Software Requirements Specification

***Attendance registering system using face recognition***

Version 1.0

June 23 , 2020

By :

***Btoul Khadour***

***Ann Abdullah***

# Table of Contents

[Table of Contents i](#_Toc77487619)

[List of Figures ii](#_Toc77487620)

[1.0. Introduction 1](#_Toc77487621)

[1.1. Purpose 1](#_Toc77487622)

[1.2. Scope of Project 1](#_Toc77487623)

[1.3. Glossary 2](#_Toc77487624)

[1.4. References 2](#_Toc77487625)

[1.5. Overview of Document 2](#_Toc77487626)

[2.0. Overall Description 4](#_Toc77487627)

[2.1 System Environment 4](#_Toc77487628)

[2.2 Functional Requirements Specification 5](#_Toc77487629)

[2.2.1 student Use Case 5](#_Toc77487630)

[Use case : checking classes dates , pre-registering attendance 5](#_Toc77487631)

[2.2.2 teacher Use Case 6](#_Toc77487632)

[Use case : checking classes dates , confirm/reject registering attendance 5](#_Toc77487631)

[2.2.3 Sys-admin Use Case 7](#_Toc77487634)

[Use case: add/update 7](#_Toc77487635)

[Use case: delete 7](#_Toc77487635)

[2.3 User Characteristics 15](#_Toc77487648)

[2.4 Non-Functional Requirements 15](#_Toc77487649)

[3.0. Requirements Specification 17](#_Toc77487650)

[3.1 External Interface Requirements 17](#_Toc77487651)

[3.2 Functional Requirements 17](#_Toc77487652)

[3.2.1 student usage 17](#_Toc77487653)

[3.2.2 teacher usage 18](#_Toc77487654)

[3.2.3 Sys-admin adding object 18](#_Toc77487655)

[3.2.4 Sys-admin updating objec 19](#_Toc77487656)

[3.2.5 Sys-admin deleting objec 19](#_Toc77487657)

[3.3 Detailed Non-Functional Requirements 23](#_Toc77487665)

[3.3.1 Logical Structure of the Data 23](#_Toc77487666)

[3.3.2 Security 25](#_Toc77487667)

[Index 26](#_Toc77487668)

# List of Figures

[Figure 1 - System Environment 5](#_Toc77487669)

[Figure 2 - student use case 6](#_Toc77487670)

[Figure 3 - teacher use case 7](#_Toc77487671)

[Figure 4 - Admin use case 8](#_Toc77487672)

[Figure 5 - Sys-Admin use case](#_Toc77487672) 8

[Figure 6 - Sys-Admin use case 9](#_Toc77487672)

[Figure 7 - Logical Structure of the Data](#_Toc77487672) 14

# *Introduction*

## Purpose

The purpose of this document is to present a detailed description of the Attendance recording using face recognition System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the IT-Engineering department for its approval.

## Scope of project

The software system will be Attendance Recording System Using face Recognition for the department of the IT-Engineering . this system will be designed to maximize time utilization in each class ,

And minimizing the overall students’ Distraction in class

By automating the attendance registering action which otherwise will be taken manually and automatically setting the phone “for student ” on silent mood in the specific date and time of the classroom

More specifically , this system will provide an android application for both student and teacher , each with a different privileges .

The system will organize the dates for each teaching subjects for both students and teachers , allow the students to pre-register their attendance by as easy as taking a picture of their faces by their phones , allowing the teachers to confirm/reject the Attendance “ otherwise the system will automatically confirm all pre-registered students” and monitor the whole operation within their phones By looking at pictures taken by students in real time . GPS will be used to restrict students to be in a specific locations “ classrooms “ for the entire class duration .

Forms of the accepted students in the final exam will be automatically generated by this system as it provides a web application to mange the system controlled by Sys-Admin “or a DBA “ .

the system also contains database to store a list of student teachers classes attendance percentage of each student and classes dates .

## Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| IT | informatics technology |
| Database | Collection of all the information monitored by this system. |
| DBA | Database administrator a consoles the database |
| Os | Operating system |
| Gpu | Graphics Processing Unit |
| info | Information |
| Software Requirements Specification | A document that completely describes all of the functions of a proposed system and the constraints under which it must operate. For example, this document. |
| Stakeholder | Any person with an interest in the project who is not a developer. |
| User | Student or teacher |
| API | An application program interface (API) is a set of routines, protocols, and tools for building software applications. |

## Reference

<https://www.youtube.com/user/kudvenkat>

<https://github.com/ageitgey/face_recognition>

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

## Overview of Document

The next chapter, the Overall Description section, of this document gives an overview of the functionality of the product. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next chapter.

The third chapter, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product.

Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

# *2.0* *Overall Description*

## 2.1 System Environment

student

Teacher

Sys-Admin

Attendance registering

Web application

Attendance registering system using face recognition

Figure 1 - System Environment

Attendance registering system has three active actors and one controlling system.

The student ,or the teacher accesses the Attendance registering system through an android application. The Sys-Admin accesses / controls the entire system directly via a web application.

## 2.2 Functional requirements Specification

This section outlines the use case for each actor separately.

### 2.2.1 Student Use Case

Use Case : checking classes dates , pre-registering attendance

pre-registering attendance

Diagram :

Student

checking classes dates

Figure 1 system Environment

Figure 2 student use case XRef 3.2.1

**Brief Description**

The Student will be able to check his/hers classes dates , in addition to pre-registering attendance

**Initial Step-By-Step Description**

Before this use case can be initiated, the Student has already accessed application via username and a password

1. The student “ after logging in “ can surf his/hers classes dates
2. The Student presses a button to register his/hers attendance
3. The system confirms that the taken photo contains a face to recognize
4. The system sends the picture to the server for processing
5. The system sends the picture to the teachers application to comfier the attendance
6. The system marks the student’s attendance as approved/rejected

### 2.2.2 Teacher Use Case

Use Case : checking classes dates , confirm/reject registering attendance

Diagram

Teacher

checking classes dates

Figure 3 teacher use case XRef 3.2.2

reject registering attendance

confirm registering attendance

**Brief Description**

The teacher either confirms the attendance of students or rejects it

**Initial Step-By-Step Description**

Before this use case can be initiated, the Author has already logged int to the application using a username and a password.

1. The teacher “ after logging in “ can surf his/hers classes dates
2. The system sends the teacher all the confirmation requests from students
3. the teacher confirm all the request from students or confirm/reject marked students .

### Sys-admin use case

Use case : Add/remove students , adding/remove teachers , Add/remove classes

Add/remove classes dates

Diagram :

Figure 5 Sys-Admin use case

Figure 4 Sys-Admin use case XRef 3.2.3

Add/remove students

Sys-Admin

,adding/remove teachers

Add/remove classes

Add/remove classes dates

Update info

**Update Information use cases**

#### **Use case: Add/Update** student/teacher/class/class Date

**Diagram:**

Figure 5 Sys-Admin use case XRef 3.2.3

Sys-Admin

Update student/teacher/

class/class Date

**Brief Description**

The Sys-Admin enters a new student/teacher/class/class Date or updates information about a current student/teacher/class/class Date.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Sys-Admin has already logged int to the application using a username and a password and in the editing page.

1. The Sys-Admin selects to *Add/Update* student/teacher/class/class Date.
2. The system presents a choice of adding or updating.
3. The Sys-Admin chooses to add or to update.
4. If the Sys-Admin is updating an student/teacher/class/class Date , the system presents a list of student/teacher/class/class Date to choose from and presents a grid filling in with the information; else the system presents a blank grid.
5. The Sys-Admin fills in the information and submits the form.

The system verifies the information and returns the Sys-Admin to the web application main

#### **Use case: deleting** student/teacher/class/class Date

**Diagram:**

Sys-Admin

deleting student/teacher/

class/class Date

Figure 6 Sys-Admin use case XRef 3.2.3

**Brief Description**

The Sys-Admin deletes a student/teacher/class/class Date

**Initial Step-By-Step Description**

Before this use case can be initiated, the Sys-Admin has already logged int to the application using a username and a password and in the editing page.

1. The Sys-Admin selects student/teacher/class/class Date.
2. The system checks if the student/teacher/class/class is connected to an other object
3. The sys-admin confirms the deletion

## 2.3 User Characteristics

The user is expected to be an IT-Engineering student/Teacher and be able to use An android phone.

The main screen of Student’s application will a schedule of all the featured classes in a particular date and a button to “register attendance.”

The main screen of Teacher’s application will a schedule of all the featured classes in a particular date and a button to that sends the application to a screen of all the pre-registered students with two more features :

1 accept all

2 accept/reject marked students

The Sys-admin is expected to be IT specialist and to be able to keep maintaining the database.

The main screen of Sys-Admin web application will a list of all the functionalities of the system “Adding/ Deleting users “ .

The detailed look of these pages is discussed in section 3.2 below.

## 2.4 Non-Functional Requirements

The System will be on a server with high speed Internet capability. The physical machine to be used will be determined by the IT-Engineering Department . the use of an Nvidia GPU is highly recommended to speed up the algorithm that identifies faces in images.

The PC containing the system will have to be associated with SQL Server to run the database and running on windows OS.

# *3.0.* *Requirements Specification*

## 3.1 External Interface Requirements

The system will deal with three types of input :

First of all the sys-admin will add the info of the teacher (ID , first name , last name ,Email , phone number , device ID , user name and password ) , as you will see in the table in 3.3.1 .

Then the sys-admin also will add the info of the student (ID , first name , last name ,Email , phone number , device ID , user name and password ) , as you will see in the table in 3.3.1 .

and the third type will be the image that the students send to confirm attende , that image will be scanned and checked if it belongs to student in the DB or not .

And finally the system will export one type of output that contain the tables of attende for each class throughout the whole semester .

## 3.2 Functional Requirements

The Logical Structure of the Data is contained in Section 3.3.1.

### 3.2.1 student usage

|  |  |
| --- | --- |
| **Use Case Name** | Student |
| **XRef** | Section 2.2.1, student use case |
| **Trigger** | To register attendance the student must open the application in the exact date and time of the class |
| **Basic Path** | 1. The student sees the featured classes 2. The student presses the register attendance button 3. The student takes a picture 4. The system analyses the picture for a face in it 5. The system sends the picture to server to process 6. The server sends the student a confirmation message |
| **Alternative Paths** | In step 2 if the system did not find a face the system shows an error message and goes to application’s main  In step 5 if the system could not recognize the face in the taken image shows an error message and goes to application’s main |
| **Postcondition** | The attendance is taken |

### 3.2.2 Teacher usage

|  |  |
| --- | --- |
| **Use Case Name** | Teacher |
| **XRef** | Section 2.2.2, Teacher use case |
| **Trigger** | The teacher opens the application |
| **Precondition** | To comfier/reject pre- register attendance the Teacher must open the application in the exact date and time of the class |
| **Basic Path** | 1. The teacher sees the featured classes 2. The teacher presses the check attendee button 3. The teacher confirms / rejects students |
| **Alternative Paths** | In step 2 if the teacher ignored this step the system will automatically confirms all the pre-registered students |
| **Postcondition** | The attendance is recorded in the database |

### 3.2.3 Sys-Admin adding object usage

|  |  |
| --- | --- |
| **Use Case Name** | Sys-Admin |
| **XRef** | Section 2.2.3, Sys-Admin use case |
| **Trigger** | The Sys-Admin opens the web application |
| **Basic Path** | 1. The Sys-Admin opens the web application 2. The Sys-Admin selects add new object 3. The sys-admin selects what object to add 4. The system shows a list of field containing the info of the object 5. The sys-admin enters the info 6. The sys-admin hit the enter button |
| **Exception Paths** | If the sys-admin entered false info the system will shows an error message |

### 3.2.4 Sys-Admin updating object usage

|  |  |
| --- | --- |
| **Use Case Name** | Sys-Admin |
| **XRef** | Section 2.2.3, Sys-Admin use case |
| **Trigger** | The Sys-Admin opens the web application |
| **Basic Path** | 1. The Sys-Admin opens the web application 2. The Sys-Admin selects update object 3. The sys-admin selects what object to update 4. The system shows a list of field containing the old info of the object 5. The sys-admin enters the new info 6. The sys-admin hit the enter button |
| **Exception Paths** | If the sys-admin entered false info the system will shows an error message |

### 3.2.5 Sys-Admin deleting object usage

|  |  |
| --- | --- |
| **Use Case Name** | Sys-Admin |
| **XRef** | Section 2.2.3, Sys-Admin use case |
| **Trigger** | The Sys-Admin opens the web application |
| **Basic Path** | 1. The Sys-Admin opens the web application 2. The Sys-Admin selects delete object 3. The sys-admin selects what object to delete 4. The sys-admin hit the comfier button |
| **Exception Paths** | If the sys-admin tries to delete an object that connected to other objects the system will show an error message |

## 3.3 Detailed Non-Functional Requirements

### 3.3.1 Logical Structure of the Data

The logical structure of the data to be stored in the database is given below.

Figure 7 Logical Structure of the Data

Sys-admin

Teacher

Student

Regester

attendance

sent to

Confirmer/reject

Control

**student Data Entity**

|  |  |  |
| --- | --- | --- |
| **Data Item** | **Type** | **Description** |
| ID | int | The collage num |
| FirstName | Text | First name |
| LastName | Text | Lastname |
| Email | Text | Email address |
| PhoneNumber | Int | Phone number |
| DeviceID | Text | The id of the student’s phone |
| UserName | Text | Username to log name |
| Password | Text | Password to log in |

**Teacher Data Entity**

|  |  |  |
| --- | --- | --- |
| **Data Item** | **Type** | **Description** |
| ID | int | The collage num |
| FirstName | Text | First name |
| LastName | Text | Lastname |
| Email | Text | Email address |
| PhoneNumber | Int | Phone number |
| DeviceID | Text | The id of the student’s phone |
| UserName | Text | Username to log name |
| Password | Text | Password to log in |

The sys-admin is the only person with the privileges to enter the database ‘ sql server ‘

And the web application that’s why it has no entity in the data base

### 3.3.2 Security

The server on which the system resides will have its own security to prevent unauthorized *users* .

The sys-admin is the only person with the privileges to enter the database ‘ sql server ‘ and the web application .

The api layer will provide a high level of security .

The username and the password we entered for both (student and teacher) will secure the account’s for each (student / teacher) .