



## STUDENTS' ATTENDANCE MANAGEMENT IN HIGHER INSTITUTIONS USING AZURE COGNITIVE SERVICE AND OPENCV FACE DETECTION & RECOGNITION ATTENDANCE SYSTEM

Edison Kagona and Sani Usman

Department of Computer Science & Information Technology, Faculty of Science & Technology, International University of East Africa; P.O.Box 35502 Kampala

### Cite this article:

Edison K., Sani U. (2022), Students' Attendance Management in Higher Institutions Using Azure Cognitive Service and Opencv Face Detection & Recognition Attendance System. British Journal of Computer, Networking and Information Technology 5(1), 43-55. DOI: 10.52589/BJCNIT-ALQQMEEE

### Manuscript History

Received: 13 Nov 2021

Accepted: 1 Dec 2021

Published: 14 Feb 2022

**Copyright** © 2022 The Author(s). This is an Open Access article distributed under the terms of Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0), which permits anyone to share, use, reproduce and redistribute in any medium, provided the original author and source are credited.

**ABSTRACT:** *This research aimed at studying the current methods of attendance used at higher institutions of learning in Uganda and the feasibility of using facial biometrics as a new method of capturing attendance. Facial biometrics is distinct from other biometrics because it can be carried out without the consent of the person involved. As a result, the researcher developed a face recognition attendance system using OpenCV and Microsoft Azure CS. Questionnaires, interviews, and observations were used to capture data for the research. The data were analyzed using SPSS to get the requirements and systems functionalities. Object-Oriented Design tools were used to model the architecture of the system. Data Flow Diagram, Use-Case Diagram, Activity Diagram, and Flow Chart were used for processing whereas Entity Relation Diagram was used for data modeling. The system was designed to facilitate attendance management of a large number of attendees with ease. Efficiency and reliability were essential features of the system. Data visualization was provided to help management make informed and timely decisions on management matters that are related to attendance. The system was developed using python Tkinter, OpenCV, and Azure CS as mentioned above. The data (images) used by the system were stored in the cloud for accessibility by multiple users. The system was tested thoroughly using various testing types to uncover and fix errors and to minimize the severity of failures.*

**KEYWORDS:** Students' Attendance Management, Higher Institutions, Azure Cognitive Service, Opencv Face Detection, Recognition Attendance System



## INTRODUCTION

### Background of the Problem

Face recognition is a biometric technique that entails the identification of a given face. In recent years, face recognition has attracted much attention and its research has rapidly expanded by not only engineers but also neuroscientists and has potential applications in computer vision, communication, and automatic access control systems (R. Rajalakshmi, 2013). Face detection is an important part of recognition and its step of automatic face identification. However, face detection is not straightforward because it has lots of variations of image appearance, such as pose variation (front, non-front), occlusion, image orientation, illuminating conditions, and facial expression (Sonia Ohlyan, 2013).

There are other biometric ways to identify individuals and their claimed identity and these include fingerprint scanning, retina scan, DNA analysis, etc. What makes face recognition different from other biometric techniques—the surreptitious answer—is face recognition does not require an individual's participation for recognition (Shailesh Wadhankar, 2018). This holds the key for considering face recognition as the primary biometric technique for authentication and identification. Use cases for face recognition are spread across various domains which range from national security, cybersecurity to smart home security systems. In the past 10 to 15 years, the data generated from research has grown manifolds and so has the safety and security concerns of individuals' lives; hence, face recognition has been in consideration for quite some time to address these concerns (Peili, 2019).

Face recognition can be divided into two—face authentication/identification, and face recognition. Face authentication involves the process of matching the queried face with a given dataset of images to authenticate identity. Face recognition is a category of biometric software that maps an individual's facial features mathematically and stores the data as a face dataset. The software uses deep learning algorithms to compare a live capture or digital image to the stored dataset to verify an individual's identity (TechTarget, 2019).

China's facial recognition technology sector has found itself on a new development path: exporting artificial intelligence (AI) technologies to the vast land of Africa under the framework of the China-proposed Belt and Road initiative. Cloud Walk Technology Co, a domestic start-up based in Guangzhou, capital of south China's Guangdong Province, signed a strategic cooperation agreement with the Zimbabwean government in March on a mass facial recognition project, marking the entry of China's AI technology into Africa (Hongpe, 2018). The rollout of Huawei's smart CCTV technology into Uganda and Kenya is the latest example of China exporting its artificial intelligence to developing countries, particularly in Africa. Based on the recent news, Uganda police have confirmed that Chinese telecommunications giant, Huawei, is rolling out a massive surveillance system that uses facial recognition and other artificial intelligence software to fight crime in the central Africa country (Tom Wilson, 2019). The project, which includes the nationwide installation of the Huawei closed circuits television cameras, has been rolled out in more than 200 cities around the world. In Nairobi, the capital of neighboring Kenya, Huawei's small spherical cameras are ubiquitous in the city centre, perched on slender lamp posts at most intersections.



For the case of the United States, facial recognition systems are currently being studied or deployed for airport security and it is estimated that more than half the United States population has already had their faces captured (Tom Wilson, 2019). The human face is crucial for identity and it is the feature that best distinguishes a person. Face recognition is an interesting and challenging problem and impacts important applications in many areas such as identification for law enforcement, authentication for banking and security systems access, and personal identification, among others (Peng Wu, 2019). Face recognition is an easy task for humans but it is an entirely different task for a computer. Very little is known about human recognition to date on how to analyze an image and how the brain encodes faces and the inner features (eyes, nose, mouth) or outer features (head shape, hairline) which are used for successful face recognition. Neurophysiologists David Hubel and Torsten N. Wiesel (David H., 2004) have shown that our brain has specialized nerve cells responding to specific local features of faces, such as lines, edges, angles, or movement. Since we don't see the world as scattered pieces, our visual cortex must somehow combine the different sources of information into useful patterns.

Nowadays, face detection is used in many places, especially websites hosting images like Picassa, Photobucket, and Facebook. The automatically tagging feature adds a new dimension to sharing pictures among the people who are in the picture and also gives the idea to other people about who the person in the image is.

In this project, we intend to implement a simple but effective face identification or detection and recognition of students' attendance system using Azure cognitive services. With this system in place, the university will be able to implement all attendance related policies that could not be implemented due to lack of an efficient attendance management system.

## **METHODOLOGY**

This chapter describes the approach that was followed to achieve the specified objectives of this research with an elaboration of the reason as to why the chosen approach is preferred. It consists of the techniques and tools that were used in data collection, study, and analysis of the research to further comprehend the requirements of the system. The feasibility analysis tools and techniques of the proposed system, the system implementation, testing, and validation processes are all covered in this chapter.

### **System Study Tools and Techniques**

A critical step towards the successful development of an effective information system is the careful gathering and analysis of relevant data. Under the system study, data on the attendance management system were collected. Several fact-finding tools and techniques can be employed to accomplish this objective; however, to get an effective as well as a timely result, three of these fact-finding techniques were used. These are questionnaire surveys, interviews, and observation.

### **Questionnaire Surveys**

A questionnaire survey was used to discover issues with the current system and the needs of the proposed system and was conducted using a form that contains a series of questions for



which we wished to find an answer (questionnaire). The questions were answered by selected respondents from among the system users (admins, lecturers, and students), either verbally or written. It is an effective tool that was used to gather standardized data from respondents. The data collected were then represented in graphs and charts for further analysis.

### **Interviews**

A semi-structured interview was conducted to gather information from system administrators and the university management on the current system, and to clarify relevant management needs on the proposed system. Interviews are generally conducted using an interview guide which contains a list of topics, themes, and areas to be covered. The interview enabled us to broadly discover issues with the current system and to explore a participant's unique point of view and motives.

### **Observations**

This fact-finding technique was used to understand the current operating procedures of managing attendance. The challenges that are inbuilt into the current conventional attendance methods used were fully understood using this technique. It mainly focused on understanding the efforts involved in the traditional methods and how this could be electronically improved. It was also used to verify the information that was gathered using the questionnaires, and observations fact-finding techniques and was conducted using an observation guide.

### **System Analysis**

This involved studying the data collected to extract the desired information. The data collected was studied to identify the functional and non-functional requirements of the system. Several tools could be applied to analyze the data acquired; however, the analysis of the quantitative data gathered under this study was performed using Statistical Package for the Social Sciences (SPSS). The qualitative data collected, on the other hand, was analyzed using non-statistical/non-numerical methods of data analysis as these data are not quantifiable.

### **System Design**

In the system design phase, user requirements were transformed into some suitable form that helped system developers in system implementation, starting with the logical system design and then proceeding to the physical system design. Object-Oriented Design (OOD) were used to carry out these tasks because OOD provides various visual tools and diagrams for process and data modeling that are very effective for communication and can be easily understood among various stakeholders. OOD provides tools for process modeling as well as data modeling. Process modeling deals with the representation of how data flows from within the system. The tools that were used for this are Context Diagram, Data Flow Diagram (DFD), and Use Case Diagram. Data modeling, on the other hand, deals with the representation of how data was stored in the database. It consists of the conceptual data model, logical data model, and the physical data model. The tool that will be used for this is an Entity Relationship Diagram (ERD).



## **System Implementation**

The system was developed using Microsoft Azure cognitive service (Azure CS) and OpenCV as described in the project title. To minimize cost, face detection and other image processing tasks were handled by OpenCV to reduce the number of API calls to Azure CS as OpenCV is free while Azure CS is not.

Python3 was the chosen programming language for this project because its community widely supports Azure CS and OpenCV which simplifies the development process. The user interface was developed using the Tkinter Python module.

The Tkinter module ("Tk interface") is the standard Python interface to the Tk GUI toolkit from Scriptics (formerly developed by Sun Labs). Both Tk and Tkinter are available on most UNIX platforms, as well as on Windows and Macintosh systems. Starting with the most recent release (8.0), Tk also offers the native look and feel on all platforms (Lundh, 1999).

The data stored in the system has to be accessed by multiple users (admins and lecturers). Therefore, it becomes a necessity to store this data in the cloud. There are various cloud storage services available. However, to avoid integrating many APIs into our system as it could potentially introduce vulnerabilities to the system, the cloud storage provided by Azure CS was used to store images and the Azure SQL database was used for storage of text data.

## **System Testing and Validation**

To guarantee the quality of a software system, it is necessary to test the system using various testing techniques. Developing an error-free system is unattainable; it is important to execute software to find and fix as many errors as possible before it is deployed.

The tests were designed in a way that ensures that the system meets its specifications such as the functionalities that the users expect of it, performance, understandability, and other system features. It was performed at different stages of the system development. Each component of the system was tested individually at the time of development and was also tested when integrated with other components.

There are many software system testing types. However, not all the testing types were used in this project as we do not need to apply all these software testing types to achieve our objective. The testing types that were implemented in this project are:

### **Unit Testing**

Individual units/components of the system were executed to make sure that it performs its expected functionality correctly. Generally, a software system is built in modules. This test focuses on these small modules of the system; hence, it was implemented on each module.

The approach used depends on the unit being tested but in general, it involves providing the module with mock data to see what it outputs.





## **Integration Testing**

This test was carried out to uncover bugs associated with component interaction. It was carried out after all parts of the system were integrated. It was conducted to ensure that the backend and the frontend part of the system work properly together.

As mentioned earlier, a typical software system is constructed in modules. These modules depend on one another and work together to achieve the desired system functionality. There is a need to test the modules after they have been put together to ensure that the modules work together as expected.

## **Performance and Stress Testing**

Under this type of test, the run-time system performance was investigated with emphasis on the time it takes for the system to perform the necessary processing and respond to the user.

Performance testing is an important testing type in this project. This is because the system that was delivered is expected to be real-time. Therefore, the system must perform its computations within a short predictable time. The system was tested to ensure that it does not take more than the expected time to respond.

Performance testing occurs throughout all steps in the testing process. Even at the unit level, the performance of an individual module may be assessed as white-box tests are conducted (Kalaisankaran, 2013).

Stress testing was carried out to ensure that the system can handle stress such as too many database queries being executed without performance degradation. It was conducted to guarantee that the system can perform as expected under peak loads.

The performance and stress testing were done by users because it is the user that has the perception of how the system should perform and how much stress it should handle. The participant (user) interacted with the system and evaluated the system response time relative to him/her.

## **User Acceptance Testing**

This test was conducted to confirm that the system performs and behaves as the users expect it to. The system user-friendliness was tested under this test. The understandability of the system (how easily new users can operate the system) were also considered.

The user acceptance test was conducted by both system developers and end-users. The developers tested the system to confirm that it provides all the functionalities as specified in the software requirement specifications. End-users participated in this test as well to confirm that no functionality has been omitted and to assess the user-friendliness of the system. There was no defined methodology for how end users were going to perform this test.



## **System Study, Analysis, and Design**

This section describes the result of the research and investigations carried out using the tools and techniques stated in the preceding chapter 3. The data collected as well as the meaning and conclusion derived from it and the design of the system is presented in this chapter.

### **System Study**

The result of the research gathered showed that attendance at most institutions in Uganda is carried out manually. Students from universities were selected and – lecturers from --- institutions were selected. The universities were selected using non-probability sampling because of the effect and restrictions of COVID-19. A total of --- questionnaires were distributed using survey monkeys and the response rate was 100%. --- faculty deans and -- quality assurance officers were interviewed and the current system methods of attendance at - -- universities were observed. The study from students gathered data shows that ---% of the institutions used manual attendance systems as ---% uses automated attendance management systems and the remaining ---% do not use either of the two; in other words, attendance is not considered at their institutions. However, the analysis of data gathered from lecturers shows different results. ---% of the lecturers claimed that their university uses an automated attendance system and ---% reported that a manual system is used in their universities.

An interview with a dean and quality assurance shows the significant effect of attendance in quality education. Students' results are directly linked to students' attendance. However, the significant challenge faced in implementing attendance related policies is the inefficiency of the current systems to provide timely information to support management decisions.

To conclude, the research shows the need for a reliable attendance system using biometrics as none of the respondents' universities were found to have any and facial biometrics was recommended by ---% of the respondents.

### **System Analysis**

This sub-section defines the specifications of the proposed systems as gathered in the system study. The specifications were broken down into three sections:

#### **User Requirements**

- i. The system should provide full support for management decisions.
- ii. The data used in the system should be stored in a way that will give simultaneous access to multiple users.
- iii. The system should only store the computed facial features of students when training and not their actual pictures.
- iv. The system should be available for use at any hour of the day. Unjustifiable system shutdown should be avoided and justifiable ones should be kept at a minimum.
- v. Operating instructions and maintenance manuals should be provided to users and delivered with the system.



- vi. Users should be trained on how to take full advantage of the functionalities provided by the system.

### **Functional Requirements**

- i. The system should verify, using face authentication, the face of a lecturer and should display the lecturer information related to his/her class only.
- ii. The lecturer should be able to view the list of his class students and search for those using either name or an identification number.
- iii. The lecturer should be able to start and end a class.
- iv. The system should support the lecturer in managing his/her lectures by providing him/her with tools to monitor students such as reading students' emotions, zooming students' images and viewing individual faces, information about classes missed, and classes conducted, and information about turn-up percentage of students.
- v. The system should provide a feature to allow lecturers to take and save a snapshot from the class live video when conducting events such as tests and examinations.
- vi. The system should allow the lecturer to see on-demand the list of absentees and attendees separately.
- vii. The system should provide the lecturer with details regarding his class in a visual format using graphs and charts. The details include the gender of his students, the ratio of attendees to the total number of students, the ratio of faces found to the faces recognized, and the weekly start time of previously conducted classes.

### **Non-functional Requirements**

- i. Performance Requirements: The system should be capable of handling both large and small workloads and should have a quick response time. Performance degradation should be avoided.
- ii. Efficiency: The system should avoid unnecessary use of system resources.
- iii. Security Requirements: The system should protect and avoid as much as possible any unauthorized access to the user's data.
- iv. Maintainability: The system should be designed in such a way that it is easy to incorporate future requirements and in such a way that updating the system should not be a bottleneck.
- v. Portability: The system should be easily portable to any system that has met the minimum system requirements.
- vi. Correctness: The system should perform its computations with accuracy and precision.
- vii. Reliability: The system should have a minimum of downtime. The service of the system should be readily available for users.



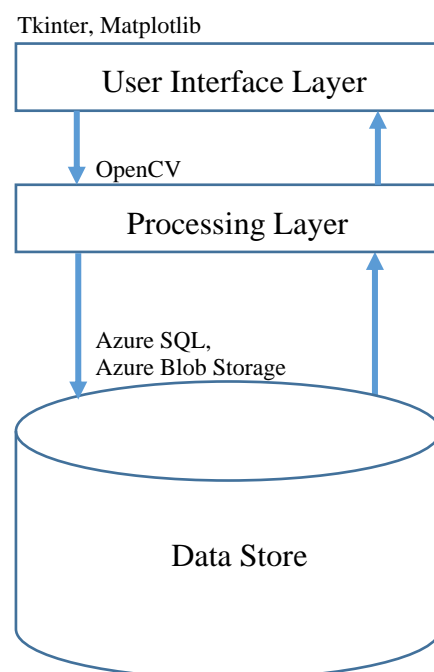
- viii. Ease of Use: The system should be easy to understand and used by all categories of users.

## System Design

System design contains the diagrammatic representations of the abstract system architecture, the flow of data within the system, and how data is processed and stored.

### Architectural Design of the System

As with most management systems, this system was implemented using the three-tier architecture model. The architecture comprised a graphical user interface at the top tier; it also comprised a complex middle-tier consisting of the application logic such as local image processing, and acting as a communication interface between the GUI and data store. The bottommost tier is the data store tier which consists of the application database and image storage hosted on the Azure cloud. Diagrammatically, the architecture is as shown below:



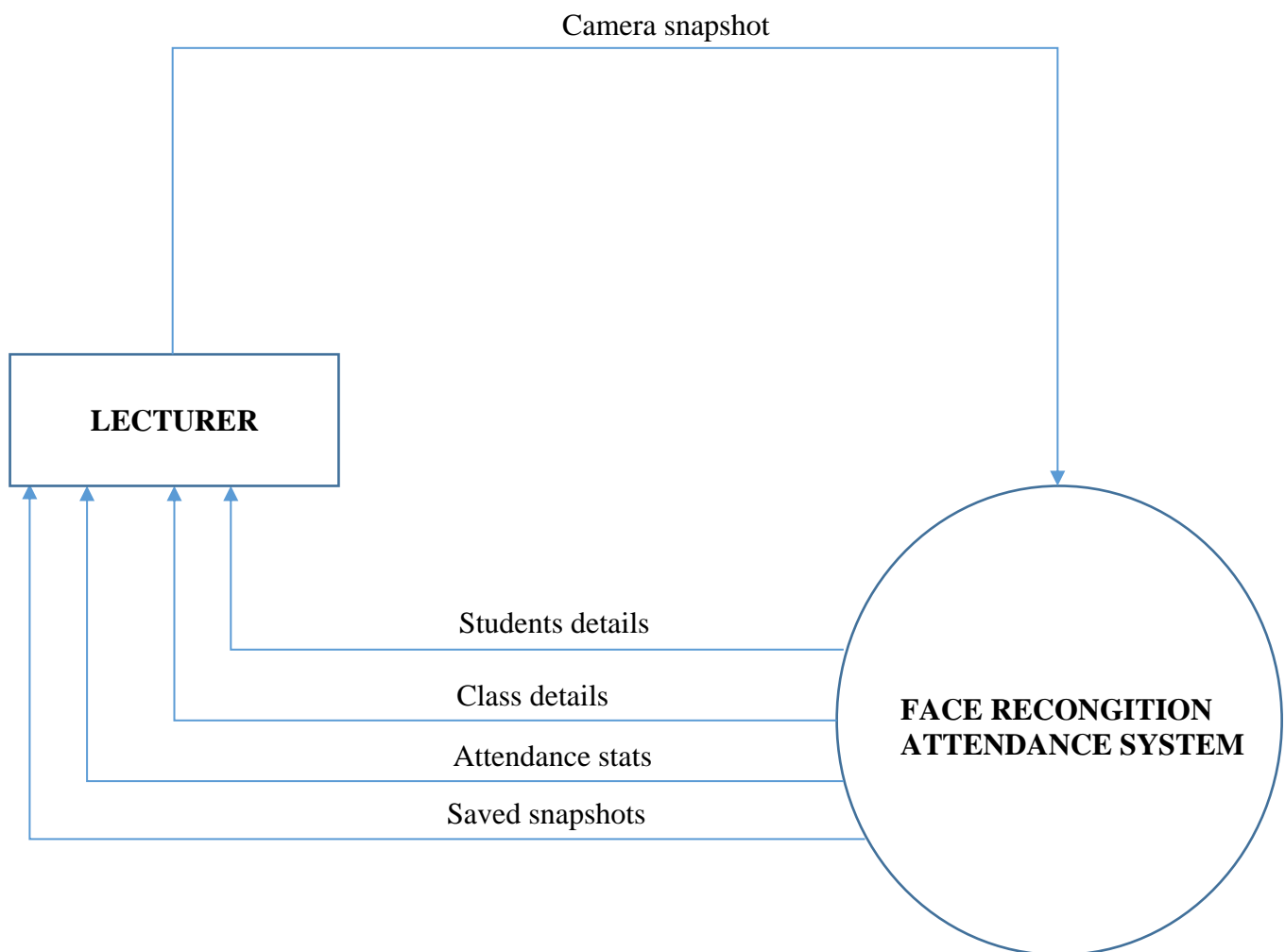
**Figure 1: System Architecture**

## Process Modelling

Process modeling tools were used to provide a graphical representation of the flow of data to and from the system, together with the logic that manipulates the data, data storage, and an abstract representation of the system output to an external entity.

### 1(i) Context Diagram

Context diagram (Level 0 DFD) was used to show the boundary of the system and its environment, and the external entities that interact with the system. The context diagram of the system is illustrated below:





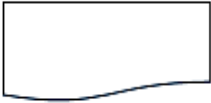


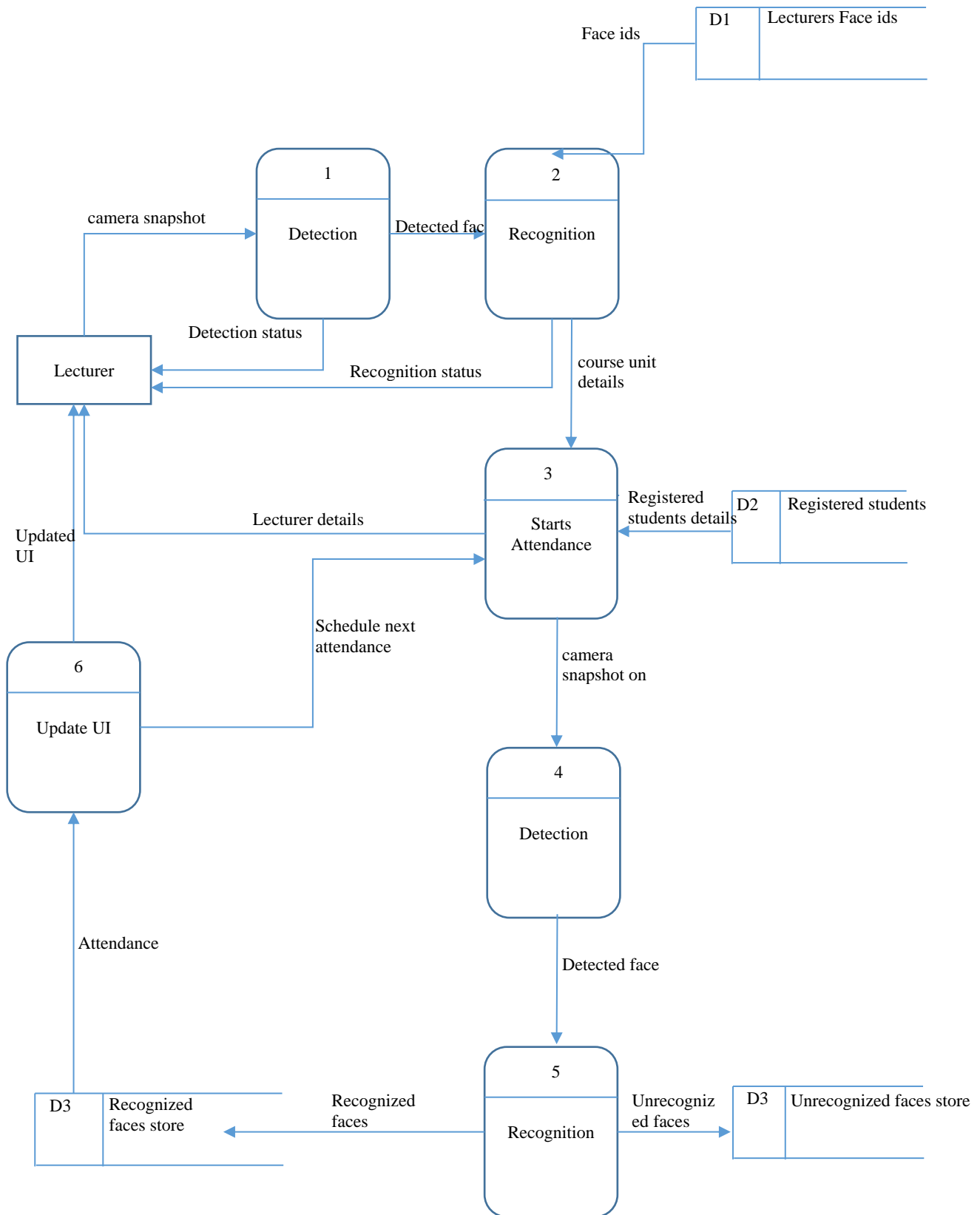
**Figure 2: Context Diagram of the Face Attendance System**

### 1(ii) Data Flow Diagram

DFD was used to show the movement of the data to and from the system. It shows a graphical representation of the data that will be given as an input to the system, the necessary processing that will be performed by the system, the data to be stored, and the output to an external entity. Symbols used in DFD are described below:

**Table 1: Symbols used in DFD**

| <b>Name</b>     | <b>Symbol</b>   | <b>Description</b>   |
|-----------------|---|--|
| External Entity |    | Entities outside the system that supply input to the system or receives output from the system   |
| Data Flow       |    | Shows data passage within the system. The arrow head shows the direction of the movement   |
| Process         |  | Shows a task performed by the system. Each process receives one or more input and produces one or more output either to an external entity, to another process or to data store. |
| Data store      |  | Data store is a data repository. Processes enter data into a data store or retrieve data from data store.  |
| Data report     |  | Represents reports that are generated by the system and are produced as output to an external entity   |



**Figure 3: Data Flow Diagram of the Face Attendance System**



### (iii) Data Dictionary of Level 1 DFD

The design objects used in the construction of the level 1 data flow diagram above are described below. The data dictionary clearly explains the processes, data stores, data flows, and external entities as used in the DFD above.

**Table 2: Description of Processes**

| Process           | Description   |
|-------------------|---|
| Detection         | The detection process finds and extracts the location of faces if there is any in an image        |
| Recognition       | The extracted faces from detection tasks are processed to recognize the detected face.            |
| Starts Attendance | Fetches the face IDs of the students that belong to this class and starts the camera for snapshot |
| Update UI         | Updates the screen of users with the newly captured details of the class                          |

**Table 3: Description of Data Stores**

| Data Store          | Description  |
|---------------------|--|
| Lecturers Face IDs  | Stores the list of all the face IDs of lecture used for recognition purpose  |
| Registered Students | Stores the list of students, their face IDs and the courses they undertake   |
| Unrecognized Faces  | Stores images of people whose faces are not recognized. In other words, it stores the faces of unknown people present in a lecture |

**Table 4: Description of External Entities**

| External Entity | Description   |
|-----------------|---|
| Lecturer        | The lecturing is responsible for starting the system and then placing his/her face near the camera for a snapshot. The systems detect and recognized the face from the snapshot and give the lecturer access to the information related to his/her course |

### Data Modelling

Data modeling tools were used to model how data is stored in the system and the relationship between the data stored. The system was developed using a relational database; therefore, an Entity Relation Diagram is best suited for modeling the systems' database. The ERD was developed by identifying the data requirements, and entities, and attributes of the system.



### 1(iii) Data Requirements

- (a) **Absentees:** This stores information about absent students. The details captured include the students\_personID, course\_units\_code, lecture\_date.
- (b) **Course\_units:** This stores information about course units such as Research Methods, Object-Oriented Programming, to mention but a few. The details captured include the code, name courses\_code, credit\_units, and start\_date.
- (c) **Courses:** This stores information about courses such as Software Engineering, Computer Science to mention but few. The details capture includes the code, name, and faculties\_ID.
- (d) **Faculties:** This stores information about faculties. The details captured include the ID and name.
- (e) **Lecturers:** This stores information about lecturers. The details captured include the personID, firstname, last name, and course\_units\_code.
- (f) **Students:** This stores information about students. The details captured include the personID, firstname, last name, other names, registrationNo, gender, email, and phone.
- (g) **students\_has\_course\_units.** This stores information about the course units that a student undertakes. The details captured include the students\_personID, course\_units\_code, and status.

### 1(iv) Identification of Entities and Attributes

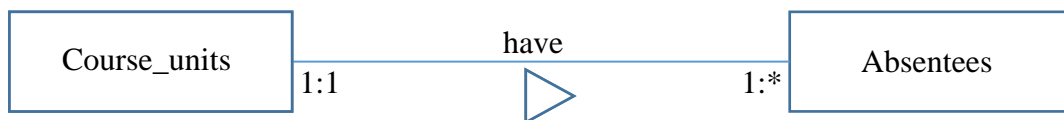
**Table 5: Entities and Associated Attributes**

| Entities     | Attributes  |
|--------------|---|
| Absentees    | students_personID<br>course_units_code<br>lecture_date          |
| course_units | code (pk)<br>name<br>courses_code<br>credit_units<br>start_date |
| Courses      | code(pk)<br>Name<br>faculties_ID                                |
| Faculties    | ID(pk)<br>faculty   |





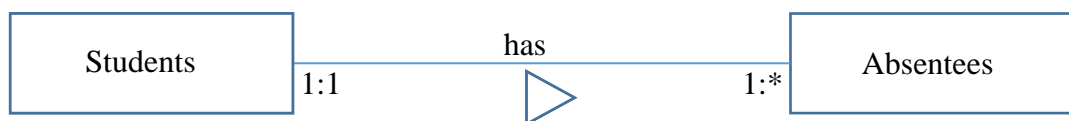
|                           |  |
|---------------------------|--|
| Lecturers                 | person ID(pk)<br>firstname<br>last name<br>course_units_code   |
| Students                  | person ID(pk)<br>firstname<br>last name<br>other names<br>registrationNo<br>gender<br>email<br>phone |
| students_has_course_units | students_personID<br>course_units_code<br>status   |



**Figure 5: Students-Absentees Relationship**

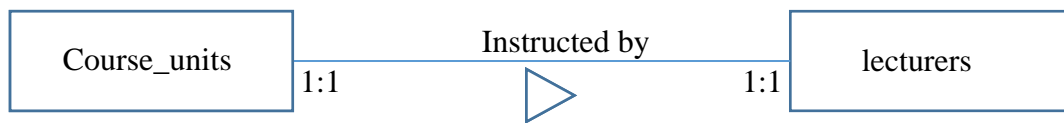
**1(v) Modeling of Relationship Between Entities**

The binary relationship between data entities is graphically illustrated here. The diagram shows as well the multiplicity by which these entities are related.



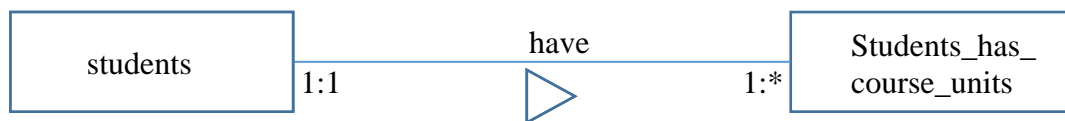
**Figure 6: Course Units-Absentees Relationship**

A student has one or more absentee records and each absentee record belongs to one student. Course unit has one or more absentees and each absentee record belongs to only one course unit.



**Figure 7: Course units-Lecturers Relationship**

A course unit has only one instructor (lecturer) and one lecturer instructs only one course unit.



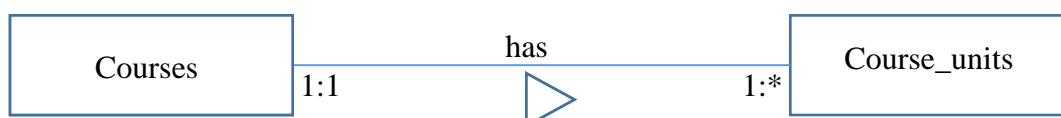
**Figure 8: Students-Students\_has\_course\_units Relationship**

Students have one or more student\_has\_course\_units and one students\_has\_course\_units instance belongs to only one student.



**Figure 9: Faculties-Courses Relationship**

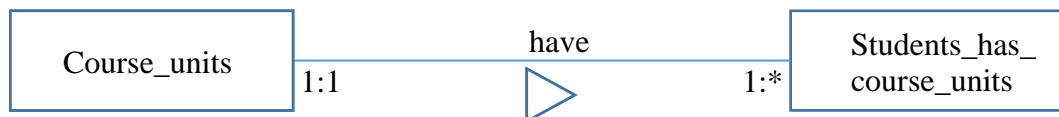
Course units have one or more student\_has\_course\_units and one students\_has\_course\_units instance belongs to only one course\_unit. This and students-students\_have\_course\_units relationship was put in place to fix the anomalies that may arise in many-to-many relationships between students and course units.



**Figure 10: Course\_units-Students\_has\_course\_units Relationship**



A faculty offers many courses and each course is offered by only one faculty. A course has many course\_units and each course\_unit belongs to only one course.



**Figure 11: Course-Course\_units Relationship**

### 1(vi) Entity Relationship Diagram

ERD depicted below shows a graphical representation of the relationships of multiple data entities in a single diagram, the attributes of each entity as well as the multiplicity by which they are related.

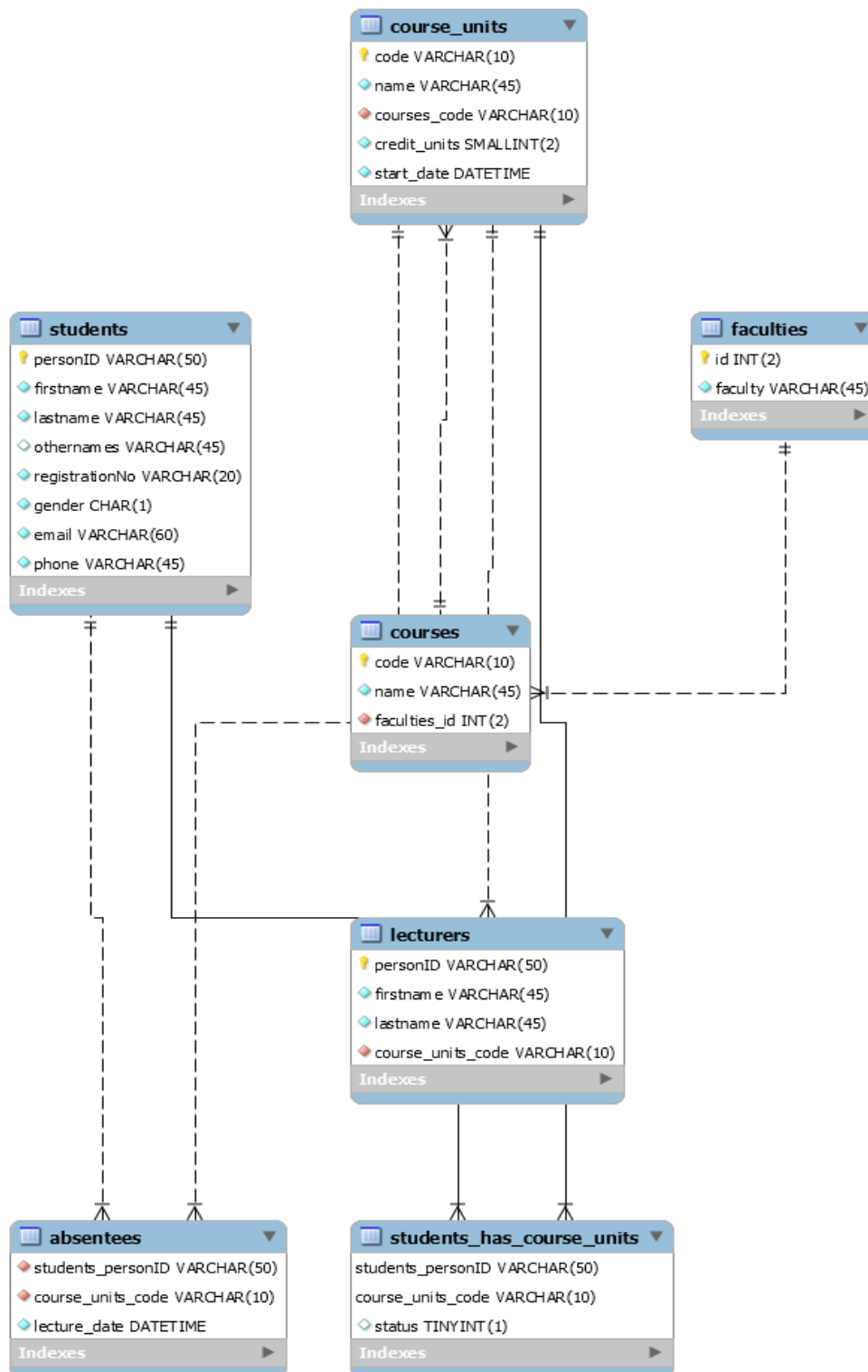


Figure 12: Entity Relationship Diagram



## 1(vii) Structure of Relations for the Database

**Table 6: Absentees Table Structure**

| Field Name        | Data Type   | Constraints   |
|-------------------|-------------|---|
| students_personID | VARCHAR(50) | Not Null, foreign key REFERENCES students(personID) |
| course_units_code | VARCHAR(10) | Not Null, foreign key REFERENCES course_units(code) |
| lecture_date      | DATETIME    | Not Null, default CURRENT_TIMESTAMP                 |

**Table 7: Course\_Units Table Structure**

| Field Name   | Data Type   | Constraints                                    |
|--------------|-------------|--|
| code         | VARCHAR(10) | Not Null, primary key                          |
| name         | VARCHAR(45) | Not Null                                       |
| courses_code | VARCHAR(10) | Not null, foreign key REFERENCES courses(code) |
| credit_units | SMALLINT(2) | Not null                                       |
| start_date   | DATETIME    | Not Null, default CURRENT_TIMESTAMP            |

**Table 8: Courses Table Structure**

| Field Name   | Data Type   | Constraints                                    |
|--------------|-------------|--|
| code         | VARCHAR(10) | Not Null, primary key                          |
| name         | VARCHAR(45) | Not Null                                       |
| faculties_ID | INT(2)      | Not null, foreign key REFERENCES faculties(id) |

**Table 9: Faculties Table Structure**

| Field Name | Data Type   | Constraints           |
|------------|-------------|-----------------------|
| IM         | INT(2)      | Not null, primary key |
| faculty    | VARCHAR(45) | Not null, Unique      |

**Table 10: Lecturers Table Structure**

| Field Name | Data Type   | Constraints           |
|------------|-------------|-----------------------|
| personID   | VARCHAR(50) | Not null, primary key |
| firstname  | VARCHAR(45) | Not null              |
| last name  | VARCHAR(45) | Not null              |



|                   |             |   |
|-------------------|-------------|---|
| course_units_code | VARCHAR(10) | Not null, foreign key<br>REFERENCES<br>course_units(code) |
|-------------------|-------------|---|

**Table 11: Students Table Structure**

| Field Name     | Data Type   | Constraints           |
|----------------|-------------|-----------------------|
| personID       | VARCHAR(50) | Not null, primary key |
| firstname      | VARCHAR(45) | Not null              |
| last name      | VARCHAR(45) | Not null              |
| other names    | VARCHAR(45) |                       |
| registrationNo | VARCHAR(20) | Not null, unique      |
| gender         | CHAR(1)     | Not null              |
| email          | VARCHAR(60) | Not null, unique      |
| phone          | VARCHAR(45) | Not null, unique      |

**Table 12: Students\_has\_course\_units Table Structure**

| Field Name        | Data Type   | Constraints             |
|-------------------|-------------|-------------------------|
| students_personID | VARCHAR(50) | Not null, composite key |
| course_units_code | VARCHAR(10) | Not null, composite key |
| status            | TINYINT(1)  | Not null, default 0     |

### System Implementation, Testing, and Validation

This chapter comprises the tools used to develop the proposed system and the techniques used to test and validate the system inputs as well as the outputs results provided by the system. It also consists of sample screenshots of the user interface and a description of the minimum requirements needed for a machine to be able to run the system.

### Functions Provided by the System

After the system successfully detects and recognizes the lecturer's face as a means of authentication, the system then navigates to the home screen. From the home screen, the system enables the lecturer to start class and then a live video camera will start. The lecturer can scale up and scale down the video, he/she can save a picture snapshot of the camera, and can change the view mode to see only faces from the video. The system reads and provides lecturers with the students' sentiments to assist him/her while conducting his/her lectures.

The lecturer views a list of students that undertake this particular course unit with their registration number, gender, phone, email, status (retake/fresh), and the percentage of the class attended by the student. The system also enables the lecturer to trigger action to see the list of only students that are present at the moment or to see the list of only the absentees. The lecturer is allowed to filter students' lists as well using either gender or status.



The lecturer views information that is related to the class in general such as the number of lectures conducted, the number of lectures missed, percentage turn-up of the entire class, the weekly attendance of the entire class, the weekly lecture start time and attendance per lecture hour.

The process of capturing and saving attendance is completely automated and does not need any further action from the lecturer.

## Sample Screenshots

### Login Screen

The login screen is the first page that appears when the application is launched. A camera automatically starts and captures live video coverage. A snapshot from the camera is processed to detect a face and if a face is found, the detected face is then forwarded for recognition. A successful recognition means the user is authenticated and is allowed to access the system.

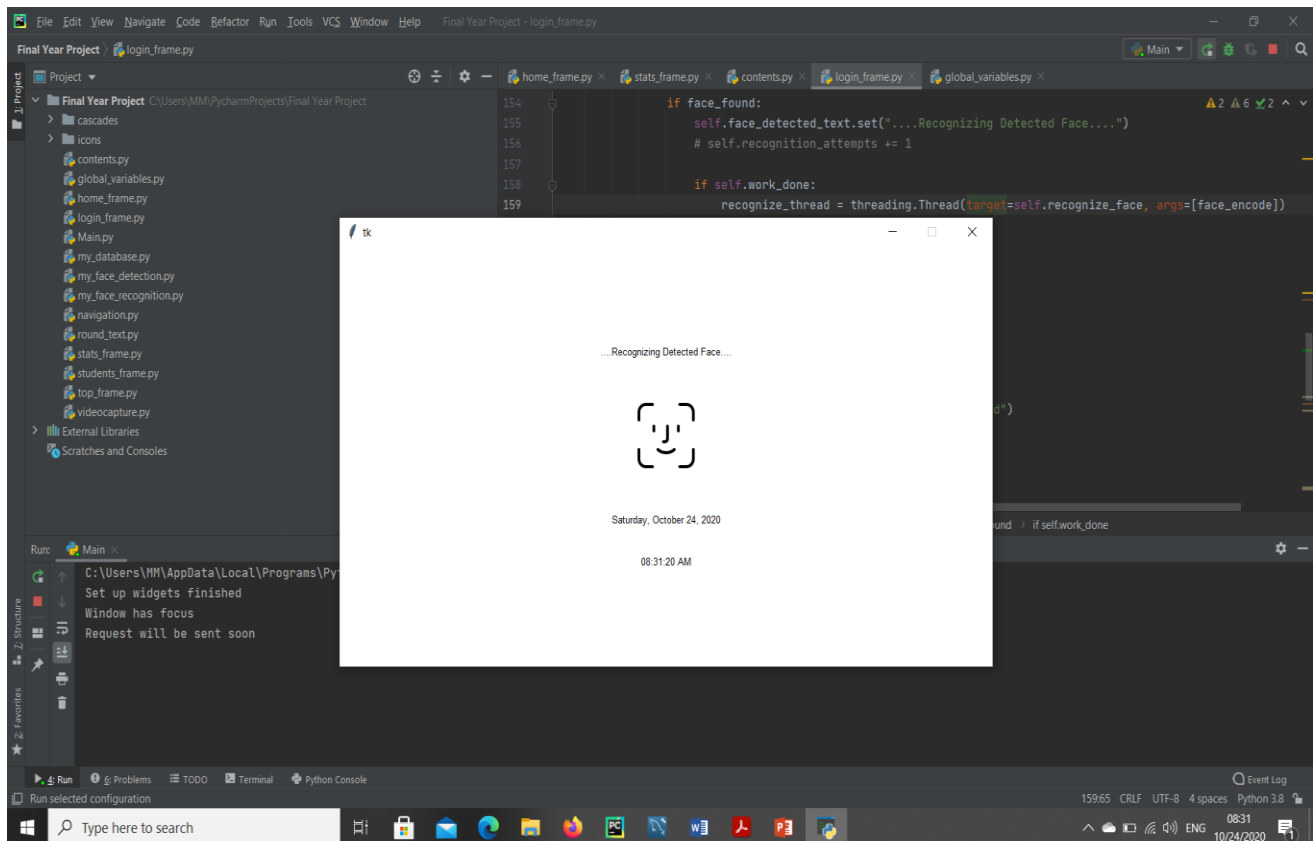
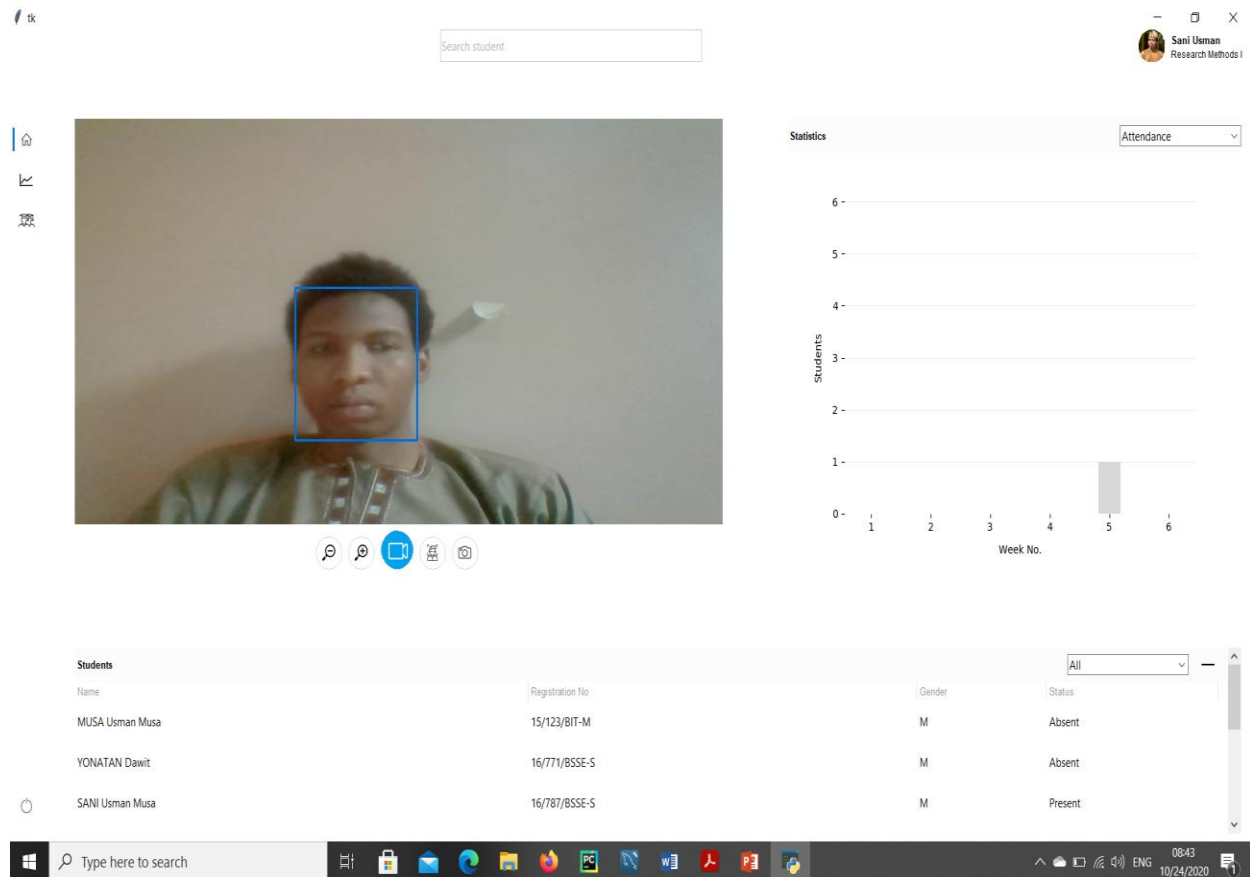


Figure 13: Login Screen

## Home Screen

After successful recognition of the lecturer's face, the system displays the home screen where the lecturer starts a class.



**Figure 14: Home Screen**

## Stats Screen

Displays statistical information about the attendance of the ongoing lecture and the previous lectures as well.

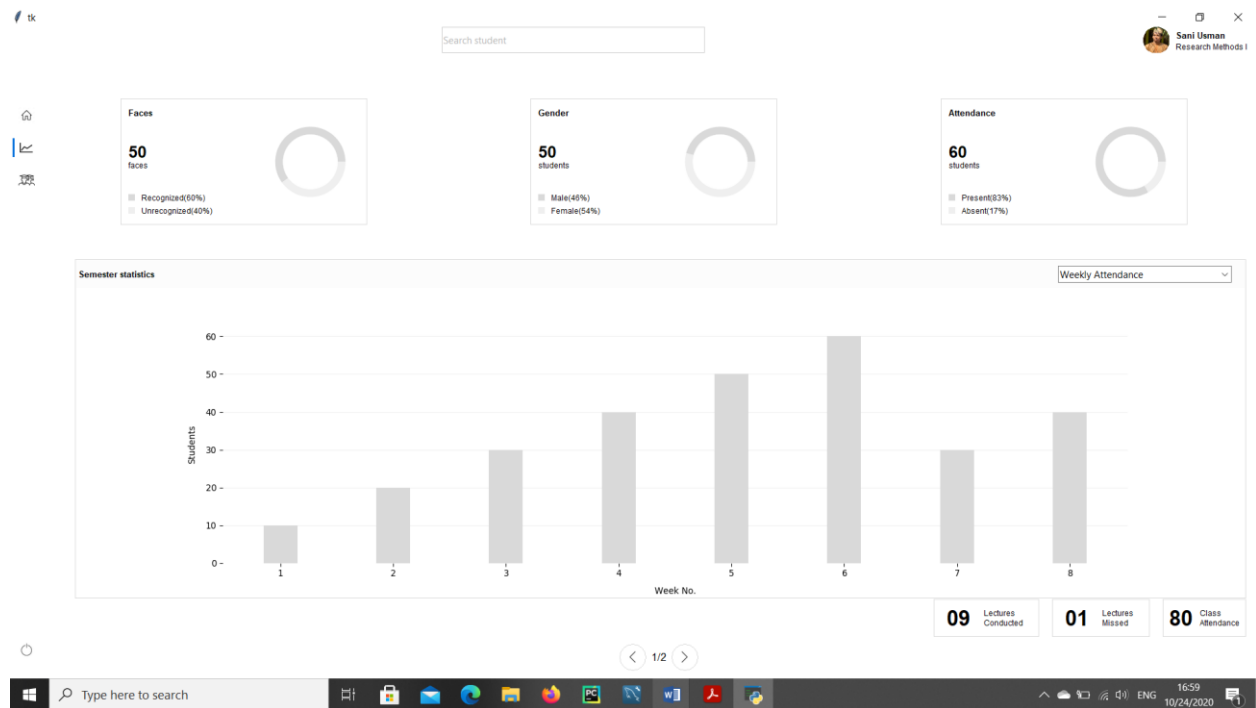


Figure 15: Stats Screen

### Students Screen

Displays the list of students that undertake the course unit. The details displayed include the name, registration number, gender, phone, email, and attendance percentage.

The Students Screen displays a list of students with the following details:

| Name               | Registration No | Gender | E-mail               | Phone          | Attendance(%) |
|--------------------|-----------------|--------|----------------------|----------------|---------------|
| MUSA Usman Musa    | 15/123/BIT-M    | M      | mmumkno@gmail.com    | +2348020607278 | N/A           |
| YONATAN Dawit      | 16/771/BSSE-S   | M      | yoni@gmail.com       | +256787275358  | N/A           |
| SANI Usman Musa    | 16/787/BSSE-S   | M      | usmusa13@gmail.com   | +2348020320987 | N/A           |
| KHADIAH Usman Musa | 16/868/BSSE-S   | F      | usmusa12@gmail.com   | +256705843733  | N/A           |
| AHMAD Usman Musa   | 16/888/BSCS-S   | M      | ahmad@gmail.com      | +2348033120243 | N/A           |
| FATIMA Usman Musa  | 19/789/BSCS-S   | F      | fatima122@gmail.com  | +2561234564    | N/A           |
| ISLAM Usman Musa   | 17/123/BIT-S    | F      | walawasani@gmail.com | +2348069602771 | N/A           |

Figure 16: Students Screen



---

## **System Testing and Validation Result**

The system was tested during and after the development process to ensure that the system meets the functional and non-functional requirements stated and to uncover and fix errors associated with the functionalities provided by the system. Unit testing was carried out on individual modules of the system before they were integrated with other modules. The modules that perform computations were invoked using mock data and the results returned by these modules were checked for validity. Likewise, the system was tested after all the modules have been integrated, a testing technique known as integration testing. This testing technique is aimed at identifying and fixing defects that are related to component interactions. The performance of the system was investigated as well using a testing type known as performance testing. Stress testing was conducted to ensure that the system can handle both small and peak loads.

A test was conducted to confirm that the system performs and behaves as the users expect it to. This test is known as user acceptance testing and it is conducted by end-users and system developers as well. The table on the next page shows the requirements traceability matrix of the system with test cases, test data, test steps, expected results, status, and test execution date.

**Table 13: Requirements Traceability Matrix**

|             |                |                     |   |               |  |   |  | TEST EXECUTION |        |      |
|-------------|----------------|---------------------|---|---------------|--|---|--|----------------|--------|------|
| TEST CASE # | REQUIREMENT ID | TEST CASE           | TEST DESCRIPTION  | TEST CASE ID. | TEST STEPS   | TEST DATA   | EXPECTED RESULT  | TESTED BY      | STATUS | DATE |
| 1           | Rq01           | Verify Log in       | It includes testing the login of the system using 2 scenarios: Testing with a known face and testing with an unknown face | Rq01: TC1     | 1) Start the system<br>2) Stand in front of the webcam for the system to take a snapshot                   | Known face snapshot   | Successful log in  | Sani Usman     |        |      |
| 2           |                |                     |   | Rq01: TC2     | 1) Start the system<br>2) Stand in front of the webcam for the system to take a snapshot                   | Snapshot of an unknown face   | FAILED login. Notify user that the face is not recognized                          | Sani Usman     |        |      |
| 1           | Rq02           | Search for students | This test validates the search functionality provided by the system.  | Rq02: TC1     | 1) Go to the home screen<br>2) At the top entry, enter the student's registration number<br>3) Press enter | <ul style="list-style-type: none"> <li>Registration Number: 16/868/BS SE-S</li> </ul> | Highlight the student with the given registration number from the list of students | Sani Usman     |        |      |



|   |  |  |  |              |  |  |   |               |  |  |
|---|--|--|--|--------------|--|--|---|---------------|--|--|
| 2 |  |  |  | Rq02:<br>TC2 | 1) Go to the home screen<br>2) At the top entry, enter the student's name<br>3) Press enter                | • Name:<br>Sani<br>Usman<br>Musa   | Highlight the student with the given name from the list of students | Sani<br>Usman |  |  |
| 3 |  |  |  | Rq02:<br>TC3 | 1) Go to the home screen<br>2) At the top entry, enter the student's registration number<br>3) Press enter | Registrati<br>on<br>Number:<br>Any<br>invalid<br>registratio<br>n number | Alert the user that no student with such registration number        | Sani<br>Usman |  |  |
| 4 |  |  |  | Rq02:<br>TC4 | 1) Go to the home screen<br>2) At the top entry, enter the student's registration number<br>3) Press enter | • Name:<br>Any<br>invalid<br>name  | Alert the user that no student with such registration number        | Sani<br>Usman |  |  |





|   |      |                           |   |           |   |              |  |            |  |  |
|---|------|---------------------------|---|-----------|---|--------------|--|------------|--|--|
| 1 | Rq03 | Starts and Stop Class     | This test validates whether the system starts the webcam on click of the start button and stops the webcam when clicked again                 | Rq05: TC1 | 1) Go to the home screen<br>2) Click the webcam icon                              | No test data | If the webcam is already on, the system should stop the webcam else the system should start the webcam | Sani Usman |  |  |
| 1 | Rq04 | Providing lecture support | The system provides the user with some tools to help in managing his lectures: emotions, zooming students images and viewing individual faces | Rq04: TC1 | 1) Go to the home screen<br>2) Click the webcam icon                              | No test data | The system reads the emotions of the students and updates the emotions bar graph                       | Sani Usman |  |  |
|   |      |                           |   | Rq04: TC2 | 1) Go to the home screen<br>2) Click the webcam icon<br>3) Click on the Zoom icon | No test data | The system zooms the video frames  | Sani Usman |  |  |



|   |      |                       |   |              |   |              |  |            |  |  |
|---|------|-----------------------|---|--------------|---|--------------|--|------------|--|--|
|   |      |                       |   | Rq04:<br>TC3 | 1) Go to the home screen<br>2) Click the webcam icon<br>3) Click on the shrink icon | No test data | The system shrinks the video frames  | Sani Usman |  |  |
|   |      |                       |   | Rq04:<br>TC3 | 1) Go to the home screen<br>2) Click the webcam icon<br>3) Click on the faces icon  | No test data | The system shows only faces from the video frame   | Sani Usman |  |  |
| 1 | Rq05 | Save camera snapshots | The system saves a snapshot on the lecturers' local machine | Rq05:<br>TC1 | 1) Go to the home screen<br>2) Click the webcam icon<br>3) Click on the camera icon | No test data | A save dialog box pops up and the user chooses the storage location. The system saves the image in the given location and shows the image using the default photo viewer in the system | Sani Usman |  |  |



|   |      |                 |   |           |  |              |  |            |  |  |
|---|------|-----------------|---|-----------|--|--------------|--|------------|--|--|
| 1 | Rq06 | Filtering lists | Students list is filtered to see absentees only, attendees only or both | Rq04: TC1 | 1) Go to the home screen<br>2) Click the webcam icon<br>3) Click on the drop-down at the top right of the students' list and choose a filtering criteria | No test data | Students list gets updated and shows only students that match the filtering criteria | Sani Usman |  |  |
|---|------|-----------------|---|-----------|--|--------------|--|------------|--|--|



## **System Requirements**

This describes the minimum hardware and software requirements needed by the system for proper functioning. The hardware minimum specifications are a webcam, processor speed of 2.4Hz or higher, computer memory of 2GB or higher, and 10GB of hard disk space. The software minimum specifications are An operating system (Windows, Mac OS, and Linux).

## **CONCLUSION**

A facial biometrics attendance system was developed to simplify the process of capturing attendance and to improve on the integrity of the procedures used in attendance management by using biometrics as an attendance criterion. The development process strived to ensure that the system requirements specified were achieved.

The study conducted during this project reflected the significance of attendance at institutions of learning and the compelling need for an efficient and reliable attendance system. Facial biometric is distinctive from other biometrics as it can be deployed without individuals' participation. This makes facial biometrics most appropriate to be deployed as an attendance mechanism. Therefore, it is highly recommended that institutions should deploy facial biometrics as a new mechanism to capture reliable attendance data in the most efficient form.

## **Future Work**

The features that could be integrated into the system to fill up the gaps left by this system are listed below:

- i. An algorithm to detect when a face is not from a physical person to remove the need for a 3D camera and/or infrared camera.
- ii. Save the pictures of an unknown person and immediately alert the security officers.
- iii. Send a reminder to the student when his/her attendance drops below a certain threshold.
- iv. Send a notification to management when a lecture is missed or when it starts late.
- v. Allow for a lecturer to be registered for more than one course unit. Upon login, the system should determine using the login time the right course unit.
- vi. A feature for creating and managing timetables.

## **Challenges and Limitations**

The following challenges were faced by the researcher in the process of conducting the research:

- i. Difficulties in gathering data with COVID-19 restrictions in place.
- ii. Limited financial resources to pay for Azure services.



## REFERENCES

- (2020, August 14). Retrieved from Tutorialspoint:  
[https://www.tutorialspoint.com/management\\_information\\_system/mis\\_quick\\_guide.htm](https://www.tutorialspoint.com/management_information_system/mis_quick_guide.htm)
- Agam, G. (2006). *Introduction to Programming with OpenCV*. Illinois: Illinois Institute of Technology.
- Altun, A. A., & Kolus, C. (2018). Face Verification System In Mobile Devices By Using Cognitive Services. *International Journal of Intelligent Systems and Applications in Engineering*, 294-298.
- Azure SQL Database documentation*. (2020, March 31). Retrieved from Microsoft Azure:  
<https://docs.microsoft.com/en-us/azure/sql-database>
- Bendoly, E. (2008). *Excel Basics to Blackbelt: An Accelerated Guide to Decision Support Designs*. Cambridge University Press.
- Berisha-Shaqiri, A. (2014). Management Information System and Decision-Making. *Academic Journal of Interdisciplinary Studies*, 19-23.
- Bisaso, R. (2009). International Federation for Information Processing IFIP. *Information Technology for Educational Management at a Ugandan Public University*, 83-94.
- Bisaso, R., & Visscher, A. (2005). *Computerized Schools Information Systems Usage in an Emerging Country - Uganda*. Enschede.
- Cognitive Services*. (2019, November 12). Retrieved from Microsoft Azure:  
<https://azure.microsoft.com/en-us/services/cognitive-services/>
- Danny, S. (2020, April 1). *SPSS*. Retrieved from Wikipedia:  
<https://en.m.wikipedia.org/wiki/SPSS>
- DAVID H, H. T. (2004). *Brain and Visual Perception: The Story of a 25-year Collaboration*. Oxford Scholarship.
- Dinalankara, L. (August, 2017). *Face Detection & Face Recognition Using Open Computer Vision Classifies*. Plymouth: ResearchGate.
- Dinca, A. D. (2017). *Face Detection & Recognition Report*. Birmingham: University of Birmingham.
- Emami, S. (2010). *Face Detection and Recognition using OpenCV*. Retrieved from [www.shervinemami.info](http://www.shervinemami.info): <http://shervinemami.info/faceRecognition.html>
- Emami, S., & Suci, V. P. (2012). Facial Recognition using OpenCV. *Journal of Mobile, Embedded and Distributed Systems vol.IV no.1*, 38-43.
- Face Recognition Homepag*. (2019). Retrieved from <http://www.face-rec.org/algorithms/>
- Hassan, S., & Asghar, D. M. (2015). *Web-base Attendance Management System*. Khyber Pakhtunkhwa: ResearchGate.
- Hewitt, R. (2010). *Seeing with OpenCV, Article*. Retrieved from [www.cognotics.com](http://www.cognotics.com):  
[http://www.cognotics.com/opencv/servo\\_2007\\_series/part\\_1/index.html](http://www.cognotics.com/opencv/servo_2007_series/part_1/index.html)
- Hongpe, Z. (2018). *Chinese facial ID tech to land in Africa*. Global Times.
- Jacksi, K., Ibrahim, F., & Zebari, S. (2018). Student Attendance Management System. *Scholars Journal of Engineering and Technology (SJET)*, 49-53.
- Kaehler, A., & Bradski, G. (2017). *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. California: O'Reilly Media Inc.
- Kalaisankaran, B. (2013). *Students Attendance Management System*. Pollachi: Dr. Mahalingam College Of Engineering And Technology.
- Kim, Y., & Lee, S. (2020). Exploration of the Characteristics of Emotion Distribution in Korean TV Series: Common Pattern and Statistical Complexity. *Digital Object Identifier*, 69438-69447.



- Kumar, M. (2016). An Incorporation of Artificial Intelligence Capabilities in Cloud Computing. *International Journal Of Engineering And Computer Science* , 19070-19073.
- Laganier, R. (2014). *OpenCV Computer Vision Application Programming Cookbook Second Edition*. Birmingham: Packt Publishing Ltd.
- Lundh, F. (1999). *An Introduction to Tkinter*.
- M. Vineetha Sai, G. Varalakshmi, G. Bala Kumar, J. Prasad. (2017). *Face Recognition System With Face detection*. Kakinada: Pace Institute of Technology and Sciences.
- Majekodunmi, T. O., & E. Idachaba, F. (2011). A Review of the Fingerprint, Speaker Recognition, Face Recognition and Iris Recognition Based Biometric Identification Technologies. *Proceedings of the World Congress on Engineering 2011 Vol II*. London: eprints.covenantuniversity.
- Management Information System*. (2020, 04 24). Retrieved from Wikipedia: [https://en.m.wikipedia.org/wiki/Management\\_information\\_system](https://en.m.wikipedia.org/wiki/Management_information_system)
- Management Information System*. (2020, April 24). Retrieved from Wikipedia: [https://en.m.wikipedia.org/wiki/Management\\_information\\_system](https://en.m.wikipedia.org/wiki/Management_information_system)
- Nieto, M., Johnston-Dodds, K., & Simmons, C. W. (2002). *Public and Private Applications of Video Surveillance and Biometric Technologies*. California: California Research Bureau.
- Peili, P. J. (2019). *Face Recognition in Low Quality Images: A Survey*.
- Peng Wu, H. Z. (2019). Some Analysis and Research of the AdaBoost Algorithm. *International Conference on Intelligent Computing and Information Science* (pp. 1-5). Intelligent Computing and Information Science.
- Qualitative Data Analysis*. (2020, April 2). Retrieved from Research Methodology: <https://research-methodology.net/research-methods/data-analysis/qualitative-data-analysis/>
- R. Rajalakshmi, M. (2013). *A Novel Approach to Face Recognition with Pose and Illumination Variation Using Support Vector Machine as Classifier*. International Journal of Innovative Technology and Exploring Engineering (IJITEE) .
- Shailesh Wadhankar, S. S. (2018). *Real Face Detection and Recognition: The Live Experiment* . International Journal of Computer Applications (0975 – 8887) .
- Sharma, H. (2020, March 31). *Microsoft Azure SQL Database*. Retrieved from IntelliPaat: <https://intellipaat.com/blog/tutorial/microsoft-azure-tutorial/azure-sql/>
- Sheth, A., Yip, H. Y., Iyengar, A., & Tepper, P. (2019). Cognitive Services and Intelligent Chatbots: Current Perspectives and Special Issue Introduction. *Digital Object Identifier*, 6-12.
- Sonia Ohlyan, S. S. (2013). *A Survey On Various Problems & Challenges In Face Recognition* . International Journal of Engineering Research & Technology (IJERT) .
- TechTarget. (2019). *facial recognition*. Retrieved from [www.searchenterpriseai.techtarget.com](http://www.searchenterpriseai.techtarget.com): <https://searchenterpriseai.techtarget.com/definition/facial-recognition>
- Tom Wilson, M. M. (2019). *Uganda confirms use of Huawei facial recognition cameras*. Retrieved from Financial Times: <https://www.ft.com/content/e20580de-c35f-11e9-a8e9-296ca66511c9>
- Types of Software Testing: Different Testing Types With Details*. (2019, November 10). Retrieved from Software Testing Help: <https://www.google.com/amp/s/www.softwaretestinghelp.com/types-of-software-testing/amp>.





---

*What is Data Modelling? Conceptual, Logical, & Physical Data Models.* (2020, April 4).

Retrieved from Guru99: <https://www.guru99.com/data-modelling-conceptual-logical.html>

Wikipedia. (2019). *Three-dimensional face recognition*,. Retrieved from

[www.en.wikipedia.org: http://en.wikipedia.org/wiki/Three-dimensional\\_face\\_recognition](http://en.wikipedia.org/wiki/Three-dimensional_face_recognition)