|  |
| --- |
| **Project: Admission Office Software** |
| **Requirement Analysis Document** |
|  |
| This project aims at providing PKFokam Institute of Excellence with a Software to support the management of the academic maters of the University |
|  |
| **Author: Mr. Herman Mekontso** |
| **Lecturer CS/IT – PKFokam Institute of Excellence, CS Reseacher** |
|  |



Table of content

[PART I: ACADEMIC REGISTRATION 4](#_Toc33822927)

[I.1. PROJECT OVERVIEW, PURPOSE, and SCOPE 4](#_Toc33822928)

[I.2. PROJECT DESCRIPTION 4](#_Toc33822929)

[1.2.1 Project context 4](#_Toc33822930)

[1.2.2 User characteristics 5](#_Toc33822931)

[1.2.3 Assumptions 5](#_Toc33822932)

[1.2.4 Constraints 5](#_Toc33822933)

[1.2.5 Dependencies 5](#_Toc33822934)

[I.3. REQUIREMENTS 5](#_Toc33822935)

[1.3.1 Functional Requirements 5](#_Toc33822936)

[I.3.2. Nonfunctional Requirements 6](#_Toc33822937)

[I.4. USE CASES 7](#_Toc33822938)

[I.4.1. Actors 7](#_Toc33822939)

[I.4.2. List of use Cases 8](#_Toc33822940)

[I.4.3. Use Case Diagram 8](#_Toc33822941)

[I.5. Sequence System Diagram 9](#_Toc33822942)

[I.5.1. Use Case: Create a new student 9](#_Toc33822943)

[I.5.2. Use Case: Search a student 10](#_Toc33822944)

[I.5.3. Use Case: Delete an existing student 11](#_Toc33822945)

[I.5.4. Use Case: Update details information of an existing student 11](#_Toc33822946)

[I.5.6. Use Case: Course Registration 13](#_Toc33822947)

[PART II : GRADE MANAGEMENT 14](#_Toc33822948)

[II.1. Introduction 14](#_Toc33822949)

[II.1.1 – Purpose 14](#_Toc33822950)

[II.1.2. – Scope 14](#_Toc33822951)

[II.2 – Use cases 14](#_Toc33822952)

[II.2.1. – Actors 14](#_Toc33822953)

[II.2.2 – List of use cases 15](#_Toc33822954)

[II.2.3 - Use cases diagrams 16](#_Toc33822955)

[II.2.3.1 - General Use case diagram 16](#_Toc33822956)

[II.2.3.2 - Administrator detailed use case diagram 16](#_Toc33822957)

[II.2.3.3 - Teacher and Students detailed use case diagram 17](#_Toc33822958)

[II.2.3.4 - Staff detailed use cases diagram 17](#_Toc33822959)

[PART III : ATTENDANCE MANAGEMENT 18](#_Toc33822960)

[II.1. INTRODUCTION 18](#_Toc33822961)

[III.2. PURPOSE 18](#_Toc33822962)

[III.3. SCOPE 18](#_Toc33822963)

[III.4. PROCEDURES AND PROCESSES 18](#_Toc33822964)

[III.4.1. Gathering data from the available time table 18](#_Toc33822965)

[III.4.2. Making sure students attend courses with all the prerequisites 18](#_Toc33822966)

[III.4.3. Providing a room for a lecture at the given time 18](#_Toc33822967)

[III.5. INPLEMENTATION OF THE PROCESSES IN THE CURRENT SYSTEM 18](#_Toc33822968)

[III.6. CURRENT SYSTEM PROBLEMS 19](#_Toc33822969)

[III.7. FEATURES OF THE SYSTEM TO BE DEVELOPED 19](#_Toc33822970)

[III.8. FEASIBILITY STUDY 19](#_Toc33822971)

[III.9. SYSTEM ANALYSIS 20](#_Toc33822972)

[III.9.1. USE CASE DESCRIPTION 20](#_Toc33822973)

[III.9.2. SYSTEM ACTORS 21](#_Toc33822974)

[III.9.3. SEQUENCE DIAGRAM 22](#_Toc33822975)

[PART IV: COURSE MANAGEMENT 26](#_Toc33822976)

[IV.1. Introduction 26](#_Toc33822977)

[IV.2. System Analysis 26](#_Toc33822978)

[IV.2.1. Actors 26](#_Toc33822979)

[IV.2.2. Functionalities 26](#_Toc33822980)

[PART V. PROGRAMS MANAGEMENT 27](#_Toc33822981)

[V.1. Introduction 27](#_Toc33822982)

[V.2. System Analysis 27](#_Toc33822983)

[V.2.1. Actors 27](#_Toc33822984)

[V.2.2. Functionalities 27](#_Toc33822985)

[PART VI: STAFF MANAGEMENT MODULE 29](#_Toc33822986)

[VI.1. Introduction 29](#_Toc33822987)

[VI.2. System Analysis 29](#_Toc33822988)

[VI.2.1. Actors 29](#_Toc33822989)

[VI.2.2. Functionalities 29](#_Toc33822990)

[PART VII: ADMINISTRATION MODULE 30](#_Toc33822991)

[VII. 1. Introduction 30](#_Toc33822992)

[VII.2. System Analysis 30](#_Toc33822993)

# PART I: ACADEMIC REGISTRATION

## I.1. PROJECT OVERVIEW, PURPOSE, and SCOPE

This sub-project aims at managing the academic registration of every student to the courses he/she is allowed to take for each new semester. The registration process will be done, for courses having one/more, depending on the grades he obtained in the prerequisite(s). In the conventional system, the details of the students are managed and a number of different applications are used to extract records from and add records to the appropriate file. But this schema has a number of major limitations and disadvantages such as data redundancy, data inconsistency, insecure data, incorrect data, time consuming, and many others. This results in an unsatisfactory registration system. The proposed system will have the capabilities allowing the administrator the ability for creation, modification, and other functions, of courses, lecturers, students and rooms by implementing the methods for reusability of the content.

The project is to maintain the student information. The problem faced by the organization earlier in the system used to maintain this information cause a lot of redundancy and inflexibility. In this project, JEE is used as backend for maintaining the information. This ensures flexible accessing to the database.

As noted above, the student entity is the core of the system, and it stores information about the students, lecturers, rooms, and courses offered by the university. It is through this selection that information is stored and accessed regarding students’ progress in the university.

The main benefits of this project are:

1. It helps in tracking students’ personal and educational information
2. The staff of the university can get the detailed information of various students registered in the school

The requirement of the user is to:

1. Login to the system via the first page of the application
2. Access/search information
3. Get help about the Academic Registration System how to use the different features of the system.

## I.2. PROJECT DESCRIPTION

### 1.2.1 Project context

This Academic Registration program is to be used by an administrator that has full access to the data and information of the software. There are also users with different access rights that will only be able to read and view the data. These are the other members of the administration, the teachers and the students. Then the product comes in place as an improvement tool for the management of the registration of courses of the students at the beginning of each semester.

This project is linked to other parts of a general architecture that is the secretary management. It is not independent as it uses the information from the Grade Management System to determine for example if a student has passed the prerequisite(s) of the course he wants to register or to determine the courses that a student has to reenroll. The system is interconnected with a main server that will store and make the data accessible to a wide variety of users with different access rights. A web interface is to be developed to give access to other users to the data they need to see.

### 1.2.2 User characteristics

The main user of the system is the secretary of the university. Dealing with documents and making report is her main activity. The secretary does not need a special competence to use the system. The general customer might need a previous experience with Microsoft Excel and Word or comparable software. The main user just need to be properly organized and attentive to the operations realized on the software.

### 1.2.3 Assumptions

The program is to be installed on a personal computer connected to a server for more security and easy data access. The computer should have an operating system of Windows XP and above and should support JEE. JEE is the main language used in the design of this program. JEE turns out to be the most suitable programming language considering the scope of the project. The system is to be used at the beginning of each semester to deliver the final printed documents. The data generated by the software will be accessed by intranet by any person for reading. This is the main reason it was imposed to be connected to a server.

### 1.2.4 Constraints

The main problem when implementing this system is the problem of levels (freshman, sophomores, etc) of each student. The system should be able to identify what course a student is allowed to register. To solve this information about the level of the student and the minimum semester at which a course can be taken should be entered.

### 1.2.5 Dependencies

* The data about the grade comes from the Grade Management System thus mistakes from that will propagate to the project;
* This project is dependent to the project Attendance management of the overall system;
* The data that will be used to compute the required documents are stored in the server. This one must be able to provide services at any time;
* The courses to be registered depend on the opened time table. A student will not register to a course if it is not available at the semester it should due to the lack of lecturer for example.

## I.3. REQUIREMENTS

### 1.3.1 Functional Requirements

1. The system should provide a security measure by requiring both a username and a password to avoid intrusion in the system;
2. The system should take new courses into consideration and provide a means to add it in the list of available courses;
3. The system should be able to register a course even though it shouldn’t (for example in the case where many students failed and the course is reopened the next semester);
4. The program must be able to retrieve the courses a particular student has already taken and those he hasn’t yet;
5. The system should be able to save the information about all the students and courses of the university;
6. The system should register a course if and only if the student passed the prerequisite if any;
7. The program should be able to print the necessary documents, in this case for example the registration file of every student;
8. The application should provide a read only interface for users other than the administrator;
9. The system should provide a Help screen to provide instructions on how to use the program.

### I.3.2. Nonfunctional Requirements

#### I.3.2.1 User interface

The client recommended making a simple user interface with light colors. This is therefore to be implemented and in which each operation must give a considerable feedback. The software must take a significant place on screen display. The user should be able to obtain results with the less amount of clicks possible.

#### I.3.2.2. Usability

The principal requirements concerning the usability are:

* The creation of a user manual coming along with the program and gives details about how to make any operation and what happen during each process;
* Giving a video as application tutorial to explain how the program is to be used.;
* Some indications are put on the different action buttons to express their utility;
* The error messages are to be as explicative and complete as possible for the sake of utilization easiness.

#### I.3.2.3. Performance

* An important number of persons can connect to the server to access the grades from the web interface. This means that the server will be highly demanded.
* The application can be used at any time by the administrator. There is no constraint and it can run for any time without perturbation on other activities done on the computer. The user asked the system to be usable at any time and in any condition.
* The program will be designed to work as fast as possible to save time to the secretary in her day to day work.

#### I.3.2.4. Maintainability

The software should be flexible enough to enable the addition of new functionalities or sub modules anytime it is needed.

#### I.3.2.5. Security

As explained before, a double authentication has been set to ensure a total security against intrusion. The server is already protected at his level. Physically, the server resides and the PKFOKAM Research center. Therefore, the main place of computation that is the secretary computer is the one concerned by this double authentication. It ensures total security as there is a personal code from the person using the software and a code generated by the software to ensure no data intrusion.

#### I.3.2.5 Back up system

The software should offer a way to automatically or semi automatically backing up the database.

## I.4. USE CASES

### I.4.1. Actors

The actors of our system are:

* The system administrator
* The students and the staff (with different profiles)
* The admission officer;
* The academic affair officer;
* The Dean of students.

#### I.4.1.1. The admission officer

This user has the right on all the elements generated by the system. He or she has the rights on all the information that are kept in the server and can access and modify them at any moment.

#### I.4.1.2. The Staff/Students

These users have no main action on the system but are capable of viewing information about the students, teachers, and courses of the university.

#### I.4.1.3 The academic affairs officer

The academic officer should be able to access all the information on students to be able to make a decision, such as validating a registration form, getting disciplinary information on a student, etc.

#### I.4.1.4. The Dean of Students

The Dean of students is in charge of the following up of students, mainly their discipline. He has to have a view on some information of students to be able to give them some advices and follow them up.

#### I.4.1.5. The system administrator

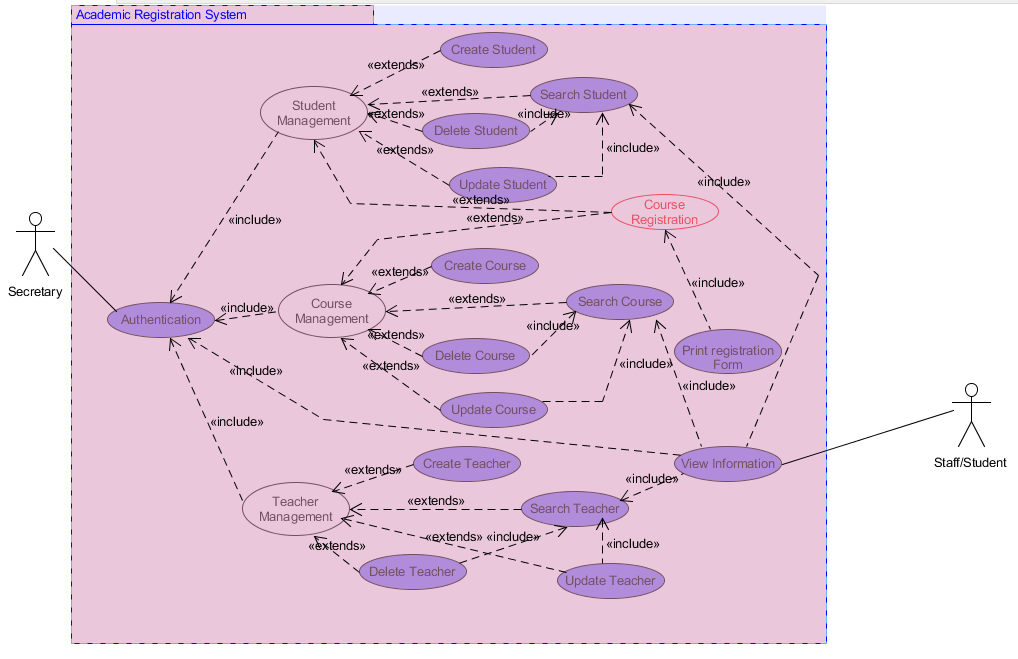
The system administrator will be in charge of performing technical configurations of the application, as setting the parameters of new data sources, configuring the backup module, and managing the roles.

### I.4.2. List of use Cases

* Create a new student;
* Delete an existing student (this functionality will be implemented after some discussions with the Rector);
* Search searching a student to display some more detailed information;
* Update personal profile of a student;
* View the details of a student. Actually this should be a sub module leading to a space to manage the student life;
* Course Registration.

The create, delete, search, and update use cases are general for students, teachers and courses. The user will also have the possibility to access a web interface for display of information and also access printable files like course registration form for reading and printing.

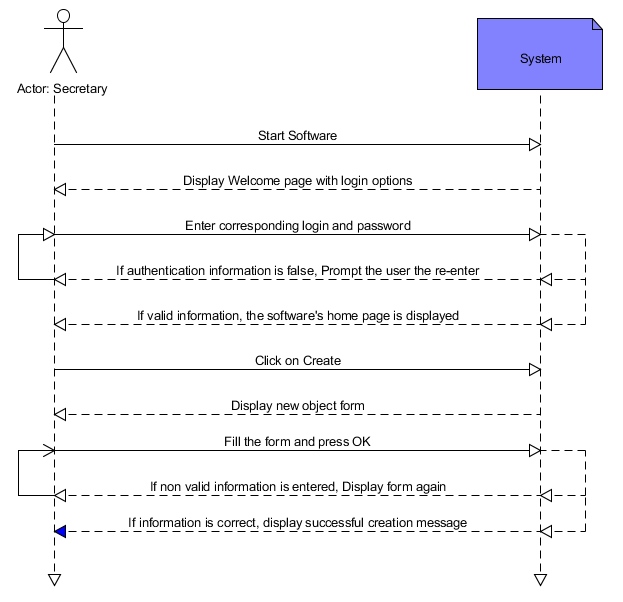
### I.4.3. Use Case Diagram



## I.5. Sequence System Diagram

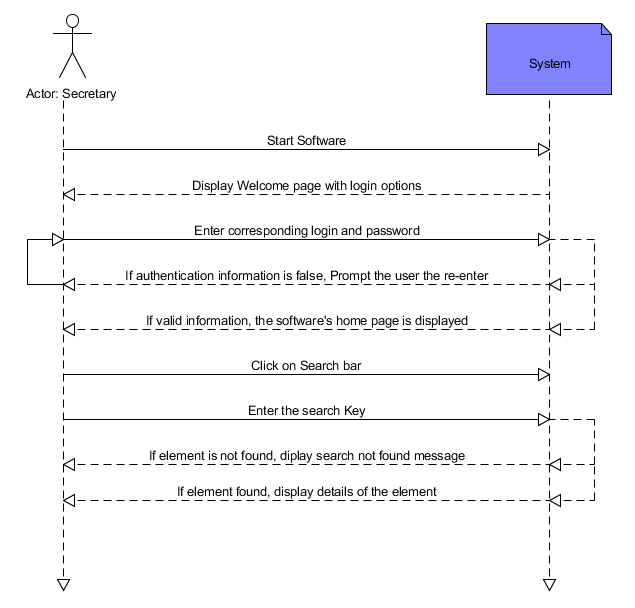
To further illustrate how the user(s) interact with the system, the sequence diagram for the use cases is shown below. For the general use cases (create, delete, …), the system will react in the same way so the diagram will not be shown for all 3 majors (Students, Teachers, Courses) because only the parameters change.

### I.5.1. Use Case: Create a new student



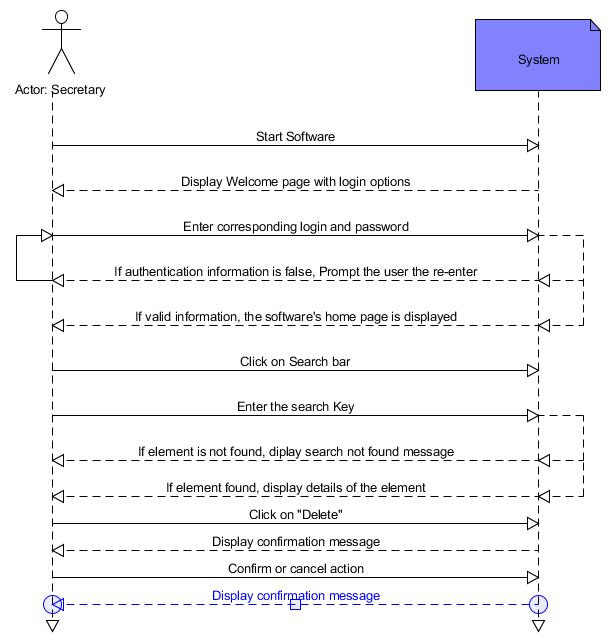
For the admission officer to create an object (student, teacher, or course), he will follow the following schematic. First he will launch the software. The login page of the software will be displayed along with a welcome message. The user then enters his/her corresponding login information. If the information entered is invalid, another prompt will be thrown to the user. If the user entered correct information, the home page of the software is displayed. The user then selects the object to be created and the system displays the new object creation form which the user fills. If all information is correct, a successful creation message is displayed. If not the form is shown back to the user with the erroneous fields highlighted. When the user is finished he exits the software.

### I.5.2. Use Case: Search a student



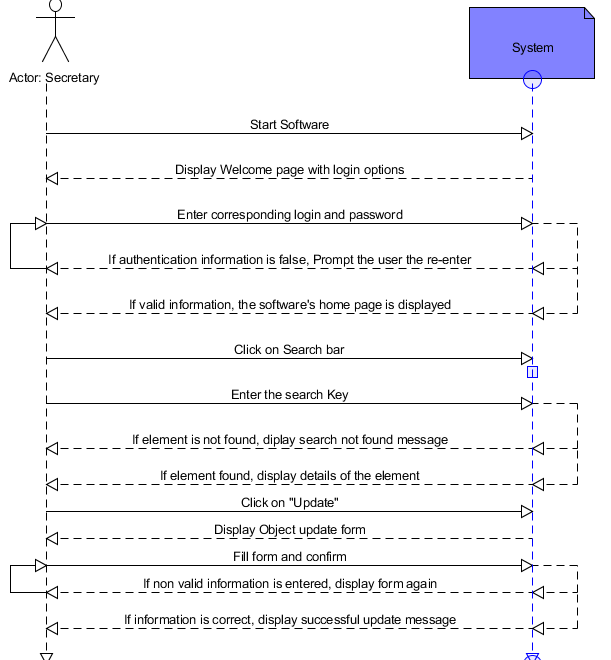
For the administrator to search an object (student, teacher, or course), he will follow the following schematic. First he will launch the software. The login page of the software will be displayed along with a welcome message. The user then enters his/her corresponding login information. If the information entered is invalid, another prompt will be thrown to the user. If the user entered correct information, the home page of the software is displayed. The user then clicks on the search bar present at every level of the software and enters the key. The software will return all elements having that key as property. The user then has the possibility to click on an element of the returned list for more details. If no element is returned, a message is displayed. The user may now exit the program.

### I.5.3. Use Case: Delete an existing student



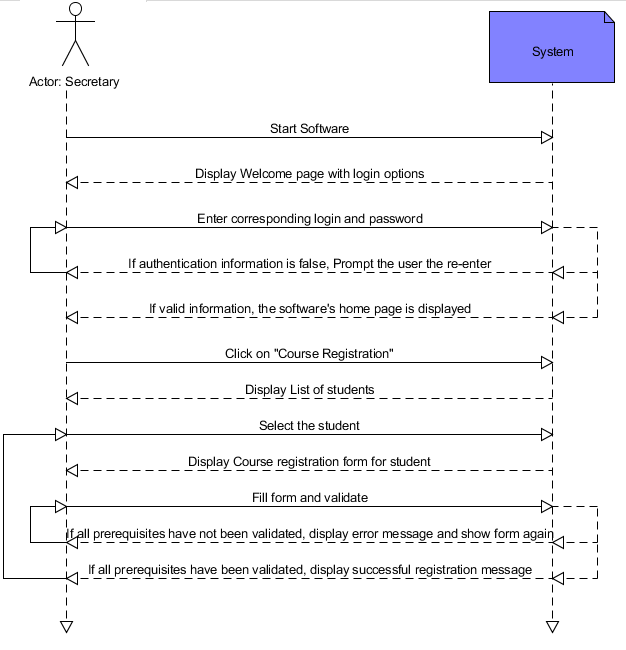
For the administrator to delete an object (student, teacher, or course), he will follow the following schematic. First he will launch the software. The login page of the software will be displayed along with a welcome message. The user then enters his/her corresponding login information. If the information entered is invalid, another prompt will be thrown to the user. If the user entered correct information, the home page of the software is displayed. The user then clicks on the search bar present at every level of the software and enters the key. The software will return all elements having that key as property. The user then has the possibility to click on an element of the returned list for more details and click on delete. A confirmation message with the possibility to cancel will be shown to the user. The user confirms his action and may now exit the program.

### I.5.4. Use Case: Update details information of an existing student



For the administrator to update an object (student, teacher, or course), he will follow the following schematic. First he will launch the software. The login page of the software will be displayed along with a welcome message. The user then enters his/her corresponding login information. If the information entered is invalid, another prompt will be thrown to the user. If the user entered correct information, the home page of the software is displayed. The user then clicks on the search bar present at every level of the software and enters the key. The software will return all elements having that key as property. The user then has the possibility to click on an element of the returned list for more details and click on update. A form with all the editable information about the object will be displayed and the user will have the possibility to make changes. If the updated information is correct a confirmation message is shown, else the field is displayed again with an error message.

### I.5.6. Use Case: Course Registration



For the administrator to register courses to students, he will follow the following schematic. First he will launch the software. The login page of the software will be displayed along with a welcome message. The user then enters his/her corresponding login information. If the information entered is invalid, another prompt will be thrown to the user. If the user entered correct information, the home page of the software is displayed. He then selects “Course Registration” and the system displays the list of undergraduates. The user may then proceed by selecting the student(s) and the course registration form is shown. Upon filling it the system verifies the grade obtained by the student in the prerequisites, if any, of the registered courses. If all prerequisites have been validated, a successful message is displayed. Else the form is shown again with the erroneous courses highlighted. The user may now proceed by closing the program.

# PART II : GRADE MANAGEMENT

## II.1. Introduction

### II.1.1 – Purpose

The software we are designing is a secretary manager that will assist the secretary officers in the accomplishment their day-to-day tasks. This is done through the automation of a large part of their activities as the statistics, the document layouts and the other repetitive tasks. The main focus of this design document is the grade management sub-project. This one consists in the optimal storage of grades and the delivery of the required documents having concern with these grades. The software must then help the secretary officer to collect the different grades and information around them, compile them and make the necessary calculations around them, and finally enables the delivery of the different documents that will use or show the grades of the different students or group of students.

### II.1.2. – Scope

The objectives of the software in its creation are to give the opportunity to the user to encrypt and decrypt the exam papers of the different students, collect the grades in a minimal number of steps, store the grades in a way that makes easy their retrieval, fully automatize all the calculations concerning the grades such as the semester, or the final results, or the GPA, enable the user to print in an appropriate layout the different printable documents that make use of the grades of the students. Another objective that is targeted is providing access to all the different grades via the internet, and allowing the other members of the administration of the institute to read and print the documents about grades.

## II.2 – Use cases

A use case describes a functional requirement of the product software. A functional requirement is an operation that the software has to provide. Therefore providing and explaining all the use cases is the same thing as exhaustively answering to the question: what the software does? From our point of view, the best way to present them is to group them by actors when there are many actors involved.

### II.2.1. – Actors

#### II.2.1.1 – System under Design

The system under design is the secretary management software. This system takes in account the computer of the secretary officer and the server on which are stored the different information generated via the system. The main actions that this system will execute are the encryption, the decryption, the storage, the calculations and the document delivery.

#### II.2.1.2 – Admission Officer

The admission officer is the secretary officer. This user has the right on all the elements generated by the system as the collection of grades and their modification, the encryption and decryption protocols, the creation of the sub-users accounts (meaning the other members of the administration that will be able via the web interface to download and print the documents referencing the grades of the students such as the semester report, the academic record and the final semester grades). The admission officer user has the rights on all the information that are kept in the server and can access and modify them at any moment.

#### II.2.1.3 – Public user

What we mean by “Public User” include everybody who is not the administrative user, but who can need to access and information about the student grades. We can group them in two categories:

* The staff user (a teacher, a head of department, dean, academic affair officer) that is created by the administrative user and that can access the documents related to the grades such as the semester reports, the final semester grades and the academic record. These documents cannot be accessed nor printed by the other class of public users. The staff users also have the possibility to see all the grades of the students for the different exams.
* The second category of public users includes any other person not member of the staff. This can be a student or any other person from inside the institute of outside. They have the reading access to all the grades collected on the server via the web interface.

### II.2.2 – List of use cases

#### II.2.2.1 – Admission Officer use cases list

The list of use cases that will be of the concern of the administrator are:

* Grades retrieval and modification
* Semester report setup
* Academic record setup
* Final grade billboard setup

An important detail to notify is the fact that the teacher and student registration are done in another part of the project but their registration will give them the opportunity to access to their personal data on the web interface of the application.

#### II.2.2.2 - Teacher and students use cases list

The list of use cases that will be of the concern of the students and teacher are:

* Web interface for grade display. (This web interface can be considered as a whole other system and details will be given further.)

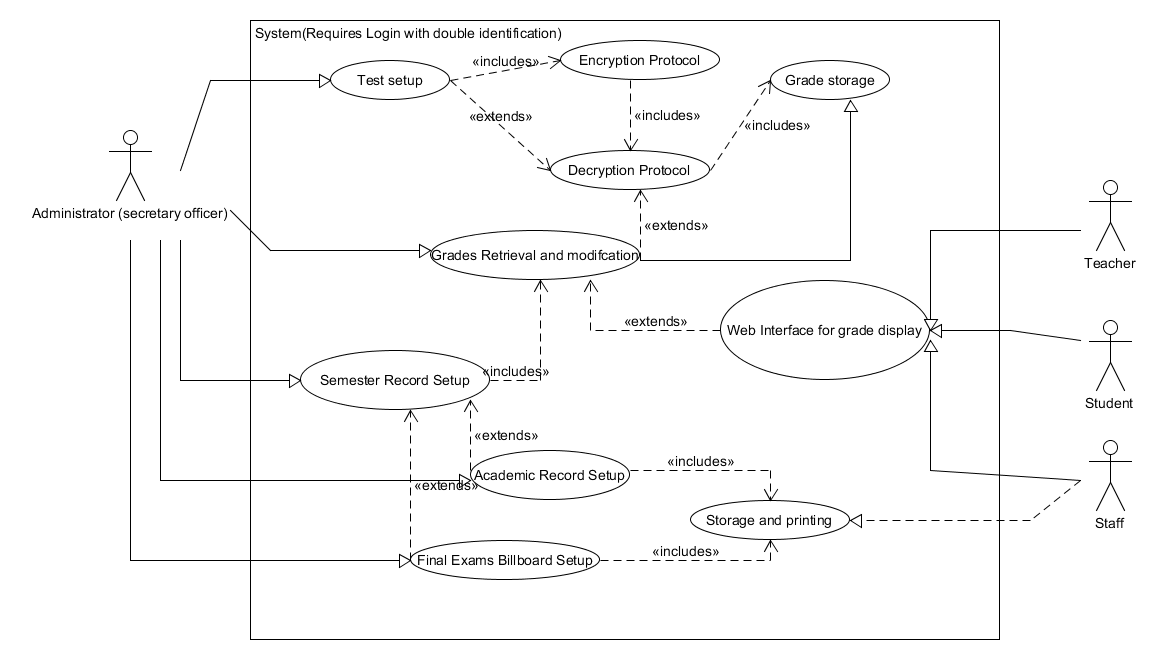
#### II.2.2.3 - Staff use cases list

The list if use cases that will be of the concern of the staff members are:

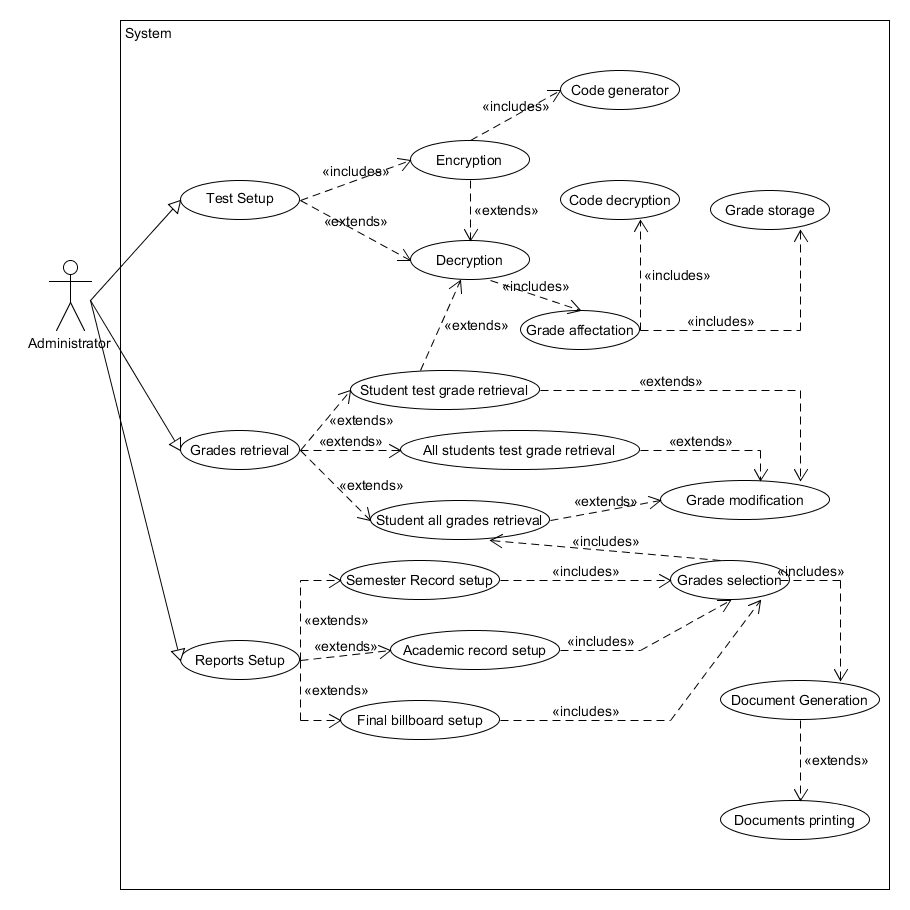
* Web interface for grade display.
* Access to the printable files for reading and printing (storage and printing).

## II.2.3 - Use cases diagrams

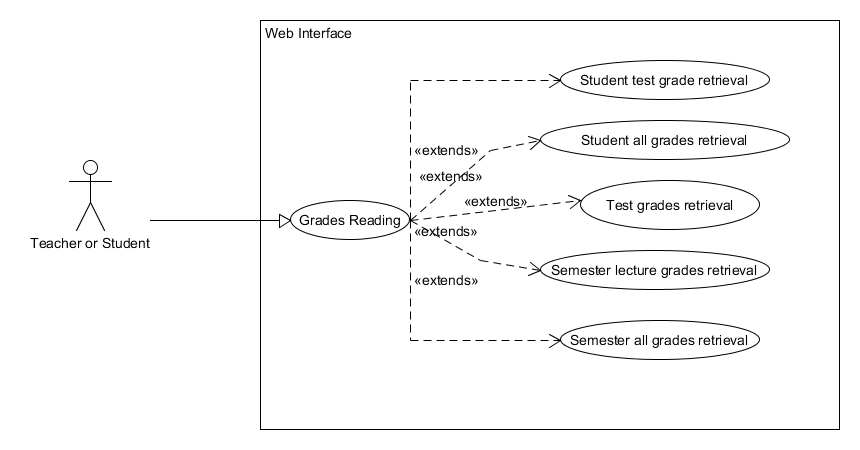
### II.2.3.1 - General Use case diagram



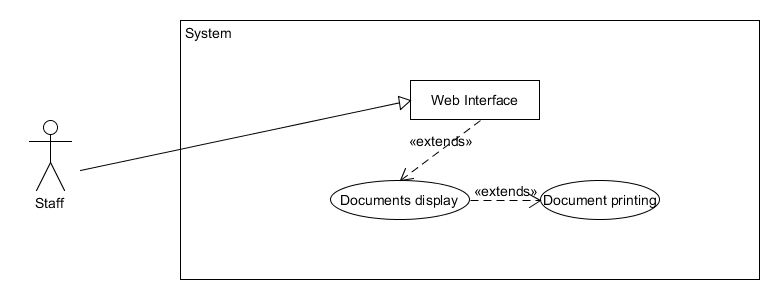
### II.2.3.2 - Administrator detailed use case diagram



### II.2.3.3 - Teacher and Students detailed use case diagram



### II.2.3.4 - Staff detailed use cases diagram



# PART III : ATTENDANCE MANAGEMENT

## II.1. INTRODUCTION

Attendance management is one of the most important and tedious processes in every academic institute especially at the university level where managing the availability of a lecturer, especially a part time offering a particular course, accessing the attendances of students to scheduled courses, and providing related statistics is not an easy task. That said, attendance management in PKFOKAM institute is one of the primary tasks to be achieved in time and therefore needs automation of the procedures and solutions which used to be manual.

## III.2. PURPOSE

The purpose of this study is to automate the traditional way of taking attendance and to generate reports automatically at the end of each session or in-between sessions.

## III.3. SCOPE

The scope of this project is to develop as web based application which will work for the whole institute. Thus, the project can be modified to operate online.

## III.4. PROCEDURES AND PROCESSES

The procedures here consist of three main parts:

### III.4.1. Gathering data from the available time table

Here, the secretary makes sure the attendance list of each course given by a particular lecturer is consistent with what is in the time table with no clashes such as the same lecturer giving two courses at the same time, or two lectures taking place in the same room. Thus the key aspect here is to be consistent with time and room assignment with respect to availability.

### III.4.2. Making sure students attend courses with all the prerequisites

Because each attendance list also contains the list of students offering a particular course given by a particular lecturer, the students present in the list must be confirmed to have validated all the prerequisite needed to attend that course. This part will be completed with help of the **academic registration project.**

### III.4.3. Providing a room for a lecture at the given time

This process involves allocating a particular room for a given lecture at a given time without any clash. So there should be no point in time when a lecture has no room to teach his students.

## III.5. INPLEMENTATION OF THE PROCESSES IN THE CURRENT SYSTEM

In the current system almost all work is done on manually. The whole session attendance is stored in log books and on paper, and at the end of the session the reports are generated. There is no interest in generating reports in the middle of the session or as per the requirement because it takes more time in calculation. As such, it can happen that at the end of a session (semester) the students who don’t have 75% attendance for example gets a notice without any warning before.

## III.6. CURRENT SYSTEM PROBLEMS

* **Not User Friendly**: The current system is not user friendly because the retrieval of data is very slow and data is not maintained efficiently.
* **Report generation constraints**: We require more calculations to generate the report so it is generated at the end of the session

And the student has no chance to improve their attendance.

* **Inaccuracy**: All calculations to generate report is done manually so there is greater chance of errors.
* **Insecurity:** The existing system requires lot of paper work. Loss of even a single register/record leads to difficult situations because all the papers are needed to generate the reports.
* **Time consuming**: Most work is done manually so reports cannot

Be generated immediately at the middle of the session as required.

As a result of the above problems encountered, the following are the objectives to attain for an automated system:

* Determining Which students are eligible to take a course
* Automatic generation of the attendance list of each course per week with the number of class hours done by each lecturer.
* Determine in which room a course is to take place
* Manage flexibility with respect to lecturer’s availability.

## III.7. FEATURES OF THE SYSTEM TO BE DEVELOPED

* **User Friendly**: The proposed system will be user friendly because the retrieval and storing of data will be fast and efficiently maintained. Moreover, the graphical user interface will be provided, which will facilitate user processes.
* **Easily generated reports**: reports will be easily generated in the proposed system so the user can have them immediately at any required time. The user will then be able to give a warning to any student to be aware of his/her attendance status.
* **Minimum manual processes**: The proposed system will minimize manual processes. Data will be fed as most as possible into the computer immediately and reports could be generated through the computers. Moreover, work will be more secured and less demanding because there will be no need to keep data on papers.

## III.8. FEASIBILITY STUDY

The main purpose of feasibility study is to consider each and every possible factor associated with the project and determine whether the investment of time and other resources yield desired results. It also includes determining the investments, manpower and costs incurred on this project. The following depicts how feasible the project is:

**Economically Feasible**

The system being developed is economic with respect to institute’s point of view. It is cost effective in the sense that has eliminated the paper work completely.

The system is also time effective because the calculations are automated which are made at the end of the month or as per the user requirement.

The result obtained contains minimal errors and are highly accurate as the data is required due to automation.

**Technical feasibility**

The technical requirement for the system is economic and it does not use any other additional Hardware and software.

**Behavioral Feasibility**

The system working will be quite easy to use and learn due to its simple but user friendly interface. Users will require no special training for operating the system.

## III.9. SYSTEM ANALYSIS

From the feasibility study previously done, a proposed solution for the automated system will be built on a web-based platform. Meaning that; the system should be able to cater access of multi-users from anywhere within the institution’s intranet. Thus, provides rapid access and facilitates quick responses to any inquiries and reports of the students’ attendances.

Here, the analysis done will satisfy the functional requirements of the proposed online system.

### III.9.1. USE CASE DESCRIPTION

#### III.9.1.1. Login

This use case will enable all distinguished users to access the platform with respective credentials which will give them access to just what is at their disposal.

#### III.9.1.2. Logout

This enables users to completely end his/her session thereby preventing insecurity. All users will as well access this functionality.

#### III.9.1.3. Generate attendance list

This use case will be accessed just by the person responsible of attendance management (the secretary) through which he/she will be able to access other functionalities to generate a list of attendance for a particular course by a lecturer. The list of students attending the course will be modified as well as the classrooms with the inclusion of the following use cases which will be accessed just by the secretary.

#### III.9.1.4. Add a classroom

This use case will enable classrooms to be assigned to a particular course at a particular time to avoid clashes;

#### III.9.1.5. Search a classroom

This use case will be used to enable any search brought to the add classroom use case.

#### III.9.1.6. Mark attendance

This use case will be available just by the lecturer to confirm the presence or absence of any student for a particular course. This is very import and will be secured given that all statistics will be generated from this data.

#### III.9.1.7. Post attendance

Here, all the information on the attendance of any student will be posted by just the faculty head at the required time. This will enable the parents and students to view their attendance information.

#### III.9.1.8. View Subject Wise Attendance

This will enable the students and parents to track the student’s attendance on a particular subject.

#### III.9.1.9. View Cumulative attendance

This use case provides a general progressive attendance information to the student and parent as well.

#### III.9.1.10. Generate Class Wise Attendance Report

This use case will be used by the accessed just by the faculty staff to provide a complete report concerning the attendance of a particular student. This functionality will therefore require data from the mark attendance use case.

### III.9.2. SYSTEM ACTORS

#### III.9.2.1. Lecturer

Lecturers will have the authority to add the attendance via the Web. Each lecturer who will be assigned classes will record the attendances on a lecture basis. The records will be stored in the database. All the records can then be queried as reports or updated by the lecturer for that particular class using full security measures.

#### III.9.2.2. Student

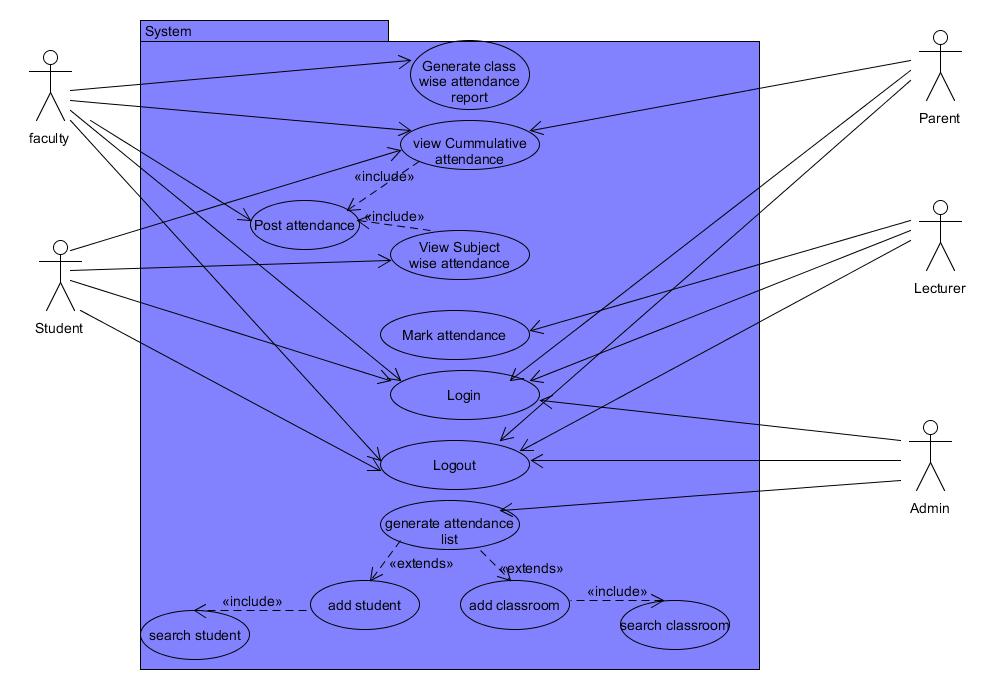
Students can only view their attendance records. If they happen to be absent but later on they provide some proofs for their absences, requests can be made to the lecturers to update the records in the system. However, if they fail to present the proofs, they will be receiving warning letters from the Department of Academic Affairs.

#### III.9.2.3. Administrator (secretary)

The administrator, in this case will be the Department of Academic Affairs/Secretary of PKF institute. The administrator will be granted access to the system as far as generating the attendance list and generating the attendance report (to some extend) is concerned. She/he can also view the attendance records on daily or monthly basis and print the reports in real-time processing.

#### III.9.2.4. Parents

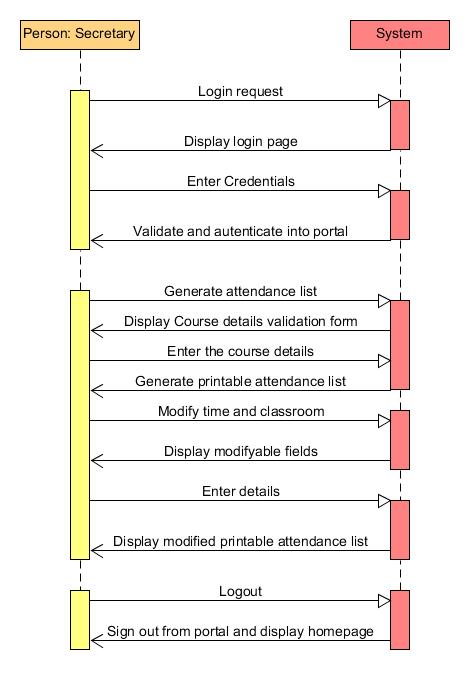
The will be granted access to the system as far as viewing subject wise attendance and viewing cumulative attendance is concerned.

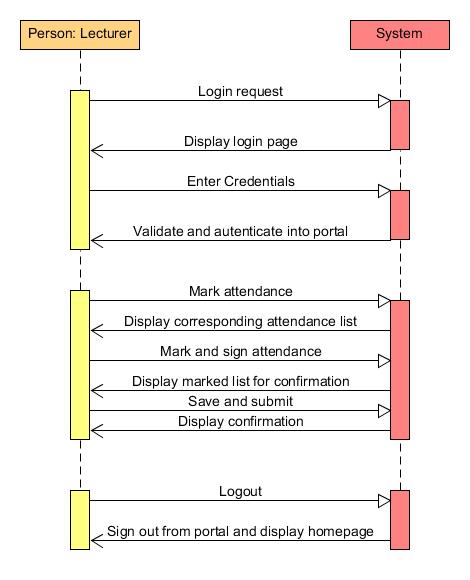


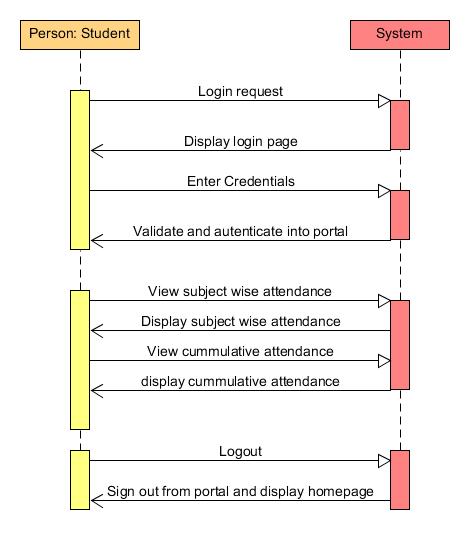
**Figure 1:** The use case diagram for the Attendance Registration System (ARS)

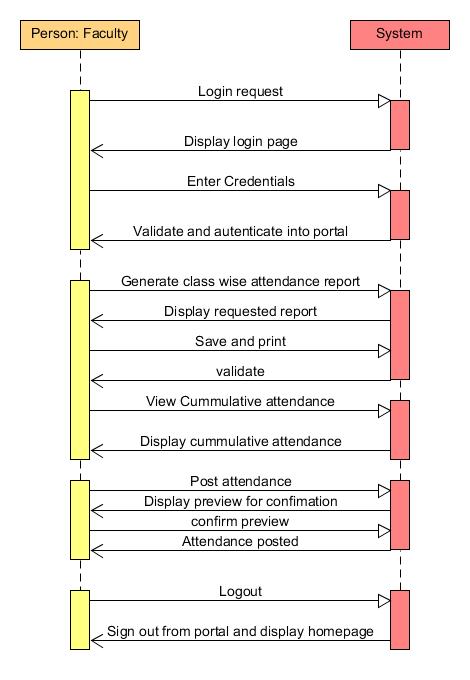
### III.9.3. SEQUENCE DIAGRAM

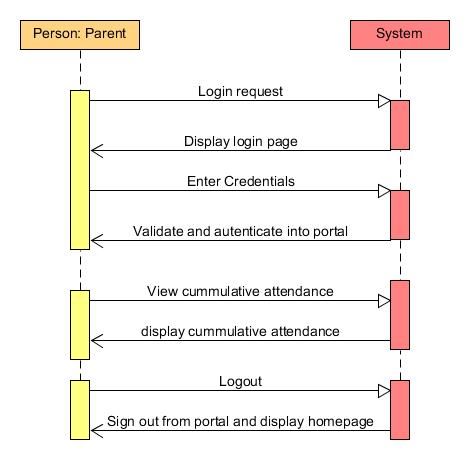
Below are sequence diagrams of corresponding actors as described in the use case detailing individual interactions with the system:











# PART IV: COURSE MANAGEMENT

## IV.1. Introduction

The Course Management System is a part of the School Management Software which will allow, the admission officer to manage courses and open them each semester they are offered. This module will be communicating with the student registration module.

## IV.2. System Analysis

### IV.2.1. Actors

There is only one actor dealing with this part of the software: the admission officer.

### IV.2.2. Functionalities

This Module will be used to manage the courses and open them any time it is needed. In more details, through this module, the admission officer should be able to:

* Add a new course to the system;
* Edit an existing course in the system;
* Open a course for a given semester;
* Get the list of students registered to a course a given semester.

#### IV.2.2.1. Add a new course

When a new course is to be created, the admission officer will have to create it providing the following information:

* The course code;
* The course title;
* The course description;
* The office hours;
* And the pre and requisites of the course.

#### IV.2.2.2. Edit an existing course

This functionality will enable the admission officer to edit a course, modifying any of the aforementioned information.

#### IV.2.2.3. Open a course for a given semester

When a course to be opened for a given semester, the admission officer should be able to do it. He or she should specify the following information:

* The semester the course is to be opened;
* The lecturers and the assistant to lecture the course.

# PART V. PROGRAMS MANAGEMENT

## V.1. Introduction

The Programs Management System is a part of the School Management Software which will allow, the admission officer to manage the programs and courses opened for each program each semester. This module will be communicating with the student registration module and the course management module.

## V.2. System Analysis

### V.2.1. Actors

There is only one actor dealing with this part of the software: the admission officer.

### V.2.2. Functionalities

This Module will be used to manage the programs and courses opened for each program each semester. In more details, through this module, the admission officer should be able to:

* Add a new program to the system;
* Edit an existing program in the system;
* Manage courses opened for each program for a given semester;
* Get the list of students registered to a program a given semester;
* Get the list of courses opened for a program a given semester.

#### V.2.2.1. Add a new program to the system

When a new program is to be created, the admission officer will have to create it providing the following information:

* The program title;
* The department in which the program is created;
* The faculty where the program is created.

#### V.2.2.2. Edit an existing program

This functionality will enable the admission officer to edit a program, modifying any of the previously enumerated information.

#### V.2.2.3. Manage courses opened for each program for a given semester

Courses are registered to each program available a given semester. The same course can be added to different programs at different level, and have different credits. A course is always added to a specified level. Moreover, the same course can exist in the system on different names. It is highly not recommended to rename and existing course. Instead, the end user should create a different course with a different title.

Through this sub module, the end user should be able to:

* Add a new course to a given program a given semester;
* Edit an existing course for a given program a given semester;
* Remove a course from a program a given semester.

##### Adding a course to a program a semester

After selecting the program for the specified semester, the end user should provide the following information:

* Specify the course to be added;
* Specify the level (Freshman, Sophomore or Junior) where the course is to be added;
* Specify the number of credits of the course;
* Specify the passing grade of the course;
* State whether the course is a core course or not.

##### Editing a course from a program a given semester

After selecting the program for the specified semester, the end user can modify provide the previously mentioned information (except the course itself of course).

##### Remove a course from a program a given semester

The end user should be able to remove a course from a program a given semester at any time.

# PART VI: STAFF MANAGEMENT MODULE

## VI.1. Introduction

This module will be used to manage the staff of PKFokam Institute of excellence. The staff includes the administration members (the President, the Rector, the Deans, the Head of departments, the Academic Affairs Officer, the Administrative and Financial Offier), lecturers (part time and permanents), and other supports like cleaners, etc.

## VI.2. System Analysis

### VI.2.1. Actors

Actors will be any person to whom the permission has been granted by the Rector will grant habilitations to carry out these operations.

### VI.2.2. Functionalities

This Module will be used to manage the staff. Since here we have different type of staff (administration, lecturers and helpers), a specific submodule should be implemented for each. We can the group the functionalities into the three following submodules:

* Administration staff management;
* Lecturers management;
* Helpers management.

#### VI.2.2.1. Administration Staff Management

This sub module will allow end users to manage the school staff. The school staff consists of the management board. Through this sub module, the end users should be able to:

* Create a new administrative staff;
* Update an existing administrative staff;
* Search and retrieve details on a potential administrative staff.

#### VI.2.2.2. Lecturers Management

This sub module will allow end users to manage the lecturers of the school as well as some of their activities. Through this sub module, the end users should be able to:

* Create a new lecturer;
* Update an existing lecturer;
* Search and retrieve details on a potential lecturer;
* Get the list of courses lecture by a lecturer a given semester.

#### VI.2.2.3. Helpers Management

This sub module will allow end users to manage the helpers of the school. This include the librarian, cleaners, guards, etc. This module will deal with the creation and the edition of helpers.

# PART VII: ADMINISTRATION MODULE

## VII. 1. Introduction

This module is dedicated to configuration of the software, as well as the management of different user profiles and their habilitations.

## VII.2. System Analysis

The configuration concerns the configuration of departments and semesters:

* a new semester should be explicitly created at the start of every semester;
* If there is a new department, it has to be created and configured in the system;
* The maximum number of credits a student can take per semester should be configured;
* Closing a semester to delete information which will not be useful the following semester.

In addition to the configuration side, this module should enable the management of different user profiles.