Faculty of Computers and Information – Suez University

Requirements Document

For Face Recognition Attendance System

Team 2
2021

Under Supervision of

Dr. Haitham Farouk

Dr. Hussien Sharaf

With Assistance of

Eng. Amany Draz

Eng. Rania Mahmoud

Team Members

Mark Adel Rateb

Abd El-Rahman Mohammad Mosa

Mahmoud Ramadan Mohammed

Ahmed Mahrous Abbas

Menna-Allah Mohammad Shazly

Nada Nasr Mohammad

Abd El-Rahman Raafat Amin

Preface

This document acts a user manual or cheat note to some college managerial operations automation through an integrated computer system. The system target to make these operations more efficiently with more facilities and capabilities in order to reduce the time, effort, and other resources as well as treating conflictions in time and attendance management between courses instructors and students.

The main reason of this system is to isolate and automate managerial and operations processes by making these processes automated and isolated to leave most of available space to educational activities, to get the optimal beneficial use of available resources to instructors as well as students.

In order to get the system in the most sophisticated form, this required to go several versions of software program development and documentation to user and system requirements to achieve customer satisfaction within available resources and budget.

The following table demonstrate a brief comparison between different system visions as follows:

1.0	Only a GUI interface with a temporary data records instead of design and implementation of actual database through database management system
2.0	Integrate an application GUI with an actual Database design through SQL lite engine by sqlalchemy
3.0	All previous versions were based on uploading students images to the website in this version we add a feature that the student can take a live image in the real time
4.0	To improve image matching algorithms acceleration, we add video streaming capabilities to ensure data integrity
5.0	Maintain system security issues and resolve the remaining bugs and problems

Introduction

The proposed system will be capable to perform the following operations:

1. Manage college student attendance in an automated manner
2. Organize student's attendance in a hierarchal form to provide efficient representation and ease of use within a service orientation towards the instructors to make them do the minimal effort in order to save their time

It consists of three modules: microprocessor-based system, web API, and user GUI web interface to interact with the system through web GUI and microprocessor-system peripherals to achieve system usability and user-friendly experience.

We used the following tools and frameworks:

1. Flask
2. Sqlalchemy
3. OpenCV
4. numPy
5. Face detection
6. Face Recognition

To provide a sufficient tool kit to the project, to ensure efficient performance and sophisticated results as well.

Glossary

Acronym/ Expression	Description
Microprocessor	It is a computer processor that is implemented on a single (or more) integrated circuit (IC) dies of MOSFET construction in a single package. The microprocessor is a multipurpose, clock-driven, register-based, digital integrated circuit that accepts binary data as input, processes it according to instructions stored in its memory, and provides results (also in binary form) as output. Microprocessors contain both combinational logic and sequential digital logic.
API	An application programming interface (API) is a computing interface that defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc. It can also provide extension mechanisms so that users can extend existing functionality in various ways and to varying degrees.
REST web service	Representational state transfer (REST) is a software architectural style that defines a set of constraints to be used for creating Web services. Web services that conform to the REST architectural style, called RESTful Web services, provide interoperability between computer systems on the internet. RESTful Web services allow the requesting systems to access and manipulate textual representations of Web resources by using a uniform and predefined set of stateless operations.
SOAP web service	SOAP (formerly an acronym for Simple Object Access Protocol) is a messaging protocol specification for exchanging structured information in the implementation of web services in computer networks. Its purpose is to provide extensibility, neutrality, verbosity and independence. It uses XML Information Set for its message format, and relies on application layer protocols, most often Hypertext Transfer Protocol (HTTP), although some legacy systems communicate over Simple Mail Transfer Protocol (SMTP), for message negotiation and transmission.

Raspberry Pi 4 Model B	A single-board computer that was released in February 2016 with a 1.2 GHz 64-bit quad core ARM Cortex-A53 processor, on-board 802.11n Wi-Fi, Bluetooth and USB boot capabilities.
ARM processor	ARM (stylized in lowercase as arm, previously an acronym for Advanced RISC Machines and originally Acorn RISC Machine) is a family of reduced instruction set computing (RISC) architectures for computer processors, configured for various environments. Due to their low costs, minimal power consumption, and lower heat generation than their competitors, ARM processors are desirable for light, portable, battery-powered devices—including smartphones, laptops and tablet computers, as well as other embedded systems.
RISC Machine	A reduced instruction set computer, or RISC, is a computer with a small, highly optimized set of instructions. The main distinguishing feature of RISC architecture is that the instruction set is optimized with a large number of registers and a highly regular instruction pipeline, allowing a low number of clock cycles per instruction (CPI). Another common RISC feature is the load/store architecture, in which memory is accessed through specific instructions rather than as a part of most instructions in the set
Web Service	▪ A service offered by an electronic device to another electronic device, communicating with each other via the World Wide Web, or ▪ A server running on a computer device, listening for requests at a particular port over a network, serving web documents (HTML, JSON, XML, images), and creating web applications services, which serve in solving specific domain problems over the Web (WWW, Internet, HTTP)
Client–server model	Client–server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. Often clients and servers communicate over a computer network on separate hardware, but both client and server may reside in the same system. A server host runs one or more server programs, which share their resources with clients. A client does not share any of its resources, but it requests content or service from a server. Clients, therefore, initiate communication sessions with servers, which await incoming

	requests. Examples of computer applications that use the client-server model are email, network printing, and the World Wide Web.
MVC	Model–view–controller (MVC) is a software design pattern commonly used for developing user interfaces that divides the related program logic into three interconnected elements. This is done to separate internal representations of information from the way's information is presented to and accepted from the user. Traditionally used for desktop graphical user interfaces (GUIs), this pattern has become popular for designing web applications.
Software Architecture	The architecture of a system is its 'skeleton'. It's the highest level of abstraction of a system. What kind of data storage is present, how do modules interact with each other, what recovery systems are in place.
Software Design	Software design is about designing the individual modules / components. What are the responsibilities, functions, of module x? Of class Y? What can it do, and what not? What design patterns can be used?
So, in short, Software architecture is more about the design of the entire system, while software design emphasizes on module / component / class level.	
Face Detection	Face detection is a computer vision problem that involves finding faces in photos. It is a trivial problem for humans to solve and has been solved reasonably well by classical feature-based techniques, such as the cascade classifier.
Face Recognition	It is a technology capable of matching a human face from a digital image or a video frame against a database of faces, typically employed to authenticate users through ID verification services, works by pinpointing and measuring facial features from a given image.
NumPy	NumPY is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.
Flask	Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing

	third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.
SQL	Structured Query Language is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS). It is particularly useful in handling structured data, i.e., data incorporating relations among entities and variables. SQL offers two main advantages over older read–write APIs such as ISAM or VSAM. Firstly, it introduced the concept of accessing many records with one single command. Secondly, it eliminates the need to specify how to reach a record, e.g., with or without an index.
SQLAlchemy	SQLAlchemy is an open-source SQL toolkit and object-relational mapper (ORM) for the Python programming language released under the MIT License. SQLAlchemy's philosophy is that relational databases behave less like object collections as the scale gets larger and performance starts being a concern, while object collections behave less like tables and rows as more abstraction is designed into them. For this reason it has adopted the data mapper pattern (similar to Hibernate for Java) rather than the active record pattern used by a number of other object-relational mappers.
OpenCV	OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify

	objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.
Software Development Models	The software development models are the various processes or methodologies that are being selected for the development of the project depending on the project’s aims and goals. There are many development life cycle models that have been developed in order to achieve different required objectives. The models specify the various stages of the process and the order in which they are carried out. The selection of model has very high impact on the testing that is carried out. It will define the what, where and when of our planned testing, influence regression testing and largely determines which test techniques to use.
Python	Python is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.
HTTP cookie	An HTTP cookie (also called web cookie, Internet cookie, browser cookie, or simply cookie) is a small piece of data stored on the user's computer by the web browser while browsing a website. Cookies were designed to be a reliable mechanism for websites to remember stateful information (such as items added in the shopping cart in an online store) or to record the user's browsing activity (including clicking particular buttons, logging in, or recording which pages were visited in the past). They can also be used to remember pieces of information that the user previously entered into form fields, such as names, addresses, passwords, and payment card numbers. Cookies

	perform essential functions in the modern web. Perhaps most importantly, authentication cookies are the most common method used by web servers to know whether the user is logged in or not, and which account they are logged in with. Without such a mechanism, the site would not know whether to send a page containing sensitive information, or require the user to authenticate themselves by logging in. The security of an authentication cookie generally depends on the security of the issuing website and the user's web browser, and on whether the cookie data is encrypted.
WebSocket	The WebSocket protocol enables interaction between a web browser (or other client application) and a web server with lower overhead than half-duplex alternatives such as HTTP polling, facilitating real-time data transfer from and to the server. This is made possible by providing a standardized way for the server to send content to the client without being first requested by the client, and allowing messages to be passed back and forth while keeping the connection open. In this way, a two-way ongoing conversation can take place between the client and the server. The communications are usually done over TCP port number 443 (or 80 in the case of unsecured connections), which is of benefit for those environments which block non-web Internet connections using a firewall. Similar two-way browser-server communications have been achieved in non-standardized ways using stopgap technologies such as Comet.
Embedded system	An embedded system is a computer system—a combination of a computer processor, computer memory, and input/output peripheral devices—that has a dedicated function within a larger mechanical or electrical system. It is embedded as part of a complete device often including electrical or electronic hardware and mechanical parts. Because an embedded system typically controls physical operations of the machine that it is embedded within, it often has real-time computing constraints. Embedded systems control many devices in common use today. Ninety-eight percent of all microprocessors manufactured are used in embedded systems. Modern embedded systems are often based on microcontrollers (i.e., microprocessors with integrated memory and peripheral interfaces), but ordinary microprocessors (using external chips

	for memory and peripheral interface circuits) are also common, especially in more complex systems. In either case, the processor(s) used may be types ranging from general purpose to those specialized in a certain class of computations, or even custom designed for the application at hand. A common standard class of dedicated processors is the digital signal processor (DSP).
Wi-Fi	Wi-Fi is a family of wireless network protocols, based on the IEEE 802.11 family of standards, which are commonly used for local area networking of devices and Internet access. Wi-Fi uses multiple parts of the IEEE 802 protocol family and is designed to interwork seamlessly with its wired sibling Ethernet. Compatible devices can network through wireless access points to each other as well as to wired devices and the Internet. The different versions of Wi-Fi are specified by various IEEE 802.11 protocol standards, with the different radio technologies determining radio bands, and the maximum ranges, and speeds that may be achieved. Wi-Fi most commonly uses the 2.4 gigahertz (120 mm) UHF and 5 gigahertz (60 mm) SHF ISM radio bands; these bands are subdivided into multiple channels. Channels can be shared between networks but only one transmitter can locally transmit on a channel at any moment in time.
MOSFET	The metal–oxide–semiconductor field-effect transistor (MOSFET, MOS-FET, or MOS FET), also known as the metal–oxide–silicon transistor (MOS transistor, or MOS), is a type of insulated-gate field-effect transistor that is fabricated by the controlled oxidation of a semiconductor, typically silicon. The voltage of the covered gate determines the electrical conductivity of the device; this ability to change conductivity with the amount of applied voltage can be used for amplifying or switching electronic signals. It is the basic building block of modern electronics, and the most frequently manufactured device in history, with an estimated total of 13 sextillion (1.3×10^{22}) MOSFETs manufactured between 1960 and 2018. It is the dominant semiconductor device in digital and analog integrated circuits (ICs), and the most common power device. It is a compact transistor that has been miniaturized and mass-produced for a wide range of applications, revolutionizing the electronics

	industry and the world economy, and being central to the digital revolution, silicon age and information age. MOSFET scaling and miniaturization has been driving the rapid exponential growth of electronic semiconductor technology since the 1960s, and enables high-density ICs such as memory chips and microprocessors.
OAuth	OAuth is an open standard for access delegation, commonly used as a way for Internet users to grant websites or applications access to their information on other websites but without giving them the passwords. This mechanism is used by companies such as Amazon, Google, Facebook, Microsoft and Twitter to permit the users to share information about their accounts with third party applications or websites. Generally, OAuth provides clients a "secure delegated access" to server resources on behalf of a resource owner. It specifies a process for resource owners to authorize third-party access to their server resources without providing credentials. Designed specifically to work with Hypertext Transfer Protocol (HTTP), OAuth essentially allows access tokens to be issued to third-party clients by an authorization server, with the approval of the resource owner. The third party then uses the access token to access the protected resources hosted by the resource server.
Hypertext Transfer Protocol	The Hypertext Transfer Protocol (HTTP) is an application layer protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access, for example by a mouse click or by tapping the screen in a web browser. HTTP functions as a request–response protocol in the client–server computing model. A web browser, for example, may be the client and an application running on a computer hosting a website may be the server. The client submits an HTTP request message to the server. The server, which provides resources such as HTML files and other content, or performs other functions on behalf of the client, returns a response message to the client. The response contains completion status information about the request and may also contain requested content in its message body.

User Story

(1)	A registered student can submit his attendance to lecture
(2)	A registered student can request a permission
(3)	An instructor can accept the student request
(4)	An instructor can add university
(5)	An instructor can add faculty
(6)	An instructor can add course
(7)	An instructor can request attendance report for the whole students
(8)	A registered student can submit his attendance to laboratory session
(9)	An instructor is able to log in the system using his name and a unique instructor ID in order to retrieve the request and required data specifically
(10)	An instructor can request an attendance report for each student separately
(11)	A registered student can submit his attendance to midterm exam
(12)	A registered student can submit his attendance to recitation
(13)	An instructor can request an attendance report for each student separately in a specified time interval
(14)	A registered student can submit his attendance to laboratory exam
(15)	A registered student can submit his attendance to final exam
(16)	A registered student can submit his attendance to specific quiz
(17)	A registered student can request a report with all his grades till the moment of the request
(18)	A registered student can differentiate (Lecture – recitation – laboratory session – laboratory exam – midterm exam – final exam - specific quiz) the attendance event using common shared code among the students and instructor, feeding this integer code to the system

	using a keypad which acts as an interface between the system between the system and the student.
(19)	A registered student can submit his attendance to event through different method from the face recognition such as finger print sensor or even a unique ID given to each student, feeding this ID to the system using a keypad which acts as an interface between the system between the system and the student. This method will be provided if the camera sensor can not properly due to illuminations and reflections of light rays as well as poor light source in the surrounding environment of the system.
(20)	An instructor can to retrieve and track the academic history of a specified student in order to giving him the whole picture from high view to judge or evaluate each student with clear insight, also it may be used with other purposes such as giving them an appropriate course project due to the skills which are gained by practicing in the work of previous courses in previous years
(21)	Any student is able to stream a live video to the dedicated window on the administration website in order to submit the attendance in order that the uploaded image is not sufficient to recognize the standing person
(22)	Any student is able to capture a live personal photo to the dedicated window on the administration website in order to submit the attendance to validate the student attendance mor efficiently
(24)	Any student is able to register to our system through a dedicated tab in our website to achieve data and results monitoring and analysis by the student to priories the student tasks to achieve the optimal efforts to provide optimal results
(25)	Any student is able to upload a photo to the dedicated window on the administration website in order to submit the attendance

User Requirements Definition

ID	Requirement	Category	Requirement's Basic Attributes			Requirement's Assessment Attributes
			Type	Classification	Source	Priority
(1)	Any registered student can take attendance	Student	Functional	Usability	External	High
(2)	Each student will have a unique ID to help in organization and management	Student	Functional	Usability	External	High
(3)	Doctor Login Requiring doctor full name as well as doctor ID	Doctor	Functional	Usability	External	High
(4)	Add course is a feature available to the doctor to add courses in order to manage and organize the system more efficiently	Doctor	Functional	Usability	External	High

(5)	Add Faculty is a feature available to the doctor to add courses in order to manage and organize the system more efficiently by wrapping up a group of related courses	Doctor	Functional	Usability	External	High
(6)	Add university is a feature available to the doctor to add courses in order to manage and organize the system more efficiently if the doctor instruct courses in different universities or different branches	Doctor	Functional	Usability	External	High
(7)	Print Attendance report due to specified time interval as well as the availability to show each student record separately	Doctor	Functional	Usability	External	High
(8)	Data recodes update in real-time manner to enable the doctor to check	System	Functional	CMS	External	High

	the attendees in lecture time at any moment					
(9)	Give green flag to indicate that the system has been recognized the student successfully	System	Functional	CMS	External	High
(10)	Give red flag to indicate the system is unable to recognize a student	System	Functional	CMS	External	High
(11)	Send remainder to remember doctor to lecture time and deadlines	System	Functional	Usability	External	High
(12)	Send warning Is a mechanism of notifying students when they are near to the limit of allowed number of absences	Doctor	Functional	Usability	External	High
(13)	Give permissions is a feature which is available to the doctor in order to	Doctor	Functional	Usability	External	High

	know the reasons of student absence					
(14)	Request permission is a feature which is available to the doctor in order to notify the doctor why he cannot attend a specific lecture	Student	Functional	Usability	External	High
(15)	The database may get crashed at any certain time due to virus or operating system failure. Therefore, it is required to take the database backup	System	Non-Functional	CMS	External	High
(16)	The system is available 100% for the user and is used 24 hrs a day and 365 days a year. The system shall be operational 24 hours a day and 7 days a week.	System	Non-Functional	CMS	External	High
(17)	The system requires a database in order to store persistent	System	Non-Functional	CMS	External	High

	data. The database should have backup capabilities.					
(18)	Mean Time to Repair (MTTR) - Even if the system fails, the system will be recovered back up within an hour or less.	System	Non-Functional	CMS	External	High
(19)	The system should accurately provide real time information taking into consideration various concurrency issues. The system shall provide 100% access reliability.	System	Non-Functional	CMS	External	High
(20)	The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database. The	System	Non-Functional	CMS	External	High

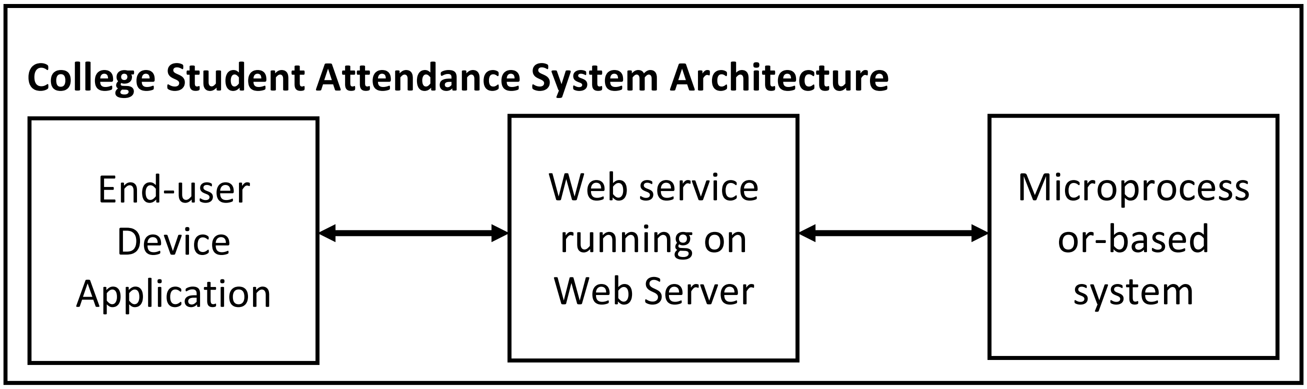
	database will be MySQL					
(21)	Power availability as there is an embedded unit in our system which acts as an interface between the system and student, we must provide stable power source in order to keep it operating well	System	Non-Functional	Usability	External	High
(22)	Construct a system with enough computing capabilities in order to do image enhancement locally on the microprocessor chip to enhance performance and overall system efficiency	System	Non-Functional	Usability	External	High
(23)	Any student is able to upload a photo to the dedicated window on the administration website in order to	Student	Functional	Usability	External	High

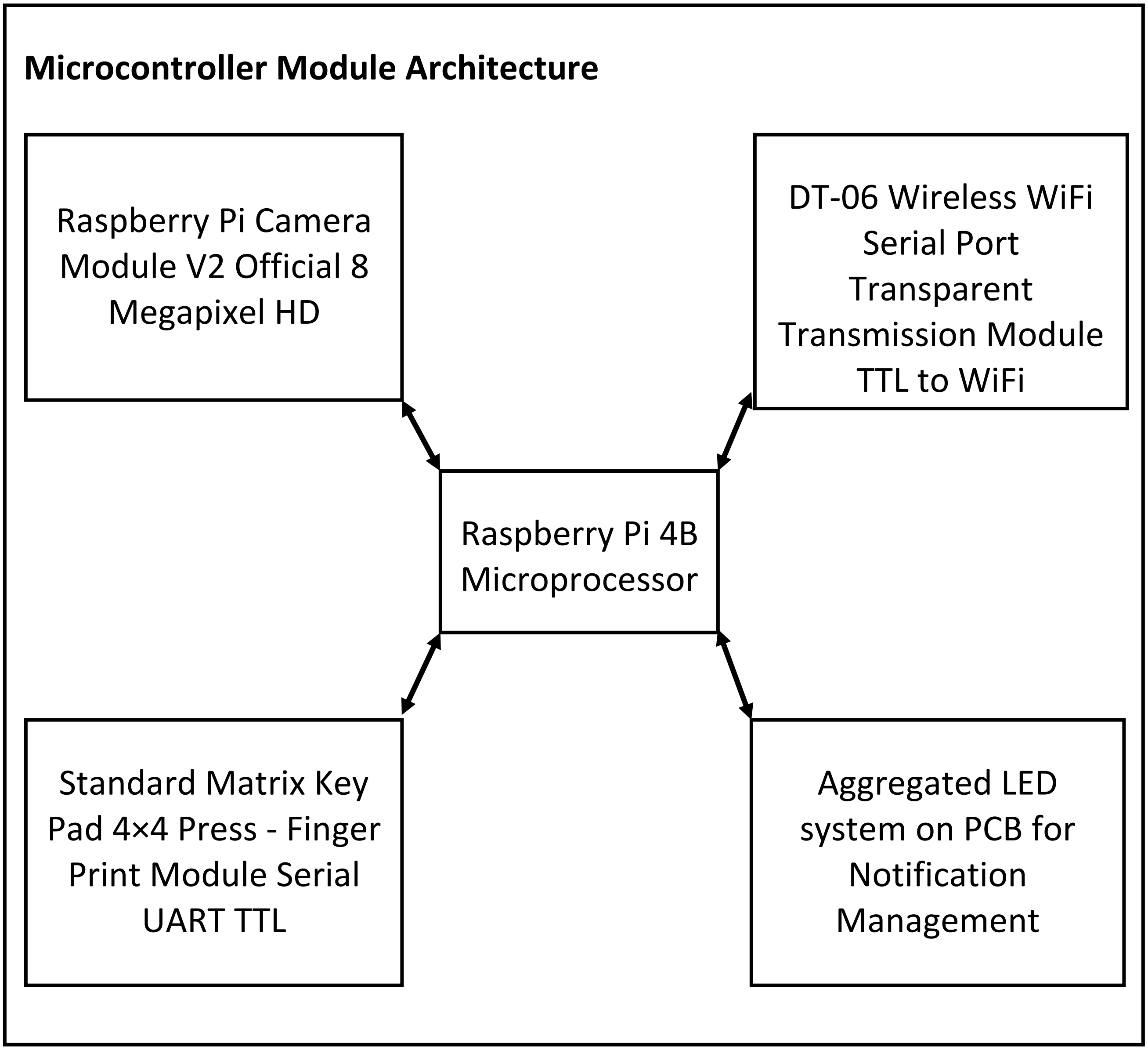
	submit the attendance					
(24)	Any student is able to register to our system through a dedicated tab in our website to achieve data and results monitoring and analysis by the student to priories the student tasks to achieve the optimal efforts to provide optimal results	Student	Functional	Usability	External	High
(25)	Any student can label a show password box to view the password as plain text instead of encrypted style to ensure that the password is correct in its syntax	Student	Functional	Usability	External	High
(26)	An error message must be shown in the registration form to accomplish data integrity to maintain our database robust if a required field is left empty and a user	System	Functional	CMS	Internal	High

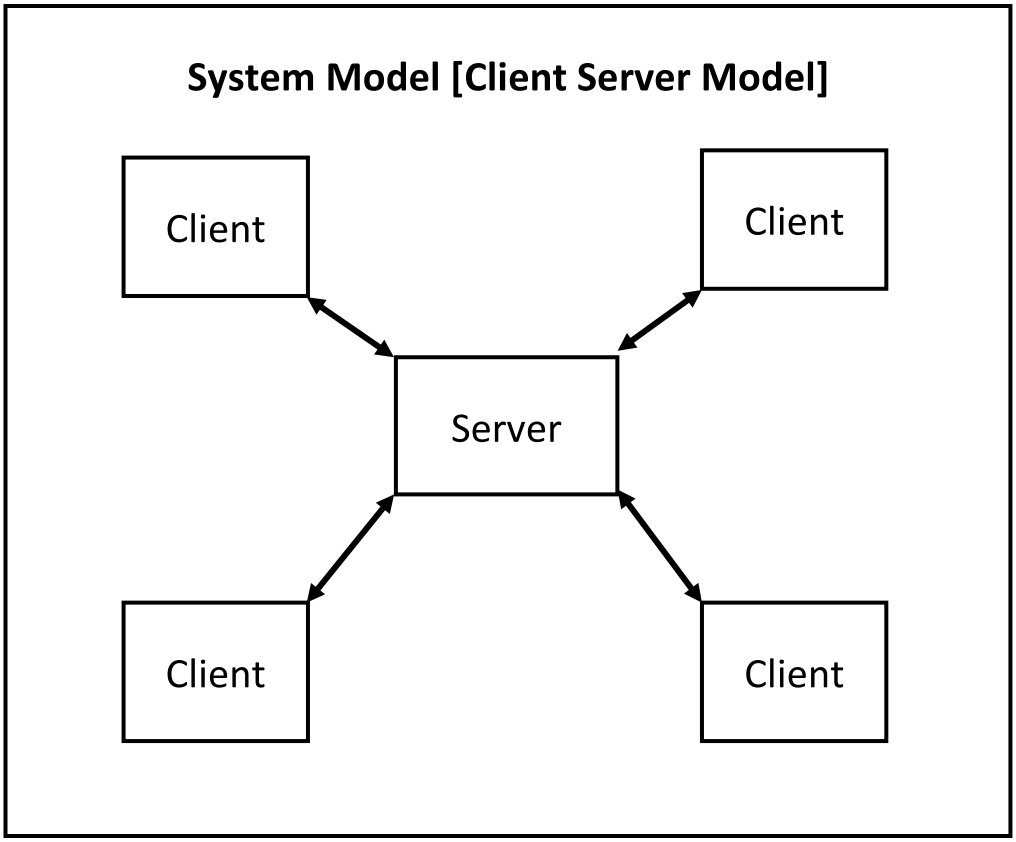
	must fill it to keep our database design or business rules					
(27)	The data resources cannot be accessed through URL injection the user must be authorized in one of two ways by storing his information by website cookies or by registration	System	Functional	CMS	External	High
(28)	A cookie software must be provided to ensures and provide data integrity	System	Functional	CMS	Internal	High
(29)	An error message may be required to be shown if the image clear enough to recognize the person who in the uploaded image	System	Functional	CMS	External	High
(30)	Any student is able to capture a live personal photo to the dedicated window on the administration website in order to	Student	Functional	Usability	External	High

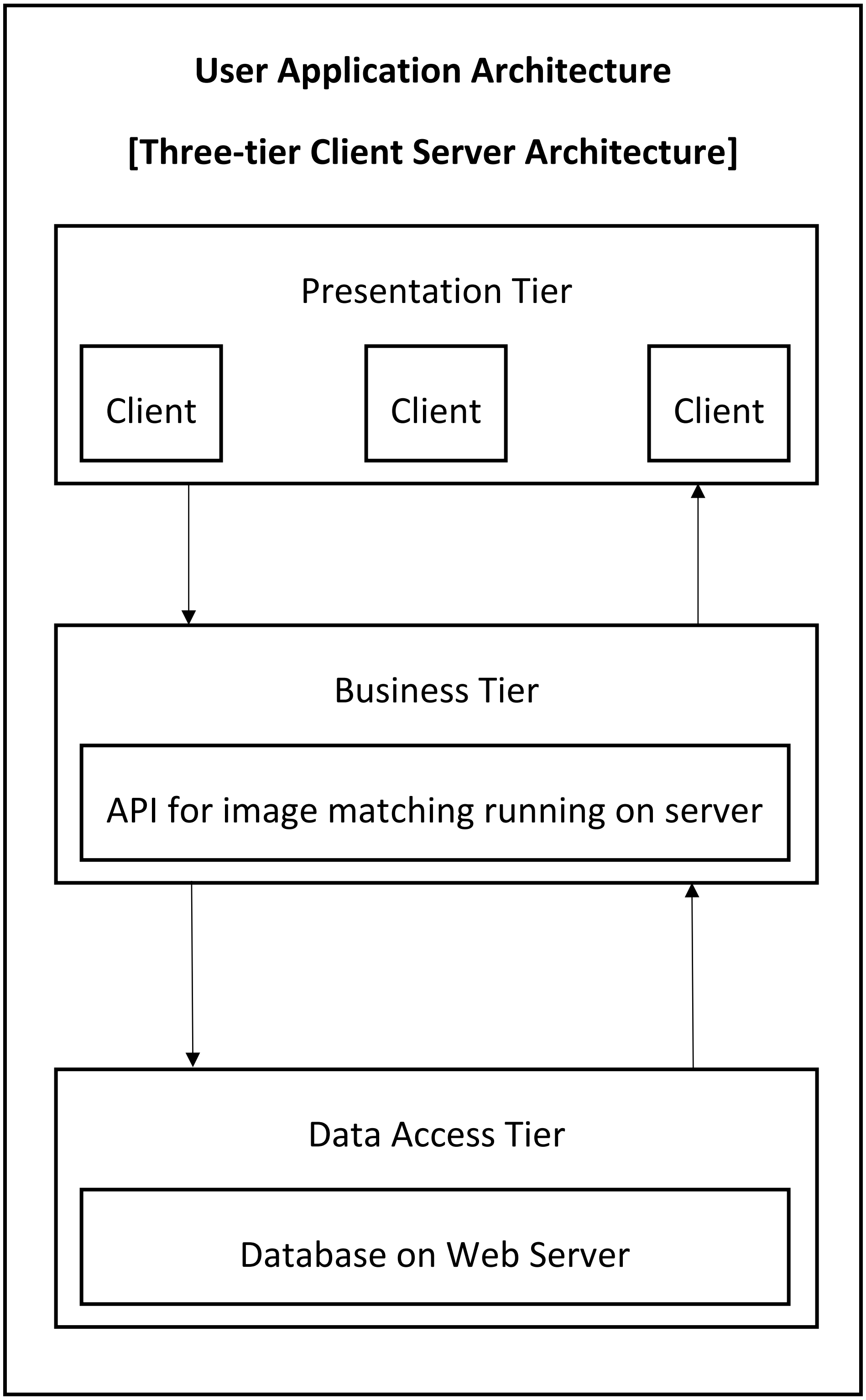
	submit the attendance to validate the student attendance mor efficiently					
(31)	Any student is able to stream a live video to the dedicated window on the administration website in order to submit the attendance in order that the uploaded image is not sufficient to recognize the standing person	Student	Functional	Usability	External	High
(32)	A reset password option must be provided to both student and instructor to provide resources accessibility	Student Doctor	Functional	Usability	External	High

System Architecture


College Student Attendance System Architecture
End-user Device Application
Web service running on Web Server
Microprocessor-based system

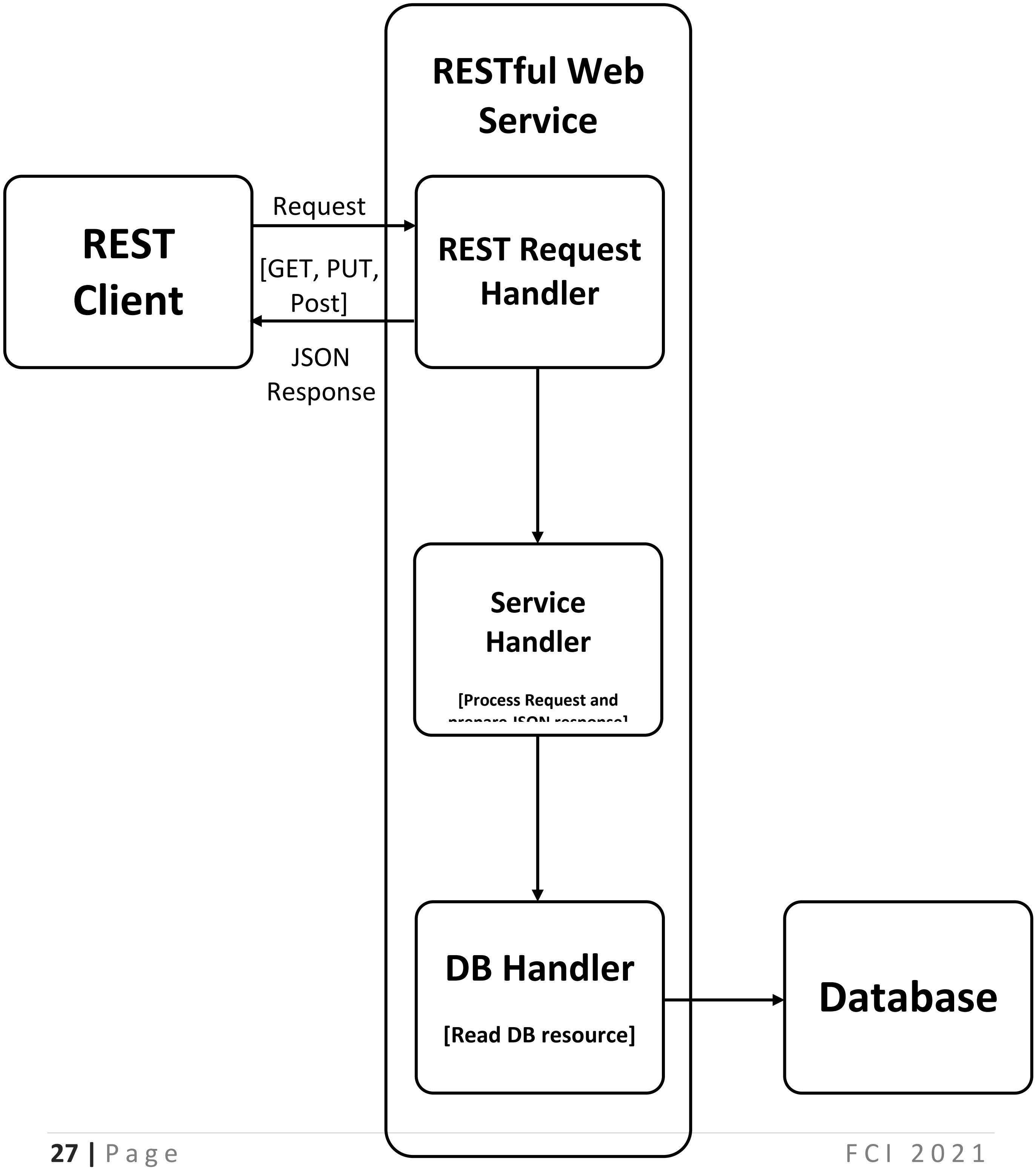

Microcontroller Module Architecture
Raspberry Pi Camera Module V2 Official 8 Megapixel HD
DT-06 Wireless WiFi Serial Port Transparent Transmission Module TTL to WiFi
Raspberry Pi 4B Microprocessor
Standard Matrix Key Pad 4×4 Press - Finger Print Module Serial UART TTL
Aggregated LED system on PCB for Notification Management


System Model [Client Server Model]
Client
Client
Server
Client
Client



User Application Architecture
[Three-tier Client Server Architecture]
Presentation Tier
Client
Client
Client
Business Tier
API for image matching running on server
Data Access Tier
Database on Web Server

Web Service API Architecture [running on Web Server]



System Requirements Specification (SRS)

1. The system should be running and continuously taking pictures via a qualified camera during the lecture time and keep adding the attendance of the people it identifies until the end of the lecture. This gathered information is then stored and structured in a database. Student's data will be added to the attendance table with the time of attending and the lecture number.

2. A database must be made to hold all the important information about and required by the system. This database should be a relational database that holds tables about different entities of the system with all the attributes or fields necessary about that specific entity. One of these entities is the student; the system should have the full name, student id, email, university, faculty, and the different courses that the student is registered in. Also, the face encodings of that specific student so that the system knows his face. Another entity is the Course and each of these must have an id and the course code and name. Also, the Faculty is an important entity and it has a many-to-many relationship with the University entity.

3. The database should follow the ACID (atomicity, consistency, isolation, durability) properties which is a set of properties of database transactions intended to guarantee data validity despite errors, power failures, and other mishaps. In the context of databases, a sequence of database operations that satisfies the ACID properties (which can be perceived as a single logical operation on the data) is called a transaction. For example, a transfer of funds from one bank account to another, even involving multiple changes such as debiting one account and crediting another, is a single transaction. Therefore, the database must have a backup and replications. The Quality of the database is maintained in such a way so that it can be very user friendly to all the users of the database. The database will be SQLite.

4. Different universities can be added to the system and work as a distributed system with maintaining the integrity and security of each one of them. The admins or university staff at each university can only work with the data for this specific university and not be able to comprise or manipulate any other data. Also, each doctor or faculty staff is authorized to access the least amount of data possible. A report is generated with the requested data for the admin so that they can seek information anytime they need. Also, the attendance for each course must be restricted to only the doctors responsible for this course and can have live updates in real-time of their attendance. Also, the system should be running and available all the time of the day and every day.

5. The university staff must provide the necessary information about the different faculties, doctors, and courses for the university before the system production. This must be achieved through adding the required data directly to the database on the server through some python functions. These data must be accurate and complete so that when going to production, the system is ready to handle user requests. These data should include all faculties in the university, all doctors applied in each of the faculties, and the courses required in each of the faculties with which doctors are working on them.

6. The student must register on the system to be able to identify him before he can take attendance and that happens via a web form the student must fill and provide his picture to upload it on the server. Each student will be able to choose his faculty while registering. This web form contains all columns of the student table besides the courses he/she applied to.

7. After student registration, the system should analyze the picture he provided and detect his face and reject the picture if the picture turns out to have multiple faces or if doesn’t contain a face at all. And if the picture passes these conditions, the system should get the encodings of that face and store them in the database. Another important edge case is if this specific encoding is already in the database, in this case the picture is also rejected.

8. For testing and usability measurements, the student can verify attendance by either uploading a picture or live video feeding to the system throughout his\her webcam.
9. To log in as a doctor or admin the user must provide his full name or username as well as his password. Then the system checks if the user data is registered in the database. If the data is valid then redirect the user to the admin panel otherwise show an error message, and if the user tries to access the admin panel without login, he will be redirected to the login page.

10. Any passwords or sensitive information cannot be stored in plain text but rather must be hashed to secure it. Using argon2 hash generator algorithm to generate a hash of each password in the database to protect the user's password in case of a data leak. the hash value will be unique for each password and argon2 is not fast to crack while md5 or sh1 are easy to brute-force thousands of keys in on each second.

11. When a user submits the registration form all the user data must be carefully checked and sanitized to prevent SQL injection, stored cross-site scripting, and object deserialization. which can escalate further to remote code execution or database leak.

12. Doctors can change their passwords while keeping security constraints like password length and including lower and upper letters and special characters. the new password is sent to the server and on the server-side constraints are checked again to make sure the new pass is valid then it generates the argon2 hash for the pass and update the database. all this is done only if is not log in this he has no access to this page.

13. Each Doctor can add his courses to the database. Each course has a unique code and table of students. Students will be able to register for these courses and then attend it. If the student provided a course that does not exist in the database, that course should be simply ignored.

14. Doctors will send warnings to students if they are near to the allowed limit of absence in any course. It will make it easier for doctors to communicate with students and easier for students to keep track of their absence. The allowed limit of absence will be set by the faculty staff. Also, the doctor will receive remainders or warnings if there is a deadline or some of the students are about to violate one of the university attendances rules (e.g., absence more than three lectures in a row). Therefore, the students should be informed and send the reasons for their absence.

15. The system should which course to take attendance into and which lecture of that course via knowing the faculty schedule before hand or the doctor will configure these data before the lecture starts. But for testing sake, we will let the student provide the name of the course and the lecture number before submitting his\her attendance.

16. Allow the user to view their input of the password field in plain text to be able to notice any misspelled word in their input. And that can be achieved by a “show password” checkbox that the user can enable or disable.

17. Prevent unauthorized URL traversal like access admin page without login this done by check for user credential in user session on every page if the user was not unauthorized to view that page, he will be redirected to the login page. The session id is stored in a cookie on the user browser this allows the server to distinguish each user.

18. In uploading pictures, either in registration or attendance, the system should check the image type and the extension in the filename. And if the type is in the allowed types (e.g., jpg, png, jpeg) then try to resize the picture and if successful then proceed to the next steps. This must be done in both the client side and the server side to avoid malicious files to be uploaded.

19. The system will send an error message to the student if the uploaded image wasn't clear enough to identify him/her. This will increase the system's accuracy. Each student will know the problem with his/her registration and

this will make a better user experience.

20. Any student is able to stream a live video to the dedicated window on the administration website in order to submit the attendance in order that the uploaded image is not sufficient to recognize the standing person.

21. There is a green led attached to $10\ Kilo\ ohm$ to protect the led from over-current operation. This governed by ohm law which states that the current through a conductor between two points is directly proportional to the voltage across the two points. Introducing the constant of proportionality, the resistance, one arrives at the usual mathematical equation that describes this relationship: $I = \frac{V}{R}$ where I is the current through the conductor in units of amperes, V is the voltage measured across the conductor in units of volts, and R is the resistance of the conductor in units of ohms. More specifically, Ohm's law states that the R in this relation is constant, independent of the current. There are two led in our microprocessor-based module depending on the validation of the image matching operation which indicates whether the student is recognized or not which led to update the database field or not as well.

22. The power supply requirements differ by Raspberry Pi model. All models require a 5.1V supply, but the current supplied generally increases according to model. All models up to the Raspberry Pi 3 require a *microUSB* power connector, whilst the Raspberry Pi 4 uses a USB-C connector. Exactly how much current (mA) the Raspberry Pi requires is dependent on what you connect to it. The current requirements for Raspberry Pi 4 Model B are specified as the following:
    - Recommended PSU current capacity: *3.0 A*
    - Maximum total USB peripheral current draw: *1.2 A*
    - Typical bare-board active current consumption: *600 mA*

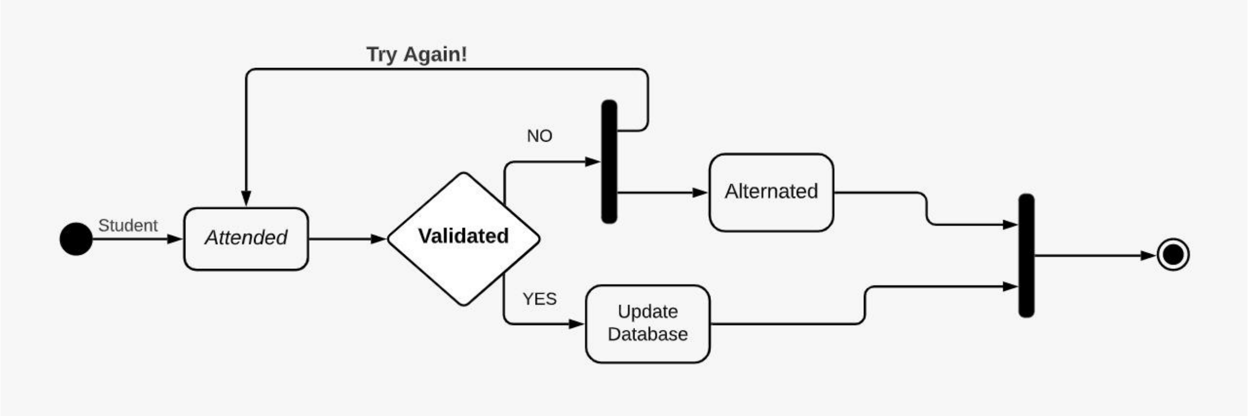
    The power requirements of the Raspberry Pi increase as you make use of the various interfaces on the Raspberry Pi. The GPIO pins can draw 50mA safely, distributed across all the pins; an individual GPIO pin can only safely draw 16mA. The HDMI port uses 50mA, the camera module requires 250mA, and keyboards and mice can take as little as 100mA or over

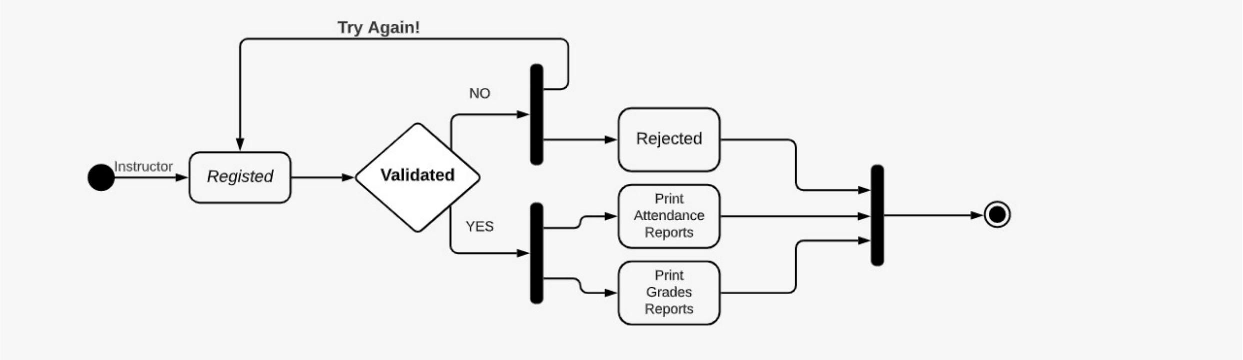
1000mA! We have to provide stable electrical power source with the previous requirements and specification from two separated devices to ensure power availability and system efficient performance.

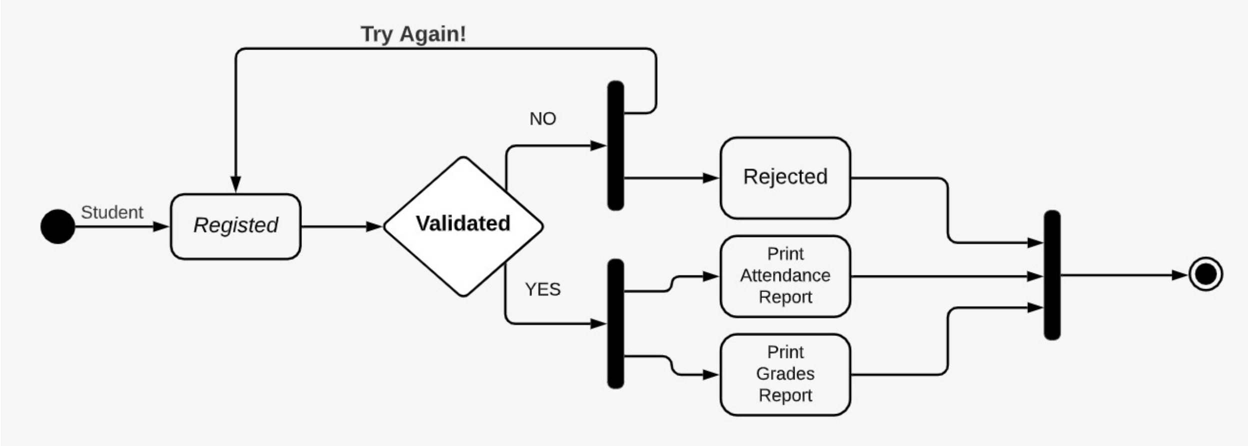
23. Construct a system with enough computing capabilities in order to do image enhancement locally on the microprocessor chip so we must provide a microprocessor with enough processing capabilities such as sufficient RAM size to store the image in an appropriate manner as well as the raspberry pi 3 board come with ARM processor to provide processing at an efficient time and many other resources with appropriate specifications as following:
    - Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
    - 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
    - 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
    - Gigabit Ethernet
    - 2 USB 3.0 ports; 2 USB 2.0 ports.
    - Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
    - 2 × micro-HDMI ports (up to 4kp60 supported)
    - 2-lane MIPI DSI display port
    - 2-lane MIPI CSI camera port
    - 4-pole stereo audio and composite video port
    - H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
    - OpenGL ES 3.0 graphics
    - Micro-SD card slot for loading operating system and data storage
    - 5V DC via USB-C connector (minimum 3A*)
    - 5V DC via GPIO header (minimum 3A*)
    - Power over Ethernet (PoE) enabled (requires separate PoE HAT)
    - Operating temperature: 0 – 50 degrees C ambient

System Models

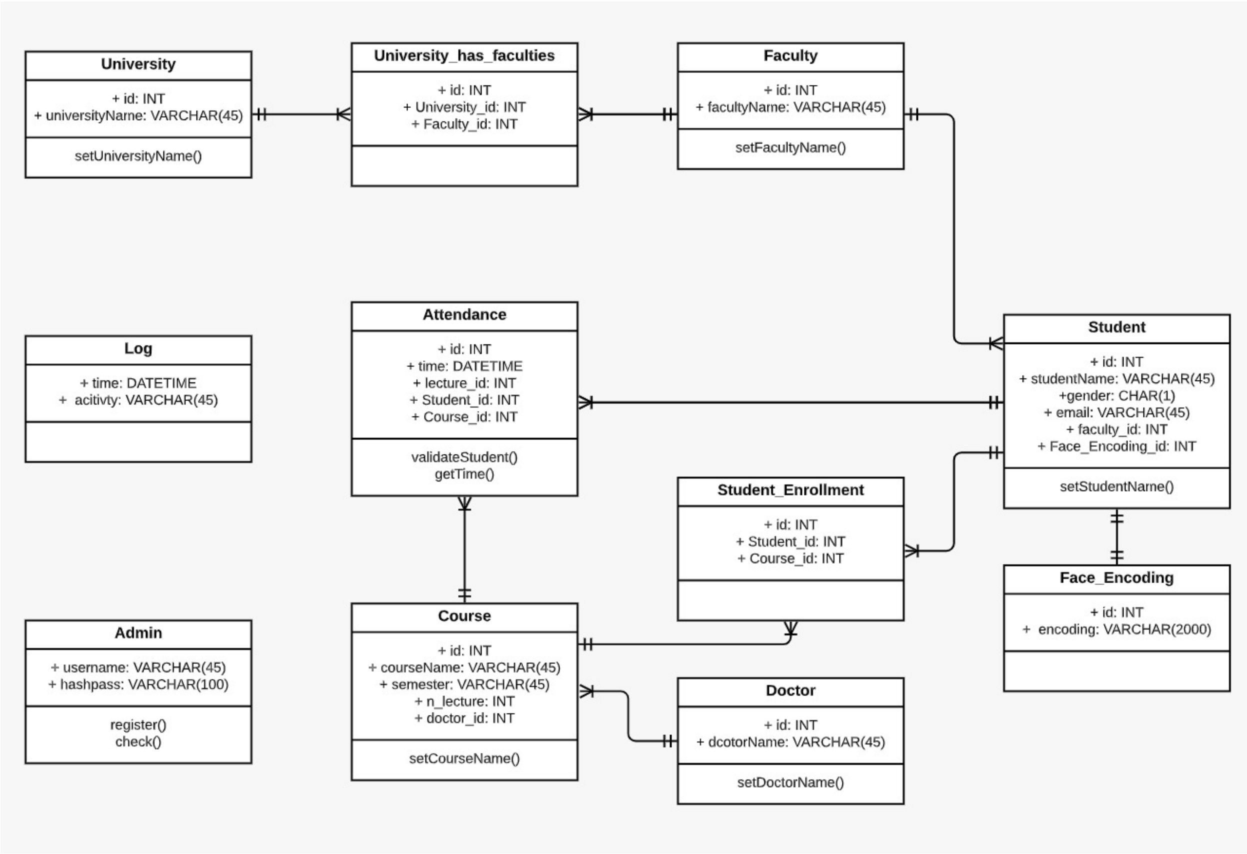
- Activity Diagrams



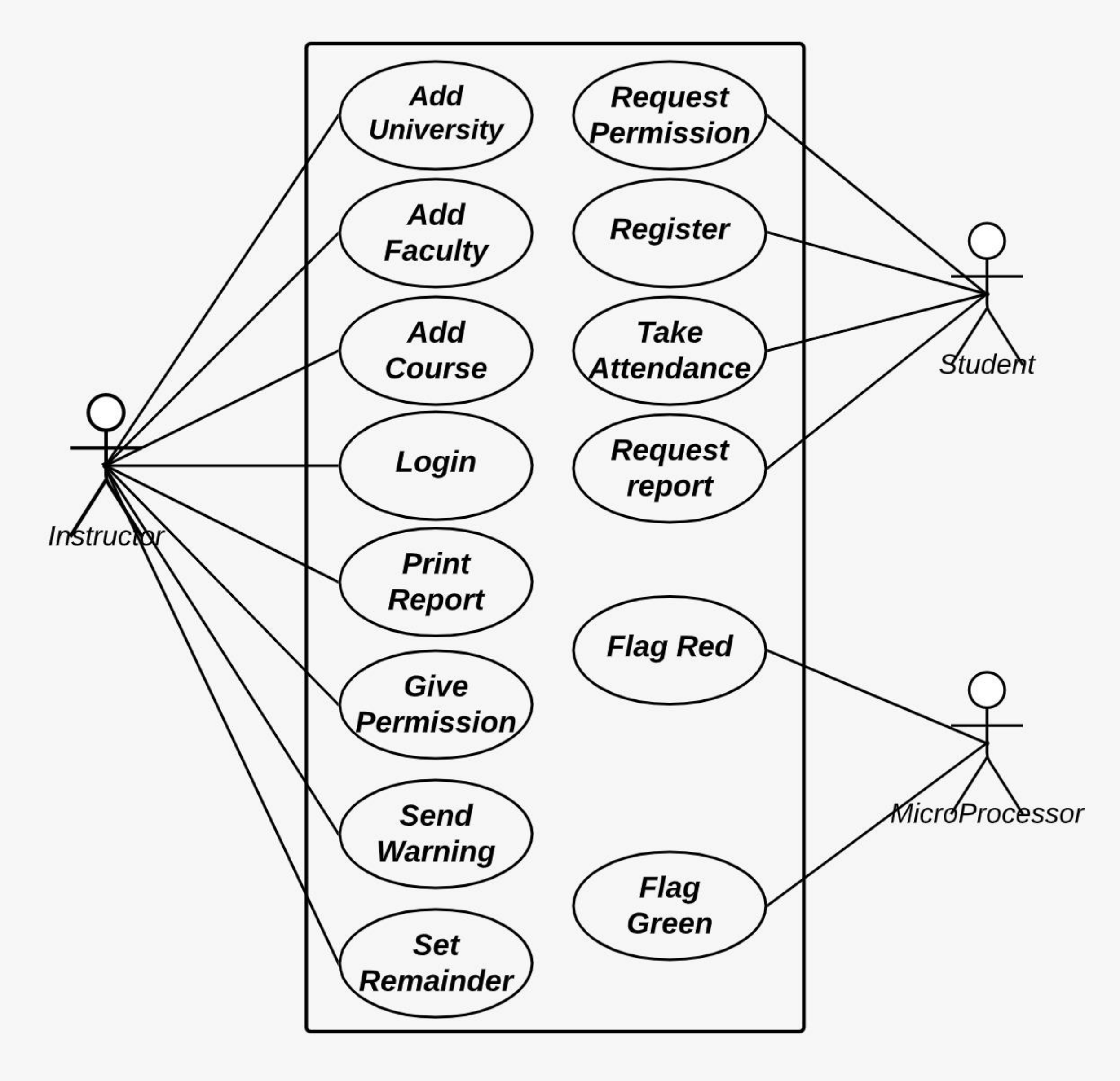




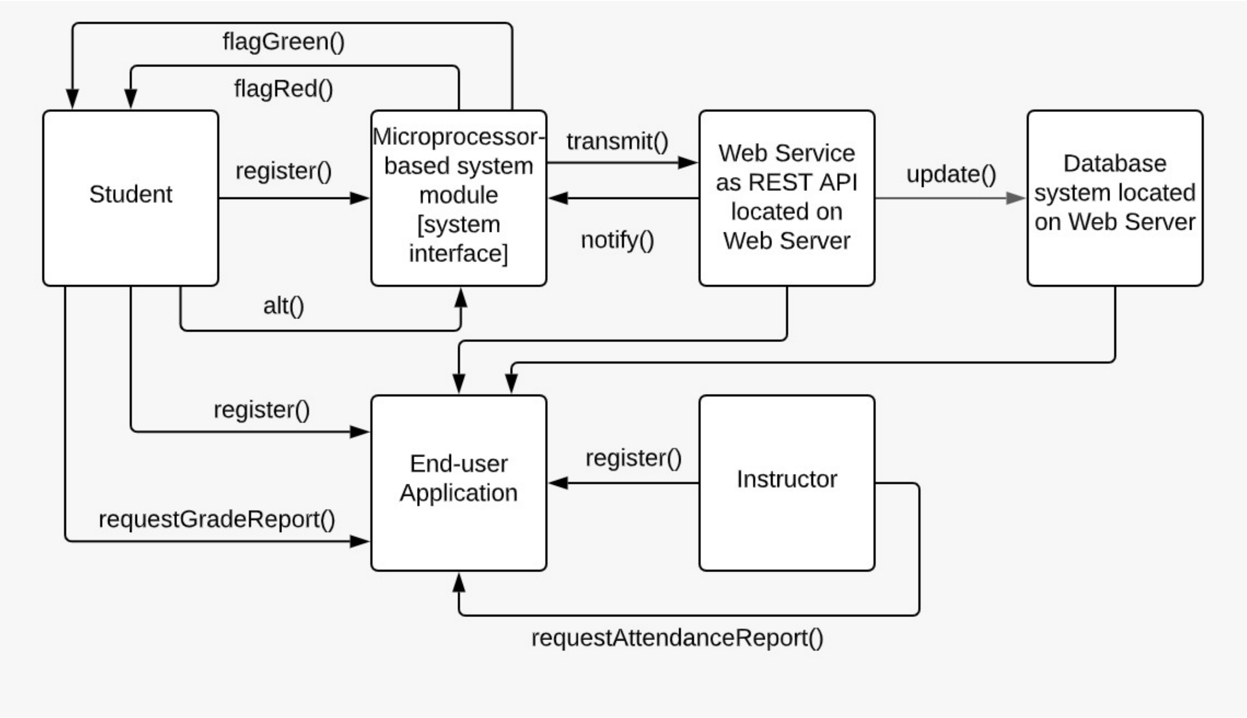
- Class Diagram



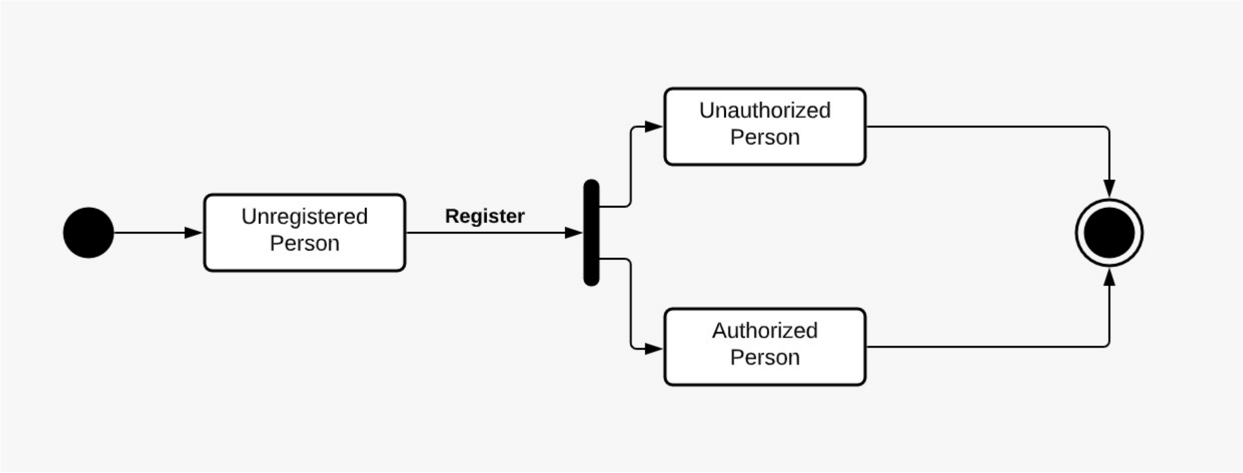
- USE CASE Diagram



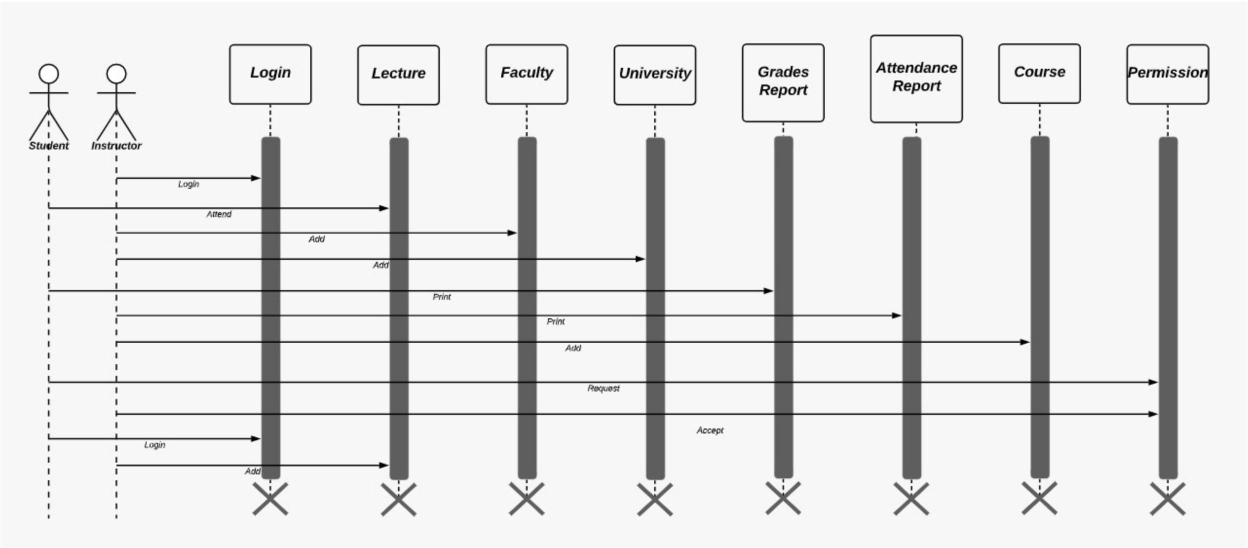
- Communication Diagram


flagGreen()
flagRed()
Student
register()
Microprocessor-based system module [system interface]
transmit()
notify()
Web Service as REST API located on Web Server
update()
Database system located on Web Server
alt()
register()
End-user Application
register()
Instructor
requestGradeReport()
requestAttendanceReport()

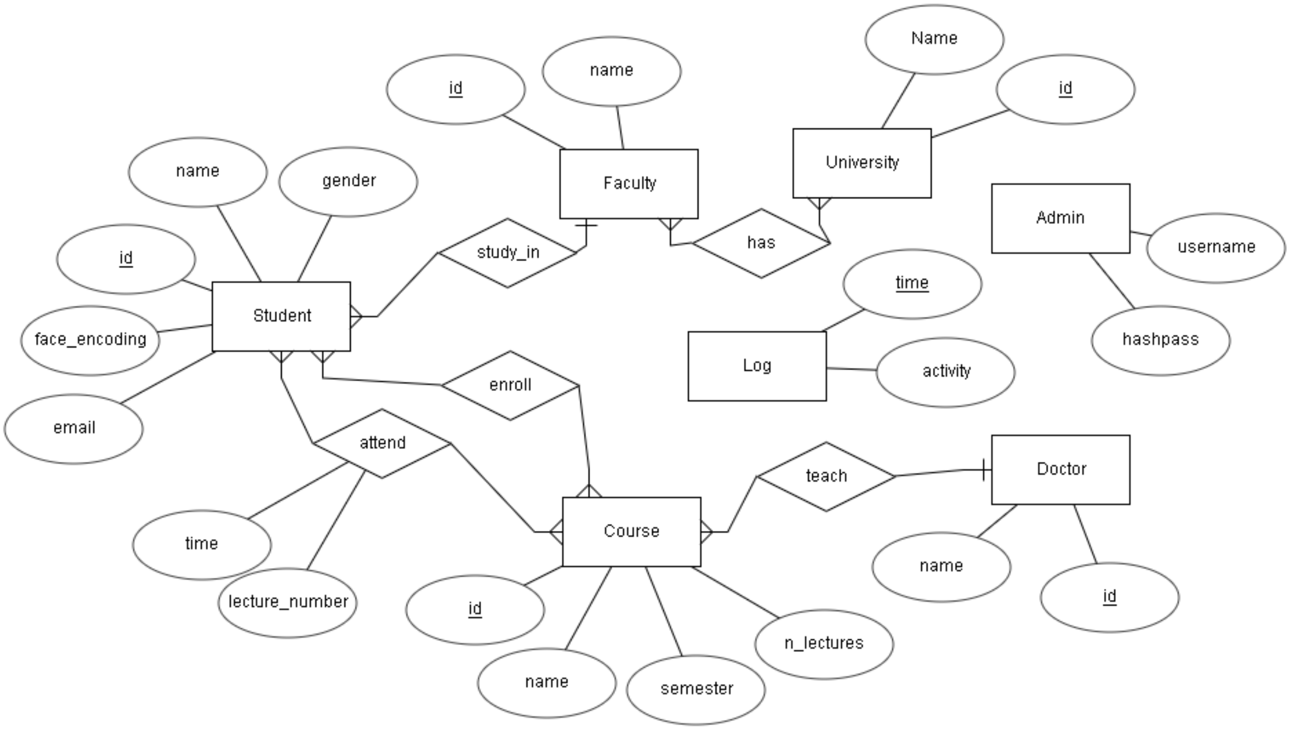
- State Diagram



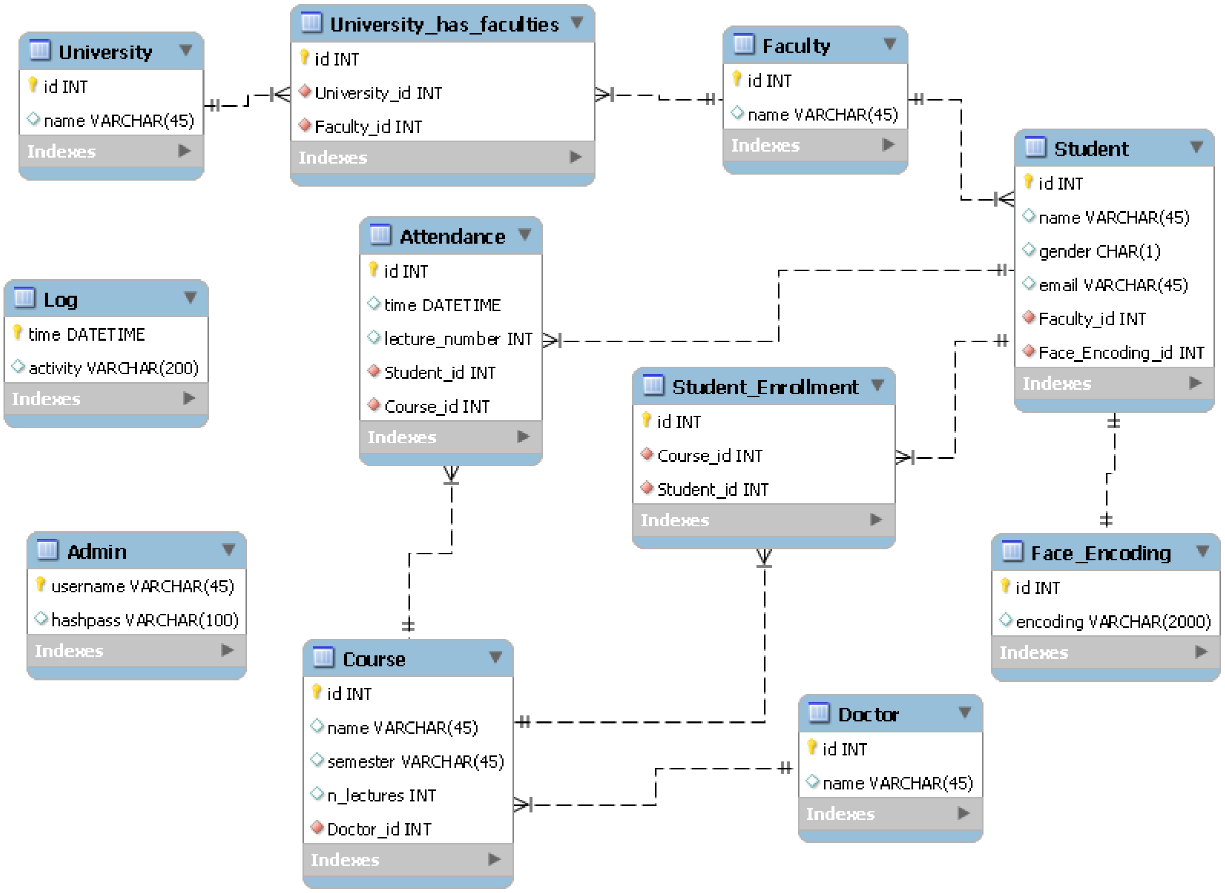
- Sequence Diagram



- ERD (Entity Relational Diagram)



- Relational Diagram



System Evolution

Face recognition is an important application of Image processing owing to its use in many fields. Identification of individuals in an organization for the purpose of attendance is one such application of face recognition. Maintenance and monitoring of attendance records plays a vital role in the analysis of performance of any organization. The purpose of developing attendance management system is to computerize the traditional way of taking attendance. Automated Attendance Management System performs the daily activities of attendance marking and analysis with reduced human intervention. The prevalent techniques and methodologies for detecting and recognizing face fail to overcome issues such as scaling, pose, illumination, variations, rotation, and occlusions. The proposed system aims to overcome the pitfalls of the existing systems and provides features such as detection of faces, extraction of the features, detection of extracted features, and analysis of students' attendance. The system integrates techniques such as image contrasts, integral images, color features and cascading classifier for feature detection. The system provides an increased accuracy due to use of a large number of features (Shape, Color, LBP, wavelet, Auto-Correlation) of the face. Faces are recognized using Euclidean distance and k-nearest neighbor algorithms. Better accuracy is attained in results as the system takes into account the changes that occur in the face over the period of time and employs suitable learning algorithms.

The system is tested for various use cases. We consider a specific area such as classroom attendance for the purpose of testing the accuracy of the system. The metric considered is the percentage of the recognized faces per total number of tested faces of the same person. The system is tested under varying lighting conditions, various facial expressions, presence of partial faces (in densely populated classrooms) and presence or absence of beard and spectacles. An increased accuracy (nearly 100%) is obtained in most of the cases considered.
In this proposed system, the system is instantiated by the mobile. After it triggers then the system starts processing the image for which we want to mark the attendance. Image Capturing phase is one in which we capture the image. This is basic phase from which we start initializing our system. We capture an image from a camera which is predominantly checked for certain constraints like lightning, spacing, density, facial expressions. The captured image is resolute for

our requirements. Once it is resolute, we make sure it is either in png or jpeg format else it is converted. We take individuals different frontal postures so that the accuracy can be attained to the maximum extent. This is the training database in which every individual has been classified based on labels. For the captured image, from every object we detect only frontal faces from viola-jones algorithm which detects only the frontal face posture of every individual from the captured image. This detects only faces and removes every other part since we are exploring the features of only faces. These detected faces are stored in the test database for further enquiry. Features are extracted in this extraction phase. The detected bounding boxes are further queried to look for features extraction and the extracted features are stored in matrix. For every detected phase this feature extraction is done. Features we look here are Shape, Edge, Color, Wavelet, Auto-Correlation and LBP. Face is recognized once we completed extracting features. The feature which is already trained with every individual is compared with the detected faces feature and if both features match then it is recognised. Once, it recognizes it is going to update in the student attendance database. Once, the process is completed the testing images gets deleted since, we are trying to design it for both the accuracy as well as efficiency co-efficient.

There may be various types of lighting conditions, seating arrangements and environments in various classrooms. Most of these conditions have been tested on the system and system has shown 100% accuracy for most of the cases. There may also exist students portraying various facial expressions, varying hair styles, beard, spectacles etc. All of these cases are considered and tested to obtain a high level of accuracy and efficiency. Thus, it can be concluded from the above discussion that a reliable, secure, fast and an efficient system has been developed replacing a manual and unreliable system. This system can be implemented for better results regarding the management of attendance and leaves. The system will save time, reduce the amount of work the administration has to do and will replace the stationery material with electronic apparatus and reduces the amount of human resource required for the purpose. Hence a system with expected results has been developed but there is still some room for improvement. The system should handle up to all students in Egypt and run 24/7 and give accurate and quick results in any case.

Appendices

What is Python?

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. Python is an interpreted, high-level, general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python programming has various frameworks and features.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Good to know

The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.

In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

What is Web Framework?

Web Application Framework or simply Web Framework represents a collection of libraries and modules that enables a web application developer to write applications without having to bother about low-level details such as protocols, thread management etc.

What is Flask?

Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects. Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have form a validation support. Instead, Flask supports the extensions to add such functionality to the application.

WSGI

Web Server Gateway Interface (WSGI) has been adopted as a standard for Python web application development. WSGI is a specification for a universal interface between the web server and the web applications.

Werkzeug

It is a WSGI toolkit, which implements requests, response objects, and other utility functions. This enables building a web framework on top of it. The Flask framework uses Werkzeug as one of its bases.

Jinja2

Jinja2 is a popular templating engine for Python. A web templating system combines a template with a certain data source to render dynamic web pages.

Flask Prerequisite.

Python 2.6 or higher is usually required for installation of Flask. Although Flask and its dependencies work well with Python 3 (Python 3.3 onwards), many Flask extensions do not support it properly. Hence, it is recommended that Flask should be installed on Python 2.7.

Install virtualenv for development environment

virtualenv is a virtual Python environment builder. It helps a user to create multiple Python environments side-by-side. Thereby, it can avoid compatibility issues between the different versions of the libraries.

The following command installs virtualenv:

- pip install virtualenv
- Once installed, new virtual environment is created in a folder.
- virtualenv venv
- venv\scripts\activate

We are now ready to install Flask in this environment.

- pip install Flask

The above command can be run directly, without virtual environment for system-wide installation.

Debug mode

A Flask application is started by calling the run() method. However, while the application is under development, it should be restarted manually for each change in the code. To avoid this inconvenience, enable debug support. The server will then reload itself if the code changes. It will also provide a useful debugger to track the errors if any, in the application.

The Debug mode is enabled by setting the debug property of the application object to True before running or passing the debug parameter to the run() method.

- app.debug = True
- app.run()
- app.run(debug = True)

Templates

The term ‘web templating system’ refers to designing an HTML script in which the variable data can be inserted dynamically. A web template system comprises of a template engine, some kind of data source and a template processor.

Flask uses jinja2 template engine. A web template contains HTML syntax interspersed placeholders for variables and expressions (in these case Python expressions) which are replaced values when the template is rendered.

Cookie:

A cookie is stored on a client's computer in the form of a text file. Its purpose is to remember and track data pertaining to a client's usage for better visitor experience and site statistics.

A Request object contains a cookie's attribute. It is a dictionary object of all the cookie variables and their corresponding values, a client has transmitted. In addition to it, a cookie also stores its expiry time, path and domain name of the site.

In Flask, cookies are set on response object. Use make_response() function to get response object from return value of a view function. After that, use the set_cookie() function of response object to store a cookie.

Reading back a cookie is easy. The get() method of request.cookies attribute is used to read a cookie.

Session

Like Cookie, Session data is stored on client. Session is the time interval when a client logs into a server and logs out of it. The data, which is needed to be held across this session, is stored in the client browser.

A session with each client is assigned a Session ID. The Session data is stored on top of cookies and the server signs them cryptographically. For this encryption, a Flask application needs a defined SECRET_KEY.

Session object is also a dictionary object containing key-value pairs of session variables and associated values.

Redirect & Errors

Flask class has a redirect() function. When called, it returns a response object and redirects the user to another target location with specified status code.

The following status codes are standardized:

- HTTP_300_MULTIPLE_CHOICES
- HTTP_301_MOVED_PERMANENTLY
- HTTP_302_FOUND
- HTTP_303_SEE_OTHER
- HTTP_304_NOT_MODIFIED
- HTTP_305_USE_PROXY
- HTTP_306_RESERVED
- HTTP_307_TEMPORARY_REDIRECT

The Code parameter takes one of following values:

- 400 – for Bad Request
- 401 – for Unauthenticated
- 403 – for Forbidden
- 404 – for Not Found
- 406 – for Not Acceptable
- 415 – for Unsupported Media Type
- 429 – Too Many Requests

Handling file upload in Flask is very easy. It needs an HTML form with its enctype attribute set to ‘multipart/form-data’, posting the file to a URL. The URL handler fetches file from request.files[] object and saves it to the desired location.

File Uploading:

Each uploaded file is first saved in a temporary location on the server, before it is actually saved to its ultimate location. Name of destination file can be hard-coded or can be obtained from filename property of request.files[file] object. However, it is recommended to obtain a secure version of it using the secure_filename() function.

It is possible to define the path of default upload folder and maximum size of uploaded file in configuration settings of Flask object.

Flask extension:

Flask is often referred to as a micro framework, because a core functionality includes WSGI and routing based on Werkzeug and template engine based on Jinja2. In addition, Flask framework has support for cookie and sessions as well as web helpers like JSON, static files etc. Obviously, this is not enough for the development of a full-fledged web application. This is where the Flask extensions come in picture. Flask extensions give extensibility to Flask framework.

There are a large number of Flask extensions available. A Flask extension is a Python module, which adds specific type of support to the Flask application. Flask Extension Registry is a directory of extensions available. The required extension can be downloaded by pip utility.

like:

- Flask Mail − provides SMTP interface to Flask application
- Flask WTF − adds rendering and validation of WTForms
- Flask SQLAlchemy − adds SQLAlchemy support to Flask application
- Flask Sijax − Interface for Sijax - Python/jQuery library that makes AJAX easy to use in web applications

Flask WTF:

One of the essential aspects of a web application is to present a user interface for the user. HTML provides a <form> tag, which is used to design an interface. A Form’s elements such as text input, radio, select etc. can be used appropriately.

Data entered by a user is submitted in the form of Http request message to the server side script by either GET or POST method.

The Server side script has to recreate the form elements from http request data. So in effect, form elements have to be defined twice – once in HTML and again in the server side script.

Another disadvantage of using HTML form is that it is difficult (if not impossible) to render the form elements dynamically. HTML itself provides no way to validate a user's input.

This is where WTForms, a flexible form, rendering and validation library comes handy. Flask-WTF extension provides a simple interface with this WTForms library.

Using Flask-WTF, we can define the form fields in our Python script and render them using an HTML template. It is also possible to apply validation to the WTF field.

Flask – SQLAlchemy:

Using raw SQL in Flask web applications to perform CRUD operations on database can be tedious. Instead, SQLAlchemy, a Python toolkit is a powerful OR Mapper that gives application developers the full power and flexibility of SQL. Flask-SQLAlchemy is the Flask extension that adds support for SQLAlchemy to your Flask application.

What is ORM (Object Relation Mapping)?

Most programming language platforms are object oriented. Data in RDBMS servers on the other hand is stored as tables. Object relation mapping is a technique of mapping object parameters to the underlying RDBMS table structure. An ORM API provides methods to perform CRUD operations without having to write raw SQL statements.

Argon2:

Argon2 is a key derivation function that was selected as the winner of the Password Hashing Competition in July 2015.[1][2] It was designed by Alex Biryukov, Daniel Dinu, and Dmitry Khovratovich from the University of Luxembourg.[3] The reference implementation of Argon2 is released under a Creative Commons CC0 license (i.e. public domain) or the Apache License 2.0.

Index