with the discussions, and finally in section V we will present the conclusions.

2. Methodology

2.1 Scrum

The SCRUM methodology has a team-oriented approach, enables and develops web and mobile systems through a collaborative and adaptive to change functional environment, based on regular deliverables of the final product.

Scrum is an iterative and incremental improvement approach that prioritizes flexibility and adaptability in complex and changing environments. Scrum is a standardized process consisting of several steps to follow and ensure high quality products, on time and within budget. In contrast, Scrum is a framework for organizing and managing work that provides a foundation to which organizations can add a unique implementation according to their approaches and needs. It has advantages such as flexible selection of requirements for sprint development. [7, 8].

- Requirements analysis: SCRUM is based on the incremental development of requirements, considering the prioritization of user stories depending on the value assigned by the customer. Also, this agile methodology is applicable when an empirical project control is required [8]. Therefore, the requirements for the development of the application were obtained through research and reliable sources from the Ministry of Health, as shown in Table 1 of the requirements analysis.

Table 1 Analysis Of Requirements

item	***Description of requirements***
1	A map showing the location of users with suspected COVID-19..
2	It is necessary to see the detailed suspicions of the users of the environment.
3	It is necessary to be able to view reports of users with suspected COVID-19.
4	It is required to know the user's information through a facial scan.
5	A login is required to access the Application.
6	The application must have a database of registered users.

- Sprint Development: A Sprint is a cycle or an iteration that is held in a project with a duration of four weeks. Each Sprint of work is developed to achieve a deliverable or increase in product value. Likewise, in the first Sprint the minimum viable version of the product as well as the elementary functionalities[7]. Therefore, for the development of the mobile application it was planned in three Sprint or deliverable as shown in Table 2 Sprint development, taking into account the prioritization of the customer.

Table 2 Sprint Development

item	***Description***
Sprint 1	R1: A map showing the location of users with suspected COVID-19.
	R6: The application must have a database of registered users.
Sprint 2	R4: It is required to know the user's information through a facial scan.
	R5: A login is required to access the application.
Sprint 3	R2: It is necessary to see the detailed suspicions of the users of the environment.
	R3: It is necessary to be able to view reports of users with suspected COVID-19 symptoms.

2.2 Technological tools

Figma: Technology that offers all the necessary tools to design projects, prototypes, generate code for hand-off and illustration. The main advantage is that it allows collaborative work, sharing components between the different files being worked on, facilitating the workflow.

Flutter: Open source mobile application development SDK created by Google. It is often used to develop user interfaces for android, iOS and Web applications. Flutter provides a method for structuring an application architecturally and to solve the problem of state management, authentication related rules can be tested in isolation from the actual authentication implementation, either through an SDK such as AWS Mobile or an HTTP call to an APIAPI[9].

Spring Cloud: Another technology used for application development was the Spring Cloud big data framework for its microservices architecture. It provides tools for developers to quickly build some of the common patterns in distributed systems[10].

Open CV: It can work well with Windows, mackintosh or Linux, Using OpenCV results in more time and resource utilization in image processing and less in rendering[11]. OpenCV library is flexible enough for the project as it can accurately and correctly detect a face in real time and highlights it by drawing a rectangle around the faces[12].

2.3 Facial Recognition

In the design of the application platform, the big data framework Spring Cloud was applied for its microservices architecture. It provides tools to quickly build some common patterns in distributed systems, such as intelligent routing, micro proxy, distributed sessions[13].

The main functional blocks include facial image ingestion, facial object storage, facial feature matching, eavesdropping service and data analysis. In the middle of function blocks there is a queue manager. We use rabbitMQ.

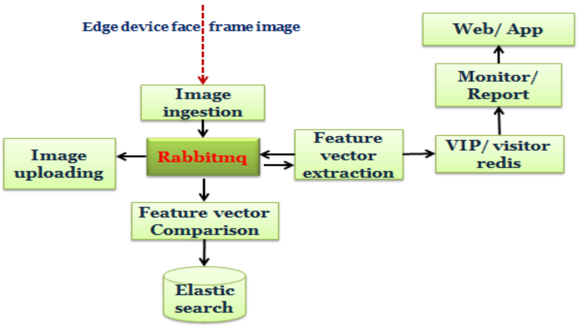


Fig. 1. Diagram of facial recognition.

The image ingestion block takes images from edge devices, for example, face recognition cameras, access control tablets, or fever detection tablets. These edge devices have built-in face detection capability, local face recognition and some control logic[14]. When a face is detected, both the face image and the frame image are sent to the cloud server for tracking.

The second block is the image upload block. It is responsible for uploading the face images or frames from rabbitMQ to the cloud storage. The third block is the feature extraction block. It receives the face image from the rabbitMQ through a communication socket. It uses a cluster of GPU machines with automatic routing load balancing and failover. Extracted feature vectors are returned to the rabbitMQ and stored in a ridis VIP/visitor database.

The feature vector comparison blocks receive the feature vector from the rabbitMQ, and call the vector search engine (VSE) to obtain face clustering information. The clustering message is stored in Elastic search. It uses socket communication with the VSE. The monitor block monitors the data in Redis, at the same time, a notification is sent to the web interface or mobile application. The monitor provides an API for statistics and heat map search.

The key algorithm of the VSE is a nearest user search algorithm. The VSE receives the face feature vector. It also receives the query face feature vector. When no valid record is found in the face feature vector database, a new record is inserted[15].

Using two cameras: a thermal camera to measure temperature and a visible light RGB camera to perform face detection and recognition. After detecting a face, the thermal camera locates the person's forehead and measures the skin temperature. To improve accuracy, a black body can be used as a reference for temperature measurement as shown in Figure 2. The device can be used as a local fever detector and face recognizer for actual identification of a face all users must be registered to facilitate management in the cloud where batch registration is supported. The results of temperature measurement and face recognition can be immediately reported locally.

However, tracking is not possible because of data access through multiple devices. Therefore, in cloud server mode, the results of facial recognition and tracking are performed in the cloud.

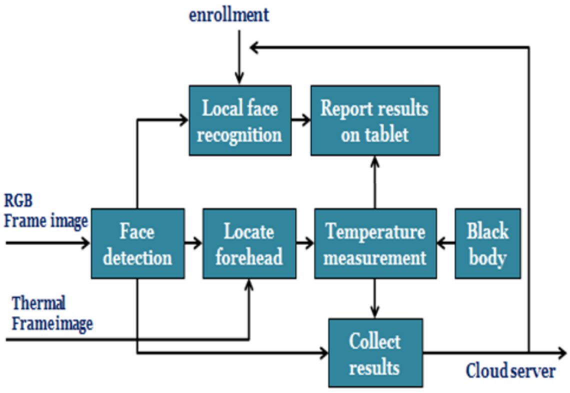


Fig. 2. Diagram of fever detection.

For tracking nearby users in study centers were based on some COVID-19 related symptoms such as fever taking temperature as an input of our data type, we added data analysis, alert type with a heat map. The most important thing was the closest contact tracking in addition to the fever encryption by device camera. If the history of this user is retrieved the results are grouped together, the contacts close to the user in query can be identified. As a result, monitoring and alerting measures can be taken to stop the spread of the virus as shown in Figure 3.

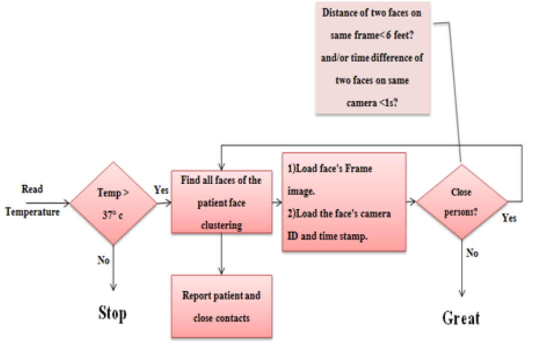

Distance of two faces on same frame< 6 feet? and/or time difference of two faces on same camera <1s?
Read Temperature
Temp > 37° c
Yes
No
Stop
Find all faces of the patient face clustering
Report patient and close contacts
1)Load face's Frame image.
2)Load the face's camera ID and time stamp.
Close persons?
Yes
No
Great

Fig. 3. Algorithm for tracing users and close contacts

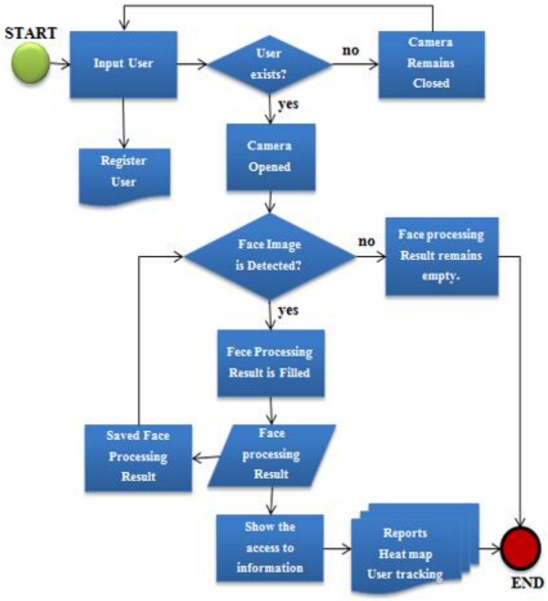

START
Input User
User exists?
no
Camera Remains Closed
yes
Register User
Camera Opened
Face Image is Detected?
no
Face processing Result remains empty.
yes
Fece Processing Result is Filled
Saved Face Processing Result
Face processing Result
Show the access to information
Reports Heat map User tracking
END

Fig. 4. Diagram of the application operation

The proposed architecture for the application is shown in Figure 5. The user performs the facial recognition using the mobile device, the image is sent to RabbitMQ for routing, it connects to the VSE algorithm API, performs the feature comparison and returns the comparison results to the device, if it is a new user, it is registered in the database.

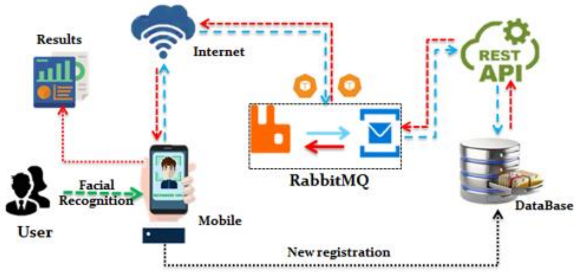


Fig. 5. Proposed architecture for the application

3. RESULTS

3.1 Prototype of the application

The prototyping cycle includes the identification of the end user's business needs, the development of a prototype of a mobile application was carried out through a system design that aims to ensure that the application meets functional specifications in the form of structured interface designs. The prototype has been designed and tested, through a system requirements analysis with the objective of identifying specific system issues and requirements, especially specifications to the implementation of the system[16].

The prototypes were used to have a clear idea of how the final product will be, taking into account the requirements of the project. It is also important for the developer to have an overview of how the structure of the final application will be, facilitating the analysis of the functionalities, giving a vision of how each requirement will be structured. The prototype was developed in Figma, where it shows a graphical view of the structure of the application, showing the startup window, user session, general map, face scan, report of users with suspicions of COVID-19.

For the development of the prototypes of the application, the prioritization of the requirements and the base functionalities for the operation of the application were taken into account with the purpose of increasing the value through the development progress, based on the above, they were divided into three Sprint.

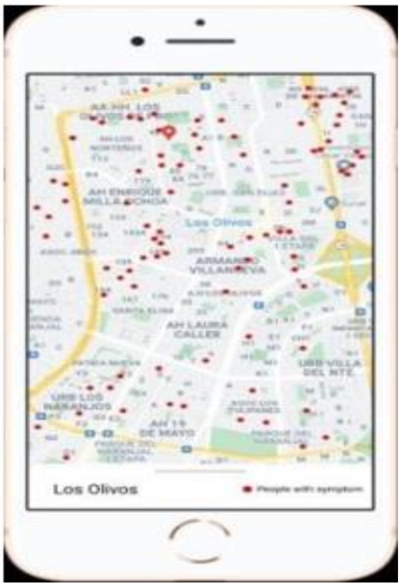


Fig. 6. Map of user location

- Sprint 1: Requirements were developed: R1, where it was possible to show a heat map of the users of the educational center taking into account the temperature of each registered user as shown in Figure 6. R6 was developed and implemented which allows access to the registered data to show possible suspicions related to COVID-19 symptoms.

- Sprint 2: Requirements R4 and R5 were developed: In R4, the functionality of being able to know the user's information through a facial scan was developed, which makes it possible to obtain the user's data in the application and compare the results obtained with the symptoms and rule out any suspicion related to COVID-19 as shown in Figure 8. In R5, the user access login was developed as the registration option for new users to access the mobile application as shown in Figure 7.

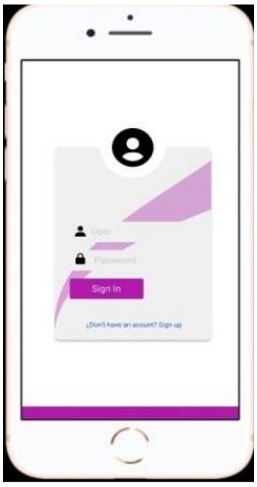
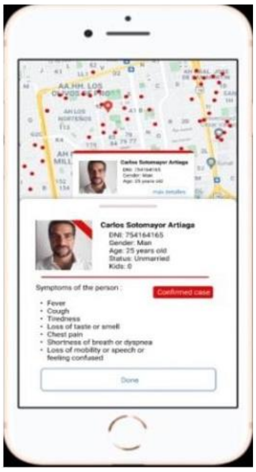


Fig. 7. Login and user registration



Carlos Sotomayor Artiaga
DNI: 754164165
Gender: Man
Age: 25 years old
Status: Unmarried
Kids: 0

Symptoms of the person :

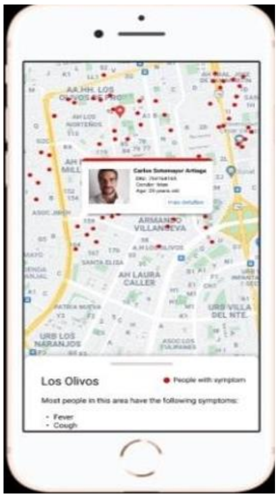
Confirmed case

- Fever
- Cough
- Tiredness
- Loss of taste or smell
- Chest pain
- Shortness of breath or dyspnea
- Loss of mobility or speech or feeling confused

Done

Fig. 8. Facil Scanning information

- Sprint 3: Requirements R2, R3 were developed where: In R2 the functionality was developed that allows to know the user's information visualized in the heat map with greater detail the results, thus alerting the location of possible users with suspicions in detail as shown in Figure 9. And finally in R3 the functionality was complemented where it allows to optener reports of all cases presented daily and keep track and track, allowing an analysis of the behavior of users with suspicions related to COVID-19 symptoms mediane statistical tools as shown in Figure 10.



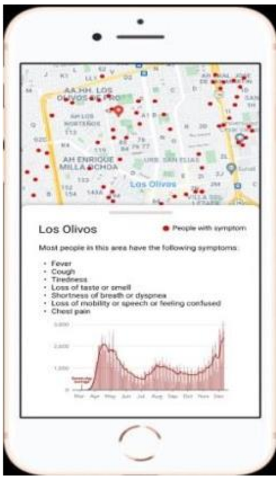
Carlos Sotomayor Artiaga
DNI: 754164165
Gender: Man
Age: 25 years old

Los Olivos
● People with symptom

Most people in this area have the following symptoms:

- Fever
- Cough

Fig. 9. Information on users with COVID-19 suspicions



Los Olivos
● People with symptom

Most people in this area have the following symptoms:

- Fever
- Cough
- Tiredness
- Loss of taste or smell
- Shortness of breath or dyspnea
- Loss of mobility or speech or feeling confused
- Chest pain

Fig. 10. Report of suspected COVID-19 cases.

4. DISCUSSIONS

4.1 About Scruu

The Scrum methodology, unlike traditional methodologies, adapts to changes allowing improvements and corrections to the initial model throughout the development of the project. In contrast, traditional methodologies stick to the initial model, preventing rapid changes that, if made, require a complicated process to be accepted. A traditional methodology only delivers the project when it is finished, while an agile methodology such as Scrum delivers functional versions of the software constantly, allowing the organization to obtain value from an early stage of the application. In the project the Scrum methodology was applied, resulting in the list of requirements to then lead the prioritization and development of the Sprint. The issues addressed during the process, discussed by team members, help to improve progress and resolve difficulties, giving us the facility to adopt to the changes that are found and not delay us with the deliverables.

4.2 About technological tools

Figma offers all the necessary tools to design prototypes, generate hand-off code and illustrate. Flutter is often used to develop user interfaces for android, iOS and Web applications. Flutter provides a method for structuring an application architecturally. Another technology used for application development was the big data framework Spring Cloud because of its microservices architecture. It provides tools for developers to quickly build some of the common patterns in distributed systems. The OpenCV library is flexible enough for the project as it can accurately and correctly detect a face in real time and highlights it by drawing a rectangle around the faces.

4.3 About facial recognition

Facial recognition on the edge device can only recognize people enrolled on the device; others are referred as visitors. Enrollment can be handled locally on the device or on the cloud platform. Considering that the cloud is more powerful, it uses GPUs to train the model and run inference to extract the face feature vector. However, more recently, the analysis of facial activity from the audio channel has been briefly studied[17].

Another source of images is the VIP image from a user input. This is useful when using some whitelist or blacklist where the image upload block takes care of uploading the faces or frames from rabbitMQ to the cloud storage. Feature extraction in face recognition can use any model such as ArcFace based model which uses a cluster of GPU machines where, the vector matching blocks receive the feature vector from rabbitmq and calls the vector search engine to get face clustering information[18]. The monitoring monitors the data. When a VIP or visitor message appears, a visitor VIP message is generated and stored in Elastic search. At the same time, a notification is sent to the mobile application interface. The monitor provides an API for cloud search statistics of registered users, high frequency visitors, heat map, etc.

Finally, for tracking and monitoring users with suspicions related to COVID-19 symptoms, the application allows facial recognition that supports encryption and fever notification, thus tracking users with fever and close contacts. In order to perform fever encryption using temperature as our data type input, and we add data analysis, alert type, message notification. This demonstrates the ease of the mobile application to integrate new devices, new data and new event processing.

5. CONCLUSIONS

A prototype of a system was developed to be able to alert suspicions related to the new strain of coronavirus by facial recognition and save their data to facilitate the monitoring of all users and close contacts. Experimental emulations show that they can detect users with fever and track nearby contacts. However, due to limitations in practical deployment, we are unable to obtain more experimental data to demonstrate the effectiveness of its operation. Hopefully, this prototype of the mobile application development can help to alert suspicions and control the spread of COVID-19 virus in educational institutions in our country and save lives. The Scrum methodology was applied throughout the development, achieving an adaptive process, which was fundamental for this project since there was no fixed path for development.

For future projects, it is recommended to implement the development and extend the functionalities of the application that will help to improve the tracking of users through new entries or data management

References

[1] H. Tahir, A. Iftikhar, and M. Mumraiz, “Forecasting COVID-19 via Registration Slips of Patients using ResNet-101 and Performance Analysis and Comparison of Prediction for COVID-19 using Faster R-CNN, Mask R-CNN, and ResNet-50,” in *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, Feb. 2021, pp. 1–6, doi: 10.1109/ICAECT49130.2021.9392487.

[2] I. O. Yahuarcani *et al.*, “Mobile Applications as Tools for Virtual Education in Indigenous Communities During the COVID-19 Pandemic in the Peruvian Amazon,” in *2020 3rd International Conference of Inclusive Technology and Education (CONTIE)*, Oct. 2020, pp. 215–221, doi: 10.1109/CONTIE51334.2020.00046.

[3] J. Oblitas and J. Jorge, “Differences in Student Satisfaction in Online Learning and Remote Teaching Courses during the COVID-19 Adaptation Stage,” in *2021 IEEE World Conference on Engineering Education*

(EDUNINE), Mar. 2021, pp. 1–5, doi: 10.1109/EDUNINE51952.2021.9429148.

[4] Z. Jiang, M. Hu, and G. Zhai, "Portable Health Screening Device of Respiratory Infections," in *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, Jul. 2020, pp. 1–2, doi: 10.1109/ICMEW46912.2020.9105969.

[5] S. R. Abdani, M. A. Zulkifley, and N. Hani Zulkifley, "A Lightweight Deep Learning Model for COVID-19 Detection," in *2020 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, Jul. 2020, pp. 1–5, doi: 10.1109/ISIEA49364.2020.9188133.

[6] W. Y. Xien, N. F. M. Radzuan, and M. N. A. Rani, "Mobile Health Monitoring and Treatment System for COVID-19 Symptoms Identification," in *2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM)*, Aug. 2021, pp. 477–482, doi: 10.1109/ICSECS52883.2021.00093.

[7] S. Ashraf and S. Aftab, "Pragmatic Evaluation of IScrum &amp; Scrum," *Int. J. Mod. Educ. Comput. Sci.*, vol. 10, no. 1, pp. 24–35, Jan. 2018, doi: 10.5815/ijmecs.2018.01.03.

[8] A. Srivastava, S. Bhardwaj, and S. Saraswat, "SCRUM model for agile methodology," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*, May 2017, pp. 864–869, doi: 10.1109/CCAA.2017.8229928.

[9] S. Boukhary and E. Colmenares, "A Clean Approach to Flutter Development through the Flutter Clean Architecture Package," in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec. 2019, pp. 1115–1120, doi: 10.1109/CSCI49370.2019.00211.

[10] W. Tan and J. Liu, "Application of Face Recognition in Tracing COVID-19 Fever Patients and Close Contacts," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec. 2020, pp. 1112–1116, doi: 10.1109/ICMLA51294.2020.00179.

[11] K. Goyal, K. Agarwal, and R. Kumar, "Face detection and tracking: Using OpenCV," in *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Apr. 2017, pp. 474–478, doi: 10.1109/ICECA.2017.8203730.

[12] C. James and D. Nettikadan, "Student Monitoring System for School Bus Using Facial Recognition," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Apr. 2019, pp. 659–663, doi: 10.1109/ICOEI.2019.8862534.

[13] S. He, L. Zhao, and M. Pan, "The Design of Inland River Ship Microservice Information System Based on Spring Cloud," in *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*, Jul. 2018, pp. 548–551, doi: 10.1109/ICISCE.2018.00120.

[14] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, Mar. 2021, doi: 10.1016/j.neucom.2020.10.081.

[15] M. Aumüller, E. Bernhardsson, and A. Faithfull, "ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms," *Inf. Syst.*, vol. 87, p. 101374, Jan. 2020, doi: 10.1016/j.is.2019.02.006.

[16] A. A. Permana, R. Taufiq, and S. Ramadhina, "Prototype Design of Mobile Application 'Hydrolite' for Hydroponics Marketplace," in *2020 7th International Conference on Electrical Engineering, Computer Sciences and Informatics (EECSI)*, Oct. 2020, pp. 45–48, doi: 10.23919/EECSI50503.2020.9251303.

[17] Z. Meng, S. Han, P. Liu, and Y. Tong, "Improving Speech Related Facial Action Unit Recognition by Audiovisual Information Fusion," *IEEE Trans. Cybern.*, vol. 49, no. 9, pp. 3293–3306, Sep. 2019, doi: 10.1109/TCYB.2018.2840090.

[18] J. Deng and S. Zafeririou, "ArcFace for Disguised Face Recognition," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct. 2019, pp. 485–493, doi: 10.1109/ICCVW.2019.00061