Ecole Supérieure Privée
d'Ingénierie et de Technologies

Plateforme des stages PFE

Fulfilled by:

Mnissi Rania

Mediouni Arij

Challakh Mohamed Amine

Ben Ali Oussama

Kadhraoui Mohamed Amine

El Hadj Sallah Abderrahmen

4InfoB1

Table of Content

List of figures:

List of tables:

General Introduction

There is no doubt that the computer revolution is the largest and most innovative that marked the life of modern humanity. Indeed, nowadays software offer solutions to all problems of life, both in professional fields as for personal applications.

In this context and as part of our Pi Dev in ESPRIT we will design and develop our first Java Enterprise Edition application named *PFE Internship*.

PFE Internship allows the processing of end-of-studies projects within an engineering school, from the beginning of the internship until the defense.

It will also display a complete dashboard that will serve as a decision support tool to allow the school to update study plans.

Our application includes six main modules:

1. M1
2. M2
3. M3
4. M4
5. M5
6. M6

Chapter 1: Analysis of existing situation

Introduction:

In the following chapter we will start by putting the project in his general frame, we will study the existent and see what the available applications are, show their disadvantages and present our project as a solution.

1- Market study:

Internships and the end of course project are key moments in the encounter between the professional world and the world of active education. Engineering courses are punctuated by three internships and an end of course project, according to a study commissioned by https://www.naceweb.org .

- 53 percent reported that they obtained their current positions directly from their internships
- Millennial Hiring Report indicates that graduates who complete three or more internships are more likely to secure full-time employment, with 81.1 percent

2- Description of existing:

All institutions of higher education offers to their students the opportunity to do internships in public or private enterprises.

Indeed after the completion of the administrative procedures of an internship, the students are assigned to one or more supervisors depending on the type of the internship. Students are then integrated during the internship period in host companies under the guidance of the company supervisor and also the supervisor of the university.

The internship tracking process is done manually. Indeed the exchange of information and documentation between the supervisor and the student is done by mailing or by papers handwritten.

The final evaluation of the project before the defences is carried out in a short time, the Jury members of an internship have no idea about the progress of an internship or the completion stain.

3- Criticism of the existing:

The existing procedure has achieved these objectives since its inception but the criterion "followed in real time "is absent, in fact:

- Student control is only performed by one part which is connected to the student's location during the internship period
- Lack of collaboration between the university and the companies
- In some cases the supervisors meet only during the defences
- The task assigned by the two supervisors sometimes differs in terms of purpose or goal
- The final evaluation of the project before the defences is carried out in a short time
- Jury members of an internship have no idea about the progress of an internship
- Loss of time in the exchange of documents
- Risk of document loss
- Lack of academic support (Enterprise)
- No weekly meeting between all the speakers.

4- PFE Internship:

Based in Tunis, our application “PFE Internship” is redefining the experience of organizing PFE internship within colleges.

In order to overcome the observed failures, we propose to computerize the process of follow-up courses. So, we opt for the development of an application that will be operated by several stakeholders and its main objective is the monitoring of internships. The application must be able to facilitate the process of traineeship the student to communicate with his or her framers documents and to seize their activities or the progress achieved.

The app also allows supervisors and jury members to track and validate the work done by the student .

Chapter 2: Requirements analysis and specifications

1- Identification of Actors:

Three major actors that will interact with our system:

- An Administrator who masters all the functionalities of the application and have all the premium rights.
- A Super-Admin is the first user created directly in the database how will create an administrator for each school
- A registered user: Students and employees.

2- Functional and non-functional needs:

In this section, we will present the requirements this system has to cover:

Functional needs	
Responsible	Service
A Student	-Register -Sign in -Change his / her account details -Manage the internship agreements -Manage the PFEs
An administrator	-Manage the students -Manage the employees -Manage the university departments
An Internship director	-Sign in -Manage the employees -Manage the internship agreements -Manage the PFEs -Plan internship defences
A Supervisor	-Sign in -Manage the PFEs
A Super-Admin	-Sign in -Manage the Administrators

A teacher	-Sign in - Manage the PFEs
A Head of departments	- Manage the PFEs

Non-Functional needs

- The access to our system should be secure with login and password.
- The system should be able to run a task in a given time.
- The system shall be compatible with different platforms.
- Ergonomics (ISO 9241-110)
  - ❖ Button labels should tell about corresponding action (avoid generic labels like " YES" or "No"
  - ❖ Using tabs and foldable elements
  - ❖ Clear and distinguishable modes

Table 1: Functional and non-Functional needs

3- Global Use Case Diagram:

A use case diagram represents the different types of the system's users and how they interact.

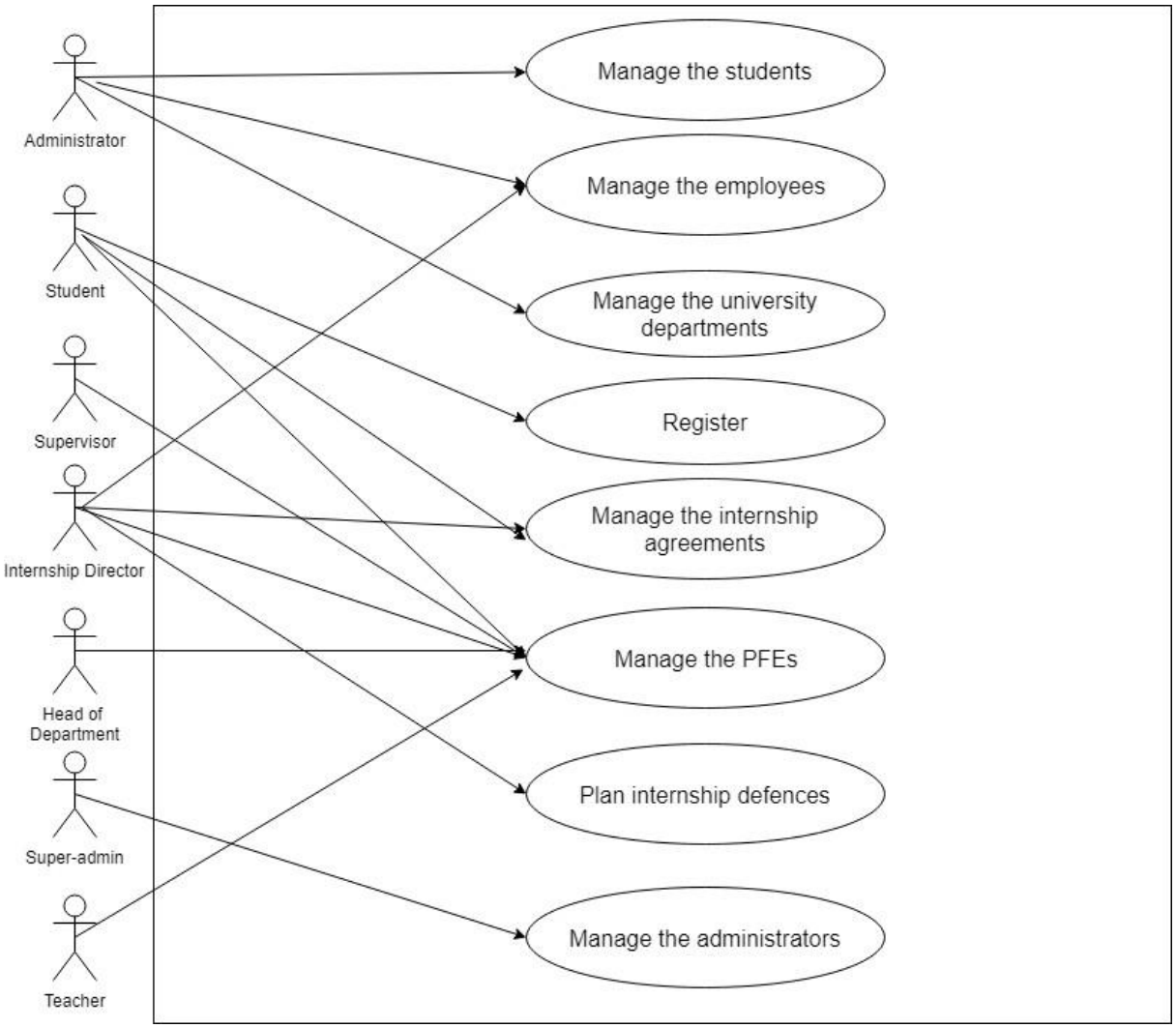


Figure 1 : Global use case diagram

4- Sequence diagram:

A sequence diagram is an interaction diagram that shows how objects operate with one another and in what order.

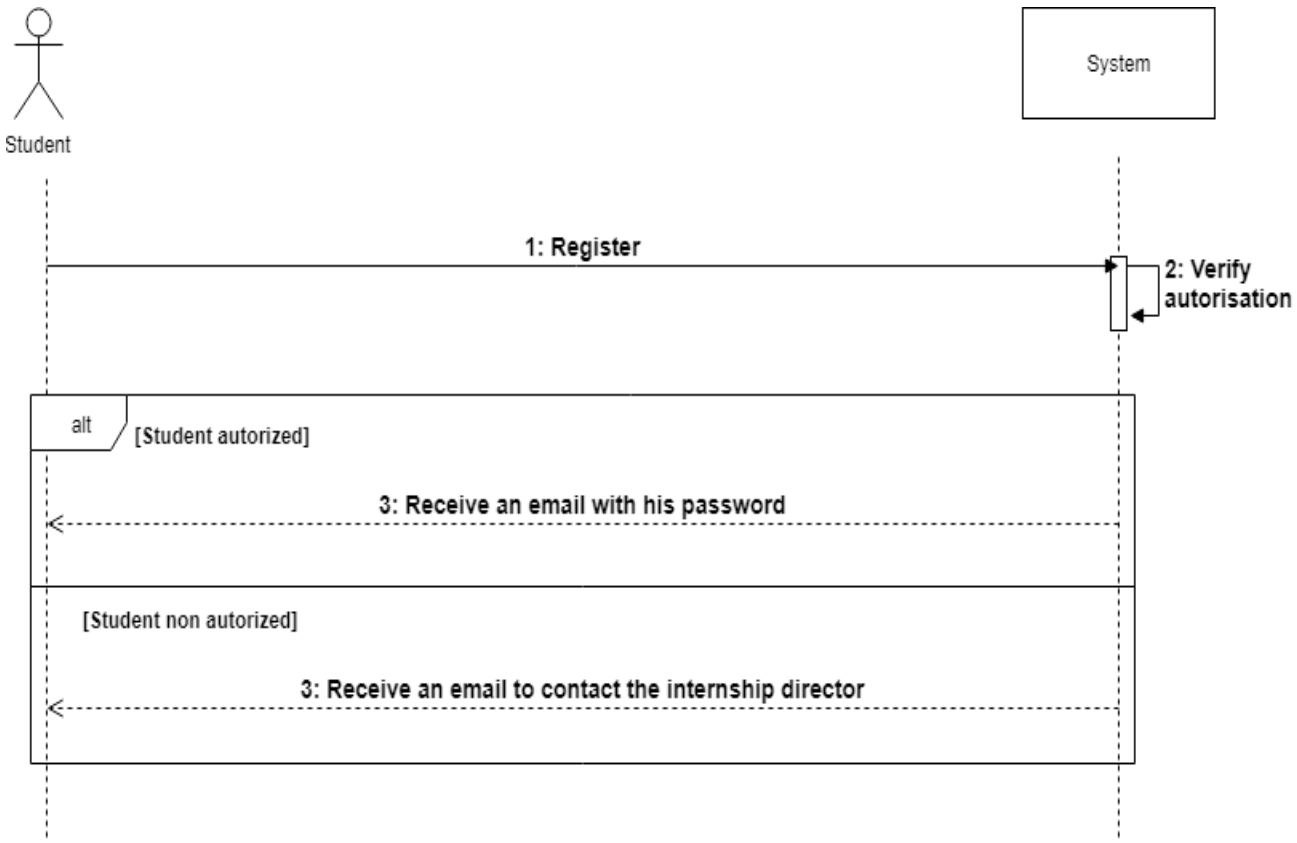


Figure 2: Sequence Diagram: Student register

To have access to the application, first the student has to register. The system will check if he has the authorization to pass his end-of-studies project. Once the student is authorized he will receive an email in his address mail of the school with an auto-generated password. Else, if he is not authorized he will receive an email witch explaines the process to get his authorization by contacting the internship director.

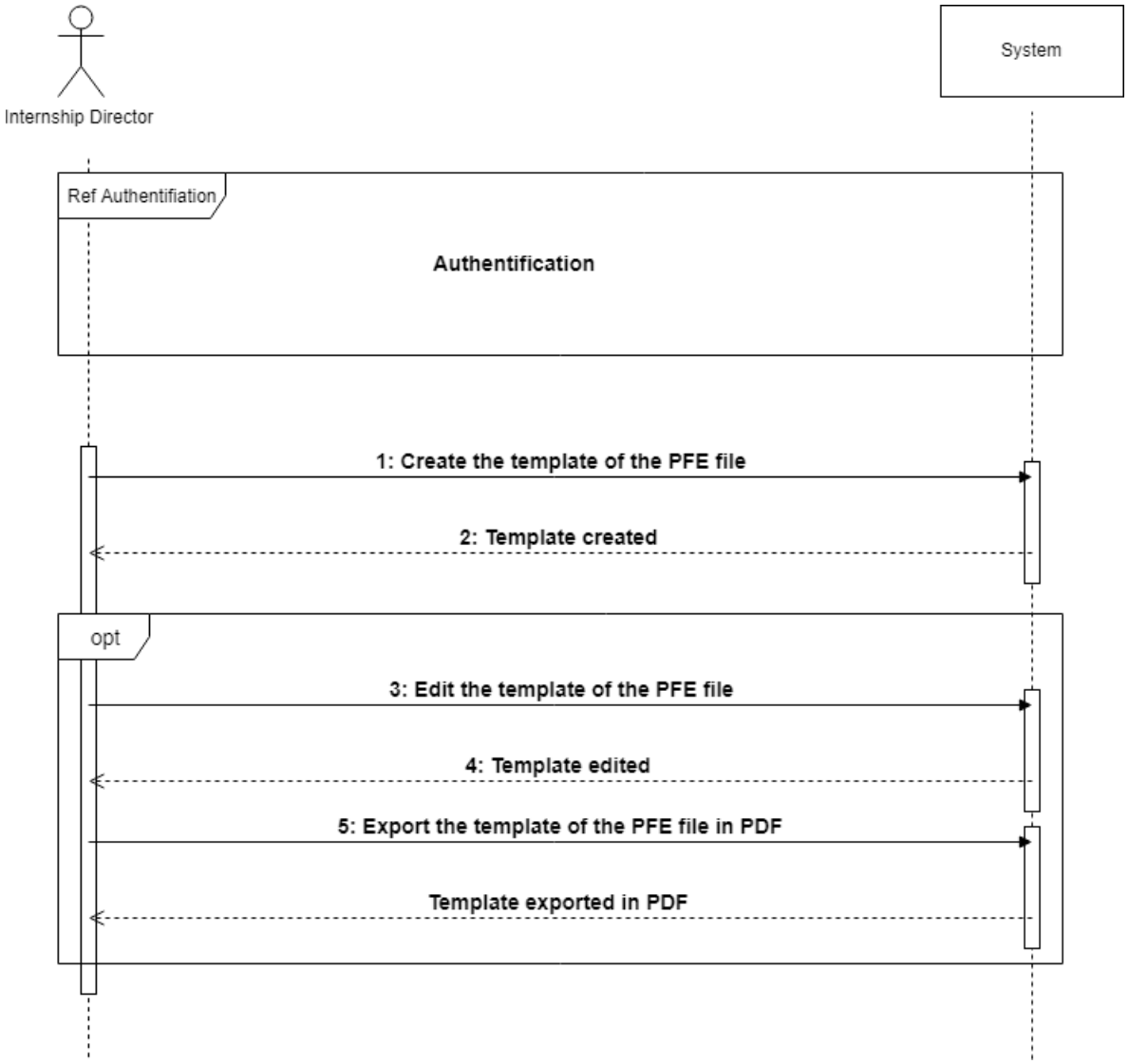


Figure 3: Sequence Diagram: Manage the template of PFE file

After being authenticated, the internship director will have access to manage the template of th PFE file in our application. He can create one with a text editor on line, edit and export it.

Chapter 3: Static analysis

1-Class Diagram

The class diagram expresses the structure of a system in term of classes and relationships between them. Similarly, a class describes a group of objects; an association describes a set of links.

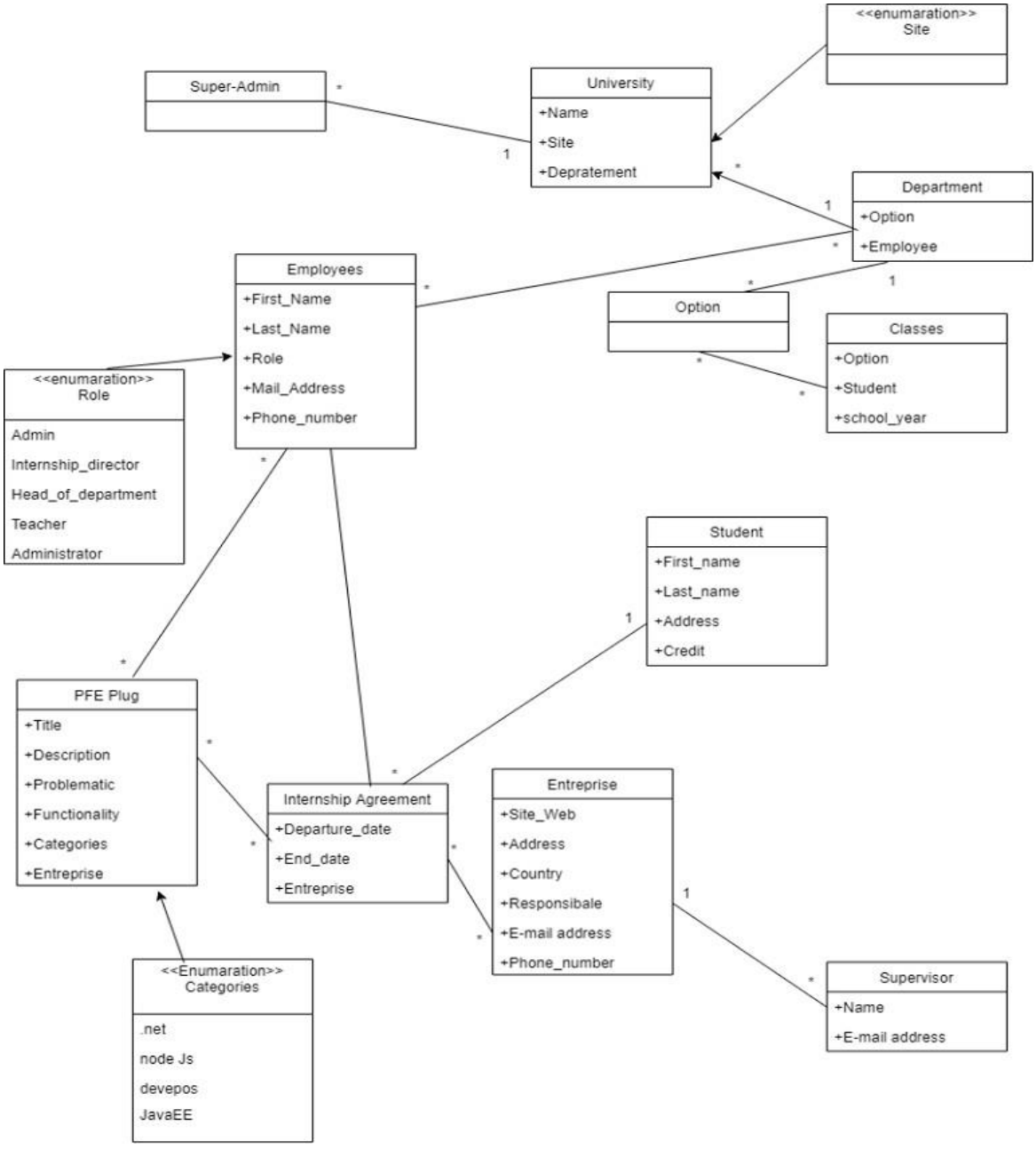


Figure 4: Class diagram

Chapter 4: Modelling and Design

1-Deployment Diagram

A deployment diagram in the UML models the *physical* deployment of components. To describe an application, a deployment diagram would show what hardware components exist (a web server, an application server, and a database server) and how the different pieces are connected (JDBC, HTTP, TCP/IP).

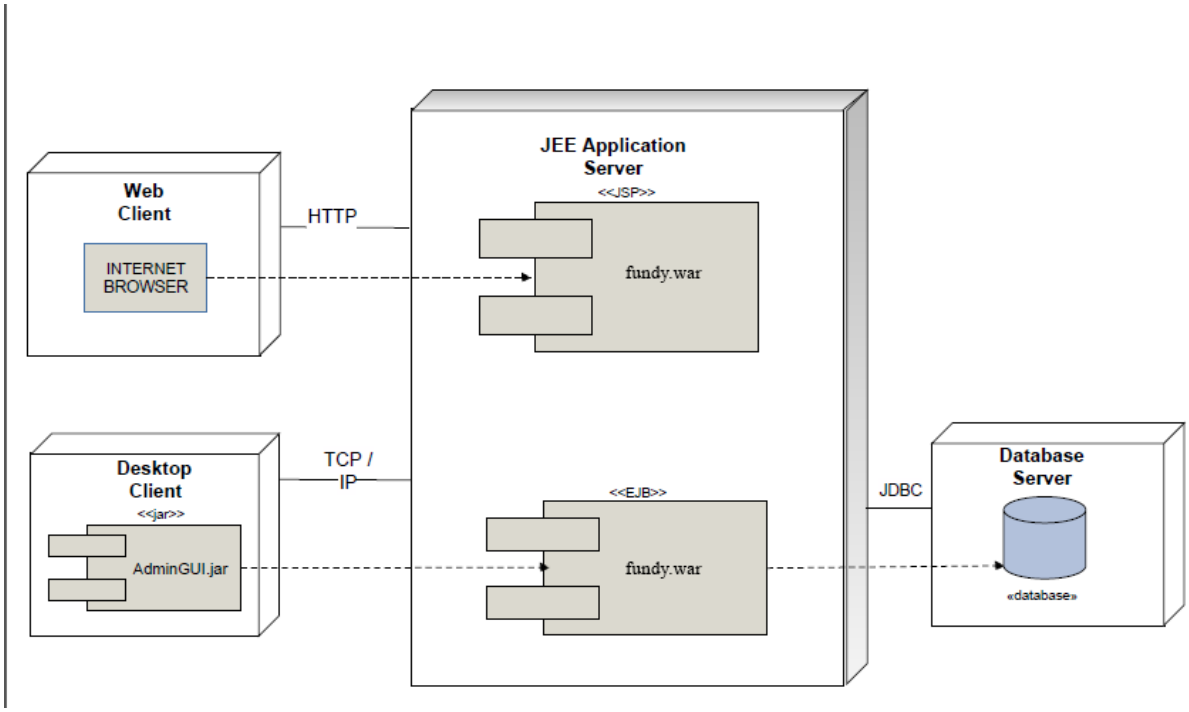


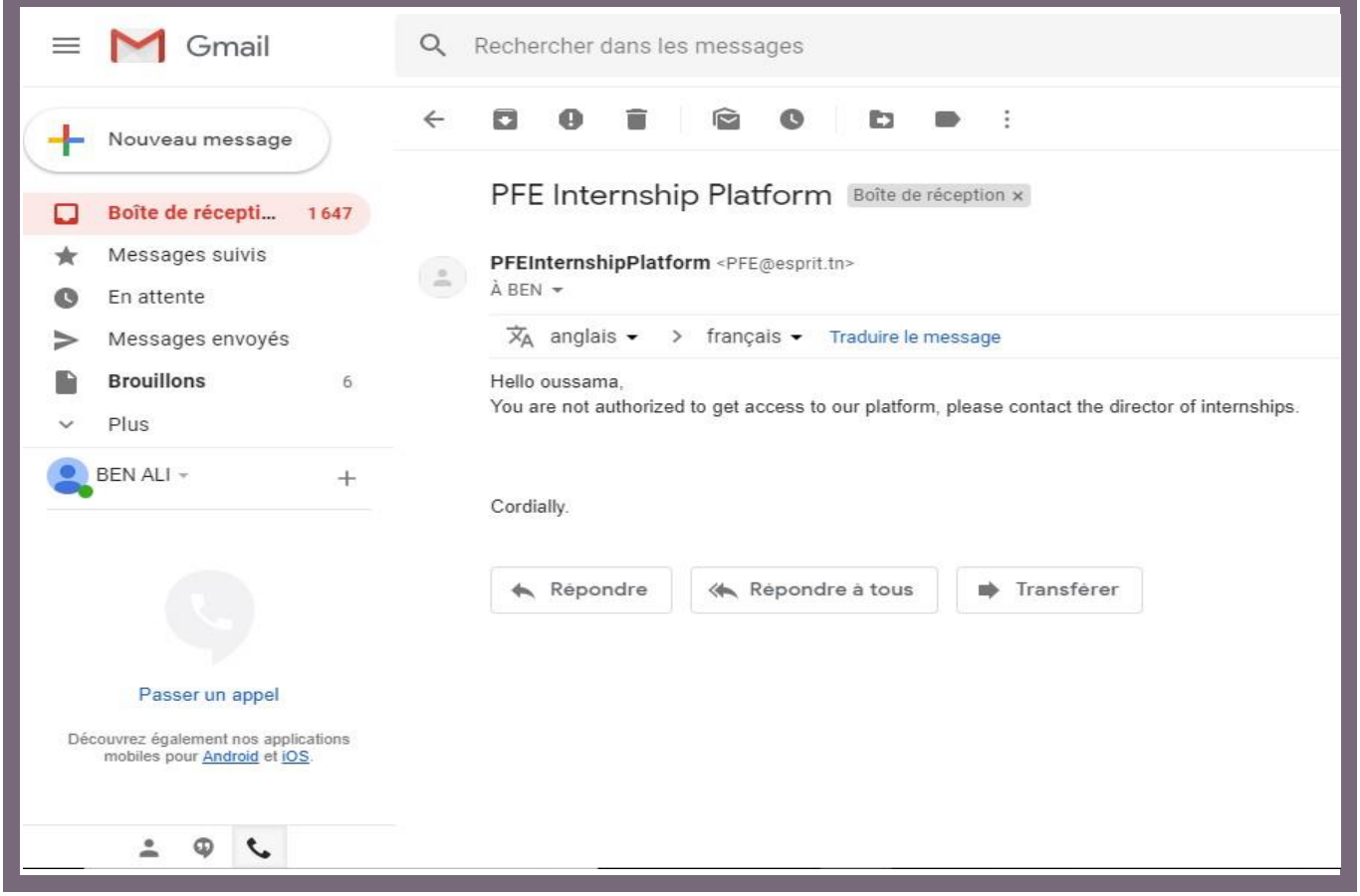
Figure 5 : Deployment diagram

2- Interfaces:

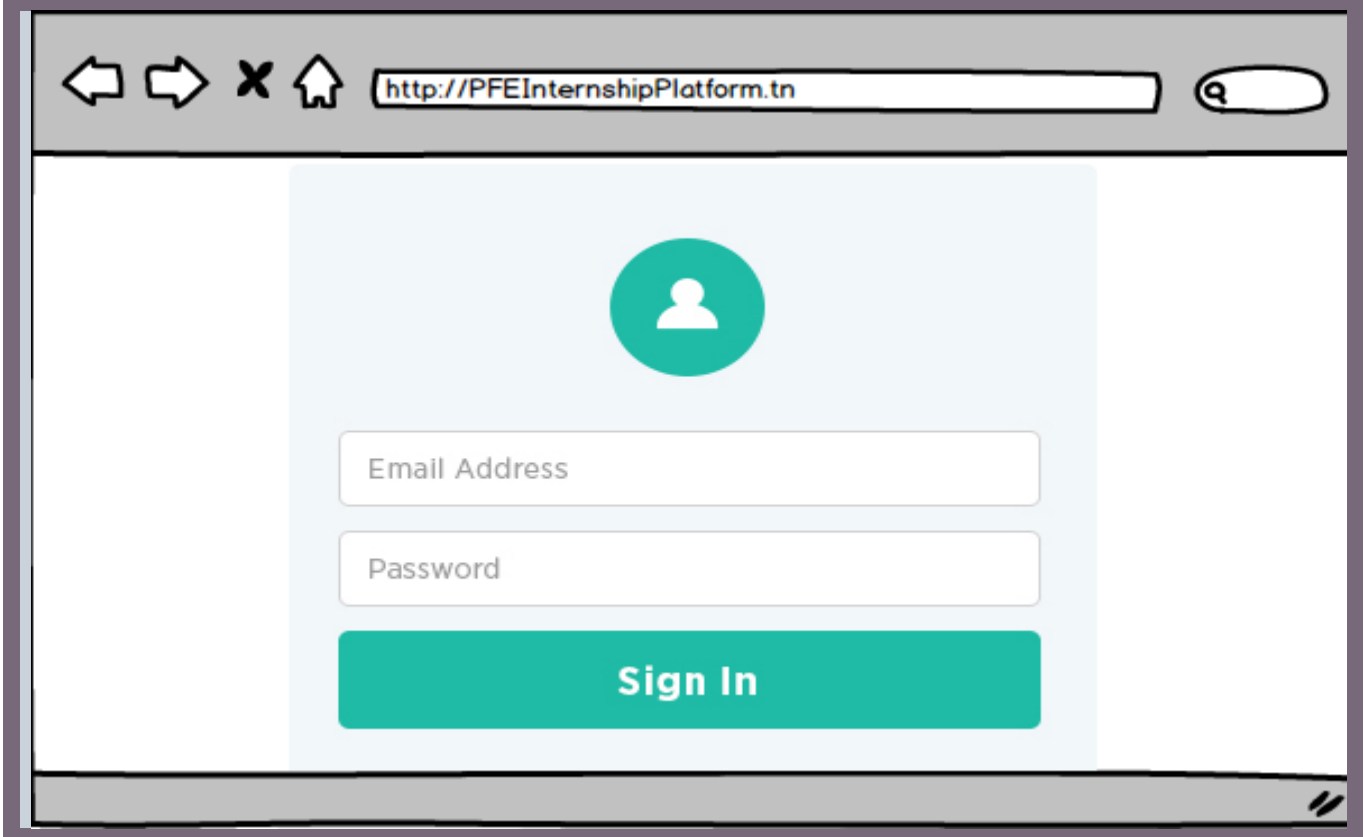
In order to validate ergonomic graphical interfaces, we used Balsamiq Mockups as a tool for graphic design.

The student must first try to register on the platform

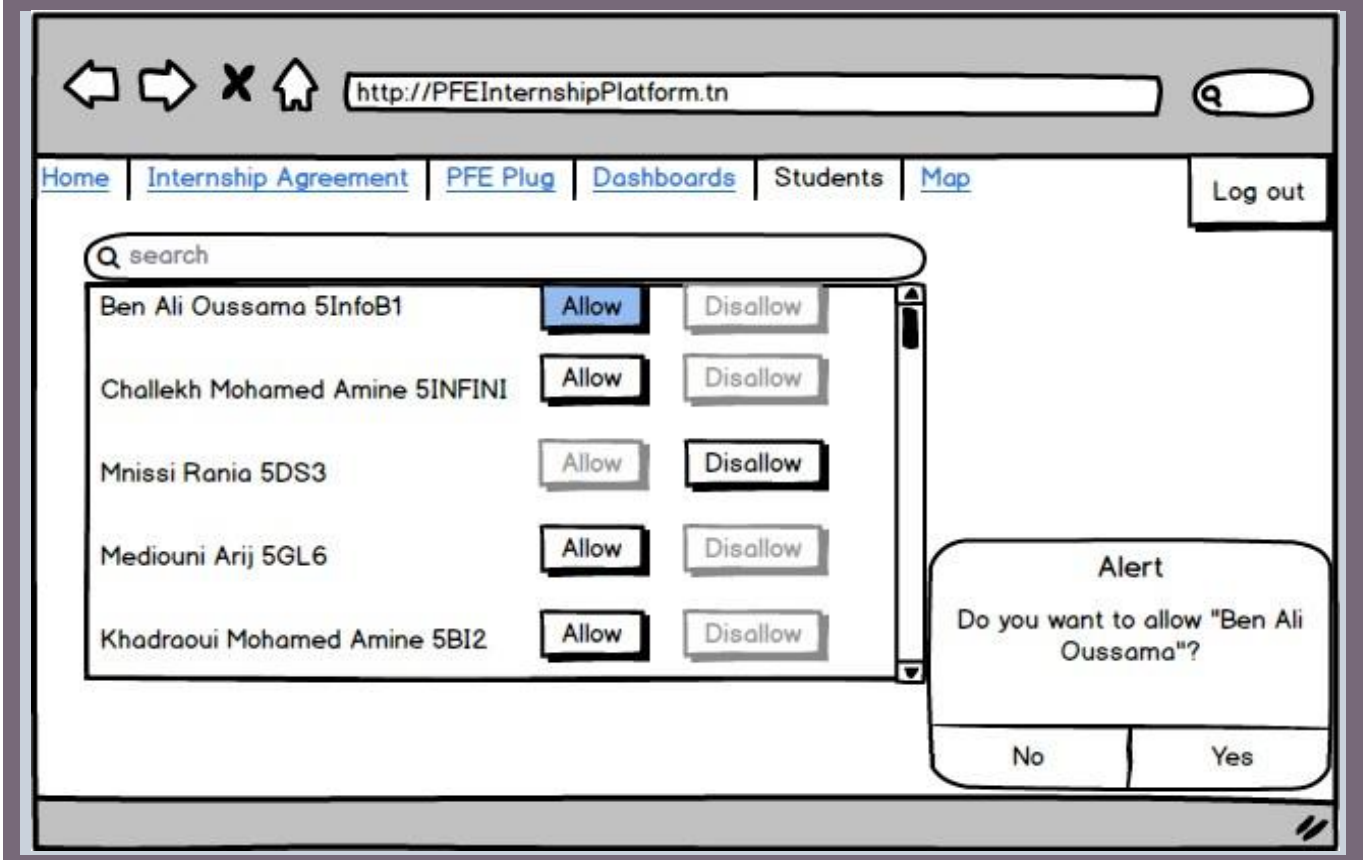
if he is not authorized to pass his PFE, he will have an email that explains to him that it is necessary to contact the director of the courses



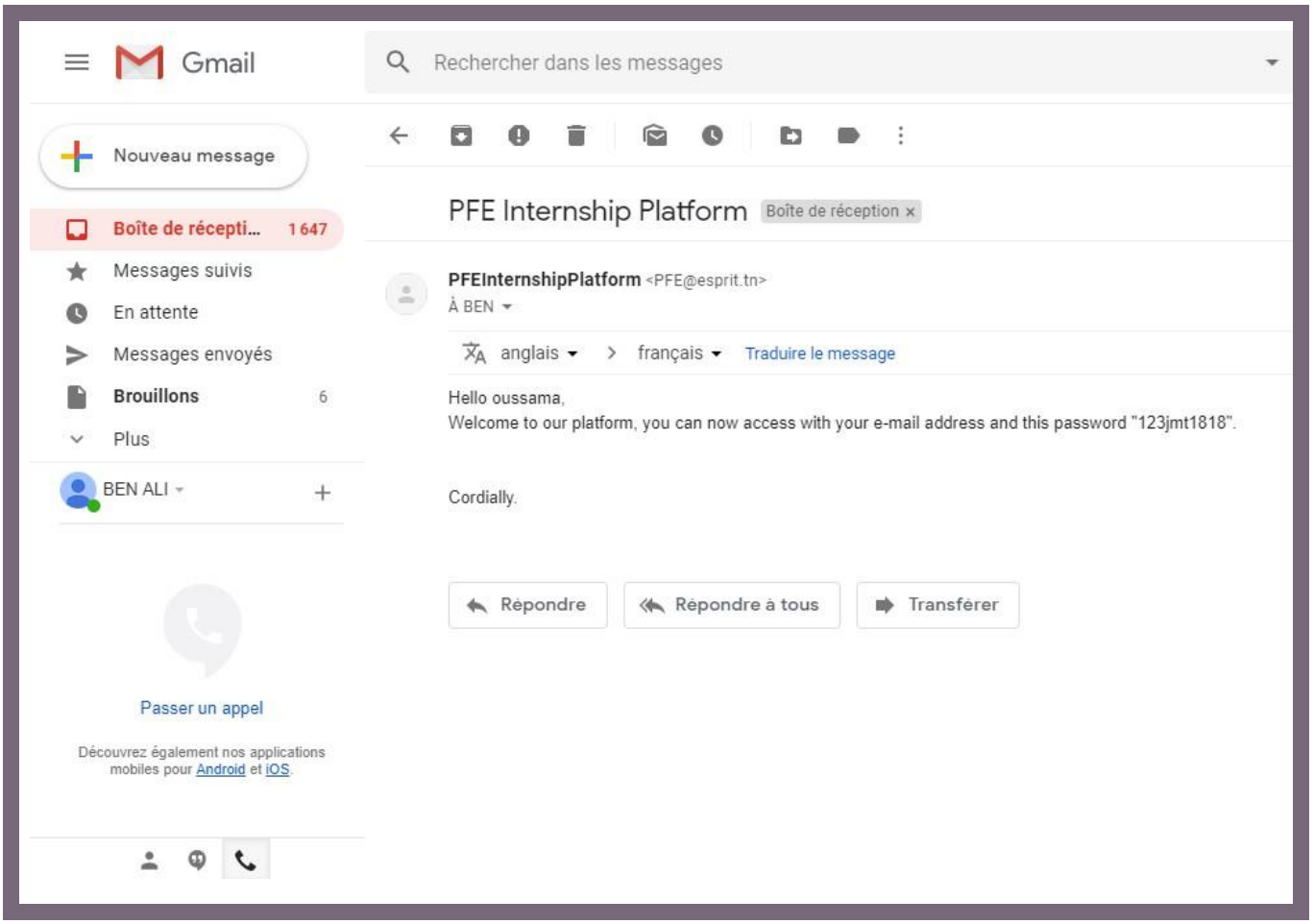
The internship director can sign in via this interface



The internship director will allow the student to register in the platform



The student is now allowed and have the access to the platform



3-Physical and logical architecture Modelling:

The standard architecture for distributed applications separates application logic into a number of tiers.

These tiers signify a logical and physical organization of components into an ordered chain of service providers and consumers.

Components within a tier typically consume the services provided by components in an adjacent provider tier and provide services to one or more components in an adjacent consumer tier.

The logical tier dimension of deployment architecture is illustrated in the following figure.

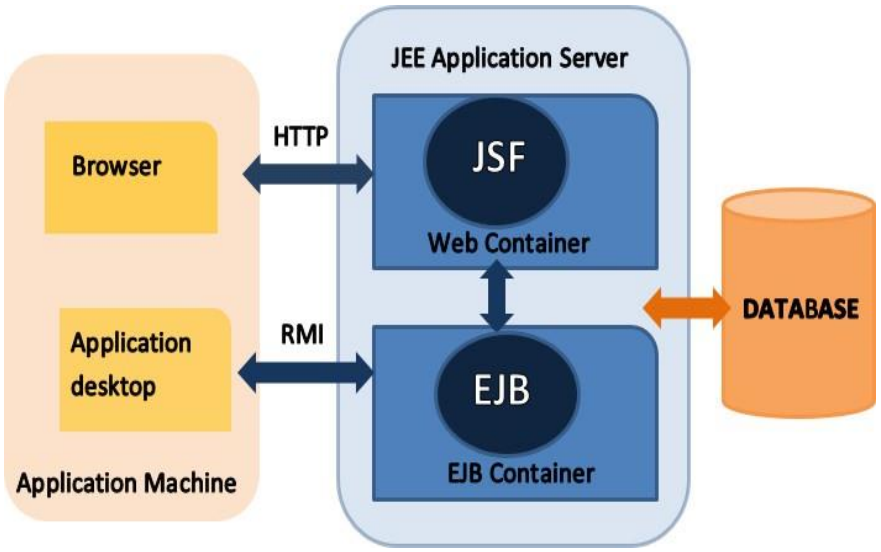


Figure 6 : Physical and logical architecture Modelling

3.1-Physical architecture:

The client tier is the front end layer in the 3-tier system and consists of the user interface.

This user interface is often a graphical one accessible through a web browser or web-based application and which displays content and information useful to an end user.

The application tier contains the functional business logic which drives an application's core capabilities.

The data tier comprises of the database/data storage system and data access layer.

3.2- logical architecture:

The three-tier architecture is a type of software architecture which is composed of three "tiers" or "layers" of logical computing.

They are often used in applications as a specific type of client-server system. It provide many benefits for production and development environments by modularizing the user interface, business logic, and data storage layers.

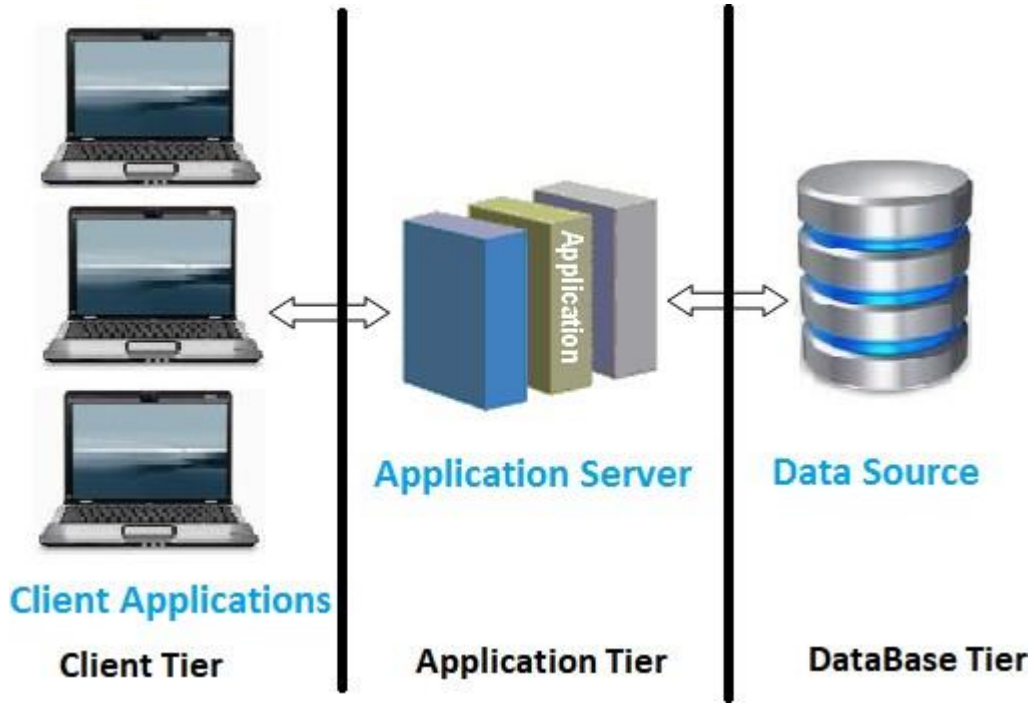


Figure 7: logical architecture

Chapter 5: Software development process

1- Chosen Software and tools:

Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

Jboss isone the best IDE for java development, we have chosen jboss 1.7 for developing the JEE module.

GitHub: For the versioning server and the orchestration between team members, we have chosen the github.

Balsamiq mockups: For creating mockups of the platform

WildFly: is an application server authored by JBoss, developed by Red Hat. WildFly is written in Java and implements the Java Platform, Enterprise Edition (Java EE) specification. It runs on multiple platforms

Final Conclusion

During the first phase of our work, we analysed the subject we were given. We studied the global context and analysed the functional requirements of our application. We have searched and selected the environmental and development tools. We also have modelled the logical and physical architecture.