

UNIVERSITY OF MACEDONIA
GRADUATE PROGRAM
DEPARTMENT OF APPLIED INFORMATICS

**DEVELOPING A WEB BASED APPLICATION FOR STUDENT
ATTENDANCE MANAGEMENT**

M.Sc. Thesis

of

Ioannis Kapantzakis

Thessaloniki, November 2018

**DEVELOPING A WEB BASED APPLICATION FOR STUDENT
ATTENDANCE MANAGEMENT**

Ioannis Kapantzakis

B.Sc. in Economic Science, Aristotle University of Thessaloniki, 2006
M.Sc. in Informatics and Management, Aristotle University of Thessaloniki, 2008

M.Sc. Thesis

submitted as a partial fulfillment of the requirements for the award of the

DEGREE OF MASTER OF SCIENCE IN APPLIED INFORMATICS

Supervisor
Assistant Professor Konstantinos Vergidis

Approved by examining board on 02/11/2018

Assistant Professor
Konstantinos Vergidis

Professor
Alexandros Chatzigeorgiou

Professor
Christos Georgiadis

.....

.....

.....

Abstract

The vast majority of Higher Educational Institutions execute the process of attendance registration, as part of their regular operation. Although this task is not part of the actual teaching process, it is necessary for bureaucratic and organizational purposes. The attendance monitoring process, is a paper-based process, according to which, the professor takes a list of the students that are present at the time of the beginning of each lecture. This process requires an amount of time which is proportional to the number of the students who are present at the classroom, meaning that, the more students present at the classroom, the more amount of time is required for the attendance registration process to be completed.

This thesis, proposes an improved, automated framework, that aims to facilitate the attendance monitoring process, making it more effective, less time consuming and less error prone. We are going to study the already existing implementations of similar systems and try to design a new one, implementing innovative techniques, that require the less possible technical support, in terms of hardware and equipment. The proposed system is going to be easily accessed by the users, and is going to provide user friendly interface in order to facilitate its use. The aim of the proposed framework is to provide a new system that automates the attendance monitoring process, helping professors and students to achieve more effective time management during the courses.

Keywords

Attendance monitoring, attendance management, mobile application, web-based service, scan, QR code, authentication, authorization, geolocation

Acknowledgments

This thesis gave me the opportunity to study and learn new technologies in the field of web and mobile development. During the research and development period, I came across to complex problems and challenges, which made me put extra effort in order to find a solution. The development of such a multidimensional system, composed of different modules, using various frameworks, gave me the opportunity to gain a more complete overview of the web application frameworks' ecosystem.

I would like to thank my supervisor, Assistant Professor Kostas Vergidis for his valuable guidance throughout this period. He was, always, willing to provide the maximum assistance and indicate the right direction during the research period.

This thesis would have been impossible without the aid and support of my family. I am profoundly grateful to my beloved wife, Eftichia, for her indefinite patience and support, and, also to my wonderful daughter, Despoina, who is an inspiration for continuous effort and self-improvement.

Table of contents

1	INTRODUCTION	9
1.1	PAPER-BASED SYSTEM AND ITS LIMITATIONS	9
1.2	AIMS AND OBJECTIVES.....	10
1.3	METHODOLOGY.....	11
1.4	RESEARCH LAYOUT	12
2	LITERATURE REVIEW	14
2.1	METHODOLOGY OF LITERATURE REVIEW	14
2.2	TOOLS USED.....	15
2.3	TECHNICAL TERMS AND KEY WORDS	16
2.3.1	<i>Barcode</i>	16
2.3.2	<i>QR Code</i>	16
2.3.3	<i>Biometrics</i>	17
2.3.4	<i>GPS</i>	17
2.3.5	<i>GEO-fence</i>	17
2.3.6	<i>LBA</i>	18
2.3.7	<i>RFID</i>	19
2.3.8	<i>NFC</i>	19
2.3.9	<i>Beacon</i>	19
2.3.10	<i>WLAN</i>	20
2.4	RELATED WORK	20
2.4.1	<i>RFID technology</i>	20
2.4.2	<i>Smartphones and mobile applications</i>	21
2.4.3	<i>Biometrics</i>	24
2.4.4	<i>Other solutions</i>	26
2.5	REFERENCED ARTICLES OVERVIEW.....	27
2.6	COMMERCIAL APPLICATIONS	28
2.7	ISSUES AND CHALLENGES	30
2.8	SUMMARY	30
3	PROPOSED FRAMEWORK.....	32
3.1	THE TRADITIONAL PROCESS	32
3.1.1	<i>Participants</i>	32
3.1.2	<i>Process</i>	32
3.2	MANUAL REGISTRATION	34
3.3	IMPROVED SYSTEM.....	36

3.4 OVERVIEW	37
4 CURRENT PROCESS	42
4.1 THE BIG PICTURE	42
4.2 DEFINING THE PROCESS BOUNDARIES	44
4.3 RESOURCES.....	45
4.4 THE ATTENDANCE MONITORING PROCESS.....	45
4.4.1 A brief overview of the process	45
4.4.2 A step-by-step illustration	45
4.4.3 Diagrammatic representation.....	47
4.5 SUMMARY	51
5 AUTOMATED PROCESS	52
5.1 REQUIREMENTS	52
5.1.1 Automation	52
5.1.2 Security.....	52
5.1.3 Easy to use	52
5.1.4 Identity verification	52
5.1.5 Location verification.....	53
5.1.6 Independent of extra equipment	53
5.1.7 Minimum software dependences.....	53
5.1.8 Provide fallback mechanisms.....	53
5.2 PROPOSED PROCESS	53
5.3 BUSINESS PROCESS DIAGRAMS	54
5.3.1 BPMN	54
5.3.2 Use cases.....	56
5.4 SUMMARY	58
6 APPLICATION DEVELOPMENT	60
6.1 SYSTEM ARCHITECTURE	61
6.2 DEVELOPMENT STACK	62
6.2.1 Alternative solutions	62
6.2.2 The chosen stack	71
6.3 DATABASE SCHEMA	73
6.3.1 Tables	74
6.3.2 Relationships	82
6.3.3 Overview	84
6.4 APPLICATION PROGRAM INTERFACE (API)	85
6.4.1 Framework	86

6.4.2 Architecture.....	86
6.4.3 Authentication and authorization.....	87
6.4.4 Attendance logs validation	90
6.4.5 Testing.....	92
6.4.6 API documentation.....	93
6.5 WEB APPLICATION	97
6.5.1 Framework	98
6.5.2 Code sample.....	99
6.6 MOBILE APPLICATION	100
6.6.1 Framework	100
6.6.2 Code sample.....	101
6.7 TECHNICAL SPECIFICATIONS.....	101
6.7.1 Mobile app	102
6.7.2 Web application	102
6.7.3 API	102
6.7.4 Database.....	103
6.7.5 Overview	103
6.8 REPOSITORIES	104
6.9 SUMMARY	104
7 PROTOTYPE SOFTWARE	106
7.1 WEB APPLICATION	106
7.1.1 User professor	106
7.1.2 User student.....	110
7.2 MOBILE APPLICATION	112
7.2.1 Login.....	112
7.2.2 Main menu	114
7.2.3 Scan QR code.....	115
7.2.4 Enrollments overview.....	117
7.2.5 Registration data validation	118
7.3 FUTURE WORK.....	121
7.3.1 Configuration screens	122
7.3.2 Students' progress overview	122
7.3.3 Manual attendance registration	122
7.3.4 Users' data security.....	122
8 SUMMARY	123
8.1 MAIN REMARKS.....	123
8.2 LIMITATIONS OF THE FRAMEWORK.....	124

8.3 FUTURE WORK.....	125
8.4 CONCLUSION.....	125

List of figures

FIGURE 1-1: THESIS LAYOUT	13
FIGURE 2-1: THE THREE STAGES OF AN EFFECTIVE LITERATURE REVIEW PROCESS (YAIR LEVY AND TIMOTHY J. ELLIS, 2006)	14
FIGURE 2-2: BARCODE LABEL.....	16
FIGURE 2-3: QR CODE (HTTPS://EN.WIKIPEDIA.ORG/WIKI/QR_CODE)	16
FIGURE 2-4: CIRCULAR GEO-FENCE AREA	18
FIGURE 2-5: POLYGONAL GEO-FENCE AREA	18
FIGURE 2-6: LBS OPERATION OVERVIEW.....	19
FIGURE 2-7: A TYPICAL ATTENDANCE MONITORING SYSTEM USING RFID TECHNOLOGY	21
FIGURE 2-8: A TYPICAL QR CODE ENABLED ATTENDANCE MONITORING SYSTEM.....	24
FIGURE 2-9: TECHNOLOGY USED	28
FIGURE 2-10: ARTICLES PER TECHNOLOGY USED	28
FIGURE 3-1: THE TRADITIONAL PROCESS OF MANUAL ATTENDANCE REGISTRATION	35
FIGURE 3-2: TIME CONSUMED IN DIFFERENT NUMBER OF STUDENTS.....	35
FIGURE 3-3: PARALLEL ATTENDANCE REGISTRATION	36
FIGURE 3-4: PARALLEL VS SERIAL PROCESS	37
FIGURE 3-5: WEB APPLICATION - STATISTICS WIREFRAME	38
FIGURE 3-6: WEB APPLICATION - STUDENTS' LIST WIREFRAME	39
FIGURE 3-7: MOBILE APPLICATION - STATISTICS WIREFRAME	39
FIGURE 3-8: DEVELOPMENT STACK	40
FIGURE 3-9: API REPRESENTATION	41
FIGURE 4-1: EDUCATIONAL INSTITUTIONS' MAIN PROCESSES	44
FIGURE 4-2: ATTENDANCE MONITORING PROCESS (ABSTRACT DIAGRAM)	47
FIGURE 4-3: ATTENDANCE MONITORING PROCESS (INTERNAL DIAGRAM).....	49
FIGURE 4-4: PARTS OF INTERNAL DIAGRAM	50
FIGURE 4-5: PART (1) OF THE INTERNAL DIAGRAM.....	50
FIGURE 4-6: PART (2) OF THE INTERNAL DIAGRAM.....	51
FIGURE 4-7: PART (3) OF THE INTERNAL DIAGRAM.....	51
FIGURE 5-1: AUTOMATED ATTENDANCE MONITORING PROCESS	55
FIGURE 5-2: PROFESSOR USING THE WEB APPLICATION	56
FIGURE 5-3: STUDENT USING THE MOBILE APP	57
FIGURE 5-4: SECRETARIAL STAFF USING THE WEB APPLICATION.....	58
FIGURE 6-1: SYSTEM ARCHITECTURE AND DATA FLOW	62
FIGURE 6-2: TOP TEN PROGRAMMING LANGUAGES ACCORDING TO IEEE SPECTRUM (JULY 2017) (HTTPS://SPECTRUM.IEEE.ORG/COMPUTING/SOFTWARE/THE-2017-TOP-PROGRAMMING-LANGUAGES).....	63
FIGURE 6-3: TOP 15 MOST POPULAR PROGRAMMING LANGUAGES OF 2017 IN NUMBER OF PULL REQUESTS (GITHUB)...	64

FIGURE 6-4: TIOBI INDEX TIME LINE	65
FIGURE 6-5: TAG ACTIVITY PER WEEK (HTTP://HEWGILL.COM/~GREG/STACKOVERFLOW/STACK_OVERFLOW/TAGS/)	67
FIGURE 6-6: POPULAR FRAMEWORKS (HTTPS://HOTFRAMEWORKS.COM/)	69
FIGURE 6-7: DEVELOPMENT STACK	72
FIGURE 6-8: USERS TABLE	75
FIGURE 6-9: ROLES TABLE	75
FIGURE 6-10: USERROLES TABLE	76
FIGURE 6-11: COURSES TABLE	76
FIGURE 6-12: CLASSROOMS TABLE	77
FIGURE 6-13: ACADEMICTERMTYPES TABLE	78
FIGURE 6-14: ACADEMICTERMS TABLE	78
FIGURE 6-15: ATTENDANCETYPES TABLE	79
FIGURE 6-16: COURSEASSIGNMENTS TABLE	79
FIGURE 6-17: ENROLLMENTS TABLE	80
FIGURE 6-18: ATTENDANCELOG TABLE	81
FIGURE 6-19: LECTURESLOG TABLE	81
FIGURE 6-20: RELATIONSHIP BETWEEN [USERS] AND [ROLES] TABLES	82
FIGURE 6-21: RELATIONSHIPS DEFINED BY [ATTENDANCELOG] TABLE	83
FIGURE 6-22: RELATIONSHIPS DEFINED BY [ENROLLMENTS] TABLE	83
FIGURE 6-23: RELATIONSHIPS DEFINED BE THE [COURSEASSIGNMENTS] TABLE	84
FIGURE 6-24: DATABASE DIAGRAM	85
FIGURE 6-25: THE MVC PATTERN (SOURCE: DOCS.MICROSOFT.COM)	86
FIGURE 6-26: TYPICAL AUTHENTICATION AND AUTHORIZATION WORKFLOW	89
FIGURE 7-1: WEB APPLICATION - LOGIN SCREEN	106
FIGURE 7-2: WEB APPLICATION - PROFESSOR'S DASHBOARD	107
FIGURE 7-3: WEB APPLICATION - PROFESSOR'S DASHBOARD DETAIL	108
FIGURE 7-4: WEB APPLICATION - PROFESSOR'S STUDENTS LIST	108
FIGURE 7-5: WEB APPLICATION - QR CODE GENERATION SCREEN	109
FIGURE 7-6: WEB APPLICATION - COURSES ASSIGNED TO PROFESSOR	109
FIGURE 7-7: WEB APPLICATION - PROFESSOR SELECTED COURSE FOR QR CODE GENERATION	110
FIGURE 7-8: WEB APPLICATION - QR CODE GENERATED FOR SELECTED COURSE	110
FIGURE 7-9: WEB APPLICATION - STUDENT'S DASHBOARD	111
FIGURE 7-10: WEB APPLICATION - STUDENT'S DASHBOARD DETAIL	111
FIGURE 7-11: WEB APPLICATION - STUDENT'S ATTENDANCE LOGS OVERVIEW	112
FIGURE 7-12: MOBILE APP - LOGIN SCREEN	113
FIGURE 7-13: MOBILE APP - NO INTERNET CONNECTION MESSAGE	114
FIGURE 7-14: MOBILE APP - MAIN MENU	115
FIGURE 7-15: MOBILE APP - THE DEVICE ASKS FORM THE USER TO ENABLE THE GEOLOCATION FEATURE	116

FIGURE 7-16: MOBILE APP - THE SCANNING PROCESS	117
FIGURE 7-17: MOBILE APP - ENROLLMENTS OVERVIEW AND DETAILS	118
FIGURE 7-18: MOBILE APP - ERROR RAISED WHEN USER IS NOT ENROLLED TO THE SPECIFIC COURSE	119
FIGURE 7-19: MOBILE APP - ERROR RAISED WHEN USER'S DEVICE IS DETECTED TOO FAR FROM THE CLASSROOM	120
FIGURE 7-20: MOBILE APP - ERROR RAISED WHEN USER ATTEMPTS TO REGISTER ATTENDANCE AFTER SPECIFIC TIME LIMIT	121

List of tables

TABLE 2-1: COMMERCIAL ATTENDANCE MONITORING SOLUTIONS	30
TABLE 6-1: TEN MOST POPULAR PROGRAMMING LANGUAGES ACCORDING TO THE TIOBE INDEX FOR DECEMBER 2017 (WWW.TIOBE.COM)	65
TABLE 6-2: WEB PROGRAMMING LANGUAGES COMPARISON	66
TABLE 6-3: WEB FRAMEWORKS COMPARISON	68
TABLE 6-4: COLUMNS DATA TYPES.....	74
TABLE 6-5: ACADEMICTERMSCONTROLLER API	93
TABLE 6-6: ACADEMICTERMTYPES API	94
TABLE 6-7: ATTENDANCELOGCONTROLLER API	95
TABLE 6-8: ATTENDANCETYPESCONTROLLER API	95
TABLE 6-9: COURSESCONTROLLER API	95
TABLE 6-10: ENROLLMENTS API	96
TABLE 6-11: QRCODECONTROLLER API.....	96
TABLE 6-12: TOKENCONTROLLER API	96
TABLE 6-13: USERSCONTROLLER API	97
TABLE 6-14: SYSTEM MINIMUM REQUIREMENTS	104

1 INTRODUCTION

Educational institutions of all levels, from primary to higher education, perform a kind of attendance monitoring of the students. Regular attendance during the academic period facilitates the progress of students. Although a very important task, attendance monitoring should not distract students and professors from the actual goal of each lecture, the goal of learning. The traditional attendance monitoring system that the vast majority of educational institutions implement, is the manual registration of students' attendance by calling each student's name and marking his or her presence or absence from the lecture, in a list of names on a paper. It can easily be concluded that, as the number of students in a classroom increases, the more time is needed in order to complete the attendance registration of all students.

1.1 Paper-based system and its limitations

The traditional attendance registration system requires the attendance data to be collected with the use of paper lists, and can be categorized into two versions. According to the first version, the professor holds a printed list with students' names and calls each name in order to register the presence or absence of the specific student. According to the second version of the traditional, paper-based attendance registration system, the professor passes the printed names list to the students, in order for them to sign next to their name. One can easily conclude that this system poses some important drawbacks.

Referring to the first version of the paper-based attendance registration system, there is an important drawback: the professor consumes valuable time to this bureaucratic, but necessary, process.

Concerning the second version of the paper-based system, students can easily sign in behalf of absent students too, achieving a false registered attendance record, in case that the professor does not verify the registered presences against the number of total present students. A student could, inadvertently, sign in the wrong name, or not sign at all, a fact that leads to a false attendance registration too. Furthermore, students consume valuable time, not watching the lecture, in order to sign and pass the document to their classmates.

A common drawback for both versions, is the administrative cost arising from the nature of this data collection system. The attendance data are stored in physical form, on a piece of paper and need to be transformed into a digital form and be, appropriately, stored and

organized. This is a time and money consuming task, that could easily be reduced, or eliminated by an automated attendance monitoring system.

1.2 Aims and objectives

The aim of this thesis is to propose an effective attendance monitoring system that automates the attendance monitoring process, verifying students' identity and location, without the need of extra equipment implementation.

In more detail, this thesis intends to deliver a complete system that will facilitate the attendance data collection, verification, storage and organization within educational institution environments. This system will consist of an Android mobile application for the attendance registration operation, alongside with a web-based environment from which students and professor will have the ability to overview various statistics.

The following requirements must be fulfilled from the attendance monitoring system which this thesis is about to deliver:

1. *Identity verification*: the system has to be able to guarantee that the identity of the registered student is verified, for example, no student should be able to register an attendance on behalf of other student.
2. *Location verification*: the system has to be able to guarantee that the registered student's current location is verified, for example, each student must be inside the predefined area of the classroom at the time he or she conducts attendance registration.
3. *No extra equipment*: the system has to be able to operate without the need of purchasing and installing extra equipment, for example smart card readers or fingerprints scanners, apart from students' personal Android smartphones.
4. *Alternative attendance registration*: students that, for some reason, do not have Android smartphone devices, should be able to, manually, register their attendance.

The main objective of this thesis, as concluded from the aforementioned requirements, is the development of a system that will be easily deployed and, will facilitate the automation of the attendance monitoring process which, now, consumes valuable time and cost.

1.3 Methodology

The method that we are going to apply in order to achieve the aim of this thesis, that is, an easy deployable and easy to use system, that performs the attendance registration in an automated way, will be presented in this section. First, we will start by examining the related literature, mentioning the alternative systems, developed for the same reason. We will try to illustrate the solutions that each one of them propose, and conclude whether these solutions meet the requirements and objectives that we set in the previous section.

Having studied the available literature, we, subsequently, will try to present and analyze the traditional attendance monitoring process, in order to exclude useful conclusions related to the effectiveness of this process and try to find potential disadvantages. The aim of studying the traditional process of attendance registration, is to become more familiar with the process, in order to, more easily, spot the points where it can be improved.

As soon as we provide a good overview of the traditional attendance monitoring process, we will continue by proposing a new, automated process which will try to meet the objectives and requirements that we set for this thesis. We are going to, theoretically, describe the execution of the proposed attendance monitoring process, providing examples of how the proposed system is going to give solutions in areas where the traditional process seems to be unable to perform effectively.

Subsequently, we are going to provide a detailed overview of the proposed system. We will present the requirements of the system, explaining each one of them in detail. Then, we are going to explain the proposed system's architecture, providing useful diagrams, illustrating the way that it should be executed in various scenarios.

After defining the framework of the proposed system, we will provide an illustration of all the most popular and powerful development tools available for the development process. Having acquired a good understanding of the available technologies in the area of web and mobile development, we will present the selected development set of tools, illustrating the advantages and disadvantages of these tools against the alternatives. Finally, after the development process, we are going to illustrate the actual application, providing various screenshots for different use cases.

1.4 Research layout

In this section, we are going to depict the thesis layout, providing a brief overview of the contents of each chapter.

Chapter 1 introduces the examined topic, explaining the problem that we are trying to solve, and, presenting the aims and objectives of this thesis. Furthermore, it analyzes the methodology used in order to achieve the goal of this research.

Chapter 2 presents the study of the available literature, related to the field of attendance monitoring systems. In this chapter we get a good overview of other applications that try to provide an automated system for the attendance monitoring process.

Chapter 3 illustrates the traditional, paper-based process of attendance registration, explaining the various steps that need to be executed in order to complete the process. Weaknesses and disadvantages of the traditional process are emphasized in order to help us understand the need of a new, automated process. Subsequently, a new framework is proposed, that has the aim to solve most of the traditional process's disadvantages.

Chapter 4 presents an, in depth, illustration of the traditional attendance monitoring process, providing various diagrams in order to facilitate the understanding of the complexity of the traditional process.

Chapter 5 starts by defining the requirements of the proposed framework and continues with a detailed illustration of the automated attendance monitoring process, providing various diagrams and use cases.

Chapter 6 presents the development process of the new, automated system. First, we provide an overview of the most popular development tools and frameworks. Then, we present the selected set of tools and provide a detailed illustration of each system's module.

Chapter 7 presents the final deliverable, providing various screenshots of the system's modules, used in different scenarios. At the end of this chapter, we present various improvements that could be applied to the developed system.

Chapter 8 is a summary of this thesis. In this chapter, we try to present a brief overview of the examined topic, illustrating the most important parts of the research conducted.

Figure 1-1, provides an illustration of the thesis's structure, compiling the presentation made in this section.

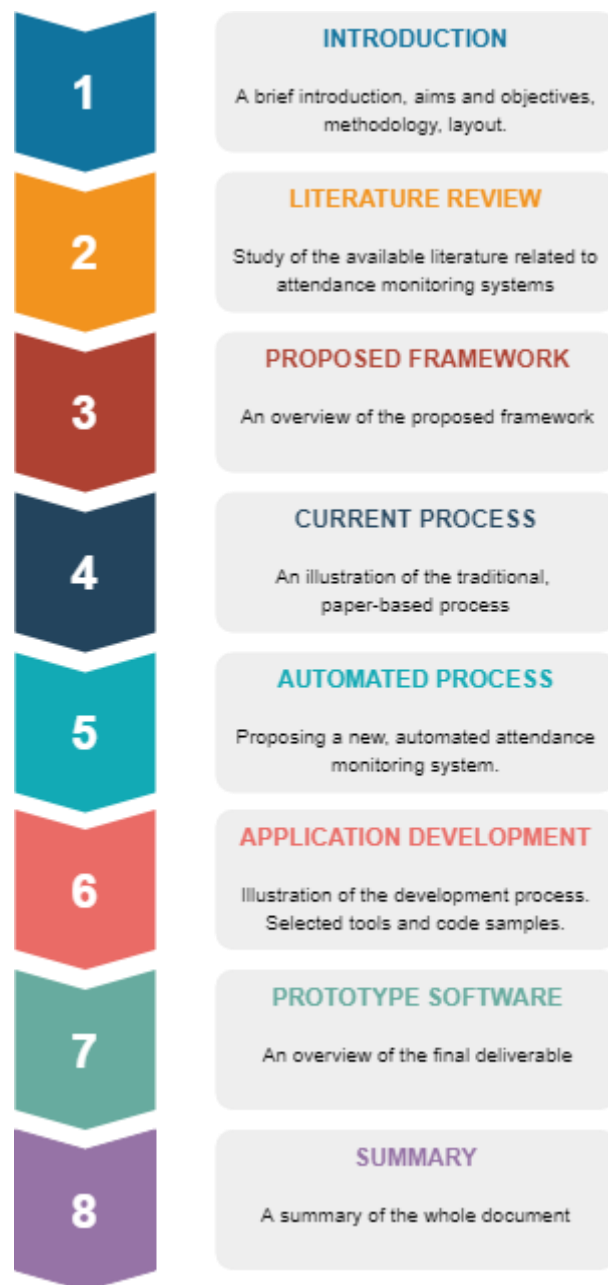


Figure 1-1: Thesis layout

2 LITERATURE REVIEW

In this chapter, we are going to present applications of various attendance monitoring systems implemented or proposed, until now. Before the actual review, we are going to present the methodology used in order to develop a complete literature review, as well as the tools used in order to organize the gathered literature more effectively. Subsequently, we are going to explain any technical terms or key words referred in the main section of the literature review. Having explained all the main concepts of the researched topic to the reader, we are going to proceed to the actual review of the existing literature.

2.1 Methodology of literature review

Yair Levy and Timothy J. Ellis (2006) proposed a three-stage process of conducting an effective literature review on an IS¹ topic. According to this proposal, a qualitative and effective literature review process should include four important characteristics: a) analyzation and combination of quality literature, b) provide a good understanding of the research topic, c) present a solid understanding of the selected research methodology as such as the reasons for selecting this methodology, and d) explain the ways that the proposed research contributes something new to the examined topic.

Figure 2-1 demonstrates the three stages of the literature review process that Yair Levy and Timothy J. Ellis proposed in 2006.

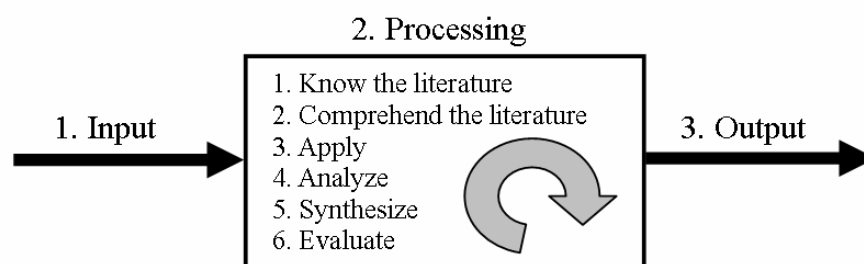


Figure 2-1: The three stages of an effective literature review process (Yair Levy and Timothy J. Ellis, 2006)

During the first step (*Input*), the researcher searches for quality literature that is relevant to the examined topic. It is important that the literature gathered comes from reliable sources and is not irrelevant to the research. For example, if the researcher searches for literature about a

¹ Information Systems

topic in corporate blogs, the findings are probably going to be not much objective. Except from that, the researcher must not limit the literature search in only one or two sources because the research may lead to one-sided conclusions.

The second step (*Processing*) aims at building a solid foundation that can be used from next researchers in order to build new research. During this step, the researcher must demonstrate a solid knowledge and comprehending of the literature that was gathered, extract the main concepts and apply the available literature to each of them. The researcher has to explain the importance of the illustrated information and combine the available literature in order to present a complete review. Finally, the researcher has to evaluate the literature and declare which of the referred citations constitute a theory, a case study, an opinion or something else.

The third, and last, step (*Output*) illustrates the literature review writing process. In order to conduct a quality writing result, Yair Levy and Timothy J. Ellis present two *argumentation models*², the first was proposed by Toulmin (1958) and the second was suggested by Williams and Colomn (2003). Further analysis of these argumentation models would be out of the scope of this research.

This methodology has been selected for the production of a complete literature review of this research because it is best applied (but not only) to IS topics research. Furthermore, a set of well-structured steps provided by this methodology facilitate the procedure of conducting an effective literature review.

2.2 Tools used

The organization of the gathered literature in a way that every document can be easily found is not an easy task. The traditional way of keeping a physical copy of each article found, raises a set of challenges. Should the documents be classified by the article title or by the article's author name? Should they be categorized by specific criteria? What if a document belongs to more than one category?

Thankfully, there are lots of software available that take care of these tricky issues. Zotero³ is the tool used for the organization of the literature of this research and it is freely available to anyone who wants to conduct such a task.

² A model referred to argumentation theory and helps the writer to develop a quality argumentation.

³ <https://www.zotero.org/>

2.3 Technical terms and key words

Before starting to present the literature related to the examined topic, we must provide a solid illustration of the technical terms referred, and give a definition for each one of them.

2.3.1 Barcode

The barcode is a binary code comprising a field of bars and gaps arranged in a parallel configuration (Finkenzeller, 2010, p. 3). Barcodes can be decoded by specifically designed machines called *barcode readers*. There are approximately ten different types of barcodes in use.



Figure 2-2: Barcode label

2.3.2 QR Code

As (DENSO, 2011) states, the QR code (Quick Response) Code is a two dimensional (2-D) matrix code that belongs to a larger set of machine-readable codes, all of which are often referred to as *barcodes*. DENSO WAVE⁴, a subsidiary of DENSO CORPORATION is the inventor of QR Code and holds a number of patents on QR Code. QR Code is freely available to anyone for any use. Figure 2-3 represents a simple QR Code label that, when scanned, provides the url for the English Wikipedia mobile main page.



Figure 2-3: QR Code (https://en.wikipedia.org/wiki/QR_code)

⁴ <https://www.denso-wave.com/en/>

2.3.3 Biometrics

Biometrics is defined as the science of counting and (body) measurement procedures involving living beings (Finkenzeller, 2010, p. 4). Biometrics can be used in order to identify individuals. As (Jain et al., 2000) state, biometric identification refers to identifying an individual based on his or her distinguishing physiological and/or behavioral characteristics (biometric identifiers).

In the following literature review, we are going to refer to some applications of biometric identification techniques in attendance monitoring systems. One of the mentioned above, is the *face recognition* technique, an identification procedure based on the analysis of face images. Another and very popular technique is the *fingerprints analysis*. Known for a long time now, fingerprints analysis is able to identify a person uniquely among others. *Iris recognition* is a technique of biometric identification based on the analysis of the iris of the human eye, the visual texture of which, stabilizes during the first two years of life and its complex structure carries very distinctive information useful for identification of individuals (Jain et al., 2000).

2.3.4 GPS

GPS stands for **Global Positioning System** and is a space-based radionavigation system owned by the United States Government (USG) and operated by the United States Air Force (USAF). It consists of space-based positioning, navigation, and timing (PNT) signals delivered free of direct user fees for peaceful civil, commercial, and scientific uses worldwide (USA Dept. Of Defense, 2008). Schiller and Voisard (2004, p. 10) describe the GPS as a satellite infrastructure serving the positioning of people and objects. The GPS system can be freely used from anyone who has a GPS receiver.

2.3.5 GEO-fence

Geo-fence can be described as a virtual fence or boundary which limits a specific geographic area using GPS or RFID systems. A geo-fence area can be a circle formatted with a specific radius around a specific point on the map, or a more complex shape defined by multiple points. Geo-fence areas could, also, be formatted by the combination of multiple Wi-Fi signals coverage. According to Sheth et al., (2009, p. 1), geofencing can be described as a Wi-Fi coverage to physical boundaries that are meaningful to users.

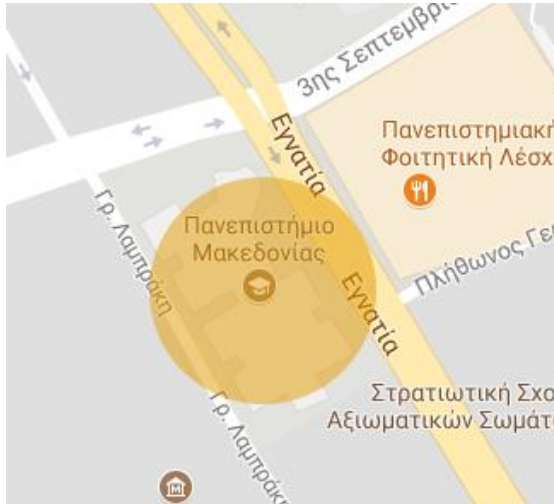


Figure 2-4: Circular geo-fence area

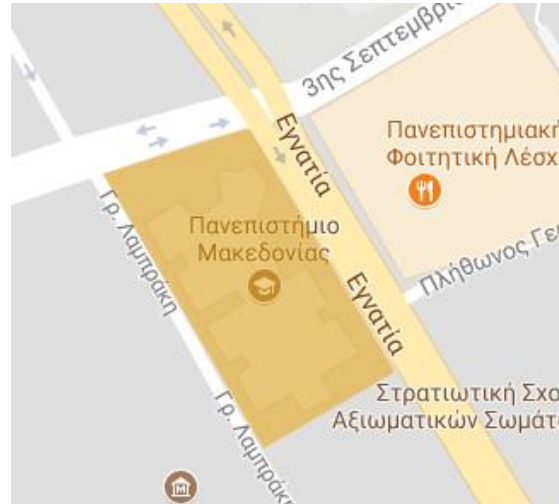


Figure 2-5: Polygonal geo-fence area

2.3.6 LBA

LBA stands for **L**ocation **B**ased **S**ervices and can be defined as services that integrate a mobile device's location or position with other information so as to provide added value to a user (Schiller and Voisard, 2004, p. 10). Virrantaus et al. (2001) states that location-based services can be defined as services utilizing the ability to dynamically determine and transmit the location of persons within a mobile network using their personal mobile device. In simple words, we can say that LBS consist of mobile applications that use the mobile devices ability to locate their current position in order to provide useful data to the user, for example, an application that finds the nearest gas station or another application that notifies the user when he or she reaches the boundaries of a pre-defined area on the map.

Figure 2-6 (Rajachandrasekar et al., 2010) represents a LBS operation. The LBA (Location Based Application) locates the current device position, executes a LBQ (Location Based Query) and receives a response with the requested data. The information (data) received from the same query for different locations are called LDD (Location Based Data). The service of getting LDD from a LBA using LBQ is a location based service.

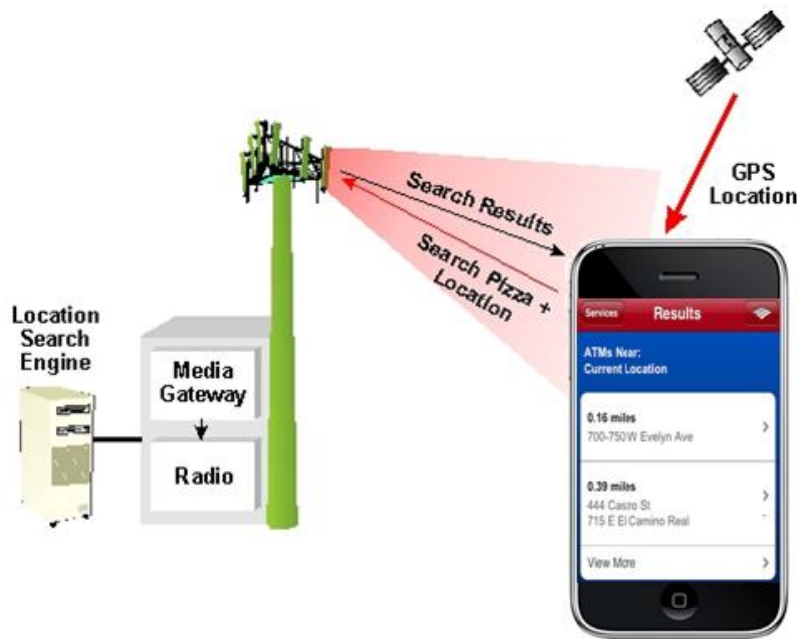


Figure 2-6: LBS operation overview

2.3.7 RFID

RFID stands for **R**adio **F**requency **I**dentification and refer to contactless ID systems (Finkenzeller, 2010, p. 1). ID systems are the systems that conduct an automated identification procedure (Auto-ID). RFID systems consist of a data-carrying device (transponder) and a reader. Unlike other systems (like smart cards), the power supply to the data-carrying device and the data exchange between the data-carrying device and the reader are achieved using magnetic or electromagnetic fields (Finkenzeller, 2010, p. 6).

2.3.8 NFC

NFC stand for **N**ear **F**ield **C**ommunication and can be described as a technology that permits the data exchange between two electronic devices in a short distance, for example, NFC technology allows contactless payments through credit cards. Data transmission between two NFC interfaces uses high-frequency magnetic alternating fields and the maximum communication range typical for NFC data transmission is 20 cm (Finkenzeller, 2010, p. 57).

2.3.9 Beacon

Beacons are small electronic devices that transmit radio signals to nearby mobile devices based on BLE (**B**luetooth **L**ow **E**nergy) technology. Beacons operate like Bluetooth devices but, they are designed to use much less energy and transmit less amount of data. Beacon

signals can be detected by Android, iPhone or Windows phones (for specific operating system versions) and are able to perform specific actions with received data.

2.3.10 WLAN

WLAN stands for **Wireless Local Area Network** and describes a local area network (LAN) that connects two or more devices using wireless connection within a specific area, for example a home or a company's network.

2.4 Related work

During the research of related literature, we have found that researchers have tried to develop effective attendance monitoring systems using various technologies like RFID, NFC, mobile phones, biometrics and many more.

2.4.1 RFID technology

Zatin Singhal and Rajneesh Kumar Gujral, (2012) have designed a remote attendance monitoring system based on RFID and GSM network. The system consists of a PC on which a RFID reader is connected. The student must own a unique card that RFID reader can read in order to record the attendance. The software that receives the data from the RFID reader is written in VB language and stores the data to a MS-Access database. As soon as an attendance gets recorded, the system uses the GSM network to send a notification via SMS (Short Message Service) to the student's or/and to the student's parents mobile devices (There is no need of internet connection).

Rajan Patel et al., (2012) also proposed an attendance monitoring system using RFID technology. The system requires the installation of a RFID reader to every class that is connected to the institute's LAN (Local Area Network). The RFID reader in each class will detect every student's personal card and pass the received data to a specific middleware (through LAN) in order to monitor the attendance of each student.

Sumita Nainan et al., (2013) and Arulogun et al., (2013) proposed similar attendance monitoring systems which include RFID tags (cards) and RFID readers. The students record their attendance while entering the class, using installed RFID readers and the data get transmitted to specific software and get stored to a database.

Parvathy et al., (2013) suggested a RFID enabled system that makes use of cloud environment. Students record their attendance using their personal card and a web-based

environment gets updated displaying appropriate data, for example, the professor can view the students name that are absent, students can view their attendance history for each period etc.

Figure 2-7 illustrates a typical attendance monitoring system that used RFID technology as described in presented literature. It is true that systems using RFID and NFC technologies facilitate the attendance monitoring procedure, offering a high degree of reliability, but, they require extra equipment. For example, every student must have a unique card that a special device can read in order to monitor the attendance. Furthermore, there is a possibility of one or more students can record the attendance of other students just by holding their personal cards.

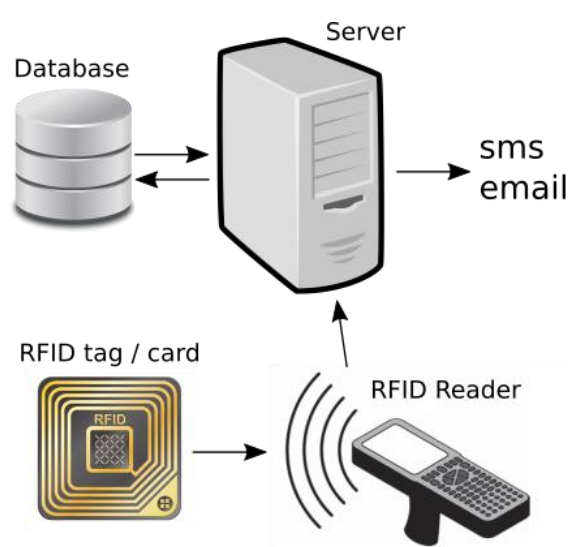


Figure 2-7: A typical attendance monitoring system using RFID technology

2.4.2 Smartphones and mobile applications

Many researchers have tried to develop effective attendance monitoring systems using RFID, or similar, technologies. These systems achieve an effective location and identity verification but require the installation of extra hardware, like RFID readers or scanners. To solve this restriction, in combination with the rapid evolution of smartphones, many researchers started to design and develop attendance monitoring systems, based on the use of mobile applications.

Shraddha S. Chawhan et al., (2013) proposed a simple mobile based attendance system according to which, the professor enters the system's Android application and checks the list of students that are present. As we can see, the only difference from the traditional attendance monitoring system is that there is no paperwork as the data are stored digitally. Therefore,

there is no time saving compared to the traditional procedure because the professor has to manually check for each student's presence.

Gautam Shanbhag et al., (2014) described a mobile based attendance marking system using biometrics. According to this proposal, the professor logs in to the system's Android application inserts his or her credentials, initializing a new session. A notification is sent to the students' mobile devices asking them to log in and insert their credentials. After a successful login, each student has to provide their fingerprints to a biometric fingerprint scanner that matches the provided fingerprints with the stored fingerprints images from the system's database. This solution uses modern technology (Android application) that everyone has access to, but requires the manual process of fingerprints matching and extra equipment (fingerprints scanner), to verify the actual person that gets recorded.

Akshay A. Kumbhar et al., (2014) proposed a simple attendance monitoring system that operates on Android devices. The process described in this research starts with the professor logging in to the application, using his credentials, and activating a new session. Now, the students are able to log in to the application from their devices and register their attendance. After a reasonable amount of time, the professor deactivates the attendance monitoring session, so as the students are not able to register any attendance anymore. Even if this proposal does not require any extra equipment, like RFID reader or scanners, it, also, does not provide an identity verification because a student could register multiple attendances for many students if they gave him / her their mobile devices and credentials.

Fadi Masalha and Nael Hirzallah, (2014) describe a three-stage attendance monitoring system that uses QR code scanning via mobile devices. According to this system, the professor logs in to the system and generates a QR code that displays to the students. The students log in to the system's mobile application from their smartphones, the application captures a picture of their face and prompts them to scan the displayed QR code. Subsequently, the application sends the scanned QR code to the server, alongside with the captured image of the student's face and the current location of the mobile device (GPS). This is a three-stage process because it verifies the identity of the student (face image), the current location (GPS service) and the lecture details (QR code). This is a complete attendance monitoring solution which guarantees the identity and location verification.

Dae-Jea Cho and Mi-Young Bae, (2014) proposed an attendance monitoring system using mobile application and QR code technology. According to the proposed process, the professor initializes the generation of a unique QR code and displays it to the students. The students, in their turn, scan the QR code with their smart phones' camera, registering their attendance.

This process eliminates the need of installing extra equipment like in RFID enabled systems but does not guarantee the physical presence of the registered student because there is a possibility that someone can send an image of the generated QR code to other students that are not present in class at that time.

Dae-jea Cho and Young-wook Cha, (2015) presented an improved version of the previous QR code enabled attendance monitoring system, including the use of Beacon technology to verify the students' current location. According to this approach, students scan the generated QR code and receive Beacon signals at the same time, verifying their physical presence in the classroom. This system can guarantee that the registered students are actually present in the class at the time of attendance check, but require the purchase of extra hardware (Beacon transmitters) and poses some software specifications restrictions for the students' smartphones (iOS 7.0 or higher / Android 4.3 or higher).

Miran Hikmat Mohammed Baban, (2015) discusses the operation of an attendance checking system that uses QR code scanning, one time at the beginning of each session and one time at the end of each session, to prevent students from registering their attendance at the beginning of the session and then leave the classroom after a while.

Deugo, (2015) proposed a QR code enabled attendance checking system that requires from the professor to scan each student's personal QR code which he or she (student) has generated from the system's web page. More specifically, students have to generate their personal QR codes from the system's website, before the session starts and print it or download it to their smartphones. On session start, the professor has to login to specific mobile application using his or her credentials and start scanning each student's QR code in order to register their attendance. This system can verify the actual students' presence, but requires a lot of time, especially in classes with many students.

Figure 2-8 represents a typical attendance monitoring system using smartphones and QR code technology, as described in many researches presented so far. The instructor (professor) initializes the process of QR code generation. The generated QR code is displayed to the students who, in their turn, scan using their smartphones' camera and send the data (QR code, location etc.) to the server. The server validates the received data and updates the database with the collected attendance data.

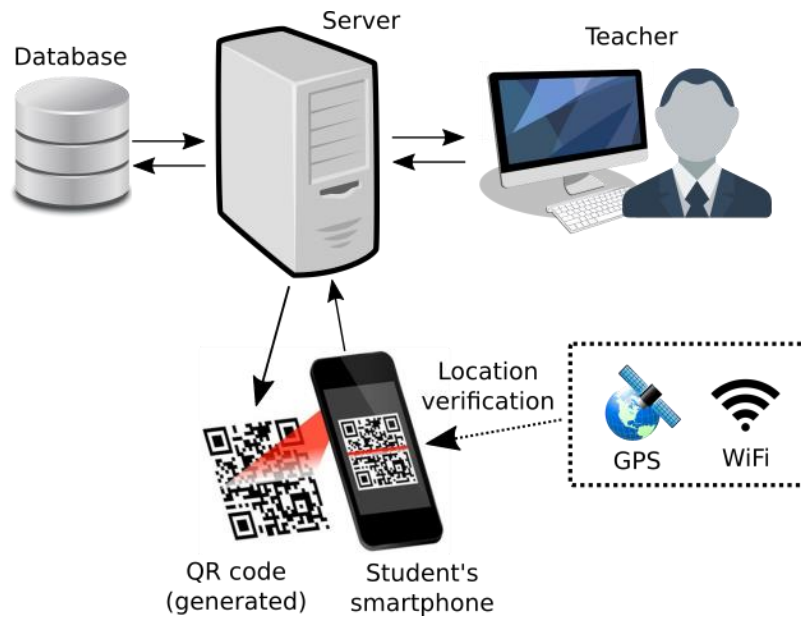


Figure 2-8: A typical QR code enabled attendance monitoring system

Sultana et al., (2015) dealt with the development of a location based attendance management system for employees. According to this approach, each employee has to install a specific Android application to his or her mobile device and login. At first login, the user has to provide the location (latitude, longitude and radius) of the working area and save the internet connection details of the office. Mobile device's location service has to be activated in order for the application to detect the user's current location. After this initial setup, every time the employee enters or leaves the predefined working area, the application notifies the system's server about the attendance details (employee id, enter or exit and timestamp).

Swapnali Pawar et al., (2016) conducted a research about attendance monitoring system for employees that uses Android application and biometrics and could be applied to educational institutes too. The research states that the employees use their mobile phone camera to capture an image of their fingerprints that is going to be compared with already stored fingerprint image in system's database. After successful matching, the application finds the current position of the mobile device using GPS and, if the location is inside acceptable boundaries on the map, the system registers employee's attendance. This system provides a complete attendance monitoring system with identity and location verification and does not require extra equipment, like RFID readers or fingerprints scanners, in order to operate.

2.4.3 Biometrics

Biometrics enabled systems can provide a high level of identity verification because they check unique characteristics of each person that cannot be replicated by others, for example

fingerprints or iris recognition. A plethora of researches, presenting attendance systems with biometrics verification, exist.

Tabassam Nawaz et al., (2009) presented a simple approach on attendance monitoring system that uses fingerprints verification. According to this application, a special fingerprints scanner is placed right next to the classroom door, and each student has to place his or her fingerprints on the scanner before entering the class. Subsequently, the system matches the provided fingerprints with previously stored fingerprints samples and mark the students' attendance. We can easily conclude that this system might get a bit slow for classes with large number of students because they must get in a line in order to provide their fingerprints before entering the class.

Kadry and Smaili, (2010) proposed a wireless attendance management system that uses iris recognition technology. Iris recognition refers to the biometric identification of a person by examining his or her iris and is based on the hypothesis that the probability of finding two people with identical iris pattern is almost zero (Tisse et al., 2002). The research indicates that the students must place their eyes in front of an eye scan sensor in order to send an image of their iris to the server in order to match received iris images with previously stored iris images. A new attendance gets marked as soon as a received iris image matches to a stored iris image. The need of purchasing and installing extra equipment in order to conduct iris recognition verification, makes this proposal ineffective compared to other systems that can operate without special equipment, for example, systems using mobile devices.

O. Shoewu and O. A. Idowu, (2012) described a simple attendance management system that uses fingerprints verification. During the enrollment, each student has to provide some data, such as name, gender, date of birth etc. alongside with his or her fingerprints to get stored in the system's database. On each lecture, students place their fingers to special fingerprints reader in order to record their attendance. The reader sends the captured fingerprints image to the server in order to compare with stored fingerprints images. If a match is found, a new record gets marked to the database. Although no doubt arises related to students' identity verification, this system requires some time, which get increased as the number of students in classroom increases, to complete the attendance monitoring procedure. Apart from that, this system requires the purchase and installation of specialized equipment for the fingerprints capturing process.

Thomas Yeboah et al., (2013) proposed an almost identical attendance monitoring system using fingerprints verification. The steps are the same, as the students have to provide their fingerprints in order for the system to find a match from the stored fingerprints samples.

Ajinkya Patil and Mrudang Shukla, (2014) proposed an attendance monitoring system that uses face recognition verification. According to this system, a camera captures an image of all students and send it to the server. Specific software executes two processing steps, first the *face detection*, in which, tries to detect how many face patterns exist in the image, and second, the *face recognition*, when it tries to match detected faces with face images stored in the database. For each detected face that gets matched with a face image stored in the database, a new attendance gets marked on the database. This novel approach presents some sophisticated techniques of face detection and recognition but is subjected to some significant restrictions, for example, this system gets less effective as the classroom dimensions increase because the camera cannot cover the whole audience in a big classroom. Another problem would be the way that the students are placed in the classroom, for example, a student may be sited behind another student in a way that the camera cannot capture his or her face in the captured image. Finally, this system requires the installation of extra equipment inside each classroom (camera).

2.4.4 Other solutions

Min Choi et al., (2014) described a sophisticated attendance check system that uses Wi-Fi networks, supporting unlimited number of concurrent connections. In fact, the term *connections*, is not accurate because the main idea of this system is that there is no actual connection between the client (student) and the AP (Access Point) (professor). According to this architecture, the professor's mobile device must have a specific application installed. Once logged in, the application sets the professor's device to act as an AP with specific range (classroom boundaries). The students' devices that are close enough to the professor's device (AP) are able to detect (not connect) the AP's SSID⁵ and can prove that are close enough to the professor (AP). This approach can be applied to bigger classrooms where RFID or Bluetooth technologies can not cover with only one access points and has the advantage that clients' (students') devices do not have to establish a connection with the professor's AP, but just to detect the AP's SSID.

Samuel A. Daramola et al., (2016) proposed an online signature verification system for student attendance monitoring. This system uses ANN (Artificial Neural Network) and the Levenberg-Marquardt back-propagation training algorithm. Students insert their personal information like name, sex, college, department etc., alongside with their signature which they draw on digital signature pads. The system collected signature samples from a population of fifty students to train the ANN. The results produced a FAR (False Acceptance

⁵ **S**ervice **S**et **I**dentifier is the name of the wireless local-area network.

Rate) of 8% and a FRR (False Rejection Rate) of 12%. The results could be improved by collecting more data in order to achieve a better training rate of the ANN, and by using more technologically advanced digital signature readers. Although very sophisticated, the signature verification system requires from the students to manually register their attendance by signing, not to a paper, but to a digital signature pad.

Makhtar et al., (2016) proposed an attendance management application that uses the GEO-fence technique. This system was proposed for employees' attendance management, but can easily be applied to students' attendance monitoring systems. According to this approach, the employee's smartphone will be GPS enabled in order to detect the current location and calculate if this location is valid, according to predefined map boundaries stored in the system's database. We can see that this system provides a location verification but does not guarantee that the device which registers the attendance, actually belongs to the employee that gets registered by this device.

2.5 Referenced articles overview

Here, we present an overview of the articles referenced in the *related work* section of the literature review. Figure 2-9 shows the number of articles for each technology that these articles propose. Smartphones seem to be the more referenced technology in articles found with ten articles referencing. Biometrics can be found in five articles as their main technology proposed. RFID technology is referenced in four articles and, finally three more articles present other kind of technology.

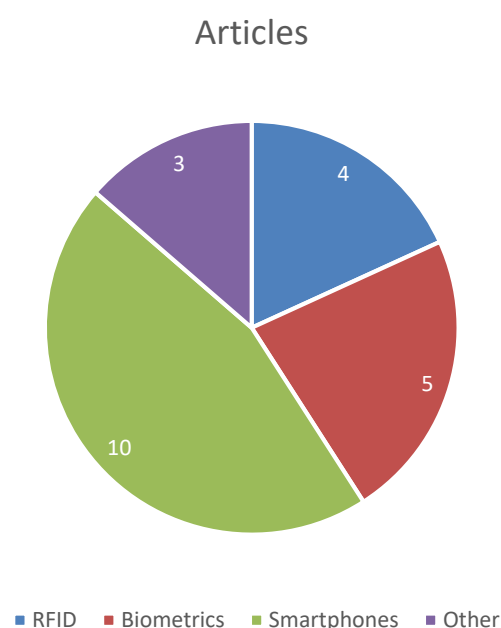


Figure 2-9: Technology used

Figure 2-10 presents the technology that was proposed in each article in more detail. Note that more than one technology may be proposed by one article. Here we can see that *mobile applications* were the solution proposed from more articles (10). *Fingerprints* and *QR-code* are referenced by 5 articles each. *RFID* technology is found in 4 articles and *GPS* is referenced by 2 articles. Finally, other technologies, such as *on-line signature*, *Wi-Fi detection*, *GEO-fence*, *iris recognition*, *GSM* and *Beacons* are proposed by 1 article each.

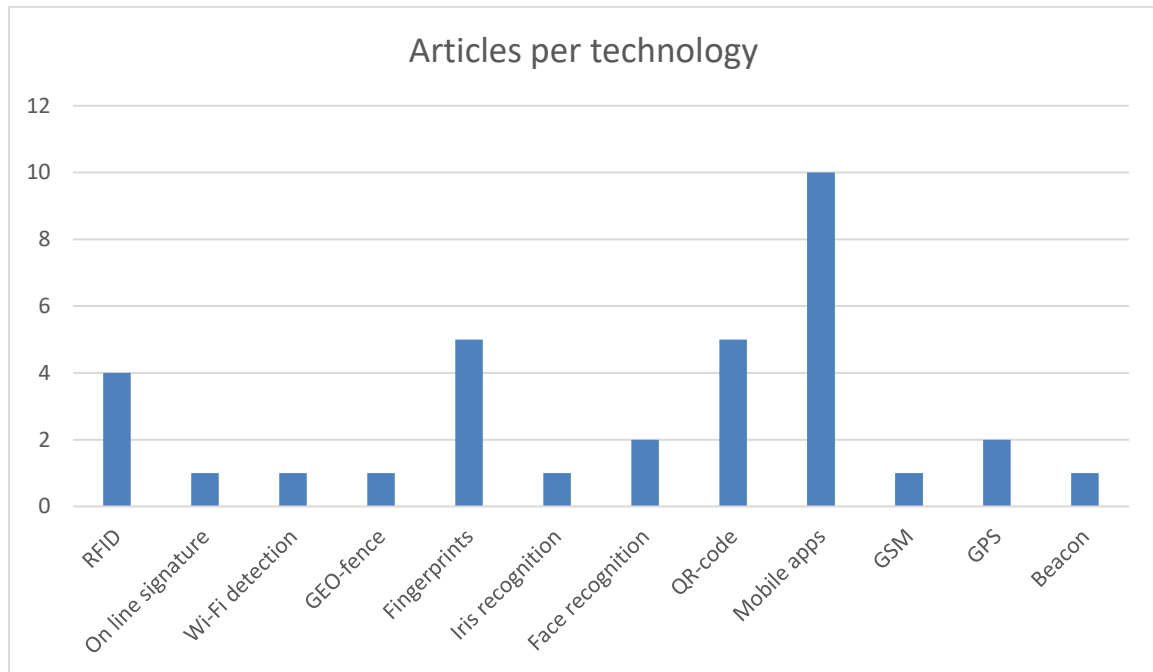


Figure 2-10: Articles per technology used

2.6 Commercial applications

During the literature review research, we have found some commercial applications that offer attendance monitoring solutions using various technologies like smart cards, biometrics, location-based services, Beacons etc.

*TDS Student*⁶ by TDS company is a student's management system that offers attendance monitoring capabilities. This product offers a web portal for students, in order to facilitate their academic life, offering an overall image of their progress. The attendance monitoring system uses smart cards for students and special hardware (card readers) for reading those cards data. Another commercial product that offers attendance management solutions is

⁶ <https://timedatasecurity.com/product/tds-student/>

*Creatrix Campus*⁷. This product uses Biometrics and RFID technology to register verified attendance data. Simac's *Presto*⁸ offers a set of smart cards and card readers, alongside with a manual attendance (digital) list. *BioStore*⁹ is another attendance registration solution that uses smart cards and/or biometrics (fingerprints) identity verification to monitor the attendance. Solutionpath developed a location-based attendance monitoring solution called *SATS*¹⁰ (Student Attendance Tracking Service) that requires no extra infrastructure such as card readers or fingerprints scanners. *Onyx Advantage*¹¹ and *Onyx Collector*¹² are similar attendance monitoring systems from Telepen company. Both systems provide a smart card-based verification, except that Onyx Advantage offers a web-based user interface for overall data analysis. Finally, *campusM*¹³ is an attendance monitoring system that uses Beacons to register students' attendance.

Table 2-1 shows the presented commercial attendance monitoring solutions and the technology used for each one of them. We can see that four of the seven (57%) make use of smart cards and card readers, two of seven (29%) use biometrics and one (14%) uses location-based techniques and one uses Beacons signals.

<i>Product</i>	<i>Smart cards / Biometrics Card readers</i>	<i>Location- based</i>	<i>Beacons</i>
<i>TDS student</i>	✓		
<i>Creatrix</i>		✓	
<i>Presto</i>	✓		
<i>BioStore</i>	✓	✓	

⁷ <https://www.creatrixcampus.com/attendance-management-system>

⁸ <https://www.simac.com/en/type-oplossingen/presto-attendance-monitoring-secondary-further-education>

⁹ <https://biostore.co.uk/solutions/identity-management/attendance-registration/>

¹⁰ <http://www.solutionpath.co.uk/attendance-monitoring-cpd>

¹¹ <http://telepen.co.uk/onyx-advantage/>

¹² <http://telepen.co.uk/onyx-collector/>

¹³ <http://www.campusm.com/platform/attendance-monitoring/>

<i>SATS</i>	✓
<i>Onyx</i>	✓
<i>CampusM</i>	✓

Table 2-1: Commercial attendance monitoring solutions

2.7 Issues and challenges

As presented above, it is clear that many researchers have tried to design, develop and apply effective attendance monitoring systems for academic institutions. Most of the presented researches propose systems that require the installation of extra equipment, such as RFID readers, fingerprints scanners, iris recognition hardware etc., in order to ensure the identity verification of the registered student. Other researches presented attendance monitoring systems that make use of mobile devices in order to eliminate the need for purchasing and installing extra equipment, verifying the location of the smartphone by the use of mobile device's GPS, but not verifying the identity of the person carrying the registered device. The challenge, and goal of this research, is the development of an attendance monitoring system that does not require the use of extra hardware and guarantees the identity and current location of the registered student.

2.8 Summary

This chapter dealt with the examination of the existing literature related to attendance monitoring systems for educational institutes. Initially, the methodology used in order to conduct the literature review, was presented. The next section of the chapter illustrated some important technical terms and key words that are heavily referred to the researches presented to the next section. Furthermore, the tools used in order to organize the articles that were found during the research, was presented in a next section. After familiarizing the reader with the technical concepts of the articles to be presented, and the tools used in order to organize them, the fourth section of this chapter dealt with the actual literature review, presenting the relative articles found, organized in sub-chapters, grouped by the main technology used in each article, and in chronological order. Subsequently, a brief review of the articles cited in the *related work* section was presented, with the help of tables and diagrams. After that, a number of commercial products offering attendance monitoring solutions for academic

environments, developed by various companies, were presented. Finally, this chapter concludes that the goal of this research is to come up with an effective attendance monitoring system that guarantees the identity and location verification of the registered students, without the need of installing extra equipment.

3 PROPOSED FRAMEWORK

In this chapter, we are going to present the methodology that will be followed in order to produce a theoretic description of a more effective attendance monitoring process. At first, we are going to describe the traditional, paper-based, attendance monitoring process. Subsequently, we will try to visualize the process by presenting a graphical representation (model) of the process. After modeling the process, we are going to analyze it and try to address specific changes in order to optimize the process, ending up with an improved version of the initially presented business process.

3.1 The traditional process

Here, we are going to describe and analyze the traditional, paper-based attendance monitoring process, explaining every participant's action in this process.

3.1.1 Participants

The participants taking place in the attendance monitoring process are:

- Professor: the professor is responsible for collecting the attendance data of every lecture.
- Students: they are obligated to attend the lectures and register their attendance.
- Secretarial staff: auxiliary staff that facilitates the overall process.

3.1.2 Process

The students' attendance monitoring process takes place in the classroom area, every day a lecture is scheduled. In order for the process to be able to begin, the professor has to have printed the list of students for the specific lecture. Having the list of students, the professor initializes the attendance monitoring process at the beginning of the lecture. There are two possible ways that the attendance monitoring process can be conducted, the first one requires that the professor passes the paper list to the students in order for them to sign next to their name, and the second way, requires that the professor calls each student's name and marks their attendance if they are present.

3.1.2.1 *Passing the list to the students*

According to this version of the process, the steps followed are these:

1. The professor passes the list to the first student
2. The student searches for his or her name in the list
3. If the name is found, the student signs next to his or her name
4. If the name is not found, the student notifies the professor in order to investigate the problem
5. If the professor has access to valid students' data, verifies that the specific student has successfully enrolled to the current lecture
6. If the student has been successfully verified, the professor marks the student's attendance
7. If the student has not been successfully verified, or the professor does not have any access to a valid students' list, the professor notes the student's name in order to verify his or her enrolment after the lecture
8. The student passes the list to the next student and the procedure continues until the list reaches the professor again
9. As soon as all students have signed the list, the professor counts the signed names and compares the number of the names signed with the number of present students at that time
10. If the number of the signatures is the same with the number of present students, the procedure ends
11. If the number of the signatures is bigger than the present students, it means that someone has registered an attendance on behalf of an absent student
12. If the number of the signatures is smaller than the present students, it means that someone forgot to register his or her attendance

3.1.2.2 Professor holds the list

If the professor chooses to hold the list and call each student's name, the process goes as follows:

1. The professor calls the first name of the list
2. If someone raises his or her hand, the professor marks the specific student's attendance
3. If no one raises his or her hand, the professor continues with the next name
4. When the professor reaches the end of the list, he or she asks the audience if everyone listened to their name
5. If one or more students inform the professor that they did not listen to their names, the professor tries to verify their enrolment (steps 5 to 7 of the previous version)
6. If everyone listened to their name, the professor counts the names marked as present and compares this number with the actual present students
7. If the number of marked names is equal to the number of present students, the process ends
8. If the number of marked names is greater than the number of present students, it means that the professor, mistakenly, marked one or more students' names that are absent

3.2 Manual registration

We are, now, going to depict the traditional attendance monitoring process and we will try to extract useful information about how to convert this process to a more effective one.

Figure 3-1 presents the process, according to which, the professor starts the attendance monitoring process by passing the students' list to the first student (Student 1). Student 1, marks his, or her attendance and passes the list to the next student (Student 2), and so on, until the list reaches the last student (Student N), who returns the list to the professor. The main disadvantage of this kind of process is that it is time consuming. The fact that only one student at a time can register his, or her attendance, makes this process serial, meaning that, student N has to wait for student N-1 to finish with his, or her attendance registration, in order to register his, or her own.

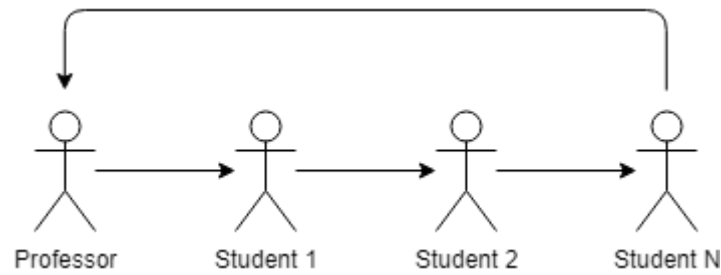


Figure 3-1: The traditional process of manual attendance registration

The fact is that the time consumed in order for the manual process of attendance monitoring to be completed, is proportional to the number of students in the classroom. Assuming that a student consumes an average of 15 seconds in order to find his, or her name on the list and mark his, or her attendance, and pass the list to the next student, it would have taken 7.5 minutes in a class of 30 students, plus, about 3 minutes for the professor to check the list, that is 10.5 minutes in total. Of course, this might not be considered as a huge amount of time spent, but, imagine a college classroom of 100 students. Figure 3-2, illustrates the amount of time spent for the manual attendance monitoring process to get completed, in different sizes of classrooms.

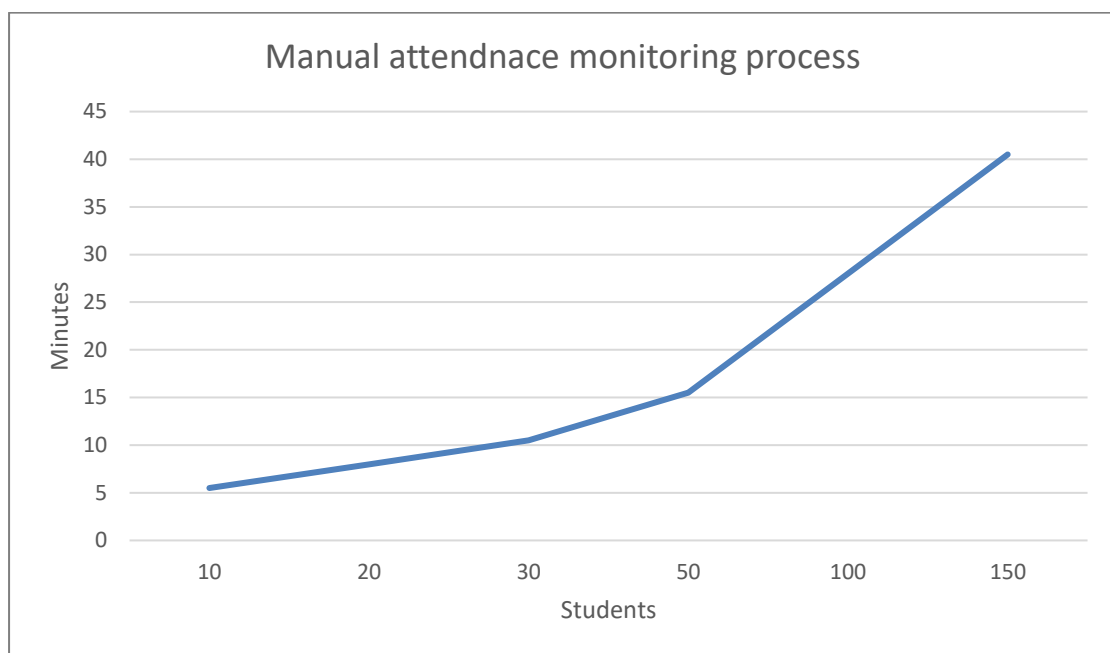


Figure 3-2: Time consumed in different number of students

Apart from the time spent in order for the traditional attendance registration to be completed, there are, also, other points in this process that can be improved, for example, the physical documents that get stacked and have to be organized and manipulated by the secretarial staff. This is a time consuming and error prone process, due to the injection of the human factor in

the process. In contrast to manual processes, automated processes, do not require any action by any human, in order to organize the incoming information. This means that, an automated process can not make a mistake, if it is programmed correctly.

3.3 Improved system

We saw above, that the main disadvantage of the manual attendance monitoring process is that it is a time-consuming process, especially in classrooms with a big number of students. What if we could implement another process that would not be serial, but parallel, meaning that all students could, simultaneously, register their attendance, as shown in figure 3-3.

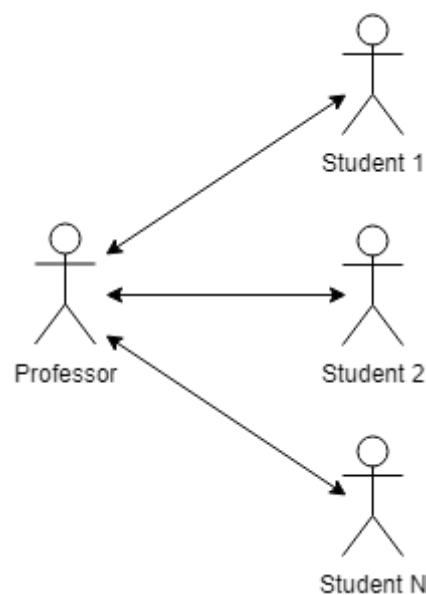


Figure 3-3: Parallel attendance registration

If we achieve to develop a process that works that way, we would, now have spent the same amount of time, for the attendance registration of all students in a classroom, in a classroom of 10 student, as in a classroom of 100 students, because, all students would have started the attendance registration process at the same time, and complete it, at about the same time also.

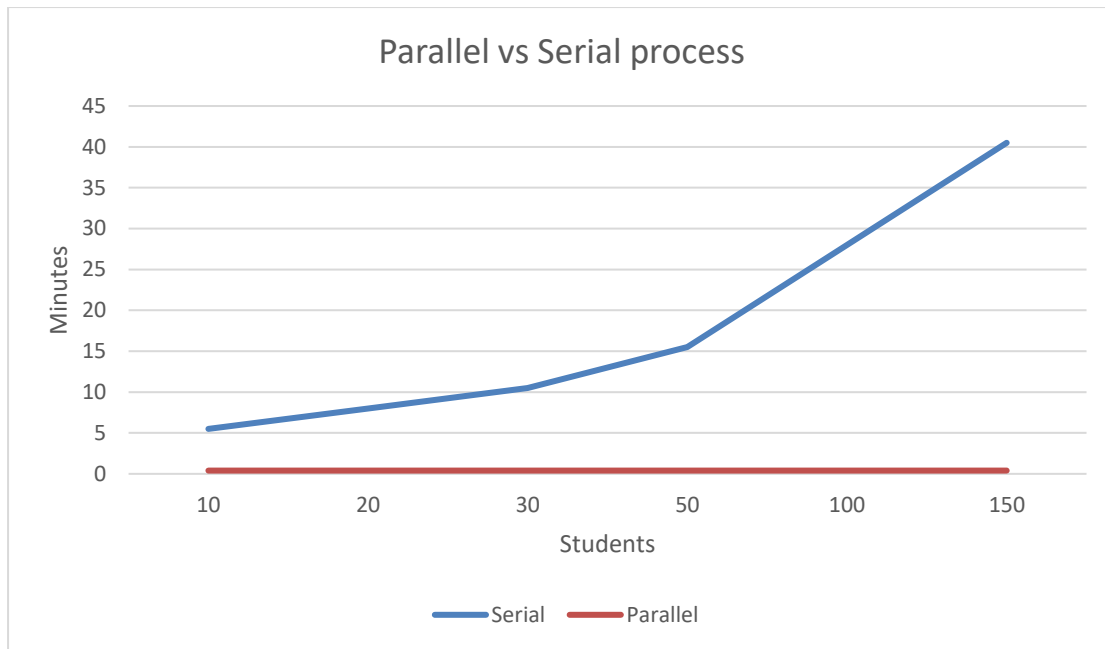


Figure 3-4: Parallel vs Serial process

Figure 3-4, compares the two kind of processes, parallel versus serial process. We see that, as the number of students grows, the amount of time spent, in the serial process increases linearly, but, in the case of the parallel process, the amount of time remains constant.

3.4 Overview

We have seen that the traditional process of attendance monitoring is a time consuming and error prone process. Developing a system that, facilitates students to, simultaneously, register their attendance, and, automatically, stores the information will make the attendance monitoring process much more efficient.

In this section, we are going to explain the design and development process of the proposed framework, and give a brief illustration of the automated system, providing some mockups.

Our new, automated system, is designed to be composed by three main modules, the *Web Application*, the *Mobile Application* and the *API*. Without providing much details, we are going to explain the role of each one of the automated system's modules. The *Web Application*, will be a simple, easy-to-use application, that will provide valuable information to the end user, such as, various statistics about the lectures and the students, lists of courses, students, enrollments and many more. The *Mobile Application* is indented to be used, be the students, and, its main feature is the attendance registration feature. The *API* is the data

provider for each one of the first two modules. The web application, as well as the mobile application will communicate with the API, receiving the data from the system's database.

Figure 3-5 depicts a wireframe of the web application statistics screen, from which the professor might be able to have a quick overview of the students' state, or, a student might overview his, or her own statistics for a specific course.

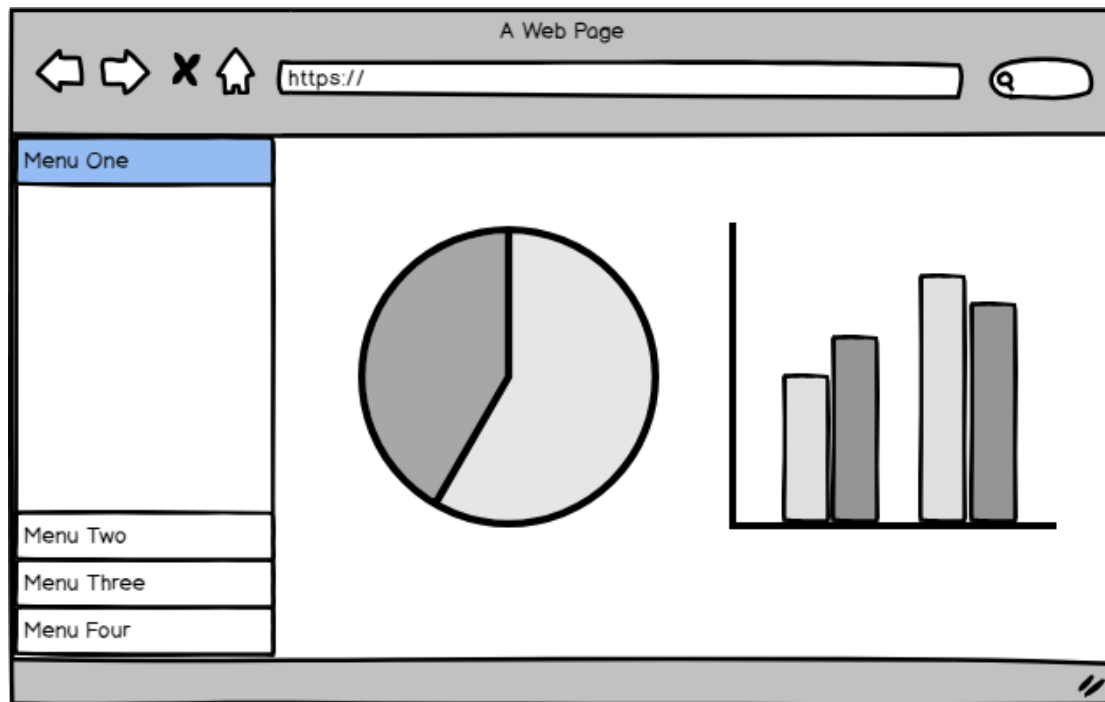


Figure 3-5: Web application - Statistics wireframe

Another view that the web application could provide to the final user, is a list of useful information, as shown in figure 3-6.

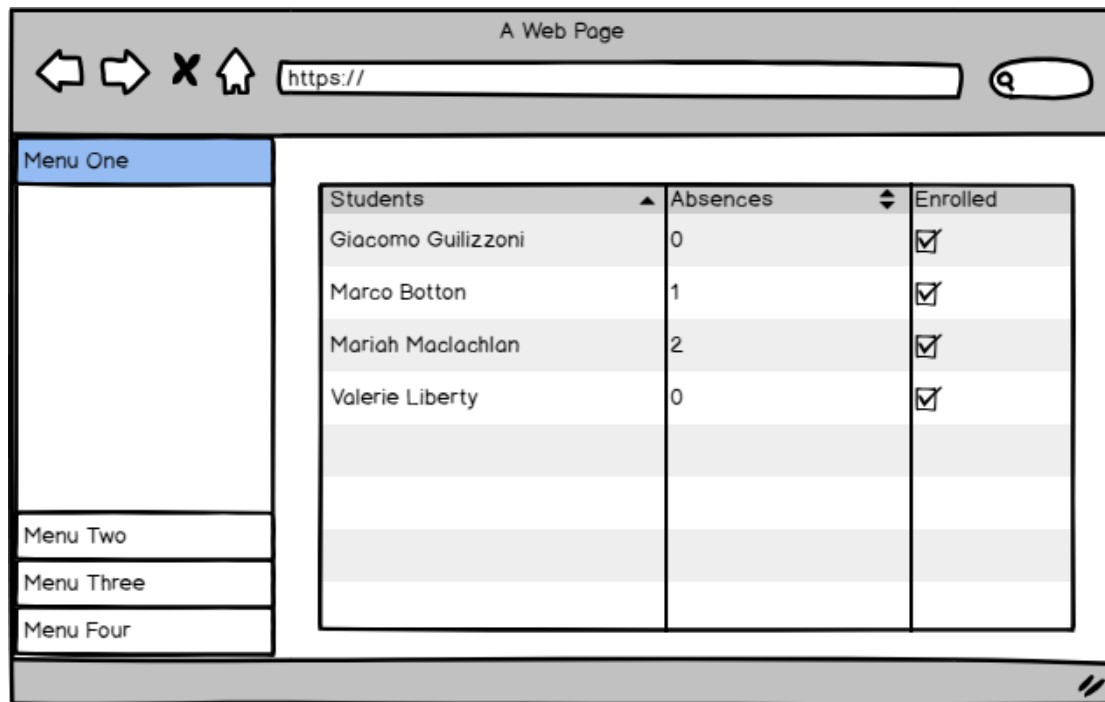


Figure 3-6: Web application - Students' list wireframe

The mobile application would not have much different user interface, and would display more or less the same information as the web application. Figure 3-7 depicts the statistics view, rendered in a mobile device.

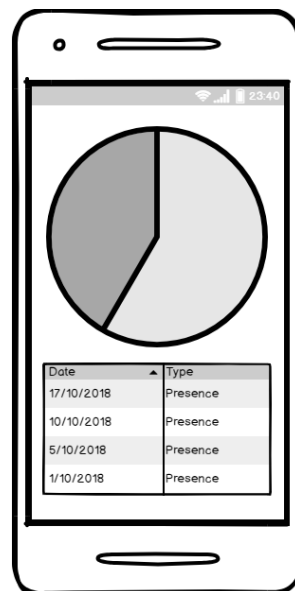


Figure 3-7: Mobile application - Statistics wireframe

Before starting the actual development of the new system, we are going to provide an in-depth analysis of each of the traditional and the proposed, automated processes. We are going

to illustrate each process, step by step, and provide informative diagrams that are going to help us to locate the traditional process's weaknesses and design a new, effective system.

After having a deep understanding of both processes, we are going to start designing the system's architecture. We should decide which technologies we are going to use in order to develop our system. First, we will present lists of the most popular and powerful tools available for each part of the development process and then we will try to pick a tool from each one of these lists, in order to compose an effective and powerful development stack, as shown in figure 3-8.

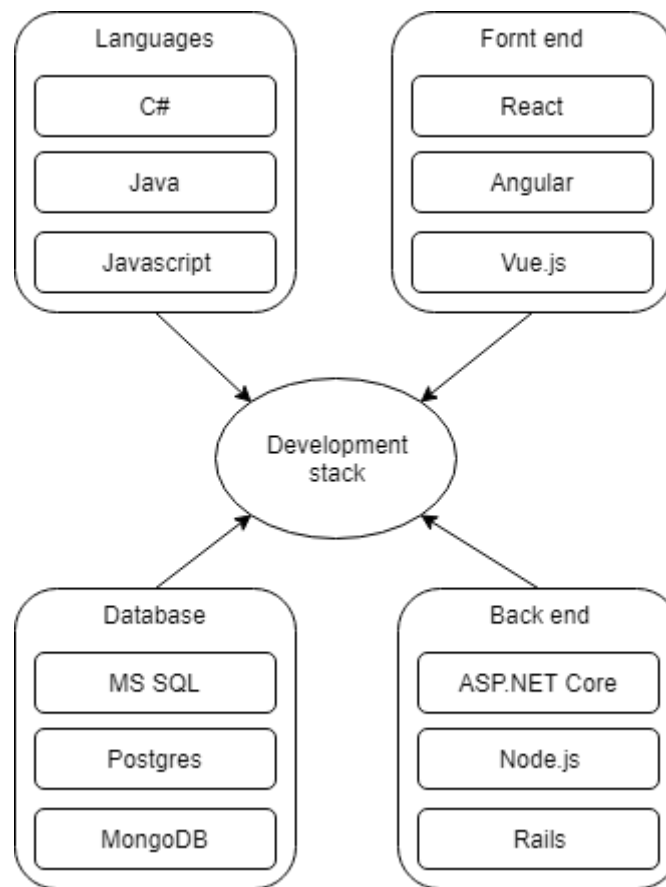


Figure 3-8: Development stack

Having selected a development stack, we can start coding in order to develop a new, automated system. As soon as we have a working version of our software, we can start documenting each one of its composing parts. We will start with the database, and more specific, with the database schema. After explaining the way, we designed and developed the database, we will proceed with the API documentation. Here, we are going to mention each one of the API's exposed calls and their return types, as shown in figure3-9.

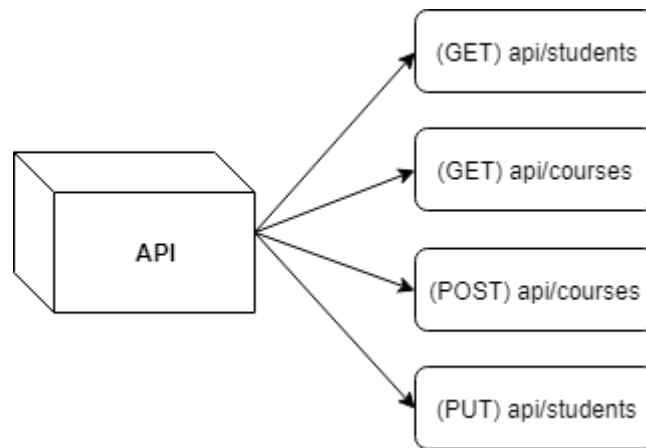


Figure 3-9: API representation

After the API presentation, we are going to refer to the web application's development process. We will present the selected development framework and we are going to provide some code samples in order to provide a better understanding of the development process. Next, we will illustrate the development process of the mobile application, presenting the selected mobile development framework and providing various code samples. Having explained the development process of the new automated system, we will mention the main technical requirements (specifications) that have to be met in order for the system to function correctly.

At a next stage, we will present the actual user interface of, both the web and the mobile application. We will provide various screenshots from real world scenarios, for example, a student's attendance registration, or a professor's overview of his, or her assigned courses and many more. We will, also, provide use cases of errors raised due to invalid data received by the application, during the registration attempt. Finally, we will try to detect and propose potential new features that would provide extra functionality to the application.

4 CURRENT PROCESS

In this chapter, we are going to describe the currently followed process of students' attendance monitoring. We are going to present each participant's role and their activities during this process. Before we start explaining the attendance monitoring process, we have to, clearly, distinguish the specific process boundaries against the wider educational institution's operation. By the end of this chapter, the reader should be able to understand the examined process and, easily, isolate it from the general group of the educational institution processes.

4.1 The big picture

In general, all educational institutions perform various processes, during their operation, that are similar across the most institutions. A university or a college operates like any organization (non-profit or not) and performs some basic procedures. The most basic and essential process is the process of *administration*. Every enterprise, organization or institute, from a multinational group of companies to a small, family business, has to be managed by one or more people. Thus, an educational institute has to be managed by someone. Some of the administrative processes that an educational institute may perform are:

- Taking decisions relative to the distribution of the available financial resources of the institution.
- Scheduling long term educational programs plan.
- Taking decisions relative to new investment plans.

Another, very important process that an educational institute performs, is the human resources management, known as *HR*. A HR department may deal with the following situations:

- Recruitment of new employees for the company.
- Training of the current personnel in order to develop their skills.
- Ensures that employees' needs get heard by the administration.

Apart from the aforementioned, *public relations* is an important process that many educational institutions, and companies, dedicate a serious amount of resources to them, in order to communicate with the rest of the world (the public). According to

<http://www.ipr.org.uk/>, through public relations, a company, or educational institution, tries to create a positive image and create a strong relationship with the audience. An educational institution could perform any of the following actions in order to promote their image to the public:

- Organizing physical events in order to achieve a two-way communication between the institution and the potential students
- Using social media interaction or blog posts in order to provide updated content to the audience
- Organizing online courses, available for free to anyone

Every company, thus an educational institution, has an *Information Technology* (IT) department that is responsible for all technological issues that arise during their operation. Some of the most important responsibilities of an IT department are:

- Store, organize, manage and secure all company data (employee's personal data, clients' data etc.)
- Provide communication (telephony, intranet, internet) inside the company
- Technical support across the company for any hardware or network failure

Last but not least, an educational institution performs *academic work*, the most important and the aim of any educational institution. The academic process includes the attendance monitoring operation as part of the students' education process. The following image illustrates the most important processes that any educational institution performs during its operation.

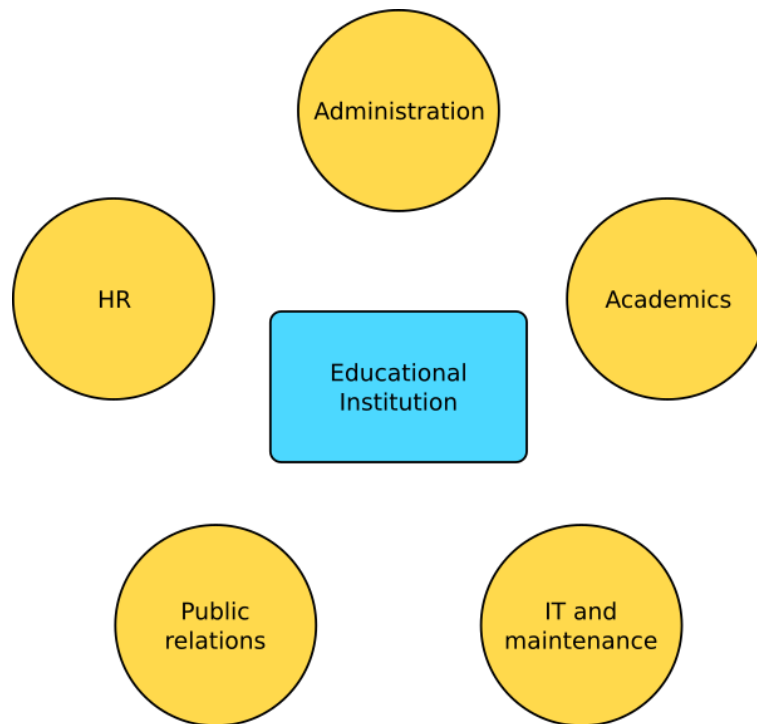


Figure 4-1: Educational institutions' main processes

4.2 Defining the process boundaries

Now that we are familiar with the general operational flow of a typical educational institution, we must set the limits of the examined process (students' attendance monitoring), within the overall process. The attendance monitoring process exists as part of the *Academics* category that we defined in the previous section. At this point we can clearly separate the examined process from the other four main groups of processes (Administration, HR, Public relations and IT).

The general process of *Academics*, except from attendance motoring process, contains other sub-processes too. In order to define more accurately the limits of the examined process within the more general process of *Academics*, we will provide the following set of assumptions:

- There is already an existing database that holds data related to courses, students, professors and classrooms
- Students' enrollment process has been successfully completed

4.3 Resources

During the attendance monitoring process, some people execute specific actions in order to achieve specific goal. These people are going to be referred as *resources*. Regarding the examined process, these are the main resources participating:

- **Professor:** The professor is the person responsible for conducting the main part of the attendance monitoring process.
- **Student:** The student holds a passive role to the whole process, because the only action he/she takes is to declare his/her presence to the classroom.
- **Secretarial stuff:** Represents the person (one or more) who is responsible for bureaucratic procedures related to the attendance monitoring process. From now on, this resource is going to be referred to as the *secretary*

4.4 The attendance monitoring process

At this point we can assume that we have accurately defined the attendance monitoring process boundaries. In this section we are going to present a brief overview of the current attendance monitoring process and, after that, we are going to continue with a step-by-step description of the process, providing related diagrams.

4.4.1 A brief overview of the process

Currently, the majority of educational institutions follow the traditional attendance monitoring process using printed lists of students' names. The professor is responsible for conducting a manual process of attendance monitoring, calling each student's name and waiting for a response in order to mark the student's presence or absence. Moreover, the list has to be processed by the secretarial stuff in order to update the institution's database with the newly gathered attendance data for this day.

4.4.2 A step-by-step illustration

The process starts at the point where the secretary prepares the list of students who are enrolled to the specific course. The secretary has to, manually, search for the students' details (name, registration number, semester, etc.), create a document that contains the students' details ordered alphabetically, or by registration number or by any other way and, finally, print the list.

The professor has to get the students' list before the session start, and this implies that the professor has to spent an amount of time in order to find the secretary's office and get the list. It could, also be possible that the professor gets the students' list by email. In that case, the professor has to spent an amount of time in order to print the list. In most cases, the list of students for a specific course stays unchanged over the semester, so the professor could have prepared a number of copies so that he, or she, does not have to try to get the students' list before each session. In this case, unpredictable events, like the enrollment of a new student in the middle of the semester, or the interruption of the course from a student, forces the professor to change the list and print it again.

The next step starts with the professor calling each student's name from the list. The professor, searches for the first name of the list and calls it in order to get a response from the student called. If the student is present at that time, he or she indicates his or her presence in the classroom and the professor marks the called name in the list as present. If the professor gets no response during the next seconds, he or she marks the student's name as absent. This step is repeated for each student's name in the professor's list, until the last name. This step is the most time consuming because it gets repeated for every student and is proportional to the total number of enrolled students at the specific course. The total amount of time consumed during this step of the process is affected from other factors, like the size and the acoustic of the classroom and even from the willing of the students to cooperate with the professor.

As soon as the professor marks the last name of the list, he or she checks if all the students have heard their name. If there is any one that did not hear his or her name, the professor notes the name, or names, on the list and continues with the lecture. This step, also demands an amount of time to be consumed.

After the lecture ends, the professor has to deliver the students' names list to the secretarial stuff. Now it's time for the secretarial stuff to, manually, update the institution's database with the data from the students' list. The secretary searches each student's name into the database and updates the attendance field with a value that declares presence or absence, depending on the value marked by the professor on the list.

When the secretary completes this process, he or she, checks if the professor has noted any student as physically present at the classroom but, his or her name was not included in the list. If such name exists, the secretary has to find out whether the list is outdated or if this student was actually not enrolled to this course. The secretary starts by querying the database in order to get the list of students enrolled to this course for this semester. Then, the secretary compares the database results with the printed list and verifies that the list is correct. In case

that the list is different from the database results and the noted name exists in the database, the secretary updates the student's attendance field and corrects the list for the next lecture. There is also a possibility that the student has successfully got enrolled to the course, but the database is not yet updated. In this case, the secretary has to manually insert the student into the database and, subsequently, update the student's attendance field for this course. In case that the student has actually not been successfully enrolled to this course, but has the right to enroll, the secretary will try to contact the student in order to find out whether the student wants to enroll or not. If the student wants to enroll, the secretary inserts the student's data into the database and updates the student's attendance field. This is the point where the attendance monitoring process ends.

4.4.3 Diagrammatic representation

In this section, we are going to illustrate the attendance monitoring process, as described in the previous section, providing various diagrams. These diagrams have been developed using the BPMN¹⁴ modeling methodology.

The first diagram illustrates the attendance monitoring process with less detail in order to provide a brief overview of the whole process. This diagram consists of two lanes, each of which represents a part of the educational institution that executes specific tasks. The first (green) lane is the secretarial stuff and the other lane (blue) represents the class (professor and students). Such non-detailed diagram is called an *abstract diagram* and is depicted in figure 4.2.

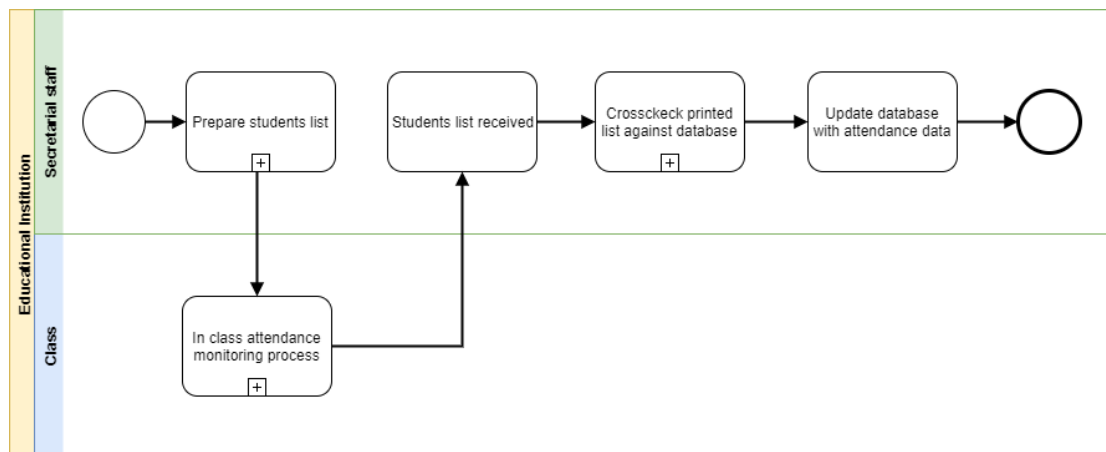


Figure 4-2: Attendance monitoring process (Abstract diagram)

¹⁴ <http://www.bpmn.org/>

In order to provide a diagram that illustrates the whole process with more detail, we can decompose the abstract diagram's processes into sub-processes ending up with a much more detailed diagram. Such diagram is called an *internal diagram*. In figure 4.3 we provide an internal diagram of the attendance monitoring process. This diagram depicts three lanes, the first one (green) is the same as the abstract's diagram first lane, the secretarial stuff. The other two lanes result from splitting the class lane of the abstract diagram class lane into two sub-lanes, the professor (blue) and the student (orange).

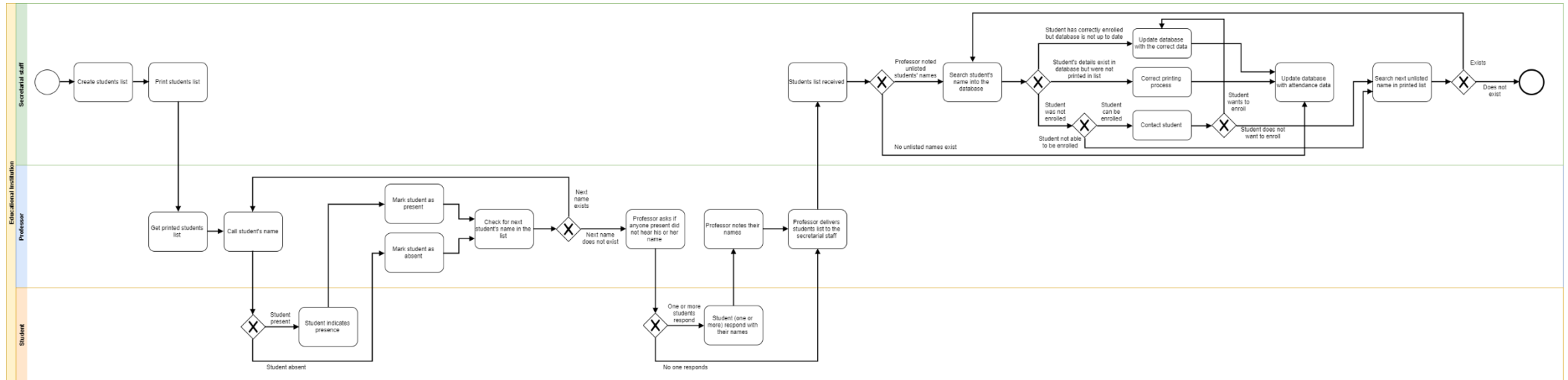


Figure 4-3: Attendance monitoring process (Internal diagram)

For the reason that the internal diagram does not fit well into a landscape document page, we are going to provide a set of figures representing parts of the internal diagram. Figure 4.4 indicates the three parts of the internal diagram that are going to be, separately, depicted.

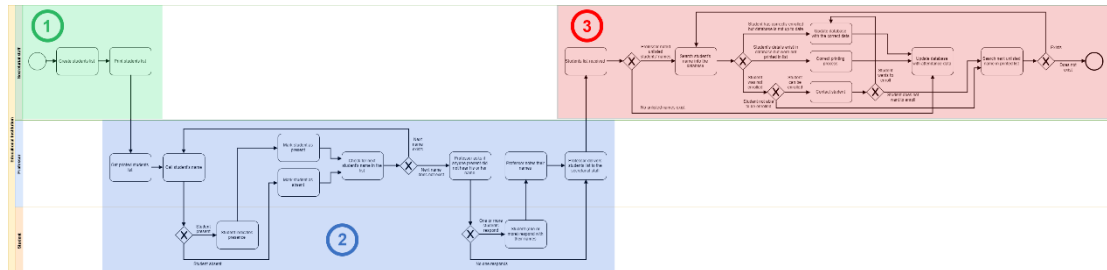


Figure 4-4: Parts of internal diagram

The first section of the internal diagram, numbered by number 1 and colored in green, includes the first steps of the attendance monitoring process, executed by the secretarial stuff and is shown in figure 4.5.

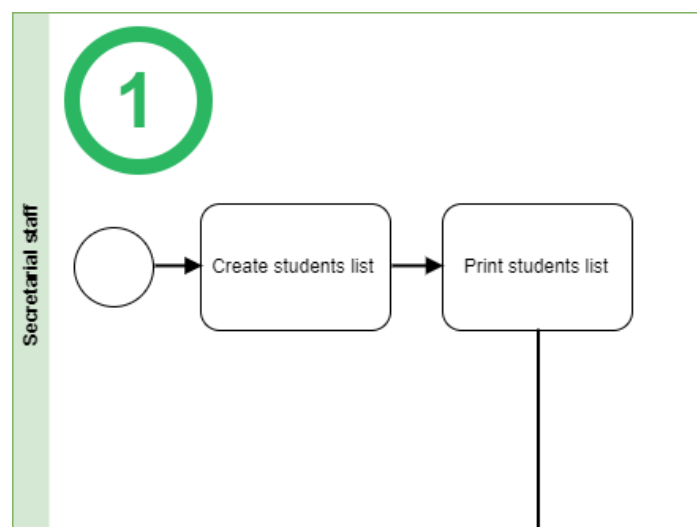


Figure 4-5: Part (1) of the internal diagram

The second part of the internal diagram is numbered by number 2 and colored in blue, illustrates the steps executed inside the classroom. In this part, two resources (lanes) execute various actions, the professor and the students. Figure 4.6 depicts part 2 of the internal diagram.

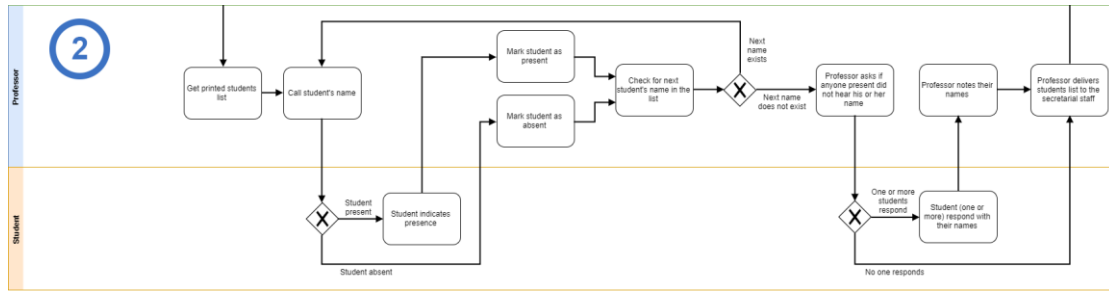


Figure 4-6: Part (2) of the internal diagram

Finally, the third part, numbered by number 3 and colored in red, illustrates the last steps of the attendance monitoring process, executed, again, by the secretarial stuff. Figure 4.7 presents the actions taken by the secretarial stuff, after the attendance monitoring process inside the classroom ends.

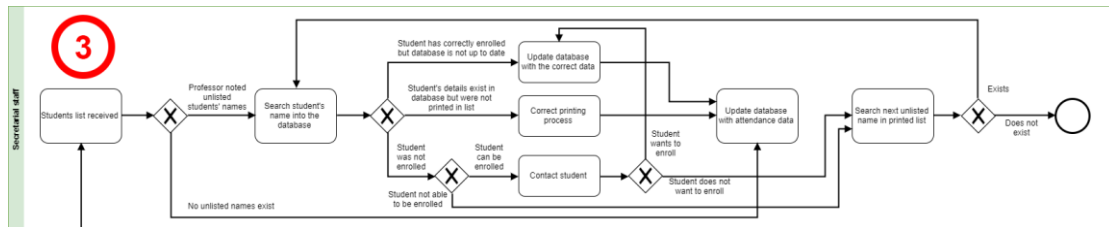


Figure 4-7: Part (3) of the internal diagram

4.5 Summary

In this chapter we tried to clearly distinguish the examined attendance monitoring process from the general operation of an educational institution. We have briefly described a wide range of a typical educational institution's operations and categorized those operations in five main groups of operations. We have placed the attendance monitoring process into the *Academics* group of operations and, subsequently, set some assumptions in order to make the process boundaries more obvious.

Then, we mentioned the resources and the actions they execute during the process. After that, we tried to illustrate a step-by-step sequence of actions taken in order to complete the attendance monitoring process. Besides the literal description of the attendance monitoring process, we provided a set of diagrams in order to depict the whole process and make it more understandable.

Finally, we see that the current attendance monitoring process includes many manual processes from different resources. Someone can easily infer that the occurrence of many manual tasks increases the possibility of error and, of course, the consumption of valuable time.

5 AUTOMATED PROCESS

In this chapter, we are going to propose an automated attendance monitoring process which will try to solve many problems, occurring during the traditional attendance monitoring process, described in chapter 4.

In order to be able to start developing an automated attendance monitoring process, we, first, have to design a theoretical model that captures all aspects of this process's business logic and meets the requirements and expectations of the potential users. Finally, we are going to depict the theoretical model into various business diagrams and provide some use cases of the proposed system.

5.1 Requirements

At this point we are going to describe the requirements of the system that is going to be developed, according to the feedback that we received from the interviews with the potential users.

5.1.1 Automation

The application should automate the traditional procedure in order to decrease the amount of manual processing of the data traded during the attendance monitoring process.

5.1.2 Security

All data that are handled by the application must be secured against accidental or deliberate actions from unauthorized users.

5.1.3 Easy to use

The user should be able to use the application without prior training. The user interface of the mobile and the web application must be as simple and minimal as possible, in order not to distract the user, and facilitate the quick completion of every action.

5.1.4 Identity verification

The system must be able to verify each user's identity, so that the person who registers an attendance is guaranteed to be the person, he or she, claims to be.

5.1.5 Location verification

The system has, also, be able to verify the actual location of the device that registers the attendance, in order to guarantee that the user is indeed physically present in the specific place at the specific time.

5.1.6 Independent of extra equipment

The system has to be able to operate as it is designed to operate without the need of extra equipment, for example, a barcode reader, a RFID device or a fingerprints scanner.

5.1.7 Minimum software dependences

The user must not be obliged to install extra software, apart from the mobile app, to his or her device, smartphone or desktop, in order to be able to use the application.

5.1.8 Provide fallback mechanisms

The system must provide some fallback mechanisms of attendance registration in case of network or hardware failure etc. For example, the user should have the choice of manual attendance registration if his or her device fails to execute automatic registration.

5.2 Proposed process

In this section we are going to introduce a theoretical model of the automated attendance monitoring process. The aim of the improved process is the automation of the most operations that are manually executed during the current attendance monitoring process.

There are some prerequisites that have to be met in order for the proposed process to operate as it is supposed to, such as that, each student has to hold an Android device, and that the classroom has to be equipped with a projection screen connected to a computer that has access to the internet or the institution's intranet.

The new process will be using a web-based application, hosted in the cloud¹⁵ or in the institution's servers, in conjunction with a mobile application installed on each student's mobile device. From now on, we are going to refer to this system as the *application*.

¹⁵ https://en.wikipedia.org/wiki/Cloud_computing

The process starts inside the classroom, where the professor has to login to the application providing his private credentials and navigating to specific menu in order to generate a unique QR code for the specific course, date and academic period. The generated QR code will be projected to the students through the projection screen.

Each student has to login to the application providing their credentials to the applications mobile app. The mobile app authenticates the student and verifies the location of the device. The student has to scan the QR code using the mobile app. The mobile app identifies the QR code, decodes the provided data and combines them with the current location of the device and the logged used, and finally, registers the attendance. All students should be able to, simultaneously, scan the QR code in order to register their attendance.

The QR code is going to be visible for a predefined period of time so that it will not be feasible for a late student to register his or her attendance. The period of time that the QR code is going to be visible is configurable by the professor. In case that one or more students do not have an Android device, the professor will have the ability to manually register their attendance from specific form in the application.

Each time a student scans the QR code, the mobile app updates the application's database with a new attendance record. Having automatically updated the database with the students' attendance records, the professor does not have to deliver any students list to the secretarial staff after the course ends. Furthermore, the secretarial staff will have the ability to generate reports from the application's database without having to manually import attendance records.

5.3 Business process diagrams

Having described the proposed attendance monitoring process's theoretical model, we can now present various diagrams in order to depict and summarize the described process.

5.3.1 BPMN

First, we are going to present diagrammatic depiction of the automated attendance monitoring process, using the BPMN specification.

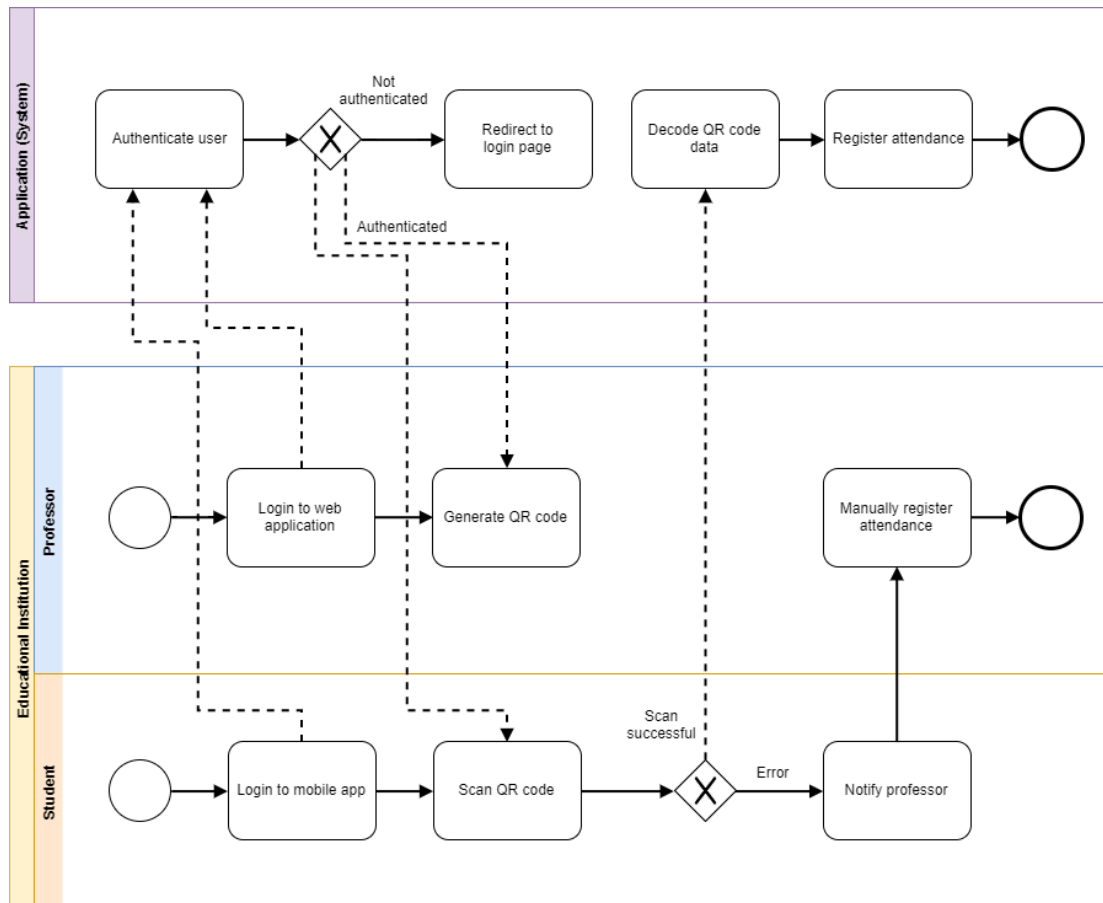


Figure 5-1: Automated attendance monitoring process

Figure 5-1 represents the proposed automated attendance monitoring process, as described in the previous section. As we can see, the BPMN diagram is consisted of two pools, the application (system) and the educational institution. Compared to the respective BPMN diagrams of the traditional attendance monitoring process, described in chapter 4, we are able to spot some important differences. First, the automated attendance monitoring process diagram does not include a lane that represents the secretarial staff, instead, the operations executed by the secretarial staff, are now executed, automatically by the application. Another major difference is that the automated attendance monitoring process diagram is much less complex than the traditional process's diagram.

It is clear that the majority of processes is now, mainly executed by the application, automatically. The amount of work executed by the application, as it is illustrated in figure 5-1 is only indicative and intentionally simplified in order to provide a clearer overview of the whole process. Each process executed by the application is an outcome of specific method calls, database queries and data validations that cannot be presented by a BPMN diagram in detail.

5.3.2 Use cases

In this section we are going to provide some use case diagrams for each user of the system (professor, student and secretarial staff).

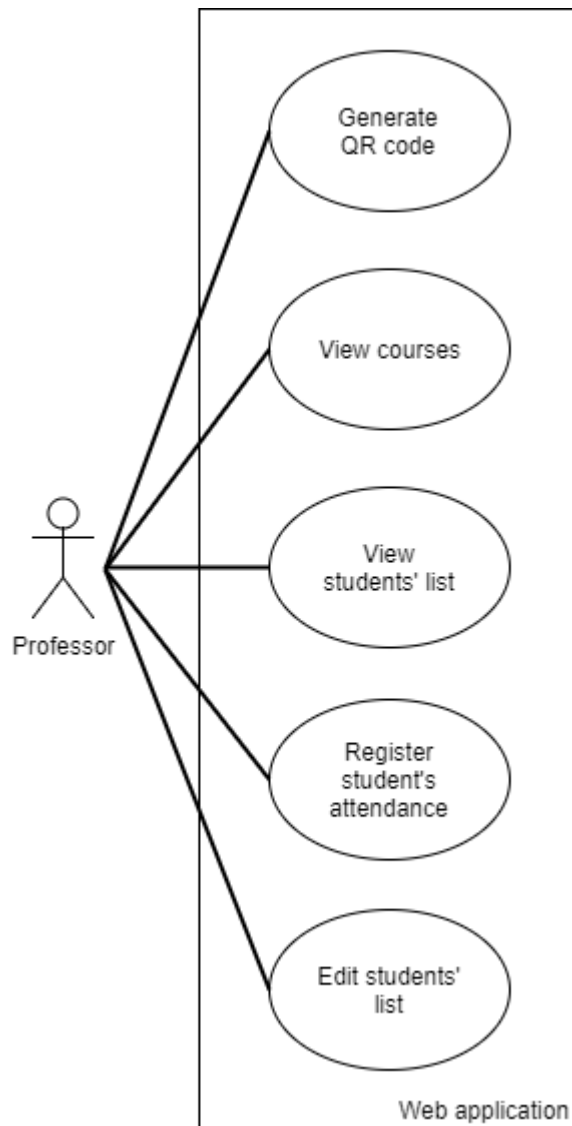


Figure 5-2: Professor using the web application

Figure 5-2 illustrates the most important operations that a professor can perform using the application. First of all, the professor will be able to generate, and display, a special QR code that holds specific data, for example, the course id, the semester, the current timestamp and other information. The professor will, also, be able to view the list of the available courses and the students list. Furthermore, the professor will have the option to edit the students list, for example, to add a new student to the students list (this action

will have to be approved by the secretarial staff on a next stage). Last but not least, the professor will be authorized to register attendance for one or more students in special cases, for example, when the student is not able to scan the generated QR code with his or her mobile device.

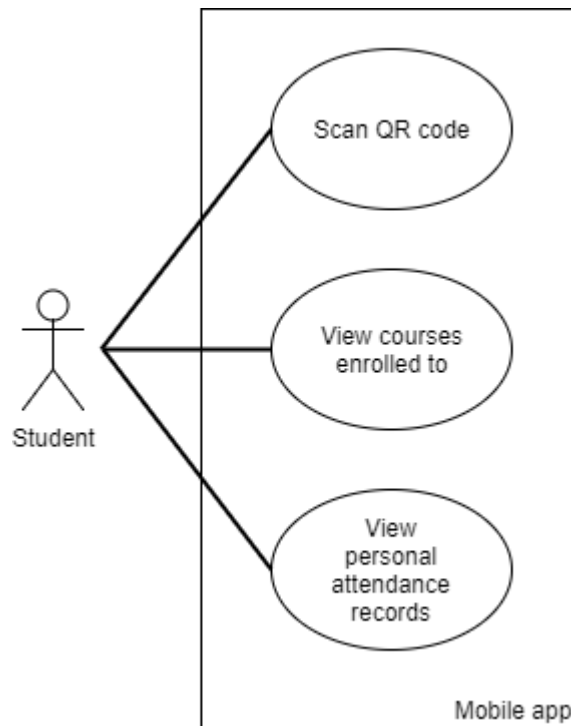


Figure 5-3: Student using the mobile app

Figure 5-3 presents the actions that a student will be able to execute using the system's mobile application. The main functionality of the mobile application is to scan the QR code that the professor has generated and displayed through a projection screen, decode the data that the QR code contains, validate the data and register the student's attendance. The student will also have the ability to view a list of the courses that he or she has been enrolled to, and, will also be able to view a history log of his or her registered attendance records.

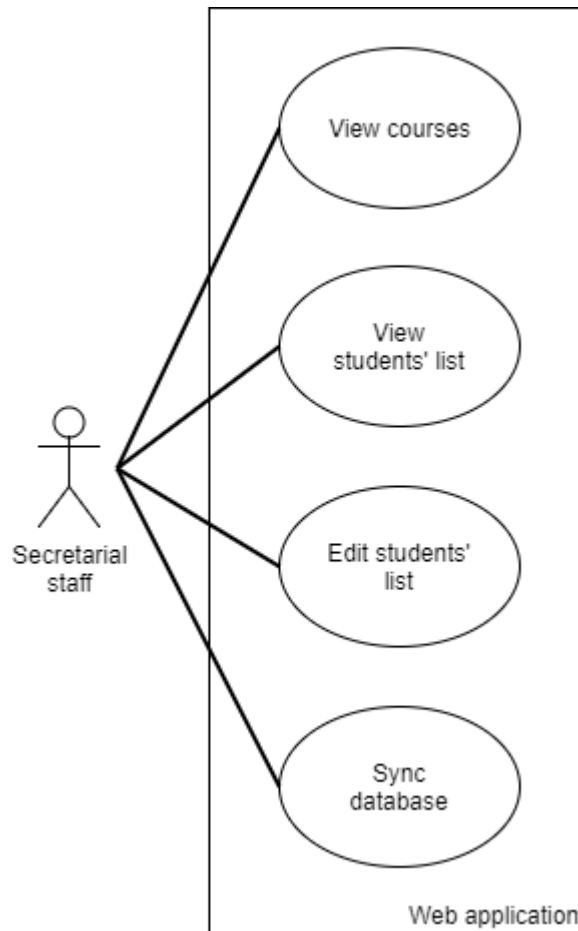


Figure 5-4: Secretarial staff using the web application

In figure 5-4 we can see the operations that the secretarial staff is able to execute using the application, such as to view the list of available courses, or the list of active students. The secretarial staff is also responsible for the maintenance of the students' details, so they will be able to edit the students list, for example add or remove one or more students from the students list. The secretarial staff will also be able to, manually, trigger an operation to synchronize the system's data with the data stored in the institute's system database.

5.4 Summary

In this chapter, we have introduced the main requirements that the proposed attendance monitoring process has to cover, such as automation, security, simplicity, identity and location verification, equipment independent etc.

After specifying the aforementioned requirements, we proceeded with the description of the proposed process in detail, defining the potential users of the system and the actions performed.

Finally, we presented a series of business diagrams in order to depict the described theoretical model of the proposed process. It is clear that the introduction of the automated execution of the majority of the, previously, manually executed operations, reduces the amount of time required in order to complete the attendance monitoring process.

6 APPLICATION DEVELOPMENT

In this chapter, we are going to present the development process of the automated attendance monitoring system, introduced in chapter 5. First, we are going to describe the system architecture and the connection of its individual items between them, as such as the data flow through the whole system.

After a theoretical introduction and diagrammatic depiction of the system, we are going to present some information about the available development options, such as the main programming languages and web frameworks. After providing a brief introduction and comparison of the available solutions, we will define the development stack that we are going to use in order to develop the attendance monitoring system.

Having come up with a specific set of development tools, we are going to illustrate the database schema, the heart of the system in terms of data organization and storage. As soon as we have provided a full picture of the system's database design, we will proceed with the description of the API¹⁶, the interface which is going to be responsible for the communication of the web and mobile applications with the database, by exposing various API calls to authenticated and authorized users only. Having provided a solid understanding of the system's API operation, we can continue with the description of the front-end applications (web and mobile) which are going to communicate with the API in order to perform specific tasks, for example, insert a new student, get the list of active courses, get authorization level of an authenticated user etc. By this communication, we say that an application **consumes** an API, in other words, the application uses any available call exposed by the API.

Continuing, we will provide some technical specifications or standards, which should be met, in order for the attendance monitoring system to operate normally. Finally, we will provide references to the public repositories which host the system's source code and detailed documentation for anyone interested in contributing to the development of the system. The chapter is going to be closed with a short summary of the development process, presenting difficulties arisen during this process and mentioning various techniques that were used and could be considered as best practices.

¹⁶ https://en.wikipedia.org/wiki/Application_programming_interface

6.1 System architecture

The automated attendance monitoring system we are going to develop, consists of three, plus one, main parts:

- Web application
- Mobile application
- API (Application Program Interface)
- Database

We say that the system consists of $3 + 1$ main parts, referring, on the one hand, to the web and mobile applications and the API, and on the other hand, to the database, because the database is the place that holds the whole system's data, and all other parts of the system communicate with the database in order to perform various actions.

Figure 6-1 provides a depiction of the aforementioned system's architecture and composition. The user (student, professor etc.) interacts with the system through one of the two available interfaces, the web, or the mobile application. As described from the use cases presented in chapter 5, not all possible actions can be performed by all users, using both applications. For example, a student will have the ability to scan a QR code using only the mobile application, in order to register his or her attendance.

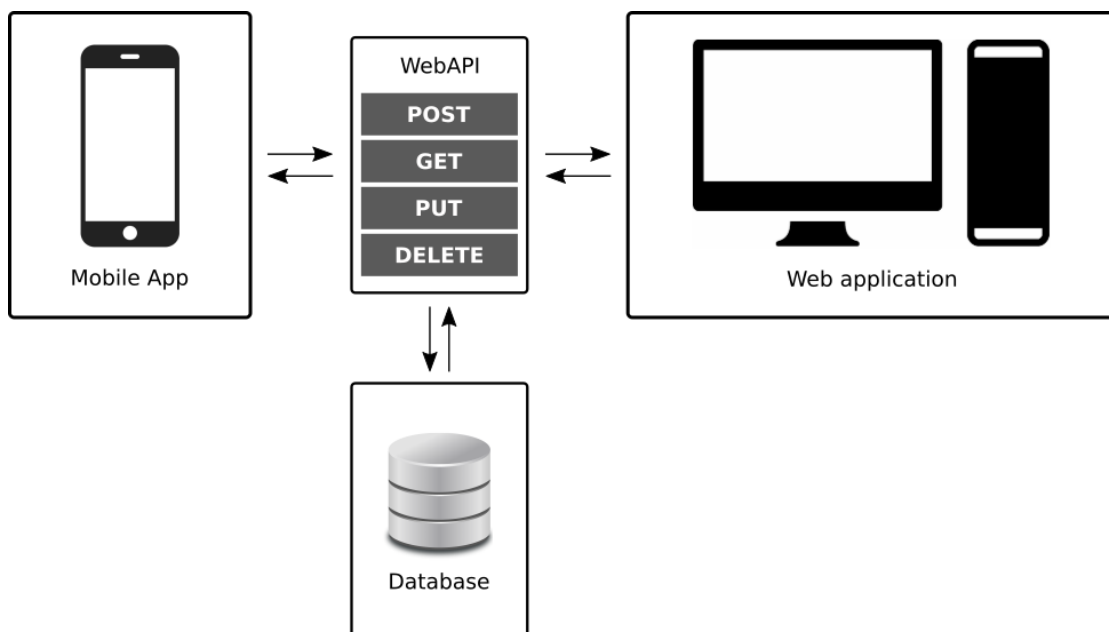


Figure 6-1: System architecture and data flow

In more detail, the system is designed in a way that, both the mobile and the web application, communicate with the database only through API calls made to the Web API application. We can describe the web API application as a middleware, responsible for the exposure of the database schema to the consuming clients (applications), by providing a list of available calls, each of which is designed to receive specific parameters in order to perform specific tasks. As we can see in figure 6-1, there is no direct communication, either between the database and the mobile application, nor between the database and the web application. This “isolation” ensures that the database is accessed by the consuming application through a unique entry point, avoiding duplicated code and providing extra security.

6.2 Development stack

Having come up with a theoretical application framework, and after we have decided the parts that our system is going to be consisted of, we have to choose which programming language we are going to use, both during the front-end and the back-end development process, the database server and any other tools that are going to be used in order to develop our application. This set of languages and tools is known as the *development stack* for our application.

6.2.1 Alternative solutions

There are so many different solutions in software development, so that a detailed comparison of all the alternatives would have occupied lots of hundredths of pages and this is not the purpose of the current thesis. Instead of a detailed comparison, we are going to provide a list of the most popular programming languages and frameworks, and we are going to present a brief comparison between them.

6.2.1.1 Programming languages

In this section, we are going to provide some rankings from different sources, about the most popular programming languages.

Figure 6-2 presents a list of the top ten programming languages from an article of IEEE Spectrum in July 2017. The results of this list emerged taking account of various factors, such as, what language employees are looking for and what is currently active in open source community.

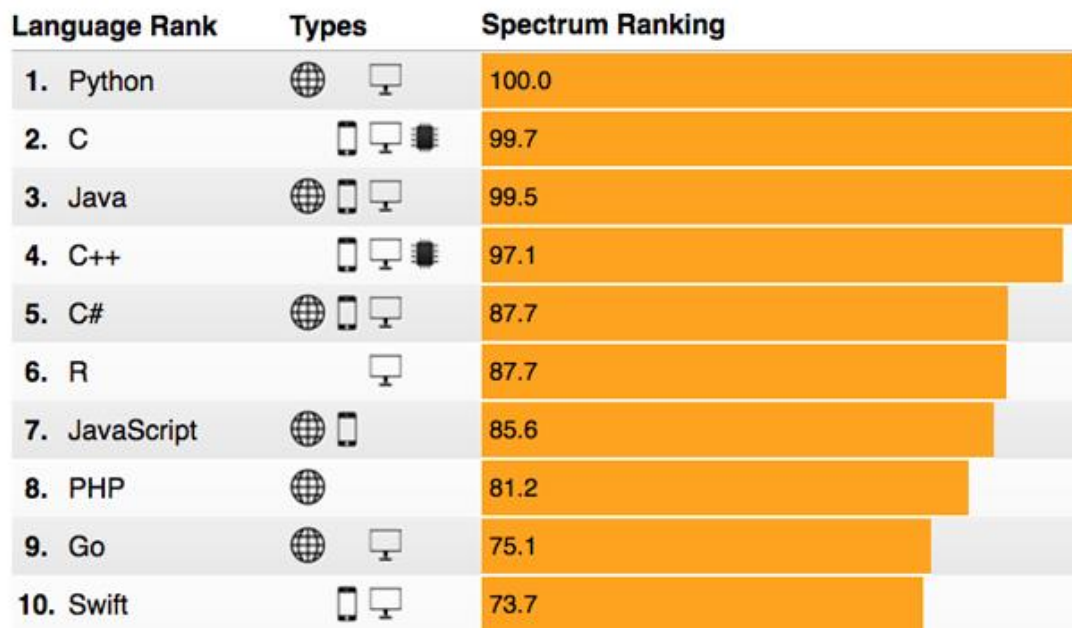


Figure 6-2: Top ten programming languages according to IEEE Spectrum (July 2017)
<https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>

Figure 6-3 presents the most popular programming languages in 2017, according to GitHub, the most popular repository hosting platform. This ranking lists the results counting the number of pull requests of each language in GitHub repositories.

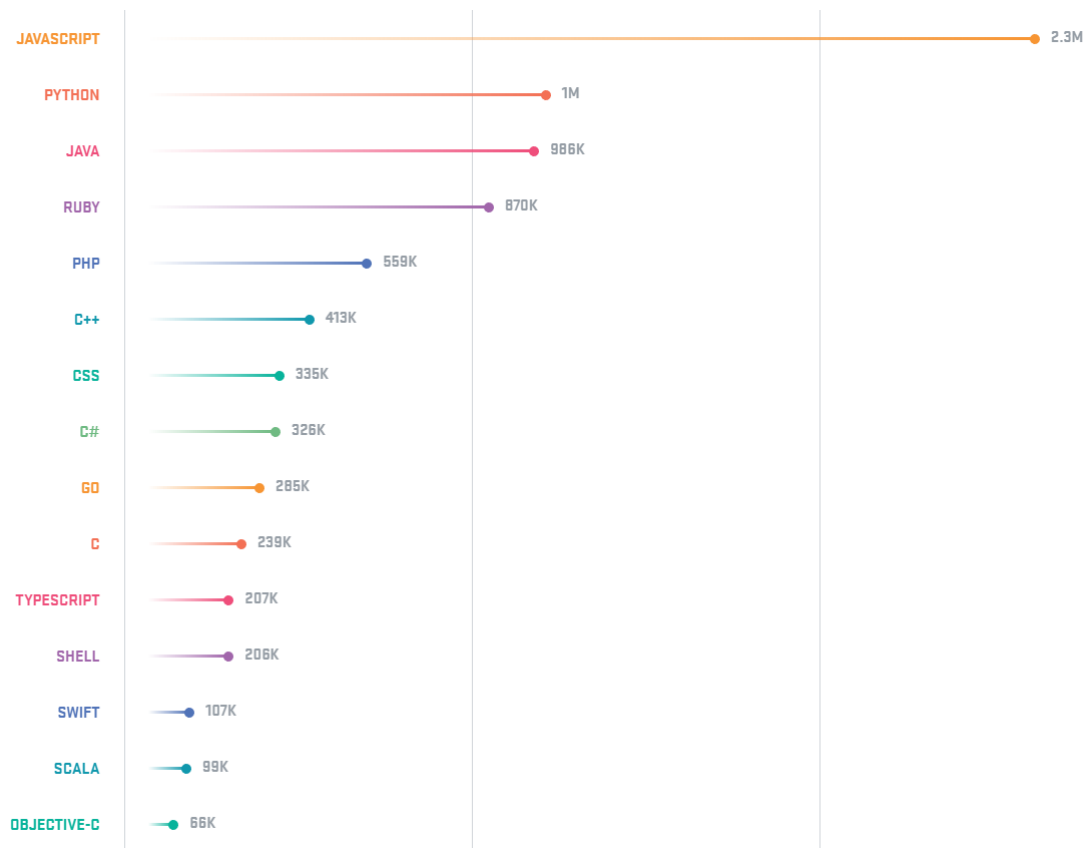


Figure 6-3: Top 15 most popular programming languages of 2017 in number of pull requests (GitHub)

Table 6-1, presents the ten most popular programming languages, according to the *TIOBE index*¹⁷, a ranking measuring the popularity of programming languages, based on search engines results of queries containing terms related to any programming language, created and maintained by the TIOBE company. The TIOBE index gets updated every month and the specific index refers to the results of December 2017.

Nov 2017	Nov 2016	Change	Programming Language	Ratings	Change
1	1		Java	13.231%	-5.52%
2	2		C	9.293%	+0.09%
3	3		C++	5.343%	-0.07%
4	5	▲	Python	4.482%	+0.91%
5	4	▼	C#	3.012%	-0.65%

¹⁷ https://en.wikipedia.org/wiki/TIOBE_index

6	8	▲	JavaScript	2.972%	+0.27%
7	6	▼	Visual Basic .NET	2.909%	-0.26%
8	7	▼	PHP	1.897%	-1.23%
9	16	▲▲	Delphi/Object Pascal	1.744%	-0.21%
10	9	▼	Assembly language	1.722%	-0.72%

Table 6-1: Ten most popular programming languages according to the TIOBE index for December 2017
(www.tiobe.com)

Figure 6-4 illustrates the evolution of the popularity of the most popular programming languages from the year 2002 up to now, according to the TIOBE index.

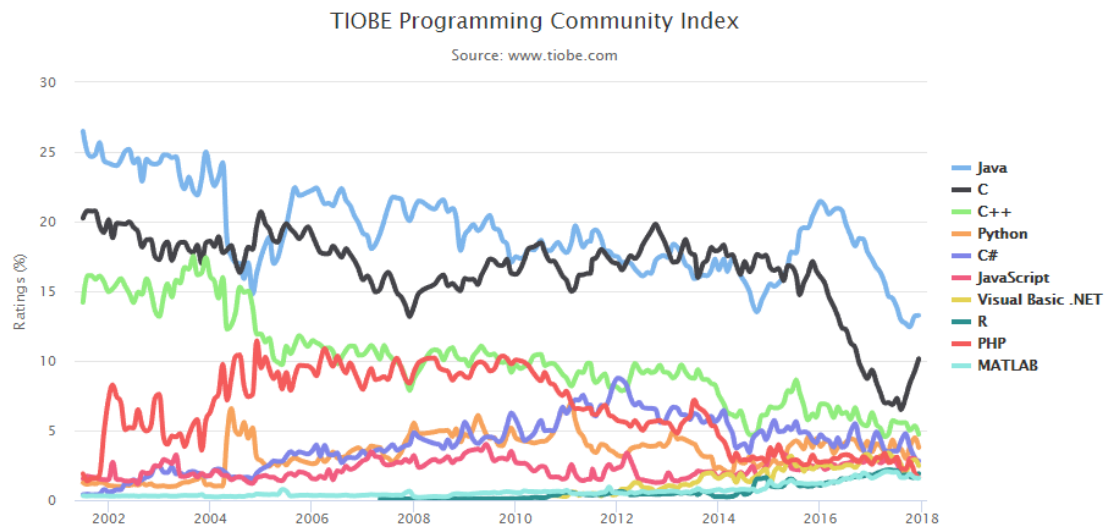


Figure 6-4: TIOBE index time line

After we have presented the most popular programming languages, we will try to make a comparison between the most popular languages for the **web**, based on the following criteria:

- Environment: the environment for which the language is designed to operate on (Client-side / Server-side).
- Object oriented: the language uses objects to represent the environment (Classes, Inheritance etc.).
- Functional: the language performs a kind of computation of mathematical functions and avoids data mutations, meaning that, each function, given the same

parameters, produces the exact same result every time and does not have side effects on other variables or objects outside its body.

- Event driven: the language listens to events in order to trigger the execution of various tasks.
- Strongly typed: a language is strongly typed when the type of each variable is declared at the time of the variable's creation and this type never change. On the other side, weakly typed languages, try to guess the variables type at runtime.
- Community: the number of people who, actively, contribute and support a programming language. For this criterion, we estimate the community size, based on StackOverflow's rankings of tags activity for each language. We provide a community rating starting from 0, meaning no community at all, and reaching up to 10 for the fully supported language by a big number of people supporting and contributing to it.
- Release date: informational field about the release date of the first version of each language.

<i>Language</i>	<i>Environment</i>	<i>Object oriented</i>	<i>Functional</i>	<i>Event driven</i>	<i>Strongly typed</i>	<i>Community</i>	<i>Release date</i>
<i>Javascript</i>	Client-side / Server-side	✓	✓	✓		7	1995
<i>Java</i>	Server-side	✓	✓	✓	✓	8	1995
<i>C#</i>	Server-side	✓	✓	✓	✓	9	2000
<i>Python</i>	Server-side	✓	✓			5	1991
<i>PHP</i>	Server-side	✓	✓			8	1995

Table 6-2: Web programming languages comparison

Table 6-2 presents a comparison of 5 of the most popular programming languages for web development. Javascript, although initially designed for client-side development, is the only

language that provides the ability for both client-side and server-side development (Node.js) and is strongly supported by a big community of programmers.

Both Java and C# are object oriented, type safe (strongly typed) languages providing many features to the programmer and supported by strong community. The main difference between the two is the target platform which, for C# is Windows, and for Java is any platform.

Finally, Python and PHP are both object-oriented languages, but they are loosely typed, making them more error prone, compared to C# and Java. It seems that Python is supported by a smaller community than PHP and the other languages.

Figure 6-5 depicts the evolution of tag activity for each programming language during time, according to StackOverflow.

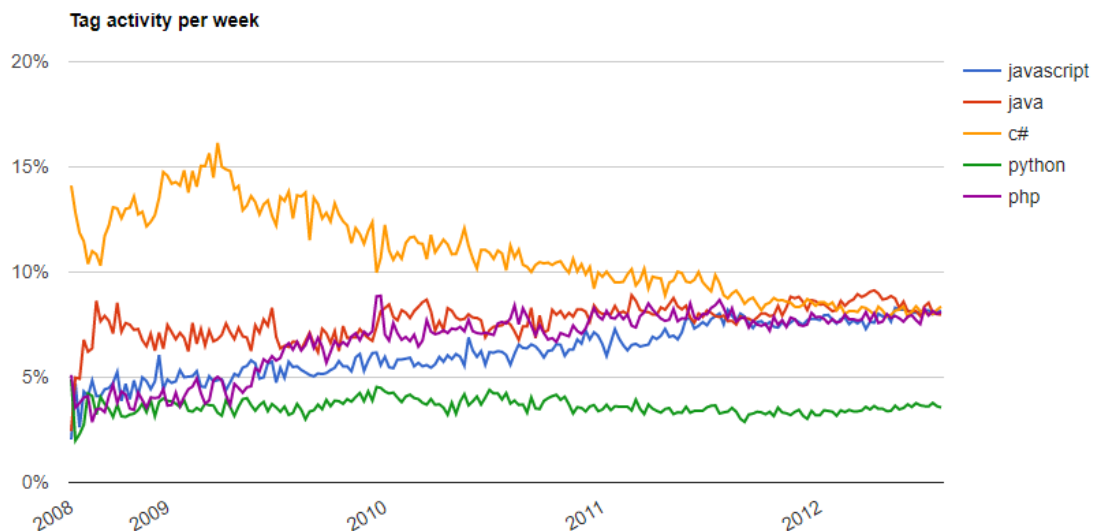


Figure 6-5: Tag activity per week (http://hewgill.com/~greg/stackoverflow/stack_overflow/tags/)

6.2.1.2 Web frameworks

A web framework is a subset of the more generic term of *software framework* and can be considered as a set of tools that aim at automating the development of web applications. Web frameworks facilitates the development of web applications providing abstractions of complex common activities such as connecting to the database, session management, file system interactions etc.

Table 6-3 presents the most popular and successful web frameworks for the most popular programming languages, providing epigrammatic information for each of them.

<i>Framework</i>	<i>Programming language</i>	<i>Front-end</i>	<i>Back-end</i>	<i>Architectural pattern</i>
<i>Angular</i>	Javascript	✓		MVC
<i>React.js</i>	Javascript	✓		Hierarchical
<i>Vue.js</i>	Javascript	✓		SPA (Single Page Application)
<i>Node.js</i>	Javascript		✓	Non-blocking, event-driven I/O
<i>Laravel</i>	PHP		✓	MVC
<i>Symfony</i>	PHP		✓	MVC
<i>Zend</i>	PHP		✓	MVC
<i>Asp.Net</i>	CLI languages		✓	MVC and others
<i>Django</i>	Python		✓	MVC
<i>Rails</i>	Ruby		✓	MVC and others

Table 6-3: Web frameworks comparison

The first four are Javascript frameworks and the three of them (Angular, React and Vue) are used for front-end development, due to the fact that Javascript is primarily executed in web clients (browsers). Despite that, since 2009, Javascript can be execute in server side too, using Node.js run time environment.

The next three frameworks are PHP frameworks used for back-end development, with Laravel being the most recent (2011) between them and perhaps the most active right now.

ASP.NET is the framework released by Microsoft in 2002, built on the CLR (Common Language Runtime) giving the ability to programers to use any of the .NET supported programming languages (C#, C++, F#, Visual Basic etc.)

The last two frameworks listed in table 6-2 are Django framework for Python and Rails framework for Ruby, also known as *Ruby on Rails*.

Figure 6-5 illustrates the most popular frameworks in a period of time starting from November 2014 up to the end of 2017, according to *hotframeworks.com*. The ranking presents the score of each framework that is measured by two factors:

- GitHub: number of stars the git repository for a framework has on GitHub
- Stack Overflow¹⁸: the number of questions tagged with the frameworks name

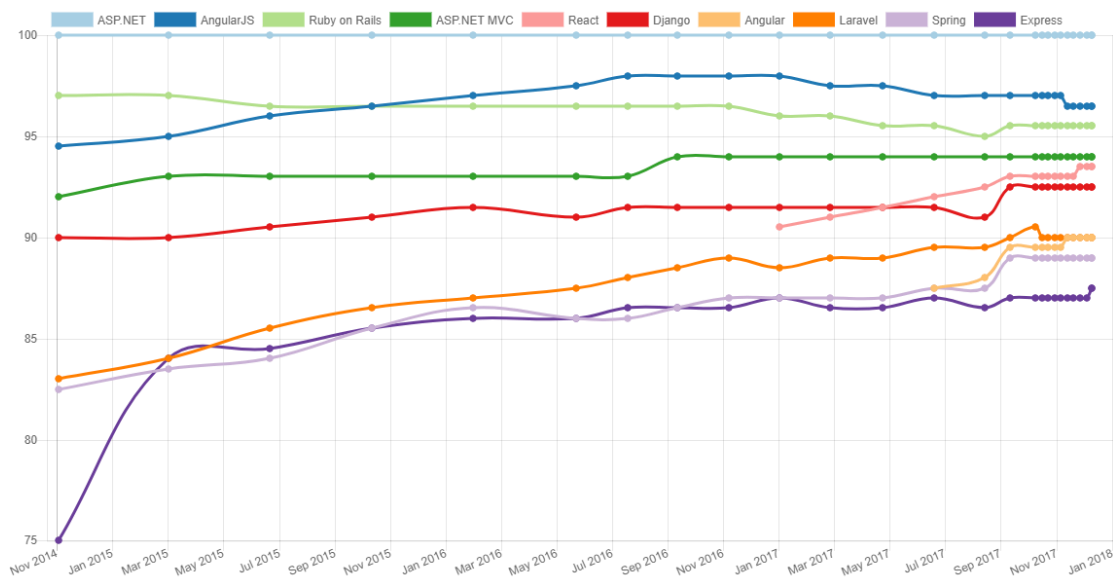


Figure 6-6: Popular frameworks (<https://hotframeworks.com/>)

6.2.1.3 Mobile application frameworks

For the development of the mobile app, we are not going to develop in native code for each platform (Java for Android, Objective-C for iOS etc.), instead, we are going to develop a *hybrid*

¹⁸ <https://stackoverflow.com/>

*mobile app*¹⁹. A hybrid mobile app is a mobile application that is written with a non-native language and is hosted inside a native mobile app.

In this section, we are going to present the main frameworks for hybrid mobile app development.

- **Xamarin:** (<https://www.xamarin.com/>) provides the ability to develop reusable code in C# and produce native Android, iOS or Windows apps.
- **NativeScript:** (<https://www.nativescript.org/>) a framework that produces native Android and iOS apps from Javascript / Typescript / Angular code.
- **Ionic:** (<https://ionicframework.com/>) a mobile app development framework that uses web technologies such as Angular, HTML5, CSS etc.
- **PhoneGap:** (<https://phonegap.com/>) another hybrid mobile app development framework that produces native code from web technologies codebase (Javascript, HTML5, CSS etc.)

6.2.1.4 Database server

Apart from the programming language and the respective framework, there are many different alternatives for the database management. In general, there are various different types of databases, but, in web development there are two main types:

- **Relational databases:** Databases that store and organize the data into tables with columns and rows as proposed by Edgar F. Codd²⁰ in 1970. All relational databases use SQL (Structured Query Language) to query the database. There are various systems that manage relational databases, called RDBMS (Relational Database Management Systems) like Microsoft SQL, PostgreSQL, MySQL etc.
- **No SQL**²¹: Are databases that do not conform to the relational database model. These databases store and manipulate data in a different way and can be categorized depending on the stored data type:
 - Document

¹⁹ <https://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/>

²⁰ https://en.wikipedia.org/wiki/Edgar_F._Codd

²¹ <https://en.wikipedia.org/wiki/NoSQL>

- Column
- Key-Value
- Graph
- Multi-model

6.2.2 The chosen stack

Now that we have an image of the available solutions for the development process, we have to pick up a set of tools that we are going to use in order to develop the automated system of attendance monitoring.

Figure 6-6 illustrates the development stack we have chosen in order to develop our application, organized in a table format, the columns of which indicate the specific application of our system, and the rows, the respective application layer.












	API	Web app	Mobile app
Front-end framework		  webpack  yarn	
Language		TypeScript	 JavaScript
Framework			
IDE	 Visual Studio®	 Visual Studio Code	
Database		 Microsoft® SQL Server®	

Figure 6-7: Development stack

Starting from the bottom, we have chosen **Microsoft SQL Server** for the database management. We chose a relational database management system because we will have to deal with heavy structured data, stored in tables with specific columns, that can be retrieved with SQL queries.

On top of the database layer, we inject the IDE (Integrated Development Environment) layer. Although it is not a part of the application itself, it provides useful tools in order to facilitate the development process. Such tools are, at least, a source code editor, a build tool and a debugger (a program that tests the program that is currently developed). For the current system, we will use two, powerful IDEs, **Visual Studio Community 2017**, for the development of the API, and **Visual Studio Code**, for the development of the Web client and the mobile app, both released by Microsoft. Visual Studio Code is a lightweight source code editor, yet very powerful and customizable, ideal for front-end development using modern frameworks like Angular, React, Vue etc. The selection of Visual Studio as the IDE for the API application development, comes as a result of the selection of the respective web framework that we are going to introduce right now.

For the development of the API we will use the **ASP.NET Core 2.0**²² framework, an evolution of the classic ASP.NET framework that Microsoft released in 2002 for the first time. ASP.NET Core is cross-platform, meaning that it can run on Windows, macOS and Linux and provides tools for easy integration with modern client-side frameworks.

For the development of the back-end of the web client we are going to use **Node.js**, a Javascript runtime that uses events in order to execute its operations. We chose to use Node.js because we need a lightweight application that communicates with the API via HTTP requests, and the implementation on a framework like ASP.NET would be an overkill.

Finally, for the development of the mobile app, we will use **Ionic**, an open source mobile development framework that provides the tools to target multiple platforms (Android, IOS, Windows Phone) using a single code base (Javascript, Angular). Ionic provides multiple plugins that can be used in order to access native device functionality (camera, GPS, Network etc.).

We have reached the programming language layer, the fourth layer, starting from the bottom of the development stack. For the development of the API, having chosen ASP.NET Core as the back-end framework, we have the ability to choose any of the programming languages supported

²² <https://docs.microsoft.com/en-us/aspnet/core/>

by the .NET, the CLI languages²³, as described in *Programming languages* section of this chapter. Thus, we are going to choose **C#**, an object oriented and strong typed programming language that provides support for functional programming through the introduction of lambda expression syntax.

Moving on to the development of the web client, we will choose **Typescript**, a superset of Javascript that provides strong typing to the Javascript language and gets compiled down to vanilla Javascript code. Except from strong typing, Typescript supports the use of interfaces, enums, next generation Javascript syntax that may not be fully adopted by all browsers, like arrow function syntax, class declaration etc. Finally, the development of the mobile app is going to be with vanilla **Javascript**.

Reaching the top of the development stack, we meet **React**, a front-end framework that we are going to use for the development of the web client, developed by Facebook, that uses reusable components and sophisticated data flow with the, optional, support of Redux library. Alongside with React, we will use **Webpack**²⁴, a module bundler that helps at combining different modules into a single file. Finally, Facebook's **Yarn**²⁵ is going to be used in order to manage our npm²⁶ packages during the front-end development.

For the other two applications (API and mobile app) we are not going to use any extra front-end framework because, on one hand, the API will not expose any use interface, only a public API for the consuming applications, and on the other hand, for the mobile app we are going to stick on pure css²⁷ styling in order to deliver a minimal user interface.

6.3 Database schema

In this section, we are going to illustrate the application's database structure (schema), a relatively simple database consisting of eleven tables. First, we are going to present the tables of the database, explaining the type of information that each table holds and the fields (columns) that each table contains in order to store this information. After that, we are going to present and analyze the relations between the database's tables.

²³ https://en.wikipedia.org/wiki/List_of_CLI_languages

²⁴ <https://webpack.github.io/>

²⁵ <https://yarnpkg.com/lang/en/>

²⁶ <https://www.npmjs.com/>

²⁷ <https://www.w3.org/Style/CSS/Overview.en.html>

6.3.1 Tables

Each table is going to be illustrated by a diagram, indicating the table name and the names of its columns. Then we are going to explain each column's purpose and the type of data that each column stores.

The kind of information that each table column stores, is described by its data type. Table 6.4 presents the different data types that our database contains. For more information about MS SQL data types you can refer to Microsoft's documentation²⁸.

<i>Name</i>	<i>Data type</i>
<i>int</i>	Integers
<i>bigint</i>	Big integers ($-2^{63} - 2^{63}-1$)
<i>varchar(n)</i>	Strings with n characters and maximum length of 8000 characters
<i>datetime</i>	Date and time data
<i>bit</i>	Stores 1, 0 or NULL
<i>float</i>	Floating point numeric data

Table 6-4: Columns data types

Every table has a column of type *int*, with name *ID*, and this column is the primary key of the table.

6.3.1.1 Users

[Users] table stores information about the users of the application, and has 5 columns as shown in the following image:

²⁸ <https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-2017>

Users
ID
Username
Password
FirstName
LastName

Figure 6-8: Users table

Apart from column ID (primary key), this table has 4 other columns:

- Username (varchar(255)): Stores the username and is not allowed to be null
- Password (varchar(255)): Stores the user's password (not encrypted) and is not allowed to be null
- FirstName (varchar(255)): The user's first name
- LastName (varchar(255)): The user's last name

6.3.1.2 Roles

[Roles] table stores information about the available roles that each user can have. The application serves different features to each user, according to their roles. For example, a user that has the role of *Professor*, could be able to view a list of the students enrolled to a specific course. In contrast, another user that does not have the role of *Professor*, should not be able to view such a list.

Roles
ID
Title
Description

Figure 6-9: Roles table

This table, as shown in figure 6-9, has 3 columns and, except from column ID, these are the following:

- Title (varchar(255)): The title of the role and is not allowed to be null
- Description (varchar(255)): Holds descriptive text about the specific role

6.3.1.3 UserRoles

[UserRoles] is a table that connects [Users] and [Roles] tables. This means that there is a many-to-many relationship between these two tables, for example, a user can have multiple roles, but, on the other hand, a role can be assigned to multiple users. The connection of these tables is defined by the intermediate table's ([UserRoles]) foreign keys²⁹ (FK). These keys are ordinary table columns that refer to a primary key of another table. These columns are going to be mentioned as FK to [Table_Name].Column_Name, meaning that, this column refers to the Column_Name column of the [Table_Name] table.

UserRoles
ID
UserID
RoleID

Figure 6-10: UserRoles table

The columns of the table are the following:

- UserID (int): FK to table [Users].ID, not allowed to be null
- RoleID (int): FK to table [Roles].ID, not allowed to be null

6.3.1.4 Courses

[Courses] table stores all the available courses.

Courses
ID
Title
Descr
IsActive

Figure 6-11: Courses table

This table has the following 3 columns (except from the ID column):

²⁹ https://en.wikipedia.org/wiki/Foreign_key

- Title (varchar(255)): This column holds the title of the course and is not allowed to be null
- Descr (varchar(500)): Optional description of the course
- IsActive (bit): This column indicates that the specific course is active or inactive. An inactive course cannot be referred by any enrollment.

6.3.1.5 ClassRooms

[ClassRooms] table holds all the available classrooms. Every classroom can be approximately, geographically, defined by its coordinates (longitude - latitude) and a number (RangeInMeters) that indicates the radius of a conceivable circle with center at the classroom's coordinates.

ClassRooms
ID
Descr
Longitude
Latitude
RangeInMeters

Figure 6-12: ClassRooms table

The table's columns are the following:

- Descr (varchar(255)): An optional description of the classroom
- Longitude (float): The longitude of the classroom
- Latitude (float): The latitude of the classroom
- RangeInMeters (float): The radius that indicates the conceivable circle that defines the classrooms geographical boundaries

6.3.1.6 AcademicTermTypes

[AcademicTermTypes] table holds the available academic term types. An academic term is a portion of the academic year, for example a quarter of the year, or a semester (six months).

AcademicTermTypes
ID
Title
Descr

Figure 6-13: AcademicTermTypes table

This table has only two columns:

- Title (varchar(255)): The title of the academic term type that is not allowed to be null
- Descr (varchar(500)): An optional description of the type

6.3.1.7 AcademicTerms

[AcademicTerms] table stores the actual academic terms, defined by their type, the starting and the ending dates.

AcademicTerms
ID
TypeID
StartDate
EndDate
Descr

Figure 6-14: AcademicTerms table

This table has 4 columns:

- TypeID (int): FK to table [AcademicTermTypes].ID, not allowed to be null
- StartDate (datetime): The date that the academic term starts, not allowed to be null
- EndDate (datetime): The date that the academic term ends, not allowed to be null
- Descr (varchar(500)): An optional description of the academic term

6.3.1.8 AttendanceTypes

[AttendanceTypes] table holds the available attendance types that a user can use. These are the following two:

- Presence
- Absence

AttendanceTypes
ID
Title
Descr

Figure 6-15: AttendanceTypes table

This table has 2 columns:

- Title (varchar(255)): The attendance type title, that is not allowed to be null
- Descr (varchar(255)): An optional description of the attendance type

6.3.1.9 CourseAssignments

[CourseAssignments] table holds the assigned courses to each professor, for specific academic term.

CourseAssignments
ID
CourseID
ProfessorID
AcademnicTermID
Notes
ClassRoomID
LecturesMinNum
LecturesTargetNum

Figure 6-16: CourseAssignments table

This table has 7 columns:

- CourseID (int): FK to table [Courses].ID, not allowed to be null
- ProfessorID (int): FK to table [Users].ID, not allowed to be null
- AcademicTermID (int): FK to table [AcademicTerms].ID, not allowed to be null
- ClassRoomID (int): FK to table [ClassRooms].ID, can be null
- Notes (varchar(500)): Optional notes to be kept for the specific course assignment
- LecturesMinNum (int): The minimum number of lectures that a student has to attend in order to be able to pass the specific course

- LecturesTargetNum (int): The number of lectures that is planned to be delivered for the specific course

6.3.1.10 Enrollments

[Enrollments] table holds the courses that each student is enrolled to, for specific academic term.

Enrollments
ID
StudentID
CourseID
AcademicTermID
StartDate
EndDate
Notes

Figure 6-17: Enrollments table

This table has 6 columns:

- StudentID (int): FK to table [Users].ID, not allowed to be null
- CourseID (int): FK to table [Courses].ID, not allowed to be null
- AcademicTermID (int): FK to table [AcademicTerms].ID, not allowed to be null
- StartDate (datetime): The starting date of the enrollment, not allowed to be null
- EndDate (datetime): The ending date of the enrollment, not allowed to be null
- Notes (varchar(500)): Optional notes for the specific enrollment

6.3.1.11 AttendanceLog

[AttendanceLog] table holds the most important data of the application, the attendance logs of each student.

AttendanceLog
ID
StudentID
CourseID
AcademicTermID
Date
AttendanceTypeID
Longitude
Latitude
Timestamp
LecturesLogID

Figure 6-18: AttendanceLog table

This table has 9 columns:

- StudentID (int): FK to table [Users].ID, not allowed to be null
- CourseID (int): FK to table [Courses].ID, not allowed to be null
- AcademicTermID (int): FK to table [AcademicTerms].ID, not allowed to be null
- AttendanceTypeID (int): FK to table [AttendanceTypes].ID, not allowed to be null
- Date (datetime): The date and time of the registration, not allowed to be null
- Longitude (float): The longitude of the coordinates retrieved from the student's device, at the time of the attendance registration
- Latitude (float): The latitude of the coordinates retrieved from the student's device, at the time of the attendance registration
- Timestamp (bigint): The actual timestamp at the time of the attendance registration
- LecturesLogID (int): FK to table [LecturesLog].ID, not allowed to be null

6.3.1.12 LecturesLog

[LecturesLog] table holds the date of each lecture delivered for specific course assignment.

LecturesLog
ID
CourseAssignmentID
Date

Figure 6-19: LecturesLog table

This table has 2 columns:

- CourseAssignmentID (int): FK to table [CourseAssignments].ID, not allowed to be null
- Date (datetime): The date and time of specific lecture, not allowed to be null

6.3.2 Relationships

In this section we are going to present the most important and, clearly, distinguished relationships between the database schema tables.

6.3.2.1 User and Roles

The first, and most clear relationship that is defined from the database schema is the relationship between three tables, [Users], [UserRoles] and [Roles]. This relationship defines the roles that each user of the application has.

The main goal of this relationship is to express a *many-to-many* relationship between the [Users] and the [Roles] table. By the term *many-to-many*, we want to express the fact that a user can have more than one role, and, vice versa, a role can be assigned to more than one user. This can be achieved by the use of an intermediate table, the [UserRoles], that stores pairs of user-role.

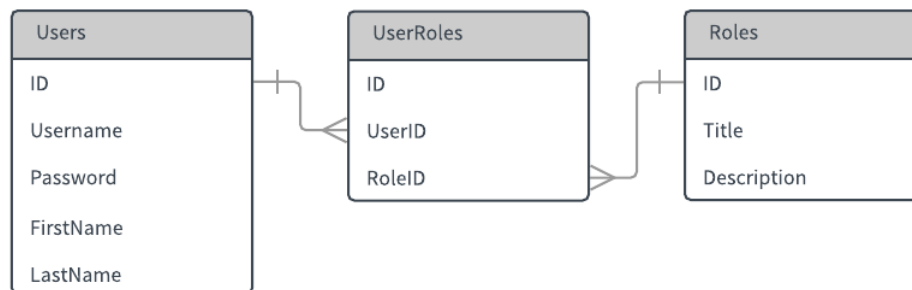


Figure 6-20: Relationship between [Users] and [Roles] tables

6.3.2.2 Attendance Logs

Another important set of relationships is the one that is defined by the [AttendanceLog] table. This table is designed to hold information about a user's attendance registration logs.

Apart from the date and geographical coordinates, this information has to combine values from other tables, such as the student's id, from [Users] table, the course's id, from the [Courses] table, the academic term's id, from the [AcademicTerms] table and the attendance's type id, from the [AttendanceTypes] table.

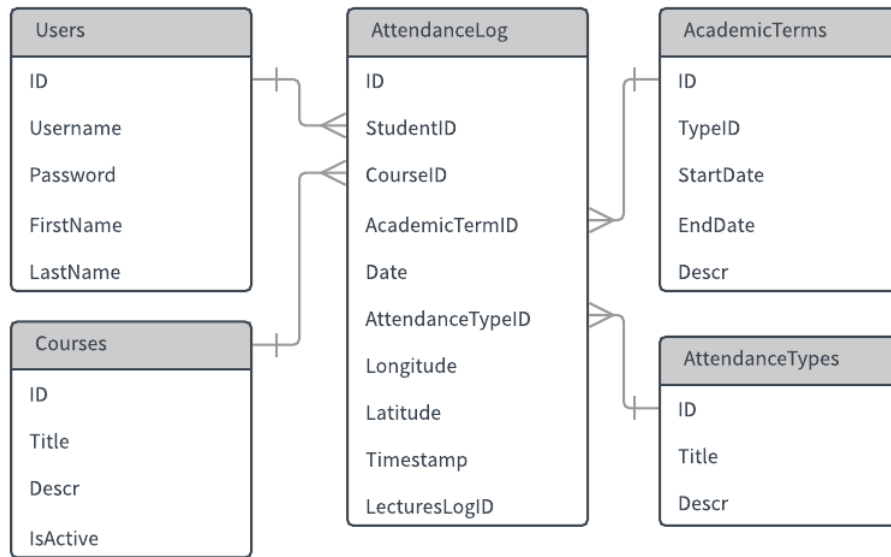


Figure 6-21: Relationships defined by [AttendanceLog] table

6.3.2.3 Enrollments

Now, we are going to illustrate the relationship that is defined by the [Enrollments] table. This table is designed to hold the enrollments of all students. An enrollment has to report a specific student, a specific course, for a specific academic term. The following figure depicts this relationship.

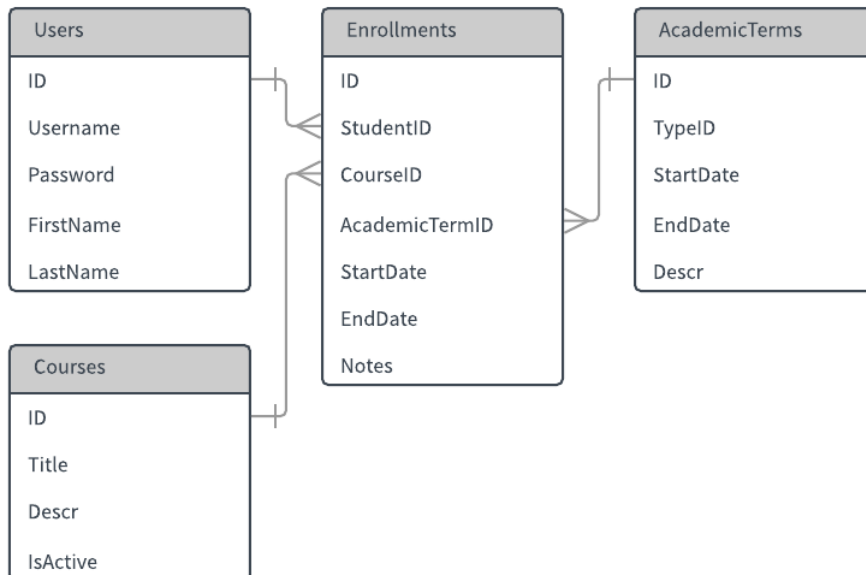


Figure 6-22: Relationships defined by [Enrollments] table

6.3.2.4 Course assignments

Finally, we are going to present the relationship that is defined by the [CourseAssignments] table. Similarly, to the enrollments relationships, the set of relationships defined by the [CourseAssignments] table, comes as result of the need to store the information of the assignment of each course to one or more professors.

As you can see in figure 6-23, a course assignment has to combine information from four other tables, [Courses], [Users], [AcademicTerms] and [ClassRooms] tables.

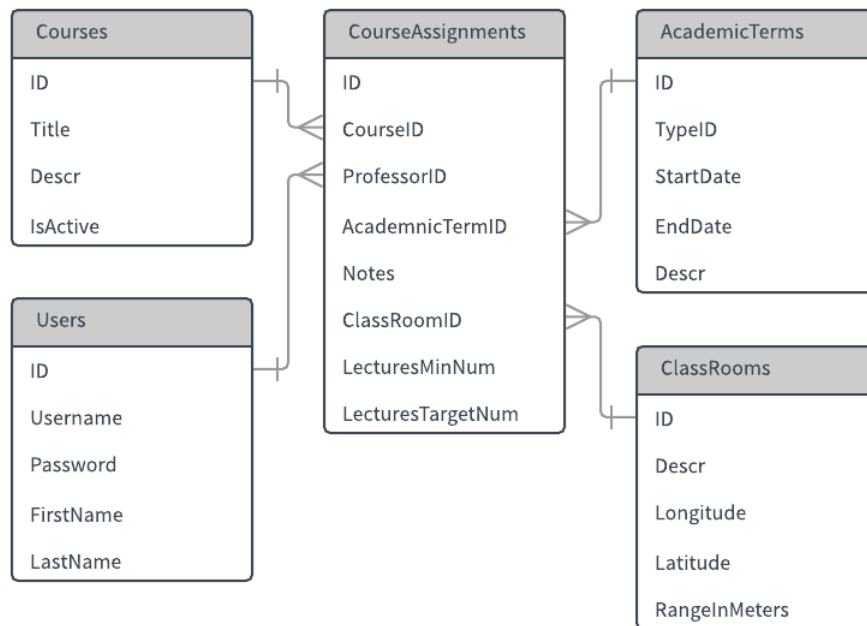


Figure 6-23: Relationships defined by the [CourseAssignments] table

6.3.3 Overview

Having presented and explained all the tables of the database schema, and the relationships between them in the previous sections, we are now going to illustrate an overview of the database schema, produced by SQL Server Management Studio (v17.8.1).

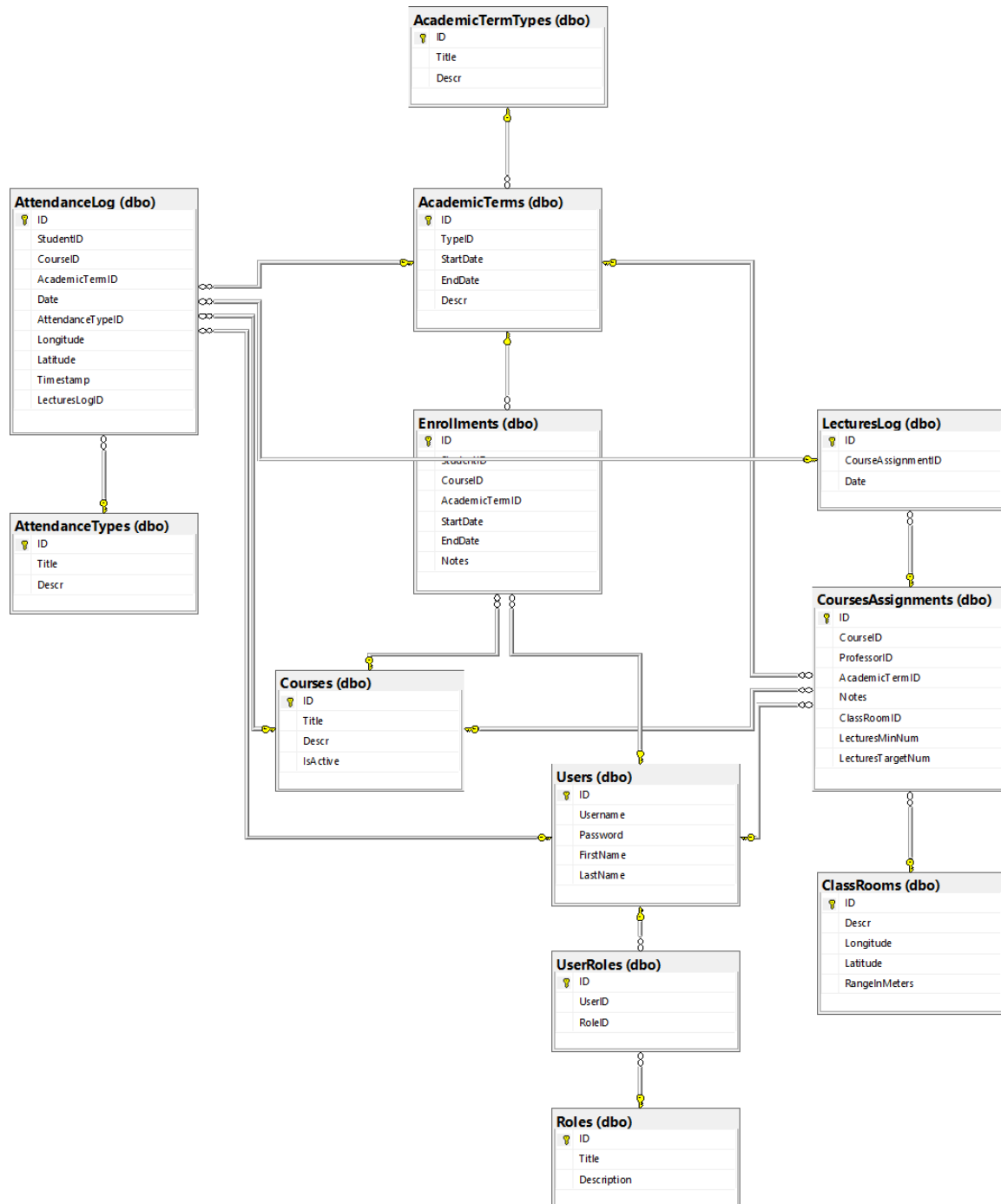


Figure 6-24: Database diagram

6.4 Application Program Interface (API)

As mentioned before, in the current chapter, the API is the middleware through which the database schema gets exposed to the consuming clients, the web based and the mobile applications. We could describe it as an abstraction of the underlying infrastructure, providing simple function calls in order to execute various tasks.

For example, an API can expose a function called `addUser` that takes specific parameters (username and password of the new user) and, when called by a client, it creates a new user to the database. The developer, who programs the client that calls the function, is not required to understand the way that this function works under the hood, in order to create a new user. This is called an abstraction.

6.4.1 Framework

The API is developed using the ASP.NET Core 2.0 framework alongside with the use of C# programming language. ASP.NET Core is cross-platform, meaning that it can run on various operating systems like macOS, Linux and, of course, Windows. It is also high performant and lighter than .NET Framework.

6.4.2 Architecture

The API was built based on the MVC architectural pattern, and more specific, ASP.NET Core MVC framework. The MVC pattern separates the application in to three different parts, the *Controller*, the *Model* and the *View*, as shown in figure 6-25.

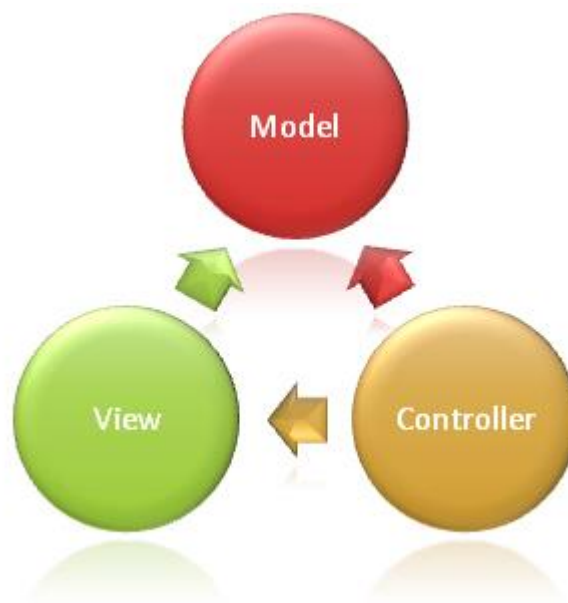


Figure 6-25: The MVC pattern (source: docs.microsoft.com)

Each of these parts has its own responsibilities. The *Model* is responsible for the application's state representation managing the data and business logic of the application. The *View* is responsible to present the final information to the user. Finally, the *Controller* is the module that

is responsible for receiving the data from the client (request), pass them to the appropriate *Model*, and, finally, picks up the appropriate *View* in order to respond to the client's request.

In our case, we do not need any views because we are developing an API that is not going to be, directly, rendered by any web browser. Instead, the API is going to be consumed, either by our web client or our mobile application.

6.4.3 Authentication and authorization

Authentication and authorization are crucial to the application security. They perform different tasks, but, combined together, offer protection against unwanted access by unauthorized clients to sensitive data.

Authentication is the process of confirming the truthiness of a claim by an entity (client) about specific data. For example, a user types his or her username and password, claiming to be the person who owns those credentials and is, previously, verified by the specific application.

Authorization is the process of specifying access rights to an entity against an application. For example, user A, may has the rights to edit his or her posts in a social media application, but user B does not have those rights.

In our case, we want, first, to authenticate, and, second, to authorize the authenticated user to use our application. As we saw in the database schema presentation, users are stored in [Users] table. This table has, among other, two fields, the *Username* and the *Password* fields. So, in order for a user to be authenticated, he or she has to provide a valid pair of username-password to the application. The following code snippet illustrates method `UserValidated()`, which searches to the database for a user that has the username and the password provided by the `LoginDetails` object.

```
1. private Users UserValidated(LoginDetails login) =>
2.     _context.Users.SingleOrDefault(
3.         x => x.Username == login.Username
4.         && x.Password == login.Password
5.     );
```

As soon as a user with the provided credentials is found in the database, the method returns this user to its calling method for further manipulation. This user is now considered **authenticated** by the application.

Having an authenticated user, we now have to determine the access rights of the specific user, or, in other words, we have to **authorize** the user with specific access rights. For the authorization process, we use *JSON Web Tokens*, or **JWT**³⁰.

According to the official site, JWT is an open source standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object.

The data shared between parties is encoded, digitally signed and, optionally, encrypted, so the information shared is verified and trusted. A JWT consists of three parts, a *Header*, a *Payload* and a *Signature*, separated by dots (.).

The *Header* contains information about the type of the token, in our case is *JWT*, and the algorithm that creates the signature. A header object that represents a JWT which uses the *HS256*³¹ algorithm to produce a signature is:

```
1. {  
2.   "alg": "HS256",  
3.   "typ": "JWT"  
4. }
```

The *Payload* hold the claims, information about an entity, like name, expiration time of the claim, subject, and arbitrary data.

Finally, the signature is produced by the encoded header, the encoded payload, a secret key and the algorithm specified in the header. The JWT can be expressed as shown in the following snippet (taken from the official JWT site):

```
1. HMACSHA256(  
2.   base64UrlEncode(header) + "." +  
3.   base64UrlEncode(payload),  
4.   secret)
```

As an example, a JWT that uses the HS256 algorithm and has the following claims,

```
1. {  
2.   "name": "John",  
3.   "age": "36"  
4. }
```

Finally looks like this (<https://jwt.io/#debugger>):

³⁰ <https://jwt.io/introduction/>

³¹ <https://community.auth0.com/t/jwt-signing-algorithms-rs256-vs-hs256/7720/2>

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bWl1IjoiaSm9obiIsImFnZSI6IjM2In0.vDrh29DpxIu-D73G5GW_UGktoyS3kKg8-CCwHiZEG-8

Now that we have presented a brief analysis of what a JWT is, we can illustrate the process of user authorization from a higher level.

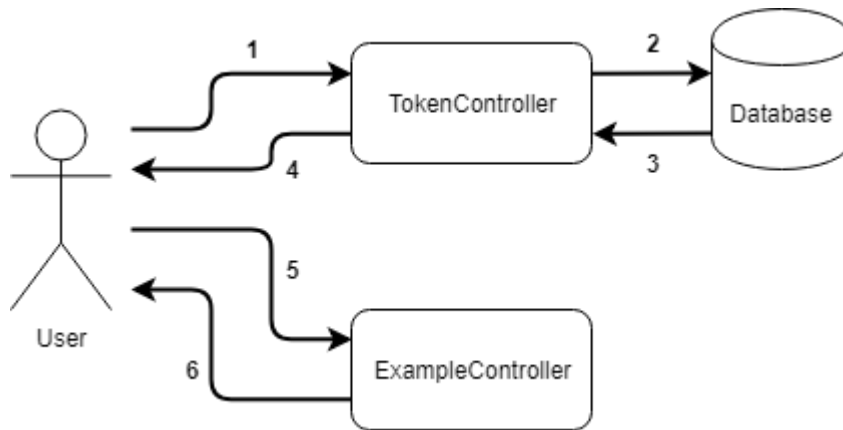


Figure 6-26: Typical authentication and authorization workflow

As illustrated in figure 6-26, the user has to be authenticated and then, authorized, in order to retrieve data from our API. The following steps are included in the authentication – authorization workflow of our API:

1. The user provides valid credentials (username and password) and a request is sent to the *TokenController*
2. The *TokenController* sends a request to the database to search for a user with the provided credentials
3. The database sends the results back to the *TokenController*
4. If the database has found a user with the provided credentials, the *TokenController* creates a new JWT and sends it back to the client (user)
5. The user makes a call to another available public method (*ExampleController*) of our API, providing the JWT received from the *TokenController*
6. The *ExampleController* validates the provided JWT, executes the requested task and responds with the appropriate message to the client

6.4.4 Attendance logs validation

In this section, we are going to describe the way that the application validates the incoming data, related to the attendance registration process. There are three conditions that have to be met in order for an attendance registration data set to be considered as valid:

- The user's device geographical coordinates have to be detected close enough to the classroom that the lecture takes place
- The registration attempt has to be executed inside a predefined amount of time from the beginning of the lecture
- The user that attempts to register an attendance for a specific lecture, has to be enrolled to this course

As shown in the next code snippet, `ValidateRegistrationData` method validates specific attendance registration data and return an object of type `AttendanceRegistrationDataValidationObject`.

```
• public static AttendanceRegistrationDataValidationObject
• ValidateRegistrationData
• (
•     AttendanceRegistrationData data,
•     attWebApiContext context,
•     IConfiguration configuration
• )
• {
•     var valid = true;
•     var error = string.Empty;
•     if (!LogHelper.ValidateNewAttendanceLog(data.Attendance, context))
•     {
•         valid = false;
•     }
•     else
•     {
•         // Validate location and timestamp
•         if (data.CoursesAssignmentID != null)
•         {
•             // Classroom's coordinates
•             var classRoom = (from c in context.ClassRooms
•                             join ca in context.CoursesAssignments
•                             on c.Id equals ca.ClassRoomId
•                             where ca.Id == data.CoursesAssignmentID
•                             select c).SingleOrDefault();
•
•             if (
•                 classRoom != default(ClassRooms)
•                 && classRoom.Longitude != null
•                 && classRoom.Latitude != null
•                 && classRoom.RangeInMeters != null
```

```

    )
    {
        double thisLon = (double)classRoom.Longitude;
        double thisLat = (double)classRoom.Latitude;
        var distanceBuffer = (double)classRoom.RangeInMeters;

        var minutesBuffer = configuration
            .GetSection("App")
            .GetSection("RegisterAttendance")
            .GetSection("TimeBufferInMinutes")
            .Value;

        var distance = LogHelper
            .DistanceInMeters
            (
                thisLat,
                thisLon,
                data.GeoLat,
                data.GeoLon
            );

        var enrollments = GetEnrollmentsForUserId
            (
                data.Attendance.StudentId,
                context
            );

        if (distance > distanceBuffer)
        {
            valid = false;
            error = @"Your device is detected
                too far from the classroom!";
        }
        else if (
            TimeSpan
            .FromTicks(
                DateTime.Now.Ticks
                - data.SessionStartTimestamp
            )
            .TotalMinutes
            > Convert.ToInt32(minutesBuffer)
        )
        {
            valid = false;
            error = $"It's been over ${minutesBuffer} minutes
                since the beginning of the course";
        }
        else if (
            !enrollments
            .Any(e => e.CourseId == data.Attendance.CourseId)
        )
        {
            valid = false;
            error = $"You are not enrolled to
                {context
                    .Courses
                    .SingleOrDefault(
                        c => c.Id == data.Attendance.CourseId
                    )
                }";
        }
    }
}

```

```

    •         }
    •     }
    •     else
    •     {
    •         valid = false;
    •     }
    • }
    • else
    • {
    •     valid = false;
    • }
    •
    • }
    • return new AttendanceRegistrationDataValidationObject {
    •     valid = valid,
    •     error = error
    • };
    • }

```

The following code snippet calculates the distance in meters between two points defined by their geographical coordinates (longitude, latitude).

```

1. public static double DistanceInMeters(
2.     double lat1,
3.     double lon1,
4.     double lat2,
5.     double lon2
6. )
7. {
8.     double theta = lon1 - lon2;
9.     double dist = Math.Sin(LogHelper.Deg2rad(lat1))
10.         * Math.Sin(LogHelper.Deg2rad(lat2))
11.         + Math.Cos(LogHelper.Deg2rad(lat1))
12.         * Math.Cos(LogHelper.Deg2rad(lat2))
13.         * Math.Cos(LogHelper.Deg2rad(theta));
14.     dist = Math.Acos(dist);
15.     dist = LogHelper.Rad2deg(dist);
16.     dist = dist * 60 * 1.1515;
17.     dist = dist * 1609.344;
18.     return (dist);
19. }
20.
21. public static double Deg2rad(double deg) =>
22.     (deg * Math.PI / 180.0);
23.
24. public static double Rad2deg(double rad) =>
25.     (rad / Math.PI * 180.0);

```

6.4.5 Testing

A very important part of the software development process is the testing process. There are many different kinds of test in software development like unit testing, integration testing, system testing, interface testing, acceptance testing and many others. In our development process, we

execute unit testing using a popular, open source unit testing framework for the .NET framework, **xUnit**.

xUnit provides the ability to create classes that contain several methods which perform various tests on specific methods of our application. The following code snippet shows the `LogHelperTest` class that has a method called `Test1`. `LogHelperTest` is responsible for testing methods from the `LogHelper` class of our application. `Test1` method is responsible for testing the `DistanceInMeters` method of the `LogHelper` class, so it calls it with predefined parameters and expects the result to be greater than zero.

```
1. public class LogHelperTest
2. {
3.     [Fact]
4.     public void Test1()
5.     {
6.         // UOM
7.         var lat1 = 40.625376;
8.         var lon1 = 22.960046;
9.
10.        // Home
11.        var lat2 = 40.596757;
12.        var lon2 = 22.986897;
13.
14.        // Calculated distance
15.        var actual = LogHelper.DistanceInMeters(lat1, lon1, lat2, lon2);
16.
17.        Assert.True(actual > 0);
18.    }
19. }
```

6.4.6 API documentation

In this section we are going to present the various calls exposed by our API, grouped by the controller that exposes them.

<i>AcademicTermsController</i>		
<i>Method</i>	Route	Description
<i>GET</i>	api/AcademicTerms	Get all the academic terms

Table 6-5: AcademicTermsController API

<i>AcademicTermTypesController</i>		
<i>Method</i>	Route	Description
<i>GET</i>	api/AcademicTermTypes	Get all the academic term types

Table 6-6: AcademicTermTypes API

<i>AttendanceLogController</i>		
<i>Method</i>	Route	Description
<i>GET</i>	api/AttendanceLog	Get all the attendance logs
<i>GET</i>	api/AttendanceLog/{id}	Get attendance log by id
<i>GET</i>	api/AttendanceLog/GetStudent/{id}	Get attendance logs for specific student id
<i>GET</i>	api/AttendanceLog/GetCourse/{id}	Get attendance logs for specific course id
<i>GET</i>	api/AttendanceLog/GetAcademicTerm/{id}	Get attendance logs for specific academic term id
<i>GET</i>	api/AttendanceLog/GetEnrollmentAttendanceLogs/{id}	Get attendance logs for specific enrollment id
<i>POST</i>	api/AttendanceLog	Add a new attendance log

<i>PUT</i>	api/AttendanceLog/{id}	Update attendance log by id
<i>DELETE</i>	api/AttendanceLog/{id}	Delete attendance log by id

Table 6-7: AttendanceLogController API

<i>AttendanceTypeController</i>		
<i>Method</i>	Route	Description
<i>GET</i>	api/AttendanceTypes	Get all the attendance types

Table 6-8: AttendanceTypesController API

<i>CoursesController</i>		
<i>Method</i>	Route	Description
<i>GET</i>	api/Courses	Get all courses
<i>POST</i>	Api/Courses/GetCurrent	Get all active courses for specific professor and academic term

Table 6-9: CoursesController API

<i>EnrollmentsController</i>		
<i>Method</i>	Route	Description

<i>GET</i>	api/Enrollments	Get all enrollments
<i>GET</i>	api/Enrollments/{id}	Get enrollment by id
<i>GET</i>	api/Enrollments/GetByUserId/{id}	Get enrollments by user id
<i>POST</i>	api/Enrollments	Add new enrollment
<i>PUT</i>	api/Enrollments/{id}	Update enrollment by id
<i>DELETE</i>	api/Enrollments/{id}	Delete enrollment by id

Table 6-10: Enrollments API

<i>QRCodeController</i>		
<i>Method</i>	Route	Description
<i>GET</i>	api/QRCode/{id}	Get QR code for specific course assignment id

Table 6-11: QRCodeController API

<i>TokenController</i>		
<i>Method</i>	Route	Description
<i>POST</i>	api/Token	Create a new token for authorized user credentials

Table 6-12: TokenController API

<i>UsersController</i>		
------------------------	--	--

<i>Method</i>	Route	Description
<i>GET</i>	api/Users	Get all users
<i>GET</i>	api/Users/{id}	Get user by id
<i>GET</i>	api/Users/Students	Get all users that have the role of student
<i>GET</i>	api/Users/Professors	Get all users that have the role of professor
<i>GET</i>	api/Users/ProfessorsStudents/{id}	Get list of students for specific professor and the current academic term
<i>GET</i>	api/Users/ProfessorsAssignedData/{id}	Get professor's assigned data (courses and students)
<i>POST</i>	api/Users	Add a new user
<i>PUT</i>	api/Users/{id}	Update user by id
<i>DELETE</i>	api/Users/{id}	Delete user by id

Table 6-13: UsersController API

6.5 Web application

In this section, we are going to present the *web application*, the application that, in combination with the API, presents the information stored to the database, to the final user via a web browser.

6.5.1 Framework

We chose to use one of the most popular front-end frameworks for the development of the web application, *React*³². React uses composable and reusable structures to construct the user interface (UI) and manage the application's state, called *components*. Components combine structure and logic into one logical piece. The structure is expressed using HTML and the logic is written in Javascript. The following snippet presents a piece of code from the main page of the web application.

```
1. loadingDiv = <div id="loadingDiv">
2.     <div id="innerLoading">
3.         <BeatLoader
4.             color={'#333'}
5.             loading={this.state.loading}
6.         />
7.     </div>
8. </div>;
```

We can see that HTML is used to express the structure of the component, while Javascript is used to handle the state. When rendered, this component produces a loader indicating that the page is loading some data at the moment.

Except from React, which is the main tool used for the development of the web application, other auxiliary tools were used to execute various tasks during the development process. Such tools are *Typescript*, *Yarn* and *NodeJS*.

*Typescript*³³ is an open source language, developed by Microsoft and is a superset of Javascript, as mentioned in the official site. We say that Typescript is a superset of Javascript because we can easily write pure (vanilla³⁴) Javascript in a Typescript file and still works perfectly right. This happens because, Typescript, when compiled, produces pure Javascript code. The aim of using Typescript is that, Typescript is *statically typed*³⁵, meaning that, the developer needs to define the type of the variable before he or she can use it, in contrast to Javascript, which is a dynamic typed language. This decreases the possibility of a potential error that could be thrown on run time, because Typescript code has to be compiled before use. The following snippet presents a case when the developer declares a variable, assigning a value of type `string`, and in the next line, the same variable gets a new value of type `number`.

³² <https://reactjs.org/>

³³ <https://www.typescriptlang.org/>

³⁴ https://en.wikipedia.org/wiki/Vanilla_software

³⁵ <https://www.sitepoint.com/typing-versus-dynamic-typing/>

```
1. var name = 'John';
2. name = 2; // Valid code in Javascript
```

While this is perfectly valid in Javascript, if we try to compile the above code from a Typescript file, we will get an error.

Another useful tool for the development process is *Yarn*³⁶. Yarn is a tool developed by Facebook and is used as a Javascript package manager. A package manager is a software that is responsible for installing, updating, removing, and many more, computer programs (packages). Another very popular Javascript package manager is NPM. We chose to use Yarn because it is faster and more reliable than NPM.

Finally, NodeJS is the base of our web application. Yarn runs on top of NodeJS. Our web application uses NodeJS as a runtime to execute various tasks, but we are not going into further details.

6.5.2 Code sample

The main task of our web application is to request data from the server and display them to the user. The way that the application executes this task is with the use of the *fetch API*³⁷. The following snippet, presents the task of retrieving the QR code for a specific course assignment.

```
1. generateQRCode(row: any) {
2.   let headers = api.getHeaders(api.ContentType.text);
3.   this.setState({
4.     loadingQrCode: true
5.   });
6.   if (headers !== null && row.hasOwnProperty('CourseAssignmentId')) {
7.     fetch(`${Config.serverUrl}api/QRCode/${row.CourseAssignmentId}`, {
8.       method: "get",
9.       headers: headers
10.    })
11.    .then(response => {
12.      if (response.ok) {
13.        return response.text();
14.      } else {
15.        throw new Error("Something went wrong...");
16.      }
17.    })
18.    .then(text => this.setState({
19.      qrCode: text,
20.      viewQRCode: true,
21.      isLoading: false,
22.      loadingQrCode: false
23.    })))
24.    .catch(error => this.setState({ error, isLoading: false }));
25. }
```

³⁶ <https://yarnpkg.com/lang/en/>

³⁷ https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API

As you can see, we call the `fetch` API, passing the appropriate route of our API (`api/QRCode/{id}`), alongside with other parameters. When the server responds, we check the response object and display the QR code to the web browser.

Summarizing, we have seen the tools and technologies used to develop the web application. We have also presented sample code, illustrating the use of the *fetch API*, which is used in order to communicate with our API and retrieve data from the server.

6.6 Mobile application

The third of the three main component of our system is the mobile application. The other two are the API and the Web application. Mobile application, as well as the Web application communicates with the API in order to retrieve specific data stored in the database, in order to display then to the final user. The main difference between the mobile and the web application is the target device. As their names imply, each of them targets specific device category, mobile application is designed to run on mobile devices (smartphones, tablets, or any other device running Android), and, web application is designed to run on web browsers.

6.6.1 Framework

The framework used to develop the mobile application is, as mentioned earlier in this chapter, the *Ionic framework*³⁸. Ionic framework was selected because it offers a familiar development environment to developers closer to the Javascript ecosystem. Ionic uses the *Angular*³⁹ front-end framework, a framework mainly maintained by Google and is equally popular and powerful with React. In contrast to the web application, we use NPM to manage the Javascript packages of our mobile application.

While developing in Javascript language, we use the *Apache Cordova*⁴⁰ framework to build and run our mobile applications natively. Cordova is a mobile application development framework that makes it possible to develop applications using pure web technologies, like HTML, CSS and Javascript, and build a native mobile application, targeting specific platforms like Android, IOS, Windows phone etc.

³⁸ <https://ionicframework.com/>

³⁹ <https://angular.io/>

⁴⁰ <https://cordova.apache.org/>

6.6.2 Code sample

We mentioned that the main reason of choosing Ionic over other mobile application development frameworks is the ability to use the Javascript language. As well as in our web application, the most common task is the call to the API to retrieve specific data. While, in the web application, we write code in Typescript, here, in the mobile application we write code in plain Javascript.

The following code snippet illustrates the way we call the API in order to retrieve a valid token to authorize the user. Instead of using the fetch API (we could use it), we used the post method, passing the API route (api/Token), alongside with the body and the appropriate headers. If the credentials provided by the user are valid, a new token is generated from the server and sent to the client (mobile application). Having a valid token, the user is redirected to ScanPage.

```
1.  this.http.post(  
2.    `${Config.serverUrl}api/Token`,  
3.    JSON.stringify(body),  
4.    {headers: headers}  
5.  )  
6.  .subscribe(data => {  
7.    if (data.ok === true) {  
8.      let jsonString = data.text();  
9.      let invalid = jsonString.replace(/\"/g, '')  
10.         === 'InvalidCredentials';  
11.      if (!invalid) {  
12.        let obj = JSON.parse(jsonString);  
13.        this.storage.set(`${Config.storageKeys.userDetails}`, obj);  
14.        this.navCtrl.setRoot(ScanPage);  
15.      } else {  
16.        alert('Not authorized');  
17.      }  
18.    } else {  
19.      alert('Error');  
20.    }  
21.    loading.dismiss();  
  });
```

We have seen that the development of a mobile application using the Ionic framework does not differ so much from the development process of the web application because we use the same programming language, Javascript.

6.7 Technical specifications

Having described the tools and technologies used in order to develop the three main components of our system (API, Web application, Mobile application), we are going to mention the minimum technical requirements that have to be satisfied in order for the system to be able to function correctly.

6.7.1 Mobile app

We have seen that the Cordova framework can target multiple operating systems and build a native mobile application, but, in our case, we have built a mobile application targeting the **Android** operating system only.

Another requirement is that the mobile device which is going to run our application should be able to connect to the **internet**, in order to communicate with the API. Another obvious reason of the necessity of an internet connection is the installation of the mobile application from the Play Store.

The main functionality of the mobile application is the scanning of the QR code that is going to be displayed on a big surface inside the classroom. For this reason, a mobile device with a **camera** installed is required.

6.7.2 Web application

The web application does not have to run on specific operating system, it just needs to be rendered by a web browser. As we explained earlier in this chapter, the web application is developed using, mostly, cutting-edge technologies like React. We have also used latest styling techniques (CSS3) to build a beautiful user interface. These technologies require a relatively **modern web browser** in order to be able to function correctly.

The web application, on the other hand, needs to be installed on a web server which has **NodeJS** installed. Our web application is using NodeJS to serve its files to the client's web browsers.

6.7.3 API

The API is a fully backend ASP.NET Core application that needs to be installed on a server in order to be able to handle clients' requests, sending back the appropriate response. The **ASP.NET Core Runtime** has to be installed on the server in order to be able to execute the API's executable files.

Except from the ability to execute the application, the server has to be able to serve the appropriate response to the clients' requests. In order to be able to respond to clients' requests, we need to have the **IIS**⁴¹ installed on the server.

⁴¹ <https://www.iis.net/>

Although ASP.NET Core is a cross platform framework, meaning that it can run on various operating systems, IIS needs to be installed on a server that runs the **Windows** operating system.

6.7.4 Database

The database of the attendance monitoring system is developed on MS SQL, so the server that hosts the database has to have the **SQL Server** installed on it.

6.7.5 Overview

Table 6-14 summarizes the minimum requirements that have to be met so that the attendance monitoring system can function properly.

<i>System minimum requirements</i>		
<i>Application</i>	Description	Versions
<i>Mobile</i>	Android	4.1 and above
<i>Mobile</i>	Device with camera installed and available internet connection	
<i>Web (front-end)</i>	Modern browser	IE9, Edge (all), Chrome (all), Firefox 3.5, Opera 11.5
<i>Web (back-end)</i>	NodeJs	6.9.1
<i>Application server (API)</i>	Windows	7 SP1
<i>Application server (API)</i>	ASP.NET Core Runtime	2.1

<i>Web server (API)</i>	Windows Server	2012
<i>Web server (API)</i>	IIS	8.0
<i>Database server</i>	SQL Server	2014

Table 6-14: System minimum requirements

6.8 Repositories

The whole system is available for everyone on Github in three separate public repositories:

- API [<https://github.com/kapantzak/AttendanceWebAPI>]
- Web Application [<https://github.com/kapantzak/AttendanceWeb>]
- Mobile Application [<https://github.com/kapantzak/AttendanceMobileApp>]

6.9 Summary

In this chapter, we tried to illustrate the development process of the attendance monitoring system. First, we explained the system's architecture, the main individual parts that compose the system, and how they communicate with each other.

Having understood the system's architecture, we presented the most important available technologies for us to develop the system. We have presented various programming languages and web development frameworks, comparing them, illustrating the pros and cons of each one of them. After a brief introduction of the most popular technologies in web development field, we have ended up with the final set of tools, the development stack.

With a specific set of development tools defined, we continued with the presentation of the application's database schema, the place where we hold the application's data. We, then proceeded with the illustration of the exported set of routes, available to any authenticated and authorized client, the API. After the presentation of the API's calls, we continued with the web application, the application that gives the ability to the users to access our system through a web browser. Alongside with the illustration of the above systems, we have provided various sample code in order to depict the development process.

Finally, we mentioned the minimum requirements that have to be met in order for our system to be fully functional and provided a centralized table with the system's technical specifications.

7 PROTOTYPE SOFTWARE

In this chapter, we are going to illustrate the user interface of the developed system. We are going to provide various screenshots related to specific use cases, like user login, attendance registration, courses overview and other.

7.1 Web application

We will start with the web application, which can be accessed using a, relatively, modern web browser. The first screen that, users view, when they open a web application, is, in most cases, a login screen. The same applies to our web application, users are prompted to type their credentials (username and password), as they exist in the database, as shown in figure 7-1.

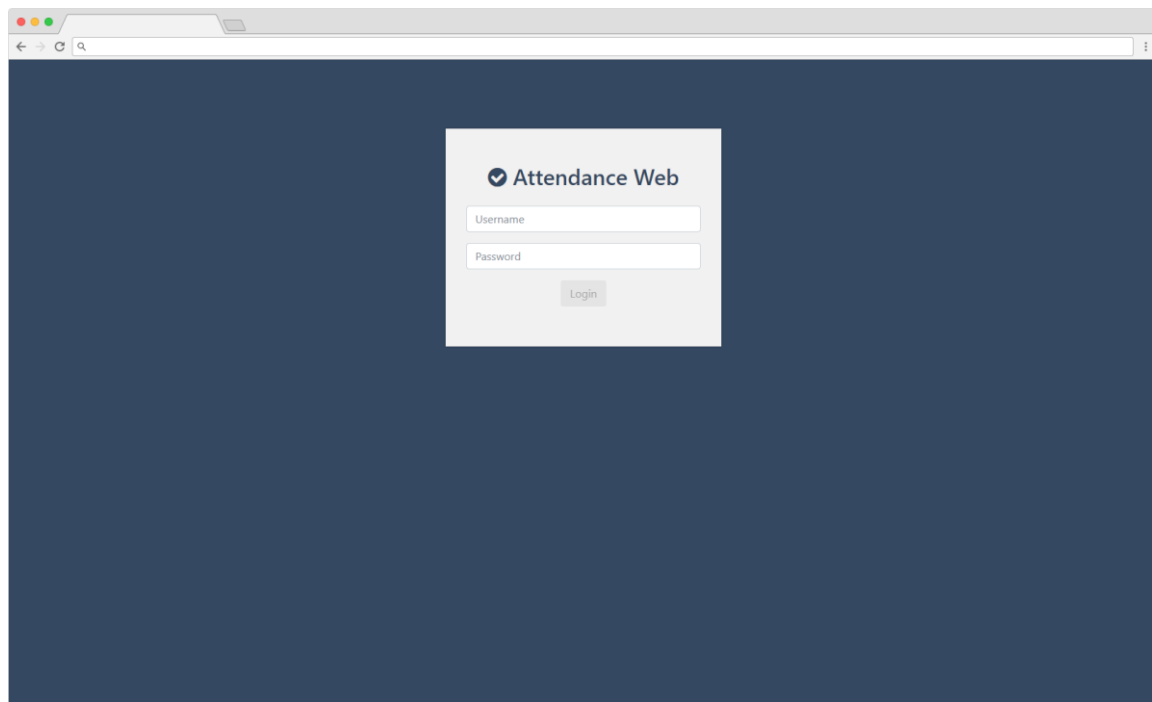


Figure 7-1: Web application - Login screen

7.1.1 User professor

As soon as users perform a successful login, they are redirected to the dashboard, a special screen which gathers various type of information, displayed in tables, charts and other forms. All users do not view the same information, instead, users are authorized to view specific information, according to their access rights, defined by the roles assigned to them. We are going to start with

the role of professor, that access users to view information related to students enrolled to courses that are assigned to them and other.

Figure 7-2 presents the dashboard screen of a user that has the role of professor. Professors view a list of courses assigned to them, and, in addition, a list of students enrolled to each of these courses, as shown in figure 7-3.

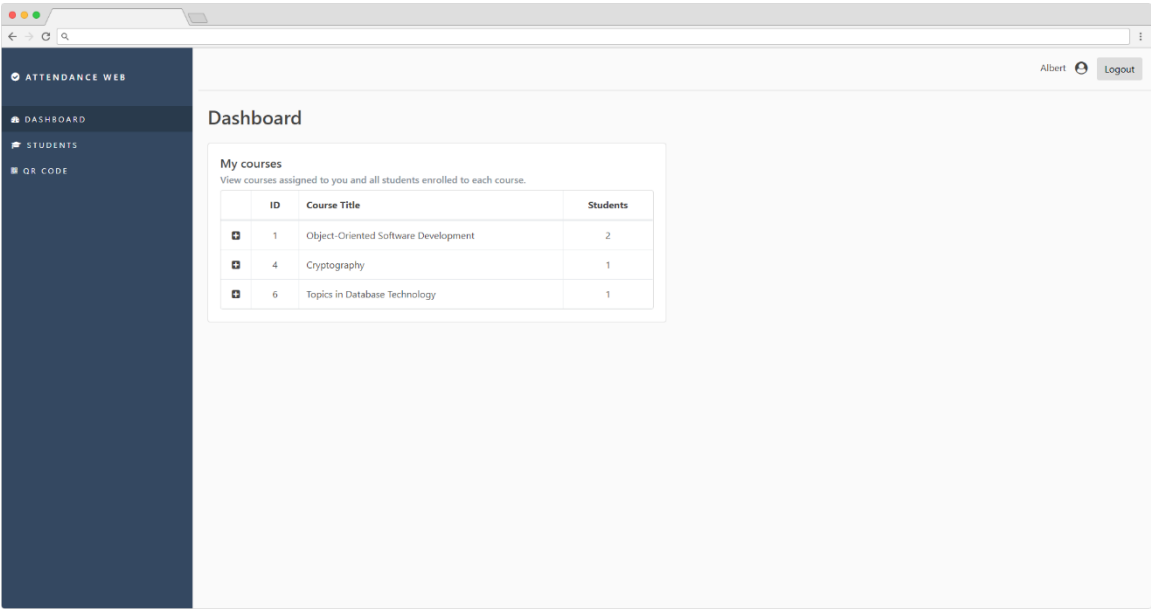


Figure 7-2: Web application - Professor's dashboard

Dashboard

My courses

View courses assigned to you and all students enrolled to each course.

	ID	Course Title	Students
⊞	1	Object-Oriented Software Development	2
	ID	First Name	Last Name
	2	Alexandros	Chatzigeorgiou
	3	Stylianos	Xinogalos
⊕	4	Cryptography	1
⊕	6	Topics in Database Technology	1

Figure 7-3: Web application - Professor's dashboard detail

Students screen, is a special screen, visible only to users assigned with the role of professor, that lists the total of students related to the specific professor through their enrollments. Figure 7-4 illustrates the *Students* screen.

ID	First Name	Last Name
2	Alexandros	Chatzigeorgiou
3	Stylianos	Xinogalos
9	Sofia	Petridou
11	Georgios	Evaggelidis

Figure 7-4: Web application - Professor's students list

Another screen, visible only to professors, is the *QR code* screen, where, professors can view a list of the courses assigned to them, as shown in figure 7-5.

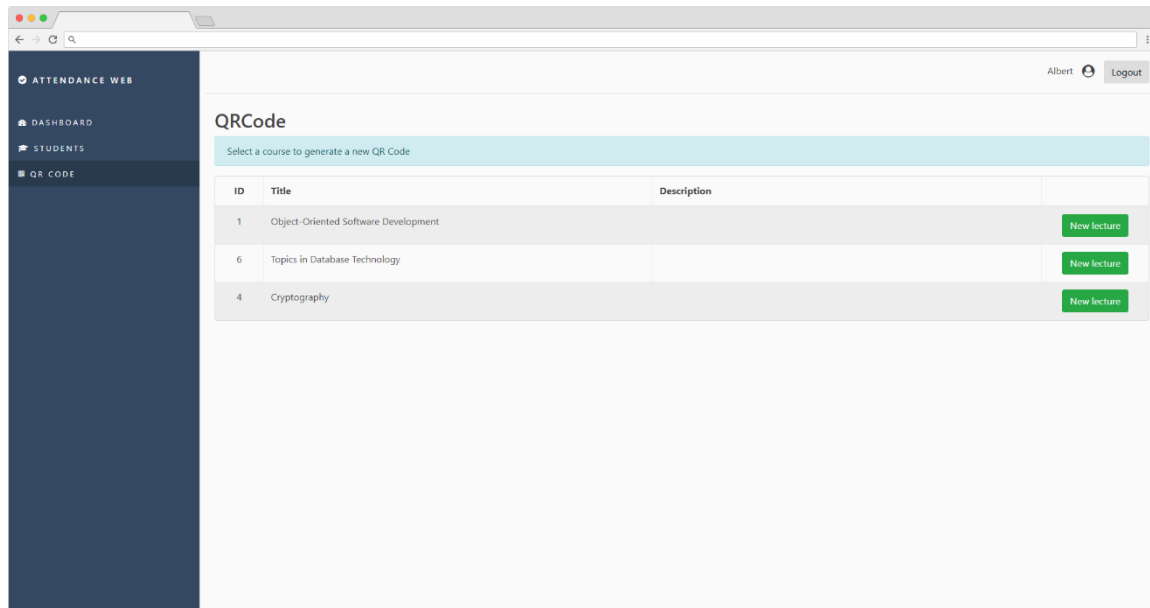


Figure 7-5: Web application - QR code generation screen

This is the place where, professors come to generate a new QR code for a new lecture of a specific course.

Figures 7-6 to 7-8 illustrates the process of a new QR code generation for specific course. Let's say that, professor A wants to start a new session for the course *Object-Oriented Software Development*. He, or she has to click the appropriate button, placed at the right of each course row.

ID	Title	Description	
1	Object-Oriented Software Development		New lecture
6	Topics in Database Technology		New lecture
4	Cryptography		New lecture

Figure 7-6: Web application - Courses assigned to professor

The first time the professors clicks the button, it turns to orange, indicating that it is going to generate a new QR code, and, therefore create a new record to the [Lecturers] table of the database, for the specific course assignment.

ID	Title	Description	
1	Object-Oriented Software Development		QR code
6	Topics in Database Technology		New lecture
4	Cryptography		New lecture

Figure 7-7: Web application - Professor selected course for QR code generation

If the professor clicks the orange button once more, it will generate a new QR code, as shown in figure 7-8.

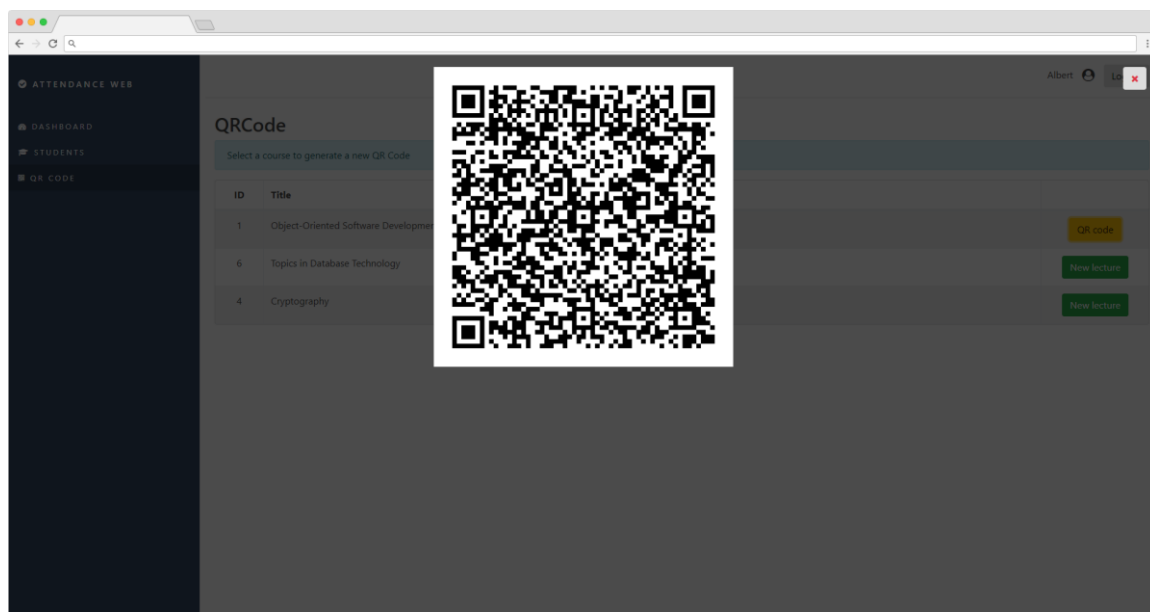


Figure 7-8: Web application - QR code generated for selected course

7.1.2 User student

We are, now, going to present the user interface that a user of type student will view, when using the web application.

The first difference, comparing to the professors' views, is the dashboard screen. Students are not allowed to view lists of students and their enrollments. Instead, students, view a list of their active enrollments. Furthermore, each item of this list, has a number, indicating the student's attendance logs for the specific course, as shown in figure 7-9.

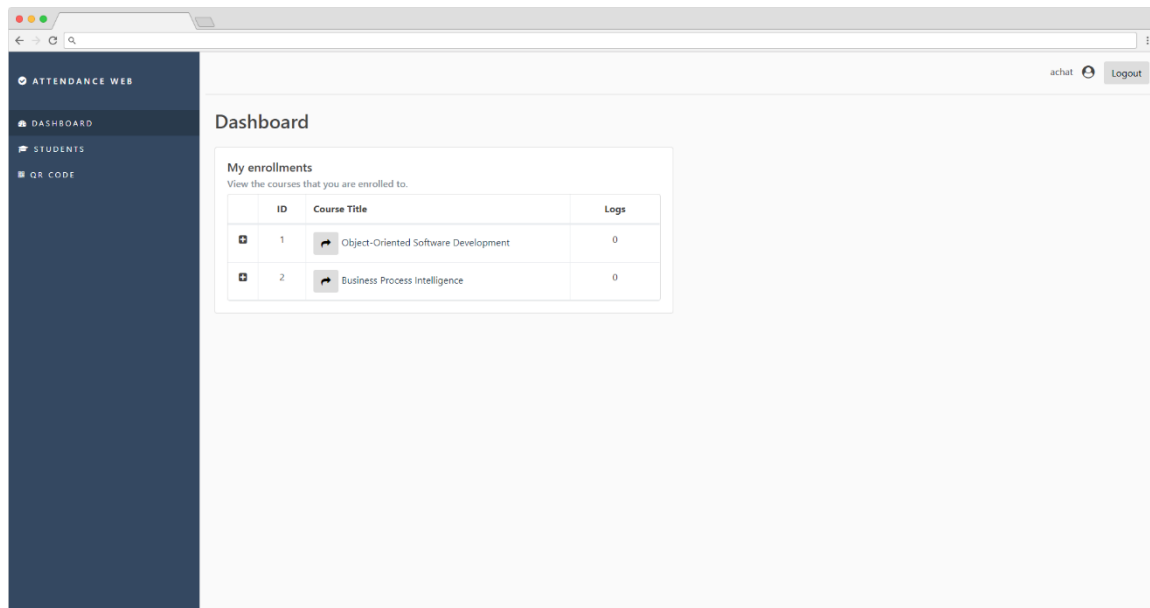


Figure 7-9: Web application - Student's dashboard

Each item of this list can be expanded, revealing a detailed list of the attendance logs for the specific course, as presented in figure 7-10.

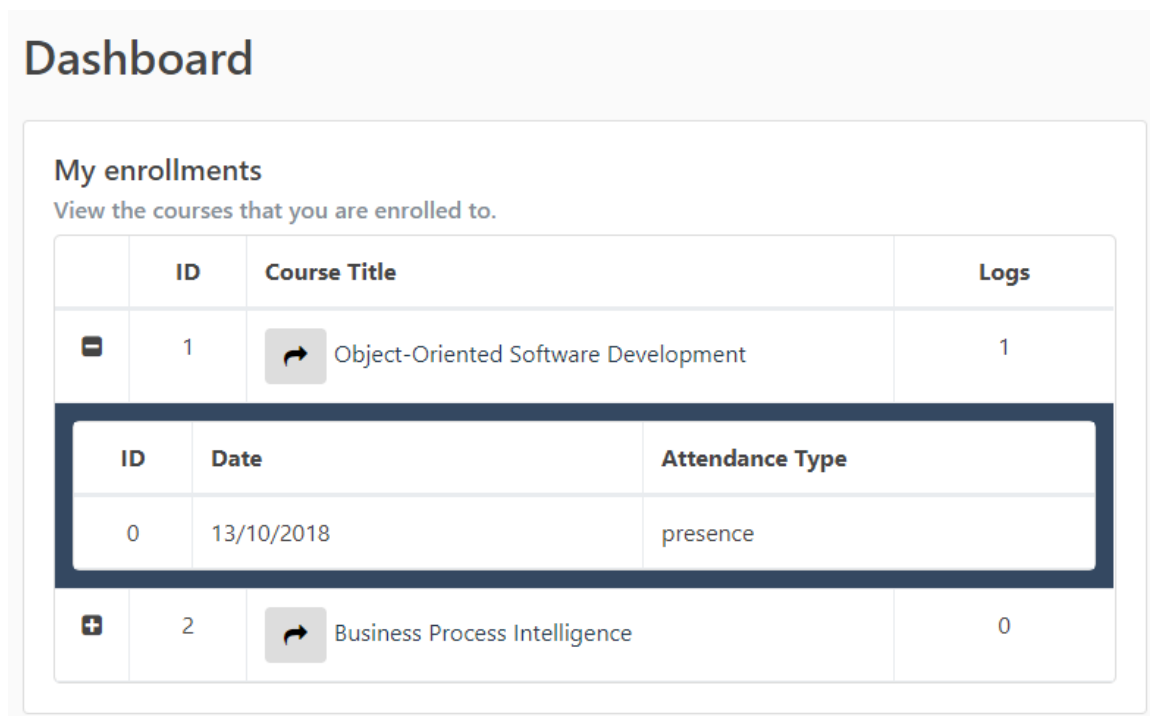


Figure 7-10: Web application - Student's dashboard detail

Except from viewing a detailed list of the attendance logs for each course, students can click on the arrow to the left side of each item in the courses list, navigating to an enrollment-specific

overview page, where they can view statistics about the number of attendances registered for the specific course and the absences for this course, as shown in figure 7-11.

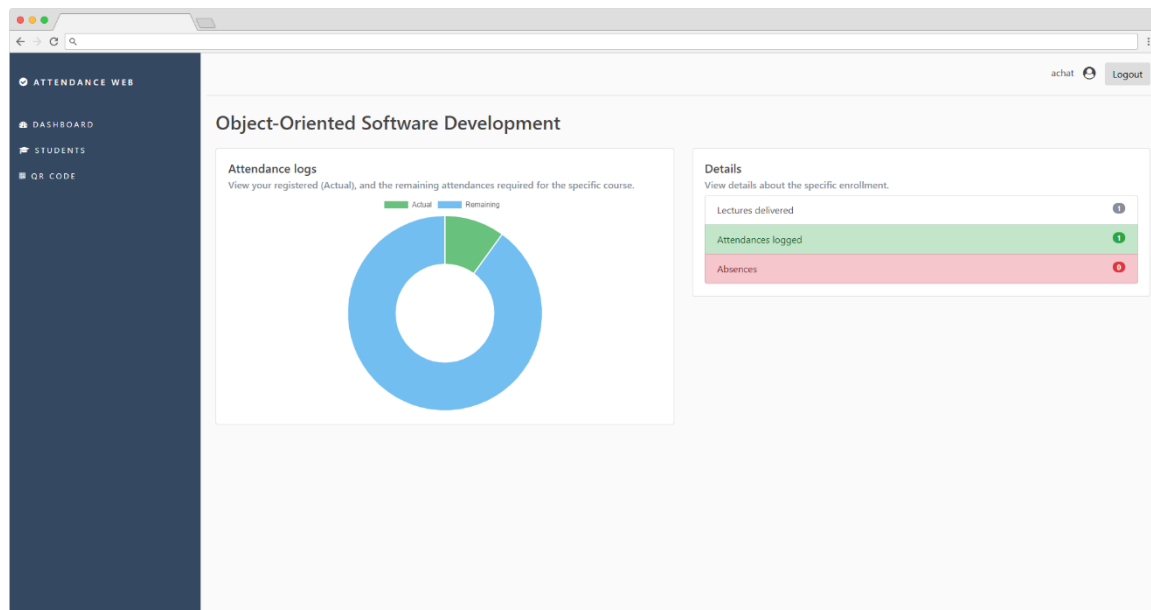


Figure 7-11: Web application - Student's attendance logs overview

7.2 Mobile application

In this section, we are going to illustrate the process of attendance monitoring, providing some screenshots from the actual mobile application. First, we are going to present the main screens of the application, explaining the tasks that the user can perform through each one of them. Then, we are going to provide various scenarios of unsuccessful attendance registration attempts, due to geographical limitations, time limitations and more.

7.2.1 Login

The first set of screens, displayed to the user, when the application launches, is the introductory screen, and then, the login screen, as shown in figure 7-12. Almost every application has this kind of screens in order to welcome the user and ask for credentials, like username and password, if needed.

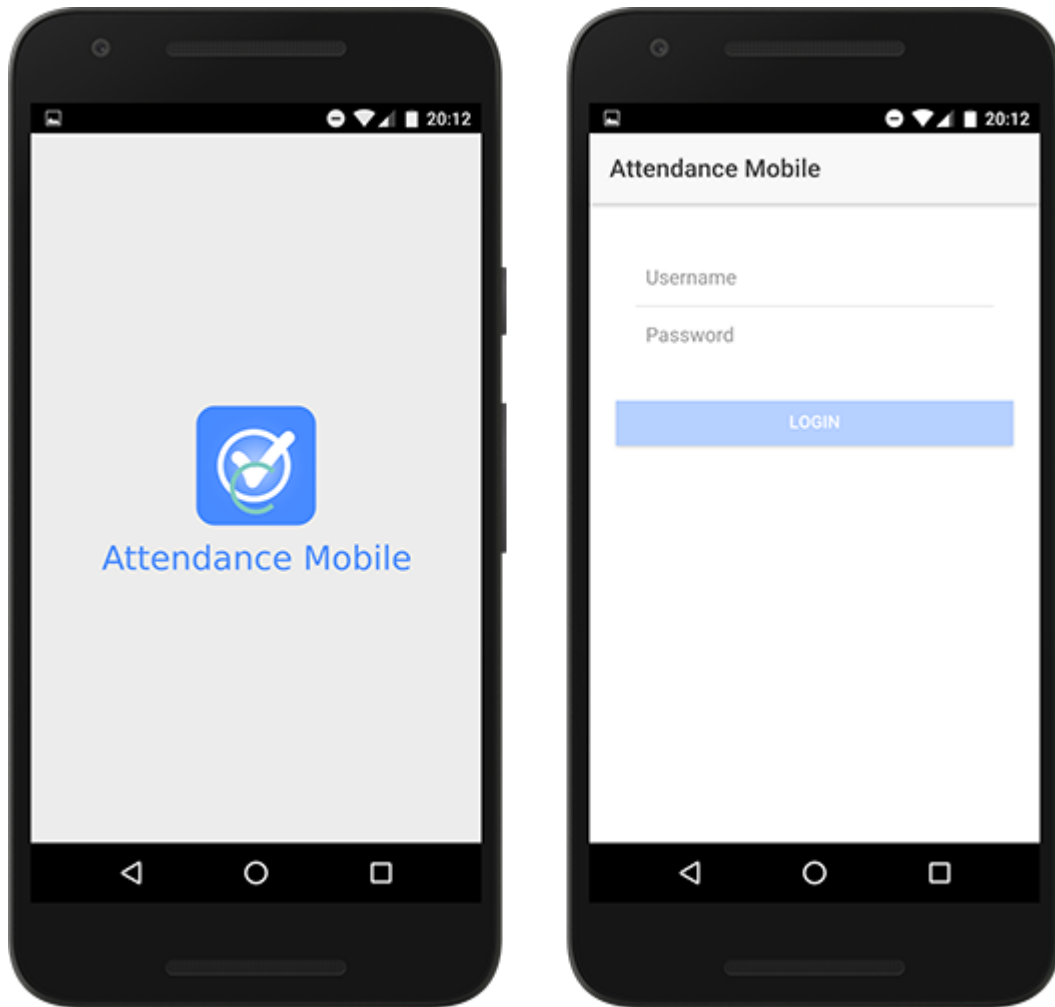


Figure 7-12: Mobile app - Login screen

If the device could not find an available internet connection, the application displays the appropriate message, instead of the login screen, as shown in figure 7-13



Figure 7-13: Mobile app - No internet connection message

7.2.2 Main menu

At the time of writing, the mobile application has only two main pages available, the *Scan* page and the *Enrollment* page. Figure 7-14 illustrates the main menu of the mobile application, that has three options, the two main pages mentioned before, and in addition, the *Logout* option.

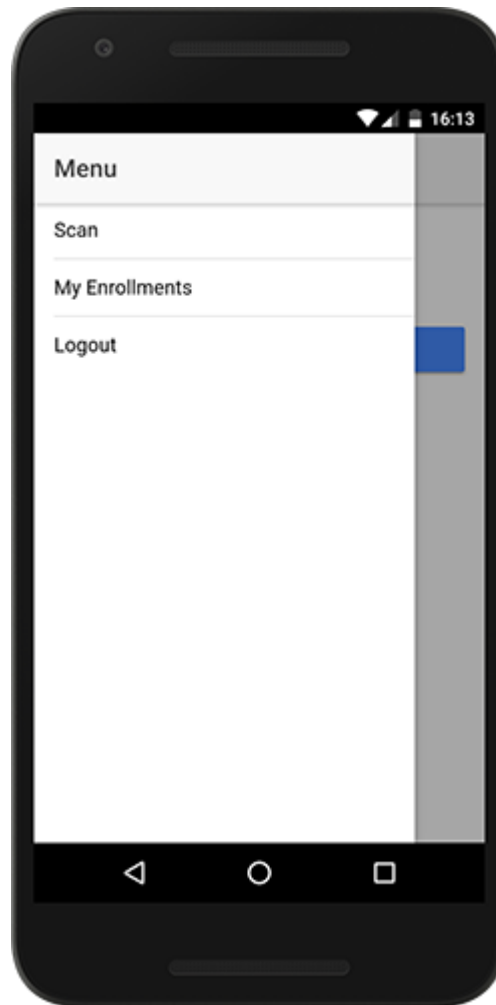


Figure 7-14: Mobile app - Main menu

7.2.3 Scan QR code

As soon as the user, achieves a successful login, the geolocation availability of the device is checked. If the device is not able to use the geolocation feature, the following message is displayed to the user:

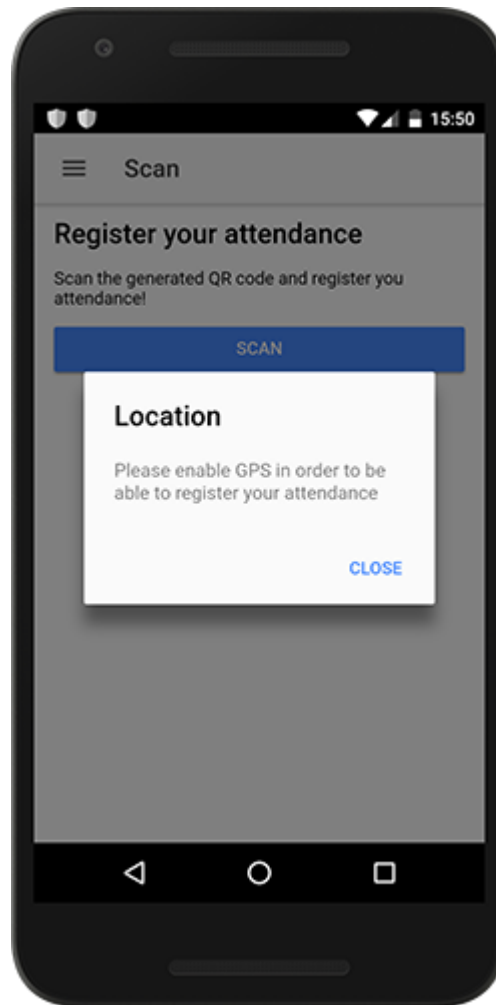


Figure 7-15: Mobile app - The device asks form the user to enable the geolocation feature

As soon as the user enables the GPS, the application executes a redirection to the scan screen. Here, the user is prompted to access the device's camera in order to scan the QR code. When the user presses the *Scan* button, the device's camera gets activated in order to read the QR code. When the user turns the device towards the QR code, the camera reads the data and, after some validations, asks the user to confirm the registration. If the user clicks *Yes*, the mobile application sends the data to the API, the API performs the registration and responds with the appropriate message of success or failure. This process is illustrated in figure 7-16, as a series of screenshots of the actual scanning process.

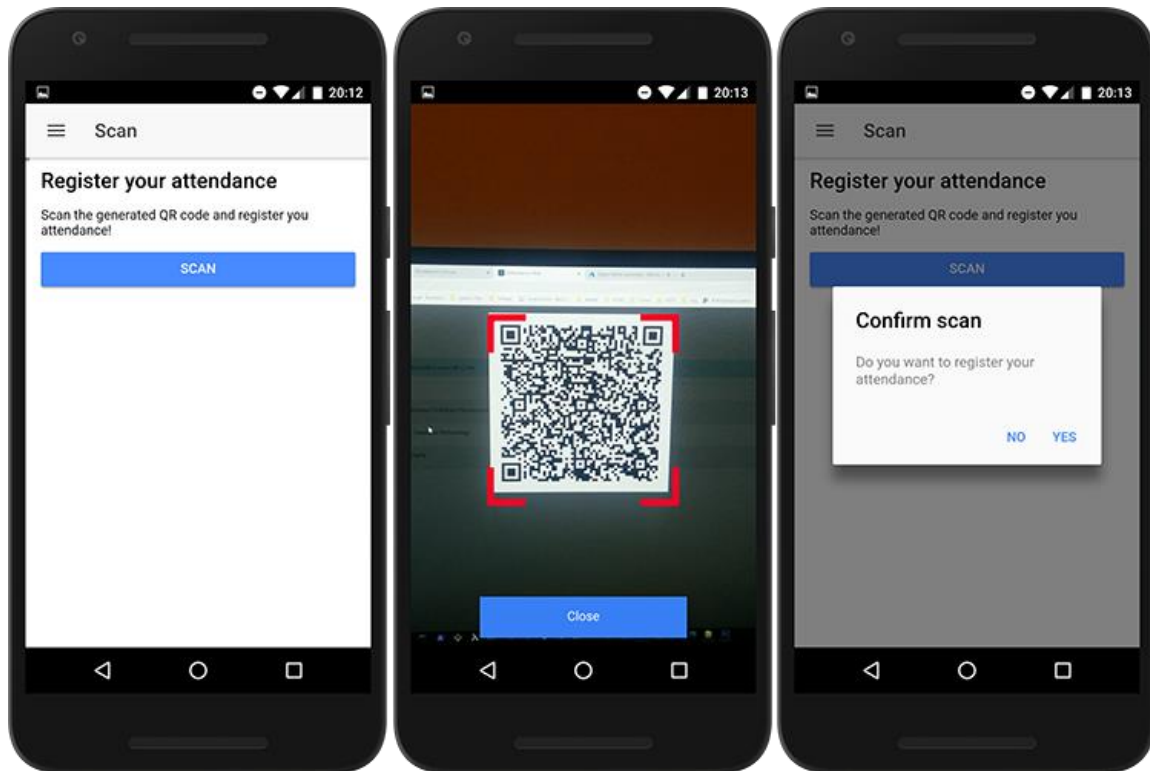


Figure 7-16: Mobile app - The scanning process

7.2.4 Enrollments overview

The user has the ability to view a list of active enrollments, meaning the courses that the user attends at the current academic term, as shown in figure 7-17, at the left screen, with the title of *My Enrollments*.

In addition, the user can view details about each enrollment by clicking on the arrow at the left side of each item list of the enrollments list. The *Enrollment details* screen gets visible, displaying useful information about the selected enrollment. In *Enrollment details* screen, user can view the *Logs chart*, a donut chart displaying the number of actual (colored with green) attendance logs that the user has registered for the specific course, against the remaining (colored with blue) logs that the user has to register in order to successfully complete the specific course. Below the *Logs chart*, the user can view similar information expressed in absolute numbers. Finally, the *Info* section displays generic information about the course, like the course title, the date range that the course is active and other data.

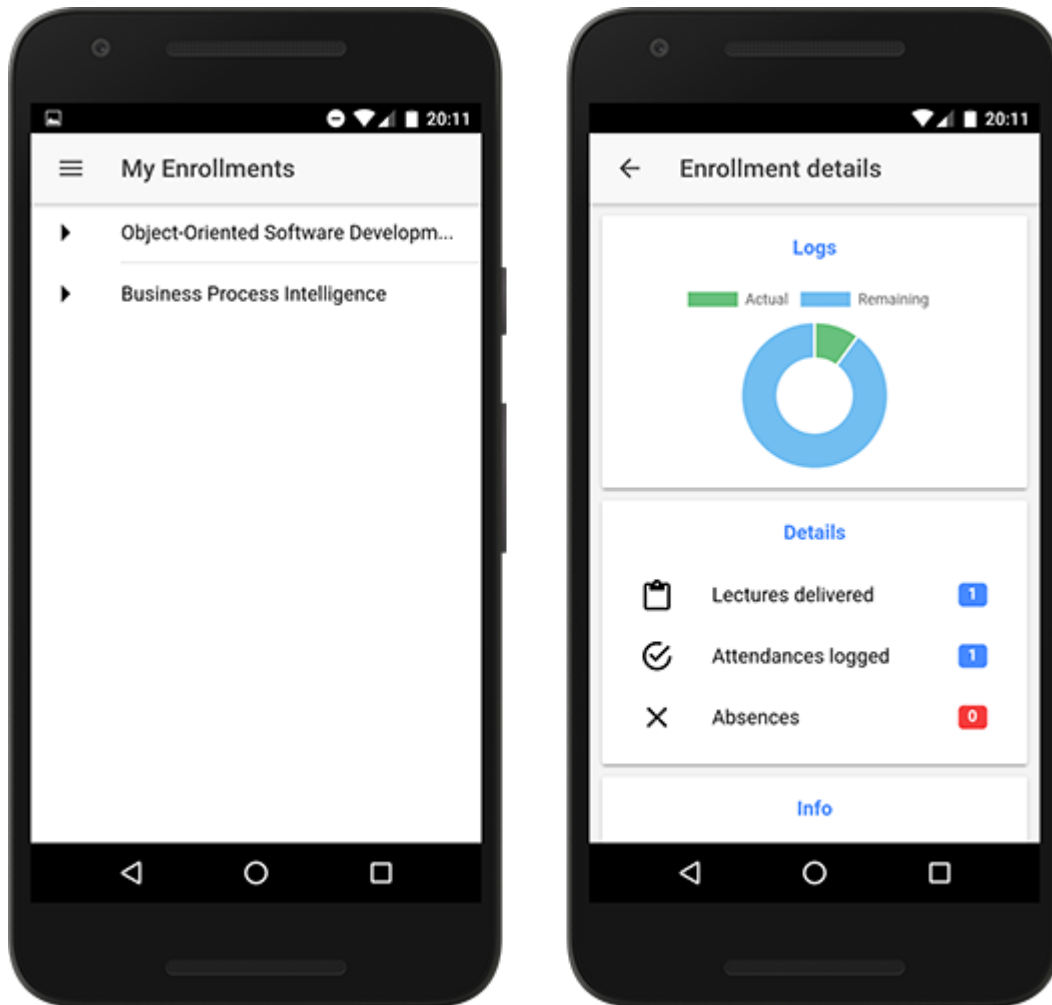


Figure 7-17: Mobile app - Enrollments overview and details

7.2.5 Registration data validation

As mentioned before, when the user scans the QR code, the API performs some validations on the incoming data. The API has to check three things in order to consider the registration as valid:

- User must be enrolled
- User has to be inside the classroom
- User has to be present at the time of QR code generation

7.2.5.1 Not enrolled

The first check that the API has to perform is that the user who requests to register an attendance log, for a specific lecture, has to be enrolled to the specific course. Figure 7-18 presents the

attempt of a user to register an attendance for the *Object-Oriented Software Development* course, without having been enrolled to this course first. The API responds with the following message:

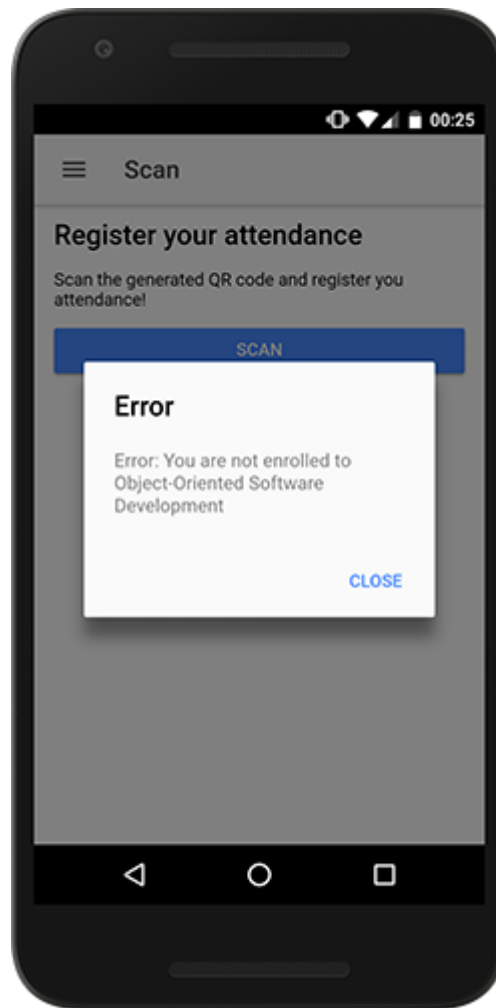


Figure 7-18: Mobile app - Error raised when user is not enrolled to the specific course

7.2.5.2 Out of geographical boundaries

The second check that the API has to perform, is the geographical location of the user's device at the time of registration attempt. Each classroom, stored in the application's database, is an entity that holds various information. Among others, a classroom's record holds a set of values that define the geographical coordinates of the classroom. Except from the coordinates, a classroom's record holds a number, indicating the radius in meters, from the point defined by the coordinates, shaping a geofence around the classroom's center. For an attendance registration to be considered as geographically valid, the user's device has to be detected inside this conceivable circle.

If the user's device is detected outside the predefined geofence, the application responds with the error message shown in figure 7-19.

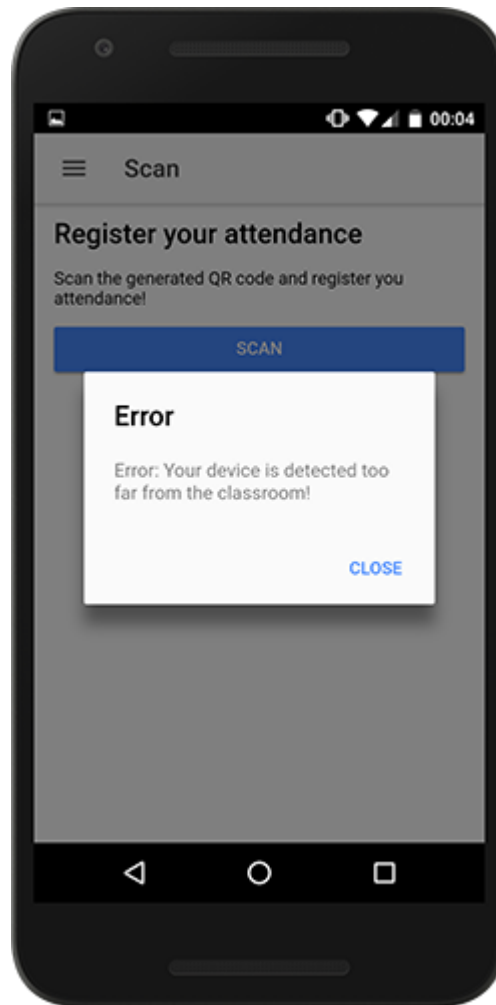


Figure 7-19: Mobile app - Error raised when user's device is detected too far from the classroom

7.2.5.3 Not in time

Finally, the API has to check the time that the user attempts to register an attendance. Someone could have taken a picture of the generated QR code, and gave that picture to another student who came to the classroom after the lecture ended, and tried to scan the picture at that time.

Every time a user attempts to register an attendance, the API checks that the difference of the current timestamp and the time of QR code generation does not exceeds a predefined amount of time, in our example this *buffer time* is 30 minutes, as shown in figure 7-20.

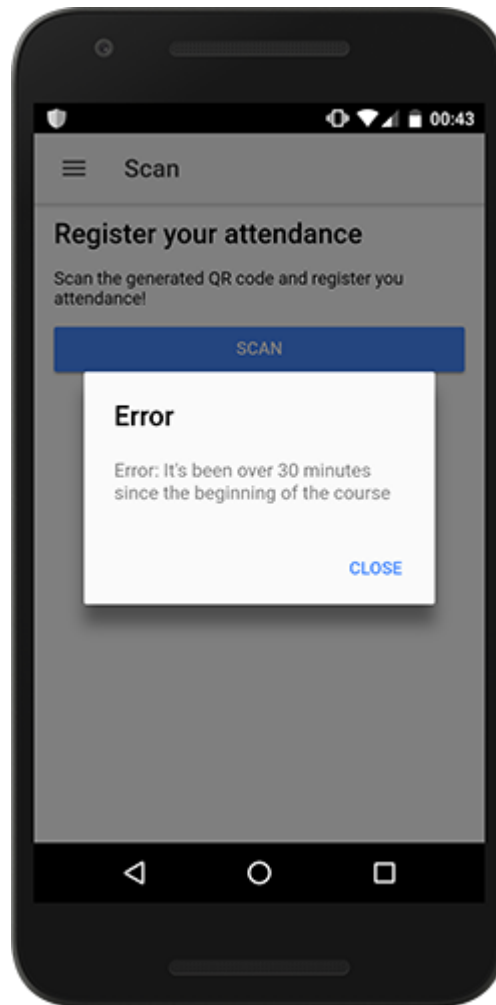


Figure 7-20: Mobile app - Error raised when user attempts to register attendance after specific time limit

7.3 Future work

Our system is, currently, performing the most basic tasks of a typical attendance monitoring system, as described in this chapter. The most important feature is the registration of the student's attendance during the lecture. Also, the student has the ability to view a list of the courses that he, or she is enrolled to, and, in addition, students can monitor their current status of lectures attended. From the side of the professor, users can view a list of all courses assigned to them, as well as a list of students enrolled to each of those courses.

Because this is a software developed for academic purposes, there is room for further improvements and additions. We are, now, going to present a list of potential improvements and new features that could add more value to our system, according to our opinion.

7.3.1 Configuration screens

Currently, the whole system is set up based on existing data in the database. Users, roles, courses, academic terms and all other entities of the system are initially inserted into the database *by hand*. By hand, we mean that we inserted the initial data using a database management system. But this is not useful at all to a typical, administrator, user of our system. Here comes the need of the addition of a set of new configuration screens, screens that would facilitate the task of the insertion, update and deletion of information in the database. So, for example, we could add a screen that handles the configuration of the system's users. The administrator of the system should be able to login to the web application and add new users and assign roles to them, or, add a new course, assign it to a professor and enroll students to this course, and many more.

7.3.2 Students' progress overview

Professors should be able to monitor the progress of their students. At the moment, professors can login to the web application and view a list of students enrolled to their assigned courses, but nothing more. There should be an overview screen where, professors, could navigate and view various statistics about their students. There probably could be various charts displaying information like the percentage of students that have more than one absence, or the number of students for each course assigned to the professor and many more.

7.3.3 Manual attendance registration

Now, students have to scan the generated QR code using their device's camera. What if their device could fail to open the camera at the moment? There should be the ability to, manually register their attendance for the specific course. This could be achieved, either by adding a new screen to the web application, from where professors could, manually add an attendance to their students, or by a new page to the mobile application, from where students could send a notification to their professor in order to confirm their attendance.

7.3.4 Users' data security

At the moment, users' sensitive data are stored in the database in plain text format. This means that they do not get encrypted before they get stored in the database. This is not a good practice in production mode, but, this is not the case in our system. The developed attendance monitoring system should not be considered to be a ready-to-use in production software, instead, a lot of changes has to take place in order to secure the whole system.

8 SUMMARY

In this chapter, we are going to provide a summary of the current thesis, present the main contributions of this research, illustrate potential limitations and provide possible improvements on the developed framework.

8.1 Main remarks

In this section, we are going to present the main remarks of the current research, providing a summary of each chapter.

In chapter 2, we presented an overview of the available literature related to the field of attendance monitoring systems. In this chapter, we have studied the most important and innovative implementations in the field of attendance monitoring, and we have concluded that, most of the systems developed, require sophisticated hardware in order to function correctly, for example, many systems required the use of fingerprint scanners, or eye scan sensors.

In chapter 3, we gave a brief illustration of the traditional attendance monitoring process, examining two alternative versions of the process. In both versions, we have concluded that, the traditional, paper-based process of attendance registration, is a time-consuming process. We have explained that, the traditional process is a serial process, that requires from each participant to complete his or her task in order for the next participant to start his or her task. The serial nature of the traditional process, makes it time-consuming. Chapter 3 continues by proposing a new, improved attendance monitoring process, that will give the ability to the participants to operate in parallel, saving valuable time, especially in cases of classrooms with a large number of students.

In chapter 4, we tried to distinguish the attendance monitoring process from the other processes that take place during the operation of an educational institution. We have, also, tried to define accurate boundaries of the process, making it more obvious and separate from the rest processes. Then, we presented the main participants of the traditional process, and provided an in-depth illustration of the paper-based attendance monitoring process through various diagrams and figures. The detailed illustration of the manual process, made obvious many disadvantages of the traditional process, for example, the fact that the traditional system requires too many manual processes, makes it error prone, and time consuming.

In chapter 5, we presented the requirements of the proposed framework. The requirements presented are the following: automation, security, simplicity, identity and location verification

and equipment independence. After defining the requirements of the proposed system, we have continued with the illustration of the automated process, defining separate steps and the system's users. After a detailed illustration of the proposed system, we can conclude that this process, dramatically decreases the amount of time needed in order for the attendance registration process to be completed.

In chapter 6 we explained the system's architecture, meaning the parts composing it and the way they communicate with each other. We proceeded with the illustration of the development of each one of those parts, providing code samples and informative data tables, when necessary. Finally, we defined a list of technical requirements, of hardware and software, that have to be met in order for the whole system to be fully functional.

In chapter 7, we provided various depicts and illustrations of the developed system. We had an overview of the process in real world scenarios, illustrated through various screenshots, from the web application, as well as from the mobile application too. In this chapter we saw that the developed system can facilitate the process of attendance monitoring, providing easy-to-use user interfaces and various automations. Although an improved system, relative to the traditional process, the new system has the potential to be even better with the addition of various new features in the future.

8.2 Limitations of the framework

As mentioned in chapter 6, the mobile application, which is a main part of the developed framework, is developed targeting the Android platform. So, an obvious limitation of the system is the mobile device's operating system of each student. A student, for example, with a mobile device with iOS operating system will not be able to use the mobile application.

Another limitation of the framework is the actual existence of the mobile device. If a student does not have any mobile device, he, or she will not be able to register his, or her attendance through the automated system.

The availability of internet connection, is another one limitation of the system. The mobile application is not able to login without a working internet connection. Apart from the mobile application, the web-based application, will not be able to connect to the API and generate a new QR code for a lecture.

Finally, the QR code that is generated from the API and is displayed through the web application, has to be projected to a big surface, in order to be visible to the whole classroom, so that the

students can target their devices' camera and scan it. The availability of a projection device, connected to the professor's laptop, and a big surface, is another limitation of the proposed framework.

8.3 Future work

As presented to the previous section, there are various limitations regarding the use of the proposed framework. These limitations reinforce the view that there is room for improvement in the future.

A feature that could be very useful in the proposed framework, would be the ability of working of line, and, when available, the mobile application would synchronize the stored data with the system's database.

Another improvement of the system, would be the release of the mobile application to other platforms, except from Android, like iOS, Windows phone etc. In fact, it is already feasible, due to the mobile application development framework that we have used, Ionic. Ionic provides the ability to the developer, to produce executables for various mobile platforms.

8.4 Conclusion

This thesis proposed and developed a new, automated system that would facilitate the process of attendance monitoring of educational institutions. The proposed framework provides solutions to problems occurred by the traditional attendance monitoring process, mainly, regarding the amount of time spent in order for the process of attendance registration to be completed. The automation provided by the proposed framework, reduces the amount of time required to complete the process. It is, also, ensures that, tasks that, previously, have been executed manually and tended to be error prone, now would produce no errors.

References

- Ajinkya Patil, Mrudang Shukla, 2014. Implementation of classroom attendance system based on face recognition in class. *Int. J. Adv. Eng. Technol.* 7, 974.
- Akshay A. Kumbhar, Kunal S. Wanjara, Darshit H. Trivedi, Anay U. Khairatkar, Deepak Sharma, 2014. Automated attendance monitoring system using android platform. *Int. J. Curr. Eng. Technol.* 4, 1096–1099.
- Arulogun, O.T., Olatunbosun, A., Fakolujo, O.A., Olaniyi, O.M., 2013. RFID-based students attendance management system. *Int. J. Sci. Eng. Res.* 4, 1–9.
- Dae-Jea Cho, Mi-Young Bae, 2014. A Study on Development of OTIP System using QR Code based on Smartphone.
- Dae-jea Cho, Young-wook Cha, 2015. Improvement of OTIP System using Location Information Based on Smartphone. *Int. J. Comput. Commun. Instrum. Eng.* 2. <https://doi.org/10.15242/IJCCIE.ER915203>
- DENSO, 2011. QR Code Essentials.
- Deugo, D., 2015. Using qr-codes for attendance tracking, in: *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp)*, p. 267.
- Fadi Masalha, Nael Hirzallah, 2014. A students attendance system using QR code. *Int. J. Adv. Comput. Sci. Appl.* 5, 75–79.
- Finkenzeller, K., 2010. *RFID Handbook. Fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*, 3rd ed. ed. Wiley, Chichester, West Sussex ; Hoboken, NJ.
- Gautam Shanbhag, Hussain Jivani, Sushil Shahi, 2014. Mobile Based Attendance Marking System using Android and Biometrics. *IJIRST-Int. J. Innov. Res. Sci. Technol.* 1, 87–90.
- Jain, A., Hong, L., Pankanti, S., 2000. Biometric identification. *Commun. ACM* 43, 90–98.
- Kadry, S., Smaili, M., 2010. Wireless attendance management system based on iris recognition. *Sci. Res. Essays* 5, 1428–1435.
- Makhtar, M., Rosly, R., Fadzli, S.A., Shamsuddin, S.N.W., Jamal, A.A., 2016. IMPLEMENTATION OF MOBILE ATTENDANCE APPLICATION USING GEO-FENCE TECHNIQUE. Faculty of Informatics and Computing, University Sultan Zainal Abidin, Tembila Campus, Terengganu, Malaysia, *ARNPJOURNALS*.
- Min Choi, Jong-Hyuk Park, Gangman Yi, 2014. Attendance Check System and Implementation for Wi-Fi Networks Supporting Unlimited Number of Concurrent Connections. *Int. J. Distrib. Sens. Netw.* 11, 508698. <https://doi.org/10.1155/2015/508698>
- Miran Hikmat Mohammed Baban, 2015. UNIVERSITY SEMINARS ATTENDANCE CHECKING SYSTEM USING QRCODE IMAGE SCANNER.

- O. Shoewu, O. A. Idowu, 2012. Development of attendance management system using biometrics. *Pac. J. Sci. Technol.* 13, 300–307.
- Parvathy, A., Rajasekhar, B., Nithya, C., Thenmozhi, K., Rayappan, J.B.B., Raj, P., Amirtharajan, R., 2013. RFID in cloud environment for attendance monitoring system. *Int J Eng TechnolIJET* 5.
- Rajachandrasekar, R., Ali, Z., Hegde, S., Meshram, V., Dandapantula, N., 2010. Location-Based Query Processing: Sensing Our Surroundings. *Dep. Comput. Sci. Eng. Ohio State Univ.*
- Rajan Patel, Nimisha Patel, Mona Gajjar, 2012. Online students' attendance monitoring system in classroom using radio frequency identification technology: a proposed system framework. *Int. J. Emerg. Technol. Adv. Eng.* 2, 61–66.
- Samuel A. Daramola, Morakinyo A. Adefuminiyi, Temitope M. John, 2016. Online Signature for Attendance Verification System using Levenberg-Marquardt Neural Network, Lecture notes in engineering and computer science. IAENG, International Association of Engineers, Hong Kong.
- Schiller, J., Voisard, A., 2004. *Location-Based Services*. Elsevier.
- Sheth, A., Seshan, S., Wetherall, D., 2009. Geo-fencing: Confining Wi-Fi coverage to physical boundaries. *Pervasive Comput.* 274–290.
- Shraddha S. Chawhan, Mangesh P. Girhale, Gunjan Mankar, 2013. *Mobile Phone Based Attendance System*, Wrox professional guides. Wiley Pub, Indianapolis, IN.
- Sultana, S., Enayet, A., Mouri, I.J., 2015. A Smart, Location Based Time and Attendance Tracking System using Android Application. *Int. J. Comput. Sci. Eng. Inf. Technol.* 5, 01–05. <https://doi.org/10.5121/ijcseit.2015.5101>
- Sumita Nainan, Romin Parekh, Tanvi Shah, 2013. *RFID Technology Based Attendance Management System*, McGraw-Hill networking professional. McGraw-Hill, New York.
- Swapnali Pawar, Priya, Komal Thorve, Urvashi, 2016. *Android Application for Attendance Monitoring System using Biometric*.
- TABASSAM NAWAZ, SAIM PERVAIZ, ARASH KORRANI, AZHAR-UD-DIN, 2009. Development of academic attendance monitoring system using fingerprint identification. *IJCSNS* 9, 164.
- Thomas Yeboah, Ing Edward Opoku-Mensah, Christopher Ayaaba Abilimi, 2013. *Automatic Biometric Student Attendance System: A Case Study* Christian Service University College.
- Tisse, C., Martin, L., Torres, L., Robert, M., others, 2002. Person identification technique using human iris recognition, in: *Proc. Vision Interface*. pp. 294–299.
- USA Dept. Of Defense, 2008. *GLOBAL POSITIONING SYSTEM STANDARD POSITIONING SERVICE PERFORMANCE STANDARD*.
- Virrantaus, K., Veijalainen, J., Markkula, J., Garmash, A., Katasonov, A., Terziyan, V., Tirri, H., 2001. *DEVELOPING GIS-SUPPORTED LOCATION-BASED SERVICES FOR M-COMMERCE: DREAM OR REAL CHANCE*. Lond. Univ. Exeter.

Yair Levy, Timothy J. Ellis, 2006. A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research.

Zatin Singhal, Rajneesh Kumar Gujral, 2012. Anytime Anywhere - Remote Monitoring of Attendance System based on RFID using GSM Network. publisher not identified, Place of publication not identified.