



ISSN ONLINE: 2447-0228



RESEARCH ARTICLE







OPEN ACCESS

SCHOOL ATTENDANCE MANAGEMENT SYSTEM FOR STUDENTS THROUGH FACIAL RECOGNITION

Yves César Amorim de Azevedo¹, Laís Reis de Jesus², Wilson Fogaça de Araújo³, Fernando Filipe Da Silva Costa⁴, Eliton Smith⁵ and Alexandra Amaro de Lima*⁶

^{1, 2, 3, 4, 5, 6}Paulista University- UNIP. Manaus, Amazonas, Brazil.

^{5, 6}Galileo Institute of Technology and Education of the Amazon- ITEGAM. Manaus, Amazonas, Brazil.

¹<http://orcid.org/0000-0002-3474-4417> , ²<http://orcid.org/0000-0002-8803-4330> , ³<http://orcid.org/0000-0001-6212-340X> ,
⁴<http://orcid.org/0000-0002-5964-0200> , ⁵<http://orcid.org/0000-0002-8039-7532> , ⁶<http://orcid.org/0000-0003-3918-0013> 

Email: afjblt123@gmail.com, laisreisdejesus45@gmail.com, wilson.fogaca2015@gmail.com, yvesamorim73@gmail.com, elitonsmith@gmail.com,
*xanduca@gmail.com

ARTICLE INFO

Article History

Received: November 14th, 2022

Accepted: December 28th, 2022

Published: December 31th, 2022

Keywords:

Artificial Intelligence,
Attendance record,
Facial recognition,
Extreme Programming (XP),
Ruby on Rails,
Python.

ABSTRACT

Currently, in the scenario of the city of Manaus, most of the records of attendance of students in educational institutions are made manually, thus making the process quite laborious and repetitive for teachers, added to this, this system does not present to be fully reliable, because the method used shows itself vulnerable to human error. To save time and ensure that the registration is done without any failure, this work develops and applies a system to register students using facial recognition, and also has a feature that triggers an email message warning the absence of the child to the parents responsible if the student is not present in class. The project uses the Ruby language, for the development of the attendance register management system, and the Python language in conjunction with the OpenCV library for facial recognition.



Copyright ©2022 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

In the youth and adult learning process, the students' attendance in the classroom is an important factor in the evaluation of their school performance, since it is closely linked to learning, development of social and emotional skills, besides helping in a more assertive pedagogical monitoring and contribution of data for statistics [1]. Currently, the traditional process of recording students' attendance in educational institutions has proven to be flawed in terms of its reliability and efficiency.

The school attendance method most used by institutions is to obtain the signatures of students in class, but this method has a flaw that is the possibility of the student to sign by a classmate, and thus circumvent the method, causing the same failure of the traditional way [2]. But with the emergence of tools such as facial recognition, it is possible to visualize safer solutions. Currently, most smartphones unlock the device from the identification of the user's face as a way to provide more security in accessing the

device data. We know that facial recognition is not restricted only to mobile phone authentications but has also been expanding for other purposes [3]. Thus, we can use it in the school context, making it the most assertive option to solve the problem of academic records through an automated facial recognition system, thus saving teachers' time and avoiding incorrect attendance records.

The system of registering attendance of students came with the aim of checking and recording entry and exit of students through facial recognition at school and informing them through a message to the responsible, giving more security of who enters and who leaves. Its development is based on languages like *Python*, *Ruby on Rails*, *MVC* practice and theories such as notions of face and its symmetry, types of automated software testing and agile methodology.

This work is organized as follows: Section 2 discusses the main tools and concepts of symmetry and facial recognition, image processing and digital image used for the development of the

project; Section 3 presents the materials and methods adhered in the creation of the system; Section 4 describes the results and discussions obtained from the ready-made software; Section 5 presents the conclusion; Section 6 presents the acknowledgements; and, finally, in Section 7, the references are shown.

II. THEORETICAL REFERENCE

II.1 COMPUTATIONAL TOOLS

II.1.1 Python

Currently, the most popular programming language to develop tools in artificial intelligence has been Python, which stands out for its simple syntax and excellent readability [4]. Thus, the choice for this language was due to the previous reasons and to the diversity of libraries and packages available for the project development. According to [5], python is an open source and interpreted programming language and, for this reason, it provides a faster performance in the execution of programs, because it does not compile the algorithm.

II.1.2 OPENCV

In [6], the author defines the *Open Source Computer Vision Library* as a multiplatform library, i.e., it can be implemented and used on major platforms such as *Windows, Linux* and *Mac*, and it is completely free, to academic and commercial use, for the development of applications in the area of Computer Vision. In addition, it has a library composed by more than 500 algorithms developed in C/C++, highlighted in [7] where the authors present an abundance of documentation of this library, and this is due to its popularity in projects focused on Computer Vision.

II.1.3 Ruby on Rails

The *Ruby* language was created by Yukihiro "Matz" Matsumoto and had its first public release in 1995. While [8] defines it as a programming language with dynamic typing aimed at solving Web Development problems in a simple way, besides providing features aimed at developer productivity. The main objective of this article is to present a discussion of the main concepts and concepts of the Web development framework. This programming language can be used in word processing, game creation and, mainly with the resources provided by *Ruby on Rails*

framework, web development, being this last one the proposal we will use in this work. In short, *Ruby's* philosophy is: "Make simple things easy and complex things possible".

According to [9], *Ruby on Rails* is a *web* application development framework written in the Ruby programming language. This tool was designed to make programming easier in the niche of solutions it finds itself in, based on the assumptions most developers need to start a project, including support for databases and the MVC (*Model-View-Controller*) architecture pattern. Its premise is to write less code while accomplishing more tasks than many languages and frameworks.

II.1.4 Machine Learning and "DLIB" Tool

Due to advances in the area of artificial intelligence and computational systems, computers have been able to automate tasks that until now were only performed exclusively by humans. Early developments in this area depended exclusively on the ability of the researcher to encode the required knowledge into the algorithm itself. A more effective methodology was then developed, involving the provision of examples from which the computer can extract this knowledge. Such methodology is called machine learning. [10].

Dlib is a modern C++ toolkit that contains machine learning algorithms and tools for creating complex C++ software to solve real-world problems. It is used in industry and academia in a wide range of domains, including robotics, embedded devices, mobile phones, and large high-performance computing environments [11].

II.2 MVC

The MVC architecture pattern was developed in 1979 by Trygve Reenskaug under the name *thingmodel-view-editor*, and was intended to be used in conjunction with Smalltalk - 76, an object-oriented programming language that influenced languages such as Java and Objective-C. [12].

The basis of MVC is the separation of what is presented to the user through the Graphical User Interface (GUI) and the data manipulation of the system itself. This separation is performed through three components with direct interactions between them. The components are model (*model*), view (*view*) and controller (*controller*) [13].

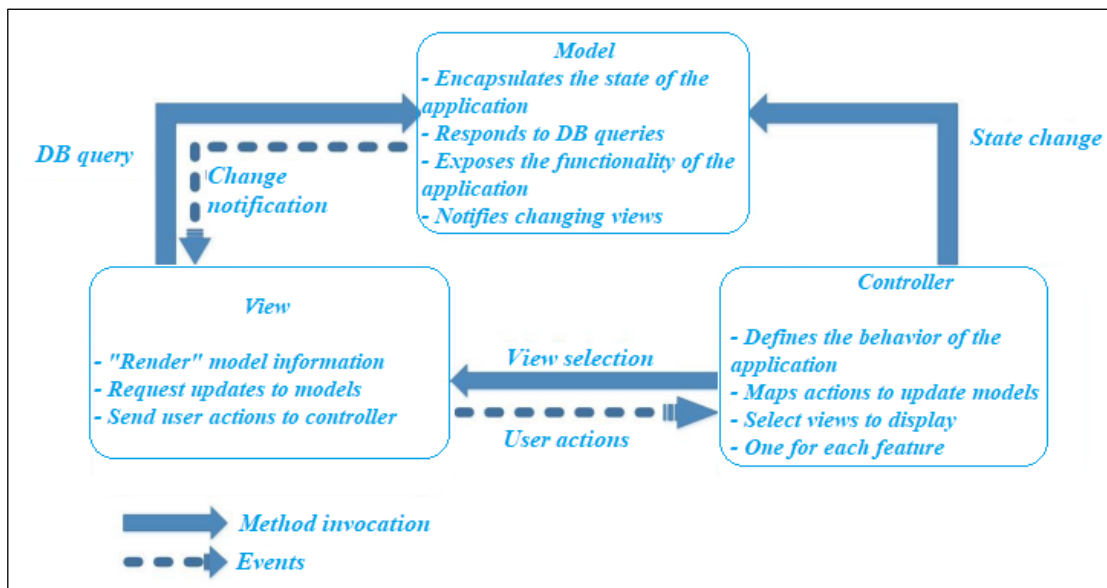


Figure 1: MVC Model.

Source: [14].

The model layer is responsible for encapsulating the data and functionality of the software. The component acts as a handler, intercepting all the operations involved in processing this data. This class of model is also known as domain; usually, the model has no idea about the user's interactions with the interface [15]. Still, the view component is responsible for displaying the information and deciding how this information will be displayed to the user [13].

The controller component is responsible for managing the interactions of the user with the interface, such as: key presses, mouse or interaction with the visible elements and passing the information to the view and the model so that the data are persisted [13]. Thus, MVC architecture assists developers in building applications by separating its main components, the data manipulation and storage, the functions that will work with the data inputs and the user view. The MVC architecture specifies where each type of logic should be located in the application [16].

II.3 AUTOMATED TESTS

Testing is an intrinsic practice to the development and is old, the need to create scripts or programs to test specific scenarios, especially for scenarios in which testing manually was impractical. However, using automated testing as a basic premise of development is a relatively recent phenomenon, beginning in the mid-1990s [17].

Currently, test automation is considered by several methodologies, especially among those identified with agile methods, as a basic practice to ensure the quality of a software. Much has been studied, new concepts and development techniques have been created and the tools are increasingly powerful and practical. All this facilitates not only the writing of tests for specific code snippets, but also for the integration of modules, graphical interfaces and databases [17].

According to ISTQB, testing is not just about executing tests and analyzing their results. This is a much more complex process that involves several steps such as test planning, analysis, design, test implementation, creating test progress reports, test results and analyzing the quality of a test object. It also involves reviewing the requirements and code. In addition to performing all this verification, this process also includes the validation of the system, ensuring that the system meets customer needs [18].

However, test automation reduces efforts in repetitive tasks, so not all tests should be automated. Before inserting this process in a project you should analyze what is the volume of tests that your software needs and the time that is spent to execute these batteries of tests. If the project in question has a very small complexity, the effort to insert this testing methodology can be more costly than the maintenance of manual testing. Added to that, the time to execute the automatic tests would not be, in this case, so less than the manual ones [19].

II.4 BASIC NOTIONS OF THE FACE

The face is the frontal part of the head, extending from the forehead to the chin, composed of the mouth, the nose, the eyes and the eyebrows. Through this set, they can convey emotions through expression, in which the individual is in his state of mind, that is, how the person feels at that moment [21].

II.4.1 Symmetry of the Face

When we talk about symmetry, immediately comes to mind the adjective of perfection, beauty, proportion, this because in the ancient age it was already sought symmetry through components

of harmonization of the physical lines of beauty in the faces considering them as beautiful.

According to the egregius theory of proportions inherited from the Roman architect Vitruvius, the proportional man has a face divided horizontally into three symmetrical parts [20]. The regions are classified as: mental, affective and instinctive [21].

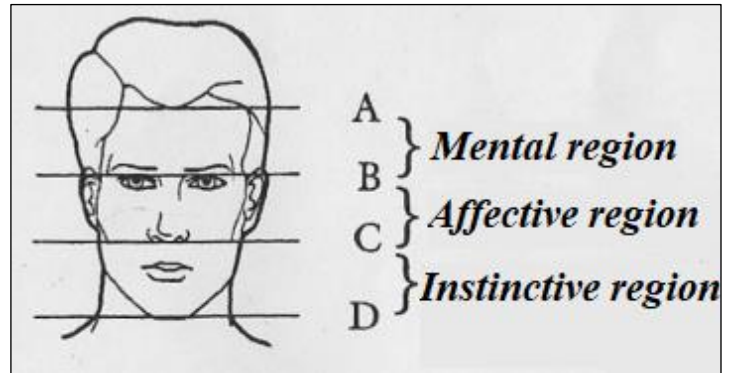


Figure 2: View of the three symmetrical parts of the human face. Source: [21].

AB mental region starts in the hair up to the eyebrows. Affective region BC starts at the eyebrows up to the nose aculeus.

The CD instinctive region starts at the aculeus of the nose and goes up to the chin.

According to the division, the width of the face should measure two-thirds of its length. The distance between the eyes should correspond exactly to the width of the nasal diameter, well aligned horizontally [22].

II.5 DIGITAL IMAGING

The image can be defined as a two-dimensional function, $f(x,y)$, where x and y are spatial coordinates (plane), and amplitude of f at any pair of coordinates (x,y) known as intensity or gray level of the image at that point. When x,y and the intensity values of f are finite and discrete quantities, called a digital image. Note that a digital image is composed of a finite number of elements, each with its specific value and location. These elements are called pictorial elements, elements that make up the image, pels and pixels. Pixels is the term called to represent an element of a digital image [23].

II.6 IMAGE PROCESSING

The processing of an image both input and output must be manipulated so that in the end it becomes an image, in the computational context this image is converted into data. After processing, these data are captured and subjected to techniques to obtain the variation of images that can be used in other processes [24].

Digital image processing goes through 3 levels up to computer vision, such as: low-level, medium-level and high-level [23]. According to [16], the low-level operates in noise reduction or improvement in image contrast. The mid-level is the segmentation or classification process. And finally, the high-level is the relation of the cognition task directly associated with human vision.

II.6.1 Biometrics

Every human being is composed of distinct characteristics such as: physical or behavioral that make them unique.

Identification through computer systems are means for automatic control of entry of reserved environments, records of activities, surveillance, etc., [25]. Therefore, biometrics is considered the science that measures the characteristics of each individual, classifying it as unique and exclusive through physical, chemical and behavioral attributes [26].

For [27] biometrics is divided into two categories, being the physical and behavioral category, which can be measured using different metrics. The first one fits, for example, in fingerprint, DNA's, face, etc. The second would refer to the psychological state, where, depending on the situation in which the person is, there would be a change in speech, walk, way of expressing and acting in general.

II.6.2 Facial Recognition

The face recognition system is a biometric that uses the face of a person through a detection equipment and compares it with an image (photo) stored in a database. This comparison is made using the detection and adjustment of high points, low points and the contour present on the face [40].

Face recognition is the process of recognizing and naming (labeling) a face. [29]. This process is a computational resource that recognizes the face and transforms it into an image.

The facial recognition uses a set of images that have been collected in a certain time, with this facilitating in the search for characteristics of the face and distinguishing it in relation to the rest of the environment, these facial photographs have to be contained in official documents, making it authentic and decreasing the invasion, making it cheaper to obtain the images. [30]. However, it can present as a disadvantage the possibility of suffering interference in its operation, due to the conditions of the environment, such as lighting, and people, such as: position of the individual, age and emotional state; with this increases the level of difficulty of implementation [31].

II.7 AGILE METHODOLOGIES

A group of seventeen *software* specialists in 2001, were discussing ways to improve the performance of their projects, presenting several agile methods as options for their software, thus was born there the term called "Agile Methodology". In this meeting they observed in their previous experiences that there was a set of common principles, that when followed their projects worked well. Thus emerged the agile alliance that established the agile Manifesto where it brought its values, its philosophy and its principles.

Agile methods are a set of practices that follow the principles of the Agile Manifesto [32]. The agile manifesto is not against traditional models [33]. Agile methods are based on incremental development [13], which contains techniques used in software engineering. However, its pattern is not proposed by traditional methodology [34].

Agile methodologies use new class of methodology developed to meet the growing market demand through more agile and light processes in increasingly shorter development cycles [35]. These methodologies are focused on individuals and the interaction between them, on the functioning of the software, on the interactivity with the customer and with fast response to changes. In this way, it is possible to be more agile by adapting the requirements with short-term delivery objectives by the time it enters the development phase [15].

An important feature found in all Agile Methods is the recognition of *software* development as a process based on experience and observation [36].

Some more used agile methodology that has the collaboration of the client and the rapid response to change and amendments, because besides having common base of their conceptions, with engineering and management, their results are satisfactory when applied, they are them: *Scrum*, *Extreme Programming* and the *Lean Software Development* [37].

II.7.1 Extreme Programming

Extreme Programming (XP) is an agile methodology for small and medium teams that develop *software* based on vague requirements and that change quickly [38]. It is a software development methodology to meet the need of customers through its speed, productivity and quality in a simple way. It is a methodology for the development, where the requirements are constantly changing [39].

III. MATERIALS AND METHODS

III.1 MATERIALS

To develop the computational solution of the problem raised in the research, various computers were used. The list of materials and peripherals that we used in the development of this project, are presented in Table 1, presented below.

Table 1: List of peripherals used.

<i>PCs</i>	<i>Laptops</i>
Processor: <i>Intel I5 9400f</i> ; Dedicated video card: <i>GTX 1650</i> , Primary Memory: <i>Asgard RGB 16GB(2 x 8GB dual-channel)</i> DDR4 and frequency 2666mhz; Operating System: <i>Windows 10 Pro</i>	Processor: <i>Intel i7 11th</i> generation; Primary Memory: 8GB; Secondary Memory: <i>SSD 250GB</i> ; Operating System: <i>Linux Ubuntu 20.04 LTS</i> ; Model: <i>Dell vostro 15 3500</i>
Processor: <i>Ryzen 7 3700x</i> ; Dedicated video card: <i>XFx QIck RADEON RX 6600XT</i> ; Primary Memory: <i>Asgard RGB 16GB(2 x 8GB dual-channel)</i> DDR4 and 3200mhz frequency; Secondary Memory: <i>Kingston 1 Terabyte capacity NVME M.2 SSD</i> ; Operating System: <i>Linux Kernel, Fedora 36</i> distribution	

Source: Authors, (2022).

A *Webcam* embedded in each notebook was used as a testing parameter in facial recognition. In the case of *PCs*, we used a dedicated device and a *smartphone*.

Dedicated *Webcam*: *ASHU H701*, Full HD resolution and video capture up to 30 frames per second.

III.2 METHODS

III.2.1 Development Project

The project consists of two applications communicating through *HTTP* requests. Thus, it was possible to obtain scenarios for each type of applicability under development. First, it was necessary to include a *web* solution to store the information of each student such as: registration number, name, enrollment number, identifier (CPF or RG), series, class and shift, besides having a photo gallery of the student in different positions. Furthermore, the

system will send notification to the parents or guardians of each student, which means that the parents' information, such as e-mail, will also be stored. In this scenario, a modern software development solution with *Ruby* programming language was used. To adhere to the best practices of *web* development in *Ruby on Rails*, the code followed "The Rails Way" principles, this went beyond its philosophy of "Don't Repeat Yourself (DRY) and Convention Over Configuration".

The paradigm is Object Oriented and with an *MVC* design pattern, moreover, it is strongly recommended by the community to use automated tests. It is possible to add *gems* to projects, which are resources that allow a more pragmatic development, since simplicity is the main focus of the *Ruby* ecosystem. With this, we design a modern Web application without worrying about major bureaucracies, as a counterpoint to classic languages. In the context of automated testing, for example, we use *RSpec* that validates the system units, as well as the functional requirements and if the system as a whole meets what is proposed.

Finally, the context of greatest value of the system, is the service written in *Python* with the popular tool *OpenCV*. The **open source Computer Vision Library** accumulates several algorithms for **Computer Vision**. Essentially, it is an **API** written in *C++*, and can be used in specific purpose projects with programming languages like *Python*, *Java*, among others.

There are several solutions for *OpenCV*, they are: facial recognition, automated inspection and surveillance, counting the number of people (pedestrian traffic in a *mall*, etc.), defect detection in the manufacturing process, object recognition, medical image analysis, among several other applications. For the purpose of this project, this tool is of paramount importance in facial recognition of each student, since its responsibility is to detect and send data to a *web* application, i.e., based on a battery of registered photos of the student, the system will be able to validate the personal integrity to assume the authentication of the student in the classroom.

Based on the modularization that the tool has, we used to make the development of the second application simpler. With this, we obtained two applications communicating with each other, on the managerial side, a platform written with *Ruby On Rails* that uses a more active and visible context service to the students that will be developed in *Python* and *OpenCV*, being each tool responsible for a solution of our research.

III.2.2 Project Infrastructure

In the context of a team product development, it is of utmost importance that the main tools used are shared among each team member, so that the workflow is efficient. Considering the need for versioning, the source code was hosted on the *GitHub* platform, which according to its own documentation, is defined by offering free and paid products aiming to store and collaborate on the developers' code. With this we used two repositories, one for each research application, being the student attendance management system and the facial recognition service.

Some resources of *software* methodology were fundamental to ensure the security of a new implementation, we emphasize in certain essential aspects: At each *commit* to the remote repository the Automated Tests and security validations must be performed. Once properly validated.

With this routine from the development members, we schematize the *CI* of the project. Detailing the tools, *GitHub* goes beyond hosting and versioning source code. One Continuous Integration automation tool for *Ruby* is *RubyCI*, providing a simple setup for the community and delivering features specific to the best practices of an efficient *Rails* infrastructure. This allowed us to configure security executors and automated tests for the student attendance management system repository. It is worth noting that we will be using a relational database with *PostgreSQL* as the *DBMS*.

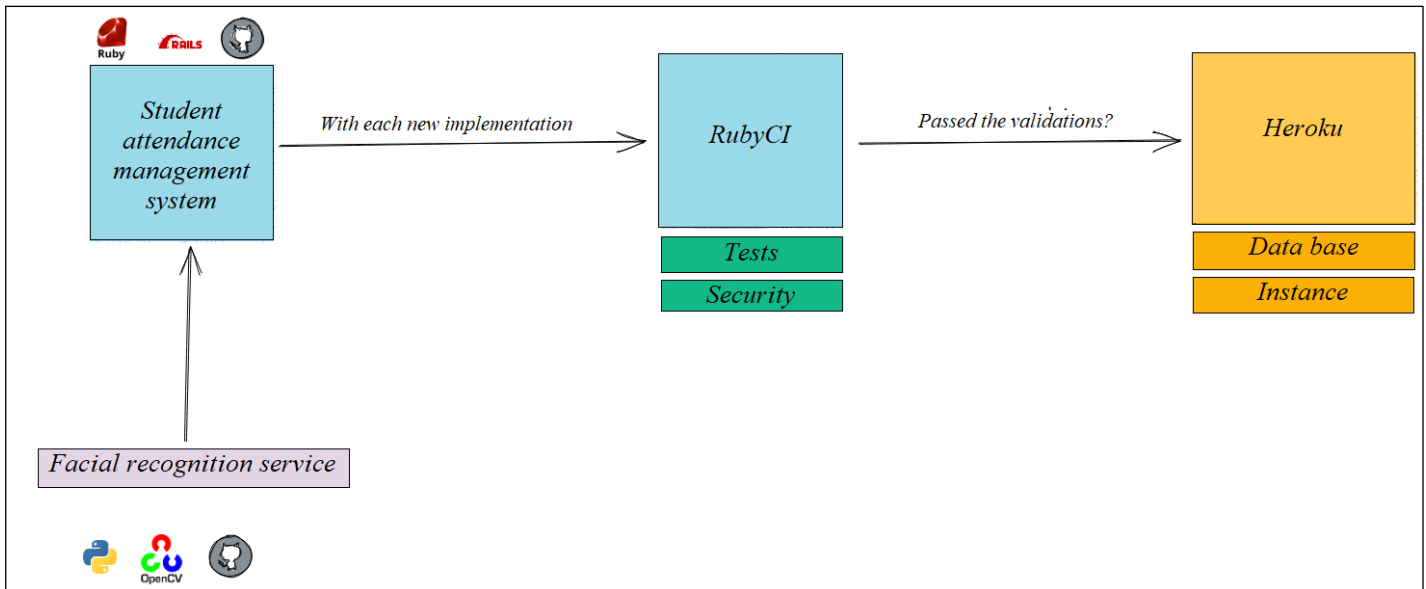


Figure 3: Project development flow.
Source: Authors (2022).

III.2.3 Agile Methodology

We used the *XP* agile methodology to have a more assertive, simplified and fluid approach in the project development activities. During the development, small tasks were made so that the junction

of them will result in the complete system. This approach, that is, the creation of small tasks, is known as "divide and conquer", widely used in the *Extreme Programming* methodology. Thus, we held weekly meetings to discuss possible solutions to the activities

created and ensure the general understanding regarding the product.

IV. RESULTS AND DISCUSSIONS

Initially, an *online* survey was raised directed to education professionals working in the city of Manaus. It was obtained in total 21 responses, indicating the type of institution and performance coming from each interviewee, among other data relating to the educational context. It is noted that 55% of professionals in the area of education who responded to the survey are from public institutions, while 45% of them were from private schools or universities. Furthermore, experience in modalities were recorded: face-to-face, hybrid and distance learning. However, Figure 4 demonstrates what the small shows the approach used in the task of recording the students' attendance, where, in some cases, it is possible to note the exclusive applicability of the manual form.

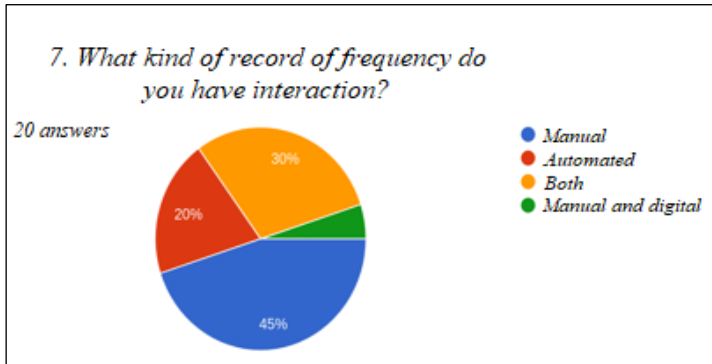


Figure 4: Type of Frequency Record Used. Source: Authors (2022).

IV.1 FACIAL RECOGNITION

The facial recognition application is a dependency of the system, so its only functionality is to receive, process and return

data. We adopted a library called "face_recognition", which has all the necessary requirements to achieve the goal of recording student attendance. In Figure 4 we have an example of its functionality with a face that identifies itself as "Yves". In this case, we used an input data (Photo of "Yves"), which processes the recognition from the sent photo, associating it with its respective identifying name. The output showed that "Yves" was identified, even if using mask.

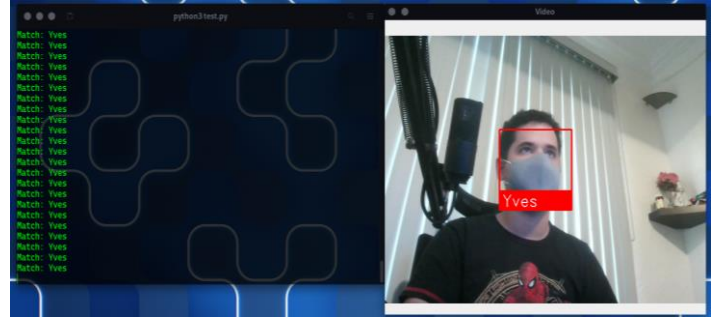


Figure 4: Example of Face Recognition. Source: Authors (2022)

IV.1.1 Modeling the System

First, the models *Student*, *SchoolClass*, *Responsible*, *FrequencyRegister* and *Register* were created. Furthermore, the cardinality was designed with the following logic:

- A *SchoolClass* can have many *Students*
- A *Student* may have only one *SchoolClass*
- A *Student* can have only one *Responsible*
- A *Responsible Party* may be responsible for several *Students*
- A *SchoolClass* can have many *FrequencyRegisters*
- A *FrequencyRegister* is related to a *SchoolClass*
- One *FrequencyRegister* can Register many *Students*
- *Students* may be present in multiple *FrequencyRegisters*

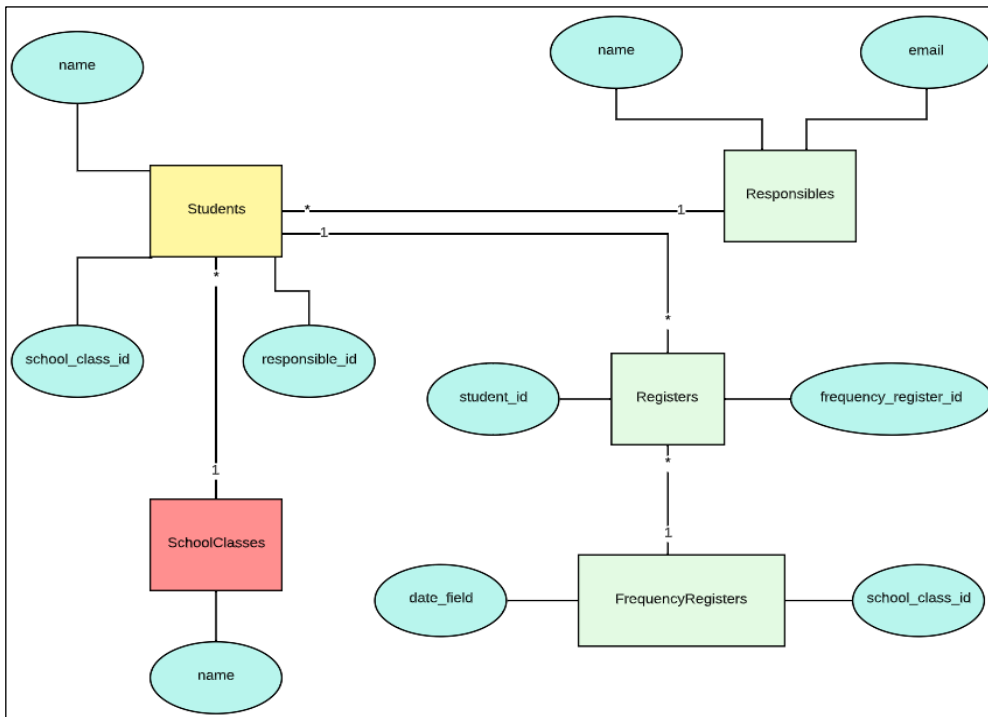


Figure 5: Data Modeling. Source: Authors (2022).

IV.2 PROTOTYPE SYSTEM

Figure 6 shows the initial screen of the system, where you can see the name of the application, a short description and a navigation menu for the main "Roles" of interaction. Each section lists the records of the related Object, i.e., by clicking on "Students", we have as a response a catalog of students of the application, the same idea applies to the others. Another point is the functionality to create a new resource, present on all pages in a manner adapted to each context, for example, in the "Responsible" page is noticeable a button to create a Responsible.

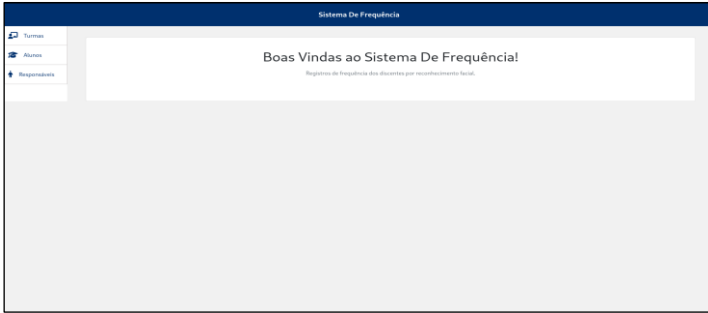


Figure 6: System Home Page.
Source: Authors (2022).

To reproduce a real scenario of the system's applicability, you must first register a Class by clicking on "Create a Class" and writing its name, for example, "Computer Science". Then, the next step is to register a responsible person, identifying him as "Yves César" and e-mail "yves_responsavel@gmail.com".

Therefore, the process of registering a student is initiated. The requirements of a student is:

- Standardized 3x4 photo, with good lighting and focus on the face.
- Name I.D.
- A responsible one.
- A class.

In this case, you can create a student with the name "Lais Reis" being the responsibility of "Yves César" and participant of the class "Computer Science". Once your registration is complete, you can start. Clicking on "Create a record of attendance", the computer camera is triggered to perform facial recognition, in this example is notable the identification of the student Lais Reis and its respective ID.

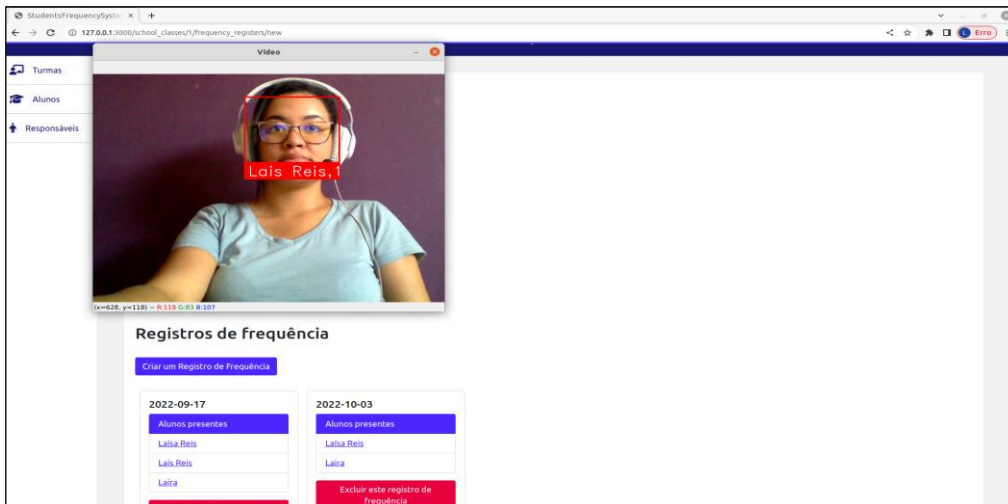


Figure 7: "Lais" Identified.
Source: Authors (2022).

By pressing the "q" key to finish running the attendance register, you can view the list of students present and a Date. By

adding this record, all the students' guardians will be notified via email about the presence in the class.

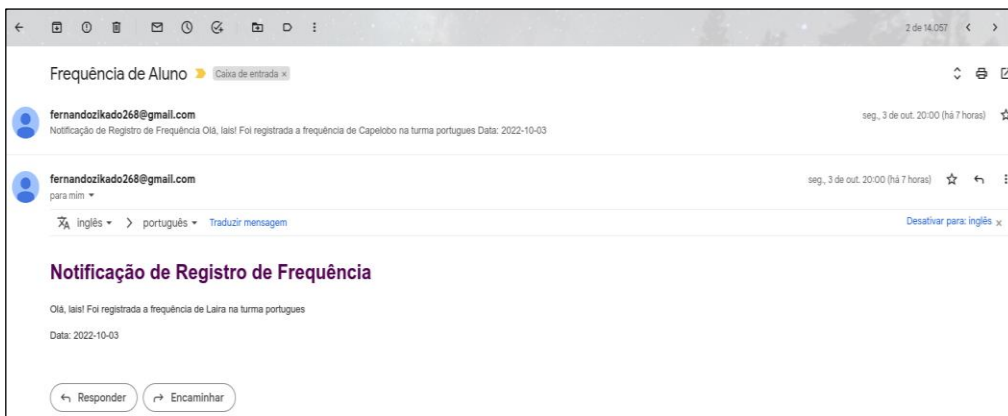


Figure 8: Frequency Notification.
Source: Authors (2022).

V. CONCLUSION

With the data survey on how schools or colleges of Manaus register their students' attendance, it can be observed that we still have failures of security and productivity of students and teachers. Among the interviewees' reports, it is generally used the traditional approach, recording manually, as well as, having the help of a system for registration. However, this model can still be problematic due to erasures, forgery of signatures or too much energy expenditure of those involved with this type of activity.

Facial Recognition is a very significant help regarding the issue of student safety, because identifying via camera the tendency is that this flow is more accurate and less bureaucratic. Moreover, it becomes possible to give the fastest *feedback* to the responsible of the student, and less energy expenditure of teachers and propensity to human error. As a limit of this project, we can highlight that the face detection tool needs standardized photos and with good lighting, so that the algorithm can in fact map the face, another problem could be the issue of twin brothers having similar traits.

In the development process, it was possible to maintain the good modern testing practices, making it safer to add each implementation in the project without losing the integrity of the functional requirements. Thinking in *Extreme Programming*, it was possible to have a more agile coding, aiming, at each *Feature*, what would bring greater value to the project in a simple way. The *Ruby On Rails Framework* was of paramount importance in terms of productivity and alignment of the methodology used, due to its ideology of making coding more objective.

Thus, the system of registering students' attendance with facial recognition was suggested as a prototype to solve the problem raised in the research, listing the identified students and notifying parents or guardians, besides providing the necessary functionality to create, edit and delete data in each part involved in the application.

VI. AUTHOR'S CONTRIBUTION

Conceptualization: Yves César Amorim de Azevedo, Laís Reis de Jesus, Wilson Fogaça de Araújo and Fernando Filipe Da Silva Costa.

Methodology: Yves César Amorim de Azevedo, Laís Reis de Jesus, Wilson Fogaça de Araújo, Fernando Filipe Da Silva Costa and Alexandra Amaro de Lima.

Investigation: Yves César Amorim de Azevedo, Laís Reis de Jesus, Wilson Fogaça de Araújo and Fernando Filipe Da Silva Costa.

Discussion of results: Yves César Amorim de Azevedo, Laís Reis de Jesus, Wilson Fogaça de Araújo and Fernando Filipe Da Silva Costa.

Writing – Original Draft: Yves César Amorim de Azevedo, Laís Reis de Jesus, Wilson Fogaça de Araújo and Fernando Filipe Da Silva Costa.

Writing – Review and Editing: Eliton Smith and Alexandra Amaro de Lima.

Resources: Eliton Smith and Alexandra Amaro de Lima.

Supervision: Alexandra Amaro de Lima.

Approval of the final text: Yves César Amorim de Azevedo, Laís Reis de Jesus, Wilson Fogaça de Araújo, Fernando Filipe Da Silva Costa, Eliton Smith and Alexandra Amaro de Lima.

VI. ACKNOWLEDGMENTS

We would like to thank our teacher and mentor Alexandra Amaro de Lima for the attentive accompaniment in the development process of this work.

VII. REFERENCES

- [1] D. J. L. Setúbal, "A importância do ensino presencial - Blog Saúde Infantil," *Instituto PENSI - Estudos Clínicos em Pediatria e Saúde Infantil*, Sep. 02, 2021. <https://institutopensi.org.br/a-importancia-do-ensino-presencial/> (accessed Sep. 20, 2022).
- [2] E. Marais, D. Argles, and B. von Solms, "Security Issues Specific to e-Assessments," presented at the 8th Annual Conference on WWW Applications (07/09/06), 2006. Accessed: Oct. 10, 2022. [Online]. Available: <https://eprints.soton.ac.uk/261433/>
- [3] L. Mukuno, E. T. Moraes, R. M. Haddad, and E. C. Almeida, "Aplicando modelo de aprendizagem supervisionada para apoio ao score de autenticação biométrica," in *Anais do Seminário Integrado de Software e Hardware (SEMISH)*, Jul. 2021, pp. 201–206. doi: [10.5753/semish.2021.15823](https://doi.org/10.5753/semish.2021.15823).
- [4] A. Netto and F. Maciel, *Python Para Data Science: E Machine Learning Descomplicado*. Alta Books, 2021.
- [5] C. P. Pertuz, *Aprendizaje automático y profundo en python: Una mirada hacia la inteligencia artificial*. Ediciones de la U, 2022.
- [6] A. Sobral, "BGSLibrary: An OpenCV C++ Background Subtraction Library," *IX Workshop de Visão Computacional (WVC'2013)*, Jun. 2013, Accessed: Oct. 10, 2022. [Online]. Available: https://www.academia.edu/5615730/BGSLibrary_An_OpenCV_C_Background_Subtraction_Library
- [7] A. M. Torres, "Ruby on Rails como ambiente de desenvolvimento de um sistema para gerenciamento das etapas associadas ao processo de tipagem HLA praticado pelo laboratório de HLA da UERJ," p. 116, 2018.
- [8] A. M. Torres, "Ruby on Rails como ambiente de desenvolvimento de um sistema para gerenciamento das etapas associadas ao processo de tipagem HLA praticado pelo Laboratório de HLA da UERJ," *Ruby on Rails as a development environment for a management system of the steps associated with the HLA typing process practiced by the HLA laboratory of UERJ*, Aug. 2018, Accessed: Oct. 01, 2022. [Online]. Available: <https://www.bdt.uerj.br:8443/handle/1/7669>
- [9] "Documentation." <https://www.ruby-lang.org/en/documentation/> (accessed Oct. 10, 2022).
- [10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning." <https://www.deeplearningbook.org/> (accessed Sep. 15, 2022).
- [11] D. E. King, "Dlib-ml: A machine learning toolkit," *The Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.
- [12] J. Galloway, P. Haack, B. Wilson, and K. S. Allen, *Professional ASP.NET MVC 5*. John Wiley & Sons, 2014.
- [13] "(82) Ian Sommerville - Engenharia de Software 9 ed | Natalia Graciano - Academia.edu." https://www.academia.edu/42787809/Ian_Sommerville_Engenharia_de_Software_9_ed (accessed Aug. 07, 2022).
- [14] M. F. de Lemos, P. C. Oliveira, L. C. Ruela, M. da S. Santos, T. C. Siveira, and J. C. de S. Reis, "APLICABILIDADE DA ARQUITETURA MVC EM UMA APLICAÇÃO WEB(WebApps)," *RE3C - Revista Eletrônica Científica de Ciência da Computação*, vol. 8, no. 1, Art. no. 1, Nov. 2013, Accessed: Oct. 10, 2022. [Online]. Available: <https://revistas.unifenas.br/index.php/RE3C/article/view/54>
- [15] M. Flower, "GUI Architectures," *martinfowler.com*. <https://martinfowler.com/eaDev/uiArchs.html> (accessed Oct. 13, 2022).
- [16] R. M. Santos, "Um-Estudo-de-Processamento-de-Imagens-com-OPENCV", Accessed: Oct. 13, 2022. [Online]. Available: https://www.academia.edu/8424043/Um_Estudo_de_Processamento_de_Imagens_com_OPENCV

- [17] P. C. Bernardo, "Padrões de testes automatizados," text, Universidade de São Paulo, 2011. doi: [10.11606/D.45.2011.tde-02042012-120707](https://doi.org/10.11606/D.45.2011.tde-02042012-120707).
- [18] M. R. S. Monteiro, "Desenvolvimento de testes automatizados para frontend," masterThesis, 2021. Accessed: Oct. 14, 2022. [Online]. Available: <https://run.unl.pt/handle/10362/123471>
- [19] L. R. P. Izabel, "TESTES AUTOMATIZADOS NO PROCESSO DE DESENVOLVIMENTO DE SOFTWARES," p. 65, 2014.
- [20] C. N. Camargos, C. A. Mendonça, and S. M. Duarte, "Da imagem visual do rosto humano: simetria, textura e padrão," *Saude soc.*, vol. 18, pp. 395–410, Sep. 2009, doi: [10.1590/S0104-12902009000300005](https://doi.org/10.1590/S0104-12902009000300005).
- [21] MARTINEZ, Valquiria. "Mistérios do rosto: Manual de Fisiognomonía". 1997.
- [22] L. G. Farkas, "Accuracy of Anthropometric Measurements: Past, Present, and Future," *The Cleft Palate-Craniofacial Journal*, vol. 33, no. 1, pp. 10–22, Jan. 1996, doi: [10.1597/1545-1569_1996_033_0010_aoampp_2.3.co_2](https://doi.org/10.1597/1545-1569_1996_033_0010_aoampp_2.3.co_2).
- [23] R. C. Gonzalez and R. E. Woods, *Processamento de Imagens Digitais*. Editora Blucher, 2009.
- [24] G. Câmara, R. C. M. Souza, U. M. Freitas, and J. Garrido, "Spring: Integrating remote sensing and gis by object-oriented data modelling," *Computers & Graphics*, vol. 20, no. 3, pp. 395–403, May 1996, doi: [10.1016/0097-8493\(96\)00008-8](https://doi.org/10.1016/0097-8493(96)00008-8).
- [25] J. I. S. da Silva, "Reconhecimento facial em imagens de baixa resolução," Feb. 24, 2015. <https://repositorio.ufpe.br/handle/123456789/16367> (accessed Oct. 13, 2022).
- [26] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics*. Springer Science & Business Media, 2007.
- [27] Department of Computer Science & Engineering Roorkee Institute of Technology, Roorkee (U.K.), India and H. Srivastava, "A Comparison Based Study on Biometrics for Human Recognition," *IOSR-JCE*, vol. 15, no. 1, pp. 22–29, 2013, doi: [10.9790/0661-1512229](https://doi.org/10.9790/0661-1512229).
- [28] Department of Computer Science & Engineering Roorkee Institute of Technology, Roorkee (U.K.), India and H. Srivastava, "A Comparison Based Study on Biometrics for Human Recognition," *IOSR-JCE*, vol. 15, no. 1, pp. 22–29, 2013, doi: [10.9790/0661-1512229](https://doi.org/10.9790/0661-1512229).
- [29] Romdhani, Sami. "Face recognition using principal components analysis." *University of Glasgow (Dissertation)*, 1996.
- [30] Junior, A. and Filho, M. "Aplicações de Processamento de Imagens a Sistemas de Segurança," p. 21. 2002.
- [31] P. S. T. Magalhães and H. D. dos Santos, "Biometria e autenticação," Oct. 2003. Accessed: Oct. 13, 2022. [Online]. Available: <http://repositorium.sdum.uminho.pt/>
- [32] "Manifesto para Desenvolvimento Ágil de Software." <http://agilemanifesto.org/iso/ptbr/manifesto.html> (accessed Oct. 13, 2022).
- [33] D. Ludvig and J. D. Reinert, "Estudo do uso de Metodologias Ágeis no Desenvolvimento de uma Aplicação de Governo Eletrônico," p. 157.
- [34] Jim, H. "Retiring Lifecycle dinosaurs: Using Adaptive Software Development to meet the challenges of a high-speed, high change environment". 2000.
- [35] P. Abrahamsson, J. Warsta, M. T. Siponen, and J. Ronkainen, "New directions on agile methods: a comparative analysis," in *Proceedings of the 25th International Conference on Software Engineering*, USA, May 2003, pp. 244–254.
- [36] L. Williams and A. Cockburn, "Agile software development: it's about feedback and change," *Computer*, vol. 36, no. 6, pp. 39–43, Jun. 2003, doi: [10.1109/MC.2003.1204373](https://doi.org/10.1109/MC.2003.1204373).
- [37] A. Hamed and H. Abushama, "Popular agile approaches in software development: Review and analysis," Aug. 2013, pp. 160–166. doi: [10.1109/ICCEEE.2013.6633925](https://doi.org/10.1109/ICCEEE.2013.6633925).
- [38] K. Beck, *Programação Extrema (XP) Explicada*. Bookman, 2004.
- [39] T. B. Pontes and D. D. B. Arthaud, "METODOLOGIAS ÁGEIS PARA O DESENVOLVIMENTO DE SOFTWARES," *Ciência e Sustentabilidade*, vol. 4, no. 2, Art. no. 2, 2018, doi: [10.33809/2447-4606.422018173-213](https://doi.org/10.33809/2447-4606.422018173-213).
- [40] M. de S. Dias, "Autenticação Segura Baseada em Biometria Voltada para a Dinâmica da Digitação," p. 70, 2009.