

**Technological** **University** **of** **the** **Shannon** – **Midwest**

**BSc.** **Internet** **Systems** **Development** **–** **Year** **4**

**Jason** **Price** **–** **K00268716**

**January** **–** **2025**

**FYP**

Contents

[Figures 6](#_Toc192272664)

[Tables 8](#_Toc192272665)

[1. Introduction 9](#_Toc192272666)

[1.1 Research 10](#_Toc192272667)

[1.1.1 Purpose 10](#_Toc192272668)

[1.1.2 Objectives 10](#_Toc192272669)

[1.1.3 Questions 11](#_Toc192272670)

[1.2 Proposed Solution 11](#_Toc192272671)

[2 Literature Review 12](#_Toc192272672)

[2.1 Attendance Systems 12](#_Toc192272673)

[2.1.1 Attendance Taking Systems 13](#_Toc192272674)

[2.1.2 Types of Attendance Taking Systems 13](#_Toc192272675)

[2.1.3 Advantages of Attendance Taking Systems 15](#_Toc192272676)

[2.1.4 Disadvantages of Attendance Taking Systems 15](#_Toc192272677)

[2.1.5 Applications in Attendance Management 16](#_Toc192272678)

[2.1.6 Traditional Identity Verification Methods 16](#_Toc192272679)

[2.1.7 Emergence of Biometric Technology 17](#_Toc192272680)

[2.1.8 Main Biometrics Used 18](#_Toc192272681)

[2.1.9 Advantages of Biometric Technology 19](#_Toc192272682)

[2.1.10 Cloud based attendance systems and Biometric Integration 19](#_Toc192272683)

[2.1.11 Conclusion 20](#_Toc192272684)

[2.2 Facial Recognition 21](#_Toc192272685)

[2.2.1 What is a facial recognition system ? 21](#_Toc192272686)

[2.2.2 Evolution of facial recognition technology 22](#_Toc192272687)

[2.2.3 Techniques for facial recognition 24](#_Toc192272688)

[2.2.4 Advantages of facial recognition systems 24](#_Toc192272689)

[2.2.5 Weakness of facial recognition systems 25](#_Toc192272690)

[2.2.6 Privacy Considerations and issues 25](#_Toc192272691)

[2.2.7 Facial recognition within attendance systems 25](#_Toc192272692)

[2.2.8 Accuracy and reliability 26](#_Toc192272693)

[2.2.9 Anti-Facial recognition masks 27](#_Toc192272694)

[2.2.10 Conclusion 28](#_Toc192272695)

[2.3 GDPR Guidelines 28](#_Toc192272696)

[2.4 QR Codes 30](#_Toc192272697)

[2.4.1 What are QR codes? 30](#_Toc192272698)

[2.4.2 Design of QR codes 30](#_Toc192272699)

[2.4.3 Uses of QR codes 31](#_Toc192272700)

[2.4.4 QR Code technology within education 31](#_Toc192272701)

[2.4.5 Limitations of QR codes 33](#_Toc192272702)

[2.4.6 Conclusion 33](#_Toc192272703)

[2.5 Final Thoughts 33](#_Toc192272704)

[2.5.1 Security and Privacy in modern attendance systems 34](#_Toc192272705)

[2.5.2 User experience 34](#_Toc192272706)

[2.5.3 Future trends within attendance systems 34](#_Toc192272707)

[3 Analysis and Design 35](#_Toc192272708)

[3.2 Analysis 35](#_Toc192272709)

[3.2.1 Introduction 35](#_Toc192272710)

[3.2.2 Requirements Gathering Process 35](#_Toc192272711)

[3.2.3 Existing attendance systems 36](#_Toc192272712)

[3.2.4 Key Insights from Analysis of Existing Systems 36](#_Toc192272713)

[3.2.5 Functional Requirements 38](#_Toc192272714)

[3.2.6 Non-Functional Requirements 39](#_Toc192272715)

[3.3 Design 40](#_Toc192272716)

[3.3.1 Systems Architecture 40](#_Toc192272717)

[3.3.2 Software Design 43](#_Toc192272718)

[3.3.3 Functional and Non-Functional Requirements 45](#_Toc192272719)

[3.3.4 User Interface (UI) Design 45](#_Toc192272720)

[3.3.5 Database Design 48](#_Toc192272721)

[3.3.6 Conclusion 50](#_Toc192272722)

[4 Implementation Report 51](#_Toc192272723)

[4.1 Tools and Technologies 51](#_Toc192272724)

[4.1.1 Programming Languages 51](#_Toc192272725)

[4.1.2 Integrated Development Environments (IDE’s) 52](#_Toc192272726)

[4.1.3 UI/UX Design Tools 54](#_Toc192272727)

[4.1.4 Libraries 55](#_Toc192272728)

[4.2 Environment Configuration 57](#_Toc192272729)

[4.2.1 Development Environment 57](#_Toc192272730)

[4.2.2 Runtime Environment 58](#_Toc192272731)

[4.3 Database Implementation 58](#_Toc192272732)

[4.3.1 Rationale for choosing Firebase Firestore 58](#_Toc192272733)

[4.3.2 Database Structure 59](#_Toc192272734)

[4.4 Feature Implementation 62](#_Toc192272735)

[4.4.1 Key Features 62](#_Toc192272736)

[4.5 Challenges Faced 66](#_Toc192272737)

[4.5.1 Technical Challenges 66](#_Toc192272738)

[4.5.2 Design and Development Challenges 68](#_Toc192272739)

[4.5.3 Time Constraints 69](#_Toc192272740)

[4.6 Conclusion 70](#_Toc192272741)

[5. Testing and Results 71](#_Toc192272742)

[5.1 Introduction 71](#_Toc192272743)

[5.2 Unit Testing of Non-Functional Requirements 71](#_Toc192272744)

[5.2.1 Performance 71](#_Toc192272745)

[5.2.2 Usability 71](#_Toc192272746)

[5.2.3 Security 71](#_Toc192272747)

[5.2.4 Compatibility 71](#_Toc192272748)

[5.3 Non-Functional Requirements Test Cases 72](#_Toc192272749)

[5.3.1 Performance Test Cases 72](#_Toc192272750)

[5.3.2 Usability Test Cases 73](#_Toc192272751)

[5.3.3 Security Test Cases 74](#_Toc192272752)

[5.3.4 Compatibility Test Cases 75](#_Toc192272753)

[5.4 Finding of Non-Functional Testing 76](#_Toc192272754)

[5.4.1 Performance Findings 76](#_Toc192272755)

[5.4.2 Usability Findings 76](#_Toc192272756)

[5.4.3 Security Findings 77](#_Toc192272757)

[5.4.4 Reliability Findings 78](#_Toc192272758)

[5.4.5 Non-Functional Testing Thoughts 79](#_Toc192272759)

[5.5 Unit Testing of Functional Requirements 79](#_Toc192272760)

[5.5.1 Functional Tests 79](#_Toc192272761)

[5.6 Functional Requirements Test Cases 79](#_Toc192272762)

[5.7 Finding of Functional Testing 83](#_Toc192272763)

[5.7.1 Login Functionality Findings 83](#_Toc192272764)

[5.7.2 View Timetable Findings 84](#_Toc192272765)

[5.7.3 Facial Recognition Findings 85](#_Toc192272766)

[5.7.4 Attendance Stats Viewing Findings 86](#_Toc192272767)

[5.7.5 Functional Testing Thoughts 87](#_Toc192272768)

[5.8 Test Data 88](#_Toc192272769)

[5.9 User Testing 89](#_Toc192272770)

[5.10 Summary of Findings 90](#_Toc192272771)

[5.11 Conclusion 90](#_Toc192272772)

[Bibliography 92](#_Toc192272773)

# Figures

[**Figure 1.1:** inflow of university students from European countries (Statista, 2015) 10](#_Toc192271938)

[**Figure 2.1:** A basic RFID System (BRADY, n.d.) 14](#_Toc192271939)

[**Figure 2.2:** Difference between Biometric Characteristics (Ashtopus Technologies, 2015) 18](#_Toc192271940)

[**Figure 2.3:** The basic structure of biometric identification technology (Li, 2015) 19](#_Toc192271941)

[**Figure 2.4:** The geometrical approach to facial recognition (Dangi, 2011) 22](#_Toc192271942)

[**Figure 2.5:** The evolution of facial recognition technology (Pattnaik, et al., 2023) 23](#_Toc192271943)

[**Figure 2.6:** Number of Facial Recognition documents by year (Zennayi, et al., 2022) 23](#_Toc192271944)

[**Figure 2.7:** Neural Network with multiple convolution layers (Raghav, 2018) 24](#_Toc192271945)

[**Figure 2.8:** HAMBox compared to other Anchor matching strategies (Wang, 2021) 27](#_Toc192271946)

[**Figure 2.9:** How a QR is structured (Schulfer, 2020) 31](#_Toc192271947)

[**Figure 3.1:** Survey for students regarding a Facial recognition sign in app. 36](#_Toc192271948)

[**Figure 3.2:** System Architecture for the student side of the application. 42](#_Toc192271949)

[**Figure 3.3:** System Architecture for the lecturer side of the application. 43](#_Toc192271950)

[**Figure 3.4:** Figma UI designs for student side of application 47](#_Toc192271951)

[**Figure 3.5:** Figma UI designs for lecturer side of application 48](#_Toc192271952)

[**Figure 3.6:** ERD Database design for the application 50](#_Toc192271953)

[**Figure 4.1:** Example of Jetpack Compose code from the student Hub Page 52](#_Toc192271954)

[**Figure 4.2:** Using the logcat to retrieve valuable testing information regarding facial ID 53](#_Toc192271955)

[**Figure 4.3:** Images of user faces being kept in Firebase Storage unit 54](#_Toc192271956)

[**Figure 4.4:** Figma Designs for StudentSignIn application 55](#_Toc192271957)

[**Figure 4.5:** compareFaces function used in the student facial recognition page 56](#_Toc192271958)

[**Figure 4.6:** Firebase rules setup 61](#_Toc192271959)

[**Figure 4.7:** Querying the “users” collection to check the role of the current user 63](#_Toc192271960)

[**Figure 4.8:** Student timetable page 64](#_Toc192271961)

[**Figure 4.9:** Image of QR code page 65](#_Toc192271962)

[**Figure 4.10:** Indexes needed for complicated “attendance” collection query’s 67](#_Toc192271963)

[**Figure 4.11:** NavGraph using backStrackEntry to pass data through pages 68](#_Toc192271964)

[**Figure 4.12:** Request to use the camera while using the app 69](#_Toc192271965)

[**Figure 5.1:** Testing access request (1) 77](#_Toc192271966)

[**Figure 5.2:** Testing access request (2) 77](#_Toc192271967)

[**Figure 5.3:** QR error message testing 78](#_Toc192271968)

[**Figure 5.4:** Sign in error message testing 78](#_Toc192271969)

[**Figure 5.5:** Testing student timetable 84](#_Toc192271970)

[**Figure 5.6:** Testing lecturer timetable 84](#_Toc192271971)

[**Figure 5.7:** Testing facial recognition 86](#_Toc192271972)

[**Figure 5.8:** Testing if signed in 86](#_Toc192271973)

[**Figure 5.9:** Testing attendance viewing 87](#_Toc192271974)

[**Figure 5.10:** Testing attendance statistics 87](#_Toc192271975)

# Tables

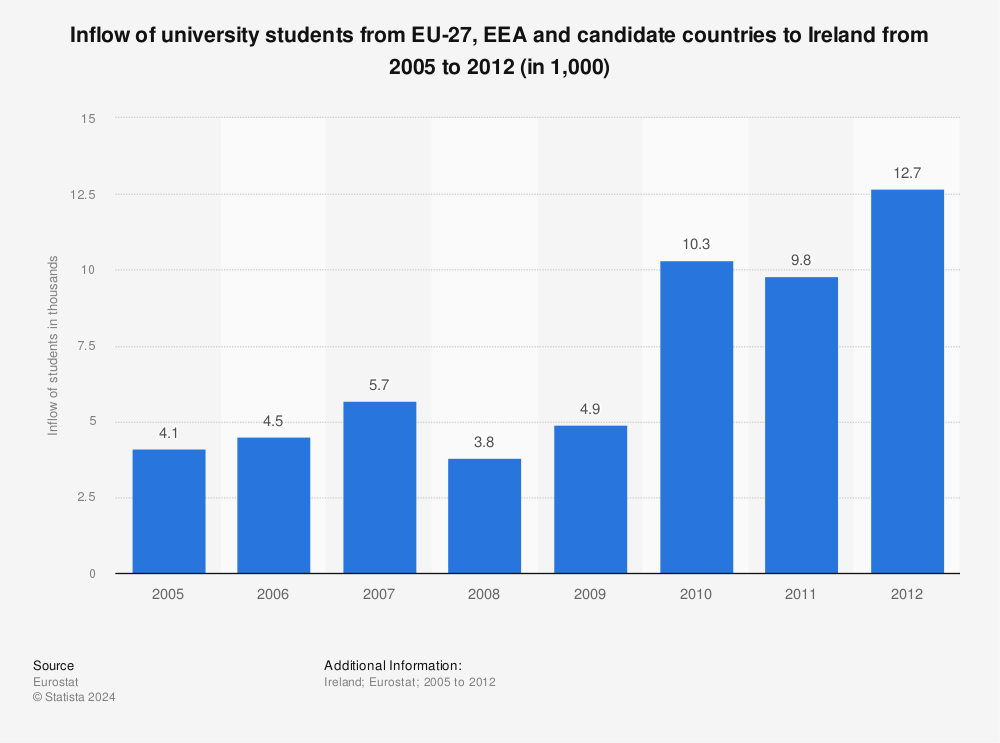
[**Table 2.1:** Authentication methods advantages and disadvantages (Faúndez-Zanuy, 2006) 17](#_Toc192272510)

[**Table 2.2:** Cloud based recognition compared to other methods (Zhang, et al., 2019). 20](#_Toc192272511)

[**Table 2.3:** Diverse ways to use QR codes within the classroom (Bala, et al., 2023) 32](#_Toc192272512)

[**Table 3.1:** Existing Attendance systems analysis 36](#_Toc192272513)

# **Introduction**

Attendance tracking is an important aspect in any institution (Zorić, et al., 2019). Traditional methods of attendance, including paper-based attendance recording, have become outdated alongside the modern digital systems we currently have (Chen & Li, 2021). These traditional methods are more limited, especially with the current rise in the number of students enrolled in colleges. **Figure 1.1** displays this trend in the number of students coming into Ireland from 2005 to 2012; there is a clear overall increase. With this growth, it becomes clear that the inadequacies of the current attendance practices are exposed, and more effective solutions become genuinely needed for the student population.  


**Figure 1.1:** inflow of university students from European countries (Statista, 2015)

## **Research**

### **Purpose**

The main purpose of this project is to develop a mobile attendance application with facial recognition and QR code technology to automate the process of taking attendance and make it more automated than manual attendance.

### **Objectives**

* **Friendly interface:**   
  To offer an easy-to-use Android app that will make the attendance process easier for students and lecturers.
* **Biometric and QR Code-Based Verification:**   
  To allow students a secured way for attendance verification via the use of QR code scanning with facial recognition.
* **Automate attendance records:**   
  To update the attendance in real time and synchronize in a protected database to ensure safe data storage and quick retrieval.
* **Enhance attendance analytics:**   
  So, lecturers can view statistics of attendance, which will help provide insights on students' engagement and participation.

### **Questions**

1. How efficient would a system with the combination of face recognition and QR codes be to automate attendance in the classroom?
2. What are the major challenges in implementing biometric verification in a mobile attendance application?
3. How is automated attendance tracking supposed to save time and enhance accuracy compared to traditional methods?
   1. **Proposed Solution**

This project will present an application that can easily integrate facial recognition with QR code functionality, allowing Android devices to connect with the centralized database for attendance recording. The application will develop a facial recognition-based automated attendance mechanism, which is in place of traditional methods and to provide a secure and reliable way of monitoring attendance among students. By doing so, it will replace the old traditional way of attendance with a more user-friendly mechanism, decreasing the workload of administrators and improving data accuracy.

In the end increased inflow into the higher education ecosystem has become a call for increased efficiency and reliability in attendance-tracking methods. Traditional paper-based attendance systems have been unable to meet the demand placed on modern educational environments because they are normally cumbersome, time-consuming, and error-prone. This project addresses these issues by developing a mobile attendance application with the features discussed previously, which will provide innovative solutions for automating and enhancing attendance tracking.

It is against this background that we are going to proceed and look at these attendance systems currently in operation, together with the pros and cons, and how technology is changing the face of effective attendance management. This will set the proper context on how the proposed application can help attendance systems in educational institutions so that the quality of learning for students in these institutions can be improved.

# **Literature Review**

## **Attendance Systems**

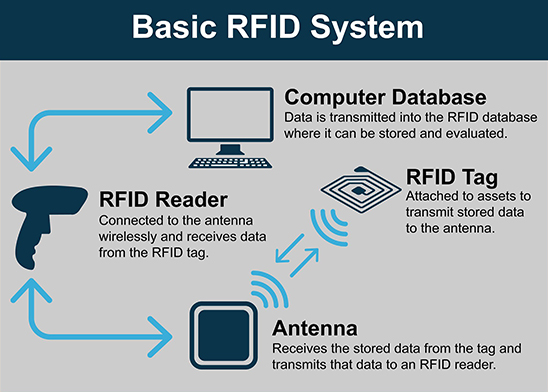
The chapter gives an overview of the various methods adopted for attendance tracking and further classifies these into non-technical and technical approaches. It examines the pros and cons of each, their relative efficiency in different environments, and applications across all sectors, including education. From this, it brings to light the inadequacies in traditional methods and how biometric and cloud-based systems are adapting to fill these gaps.

### **Attendance Taking Systems**

In this fast-growing world of technology, attendance and time management software plays an especially key role in monitoring our attendance tracking. Most of the attendance software includes multiple options for clocking in and out, from biometric terminals with touchless facial and fingerprint recognition to mobile app-based systems with remote clocking capabilities (Rucevic, 2023 ). This system, therefore, not only reduces time fraud but also assures a better way of adhering to labour laws, diminishes manual processing, and cuts administrative costs. Moreover, most of these systems also integrate with payroll software that guarantees accurate and efficient payroll processing directly through an import of attendance data. Technological change in that respect only makes the attendance systems more efficient, accessible, and adaptive to the needs of organizations of different size and requirements (Rucevic, 2023 ). Industries moving toward those solutions, sectors such as education and health care would be able to make use of increased accuracy, productivity, and flexibility in handling attendance.

### **Types of Attendance Taking Systems**

The diversity in attendance systems shows how different organizations are in the need for accuracy, efficiency, and data security in attendance tracking. The system ranges from manual to advanced technology-based solutions and hence helps organizations streamline attendance management and ensure labour regulatory compliance (Qureshi, 2024). Radio Frequency Identification (RFID) systems use cards or tags embedded with microchips. RFID tags are useful in deriving real-time attendance records without direct contact in high-traffic areas when scanned by the readers installed at the entry points. This has streamlined the collection of attendance data, and RFID systems allow the administrators to take online monitoring of the attendance records, hence proving to be efficient and hygienic (Qureshi, 2024). As shown in **Figure 2.1** the antenna receives the data from the tag transporting it to the RFID reader which then checks the database to confirm without the use of human touch. Challenges with RFID systems include the probabilities of misplacement of RFID cards, as well as the capital expenditures associated with the purchase of equipment such as RFID readers and tags. These make RFID an unattractive solution for organizations with strained budgets.



**Figure 2.1:** A basic RFID System (BRADY, n.d.)

Web-based attendance tracking systems also became extremely popular, as these could automate and hence simplify the procedure of keeping track of attendance. The IoT relates to all sorts of devices capable of sharing data in real-time and enormously enhancing decision-making in a very wide range of fields (Yadav & Bhole, 2019). The attendance systems, therefore, integrate cloud computing with for instance, software as a service (SaaS) so that the resources are managed efficiently and access to attendance records is made available promptly. This not only boosts efficiency but also overcomes the challenges that relate to the conventional methods of attendance, including impersonation and misplacement (Yadav & Bhole, 2019). However, technical glitches, such as problems related to the internet or software compatibility issues, may themselves become barriers to the online system and limit its adoption in organizations with a poor digital infrastructure.

The use of GPS technology can, therefore, provide real-time location information that confirms students in specific pre-defined areas among novel ways of taking attendance. For example, a study by (Zhao & Huang, 2020) states that the GPS attendance system gives teachers the possibility to take attendance by checking whether students are inside certain geographic boundaries during class hours. This will bring the correctness in attendance reports and close fraudulent reporting, usually done by students who mark attendance for fellow students not in class. In relation to this, by employing GPS technology integrated into the solution, the institutions can produce a flawless, more efficient, and reliable attendance management system. This eases administrative processes and improves general accountability (Nayak, et al., 2022).

### **Advantages** **of** **Attendance** **Taking** **Systems**

With their ability to bring advantages that translate far beyond improved operational efficiency and increased employee accountability, attendance-tracking systems have gained prominence in organizations. Among the most prominent benefits of using these attendance trackers would be increased accuracy and efficiency. Automation of attendance systems significantly reduces errors developed by traditional manual attendance-tracking methods. Such a feature shall allow real-time data capture and reduce administrative burdens (Nayak, et al., 2022). With such a seamless process, there is the assurance of getting accurate attendance records ready for analysis. Besides this, there is the added advantage of high security in relation to attendance systems. Most of these systems come with sound security features meant to prevent unauthorized entry and fraudulent attendance. Features such as facial identification and other biometric verification techniques, through which attendance can be recorded can be ensured to reflect only those present, according to (Nayak, et al., 2022). The level of security associated with users is very vital to ensure that the data attendance is not compromised.

The other important attendance system feature is that it creates a culture of accountability in the workforce. From this data, managers can produce performance improvement plans and put strategies in place for the future (Petrovic, 2024). Greater accountability may lead to increased productivity and a more engaged workforce. The major concerns running through present-day attendance systems are ones of hygiene and privacy. Creating an attendance software that give much importance to the privacy of employees are quite common today, thereby helping in reducing questions regarding data security (Nayak, et al., 2022). Moreover, contactless attendance will avoid physical contact, which becomes especially relevant in settings where hygiene control is a requirement.

### **Disadvantages of Attendance Taking Systems**

Though attendance systems have many advantages, they are not without their drawbacks. One major concern is that biometrics do reduce manual effort, but the time consumed is not improved and is sometimes increased, and the biometrics are not cheap in cost (Kodali & Hemadri, 2021).

Furthermore, there might be unforeseen circumstances arising with attendance taking systems. For instance, during COVID-19, when everyone started wearing face masks, there was a problem in the field of biometrics and attendance taking for facial identification (Liliana & Liliana, 2021).

There are, in addition, concerns regarding privacy, particularly through the collection of biometric data. Workers may feel uncomfortable knowing that their personal information is stored and watched. Such anxiety is aggravated by the complexity surrounding ensuring LED compliance (European Data Protection Board, 2023).

Lastly, organizations can also become so dependent on technology, and when the systems fail, great losses in data and accuracy are experienced. Another challenge is resistance to new technologies where employees may rebuff new systems that are alien to them and invasive; this may cause the implementation process to fail. Most of the time, the systems are designed considering the administration first. It is important, however, to consider the well-being of employees too (Petrovic, 2024).

### **Applications** **in** **Attendance** **Management**

Attendance management has improved with the adoption of biometric systems, facial recognition technology, due to their ability to offer high security and accuracy over the traditional manual methods. One of the key characteristics is non-intrusiveness, facial recognition assures the comparison between live images and stored data for the maintenance of attendance without making physical contact (Sajid, et al., 2014).

### **Traditional** **Identity** **Verification** **Methods**

Traditional authentication techniques, which are used to verify the identity of an individual, can be classified into two broad classes. One class, which uses physical objects for identification, includes Identity-Based Markers such as keys or certificates (Meng, 2008). Unfortunately, these techniques sometimes result in identification failures. Another class, Knowledge-Based Authentication, depends on information the users must provide to authenticate their identity (Li & Zhang, 2010). This information can come in the form of passwords or knowledge questions. Although this has been a quite common authentication method through the early years of computer networking, several flaws exist. As seen in **Table 2.1** with this kind of authentication, most notably the possibility that users might forget the information or the fact it can be guessed or cracked.

A white paper with black text

Description automatically generated

**Table 2.1:** Authentication methods advantages and disadvantages (Faúndez-Zanuy, 2006)

### **Emergence** **of** **Biometric** **Technology**

With all the drawbacks of the traditional authentication methods, biometric technology presents a more secured and effective alternative. Biometric technology is based upon the physiological or behavioural characteristics unique to individuals for the purpose of identification and verification of identities (Li & Zhang, 2010). These methods can be specifically divided into two main classes which are: Physical Characteristics, which are Physiological and Behavioural as shown in **Figure 2.2** where we can see examples of both kinds.

Physiological characteristics refer to biometric traits, which are fingerprints, face recognition, iris patterns, hand geometry, and DNA that an individual possesses in unique and non-replicable form (Li & Zhang, 2010). Most used in attendance systems, this type of physical characteristic allows for identification with a high degree of accuracy due to the matching of those distinct features against stored data.

Another category is that of the behavioural characteristics, which includes voice pitch, speaking style, typing rhythm, and signature. All these are different individual behavioural patterns, in use in systems requiring an extra level of security. The behavioural characteristics can add flexibility to the attendance systems where they allow identifying methods without contact (voice pitch and speaking style), which turns out to be vital in some environments. Such features can be continuously recorded in a non-intrusive manner using embedded sensors in smartphones, such as gyroscopes and accelerometers, thereby improving both usability and security at low cost (Papaioannou, et al., 2023).

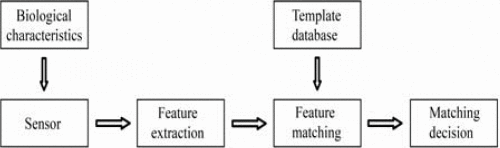
A diagram of a biometric characteristics

Description automatically generated

**Figure 2.2:** Difference between Biometric Characteristics (Ashtopus Technologies, 2015)

### **Main** **Biometrics** **Used**

Biometric identification technology applies in many fields, including criminal investigation, information communication, online shopping, online banking, medical insurance, and e-commerce (Li, 2015). **Figure 2.3** provides an example of how the biometric technology functions. As we can see the biological characteristic is picked up via a sensor and compared with a database to create a matching decision. Out of all the biometric technologies, fingerprint identification is among those that have been implemented widely. The uniqueness in the pattern of an individual's fingerprint makes possible its storage and retrieval from a database for identification purposes. However, though the fingerprint is the leading biometric, there is an increase in the use of multi-biometric identification technologies (Meng, 2008).



**Figure 2.3:** The basic structure of biometric identification technology (Li, 2015)

### **Advantages of Biometric Technology**

Compared to the traditional ways, there are a few advantages of biometric authentication systems. For example, unique identifiers like palm prints or fingerprints are not easy to be imitated, hence offer a much more secure environment for attendance (Paderes, 2015). Second, users do not need to remember passwords anymore as biological characteristics like the width of an eye and fingerprints remain almost the same for as long as decades, which significantly reduces the chances of unauthorized access.

### **Cloud based attendance systems and Biometric Integration**

There has been a trend in biometric authentication moving towards being more cloud based. In addition, cloud computing revolutionized the way organizations manage their data. It eases the storing and accessing of attendance records from a secure location.

Biometric technologies, integrated with cloud-based attendance systems, impart great benefits in data security, accessibility, and real-time attendance tracking. It may also help organizations to meet the requirements of regulations with precise and verifiable data of attendance (Qureshi, 2024). As we can see in **Table 2.2** the cloud based facial recognition has the highest detection and recognition rate compared to other facial recognition methods. This further proves that as we progress within biometric technology cloud-based operations will be used as the staple.

However, there are potential challenges with such systems; for example, most of the biometric information storage raises privacy concerns. Organizations need to handle such issues while putting in place a cloud-based biometric attendance system.

A table with numbers and text

Description automatically generated

**Table 2.2:** Cloud based recognition compared to other methods (Zhang, et al., 2019).

### **Conclusion**

In conclusion, traditional verification methods can be categorized into non-technical and technical approaches. Traditional methods of identity verification, such as knowledge-based authentication and identity-based markers, have shown high weaknesses in security and user experience. Biometric technology is a potential way to raise accuracy and efficiency in attendance management with its physiological and behaviour attributes unique to individuals.

As discussed, each of the biometric methods—be it fingerprint recognition, iris pattern recognition, or facial recognition—offers deep advantages over the traditional system. Technologies like this one elevate the bar of security and simplify the full process of taking attendance yet have their own sets of drawbacks, which include issues regarding privacy and complicated equipment requirements. Though some of the disadvantages have been shown, the use of a biometric system in attendance management revolutionized the attendance of students with an extremely secure alternative to the conventional old system of manual attendance taking.

We have also seen the current trend of using cloud-based systems integrated with biometric authentication to make attendance solutions even more functional and accessible. Of late, there has been a tremendous increase in facial recognition technologies being use, mainly because the process is non-intrusive to the user and easy to carry out; hence, their high adoption rate in modern-day attendance systems.

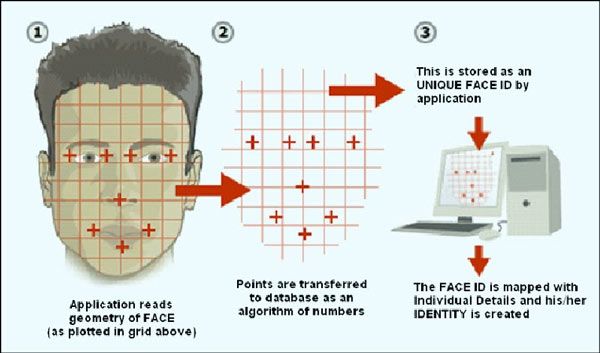
In the next section, we will take a closer look at the facial recognition technology itself: the principles underlying it, how it works, and the role of changing attendance systems. The following section will help us understand the inner mechanism of facial recognition and how this, through attendance taking, can aptly meet challenges to bring a transformation in educational results overall.

## **Facial Recognition**

Facial recognition technology has been exceedingly popular over the last couple of years and has brought a revolution in identification, precisely in attendance management and security at large. This section is going to offer an in-depth analysis regarding facial recognition systems, their underlying techniques, applications, advantages, and weaknesses. It will also address the privacy issues and accuracy of such systems, in consideration of the challenges they must meet, including countermeasures in anti-facial recognition masks.

### **What is a facial recognition system ?**

Face recognition systems are works of advanced biometric technologies using algorithms to identify and authenticate individuals by their facial features (Li, et al., 2020). Normally, the way the software works is that it would scan videos and photos to create a map of a person's facial features, normally known as a facial signature. This includes things such as the exact location of their eyes, scars, or any other facial differences (Stouffer, 2023). Generally, it starts with a camera taking a picture of a face; it then goes through an algorithm that isolates key facial features and translates these into some form of mathematical representation or a "faceprint" (Li, et al., 2020). This can be seen in **Figure 2.4** where once the image is taken points of facial landmarks are transferred to the database as algorithmic numbers and stored. The faceprint is then compared against a database of known faces to find a match. If the system finds a match with a certain amount of confidence, then it confirms the identity of the person. The accuracy in facial recognition lies in several factors like good lighting conditions, the appropriate angle of the face, and good image quality (Li, et al., 2020).

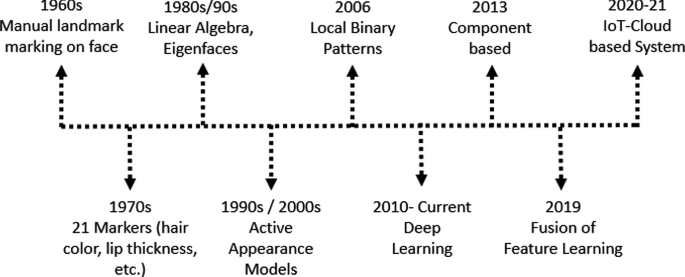


**Figure 2.4:** The geometrical approach to facial recognition (Dangi, 2011)

### **Evolution of facial recognition technology**

Facial recognition technology has gone through dramatic changes since its inception. In its early stages in the 1960’s as shown by **Figure 2.5**, the systems had been so dependent on manual human recognition, whereby operators would compare facial features in a bid to identify individuals (Gupta, et al., 2023). While this way brought in some level of success, it was time-consuming and error-prone owing that to human error.

Coming a long way from the early beginnings, fully automated facial recognition systems started taking their current form. The key turning point was the development of algorithms in the 1990s that enabled the identification of facial features using digital images (Gupta, et al., 2023). The systems then used these geometric metrics to measure the distances between the key facial landmarks, tending towards a much more standardized approach toward identification. The development of machine learning and artificial intelligence in the 2000s transformed the field of facial recognition. The objective of machine learning is to acquire new knowledge and abilities while systematically organizing a body of knowledge that can support the continual improvement of performance (Xue & Zhu, 2009). This capability has seen considerable improvement with the vast amount of deep learning techniques, which have enabled systems to be trained on a continuous basis on ever larger datasets, incrementally improving their performance. As such, they have continued to remain unmatched in face recognition since 2014 (R, et al., 2023).



**Figure 2.5:** The evolution of facial recognition technology (Pattnaik, et al., 2023)

In recent years, facial identification has found applications in various sectors, including security, law enforcement, and attendance (Li, 2015). With the increasing number of smartphones that have facial recognition, this development has become so common and an aspect of modern life that it is more accepted by society. At the same time, it arouses significant ethical dilemmas regarding human rights and misuse (Banzon, et al., 2024), This shows why responsible application and governance are needed; a concern reflected in the ever-growing number of publications on facial recognition that arise every year as shown below in **Figure 2.6** where we can see in 2020 peaking at nearly 900 facial identification documents that year..

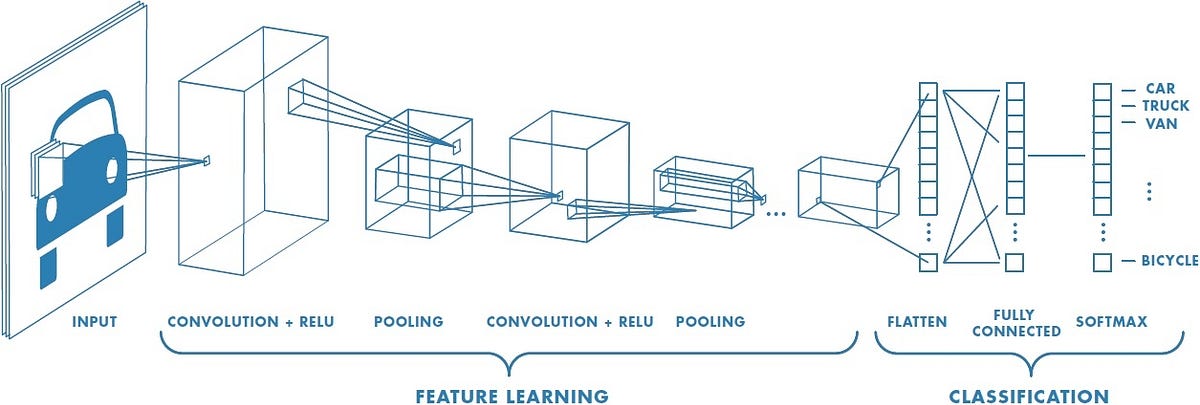
A graph of a number of documents

Description automatically generated

**Figure 2.6:** Number of Facial Recognition documents by year (Zennayi, et al., 2022)

### **Techniques for facial recognition**

Face identification techniques offer different procedures to get efficient and effective recognition. Initially, detection algorithms find faces in an image, the standard is applying the Haar cascades with AdaBoost classifiers because it allows fast detection—particularly when combined with integral images to speed up calculations (Sahu & Dash, 2020). The next step is feature extraction, whereby deep learning identifies those unique facial markers that remain unchanged in different conditions, such as the eye's position, jaw line, and nose shape (Sahu & Dash, 2020). Dimensionality reduction on faces is done using eigenfaces and other techniques by applying principal component analysis, which optimizes data representation across classes (Zhang, et al., 2014). Skin colour detection methods can be done by using models like RGB, HSV, and YCbCr colour models to separate facial regions from the rest of the environmental noise. Then, face alignment standardizes those features by accounting for variance in angle, lighting, and expression. CNNs (Convolutional Neural Networks) with deep metric learning finally have a strong matching aspect to where unique facial embeddings are encoded and then can be compared for recognition or verification. As shown in **Figure 2.7**, we see the image of the car going through multiple convolution layers as the system processes the image as an array of pixels to categorize the image. In conjunction with those methods, the facial recognition systems are robust to various challenges like pose variations, occlusions, and ambient lighting conditions.



**Figure 2.7:** Neural Network with multiple convolution layers (Raghav, 2018)

### **Advantages of facial recognition systems**

Facial recognition has been one of the important techniques in securing smartphones, given the benefits to improve the user experience and security of the device. Among those advantages, automating the process of attendance which through facial recognition happens seamlessly. Providing touchless sign-ins through facial recognition systems limits physical contact and decreases the chance of spreading viruses across working places (Liliana & Liliana, 2021). This was a major advantage during COVID-19. On all counts, the integration of facial recognition technology into smartphones not only brings enhanced security but also falls in with the ever-growing demand for efficient and user-friendly access control solutions (Jha, et al., 2022).

### **Weakness** **of** **facial** **recognition** **systems**

Despite their popularity and the promise of bringing convenience and security, facial recognition carries a few major flaws. Perhaps one of the most important is the potential for bias with respect to different demographics as evidenced by European laws to do with the use of facial identification in finding criminals within law enforcement (European Data Protection Board, 2023). Facial recognition algorithms have been found to perform poorly, particularly for minority groups, submitting a higher rate of false positives or misidentifications (Istvan, 2020). The need for extensive databases of facial images gives rise to significant concerns about privacy, as companies such as Clearview AI has amassed large repositories from publicly available sources, such as social media. This misuse by governmental and commercial entities only heightens concerns of overreach, evidenced by some countries using facial recognition to monitor and control populations (Istvan, 2020). All these weaknesses show there is a need for strict standards and ethical frameworks in the application of facial recognition technologies in ways that serve public safety without hurt to individual rights.

### **Privacy Considerations and issues**

The facial attendance systems are the epitome of privacy considerations, especially because of the sensitivity associated with biometric data (Bowyer, 2004). The collection and storage of facial data raise important ethical and privacy issues since biometric data can identify a particular person. Informed consent has become more focused on recent ethical implications. This means that users must be informed how their data is processed, who has access to the data, and how long the data will be stored. However, alleviation of these fears through encryption of data, controls on access, anonymization when possible, and just clear policies will ensure user trusts and protection (Bowyer, 2004).

### **Facial** **recognition** **within** **attendance** **systems**

Facial recognition systems have developed into a very effective and authentic way of biometric identification, particularly in education, where it is always challenging to take attendance in huge classes (Wang, 2021). Traditional manual methods of taking attendance include calling out names or keeping a paper record, which are indeed time-consuming and usually subject to mistakes, especially in classes with more than 100 students (Wagh, et al., 2015). To tackle these issues, several automated systems have been put forward, such as fingerprint scanning, iris recognition, and RFID-based technologies. However, these options often suffer from the drawbacks of cost, accessibility, and portability. For example, fingerprint and iris scanning systems demand individuals' direct interaction with devices, hence leading to delays, on the other hand, RFID systems are vulnerable to inaccuracy in attendance due to card interchanging (Sultan, et al., 2022). In sharp contrast, facial recognition technique mitigates these threats as every individual has a distinctive facial attribute, hence minimizing the chances of impersonation (Sultan, et al., 2022). This not only saves time but also enhances the credibility of data, hence overcoming traditional problems of record loss and manipulations of attendance in an academic environment.

### **Accuracy** **and** **reliability**

In this regard, facial recognition technology has improved significantly, attaining high accuracy and robustness, much of which is attributed to Universal Representation Learning and sophisticated detection schemes such as HAMBox. Universal Representation Learning improves the generalizability of face recognition algorithms by simulating variations of different kinds like low resolution and occlusion thereby improving performance on datasets such as LFW and MegaFace (Wang, 2021). HAMBox tackles the problem of misalignment in face detection by improving anchor matching strategies, which improves the accuracy of such a detection process. Such fine-tuning achieves an Average Precision of 93.3% and we can see in **Figure 2.8** that HAMBox is outperforming other anchor matching strategies by a good percentage. on the challenging WIDERFACE dataset (Wang, 2021). Such improvement ensures that face detection and recognition tasks remain reliable for real-world applications. In this regard, these technologies are on their way to becoming increasingly adaptive and reliable when used in complex environments.

A graph of a graph with a graph and a chart of a graph

Description automatically generated with medium confidence

**Figure 2.8:** HAMBox compared to other Anchor matching strategies (Wang, 2021)

### **Anti-Facial recognition masks**

With the rise in facial recognition technology, there have been masks created for anti-facial recognition and concerns about privacy and security.

An experiment was conducted to test how masks affect the ability to detect a face. Having the three training datasets, Training\_UM (unmasked images), Training\_M (synthesized masked images), and Training\_HM (mixed images). One study by (Pudyel & Atay, 2023) used feature extraction in six machine learning algorithms to see how these datasets worked with facial recognition. Models trained with unmasked images were poorly generalizing to the masked faces, with a failure to recognize as high as 45%. Models trained with masked images had a lower mean miss rate in the masked data sets when tested (37%).

Logistic Regression outperformed the other algorithms, achieving 80% in terms of accuracy when trained with half-masked images, while the K-Nearest Neighbours algorithm produced the poorest performance at 24% (Pudyel & Atay, 2023). These results bring forth the limitations of conventional facial recognition systems against methods like masks.

Results show that anti-facial recognition masks could challenge the effectiveness of facial recognition technologies, yet training with different datasets can improve the robustness of the system. The next step in research should go ahead with more advanced deep learning algorithms and larger, diverse facial image databases to cope with the complexities brought forth by such countermeasures.

### **Conclusion**

In other words, facial recognition technology has been one of the most imperative tools in transforming various fields with high efficiency and accuracy rates compared to traditional methods, particularly in attendance management. The evolution has been tremendous, from early systems of manual recognition to the current highly sophisticated automated systems, driven by both machine learning and deep learning in enhancing the reliability and effectiveness. However, all the benefits notwithstanding, facial recognition systems remain vulnerable to weaknesses regarding susceptibility to privacy issues, ethical questions, and abuse.

With the controversies and complexities surrounding facial recognition, especially in sensitive environments like education, we need to look at regulatory frameworks that ensure individual rights are protected. That would bring into play the General Data Protection Regulation, or GDPR and EU regulation on privacy and data protection. The next chapter details the implications that GDPR has on facial recognition technologies with regards to how such regulation reshapes the legal framework concerning biometric systems and the duties of organizations using such technologies. All these regulatory considerations should be considered to ensure ethical practices in the facial recognition system used in attendance and beyond.

## **GDPR Guidelines**

There is a need to develop facial recognition technology within the boundaries set by the Council of Europe and the GDPR, especially attendance systems. Among those regulations, central principles are accuracy in data and reliability in algorithms, along with necessary measures to protect fundamental rights of people using facial recognition technologies.

It would then be the developers' and manufacturers' responsibility to ensure that the respective biometric information used in training algorithms is well-represented and trustworthy. Most specifically, under Article 5 of the Convention 108+, dealing with the issue of accuracy, it talks about confronting biases relating to skin colour, age, and gender (Europe, 2023). This follows from the general rule of not bringing in accidental bias, so the technology of facial recognition must be fair with respect to gender, race, ethnic origin, religion, or political affiliation (European Data Protection Board, 2023). Thus, the developers of attendance-taking applications using facial recognition must carry out extensive tests using a very wide variety of datasets.

Furthermore, the updating of facial data on a regular basis will update the accuracy of the system. Changes in personal appearance, such as facial accessories or health conditions, might cause large variations in the reliability of recognition (European Data Protection Board, 2023). The developers need to test the performance of their algorithms in a periodical manner and perform updating with new training data so that the capability of recognition does not degrade over time. Transparency of the deployed technology means visibility by the end-user or institution, through dashboards showing system reliability metrics. The transparency will provide accountability and give more faith to the facial recognition attendance systems.

Simultaneously, the GDPR-based data protection principles include purpose limitation, data minimization, and a clearly defined period for data retention. This means that from a legal viewpoint, such principles should be built in the technical design of facial recognition systems (European Data Protection Board, 2023). In other words, if they ever hope to be compliant, such a system will also have to practically integrate considerations for privacy during development. It may also comprise things like automatic deletion of raw facial data once the necessary biometric templates are extracted and carrying out Data Protection Impact Assessments, which help in identifying and mitigating privacy risks throughout the development process. Measures like these ensure that facial recognition is done correctly and with respect for the rights of individual privacy.

The steps will go a long way in ensuring responsible use of facial recognition in attendance systems and strictly ensure, at the end, that user data security is assured and dependable with no discriminating results.

In general, facial recognition attendance systems are to be fully implemented following the GDPR rules and ethical norms that can protect the rights of the individual and make the systems fair. They also make facial recognition systems much more dependable by focusing on accuracy with lower bias and frequent updates. Transparency and accountability are also particularly important in public trust building and compliance with legal requirements.

## **QR** **Codes**

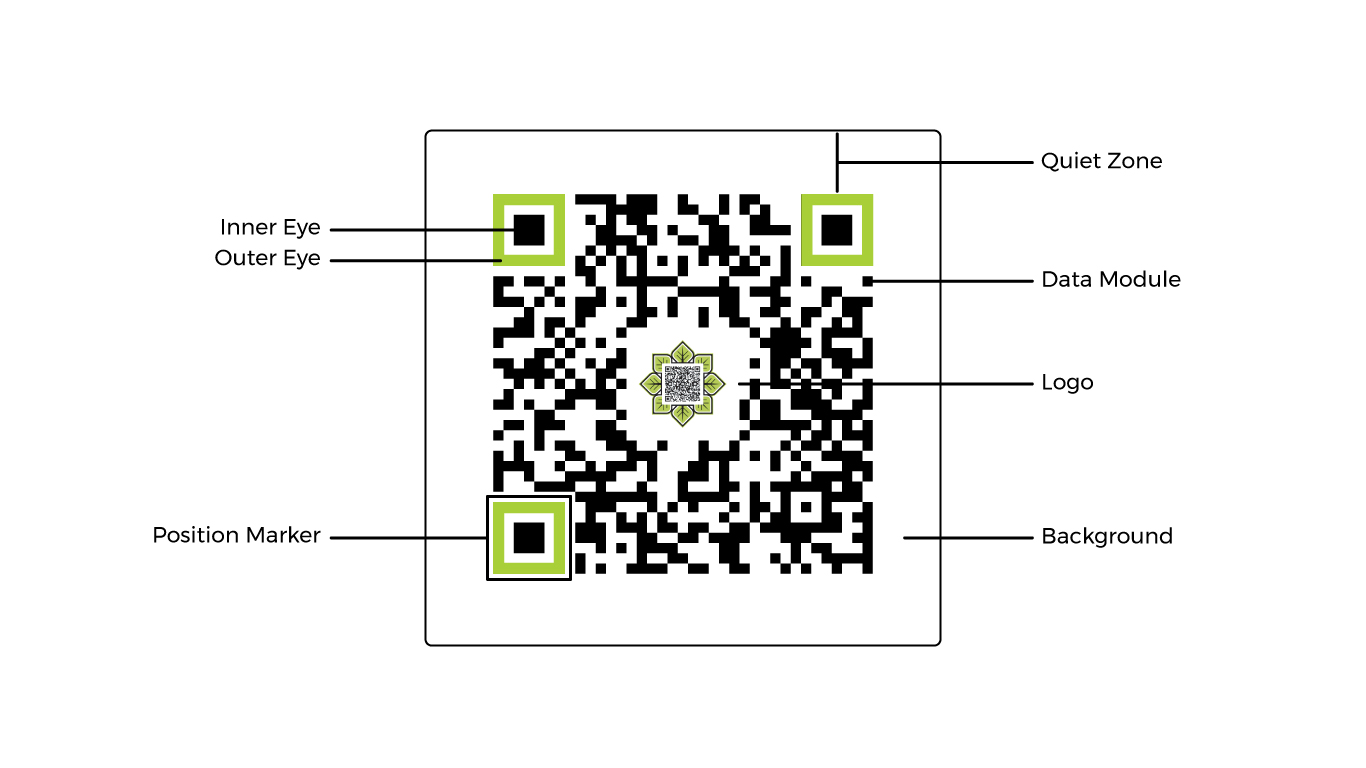
In many fields, especially education, QR codes have featured because of their capability in storing information and rendering easy access to it. These two-dimensional barcodes were invented by DENSO Wave back in 1994 and render fast data retrieval. This will highlight some of the basic features of QR codes regarding design and applications, including specific use cases within the educational landscape.

### **What** **are** **QR** **codes?**

QR codes are two-dimensional barcodes, QR codes are matrices that store information. Invented by a Japanese company called Denso Wave with the purpose of tracking automotive parts in manufacturing (WAVE, n.d.), QR codes have since been used for a variety of applications and purposes across different fields. More precisely, QR codes, unlike traditional barcodes, which can store only a small amount of information, QR codes can store a lot more such as alphanumeric characters, binary data, and special characters (Tiwari, 2016). The increased storage capacity means QR codes could store URLs, contact details, texts, and many other forms of data, hence becoming very versatile tools in communication and sharing information in our digital era.

### **Design of** **QR** **codes**

In their design, QR codes are square in shape, containing a grid-like pattern composed of black squares on a white background. The position markers and the alignment patterns, timing patterns, and data modules are built into the design of a QR code as shown in **Figure 2.9**. The three big squares at the corners of the QR code are position markers, which enable a scanner to rapidly determine the orientation of the code (Tiwari, 2016). Alignment patterns ensure the code can be read to an exact degree even if it is distorted or printed on some curved surface. Further, QR codes also include error correction capabilities, enabling them to remain functional even when partly damaged (Tiwari, 2016). Some of the design elements, in combination, allowed the QR codes to be scanned at any angle and from various distances using the inbuilt reader, hence making them very user-friendly and efficient for quite a few applications.



**Figure 2.9:** How a QR is structured (Schulfer, 2020)

### **Uses of QR codes**

QR codes have become used in many areas since they are able to deliver information easily and rapidly. In the area of marketing, companies utilize QR codes in enabling consumers to easily access promotional sites, information about products and social media to create more interaction with customers to increase sales (Tiwari, 2016). For example, in the health sector, QR codes keep patients' records, medication tracking, and laboratory results whereby health professionals can retrieve information quickly and precisely (Tiwari, 2016). QR codes have been implemented in the education sector through facilitating attendance keeping, providing access to digital learning materials, and offering engaging activities for students (Tretinjak, 2015). The area of contactless payment systems also saw them thrive in use—whereby a customer could easily make transactions through their smartphones. Applications using QR codes have continued to diversify and offer cutting-edge solutions in different sectors with technological advancement.

### **QR Code technology within education**

Due to their high data capacity, easiness in use, and versatility, QR codes have been universally used in education in the recent years. They can be used in both individual and collaborative learning processes, hence offering an interactive environment for the students to learn.

During this study, twenty-three students of vocational education participated in activities regarding QR code at the School of Electrical Engineering in Zagreb. In the commentaries about the activities, ninety-two percent of pupils declared that the activities were remarkably interesting and useful in realizing the learning outcomes (Tretinjak, 2015). Moreover, ninety-eight percent of them said that the interaction with the content was easy once the proper software was installed on the devices. This research shows that QR codes can be placed in almost every class material, be it textbooks, handouts, course website, or even class walls, to help learning (Tretinjak, 2015).

This has provided learning environments with additional resources as shown in **Table 2.3** with all the potential uses. This creates more student engagement and allows for the support of more diverse teaching methods via QR code integration.

A screenshot of a paper

Description automatically generated

**Table 2.3:** Diverse ways to use QR codes within the classroom (Bala, et al., 2023)

### **Limitations of QR codes**

Though popular and versatile, there are some barriers to the wide spreading of QR codes. First, the smartphones that have no built-in QR scanner can prove costly and thus unreachable for a lot of potential users. Another major factor is that consumers have not been informed about QR technology and thus require educational marketing campaigns to encourage its use. Also, aesthetic issues related to the appearance of QR codes and the inflated cost of maintenance to keep the devices connected pose a barrier toward their wider application. These disadvantages can decrease the efficiency of QR codes in general, in attendance management and learning management systems (ByteScout, 2019).

### **Conclusion**

Finally, the fact is that QR codes have become a force of change, revolutionizing an educational setting by increasing accessibility and further exciting students. High storage capacity and ease of use make them particularly important in simplifying various parts of education, for example, the attendance system. However, challenges of accessibility, user familiarity, and aesthetic concerns need to be met before their widespread adoption.

This being the current form, the integration of QR codes and facial recognition into the attendance system does bring out the fact that each has their own advantages and limitations. This may result in a more efficient and reliable way of tracking attendance and balancing privacy and ethical issues.

## **Final Thoughts**

With increasing acceptance of advanced technologies in attendance management at educational institutions, issues of security, privacy, and user experience cannot be overemphasized. This introduces the use of biometric systems in the form of facial recognition and QR code technologies—both opportunities and challenges. The key features of modern attendance systems include security and privacy implications of biometric data, user experience in using these technologies, and emerging trends that are going to shape the future of attendance management. In this section, we look at some of the elements that can enable us to have a better view of creating an efficient and ethically sound attendance system that respects individual rights and increases operational effectiveness.

### **Security and Privacy in modern attendance systems**

Since the management of attendance is evolving with the help of innovative technologies, one of the most necessary discussions in this regard must be about the security and privacy of personal information. Regarding this aspect, leading-edge biometric technologies, including facial recognition, should be integrated into the system, while strictly adhering to the requirements of laws such as GDPR, so that no misuse of sensitive information occurs. Since it is the duty of any biometric system to uphold security, it becomes an ethical responsibility for developers and institutions to develop such systems using strong encryption, ensuring that limitations set for data retention are strictly observed and transparency in obtaining user consent. It would, therefore, assure public confidence in the systems and would assure no privacy violation through biometric system deployment.

### **User experience**

The modern attendance system is all about user experience. In the process of undertaking biometric techniques, users shall have the opportunity to interact with the attendance system in a seamless and touchless manner to minimize manual intervention while accelerating the pace of attendance. The QR code systems take the speed aspect to another level by increasing efficiency in scanning and digitizing codes. At the same time, the design of these two systems should ensure the seamless interaction. The development of user interfaces in ways that enhance acceptability, and efficiency will become of significant importance in attendance management as it becomes more automated.

### **Future trends within attendance systems**

The future attendance systems will continue to innovate with the integration of artificial intelligence and machine-learning methodologies. Better algorithms mean that facial recognition accuracy is high, unlocking predictive capabilities in analysis of attendance trends and unearthing valuable insights for educators. More than that, the in-depth exploration of blockchain technology to secure and validate attendance records offers an extremely promising avenue for creating a transparent record-keeping system. With those technologies in place, attendance management is going to be even more simplified and make systems adaptable, scalable, and effective in any educational institution around the world.

# **Analysis and Design**

## **Analysis**

### **Introduction**

This chapter explains in detail the analysis and design framework of the StudentSignIn application.

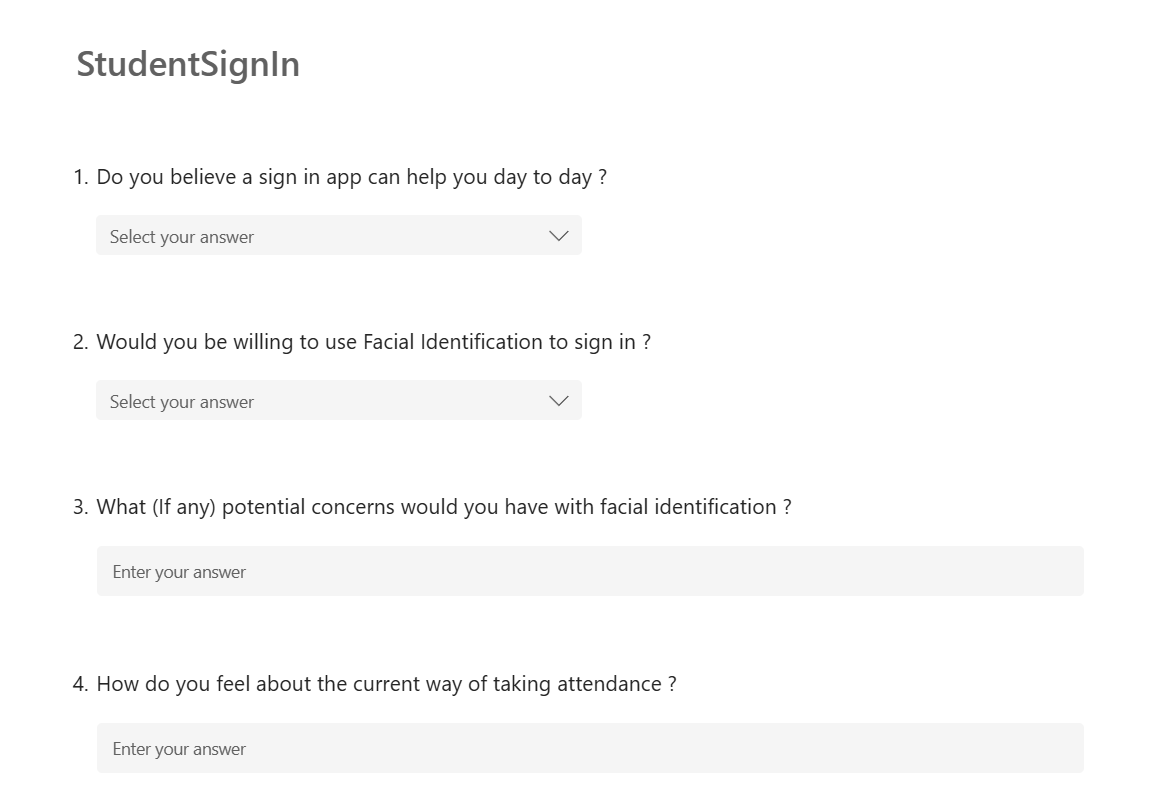
Analysis: Both functional and non-functional requirements of the system should be identified and described precisely to ensure that features impress the needs of the users. An elaborated process for the gathering of the requirements involves evaluating some of the existing solutions for attendance and researching what the users expect. This approach will ensure that the app shall amply address the loopholes in the present systems while being a practical and user-centred solution.

The chapter discusses, under the Design section, the architecture of the system, the user interface, and the structure of the database. It contains very detailed diagrams that illustrate the internal interaction of the app, such as the ERD and the system architecture visuals. These design decisions are motivated by their ability to support the app's goals of being reliable, efficient, and easy to use.

The chapter delves into the design and development of the StudentSignIn app, which would be able to meet its operational requirements through a strong analysis and design structure.

### **Requirements Gathering Process**

The tour of requirements regarding what to include in StudentSignIn, an application that should be reliable and efficient in tracking attendance, was indeed extensive. It needed to incorporate an investigation of already existing attendance systems, analysing the disadvantages that came with traditional sign-in methods, and an overview of modern technologies including biometric face recognition and QR code scanning. Furthermore, students and lecturers were to have questionnaires tailor-made to answer very particular questions about their perception of the goals of my application. The literature survey on automated attendance systems helped in learning about industry standards, the possibility of technology, and ethical concerns.



**Figure 3.1:** Survey for students regarding a Facial recognition sign in app.

### **Existing attendance systems**

The present systems were analysed in detail with the view to creating an efficient and competitive attendance-tracking solution. Below is a summary of the main attendance-taking systems currently in place, highlighting their functionalities:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| System Name | QR code scanning | Facial recognition | Real-time attendance | Role-based access | User-friendly interface | Data Security |
| Paper based sign in | No | No | No | No | No | Low |
| Mobile App (QR code) | Yes | No | Yes | Yes | Yes | Medium |
| Biometric attendance | No | Yes | Yes | Yes | Moderate | High |
| Student Sign In | Yes | Yes | Yes | Yes | High | High |
| RFID attendance | Yes | No | Yes | Yes | Moderate | Medium |

**Table 3.1:** Existing Attendance systems analysis

### **Key Insights from Analysis of Existing Systems**

Analysis of the currently existing attendance systems showed various avenues of improvement in bringing together user convenience and high-level security. Most of the current systems depend exclusively either on QR code scanning, facial recognition, or RFID technology alone, hence limiting their scope and flexibility. In this regard, StudentSignIn fills in that void with an all-inclusive approach: pairing QR code scanning for fast and easy entry with facial recognition for added security.

* **Biometric Systems:**Biometric systems assure high security and impede manipulation; however, these may be unusable in some cases due to poor lighting or technical failure. This concern had been addressed in the design of StudentSignIn by incorporating fallback mechanisms to ensure that the smooth functionality is preserved with no compromise on security.
* **QR-Based Systems:**Although quite user-friendly, QR systems are very prone to abuses such as screenshot scanning or code sharing. That is why the integration of face recognition with StudentSignIn provides an additional layer of security that makes the attendance neither fake nor manipulative. To do this a timer is set on the QR which will then generate a new QR code every 10 seconds. Stopping false positive sign ins.
* **Real-Time Attendance Tracking:**  
  Real-time tracking by lecturers will be dynamically possible, reducing administrative overhead and accuracy in records. This feature was prioritized to address the inefficiencies seen in systems that require manual updating of records.
* **Role-Based Access Control:**This will prevent unauthorized access and further enhance the integrity of data, because role-based access will ensure that no critical functionality, such as editing the attendance record, is performed except by an authorized user.
* **Enhanced Data Security:**Considering the sensitive character of students' and sessions' data, the application was developed under strong security protocols. Information is securely stored and transported with encryption, based on best practices in the protection of data and answering to legal standards such as GDPR.

Through these observations, StudentSignIn has been developed to improve upon where present systems fall short but still provide a well-performing, easy-to-use, secure solution for students and lecturers alike.

### **Functional Requirements**

The idea behind the design of the StudentSignIn system was to offer an efficient, safe, and easy way in which students and lecturers will manage attendance. Each feature has been carefully planned to solve a specific problem identified from the requirements gathering process.

* **User Authentication:**The system is accessible only to students and lecturers via valid institutional e-mail addresses. For example, specified domains such as "@student" or "@lecturer" ensure that any account creation on this is indeed controlled and restricted to genuine users. This provides a sound yet simple identity verification system targeted explicitly at educational establishments.
* **QR Code Scanning:**This allows the student to scan a different QR code to be generated by the lecturer for every session. As the system builds up, unauthorized attendance sign-in will be avoided, since it does not allow any screenshot or copied code. Linking the QR code to a session ensures that only eligible participants will be creating an attendance mark; hence, it reduces fraud while improving efficiency.
* **Facial Recognition Verification:**  
  Scanning of the QR verifies his or her identity through face recognition. The feature makes up for the deficiency in an independent QR system where only a registered student should have been allowed to log in to make the system more secure and reliable.
* **Role-Based Access:**Differentiates users' roles: a lecturer is granted the right to modify the attendance data, while students can only view their personal records. That way, it protects sensitive information and cuts off accidental changes made by unauthorized people.
* **Attendance Recording:**Attendance is recorded the instant that it is checked. Associating attendance with the course and session in real time will alleviate most needs to track manually, reduce administrative errors, and log student participation more accurately.
* **Timetable View for Students:**It will display a timetable for the student at an individualistic level-that is, showing only sessions relevant to his course. This will enable students to manage their schedules effectively, understanding when classes are due and hence keep them better organized.

### **Non-Functional Requirements**

The non-functional requirements are set so that the StudentSignIn system can work well and operate effectively, safely, and reliably. The subjects discussed will touch on performance, usability, security, reliability, and scalability, which will affirm and ensure the system meets both users' expectations and institutional standards.

* **Performance:**The application should load fast and respond immediately to the users' input without noticeable delays, even at those moments of high load when large numbers of students and lecturers may access the system. By efficient performance, it means smooth user experience, lower irritation, and it limits disruption to time-critical activities such as attendance tracking.
* **Usability:**It must be intuitively cognitive, with easy navigation for both students and lecturers. It must clearly provide in-app instructions that will lead a user through some tasks, such as the scanning of QR codes and face recognition. In such a way, it will make it possible for even less technical proficient users to use the app without increased chances of user error, hence increased adoption.
* **Security:**  
  The most concerning issue is sensitive data protection. To achieve user privacy, access shall be granted to the attendance record by the lecturer and the concerned officials only, through the planned design. Sensitive information, such as data on facial identification and the attendance record, is protected against unauthorized access and breach by means of data encryption and security protocols in communication, in accord with ethics and their legal implications.
* **Reliability:**Facial recognition and QR code scanning are core features of the system and must be super accurate to deliver valid but error-free records of attendance. It should be made sure that through the system, reliable real-time logging keeps the trust in it and avoids different problems, either wrong attendance records or non-signed sign-ins.
* **Scalability:**This means developing the application to handle increasing numbers of users without performance degradation. This kind of scalability would allow the system to respond to demands caused by either the growth of the adopting institution or by its adoption by other educational institutions. A scalable design future-proofs the system and supports its long-term viability.

## **Design**

### **Systems Architecture**

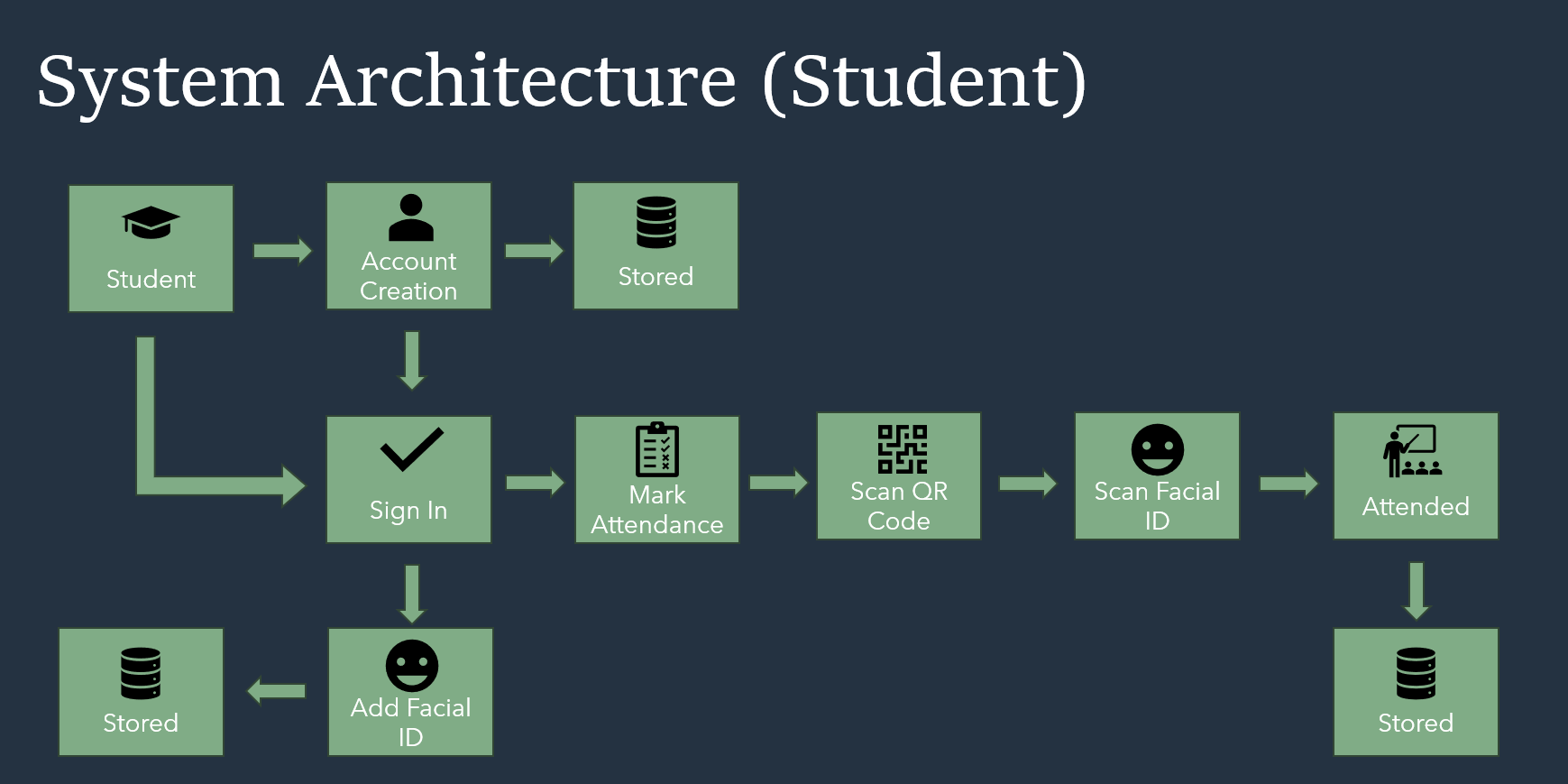
It follows the modular architecture design of the StudentSignIn application, which can clearly communicate between different modules in performing the process of attendance tracking of both students and lecturers. This ensures seamless integrations with user-faced interfaces, database systems, and verification processes.

#### **Component Design and Interactions**

* **User Interface Layer**The UI should be well-designed to make usage as smooth as possible both for students and lecturers.
  1. It logs the user in with UI, scans the QR code, and does face recognition for attendance.
  2. Lecturers will use the UI to generate QR codes and access attendance statistics.
* **viewModel Layer**The viewModel layer acts as the core of the system, coordinating tasks such as:
  1. Validating QR codes.
  2. Matching facial recognition data.
  3. Recording and retrieving attendance data.
* **Database Layer**The database layer handles the storage and retrieval of all relevant data:
  1. User profiles (e.g., student and lecturer details).
  2. Session information (e.g., QR codes, attendance logs).
  3. Timetables and schedules.
* **Communication Between Components**
  1. UI to viewModel: Communication happens as a reaction to events; these events may be a user scanning a QR code or simply logging in. In case a QR code is scanned, it sends the data to the viewModel where it interacts with the database.
  2. viewModel to Database: Communication is request-response based. The viewModel layer queries the database layer to fetch user details, verify session data, or store attendance records.
* **Validation and Security**
  1. Data flow between components includes multiple layers of validation. For instance:
     + QR codes are validated to ensure they belong to an active session.
     + Facial recognition data is verified against stored records to confirm identity.
  2. Access control functions are added into the communication design to restrict actions based on user roles (lecturer or student role).
* **Error Handling**
  1. The system includes mechanisms for handling errors at various levels. For example:
     + If a QR code is invalid, the viewModel layer sends an error message to the UI for user feedback.
     + If data retrieval fails, the system will have fallback mechanisms to ensure the system remains responsive.

#### **Student Interaction Flow**

* + The user logs into the user interface, selects a session (mark attendance).
  + The UI sends a request to the viewModel layer to initiate attendance recording.
  + The student scans the session QR code. The scanned data is sent to the viewModel, which validates it by checking the database for the active QR code session.
  + The student completes facial recognition. The captured data is sent to the viewModel, which matches it with stored records.
  + If both steps succeed, the viewModel layer updates the attendance record in the database layer and notifies the UI of the successful attendance.



**Figure 3.2:** System Architecture for the student side of the application.

#### **Lecturer Interaction Flow**

* + The lecturer logs in through the UI and selects a module from a course.
  + The UI sends a request to the viewModel to generate a QR code for the session.
  + The viewModel communicates with the database layer to store session details and provides the generated QR code to the UI.
  + The lecturer can later view attendance statistics. The UI sends a request to the viewModel layer, which retrieves the data from the database layer.



**Figure 3.3:** System Architecture for the lecturer side of the application.

### **Software Design**

The StudentSignIn application is designed in such a way that it can provide a safe, efficient, and user-friendly solution to handle attendance.

#### **Architecture Overview**

The system is divided into three primary layers: User Interface (UI), viewModel, and the Database.

* **UI Layer:**   
  It should be intuitively interactive for both a user, either a student or a lecturer. It will also be able to perform functions like login, scanning QR codes, face detection, and reviewing attendance records.
* **viewModel Layer:**   
  This layer processes user actions and ensures that rules are applied. It handles QR code scanning, facial recognition validation, and attendance logging.
* **Database Layer:**   
  Firebase is used to manage user data, attendance records, and session information. All data is stored securely in Firebase Firestore and Firebase Storage.

#### **Components and Methods**

Each component is designed to perform a specific function within the app. These are following examples of components which will be added to the app.

* **Authentication:**
  + Class: AuthManager
  + Methods: loginUser(), validateEmailDomain(), roleBasedAccess()
  + Description: Manages user sign-ins using Firebase Authentication. Ensures that only users with valid student or lecturer emails can log in.
* **QR Code Scanning:**
  + Class: QRCodeScanner
  + Methods: generateQRCode(), scanQRCode(), validateQRCode()
  + Description: Handles generating, scanning, and validating QR codes for attendance marking.
* **Facial Recognition:**
  + Class: FacialRecognitionManager
  + Methods: captureFacialData(), verifyFacialID(), compareFacialData()
  + Description: Handles the facial recognition process, comparing real-time images with stored student facial data in Firebase Storage.
* **Attendance Tracking:**
  + Class: AttendanceTracker
  + Methods: logAttendance(), updateAttendanceRecord()
  + Description: Records attendance once both QR code and facial recognition checks are successfully passed.

### **Functional and Non-Functional Requirements**

#### **Functional Requirements:**

* **Authentication:**   
  Implemented using Firebase Authentication, with role-based access control.
* **QR Code Scanning:**   
  Handled by the QRCodeScanner class, which validates the QR code before marking attendance.
* **Facial Recognition:**   
  Managed by the FacialRecognitionManager, ensuring high security and accurate identification.
* **Attendance Tracking:**   
  The AttendanceTracker class logs attendance automatically once the student is verified.

#### **Non-Functional Requirements:**

* **Performance:**   
  Optimized Firebase queries are used to ensure fast response times, even under heavy use.
* **Security:**   
  Firebase Authentication and facial recognition provide robust security.
* **Scalability:**   
  The system is designed to scale with Firebase, ensuring smooth performance with increasing users.
* **Usability:**   
  The UI is designed to be simple, intuitive, and easy to navigate for both students and lecturers.

### **User Interface (UI) Design**

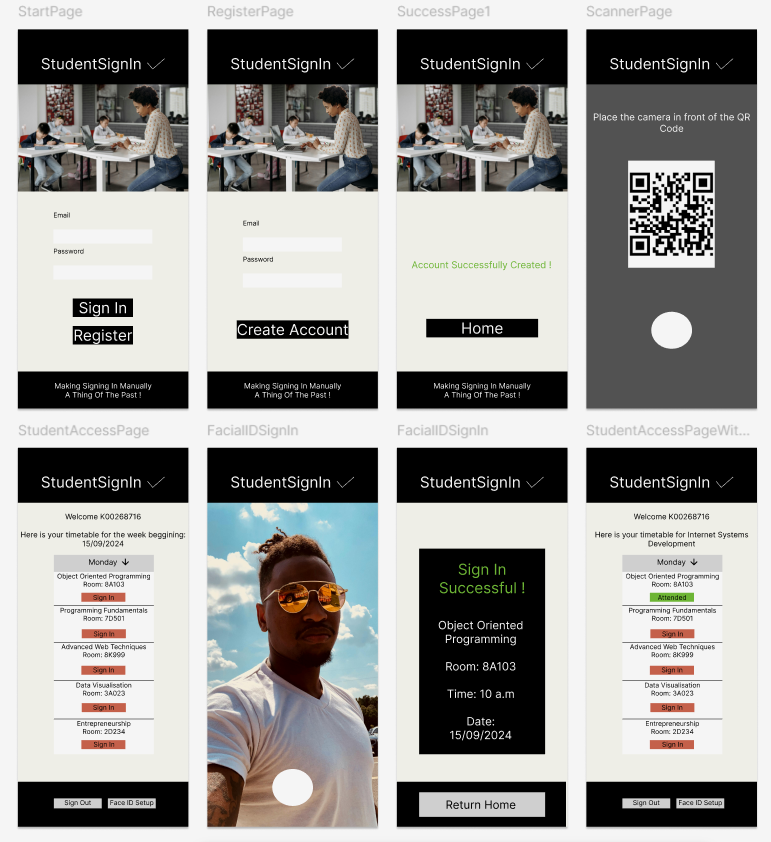
Excellence in design means simplicity, being user-friendly, and turning it into an experience free from troubles through the most important application views. The UI was designed in Figma, based on how to make the user feel comfortable during the navigation.

The Login screen is essentially the entry point into the application. It is neat but with clear prompts for entering an email address and password. Access is quite role-based; thus, each time one logs in, some kind of permissions is accorded to the users based on their status, be they a student or lecturer.

Upon logging in, users enter something called the Home Screen, where a timetable view pops right into sight, the place where the schedule of course time is given, and they can easily reach other main areas of the app, such as the QR scanner, in such a way that motion between each of those activities would be rather smooth.

Besides, the function of scanning the QR code and face identification is put into one: having succeeded with the QR scan, it automatically goes on to perform face recognition. Further refinement is achieved with clear instructions and responses smoothly guiding users through each little step.

The lecturers will have their personal dashboard where managing of students is made easy. They can add and remove student accounts from enrolled courses, thus providing better permission control in the users of this application.



**Figure 3.4:** Figma UI designs for student side of application



**Figure 3.5:** Figma UI designs for lecturer side of application

### **Database Design**

The database is designed in Firebase Firestore to support the app’s functionality efficiently, with collections made for each area of functionality:

* **Users Collection:**   
  Stores information for each user (students, lecturers, and admins), including:
* userID: Unique identifier
* email: User’s email address (for role-based access)
* role: Indicates whether the user is a student, lecturer, or admin
* courseID: Links students to their enrolled course
* faceImageURL: Storing facial image
* **Courses Collection:**   
  Contains course details, including:
* courseID: Unique course identifier
* courseName: Name of the course
* **Timetable Collection:**   
  Scheduling information for each course, including:
* courseID: This is a reference to the course the schedule belongs to.
* Day: Day of the week, e.g. Monday, Tuesday
* Time: Time of day the class is scheduled
* Location: Details of classroom or venue
* **Attendance**   
  Collection: Attendance of each session, with attributes such as
* sessionID: ID of each class session
* userID: This references the student who attended
* date: Date of the class session
* status: Attendance status (present/absent)
* **Firebase Storage:**   
  Storage unit for each person’s facial image
* /images/{UserID}.jpg: Image of user’s face

These are the relational relationships in designing the collections, which make retrieval of data with much efficiency. For example, attendance of a student could be tracked based on CourseID and UserID; this would, therefore, enable a lecturer to retrieve the list of attendance of both students and sessions.



**Figure 3.6:** ERD Database design for the application

* + 1. **Conclusion**

Attendance tracking will be provided through the StudentSignIn system in a very dependable and secured manner by integrating smooth use of QR code scanning with biometric facial recognition. It is expected that such thoughtful integration of technologies will ensure that the system meets all the standards of security, performance, and usability. The application was designed to use Firebase's capability for real-time updating, thus enabling instantaneous recording of attendance and ensuring data accuracy and security. It ensures smooth interactions because of its user-friendly interface for both students and lecturers on the application. The system can easily be scaled up when adding other integrations and caters to the needs of educational institutions for effective searching in managing student attendance.

# **Implementation Report**

This chapter presents an in-depth overview of the StudentSignIn application's development. The chapter goes through tools and technologies utilized, explaining their reasons for choice and their importance in enabling the application's primary capabilities. From sign-in and attendance registration to QR code generation and details on face recognition, every piece is examined in detail to identify their significance in building an integrated and safe student sign-in system.

## **Tools and Technologies**

The selection process for tools and technologies is a serious step for the development of the StudentSignIn application to guarantee the success of the project by meeting the goals set for the project. Below is the outline of key tools, technologies, and platforms used during the development, and justification for using them.

### **Programming Languages**

#### **Kotlin**

The main programming language used during the development of StudentSignIn was Kotlin because it is the preferred language for the development of Android applications. Compared to Java, Kotlin has some favourable feature which include clear syntax, null safety, and a modern way of programming that reduces code maintenance efforts and decreases the likelihood of runtime errors. Boilerplate code reduction increases productivity immensely, therefore allowing the developer to invest the maximum amount of time in implementing functionality rather than adding repetitive code.

Kotlin has been indispensable in the development of the StudentSignIn application because of its deep integration with Jetpack Compose, used to express a fluid UI. It was simple to construct given the simplistic nature of Jetpack Compose in developing complex feature screens such as the QR code scanner and facial recognition pages. Kotlin's compatibility with the Firebase SDK rendered seamless integration of authentication, database management, and cloud storage within the app. Its strong coroutine framework allowed some ability to achieve asynchronous operations with assurance that time-critical tasks, such as database queries, image uploads, and biometric verification, were executed without hindering the user experience.

A screen shot of a computer program

Description automatically generated

**Figure 4.1:** Example of Jetpack Compose code from the student Hub Page

Finally, Kotlin provided the ability to efficiently manage application navigation, error handling, and the processing of data. It further allowed the use of different libraries such as the Google ML Kit or Camera-X, which could easily be integrated into the project. Equipped with these state-of-the-art capabilities, Kotlin empowered the creation of StudentSignIn-a secured, reliable, and scalable application to meet all complex requirements related to the student attendance system.

### **Integrated Development Environments (IDE’s)**

#### **Android Studio**

The main development environment for building StudentSignIn was Android Studio, the official Integrated Development Environment for developing Android apps. It provided a strong tool set that catalysed development immensely, enabling the creation of a full attendance management system.

Writing and maintaining the Kotlin code was powered by the powerful code editor at the heart of Android Studio. From error detection and smart code completion to refactoring tools, this IDE supported the creation of even the most complex features, from user authentication and real-time database operations down to the implementation of biometric functionality.

The debugging tools in Android Studio have been used throughout the development life cycle. This exposes the capability to log during testing with Logcat, this has really helped a lot with troubleshooting responses, querying the database, and asynchronous operations. The in-built emulator was also used to test the app on different types of Android devices and with all sorts of configurations to see how the app behaves on different devices. This helped validate the performance of both the QR code scanning and facial recognition processes under various conditions.

A screenshot of a computer program

Description automatically generated

**Figure 4.2:** Using the logcat to retrieve valuable testing information regarding facial ID

The integration of Firebase within Android Studio further goes on to ease backend connectivity. With the Firebase Assistant tool provided in Android Studio, the application is connected seamlessly with services such as Authentication, Firestore, and Storage. Such aid has been especially helpful in keeping data synchronized in real time, securely logging users in, and storing user images for biometric verification. The ability to set up the database in this way all from within Android Studio saved a lot of time and ensured tight coupling between the front-end and the backend.

A screenshot of a computer

Description automatically generated

**Figure 4.3:** Images of user faces being kept in Firebase Storage unit

Also, the inclusion of version control with Android Studio into Git was greatly helpful in managing this project's source code. Android Studio’s built in GitHub tools allow easy creation of branches, merges, and tracking of changes, which was crucial in ensuring smooth collaboration and prevention of code conflicts when adding new features or bug fixes.

Through using the many features provided by Android Studio, StudentSignIn was able to be effectively created to offer quality code, robust functionality, and a smooth user experience. Features of the IDE enhance productivity but contribute to the scalability and reliability of the app—proving to be a solid base for the attendance management system.

### **UI/UX Design Tools**

#### **Figma**

Figma that was chosen for designing the UI and UX of the application. Figa is a digital tool used online in designing and collaboration. Its effectiveness in collaboration makes it very suitable for the iterative process of designing and testing UI elements. Figma allowed making high-fidelity prototypes of the StudentSignIn app, visualize the user flow, and get feedback before actual development was started, which helped smooth the design process and allowed the interface of the app to be developed more intuitive and accessible for its users. Figma designs we’re built early on purpose to leave room for improvement as it went through multiple iterations of how the app should look and behave. In the end the final design worked well from the visuals and in terms of the flow through the app, with a mix of nice sleek design and the professionalism you would expect from an app targeted at colleges.

A screenshot of a mobile application

Description automatically generated

**Figure 4.4:** Figma Designs for StudentSignIn application

### **Libraries**

#### **Google ML Kit for Facial Recognition**

The Google ML Kit is a machine learning library that allows for easier use of machine learning capabilities. In the application StudentSignIn, the Face Detection API developed by ML Kit is selected since it will power the Facial Recognition Feature on this app. This is the needed library to detect the faces through the camera to identify the students at sign-in.

Using ML Kit, StudentSignIn was enabled to capture live camera frames, doing further processing to trace facial features such as eyes, nose, and mouth. Later, the app compared that taken facial data against the stored images of the registered user; the app verified their identity. Marking the attendance of the students

A computer screen shot of a program code

Description automatically generated

**Figure 4.5:** compareFaces function used in the student facial recognition page

This allowed the app to implement seamless facial recognition without much lag, as it helped students sign in quickly and securely. The prebuilt machine learning models of ML Kit also came with the benefit of requiring large custom models or complicated algorithms.

#### **ZXing for QR Code Generation**

ZXing, an open-source barcode image-processing system, which is capable of scanning and generating QR codes. In StudentSignIn, ZXing is used to produce a distinct QR code for every class session to ensure only students physically in the classroom may scan and validate their attendance.

The MultiFormatWriter in ZXing is designed to encode session data in a QR code and then convert to bitmap for visually oriented presentation. On the scanning side, ZXing's BarcodeReader effectively scans such QR code using the phone's camera, ensuring efficient and accurate processing.

This implementation guarantees safe and effective sign-up through barcode, preventing any student from attempting to evade attendance monitoring through means of using screenshots or expired codes.

#### **CameraX for Camera Handling**

CameraX is an Android Jetpack library that allows easy integration of the camera into an Android application. Unlike the older Camera2 API, CameraX offers consistency across Android devices hence, when developing you do not have to concern himself with camera hardware compatibility. CameraX had been implemented to control the front camera for capturing face authentication and the back camera for scanning QR codes.

One of the major benefits of using CameraX is that it's easy to use, while providing great camera output without any fuss related to low-level camera APIs. This made the implementation of the camera feature easy and reliable inside StudentSignIn. The CameraX library will also let users see feedback when either the QR code scanning or facial recognition fails.

Since each different smartphone had different camera specifications, compatibility of the library with a wide range of Android devices was vital in making StudentSignIn work perfectly. In turn, CameraX handled the camera streams efficiently to allow operating in parallel for facial recognition and QR code scanning without lag while transitioning through different functionalities in the app.

## **Environment Configuration**

For success, the StudentSignIn application requires thorough and meticulous configuration of the development and runtime environments. This section describes the major configurations of both environments for developing, testing, and deploying the application.

### **Development Environment**

The development environment for StudentSignIn was configured to ensure an efficient workflow, enabling the development of both frontend and backend components of the application.

#### **Emulator Setup**

The Android Emulator was used during development to emulate various Android devices for testing various screen dimensions, different OS versions, and varied performance conditions.

Simulations ranging from simple and basic Android devices to high-end models were to be done on the emulator. This would cover all kinds of hardware to take the application through performance evaluation.

#### **Version Control**

GitHub was used for version control to track changes in the code and collaborative development. GitHub provided a host for the project with a remote repository where source code management for the project was done. The branching and pull request features were used to control how the new features and bug fixes would be integrated.

### **Runtime Environment**

The runtime environment regards the actual devices and services on which the StudentSignIn app runs during its usage by students and lecturers.

#### **Android Devices**

The Android-based application utilizes the students' smartphones to scan QR code and perform face recognition. The application was tested both on multiple Android smartphones and older models to work on diverse hardware platforms with a front camera whose resolution is good enough to enable face recognition.

The app requires an Android device running Android 6.0 (Marshmallow) or higher, with a working camera for the QR code scanning and facial recognition features. This is due to the use of camera API’s, ML Kit Library for facial recognition and the fact Android 6.0 is compatible with most active Android devices.

## **Database Implementation**

Having a correctly set up database is one of the core functionalities for the StudentSignIn application. Information stored includes things such as user data, records of attendance, timetable information, and face recognition data. Firebase Firestore is a NoSQL, cloud database chosen for the back end because of its scalability, synchronization, and ease of integration with Android. This section expands on the structure of the database, the decisions with respect to data modelling, and justifies the adopted approach of implementation.

### **Rationale for choosing Firebase Firestore**

* **Real-time Syncing:**   
  Firebase Firestore enables real-time updates, ensuring that attendance data and timetable changes are instantly reflected across all users.
* **Scalability:**   
  Firestore's ability to scale horizontally allows the database to handle growing numbers of users and attendance records without degradation in performance.
* **Ease of Integration:**   
  Firebase SDK integrates seamlessly with Android Studio and Kotlin, streamlining the development process.
* **Security:**   
  Firestore provides robust authentication and authorization mechanisms, which were configured to ensure that only authorized users could access or modify data.
* **NoSQL Structure:**   
  The flexible schema of Firestore was ideal for the dynamic and hierarchical data required by StudentSignIn.

### **Database Structure**

The database is organized into collections and documents, following Firebase Firestore's NoSQL structure. Below is an overview of the key collections and their importance within the application.

#### **Collections**

There are four major collections in the database: users, attendance, timetable, and sessions. Each one of these serves a different purpose and offers fluent functionality across the application.

The user’s collection holds information of all the users of the application be it students or lecturers. Every user is assigned a unique identifier (UID) and email. Further, there is a field for the role, which would specify the user as a student or lecturer, and another for the course, associating the user with a specific course. This is very much required for proper implementation of role-based access and providing personalized functionalities to the user according to a role.

The attendance collection stores the sign in of students attending lectures. Each of them will be marked with a timestamp of when the attendance was taken, a timetableEntryId pointing to which class it is for, and a userId for the student who has signed in. This collection is essential in maintaining attendance and data which can be viewed at the lecturer side of the application.

The timetable collection holds the classes for a module taken by students. Every class is given a classId and will be part of a particular course. The other fields that have been generated are day of the week, start time and end time, subject name, and lecturer Id. A totalClasses field has also been included to maintain the number of sessions in a semester. This collection has been used for the purpose of enabling both the students and lecturers to view classes in which they are enrolled on.

Finally, the sessions collection stores information used to verify attendance. Each session is identified by a classId and a timestamp that indicates the date and time of the session. A uniqueIdentifier is generated for each session, which is embedded in a QR code for attendance purposes. This unique identifier, therefore, prevents abuses such as using screenshots of QR codes, allowing only valid sessions to be recorded.

Collectively, they provide the core functionality for the StudentSignIn application, covering functionality such as attendance tracking, timetabling, and authenticating sessions through QR codes.

#### **Firebase Configuration**

The Firestore database was configured with rules to ensure secure and efficient access:

1. **Role-Based Access Control (RBAC):**

* Students can only view and modify their own attendance records.
* Lecturers have additional privileges, such as modifying attendance statuses.

1. **Validation Rules:**

* Email-based authentication restricts unauthorized access.
* Queries are filtered to prevent retrieval of unnecessary data.

A screenshot of a computer program

Description automatically generated

**Figure 4.6:** Firebase rules setup

#### **Data Flow**

The following describes the interaction between the app and the database during key operations:

**User Authentication:**

Upon login, the app verifies credentials using Firebase Authentication and fetches the user’s role and course from the user’s collection.

**Attendance Marking:**

When a student scans the QR code and completes facial recognition, a new document is added to the attendance collection with the corresponding timetableEntryId.

**Timetable Retrieval:**

Based on the user’s course, the app queries the timetable collection to display relevant classes for the day.

**Session Validation:**

QR codes generated for each session are validated against the sessions collection to ensure authenticity.

## **Feature Implementation**

The three major functionalities in the StudentSignIn application are based on QR code-based attendance, face authentication, and user-specific timetables. This section will proceed with the discussion of the implementation details of each feature, discussing major development decisions taken and technologies used.

### **Key Features**

#### **User Authentication**

The user authentication provides secure access to the application and allows differentiation between a student and a lecturer. When a user is registered, there is identification against the email and role assignment as a student or lecturer. The Firebase Authentication allowed the use of email/password-based logins which is done through the cloud, though further steps were taken to query the user's collection for details on the authenticated user. Certain risks, like unauthorized registering, were overcome by allowing only admins (Users with access to the database) to create users. Role-based access was granted to users based on the designation of the user to different parts of the application as a student will have different permissions on the app compared to a lecturer. There are error handling mechanisms which handle invalid credential errors or connectivity issues.

A computer screen shot of a program

Description automatically generated

**Figure 4.7:** Querying the “users” collection to check the role of the current user

#### **Timetable Management**

It displays the dynamic timetable of courses that the user is assigned to. Students will be able to see their own timetable, while the lecturer will be able to see timetables for all classes he teaches. The data are fetched from the timetable collection, filtering the user's course and the day of the week selected. Defaulting it sets "Monday" and gives a choice to the user to switch using the dropdown menu. The Jetpack Compose populates the timetable dynamically for a smooth user experience. With this in place, the problems associated with the efficient filtration of data and timely rendering are solved through the implementation of indexed queries for the course and day fields.

A screenshot of a computer

Description automatically generated

**Figure 4.8:** Student timetable page

#### **QR Code-Based Attendance**

This functionality will enable signing in using a QR code, which shall be provided by the lecturer. Every QR code belongs to a session and has a timestamp for validation purposes. After generating a QR code through the lecturer's app, the QR code will be saved in the sessions collection. The student app will scan the QR code using the CameraX library and verify it against the session data embedded in it. If verification is successful, it will go on to the facial identification stage. Anti-misuse measures regarding QR code screenshots, among others, have been introduced through unique identifiers and timestamp validation; QR codes will expire after some time.

A qr code on a white background

Description automatically generated

**Figure 4.9:** Image of QR code page

#### **Facial Recognition**

Facial recognition attempts to work in a way that no other student, but the enrolled one can mark the attendance. It uses the Google ML Kit for face detection and Firebase Storage for storing reference images of users as previously discussed. Tuning of the threshold (how alike the live image is to the stored facial image) was done to get the best quality facial recognition while not interfering with the overall speed and flow of the app.

#### **Attendance Records**

The application keeps attendance records that lecturers can view for analysis, while students may view their attendance history. The attendance collection will store attendance information that links a student, timetable entry, and session. Listeners were set up with Firebase Firestore so that any change would reflect instantly on the attendance dashboard for smoothness.

## **Challenges Faced**

The different stages of development in StudentSignIn have gone through their technical, design, and implementation challenges, which ranged from the integration of facial recognition within the system to how to manage the data flow in a scalable and non-vulnerable way. This section will discuss the important hurdles during the development and the strategy that was followed to resolve them.

### **Technical** **Challenges**

#### **Facial** **Recognition**

The biggest challenge with facial recognition was the ability to recognize and accurately provide seamless UI. The current face detection done with the use of Google ML Kit and Firebase Storage was less robust than originally anticipated. By setting a threshold for the matching of the face, it showed that setting the threshold value higher resulted in an increase in false positives, while setting it too low resulted in rejecting valid users. While the code does target a face and attempts to compare facial structure, it does vary in terms of success rate. In combating this, further testing of thresholds needs to be done.

#### **Data** **Synchronization** **and** **Performance**

Data across devices and users had to be kept in real time to keep track of attendance records, timetables, and user data, this presented a big challenge in keeping the database performance high. On the back of Firebase Firestore, speed and reliable data access needed structuring with things such as indexing. Too complicated a query resulted in a longer fetch time, especially when filtering timetable entries by course and day. This is where indexed fields came into play as they organise the data for faster retrieval, keeping the performance high while reducing the chances of conflicts in data.  
A screenshot of a computer

Description automatically generated

**Figure 4.10:** Indexes needed for complicated “attendance” collection query’s

#### **Passing Data**

The actual use of NavBackStackEntry, while implementing the project, posed quite a lot of challenges regarding data flow between pages due to the complexity involved in the transfer and maintenance of such critical data like userId, courseId, and courseName. These pieces of data were vital to the functionality of the app, especially recording attendance data into Firebase accurately and making sure the lecturers could use the data stored on their end of the app.

When data passed between pages was incorrect or incomplete, it created issues with the inconsistencies of the information recorded and caused errors from the intended functionality. Again, logging was the approach to debugging to trace and monitor what data was getting passed through at each step of navigation. This methodology allowed finding discrepancies and quickly making changes, so the right data would always be delivered to the proper pages.

A screen shot of a computer program

Description automatically generated

**Figure 4.11:** NavGraph using backStrackEntry to pass data through pages

### **Design and Development Challenges**

#### **Navigation and UI Consistency**

Seamless navigation among all the different screens of the application was challenging while maintaining UI consistency throughout on unique pages such as the QR code scanner, timetable viewer, and face recognition. Although Jetpack Compose had made the creation of modern UIs a lot easier, these pages presented more of a challenge due to their already complex nature.

#### **Error** **Handling and User Feedback**

Since the app relied on hardware such as cameras, robust error handling was a must. Challenges arose how to handle users have camera permissions turned off, and how to provide some kind of useful feedback to the user upon error. For example, if the user does not have camera permissions turned on an error indicating that permissions are not active. This helped improve overall user satisfaction, supported by comprehensive error messages and fallback mechanisms.

A screenshot of a video chat

Description automatically generated

**Figure 4.12:** Request to use the camera while using the app

### **Time** **Constraints**

Time constraint was the major issue because it was a final year project, and hence we were on a tight deadline. Advanced technologies like Google ML Kit were involved in the whole process and the integration had to be done through proper time management. This meant proper prioritization of features according to their impact was necessary. Features such as the QR code technology, facial recognition and signing into a class took precedence.

## **Conclusion**

The StudentSignIn application is designed for only one thing: to improve students' and staff's attendance, focusing on both efficiency and security. Leveraging on advanced technologies including QR code and facial recognition, and underpinned by strong backend by Firebase, the application bridges the limitations of old attendance systems while keeping strong focus on scalability and on security.

The reasoning behind such design was to improve attendance-taking, making sure to make it not only faster but accurate, while maintaining an strong user experience. Using Kotlin and Firebase, both of which combine to provide a solid and scalable base for the application, made possible to smoothly incorporate timetabling, sign-ins using biometrics, and synchronized data in real time.

During the development stage, several crucial challenges were met, including fine tuning for facial recognition to obtain optimal precision, protection of QR codes against spoofing, and working around the complexities of real-time data manipulation using Firebase. Despite such barriers, however, the application has found means to provide safe and accessible operation, setting up for future development, such as advanced analytics and offline capabilities.

This proof of concept demonstrates how contemporary technologies have the capability to develop an innovative and scalable attendance system, presenting a solid platform for future expansion and improvement.

# **Testing and Results**

## **Introduction**

The Testing and Results chapter looks deeply into the applications performance, safety, user-friendliness, and how it functions. This app has been designed for lecturers as well as students to increase their classes' efficiency through facial recognition technology in combination with QR codes. The testing chapter’s goal is to prove that the app achieves its objective while living up to performance, safety, and user-friendliness standards.

Testing involves functional requirements like user authentication at login, accessing timetables, marking of attendances, and the scanning of QR codes as well as non-functional requirements that focus on performance, ease of use, safety, and overall performance. Through systematic testing and thorough analysis of results, StudentSignIn can be not only effective but also dependable.

## **Unit Testing of Non-Functional Requirements**

### **Performance**

* Load Times
* QR Code Scanning Speed
* Facial Recognition Processing Speed
* Database Query Performance

### **Usability**

* Navigation Flow
* Error Message Clarity
* Responsiveness across devices

### **Security**

* QR Code screenshot prevention
* Authentication and authorization verification

### **Compatibility**

* Cross device compatibility
* Camera compatibility for facial recognition

## **Non-Functional Requirements Test Cases**

### **Performance Test Cases**

|  |  |
| --- | --- |
| **Test Case Name** | **QR Code Scanning Speed Test** |
| Test Case ID | 01 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 05/02/2025 |
| Description/Summary of Test | Test the time taken to scan / process QR code |
| Pre-Condition | User is logged in and scans a valid QR code |
| Inputs | QR code provided by lecturer |
| Post-Condition | Navigate to facial recognition scanner |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | QR code scans instantly |

|  |  |
| --- | --- |
| **Test Case Name** | **Facial Recognition Process Speed** |
| Test Case ID | 02 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 05/02/2025 |
| Description/Summary of Test | How quickly can the facial recognition match the user |
| Pre-Condition | User is logged in and has scanned a valid QR code previous |
| Inputs | Users face for recognition |
| Post-Condition | User is authenticated / failed recognition |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | Facial recognition speed is around 15-20 seconds to recognise a user |

|  |  |
| --- | --- |
| **Test Case Name** | **Database Query Performance** |
| Test Case ID | 03 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 05/02/2025 |
| Description/Summary of Test | The speed of database queries when fetching timetable and attendance data |
| Pre-Condition | User has either been enrolled in a course / teaching one |
| Inputs | Query to retrieve timetable |
| Post-Condition | Timetable is displayed in a reasonable time (under 5 seconds) |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | Timetable / attendance data displays almost instantly on the load of the Hub Page |

### **Usability Test Cases**

|  |  |
| --- | --- |
| **Test Case Name** | **Navigation Flow** |
| Test Case ID | 04 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 05/02/2025 |
| Description/Summary of Test | How easy is it to move through the app |
| Pre-Condition | User is logged in and is on a Hub Page |
| Inputs | User interaction (signing in, generating QR code etc.) |
| Post-Condition | Successfully navigate app without confusion |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | App has a forward momentum as one checkpoint leads to the next until signed into class |

|  |  |
| --- | --- |
| **Test Case Name** | **Error Message Clarity** |
| Test Case ID | 05 |
| Test Priority | Medium |
| Test Executed By | Jason Price |
| Date of Test Execution | 05/02/2025 |
| Description/Summary of Test | How clear and helpful are the error messages within the app |
| Pre-Condition | User intentionally causes an error such as entering the wrong password |
| Inputs | Error message displays |
| Post-Condition | User can move on having understood error |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | App has error messages for face detection, invalid details and expired QR Codes to help guide the user. |

### **Security Test Cases**

|  |  |
| --- | --- |
| **Test Case Name** | **QR Code Screenshot Prevention** |
| Test Case ID | 06 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 09/02/2025 |
| Description/Summary of Test | Can the user screenshot the QR code to send to users outside the classroom |
| Pre-Condition | Image is taken of QR Code |
| Inputs | Attempt to take image of QR Code |
| Post-Condition | App prevents unauthorised sign in |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | QR code regenerates every 15 seconds preventing spoofing |

|  |  |
| --- | --- |
| **Test Case Name** | **Authentication and Authorization** |
| Test Case ID | 07 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 09/02/2025 |
| Description/Summary of Test | Does the system only allow authorized users |
| Pre-Condition | Are users with invalid credentials rejected |
| Inputs | Invalid student / lecturer credentials |
| Post-Condition | User does not gain access to application |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | Users without a valid “@student” or “@lecturer” email are rejected |

### **Compatibility Test Cases**

|  |  |
| --- | --- |
| **Test Case Name** | **Camera Compatibility for Face Recognition** |
| Test Case ID | 08 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 09/02/2025 |
| Description/Summary of Test | Test the compatibility of the app’s facial recognition against multiple android devices |
| Pre-Condition | User is prompted to do facial recognition on supported Android device |
| Inputs | User’s face using different devices |
| Post-Condition | User is signed into class |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | Tested on 2 physical Android devices, both supported the use of Camera X and allowed user to proceed to sign in as well as multiple emulated Android devices |

## **Finding of Non-Functional Testing**

Non-functional testing was performed to determine performance, usability, security, and reliability of the StudentSignIn application. The findings of which are presented below.

### **Performance Findings**

Performance testing was focused on testing the responsiveness and speed of vital features like facial recognition and QR code scanning. Scanning of QR codes was analysed across varying lighting conditions and orientations. Findings revealed that QR codes were correctly identified almost instantly, though performs best in well-lit conditions. Facial recognition was found to take around 15 to 20 seconds in initial testing depending on the quality of the camera and how high/low the threshold of similarity was set to within the code itself. There was a performance drop in low-light conditions, emphasizing the need for more robust image pre-processing methods for higher accuracy and faster performance.

### **Usability Findings**

Usability testing was performed to analyse user experience in general as well as ease of navigation in the app. Testers found that the interface of the app was user-friendly with minimal prompts needed to reach its basic features. Error message clarity in cases of wrong methods of user authentication was appreciated in general user experience. Additionally, helpful hints like requests for permission for camera that was not yet granted helped in allowing users to go on with their activity in the app with ease.

|  |  |
| --- | --- |
| A screenshot of a cell phone  AI-generated content may be incorrect.  **Figure 5.1:** Testing access request (1) | A screenshot of a computer  AI-generated content may be incorrect.  **Figure 5.2:** Testing access request (2) |

### **Security Findings**

Security testing was implemented to determine vulnerabilities related to authentication and data manipulation. All old QR codes were blocked by the system perfectly, thus preventing unauthorized check-ins. Additionally, all user credentials were checked to verify their secure storage in Firebase through the implementation of authentication. Any unauthorized accounts were unable to sign in and were met with a “Incorrect email or password” message thus preventing any manipulation of student data.

|  |  |
| --- | --- |
| A screenshot of a computer  AI-generated content may be incorrect.  **Figure 5.3:** QR error message testing | A person sitting at a table with children  AI-generated content may be incorrect.  **Figure 5.4:** Sign in error message testing |

### **Reliability Findings**

Reliability testing was performed to ensure that the performance of the application was consistent on all devices and conditions. Various Android devices of varying specifications were used for testing purposes. The app performed perfectly all through, with only slight speed differences regarding facial recognition. The app was run continuously for lengthy intervals to determine if it would crash but found the app to be reliable and steady throughout use.

### **Non-Functional Testing Thoughts**

Overall, the non-functional testing of StudentSignIn demonstrated outstanding performance in all-critical aspects of QR scanning, security, and reliability. Incremental improvement in facial recognition processing speed would further improve performance. All these points of feedback shall guide future improvement.

## **Unit Testing of Functional Requirements**

### **Functional Tests**

* Log in (Student and Lecturer)
* View Timetable
* Marking of Attendance
* QR Code Generation
* Facial Recognition
* Edit timetable data (lecturer side)
* Attendance Statistics

## **Functional Requirements Test Cases**

|  |  |
| --- | --- |
| **Test Case Name** | **Student Login** |
| Test Case ID | 09 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 13/02/2025 |
| Description/Summary of Test | Validate whether student can log in using valid credentials |
| Pre-Condition | User must have a registered student account |
| Inputs | Valid student email and password |
| Post-Condition | User is successfully logged in and brought to the Hub Page |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | Users with valid “@student” emails are given access to the student Hub Page |

|  |  |
| --- | --- |
| **Test Case Name** | **Lecturer Login** |
| Test Case ID | 10 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 13/02/2025 |
| Description/Summary of Test | Validate whether lecturers can log in using valid credentials |
| Pre-Condition | User must have a registered lecturer account |
| Inputs | Valid student email and password |
| Post-Condition | User is successfully logged in and brought to the Hub Page |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | Users with valid “@lecturer” emails are given access to the student Hub Page |

|  |  |
| --- | --- |
| **Test Case Name** | **View Timetables** |
| Test Case ID | 11 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 13/02/2025 |
| Description/Summary of Test | Validate whether students and lecturers can view their specific timetables |
| Pre-Condition | Student must be logged in and assigned a course  Lecturer must be assigned to a module |
| Inputs | Sign in to get to Hub Page |
| Post-Condition | The user sees the correct timetable for their course |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | Students view their course modules whereas lecturers view which classes they have been assigned (Lecturers may teach different courses) |

|  |  |
| --- | --- |
| **Test Case Name** | **Generate QR Code for class** |
| Test Case ID | 12 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 13/02/2025 |
| Description/Summary of Test | Ensure lecturers can generate a valid QR code for attendance. |
| Pre-Condition | Lecturer must be logged in. |
| Inputs | Select class and generate QR code |
| Post-Condition | A unique QR code is generated for the session. |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | QR code worked as expected, updating the database every 15 seconds with a new Unique ID |

|  |  |
| --- | --- |
| **Test Case Name** | **Mark Attendance via QR Code** |
| Test Case ID | 13 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 13/02/2025 |
| Description/Summary of Test | Validate that students can mark attendance by scanning the QR code |
| Pre-Condition | Lecturer has generated a valid QR code.  Student is in the correct class and is logged in. |
| Inputs | Scan QR Code using the app |
| Post-Condition | Student is taken to facial recognition for further authentication |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | Originally encountered an error where QR code scanned twice which has since been fixed |

|  |  |
| --- | --- |
| **Test Case Name** | **Facial Recognition for Attendance** |
| Test Case ID | 14 |
| Test Priority | High |
| Test Executed By | Jason Price |
| Date of Test Execution | 20/02/2025 |
| Description/Summary of Test | Ensure facial recognition correctly authenticates students |
| Pre-Condition | Student has scanned a valid QR Code  Student has a registered face within the system |
| Inputs | Student positions face in front camera |
| Post-Condition | If matched -> Attendance is marked  If not matched -> Access denied |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | While functionality operates as expected (only a matched face can continue), accuracy of facial recognition remains a slight issue |

|  |  |
| --- | --- |
| **Test Case Name** | **Attendance Statistics Viewing (Lecturer Only)** |
| Test Case ID | 15 |
| Test Priority | Medium |
| Test Executed By | Jason Price |
| Date of Test Execution | 20/02/2025 |
| Description/Summary of Test | Verify that lecturers can view attendance statistics for their classes. |
| Pre-Condition | Lecturer is logged in and assigned to a module |
| Inputs | Navigate to Attendance Statistics Page |
| Post-Condition | Lecturer can see student attendance data for their module |
| Status (Fail/Pass) | Pass |
| Notes/Comments/Questions: | Attendance data updates in real time for the lecturer to view |

## **Finding of Functional Testing**

Functional testing was focused on testing the basic features of the StudentSignIn app, which included logging in, viewing the timetable, marking attendance and of course facial recognition.

### **Login Functionality Findings**

Testing of both lecturer and student login proved that only users with legitimate credentials would have easy access to their specific Hub Pages. During student login testing, the system was able to identify users with a "@student" email suffix to provide them with access to the student Hub Page. Similarly, lecturers having a "@lecturer" email suffix were correctly validated to redirect them to the lecturer Hub Page. During testing, all the login aspects acted as expected with no notable errors reported.

### **View Timetable Findings**

Both students and lecturers both could view their custom schedules. Pupils could view modules specific to their course, whereas teachers could see their own schedules of classes they teach as they may teach in multiple courses. The system was able to distinguish perfectly between teachers' schedules and pupils' schedules with no glitches or errors reported in the testing session.

|  |  |
| --- | --- |
| **A screenshot of a computer  AI-generated content may be incorrect.**  **Figure 5.5:** Testing student timetable | **A screenshot of a cell phone  AI-generated content may be incorrect.**  **Figure 5.6:** Testing lecturer timetable |

**QR Code Generation and Scanning Findings**

The functionality to generate QR codes was put through intense testing to make sure that lecturers would have the capacity to create a distinctive code for every session. Besides that, students' capacity to scan the produced QR code was tested to mark their presence. Lecturers managed to create QR codes while the system updated the database with new unique IDs every 15 seconds. Students managed to scan the QR codes perfectly, directing them to facial recognition for further checks. During testing, a glitch was found where double scanning of the QR code caused difficulties in the process of students marking attendance. This was attended to by allowing one QR code scans for every session. Another slight error found was when the QR code regenerated (as it does every 15 seconds) you would be brought back to the QRCodePage from wherever you were in the application. To fix this a button to stop the regeneration function was put into place.

### **Facial Recognition Findings**

Facial recognition technology allowed users to identify themselves by simply scanning their face to which they would be marked attended. If facial features didn't match, the system would produce an error “Face not detected” until a match was found. Facial recognition was seamless in ideal conditions but at other times faltered in cases of varying lights and odd positions of the camera, causing holdups in cases of suboptimal conditions. Errors were also found in the level of accuracy of the software due to the limitations of certain libraries. Further enhancements will be required to bring the standard of the facial recognition technology up such as manipulating the threshold of how accurate the face must be to sign in.

|  |  |
| --- | --- |
| A person taking a selfie  AI-generated content may be incorrect.  **Figure 5.7:** Testing facial recognition | A screenshot of a computer  AI-generated content may be incorrect.  **Figure 5.8:** Testing if signed in |

### **Attendance Stats Viewing Findings**

Attendance was also assessed as lecturers should be able to view and interact with students for the classes which they teach. Users managed to handle the Attendance Statistics Page with ease, where fresh records of student attendance were at their disposal at all points in time. The system produced accurate attendance records that automatically refreshed. Although more data could be included in future versions for lecturers to view.

|  |  |
| --- | --- |
| A screenshot of a cell phone  AI-generated content may be incorrect.  **Figure 5.9:** Testing attendance viewing | **Figure 5.10:** Testing attendance statistics |

### **Functional Testing Thoughts**

The functional testing of StudentSignIn demonstrated that all the necessary features including login, view of timetable, generation of QR codes for scanning, facial recognition, and statistics of attendance were working. Although some minor bugs in the shape of repeated QR codes' scanning and low-light facial recognition difficulties did arise, these were for the most part either fixed or greatly improved. Overall, the app is working to specifications, and these results shall provide good guidelines for future improvement in areas like facial recognition reliability and ease of user interface.

## **Test Data**

Testing was exhaustive in scope to include a broad range of scenarios to verify that all features from things such as lecturer and student logins to facial recognition and attendance records worked perfectly. Below is the test data which made this possible:

**Student Login**

* **Email:** jdawg@student.tus.ie
* **Password:** 22052003

**Lecturer Login**

* **Email:** jason@lecturer.tus.ie
* **Password:** 123456

**Invalid Data**

* **Email:** jason@tus.ie
* **Password:** 123456
* **Outcome:** Rejection due to being in an invalid format (neither a student nor lecturer email)

**Timetable Data (For Students)**

* **Course:** ISD
* **Modules:** ProMath, Programming Fundamentals
* **Day:** Monday
* **Outcome**: The app should display a list of modules for the ISD course on Monday. This should include ProMath and Programming Fundamentals.

**Timetable Data (For Lecturers)**

* **Lecturer:** jason@lecturer.tus.ie
* **Modules:** ProMath, Programming Fundamentals
* **Day:** Monday
* **Outcome:** The app should display a list of modules the lecturer, in this case, "jason@lecturer.tus.ie", is assigned to. This should include ProMath and Programming Fundamentals.

**Valid QR Code**

* **Generated QR Code:** Unique session for ProMath (Generated by app)
* **Outcome:** The app should successfully scan the QR Code and transition to facial recognition page.

**Invalid QR Code**

* **QR Code:** Expired, older than 15 seconds
* **Outcome:** The app should reject the QR code and prevent the student from progressing to the facial recognition page.

**Facial Recognition Data**

* **Registered Face:** Jason Price for user jdawg@student.tus.ie
* **Outcome:** The app should successfully match the registered face and proceed marking the attendance

## **User Testing**

In pursuit of better understanding the system's usability, external participants were invited to interact with the app and provide their thoughts on the application. Among them is my brother, Shane, who is not technical, and fellow students who have technical ability and testing backgrounds.

Shane found the login process straightforward but suggested providing clearer instructions for QR scanning. He also reported a slight delay in facial recognition but, overall, found the system intuitive.

My classmates reassured me that expired QRs were indeed declined, and that facial recognition was working although the accuracy could be increased. They appreciated how buttons lit up green, showing "Signed In" after scanning attendance, giving them concrete evidence of successful sign in. They also thought the app needed little explanation as it all seemed very straight forward and easy to understand.

The feedback collected through both technical and nontechnical users played a crucial role in shaping StudentSignIn and will have a strong influence on future versions of the application.

## **Summary of Findings**

This chapter explored in-depth details of testing and results for the StudentSignIn application. Carefully detailing how testing was implemented through functional as well as non-functional testing to verify that the system was in accordance with its set specifications. The chapter explored several test cases for basic function like user signing in, retrieval of schedules, validity of QR codes, facial recognition, as well as tracking of attendance. Through detailed documentation of testing results, it was easy to determine where the app was succeeding as well as identify where improvement was needed. Findings found that the system could allow students to sign in, view their schedules, scan their codes, and verify their presence through facial recognition. Additionally, lecturers could view records of attendance, thus protecting the integrity of signing in processes. Findings from these tests validated that the system was user-friendly as well as dependable in its reliability of automatic tracking of attendance. Each of these defects found in testing processes like facial recognition errors as well the double scanning of QR codes was explored in depth to improve performance of the system.

## **Conclusion**

Through testing of functional as well as non-functional specifications went as planned. Findings confirmed that the application performed well in enabling student authentication, retrieval of timetables, scanning of QR codes, as well as facial recognition-based tracking of attendance. Additionally, users found it easy to view records of attendance, further affirming the system’s usability desired utility.

As much as results were good in general, some challenges were realized in specific aspects, namely in facial recognition reliability and how to handle expired QR codes. As it was, these challenges became apparent through systematic test cases to provide room for improvement and refinement. The system demonstrated good reliability in user authentication, preventing unauthorized entries while only allowing listed students to mark their attendance. Through performance testing, it was realized that the app was quick in responding to user actions while maintaining seamless user experience.

Overall, testing confirmed the utility of StudentSignIn as an automatic tracking of attendance solution. Although limitations in some respects were realized, the system was effective in delivering its basic objectives while giving room for future improvement. Findings from testing will guide further improvement of the application to provide a solid and secure platform for both students and lecturers.

# **Bibliography**

A. & Liliana, D. Y., 2021. Facial Biometric Identification in The Masked Face. IEEE.

Ashtopus Technologies, 2015. History of Biometric Technology Solutions, s.l.: Ashtopus Technologies.

Bala, R., Harnal, S. & Gupta, M., 2023. Teachers’ Perception about the Use of QR Code in Education. IEEE.

Banzon, A. M., Beever, J. & Taub, M., 2024. Facial Expression Recognition in Classrooms: Ethical Considerations and Proposed Guidelines for Affect Detection in Educational Settings. IEEE.

Bowyer, K., 2004. Face recognition technology: security versus privacy. IEEE.

BRADY, n.d. What is RFID ?. [Online]   
Available at: https://www.bradyindia.co.in/intelligent-manufacturing/what-is-rfid  
[Accessed 01 11 2024].

ByteScout, 2019. ByteScout. [Online]   
Available at: https://bytescout.com/blog/2019/04/qr-code-advantages-and-limitations.html#1

Chen, Y. & Li, X., 2021. Research and Development of Attendance Management System Based on Face Recognition and RFID Technology. IEEE.

Dangi, R., 2011. Engineers Garage. [Online]   
Available at: https://www.engineersgarage.com/face-recognition/  
[Accessed 01 11 2024].

European Data Protection Board, 2023. Guidelines 05/2022 on the use of facial recognition technology in the area of law enforcement. [Online]   
Available at: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.edpb.europa.eu/system/files/2023-05/edpb\_guidelines\_202304\_frtlawenforcement\_v2\_en.pdf  
[Accessed 18 10 2024].

Europe, C. o., 2023. Council of Europe. [Online]   
Available at: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://rm.coe.int/guidelines-facial-recognition-web-a5-2750-3427-6868-1/1680a31751

Faúndez-Zanuy, M., 2006. Biometric security technology, s.l.: IEEE Aerospace and Electronic Systems Magazine.

Gupta, O. P., Agrawal, A. P. & Pal, O., 2023. A study on Evolution of Facial Recognition Technology. IEEE.

Istvan, Z., 2020. IEEE Spectrum. [Online]   
Available at: https://spectrum.ieee.org/facing-up-to-facial-recognition  
[Accessed 30 10 2024].

Jha, A. et al., 2022. Facial Recognition Impact in Smartphone Sector. IEEE.

Kodali, R. K. & Hemadri, R. V., 2021. Attendance Management System. IEEE.

Li, L., Mu, X., Li, S. & Peng, H., 2020. A Review of Face Recognition Technology. IEEE.

Li, P. & Zhang, R., 2010. The evolution of biometrics. IEEE.

Li, X., 2015. Application of biometric identification technology for network security in the network and information era, which will greatly change the life-style of people. IEEE.

Meng, X., 2008. Study on the Model of E-Commerce Identity Authentication Based on Multi-biometric Features Identification. IEEE.

Nayak, A. et al., 2022. Biometric A Helping Hand in Talent Management: A Modern Time Tracking Tool. IEEE.

Paderes, R. E. O., 2015. A Comparative Review of Biometric Security Systems. IEEE.

Papaioannou, M. et al., 2023. Behavioral Biometrics for Mobile User Authentication: Benefits and Limitations. IEEE.

Pattnaik, I., Dev , A. & Mohapatra , A. K., 2023. Forensic Facial Recognition: Review and Challenges. s.l., Springer, Singapore.

Petrovic, N., 2024. TopSource Worldwide. [Online]   
Available at: https://topsourceworldwide.com/blog/8-benefits-of-leave-and-attendance-management-system/  
[Accessed 30 10 2024].

Pudyel, M. & Atay, M., 2023. An Exploratory Study of Masked Face Recognition with Machine Learning Algorithms. IEEE.

Qureshi, A., 2024. Jibble. [Online]   
Available at: https://www.jibble.io/article/types-of-attendance-systems#:~:text=There%20are%20several%20types%20of,%2C%20and%20GPS%2Dbased%20systems.  
[Accessed 29 10 2024].

Raghav, P., 2018. Medium. [Online]   
Available at: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148  
[Accessed 01 11 2024].

Rucevic, T., 2023 . Softworks. [Online]   
Available at: https://www.softworks.com/blog/what-is-time-and-attendance-management-software/  
[Accessed 29 10 2024].

R, V. C. et al., 2023. Face Recognition and Identification Using Deep Learning. IEEE.

Sahu, M. & Dash, R., 2020. Study on Face Recognition Techniques. IEEE.

Sajid, M., Hussain, R. & Usman, M., 2014. A conceptual model for automated attendance marking system using facial recognition. IEEE.

Schulfer, S., 2020. Sprout QR. [Online]   
Available at: https://www.sproutqr.com/blog/how-do-qr-codes-work  
[Accessed 01 11 2024].

Statista, 2015. Inflow of university students from EU-27, EEA and candidate countries to Ireland from 2005 to 2012. [Online]   
Available at: https://www.statista.com/statistics/434834/ireland-inflow-of-university-students-from-european-countries/

Stouffer, C., 2023. Norton. [Online]   
Available at: https://us.norton.com/blog/iot/how-facial-recognition-software-works#:~:text=Facial%20recognition%20uses%20technology%20and,but%20also%20raises%20privacy%20issues.  
[Accessed 30 10 2024].

Sultan, H. et al., 2022. Real Time Face Recognition Based Attendance System For University Classroom. IEEE.

Tiwari, S., 2016. An Introduction to QR Code Technology. IEEE.

Tretinjak, M. F., 2015. The implementation of QR codes in the educational process. IEEE.

Wagh, P., Thakare, R., Chaudhari, J. & Patil, S., 2015. Attendance system based on face recognition using eigen face and PCA algorithms. IEEE.

Wang, W., 2021. The development of face recognition in accuracy and speed: A Review. IEEE.

WAVE, D., n.d. Our Story. [Online]   
Available at: https://www.denso-wave.eu/en/company/our-history  
[Accessed 24 10 2024].

Xue, M. & Zhu, C., 2009. A Study and Application on Machine Learning of Artificial Intellligence. IEEE.

Yadav, V. & Bhole, G. P., 2019. Cloud Based Smart Attendance System for Educational Institutions. IEEE.

Zennayi, Y., Bourzeix, F. & Guennoun, Z., 2022. Analyzing the Scientific Evolution of Face Recognition Research and Its Prominent Subfields. IEEE.

Zhang, H. et al., 2019. Cloud-Based Class Attendance Record System. IEEE.

Zhang, H., Liu, W., Dong, L. & Wang, Y., 2014. Sparse eigenfaces analysis for recognition. IEEE.

Zhao, C. & Huang, X., 2020. Attendance System Based on Face Recognition and GPS Tracking and Positioning. IEEE.

Zorić, B., Dudjak, M., Bajer, D. & Martinović, G., 2019. Design and development of a smart attendance management system with Bluetooth low energy beacons. IEEE.