

# Specification of Requirements according to the IEEE 830 standard

IEEE Std. 830-1998

October 22, 2008

## **Summary**

This document presents, in English, the Software Requirements Specification (ERS) format according to the latest version of the IEEE 830 standard. According to IEEE, a good Requirements Document, although not mandatory to strictly follow the organization and format given in standard 830, should include, in one way or another, all the information presented in said standard. The IEEE 830 standard is not free of defects or prejudices, and therefore has been justly criticized by multiple authors and from multiple points of view, even questioning whether it is really a standard in the usual sense that the term has in other engineering. This document is not intended to pronounce itself either in favor or against one or the other: it only reproduces, for mainly teaching purposes, how a Document of Requirements according to the IEEE 830 standard would be organized.

# Index

Summary .....	1
1. Introduction .....	3
1.1. Purpose .....	3
1.2. Scope of the System' .....	3
1.3. Definitions, Acronyms and Abbreviations.....	3
1.4. References .....	4
1.5. Overview of the Documentum.....	4
2. Overview.....	5
2.1. Product Perspective .....	5
2.2. Functionis of the Product.....	5
2.3. User Characteristics .....	5
2.4. Restrictions.....	6
2.5. Assumptions and Dependencies.....	6
2.6. Future Requirements.....	6
3. Specific Requirements.....	7
3.1. External Interfaces.....	8
3.2. Functions .....	8
3.3. Performance Requirements .....	9
3.4. Design Restrictions .....	10
3.5. Totaxes of the System .....	10
4. Appendices .....	11

# 1. Introduction

When we talk about a class platform for teachers and students, information is needed about grades, NRC, subjects, attendance, records and class tutorials. In order to optimize and make these processes more comfortable, we seek to implement a platform that facilitates the connection between teacher and student.

## 1.1. Purpose

This document is aimed at the teacher who seeks to optimize the processes of their classes, tutorials and student grades as well as the records they will need, with their new needs to use a system programmed with their database.

## 1.2. Scope of the System'

In Class

- Optimize and implement new registration processes and the storage of student information, their classes, tutorials and grades administered by the teacher that will be required for future processes that allow providing the service of adding, deleting, editing classes, grades and tutorials.
- Improve the service for the entry and elimination of grades, classes and tutorials for teachers and students.
- This document is referenced in a previous document where the client gives us their project requirements in which what the client needs is explained in more detail.

## 1.3. Definitions, Acronyms and Abbreviations

### What is a Registry:

In computing, or specifically in the context of a relational database, a record represents a single data object implicitly structured in a table.

### What is a database:

A database is a tool for collecting and organizing information. Databases can store information about people, products, orders, or other things. Many databases start as a list in a spreadsheet or word processing program.

**What is a tutorship:**

Academic tutoring is a personalized accompaniment process that aims to improve school performance, solve school problems and develop study and work habits to avoid failure, falling behind and dropping out of school.

**1.4. References**

The references carried out in the project were only based on the interview with the client of the project based on user quotes.

Interview/InClass|DeveloperBears Retrieved November 13, 2022, from [Interview InClass | DeveloperBears](#) 🐻💻

**1.5. Overview of the Documentum**

A unique username and password were provided to the teacher with which the teacher will be able to enter the teachers' menu, he will only have access with the username and password mentioned above, in this way students will be prevented from entering the teacher's menu, from there You will be able to enter, edit and delete a course, grades and manage student tutorials, from which a record of them will be created.

On the other hand, students will have access to their own menu in order to enter edit and delete a tutorial where they are required to write their data (ID, NRC, career). Once what the teacher or student requires is recorded, a record is generated in the database.

## **2. Overview**

### **2.1. Product Perspective**

The In Class system will be totally independent from other products, since it is intended to be a complete application for our client, that is, we want to provide all the options to facilitate their work with the courses they teach, in the case after a student withdraws, they can be deleted, upload grades, add more courses or students, manage their information and also obtain pdfs that show the work done by the In Class system.

### **2.2. Functions of the Product**

This subsection of the ERS shows 'a summary, in broad strokes, of the functions of the future system. For example, in an ERS for an accounting program, this subsection shows 'for the system to support' account maintenance, shows 'the status of accounts' and facilitates 'billing', not to mention the enormous detail that each of these functions requires.

Functions should be displayed in an organized manner, and graphics may be used, provided that such graphics reflect the relationships between functions and not the design of the system.

The InClass system will allow the teacher to add courses, students with their names and ID, manage their information, delete it if deemed necessary. In addition, it will allow you to enter notes for homework, papers and exams in the three units, take attendance and generate pdfs of these two records. In the case of students, there is an option to request tutorials and have them registered, a pdf will also be generated with the time and all the information entered. All this to demonstrate compliance with the activities for both the teacher and her students.

### **2.3. User Characteristics**

This subsection will describe the general characteristics of the users of the product, including educational level, experience and technical experience.

The users of the InClass system will be teachers and students, in this case our client is the engineer Santiago Valencia who teaches Usability and Requirements classes at the University of the Armed Forces ESPE and the students of his classes will also be able to access the system, that is It is

important to mention that their students will be from the software engineering career.

## 2.4. Restrictions

The limitations that were imposed on the developers by the client are:

- **Connection with the academic system:** This is necessary since in this way the teacher can initiate a connection with the database and the platform in general to make the changes he needs.
- **Generate a file that can be uploaded:** The client refers to the fact that the platform, after making the creations or modifications, is capable of creating a file either in json, doc or pdf, in this way the teacher can upload them elsewhere since will have it as file.
- That students do not have access to the teacher's menu
- Security Considerations.

## 2.5. Assumptions and Dependencies

This subsection of the ERS describes those factors that, if changed, may affect requirements. For example, requirements may presuppose a certain organization of certain units of the enterprise, or they may presuppose that the system runs on a certain operating system. If these details change in the company organization, or if certain technical details, such as the operating system, change, it may be necessary to review and change the requirements.

The InClass system is designed for a single teacher, so if one of the future requirements is to add more teachers, some system features must be changed and others added to make it work as expected. In addition, if a percentage is considered for each grade, the system must be adapted so that the percentage of the final average is well calculated. Also if you want to add another member of the university as an inspector, you should design a menu for this member and restrict access to students and the teacher to this field, it is important to mention that you should add and adapt all the necessary functionalities to that more members can be added and the application works correctly.

## 2.6. Future Requirements

The possible requirement for the future is that the system could be connected to the Moodle application automatically and all the information entered can be uploaded, this so that the teacher's work with their students will be recorded.

### **3. Specific Requirements**

#### **1. Use of the InClass system**

##### **1.1 Teacher Menu:**

In this section it is important to know that the student will not have access to said menu, for this reason it was considered that the teacher had a username and password that has been provided by the programmers, its function is to only allow the teacher to enter your menu.

Considering that in the teacher's menu they will find that they can enter, add, delete and see the information of the students such as their grades, requested tutorials.

The system was designed with the goal of being fairly easy to understand and use, as well as maintaining a connection to the database for when information needs to be uploaded or downloaded.

##### **1.2 Student Menu:**

On the other hand, this section is aimed at students where they can see their notes, information, request, edit and delete tutorials that they consider necessary or not, where in order to request one they will be asked for their name, nrc, career, and the course of the subject you want to request a tutorial.

The same request that will be visible to the teacher and thus be aware of it. All this while working together with the database to always keep the requests at the moment they are made.

##### **1.3 Databases:**

The database is the place where the changes made to both menus for both teachers and students will be received, this will keep the information such as adding a tutorial, uploading notes, creating and deleting students constantly updated.

This makes it easier for both parties, since being a database, you can work remotely anywhere as long as you have a computer with an Internet connection.

### 3.1. External Interfaces

The database system must use the interface provided by MongoDB, with which all project data will be handled, saving them in documents belonging to well-defined collections.

### 3.2. Functions

This subsection (perhaps the longest in the document) should specify all those actions (functions) that the software must carry out. Normally (though not always), they are those actions expressible as 'the system owes 'to...'. If deemed necessary, graphic notations and tables may be used, but always subject to natural language, and not the other way around.

It is important to note that, in 1983, the IEEE 830 Standard stated that functions should be expressed as a functional hierarchy (in parallel with the Fingers proposed by structured analysis). But the IEEE 830 Standard, in its latest versions, already allows this subsection to be organized in multiple ways, and suggests, among others, the following:

- For types of User
  - The system must have a login method, this must identify between student-type users and teacher-type users in such a way that after validating the data, it shows exclusive functionalities for teacher-type users and in the same way for the student-type user, for this the institution ID and a password must be requested.
  - The system must only allow the teacher-type user to add, delete, modify and manage both courses and students for each course, as well as all other functions of the user, such as obtaining attendance records, grades, registered students and tutorial registration.
  - The system must allow students the possibility of requesting tutorials, the courses in which they are registered and will be able to obtain a record of the grades per unit and final grade of each course and only theirs.
- By Objects:
  - The system must have a Teacher object which must contain the following attributes: name, id, set of courses and a set of tutorials, it must be able to add courses, delete courses, obtain a tutorial record.
  - The system must have a Course object which has the following attributes name, nrc and set of students, here it must be possible



to add students, delete students and obtain a record of course grades and attendance.

- The system must have a Tutorial object in which the following attributes must be presented: date on which it was requested, id of the student requesting the tutoring, name of the student, degree course and name of the course.
- The system must have a Student object with the attributes, name, id of the institution, attendance record and grade record, here it must be verified if the student is approved or not.
- The system must have an object called the attendance record in which the total number of attendances and the number of classes taught will be counted. In this object, student attendance will be added and it can be modified.
- The system must have a qualification record object which will be composed of units, it must allow calculating the final average. - The system must have a Unit object which refers to each period taught in a semester, it has the following attributes workshops, homework, tests and exams. Here it should be possible to obtain the note per unit.
- The system must have a qualification object in which it is made up of the value of the qualification and the percentage of the total qualification valid to perform the average per unit, here it must be possible to add, modify qualifications and obtain the final grade for the unit.
- The system must have a connection to a database so that it can be used on different computers and the information is maintained

### **3.3. Performance Requirements**

The requirements related to the load that the system is expected to bear will be detailed. For example, the number of terminals, the expected number of simultaneously connected users, number of transactions per second that the system must support, etc.

Also, if necessary, the data requirements will be specified, that is, those requirements that affect the information that will be stored in the database. For example, the frequency of use, access capabilities and the amount of records expected to be stored (tens, hundreds, thousands or millions).

- The system in an initial phase must support a large number of students, although it is not defined, it must support at least 500

simultaneous students among all the added courses, however it is not expected to exceed that number in each period.

- The data system must be saved and have communication with a database for this, MongoDB will be used in which the teacher, course, student and tutorial collections will be created. And in each collection their respective documents.
- The data entered by the user must be saved under the following conditions, usernames can only contain characters of the alphabet, ids must have the following format L##### as an example L00385510, the nrc must be a number that goes from 1000 to 10000, and both the nrc and the Id cannot be repeated within the same course

### **3.4. Design Restrictions**

Anything that restricts the decisions related to the design of the application: Restrictions of other standards, hardware limitations, etc.

The program does not allow students to enter the teacher menu since it has a username and password that will be given only to the teacher.

In the same way, the program does not allow entering data in sections that do not correspond, such as letters and numbers.

### **3.5. Attributes of the System**

The quality attributes (the "ilities") of the system will be detailed: Reliability, maintainability, portability, and, very importantly, security. It should be specified which user types are authorized, or not, to perform certain tasks, and how security mechanisms will be implemented (for example, by means of a login and password).

The system is maintainable, portable and secure with the database to which it will be connected, since the teacher and student can use it by connecting from their computers as well as entering what they need from the system.

### **3.6. Other requirements**

The system must verify the compliance of the teacher's work with the courses for which he has been assigned to teach.

## **4. Appendices**

1. All the information referring to the data of the program, whether of teachers and students, their respective qualifications, tutorials, will be managed with a database, files.
2. With the time spent and all the functionality that was needed, an approximate amount of \$170 is estimated.
3. The Java programming language will be used mainly.