International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Design and Creation of a Web Application for the Management of Students in a Private School.

[1]*Mualundama Mualundama Benjamin*, [2]*Basa Kutupu Faustin*

[1](Assistant to ISP/Luiza), [2](Doctoral student at the University of Kananga)

Higher Pedagogical Institute of Luiza

DOI: https://doi.org/10.55248/gengpi.4.1223.123426

SUMMARY

In the contemporary world, information has become the main engine of economic and social progress. The important role it plays in an organization requires it to devote significant investments and special attention. Our work consists of designing and creating a web application for managing students of a private school for foreign languages. To do this, we followed the UP software development approach and the UML modeling language. The implementation of our application is based on a database developed with MySQL using a three-tier architecture. The development environment used is Wamp.

Keywords: UP process, 2TUP, UML, PHP, Wamp, CSS3, HTML5, 3-tier architecture.

Introduction

Currently, the world is experiencing considerable technological advances in all sectors with the help of IT, which plays an important role in the development of many companies and organizations.

Before the invention of the computer, we recorded all information manually on paper media. This caused many problems such as the loss of considerable time in searching for this information or the degradation of the latter [1].

The new logic of work organization requires education and learning establishments to essentially use information as raw material in order to be more efficient. They must therefore integrate a dismantling of the information system into their strategic investments, to the extent that they structure the capture, storage, organization and communication of information.

Today, private schools, to which we will also link our study, are an integral part of the educational establishments where computers can help. Indeed, the growth of the student population requires the establishment of rational and rapid management, and to this day, the way of manual management is still dominant, hence the need for introduce IT into these training centers.

We were able to observe, in fact, during our observation within the school of foreign languages that a majority of the processing is done manually, which gives rise to a certain number of problems such as: Large volume of information processed manually, which sometimes causes errors in the preparation of documents; Difficult search on registers which causes a waste of time; Information insecurity; Possibility of error in filling out the various documents and registers; Possibility of error in calculating statistics.

Therefore, the best solution to overcome the problems is computerization to ensure instant access to data and security of the latter, which simplifies administrative work. As a result, we proposed to school officials to design an application for the management of their school.

The objective of our project presented in this report is the design and creation of a simple web application for managing the entry/exit of students as well as teachers in a training center. And to carry out this task, our choice fell on the UP (Unified Process) software development method and UML (Unified Modeling Language) as the modeling language;

This present work will be structured into four chapters:

In the first chapter, we will present, as the first part, some notions about web technologies, . In the second chapter, we will do a detailed design of our application by presenting the use cases under different diagrams. The third and final chapter will be the part of the implementation and creation of our application, where we will present the development environment, the structure of our application and some of its interfaces.

I: WEB TECHNOLOGY

The development of web applications presents certain particularities, at the technical and ergonomic level. This specificity obliges us, at the time of design, to recommend design methods and working methods dedicated to this type of applications.

This chapter is made up of two sections, the first is dedicated to defining some generalities about web technologies; the second focused on the presentation of the host organization.

1.1 Web technologies

In the following, we will define a website, a web application, and we will briefly present the difference between the two.

1.2.1 A website

A website is made up of a set of structured documents, called web pages, stored (hosted) on a computer (server) connected to the global network. There are two types of websites:

A static site: made up of web pages with fixed content. Each page is coded in HTML (HyperText Markup Language) and displays the same information to each visitor. These pages are modified through intervention by the website administrator.

A dynamic site: the pages of the site that compose it can be modified by visitors. In addition, between two visits to the same site, the content of the page may be different without action by the website administrator. The server that runs the site uses Scripting technology (such as PHP: Hypertext Preprocessor) as well as a database such as MySQL (My Structured Query Language).

1.2.2 A web application

A web application is a set of pages that interact with users, with each other, as well as with the different resources of a web server, notably databases. The architecture of a web application contains an application server which makes it possible to process logic and application state. While the architecture of a website contains three main components:

– A web server.

– A network connection.

– Client browsers.

In other words, a web application is used to manipulate information and is subject to access and role controls, and a website is used to expose information.

1.2.3 The architecture of web applications

An architecture is a generic and conceptual model which relates to a subject and which represents the functionality, structure, positioning, interrelation of the different types of elements (hardware, software, infrastructure) which compose it.

Generally, an application is divided into 3 levels (layers) of abstraction:

The presentation layer: this is the part of the application visible to users (we will talk about user interface). In our case, this layer is a web browser, which is presented in the form of HTML pages, composed of forms and buttons.

The business layer: corresponds to the functional part of the application, the one which implements the logic, and which describes the operations that the application performs on the data, according to the requests of a user carried out through the presentation layer .

The data access layer: it consists of the part managing access to the system database.

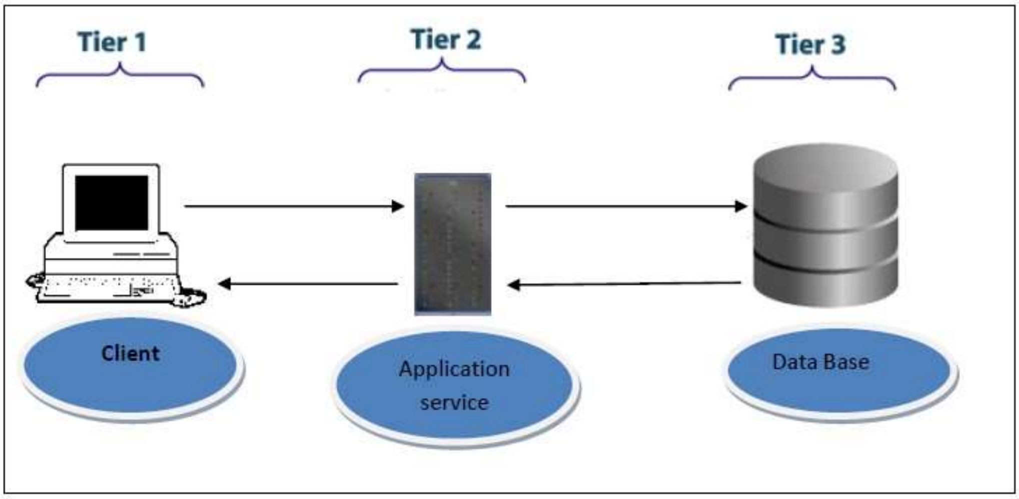
There are different architectures for a web application:

– 1-tier architecture

– 2-tier architecture

– 3-tier architecture – N-tier architecture.

In our application we used a 3-tier architecture which is illustrated in figure (1.1) below:



1.2.4 The Client/server application

A web application is said to be based on a Client - Server model if the database is located on a server which is only used for that purpose, and to interact with this database, you must use “client” software which will query the server and transmit the response that the server gave it.

The server can be installed on a different machine from the client.

Therefore, when we install a DBMS (basic data management system), we are actually installing two things:

- the server
- the customer

Each request (insert/modify/read data) is made through the client. So we may need a language to chat with the client, to give them the requests we want to make.

In the case of our application we have MySQL as the database management system, and SQL (Structured Query Language) as the database management system language.

1.2.5 The web server

A web server is software that provides web pages in response to requests from web browsers. A page request is generated when a user clicks a link on a web page or enters a URL (Uniform Resource Locator) in the browser's address field.

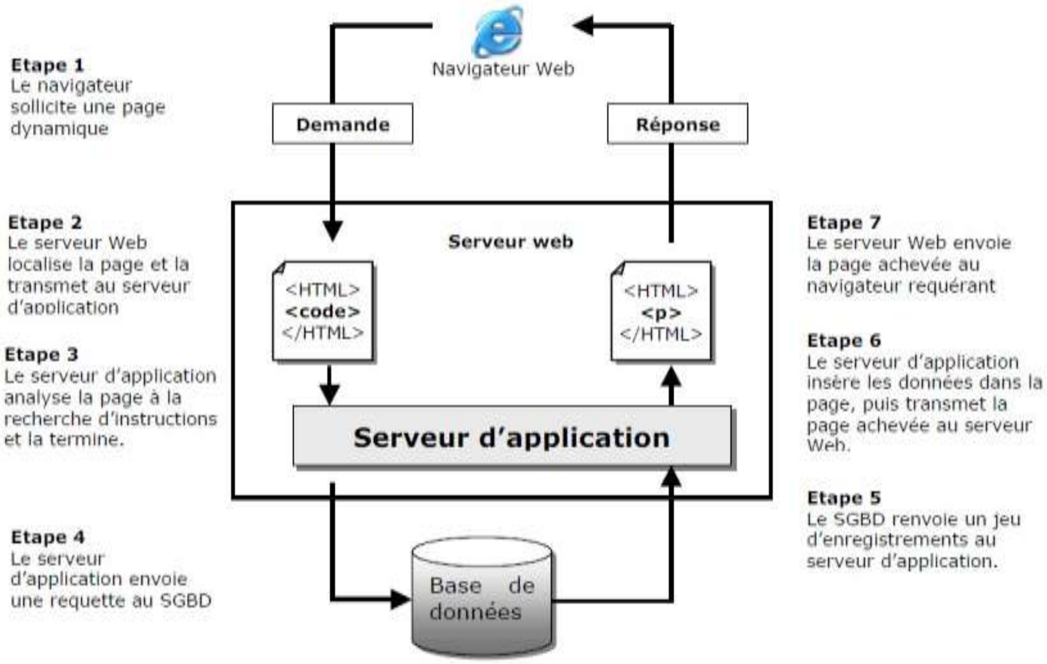
1.2.6 How a web application works

When a web server receives a request for a static web page, it simply forwards that page to the requesting browser. On the other hand, when the web server receives a dynamic page request, it forwards that page to a special software extension responsible for completing the page. This special software is called an application server. The application server reads the code on the page, completes that page based on the instructions in the code, and then removes the code.

This results in a static page that the application server sends back to the web server, which then transmits this page to the requesting browser. The browser only receives pure HTML code when the page is served to it. An application server allows us to work with server-side resources such as databases. A dynamic page can, for example, instruct the application server to extract data from the database and insert it into the HTML code of the page.

The instruction for extracting data from the database is called database query. A query is made up of search criteria written in a database language called SQL. The SQL query is written in scripts or tags on the server side of the page.

All these actions are summarized in the diagram shown in figure (1.2) below:


Navigateur Web
Etape 1
Le navigateur sollicite une page dynamique
Demande
Réponse
Etape 2
Le serveur Web localise la page et la transmet au serveur d'application
Serveur web
<HTML>
<code>
</HTML>
<HTML>
<p>
</HTML>
Etape 7
Le serveur Web envoie la page achevée au navigateur requérant
Etape 3
Le serveur d'application analyse la page à la recherche d'instructions et la termine.
Serveur d'application
Etape 6
Le serveur d'application insère les données dans la page, puis transmet la page achevée au serveur Web.
Etape 5
Le SGBD renvoie un jeu d'enregistrements au serveur d'application.
Etape 4
Le serveur d'application envoie une requette au SGBD
Base de données

Generally and after the creation and implementation of an application; A final step remains to be accomplished: hosting the application so that it becomes accessible and manipulated by users.

1.2.7 Hosting a web application

Is a service aimed at making a site or web application accessible on the Internet. In order for the pages to be visible to everyone, they must be stored on a computer permanently connected to the Internet (server). A host is like a company specializing in hosting websites on servers, its main activity on the internet consists of installing servers, securing them, keeping them up to date by installing security updates to avoid malicious attacks and repair them in the event of a breakdown.

There are several types of accommodation including:

1. Free hosting: this type of hosting offers a basic service, with fairly limited storage space, and reliability that is not always optimal.

2. Paid hosting: this type of hosting offers a quality service, with good storage capacities and great reliability. Support will also be more available and responsive in case of problems and will help with personalized responses.

In the next part we will talk about the languages used for creating applications.

1.2.8 Languages used for creating web applications

The development of a web application requires knowledge of the different languages used in web technologies: HTML for the presentation of pages, CSS (Cascading Style Sheets) for the graphic charter, JavaScript, Java or ActionScript for the automation executed by the client, as well as a language such as Java, PHP and others

1.3. The PHP language

PHP is a very popular, multi-platform, object-oriented scripting language.

It was designed to allow the creation of dynamic applications, the generation of code (HTML, XHTML (Extensible HyperText Markup Language), CSS) and data (JPEG (Joint Photographic Experts Group), PNG (Portable Network Graphics)). is most often coupled to an Apache server, although it can be installed on other Http (HyperText Transfer Protocol) servers. The PHP instructions are read and analyzed by the web server. They never reach the browser that displays the page because the web server replaces the PHP code with the content that the code was intended to generate.

Given all these elements and the constraints relating to this project, we chose to use the PHP language, which presents a good compromise between speed of implementation and performance. On the other hand, a web application with PHP is very easily deplorable and works with all operating systems, whether Mac, Linux or Windows. After having seen the different web technologies in the first part, we will move on to the second part which is the host organization.

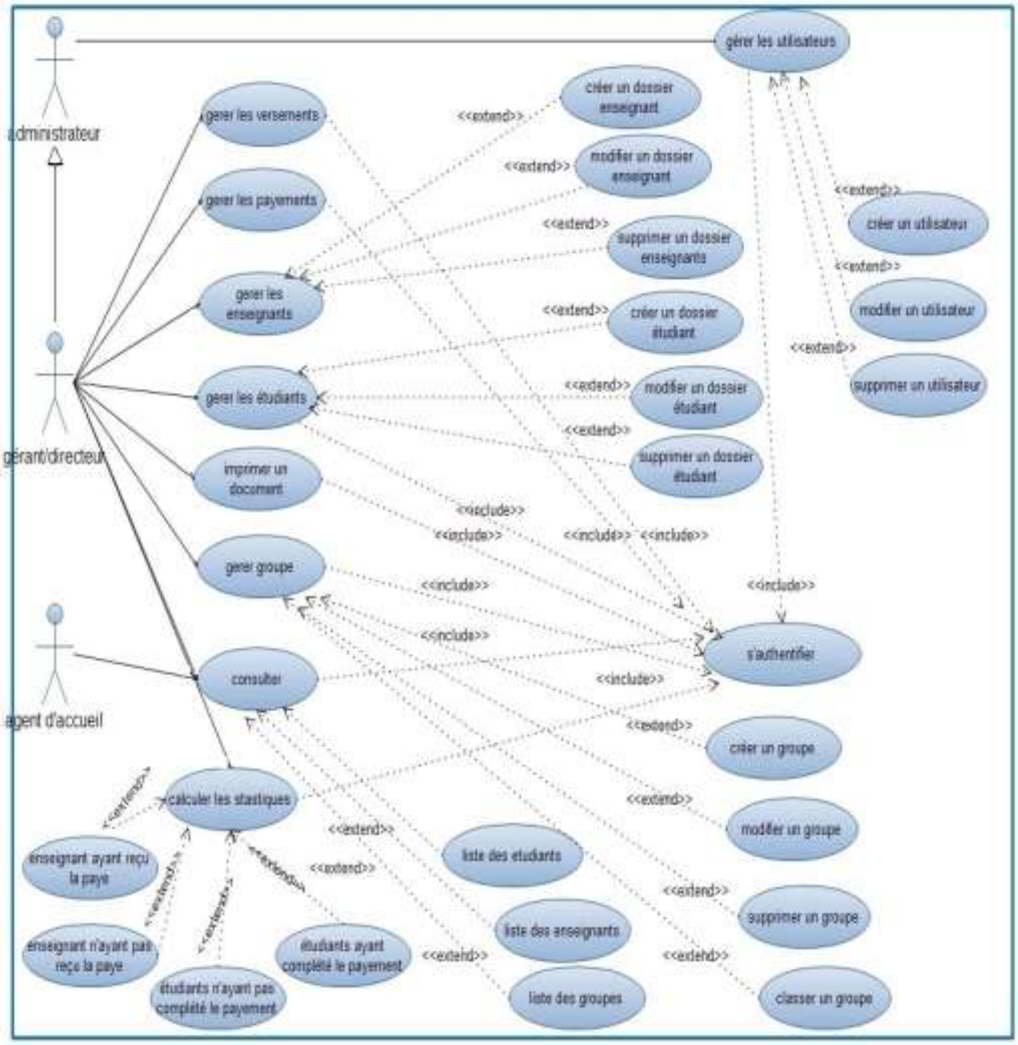
II: MODELING AND REALIZATION

2.1. Modeling

2.1.1. Identifying use cases for the application

The use case diagram identifies the functionalities provided by the system, the users who interact with the system (actors), and the interactions between them. A use case corresponds to a certain number of actions that the system must execute in response to a need from an actor.

2.1.2 The global use case diagram



Based on figure (2.2), we have distinguished the following use cases:

1. User management: for the site administrator.

2. Authentication: the application must verify that the user is who they say they are in order to authorize them access to the application.

3. Data update which includes: creation, modification and deletion of a student/teacher/group.

4. Print a document: a training certificate for the student.

5. Consult a student/teacher file: obtain information on any student (teacher) and the list of groups.

6. Calculation of statistics: number of students who have and have not paid for their training, the number of teachers who have and who have not received pay.

At the end of the expression of needs using the global use case diagram, in what follows, we detail each of the use cases presented in figures (2.2) up to figure (2.11), giving its textual description.

2.1.3. The sequence diagram

Sequence diagrams are the graphical representation of interactions between actors and the system in chronological order in the UML formulation.

These communications between classes are recognized as messages. The sequence diagram lists objects horizontally, and time vertically. It models the execution of different messages as a function of time. To create the sequence diagrams we used interaction operators. An interaction operator defines the type of a compound fragment.

The interaction operators that we previously used in sequence diagrams are:

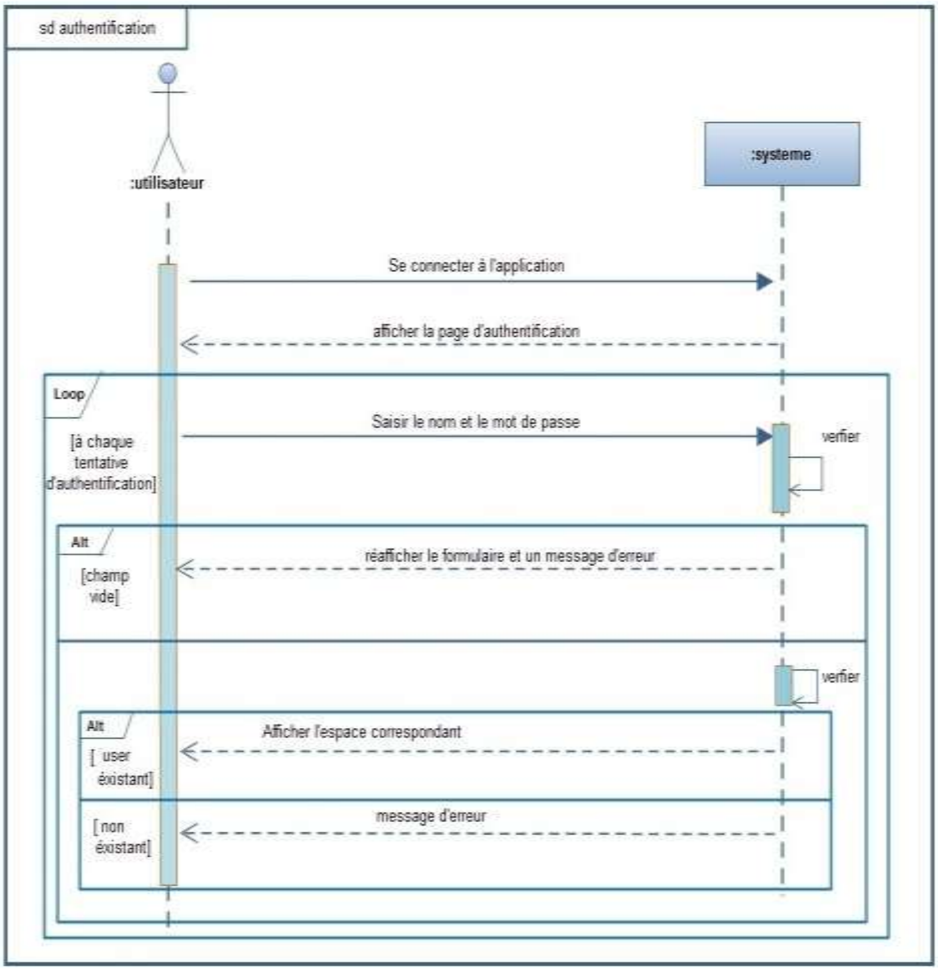
1. Reference (ref): this operator designates that the fragment refers to a case seen previously.

2. Alternative (Alt): this operator designates that the compound fragment represents a choice of behavior. At most one interaction operand will be chosen. The chosen operand must have an implicit or explicit guard expression which has the value “true” at this point of the interaction.

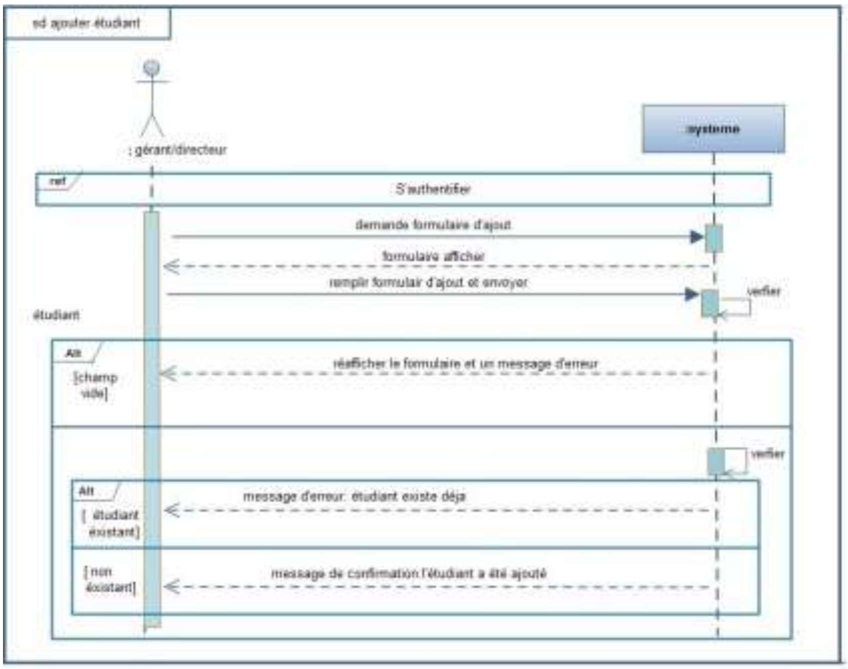
3. Loop: this operator designates that the compound fragment represents a loop. The “loop” operand will be repeated several times.

Figures (2.1) to (2.2) will respectively present the sequence diagrams: authenticate, add/modify/delete consult a student file and print a document.

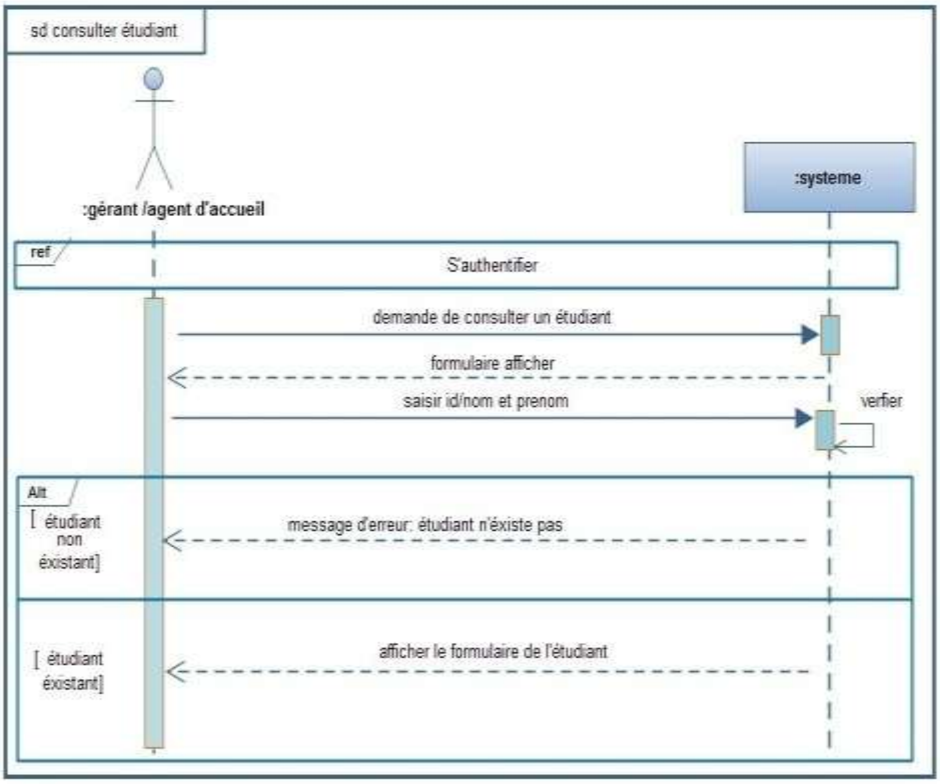
A. Authenticate Sequence Diagram


sd authentification
:utilisateur
:systeme
Se connecter à l'application
afficher la page d'authentification
Loop
[à chaque tentative d'authentification]
Saisir le nom et le mot de passe
verfier
Alt
[champ vide]
réafficher le formulaire et un message d'erreur
verfier
Alt
Afficher l'espace correspondant
[user éxistant]
message d'erreur
[non éxistant]

B. Sequence diagram add student record


sd ajouter étudiant
: gérant/directeur
:systeme
ref
S'authentifier
demande formulaire d'ajout
formulaire afficher
remplir formulair d'ajout et envoyer
verfier
étudiant
Alt
[champ vide]
réafficher le formulaire et un message d'erreur
verfier
Alt
[étudiant existant]
message d'erreur: étudiant existe déja
[non éxistant]
message de confirmation l'étudiant a été ajouté

C. Sequence Diagram Consult a Student



2.2 Realization

In the life cycle of our project, design represents a primordial and determining phase for producing a high quality application.

In what follows we present the interaction diagrams, then we will move on to the description of the participating classes in the class diagram, and we will build the relational model from the classes of the application, and this by following the rules from the transition to relationships.

2.1. Interaction diagram

The interaction diagram is a detailed sequence diagram. In this section we present the interaction diagrams, replacing the system represented by a black box in the sequence diagram with the following three types of classes:

1. The interface object (dialogue)(<D>): represents the interface between the actor and the system,

2. The control object (<Ctrl>): represents system processing triggered by an actor,

3. The entity object (<E>): represents objects written in the use case.

We will also respect the rules that we had set on the relationships between the analysis classes, but this time focusing on the dynamic interactions between objects:

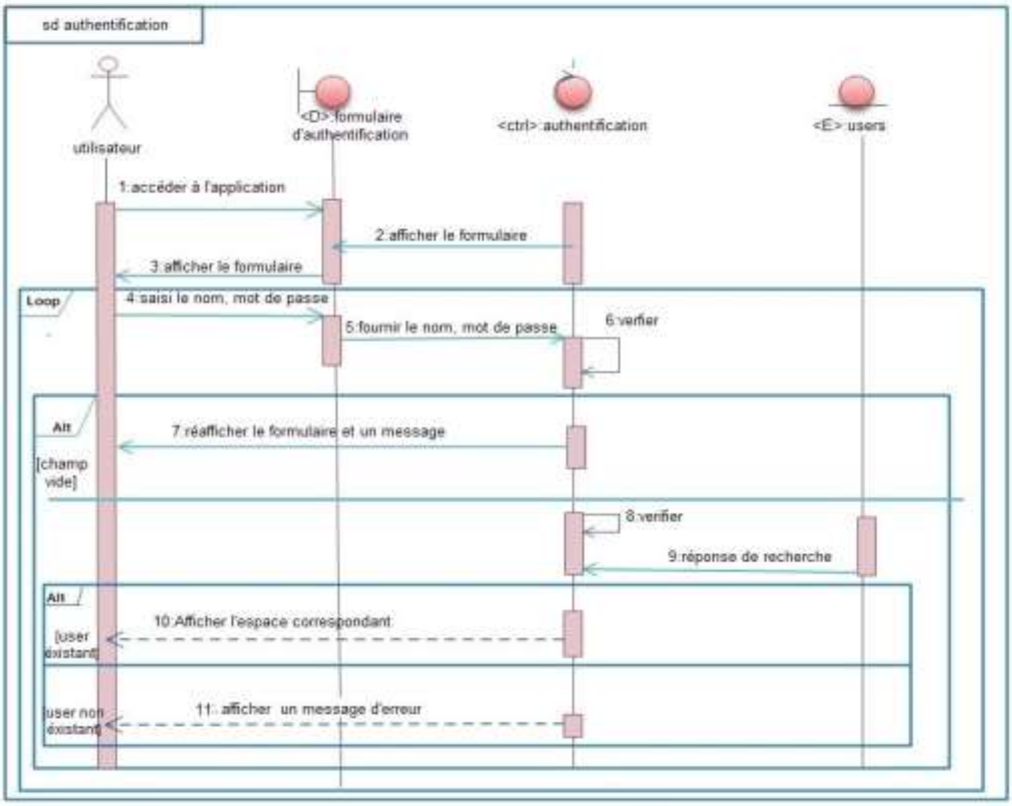
– Actors can only interact (send messages) with dialogues;

– Dialogs can interact with controls;

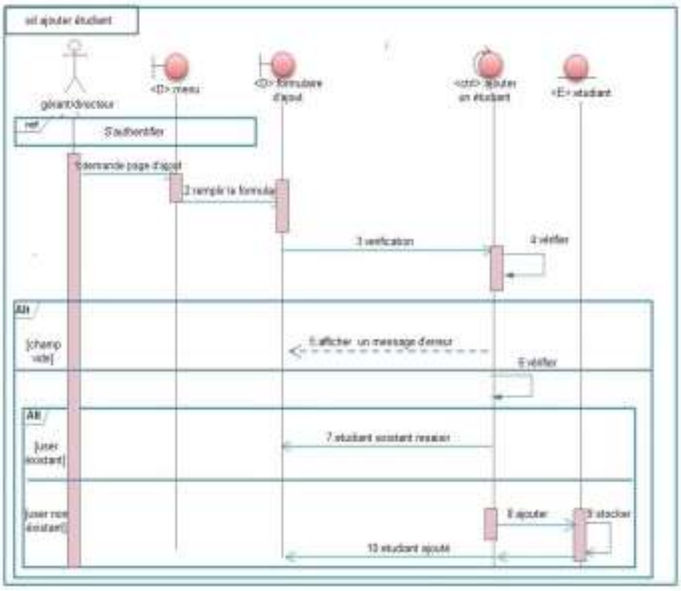
– Controls can interact with dialogs, entities, or other controls; – Entities can only interact with each other.

Figures (2.1), (2.2), (2.3), (2.4) below will respectively present the interaction diagrams for authenticating, adding, modifying and deleting a student record.

3.2.1 User authentication interaction diagram


sd authentification
utilisateur
<D>:formulaire d'authentification
<ctrl>:authentification
<E>:users
1:accéder à l'application
2:afficher le formulaire
3:afficher le formulaire
Loop
4:saisi le nom, mot de passe
5:fournir le nom, mot de passe
6:verifier
Alt
[champ vide]
7:réafficher le formulaire et un message
8:verifier
9:réponse de recherche
Alt
[user existant]
10:Afficher l'espace correspondant
[user non existant]
11: afficher un message d'erreur

2.1.2 Interaction diagram managing a student file: Case: add a student file



2.2.Description of classes

Student: this is the essential entity of the application, he first goes through the manager or reception agent and registers for training.

Teacher: he is responsible for the student groups, and receives pay from the manager.

Group: contains students who receive training, each group is supported by a teacher.

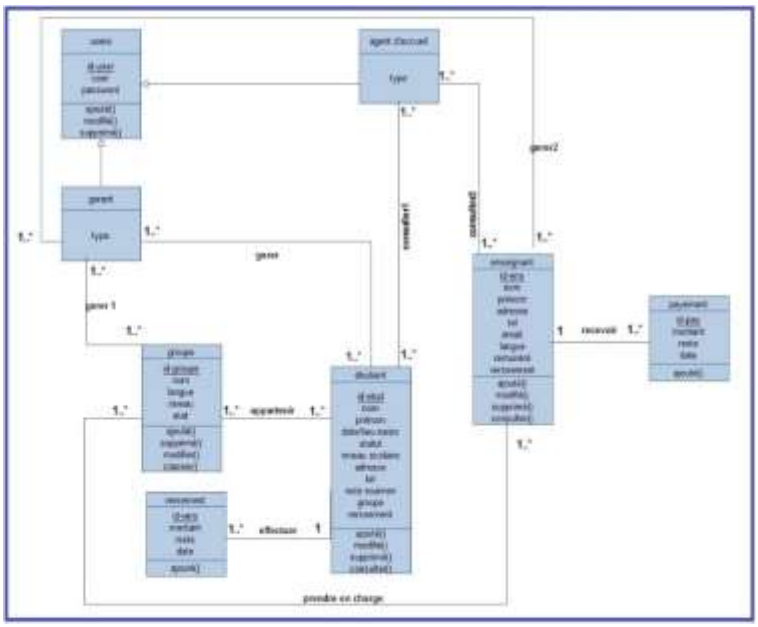
Users: – manager: he is responsible for monitoring the student/teachers/groups (registration, choosing a training course, paying training costs, etc.).

– Reception agent: this is the person who is responsible for consulting the files of

students/ teachers/groups.

Payment: students make payments for training.

Payment: Teachers receive salaries.



III: IMPLEMENTATION

To be able to properly carry out our application we have opted for some tools which we will define below.

a. PHP (Hypertext Preprocessor or Personal Home Page)

As we have already seen previously PHP is a free scripting language mainly used to produce dynamic web pages via an http server.

b. CSS (CASCADING STYLE SHEETS)

So-called CSS style sheets allow you to manage the presentation of a web page. CSS allows you to create style and layout rules.

These rules concern the positioning of elements, alignment, fonts, colors, margins and spacing, borders, background images, etc. The purpose of CSS is to separate the structure of an HTML document and its presentation.

c. XHTML (Extensible HyperText Markup Language): XHTML is a combination of HTML and XML (eXtensible Markup Language), is a markup language used to write pages for the Word Wide Web (www).

d. JAVASCRIPT: It is a programming language that offers the possibility of implementing treatments. Elaborated in web pages, and allows improvements to be made to the HTML language by allowing commands to be executed from the client side (i.e. at the browser level and not the web server). Thus the JavaScript language is strongly dependent on the browser calling the web page into which the script is incorporated, but in return it does not require a compiler.

The application was developed using a Windows web development platform which is the Wampserver package (2.2) for:

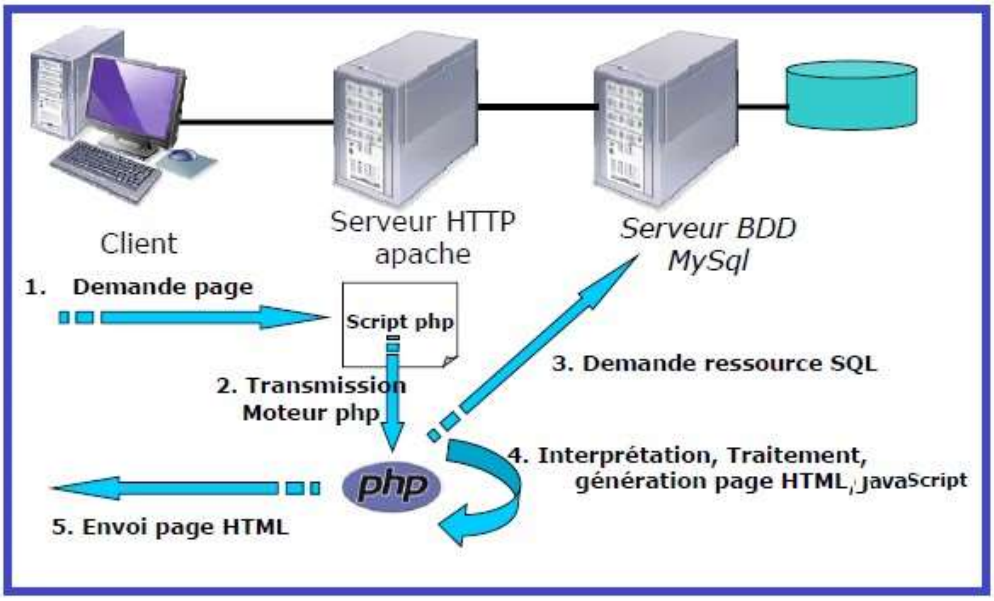
– Windows: A tool that runs under Windows XP, Windows 7.

– Apache (2.2.21): it is a web server. This is the most important of all programs, it is responsible for delivering web pages to visitors. However, Apache only manages static websites. It must therefore be supplemented with other programs.

– MYSQL (5.5.16): this is the database management system (DBMS).

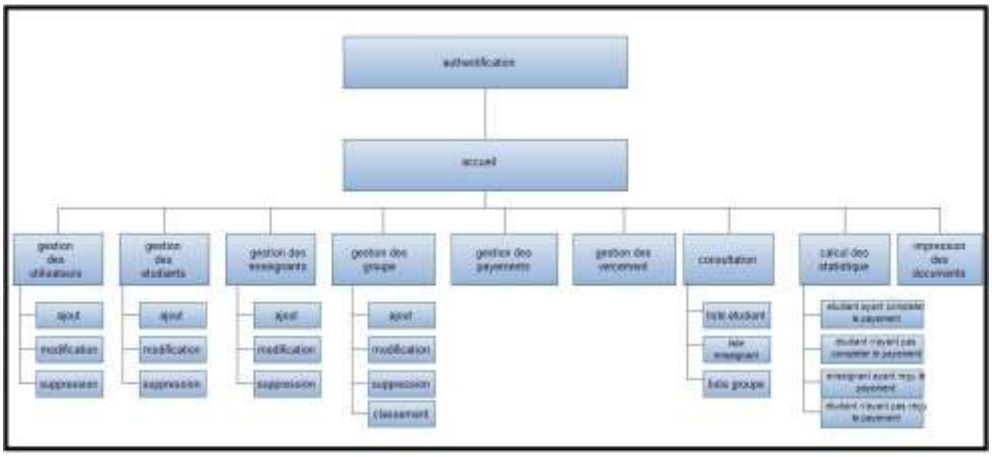
– PHP (5.3.8): this is a plug-in for Apache which makes it capable of processing dynamic web pages[11].

The diagram shown in figure (4.2) explains how these tools work in the system



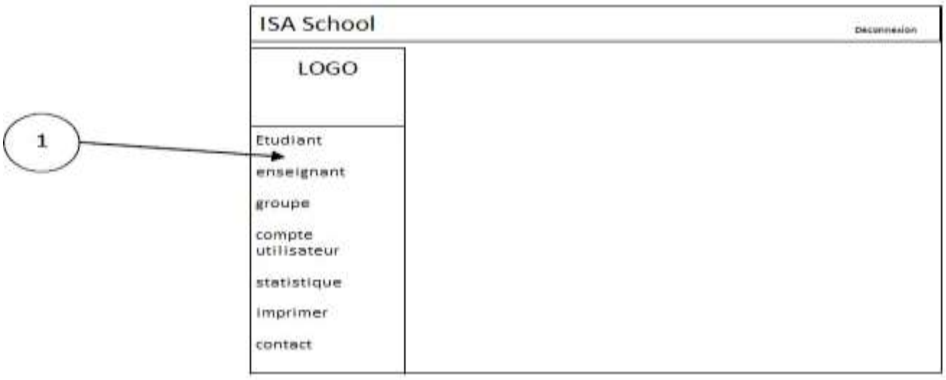
3.2. Structure and model of the application

The following diagram, presented in figure (4.2), summarizes the organization of the application pages (Navigation diagram)



Furthermore, after specifying the needs seen previously and following the structure presented in Figure (4.2), we also proceeded with a set of models developed for potential users of the application. The following figures (4.3) and (4.4) illustrate some of them:

4.3.1 Mock-up of the home page after authentication



3.3 Presentation of some application interfaces

Using the models presented previously we have reached the end of the construction of our application.

Figures (4.5) to (4.12) illustrate examples of screenshots of the different interfaces of the application.

3.3.1 Screenshot: authenticate

The user must complete both authentication fields by entering the correct contact details so that they can access the home page.

CONCLUSION AND PERSPECTIVES

Our project consists of designing and implementing a web application for managing students at the school of foreign languages. In this work, we talked about the management of the ISA School training center, and we presented the stages of development of our application. We also relied throughout the design process on the UML, which is a practical graphic tool to illustrate our approach.

The aim of the application was to show the interest in computerization of the educational and school sector, by creating a prototype for managing ISA School student files, ensuring several advantages compared to manual management, namely :

– Facilitate the daily professional exercise of the various professionals at ISA School. – Improve storage, archiving and reduce access time to files.

The application created ensures several basic functionalities, namely the creation of a file, its modification, consultation of the different files, the printing of the different documents and the calculation of statistics.

The Theme assigned to us is very instructive on an educational level and very interesting on a technological and development level. We as students at the end of the cycle It allowed us to:

– Increase our knowledge.

– Initiate to different development technologies (css, php...). – Improve our skills in object-oriented programming.

In the end, given that no one can claim to address a field as a whole, we would like to come:

– Improve interfaces so that they meet ergonomic criteria.

– Establish a database security system and limit the number of authentication attempts to the application.

– Host the application on a server.

– Develop the application to a mobile application.

Bibliography

1. http://www.memoireonline.com/ApplicationwebdeGestiondepharmacieenJava2. html Consultation date 05/22/2023
2. http://html5-tutorial.fr/developpementd'applicationweblapprochompisant\etlApprochemodulaire/ Date of consultation 7/05/2023
3. http://www.aidice--web.com/accueil/definitionsiteweb.php Date of consultation 7/05/2023
4. https://www.ideematic.com/dictionnaireweb/applicationweb Date of consultation 7/05/2023
5. http://www.Developpez.com/LelangagePHP Consultation date 06/28/2023
6. http://www.Developpez.com/UP:unifiedprocessus Consultation date 05/22/2023
7. https://msdn.microsoft.com/fr/library/aa292164($v=vs.71$).aspx Date of consultation 7/05/2023
8. http://aws.amazon.com/fr/application-hosting/ Date of consultation 06/13/2023
9. http://www.microsoft.com Date of consultation 03/04/2023
10. http://www.wampserver.com/ Date of consultation 03/04/2023
11. Samir ADOUANE. Integration of means of dynamic modification of content on the web. Master's thesis, Universit´e el Hadj Lakhdar Batna, Batna, 2007.
12. Philip BOUSQUET. Create a web application with PHP, 2008.
13. L. SHKLAR and R. ROSEN. Architecture: Principles, Protocols and Practices. Edition: John WILEY and SONS, 2003.