



Ecole Supérieure Privée  
d'Ingénierie et de Technologies

**Pi-Dev report**

**Realised by:**  
**Dhaouadi Zohra**  
**Nader Kasri**

**Edited by:**  
**Latex**

**Class: 4 GL 3**

**2017-2018**

# Devotion

We have set light, an ever-burning flame of gratitude and deep sense of obligation to our honorable teacher Hajar Berhouma for her valuable guidance, constructive criticism and inspiring attitude during our integrated project.

this program made us realize the value of working together as a team and as a new experience in a working environment, which challenges us every minute. Not forget, great appreciation go to the rest of whom helped us during the project. The whole program really brought us together to appreciate the true value of respect.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Functional branch, Specification and Analyses</b>	<b>3</b>
2.1	Customer specifications . . . . .	3
2.1.1	Customer overview . . . . .	3
2.1.2	Customer requirements . . . . .	5
2.1.3	Methodology design . . . . .	6
2.1.4	Customer specifications . . . . .	7
2.2	Internal specifications . . . . .	11
2.2.1	Functional requirement . . . . .	12
2.2.2	Non-Functional requirement . . . . .	13
<b>3</b>	<b>Global analysis</b>	<b>15</b>
3.1	Global architecture . . . . .	15
3.2	Use case . . . . .	16
3.2.1	Actor identification . . . . .	16
3.2.2	Use case analyze . . . . .	16
3.3	Class diagram . . . . .	18
3.4	Functional Architecture . . . . .	18
3.4.1	Logical Architecture . . . . .	18
3.4.2	The N-Tiers architecture . . . . .	19
3.4.3	Physical architecture . . . . .	20
3.4.4	Road map . . . . .	21
3.5	Conclusion . . . . .	21
<b>4</b>	<b>Technical branch</b>	<b>22</b>
4.1	Software tools . . . . .	22
4.1.1	Platform . . . . .	22

## Contents

---

4.1.2	Programming tools . . . . .	23
4.1.3	Application compiler . . . . .	24
4.1.4	Database deployment . . . . .	25
4.1.5	Java platform implementation . . . . .	26
4.1.6	Web services' consumption . . . . .	26
4.1.7	Application design and analysis . . . . .	27
4.1.8	Collaborative development environment . . . . .	27
4.2	Logical architecture . . . . .	27
4.3	Physical architecture . . . . .	27
4.4	Preliminary and detailed conception . . . . .	28
4.4.1	JAVA EE functional requirement . . . . .	28
4.4.2	Object sequence and object class diagrams . . . . .	28
4.4.3	JEE Analysis class diagram . . . . .	31
4.5	Design Class Diagram (JEE) . . . . .	32
4.6	Design class diagram decomposed into packages (JEE) . . . . .	32
4.6.1	Web package (JEE) . . . . .	33
4.6.2	EJB package (JEE) . . . . .	34
4.6.3	Package Persistance (JEE) . . . . .	35
4.6.4	Component Diagram (JEE) . . . . .	36
4.6.5	Deployment Diagram (JEE) . . . . .	37
4.6.6	Detailed deployment diagram(JEE) . . . . .	39
4.7	Implementation and test . . . . .	39
4.7.1	Application interface . . . . .	39
4.7.2	Home page . . . . .	40
4.7.3	User management . . . . .	41
4.7.4	Task management . . . . .	42
4.7.5	Application management . . . . .	42
4.7.6	Newsletter . . . . .	44
4.7.7	Inbox . . . . .	45
4.7.8	Contact us . . . . .	45
4.7.9	About us/Streaming . . . . .	45

## Contents

---

4.8	Implementation tips and integration . . . . .	47
4.9	Application test . . . . .	47
4.10	Conclusion . . . . .	49
<b>5</b>	<b>Mobile/Web Module</b>	<b>50</b>
5.1	Functional requirements Mobile/W.S . . . . .	50
5.2	Architecture and methodology . . . . .	51
5.2.1	Retrofit2 . . . . .	52
5.2.2	Volley . . . . .	52
5.2.3	M.V.C modeling . . . . .	52
5.3	Preliminary and detailed conception . . . . .	53
5.3.1	Logical architecture . . . . .	53
5.3.2	Physical architecture . . . . .	53
5.3.3	Object sequence diagram . . . . .	54
5.3.4	Class diagram . . . . .	55
5.3.5	Deployment diagram . . . . .	56
5.3.6	Detailed deployment diagram . . . . .	56
5.3.7	Component diagram . . . . .	56
5.4	Coding and test . . . . .	57
5.4.1	Login interface . . . . .	57
5.4.2	SignIN interface . . . . .	58
5.4.3	Library center localization interface . . . . .	58
5.4.4	Library center localization interface . . . . .	59
5.4.5	Generate QR . . . . .	59
5.4.6	ScanQR . . . . .	60
5.4.7	Resultat scanQR console . . . . .	60
5.4.8	Add comment interface . . . . .	61
5.4.9	Delete/Modify comment interface . . . . .	61
5.4.10	Mail interface . . . . .	62
5.4.11	Main and get comments interface . . . . .	62
5.5	Tip of implementation and integration . . . . .	63
5.6	Test the application . . . . .	63

## Contents

---

5.7	Conclusion . . . . .	63
<b>6</b>	<b>Asp.NET</b>	<b>64</b>
6.1	Tools and technologies . . . . .	64
6.2	Language developing tools . . . . .	66
6.2.1	C# language . . . . .	66
6.2.2	Asp.net language . . . . .	66
6.2.3	Entity Framework . . . . .	66
6.3	Design developing tool and template . . . . .	67
6.3.1	CSS3,HTML,Javascript . . . . .	67
6.3.2	Razor CsHtml . . . . .	67
6.3.3	jQuery . . . . .	68
6.4	MYSQL Connector . . . . .	68
6.5	.NET functional requirement . . . . .	68
6.6	Logical architecture . . . . .	69
6.6.1	.NET web reference logical architecture . . . . .	70
6.7	Physical architecture . . . . .	70
6.8	Object Sequence Diagram . . . . .	71
6.8.1	Login sequence diagram . . . . .	71
6.8.2	Admin Leave sequence diagram . . . . .	71
6.8.3	Agent Leave sequence diagram . . . . .	72
6.9	Class diagram . . . . .	73
6.10	Component diagram . . . . .	74
6.11	Deployment diagram . . . . .	74
6.12	Package diagram . . . . .	75
6.13	Application interfaces . . . . .	75
6.13.1	Login interface . . . . .	75
6.13.2	Agent leave interface . . . . .	76
6.13.3	Administrator leave management interface . . . . .	76
6.13.4	Mail management interface . . . . .	76
6.14	Test the application . . . . .	77

Contents

---

<b>7 Conclusions and perspectives</b>	<b>79</b>
---------------------------------------	-----------

# List of Figures

2.1 I.R.M.C Logo . . . . .	4
2.2 2-TUP approach . . . . .	6
2.3 I.R.M.C Home page . . . . .	8
2.4 Current I.R.M.C location . . . . .	9
2.5 Current Scientific Watch . . . . .	9
3.1 Global architecture . . . . .	15
3.2 Use Case diagram . . . . .	17
3.3 Class diagram . . . . .	18
3.4 Global functional Architecture . . . . .	19
3.5 Functional Architecture . . . . .	19
3.6 N-Tiers architecture . . . . .	20
3.7 Deployment diagram . . . . .	20
3.8 Road map . . . . .	21
4.1 The Enterprise Java platform (Java EE) . . . . .	23
4.2 Jboss developer studio 9 . . . . .	23
4.3 Java Development Kit . . . . .	24
4.4 Java Runtime Environment . . . . .	24
4.5 Microsoft visual studio logo . . . . .	24
4.6 Android studio logo . . . . .	25
4.7 MySQL server . . . . .	25
4.8 MySQL connector . . . . .	25
4.9 MySQL Workbench . . . . .	26
4.10 Wildfly x.9 server . . . . .	26
4.11 Client REST . . . . .	26
4.12 StartUML application designer . . . . .	27

---

## List of Figures

---

4.13 Collaborative development environment . . . . .	27
4.14 Java EE Package diagram . . . . .	28
4.15 Detailed Java EE Package diagram . . . . .	28
4.16 Global java EE deployment diagram . . . . .	29
4.17 JAVA EE functional requirement . . . . .	29
4.18 Global java EE deployment diagram . . . . .	29
4.19 Object sequence diagram "Add SessionUser" . . . . .	30
4.20 Object sequence diagram "Delete sessionTask" . . . . .	30
4.21 Object sequence diagram "Upload applications and documents" . . . . .	31
4.22 Object sequence diagram "Send mails" . . . . .	32
4.23 JEE Analysis class diagram . . . . .	33
4.24 Design Class Diagram (JEE) . . . . .	34
4.25 Web package (JEE) . . . . .	35
4.26 EJB package (JEE) . . . . .	36
4.27 Package Persistence (JEE) . . . . .	37
4.28 Component Diagram (JEE) . . . . .	38
4.29 Deployment Diagram (JEE) . . . . .	38
4.30 Deployment diagram(JEE) . . . . .	39
4.31 Sign in . . . . .	40
4.32 Sign out . . . . .	40
4.33 Home page . . . . .	41
4.34 User management . . . . .	42
4.35 Task management . . . . .	43
4.36 Application management . . . . .	44
4.37 Applications list . . . . .	45
4.38 Application comment . . . . .	46
4.39 Newsletter . . . . .	46
4.40 Inbox . . . . .	47
4.41 Contact us . . . . .	48
4.42 Mail reception . . . . .	48
4.43 About us/Streaming . . . . .	49

---

## List of Figures

---

5.1	Technical branch . . . . .	51
5.2	Web service functional requirements . . . . .	51
5.3	Retrofit2 android methodology . . . . .	52
5.4	Volley android methodology . . . . .	52
5.5	M.V.C architecture . . . . .	53
5.6	Logical architecture . . . . .	53
5.7	Physical architecture . . . . .	53
5.8	Authentication sequence diagram . . . . .	54
5.9	QrCode generator sequence diagram . . . . .	54
5.10	QrCode scan sequence diagram . . . . .	55
5.11	Comment on jobs offer sequence diagram . . . . .	55
5.12	Class diagram . . . . .	55
5.13	Deployment diagram . . . . .	56
5.14	Deployment diagram . . . . .	56
5.15	Component diagram . . . . .	57
5.16	Login interface . . . . .	57
5.17	SignIN interface . . . . .	58
5.18	Library center localization interface . . . . .	58
5.19	Library center localization interface . . . . .	59
5.20	Generate QR . . . . .	59
5.21	ScanQR . . . . .	60
5.22	Resultat scanQR console . . . . .	60
5.23	Add comment interface . . . . .	61
5.24	Delete/Modify comment interface . . . . .	61
5.25	Mail interface . . . . .	62
5.26	Main and get comments interface . . . . .	62
6.1	Restful Web-API . . . . .	65
6.2	Asp.net C . . . . .	66
6.3	Asp.net developing tool . . . . .	66
6.4	ADO.NET: Entity framework . . . . .	67
6.5	CSS3,HTML,Javascript for template design . . . . .	67

---

## List of Figures

---

6.6	Razor CsHtml . . . . .	67
6.7	jQuery . . . . .	68
6.8	MYSQL Connector . . . . .	68
6.9	.NET functional requirement . . . . .	69
6.10	.NET functional requirement . . . . .	69
6.11	.NET web logical architecture . . . . .	70
6.12	Physical architecture . . . . .	70
6.13	Login sequence diagram . . . . .	71
6.14	Admin Leave sequence diagram . . . . .	72
6.15	Agent Leave sequence diagram . . . . .	73
6.16	Class diagram . . . . .	73
6.17	Component diagram . . . . .	74
6.18	Deployment diagram . . . . .	74
6.19	Package diagram . . . . .	75
6.20	Login interface . . . . .	75
6.21	Agent leave request interface . . . . .	76
6.22	Administrator leave management interface . . . . .	76
6.23	Mail management interface . . . . .	77

# List of Tables

2.1 Basic functional requirements . . . . .	12
3.1 Actors Identification . . . . .	16
3.2 Description of consulting and add a document action . . . . .	16

## List of acronyms

I.R.M.C : The Institute for Research on the Contemporary Maghreb

2-TUP : Two Tracks Unified Process

# 1

## Introduction

The number of internauts using the web as a reference for their bibliography is constantly increasing throughout the world. Unfortunately, to the entire users at the I.R.M.C (Institute of Research on the Contemporary Maghreb), each and everyone of them experiences daily difficulties using the official site of the institute depending on the field where they're looking for reference.

After thoroughly studying the harsh context of these institute , we have decided to create a project which aims to help their followers in terms of the quality of their services, activities and essential necessities for a better deliverance of information that could be offered to them thanks to the manifold services that we'll propose in our plateform.

Our project will be divided into 3 parts where each part has its own technology : Java Entreprise Edition application for developping our remoted services, DotNet and JSF for a web application and our GUI representation to the required services.

Our integrated project is to maintain an already deployed official site for the IRMC based on the 2-TUP methodology.

The principle of this methodology is to develop a software upgraded incrementally maintaining a fully transparent list of requests for changes or corrections to implement.

The method is based on iterative development at a constant rate for a period of branches.

## Chapter 1. Introduction

---

In fact the success of any study depends on the quality of its departure. Therefore, throughout this chapter, which represents the preparatory phase of the project we concentrated on presenting an over view of the project.

# 2

## Functional branch, Specification and Analyses

### Introduction

On the aim to present you an over view on the project, we went through a global analyse. Which will be presented in the next chapter.

### 2.1 Customer specifications

#### 2.1.1 Customer overview

The Institute for Research on Contemporary Maghreb (IRMC) is a research center in the humanities and social sciences, with a regional vocation, based in Tunis. It is one of 27 French research institutes abroad (IFRE), spread over five continents, with a strong presence in the Mediterranean and the Near and Middle East, including Asia Minor and Central (Morocco, Tunisia, Egypt, Lebanon, Syria, Jordan, Turkey, Iran, Uzbekistan, Yemen).

The IRMC is under the supervision of the Ministry of Foreign and European Affairs and, since 2000, the Ministry of Higher Education and Research, and the National Center for Scientific Research (CNRS) of which it constitutes a Joint Unit (USR 3077).

The IRMC contributes, in partnership with the scientific community, notably Maghreb and European, to the development of research on the Maghreb (Tunisia, Algeria, Libya, Mauritania). Its vocation is to stimulate a transversal research catalyzing the energies and the solidarities scientists intramaghrébines and Maghreb /Europe. It also involves fostering the renewal of the social sciences, helping to prepare the next generation of researchers in research centers and universities in France and the Maghreb. As such, the IRMC is a place of training and exercise of the profession of researcher, and training in scientific practice and culture. It constitutes the first concrete experience of confrontation with senior researchers (CNRS) and conduct of a collective research for post-doctoral researchers, as well as the first sustainable immersion in a research unit for PhD students.

His fields of study cover all human and social sciences: anthropology, demography, law, economics and management, urban studies, geography, history, political science, social sciences applied to literature and philosophy, sociology.



**Figure 2.1:** I.R.M.C Logo

The head office: Tunis,

Year of creation: 1992,

Type of company: Public administration,

Cut: 11-50 employees,

Skills: Research Library, Research in Humanities and Social Sciences, Research Training,

Publishing / Publication.

Domain:

His fields of study cover all human and social sciences: anthropology, demography, law, economics and management, urban studies, geography, history, political science, social sciences applied to literature and philosophy, sociology.

## **2.1.2 Customer requirements**

### **2.1.2.1 Directory of documentation centers and libraries**

The realization from data already collected in the form of fields (type, domain, access, contact, etc.) in an excel file should be automated with a geolocated tool listing all the research libraries.

The aim is to make up for the lack of a repository documentation structures in Tunisia and at regional level. See attachments.

### **2.1.2.2 Mediterranean scientific watch**

The realization for a data collection tool (manually or via rss feeds), also geolocation, to carry out a regional documentary watch with displaying bookmarks with 3 specific color codes (Activities, Calls, Jobs / Scholarships).

The tool must allow Internet users to propose events that would be posted online only after validation of the webmaster. The tool should be able to transcriber automatically as bookmarks on an interactive map of previously integrated rss feeds. In addition to the map interface, the possibility of Sort the information as lists about any I.R.M.C.

### **2.1.2.3 Digitization, editing and indexing filmed lectures**

Designing a visual type with the editing and digitization of conferences in an adaptable format to be made available on a dedicated area.

The only requirement is the possibility of sharing these conferences in an online format and to have a reasonable volume.

#### 2.1.2.4 A mobile application

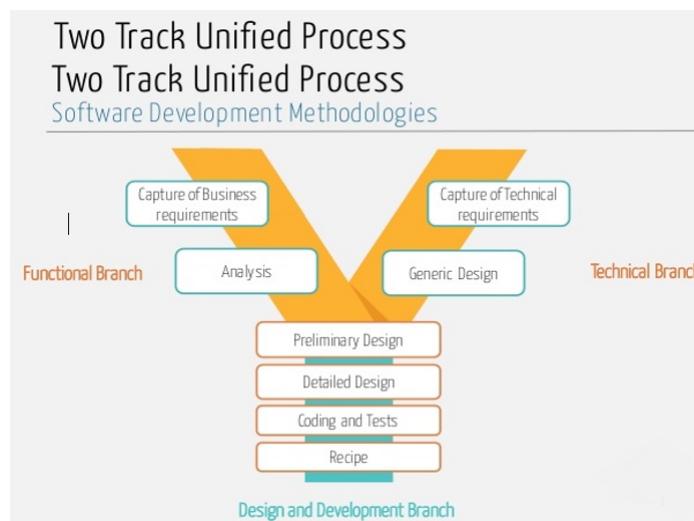
A mobile application containing useful information (team, contact, plan access), an intranet (priviledge to access documents) and the agenda of activities (system of QR code that will be added to our posters).

### 2.1.3 Methodology design

During this project we'll be followin the 2-TUP (Two Track Unified Process).

2TUP offers a development cycle in Y, which dissociates the technical aspects from the functional aspects. It starts with a preliminary study that essentially consists in identifying the actors who will interact with the system to be built, the messages that the actors and the system are exchanging, to produce the specifications and to model the context (the system is a box black, the actors surround him and are connected to him, on the axis which links an actor to the system one puts the messages that the two exchange with the direction).

The process is then articulated around three essential phases: The Unified Process method



**Figure 2.2:** 2-TUP approach

is an iterative and incremental method. The 2TUP separates the technical aspects from the functional aspects and revolves around 3 different branches: the technical branch, the functional branch and the implementation phase:

- The functional branch capitalizes on the knowledge of the business of the company. This branch captures business needs, resulting in a model focused on the end-user business.

- The technical branch capitalizes on technical know-how and / or technical constraints. The techniques developed for the system are independent of the functions to be performed.

- The implementation phase consists of bringing the two branches together, allowing for an application design and finally the delivery of a solution adapted to the needs.

## **2.1.4 Customer specifications**

### **2.1.4.1 Project description**

the Research Institute on Contemporary Maghreb (IRMC), would like to upgrade its website. As a matter of fact, they would like to collect and display all data related to Social Sciences on their website in order to create a Scientific Directory. They also need a mobile application to make their information and data more accessible.

A platform that connects internauts to IRMC actions in their community or any specified field.

### **2.1.4.2 Functionnal requirements**

For our users,a visitor can:

Log into the platform and create a profile,

Make advanced research on fields by author, date, type of article, key words..

Join a debate in an instant chat.

See writers' profiles and their actual/previous actions.

Contact the institute to ask for information.

Rate,comment documents and published articles after readying or adding it.

An I.R.M.C administrator can:

Log into the platform and add a freelancer profile.

Add a document,article,video.

See potential freelancers profiles (suggested by the platform) and invite them to join an action.

Manage Documents/Articles/Videos/users/

See statistics related to I.R.M.C and generate custom reports

#### 2.1.4.3 Analysis of the existence

People seeking to get involved with I.R.M.C activities abroad often get lost in an avalanche of overwhelming information, much of it peddled by middlemen and placement agencies. In an industry heavily dominated by corporate matchmakers, it has become difficult and expensive - to find needy and reputable local writers seeking for their target readers.

Of the same object, we finds only social networks to connect reader, But that it's not enough, so here we need a platform who is already existing but they are intended for a maintain and adding an innovative services for a better performance. Currently, the already deployed website of I.R.M.C is a static web site with a user friendly interface and quite easy to use. However, it still present different limits on a scale of its performance and data. The website regroups all the IRMC's information such its research teams and its research programs.

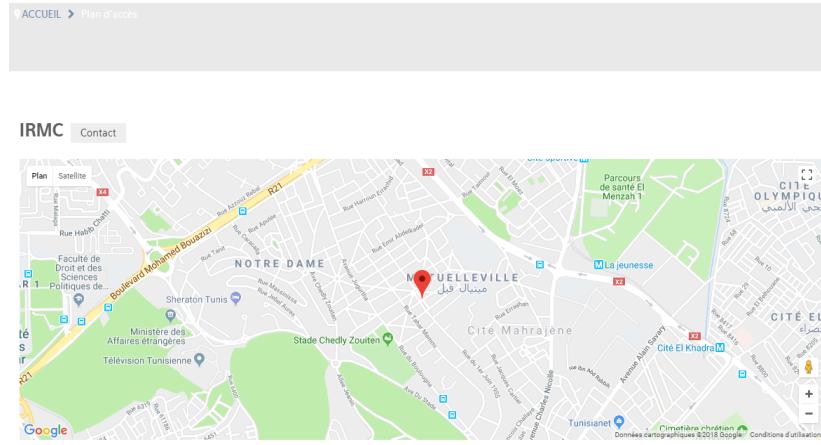
On the homepage, we have access to 2 different menus and a list of articles which are classed by theme and date.



**Figure 2.3:** I.R.M.C Home page

The most interesting section of the website, and the one that interests us the most, is the

“Resources” section. This section has a directory of Tunisian centers, as well as a Mediterranean scientific watch. The directory’s page is in a Google Map format, which allows us to zoom in and out on a world map to find scientific centers, such as universities. It also allows us to search for specific keywords yet it’s all a static implementation of data.



**Figure 2.4:** Current I.R.M.C location

The Mediterranean scientific watch is actually not ready yet, a temporary third-party solution using Timeline JS is just implemented. Unfortunately, It doesn’t assure a high availability or a load balancing service ("thumbnail not always working, redirects are buggy"), this is why, we consider the solution is not optimal.



**Figure 3:** Current Scientific Watch

**Figure 2.5:** Current Scientific Watch

In terms of resources, the IRMC’s website is quite lackluster.

Granted, it allows us to look for scientific for some of the scientific centers, but what if users

would like to look for documentation centers?

Or what if users would like to look for projects and ideas for theses?

or With the trend of browsing the Internet with a cell phone?

How can a respected research facility not have its own phone application?

As we noticed, the current solutions are either non-existent or lacking. To remedy the situation, we will implement proper solutions to these problems by adding a map for documentation centers and libraries, our own 100% coded scientific watch, an Android application as well as other features.

#### **2.1.4.4 Problematic**

We see that IRMC finds many difficulties to connect to their archive of documents.in the other side many visitors to their site have been disappointed that they can not attend an event for their network limits.

So here we found that only social networks ,we need a platform who serve us to communicate on their actions and makes it easier to inform and publicized.

#### **2.1.4.5 Proposed Solution:**

We want to level the documentation field and make readers abroad accessible to it all. We believe everyone should be able to find the cause they want to support and that doing.

This is why, it should be simple and free.We also believe that enabling direct dialogue between internauts and the local teachers' institute increases the chances of a better match of researches, leading to a better experience for both organizations and interns.We want to return reading to its roots, before all the noise, clutter and financial interests took over.

They should be able to get along by connecting and solving problems together.We added a smart geolocalisation system to facilitate the organisation of any kind of research or IRMC on way of destination action. All search services are free because we don't believe in paid matchmaking when it comes to inform and get enriched . We do however intend to stick around for the long run and will be deriving our income from optional travel-related services offered through our platform.

## 2.2 Internal specifications

Any project's requirements need to be well thought out , balanced and clearly understood by all involved , but perhaps of most importance is that they are not dropped or compromised halfway through the project. The IRMC would like to add new services to its information system. Some of these services aim to bring more coherence and fluidity to certain workflows such as the processing of applications for job offers and scholarships.

The workflow's goal is to filter applications. It begins with the entry of an offer by an IRMC researcher to which is attached a set of questions to be answered by the candidate. The candidate can apply by adding personal information such as his e-mail and attaching documents.

The researchers would be able to view these applications, approve them and contact said candidates.

The IRMC would like to centralize as much information and resources as possible by adding 2 directories:

The first would be for documentation centers and libraries. A documentation center is defined by a name, an acronym, a city, an address, a type, a field, a set of links, a phone number and an e-mail.

These centers would be reachable in an interactive map as well as in an organized list.

The second directory is a scientific watch which promotes publications for scholarships, jobs and events. A publication is defined by a description, a field and a date.

Our solution would extract data via 3 different ways:

An RSS feed, a manual input by IRMC researchers or suggestions by visitors that would need to be checked and approved by an admin. Like the other directory, publications would be reachable in an interactive map as well as in an organized list. Another important feature is an online archive of the IRMC's video conferences and scientific journals.

Finally, the IRMC would like to have its own Android application which is basically a mobile Intranet that not only allows users to sign up and sign in to have access to private documents, but also to display information about the IRMC.

### 2.2.1 Functional requirement

In this section, we present the exigency this system has to cover. The definition of a functional requirement is Any requirement which specifies what the system should offer.

In other words , a functional requirements will describe a particular behavior of function of our system when certain conditions are met . For example:make advanced research on documents as a user , adding articles and tutorials as an IRMC worker.

Module	Sub-module	Functionality Code	Functionality	Responsible
JEE	<i>Publication management</i>	Jee01	Add publication	
		Jee02	Delete publication	
		Jee03	View publication w/ Google Maps API	
		Jee04	Update publication	
	<i>RSS Feed management</i>	Jee05	RSS Feed API	
.NET	<i>Researcher profile management</i>	.NET01	Sign-up	
		.NET02	Login	
		.NET03	View profile	
		.NET04	Update profile	
	<i>Documentation Center Directory management</i>	.NET05	Add center	
		.NET06	Delete center	
		.NET07	View center w/ Google Maps API	
		.NET08	Update center	
Mobile	<i>QR Code Management</i>	Mob01	QR Code API	
		Mob02	Sign-up	
		Mob03	Login	
		Mob04	View profile	
		Mob05	Update profile	
		Mob06	View documents	
		Mob07	View IRMC information	
		Mob08	View activity calendar	
	<i>Schedule Management</i>			

**Table 2.1:** Basic functional requirements

As we'll be working on maintaining a contact sub-module between the internaut and the administrator,while generating reports and statistics for the back-end admin.

### **2.2.2 Non-Functional requirement**

Any requirement which specifies how the system perform a certain function. In other words, a non-functional requirement will describe how our system should behave and what limits there are on its functionality.

Non-functional requirements generally specify the system's quality attributes or its characteristics, for example: "Modified data in a database should be updated for all users accessing it within 2 seconds."

To resume, This application must satisfy several non-functional requirements such as:

The application may offer a security level to ensure the integrity of data from accidental or malicious damage.

Reliability: The application should, under normal conditions, perform the required functions successfully.

Usability: The user should not have to spend more than 10 second to load the map on the interface.

Availability: Users can edit or send information at any time.

#### **2.2.2.1 Security**

The application has to offer full security for the client and consultant to manage any information which means accessible and usable only in authorized ways by authorized users.

#### **2.2.2.2 Performance**

This one shouldn't come as a surprise. Quality software has to be fast. Or at least feel fast. "It's not fast enough" is a battle we never want to get into. When we think about an application being performant, we must think about specifying the following:

#### **2.2.2.3 Response times**

How long should my app take to load? What about screen refresh times or choreography?

Processing times:

How long is acceptable to perform key functions or export / import data?

Query and Reporting times:

This could be covered off with general reporting times, in case of providing an API we should probably consider acceptable query times too.

Capacity and Scalability:

How much do we need to cope with now, and how much do you think we'll need to cope with in the future. Unless you are truly blessed with an incredibly forgiving audience (or a great illustrator), you won't be able to get away with your own fail whale.

Availability

Our application must be available, users can edit or send information at any time.

Hours of operation

When does our application need to be available: If we need to do a database upgrade or a system backup, can we take the system offline while you do it?

Locations of operation

A few things to think about here: Geographic location, connection requirements and the restrictions of a local network prevail.

Reliability

Our application has to be 100% reliable, clients should rely on it social networking services.

Usability

Our application has to be easy to use, with standard looking interfaces and with clear and beautiful visual style.

and we end by,Maintainability:

Our application must be quickly maintainable.

## Conclusion

This chapter is a brief presentation of this report scope, it contains a small description of the key elements for our project.

# 3

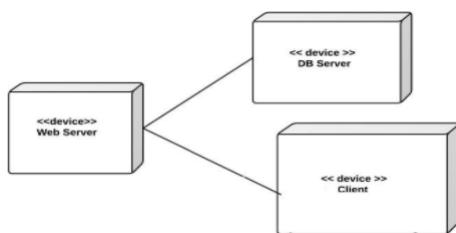
## Global analysis

### Introduction

On the aim to present you an over view on the project, we went through a global analyze. Which will be presented in the next chapter.

### 3.1 Global architecture

The software design stage comes after the requirements analysis and specification, it is a process of problem solving and planning for a software solution and it is a key step in the software life cycle.



**Figure 3.1:** Global architecture

## 3.2 Use case

The Use cases gives a clear view about the sequence of functions of our system. We did not give all the use cases specifications as many of them are either obviously similar or easily comprehensive. Now we have to concentrate upon the design of our system using UML (Unified Modelling Language)

### 3.2.1 Actor identification

Our platform is used by different kind of actors, this is where we define them:

Administrator	He is the one responsible for managing users accounts and their activities, generate reports, and managing documents for the researchers.
Internaut	-Manage an action. -Search an action, user's profile, documents, videos. - Clame. -Contact IRMC administrator

**Table 3.1:** Actors Identification

### 3.2.2 Use case analyze

In this section, we analyze one of our use cases and give detailed description. After this part, we present our global use case diagram with the most important functionalists.

**Table 3.2:** Description of consulting and add a document action

Use Case Name	Consult a document
Participating Actors	Internaut
Entry Conditions	Internaut must have the privilege to consult the specified document The Action is completed
Flow Of Events	1-consult documents page 2- Choose the "Research Action" section 3-Search
Alternative Flows	
Exit Conditions	

Use Case Name	Add a document
Participating Actors	Admin
Entry Conditions	Admin must be connected
Flow Of Events	1-consult Home page 2- Choose the "Add Action" section 3-Fill out an action form
Alternative Flows	Error message: Missing mandatory information
Exit Conditions	Message of confirmation: The action was accepted and created.

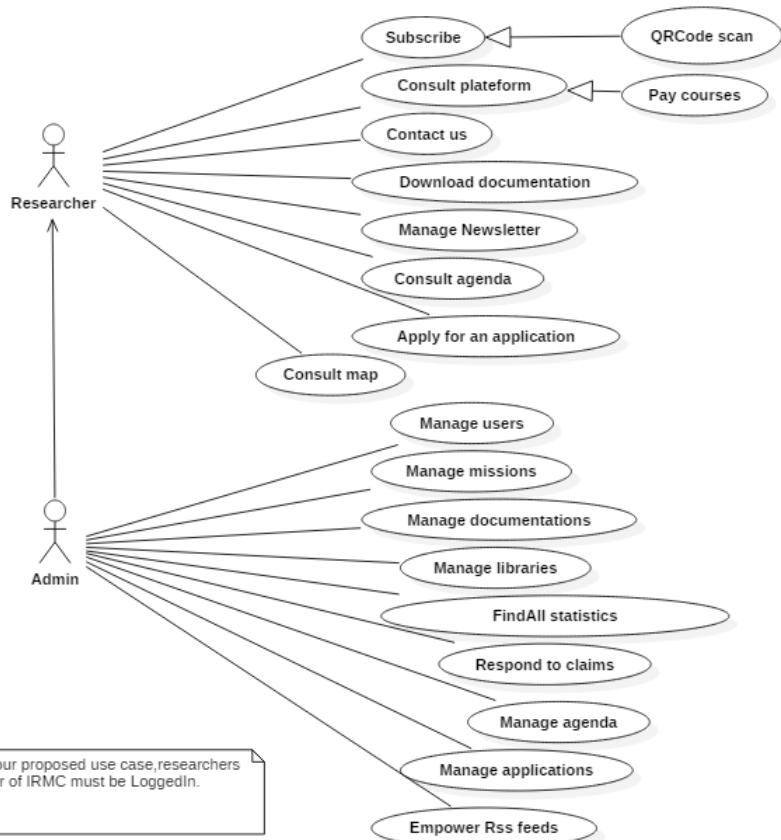
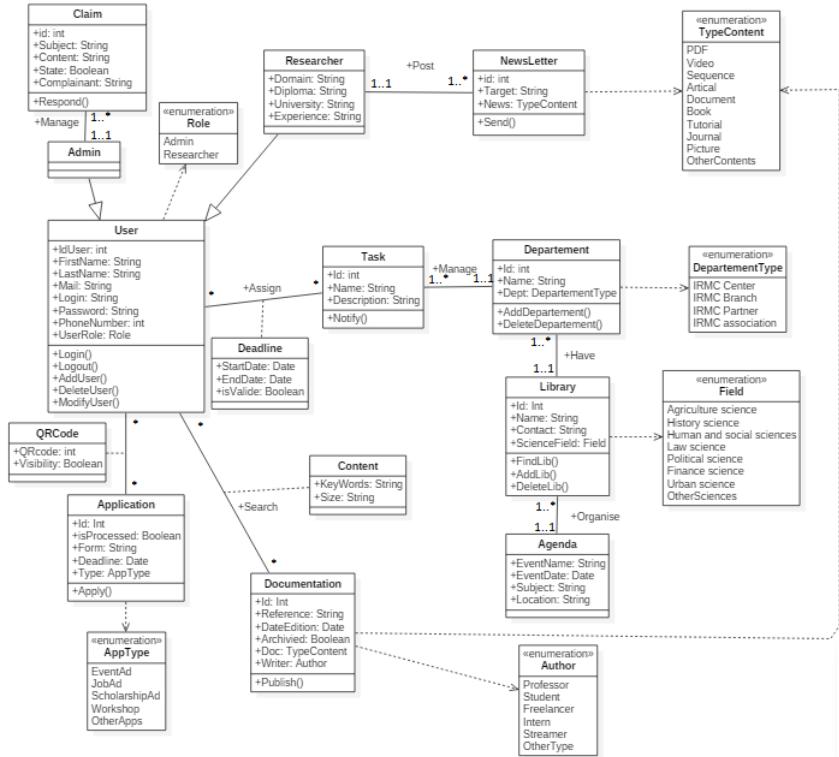


Figure 3.2: Use Case diagram

### 3.3 Class diagram

We used the class diagram to describe the structure of the system by showing the system's classes, their attributes, methods, and the relationships among objects.



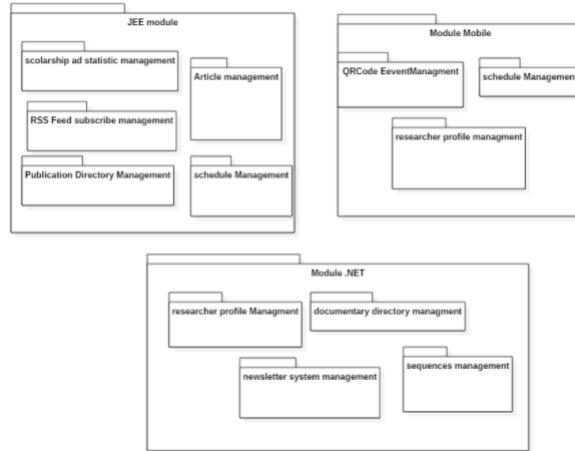
**Figure 3.3:** Class diagram

### 3.4 Functional Architecture

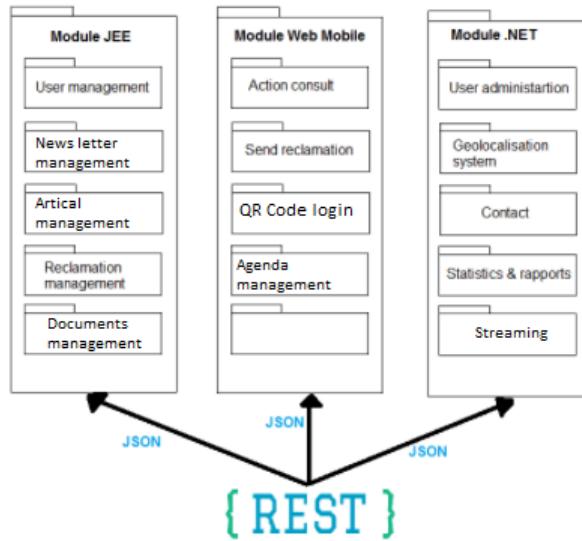
UML modeling helped us to plan our program before the programming takes place. It's used as a standard, a widely understood and well known. This made it easy for a new programmer to step into our project and be productive from day one. Now after explaining the aspects of our application.

#### 3.4.1 Logical Architecture

We define in this step the software logical architecture which is the n-tiers architecture.



**Figure 3.4:** Global functional Architecture



**Figure 3.5:** Functional Architecture

### 3.4.2 The N-Tiers architecture

The n-tiers architecture (distributed architecture, or multi-tier architecture) is a client-server architecture in which presentation, application processing, and database management layers are separated.

#### 3.4.2.1 Presentation Layer

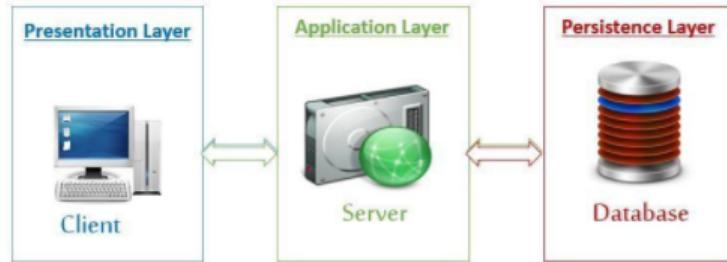
It puts out the results to the browser/client tier and all other tiers in the network. (In simple terms it is a layer which users can directly access to such as a web page, or an operating system GUI) In our application, we'll be using JSF and .NET for the web presentation layer.

### 3.4.2.2 Application Layer

It contains treatments representing the business rules and it controls the application's functionality by performing detailed processing. We'll be developing our services with JEE and SOA to maintain a lifetime code by consuming our already developed functionality.

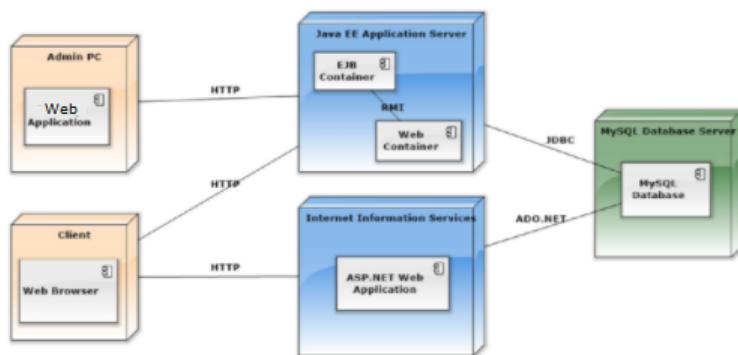
### 3.4.2.3 Persistence Layer

This layer is responsible for mapping logic objects and data components. It's the only layer that is linked with our database MySQL.



**Figure 3.6:** N-Tiers architecture

### 3.4.3 Physical architecture



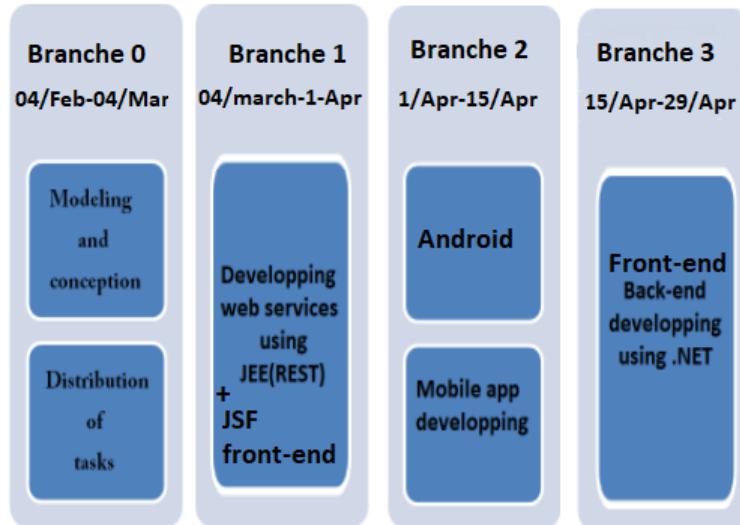
**Figure 3.7:** Deployment diagram

We had to think of an architecture that can be communicated data between these platforms. For this, we used the solution that is of use, to

communicate that all the platforms to a single database present on a server.

### 3.4.4 Road map

Our milestone of 2-TUP branches during the hole process:



**Figure 3.8:** Road map

## 3.5 Conclusion

With this project we are going to develop a platform for I.R.M.C and its prime objective is not only to help us master the new technologies such as JEE , .NET , Android and SOA consuming services but more importantly to sensitize the people about a new lifetime advanced research engine technology.

# 4

## Technical branch

### Introduction

On the aim to implement an interactive and easy to use platform. We had to collaborate the development tools in order to make that happen. We needed to set an efficient project architecture that will help us build a trade application.

### 4.1 Software tools

In this part, we will move to the process of the project implementation by shedding lights on the various mastered tools, libraries and platform.

#### 4.1.1 Platform

##### 4.1.1.1 Java Enterprise Edition Java EE

The Enterprise Java platform (Java EE) is a set of coordinated specifications and practices that enable solutions for the development, deployment, and management of multi-tiers centralized applications on a server.

Ejb 3.2 : is a server-side software component architecture for the Java EE development

platform.

JAP 2.1 : a Java application programming interface specification that describes the management of relational data in applications using Java Platform, Standard Edition and Java Platform, Enterprise Edition.

MVC 2 : a software architecture model(allows the use of multiple controllers)



**Figure 4.1:** The Enterprise Java platform (Java EE)

#### 4.1.1.2 JBoss Developer Studio 9

JBoss Developer Studio is a set of eclipse-based development tools that are pre-configured for JBoss Enterprise Middle-ware Platforms and Red Hat Enterprise Linux. JBoss is written in Java and as such is cross-platform:

usable on any operating system that supports Java.



**Figure 4.2:** Jboss developer studio 9

### 4.1.2 Programming tools

#### 4.1.2.1 Java Development Kit JDK 1.8

It represents the environment in which the Java code is compiled to be turned into byte code so that the Java virtual machine (JVM) can interpret. The primary components of the JDK are a selection of programming tools.

JDK are a selection of programming tools, including:

Javac: the compiler, which converts the source code into class file.

Jar – Archiver: puts as a single package, all files class in a JAR file.

Javadoc: the documentation generator that automatically generates documentation from source code comments.

Jdb: the debugger.



**Figure 4.3:** Java Development Kit

### 4.1.3 Application compiler

#### 4.1.3.1 Java Runtime Environment JRE

Java Runtime Environment is a Java virtual machine. It is a part at runtime Java SDK but without the development tools such as compilers and debuggers.



**Figure 4.4:** Java Runtime Environment

#### 4.1.3.2 Microsoft visual studio



**Figure 4.5:** Microsoft visual studio logo

Microsoft Visual Studio is an integrated development environment from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps.

#### 4.1.3.3 Microsoft visual studio



**Figure 4.6:** Android studio logo

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

#### 4.1.4 Database deployment

##### 4.1.4.1 MySQL server

A basic relational database server that stores information in tables, in the form of data subjects by groups. The tables are linked by relationships



**Figure 4.7:** MySQL server

##### 4.1.4.2 MySQL connector

MySQL Connector is a driver for connecting to a MySQL database server through the ODBC application programming interface which is the standard way of connecting to a database. Users can connect from common applications and programming environments such as Microsoft Access or Excel.



**Figure 4.8:** MySQL connector

#### 4.1.4.3 MySQL Workbench

MySQL Workbench is a unified visual tool for database architects, developers, and DBAs. MySQL Workbench provides data modeling, SQL development, and comprehensive administration tools for server configuration, user administration and backups.



**Figure 4.9:** MySQL Workbench

#### 4.1.5 Java platform implementation

##### 4.1.5.1 Wildfly x.9 server

WildFly, formerly known as JavaBeans open source software application server that implements the Java platform Enterprise Edition.



**Figure 4.10:** Wildfly x.9 server

#### 4.1.6 Web services' consumption

##### 4.1.6.1 Client REST

This application is deprecated by GOOGLE. The only REST client that makes connection directly on socket giving you full control over the connection and request/response headers.



**Figure 4.11:** Client REST

### 4.1.7 Application design and analysis

#### 4.1.7.1 StartUML

For the UML modeling open source software. Via this platform, it is possible to design our diagrams such as creating class diagram, use case, package, activity or sequences.



**Figure 4.12:** StartUML application designer

### 4.1.8 Collaborative development environment

#### 4.1.8.1 GitLab

GitLab is a Web-based Git repository hosting service, which offers all of the distributed revision control and source code management (SCM) functionality of Git as well as adding its own features.



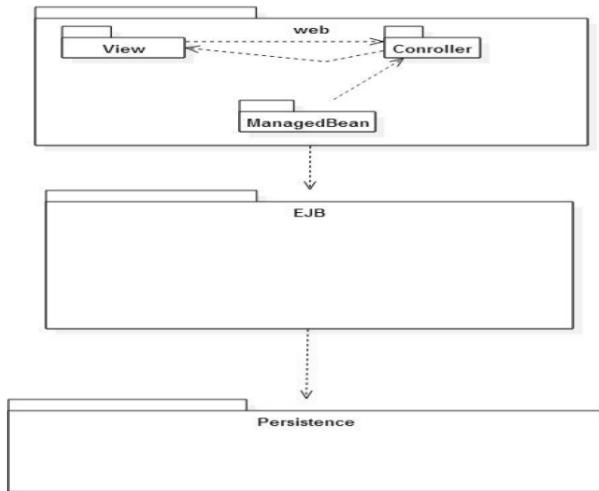
**Figure 4.13:** Collaborative development environment

## 4.2 Logical architecture

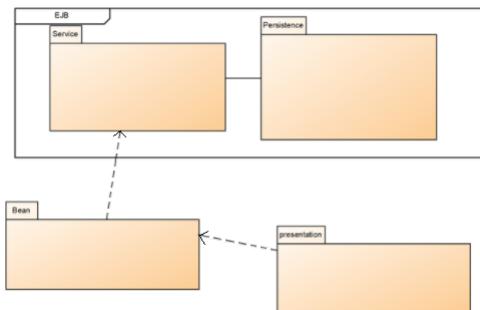
This diagram represents the Logical architecture of our application which is composed with 4 elements, our EJB which communicates with the entity and the Managed bean and the managed bean communicates with the EJB and our view (JSF).

## 4.3 Physical architecture

This diagram represents the physical architecture of our application composed with our server MySQL and 2 container, web and EJB containers.



**Figure 4.14:** Java EE Package diagram



**Figure 4.15:** Detailed Java EE Package diagram

## 4.4 Preliminary and detailed conception

### 4.4.1 JAVA EE functional requirement

-

### 4.4.2 Object sequence and object class diagrams

#### 4.4.2.1 Object sequence diagram "Authentication"

#### 4.4.2.2 Object sequence diagram "Add SessionUser"

-This diagram shows how to manage our agent inside the IRMC :

-Login()

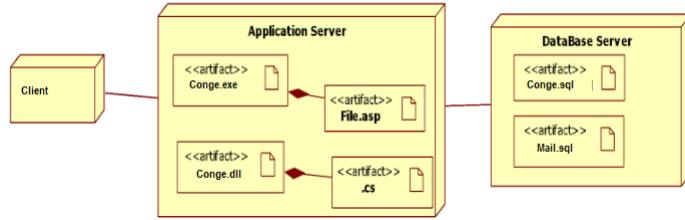


Figure 4.16: Global java EE deployment diagram

Module	Sub-Module	Functionality code	Functionality	Responsible
JEE	User management	JEE 01 JEE 02 JEE 03 JEE 04 JEE 05 JEE 06 JEE 07	Sign-in (Recaptcha) Sign-out Add user Modify user Search for user User session delete (token) View users	Zohra Dhaouadi
	Application management	JEE 08 JEE 09 JEE 10 JEE 11 JEE 12 JEE 13 JEE 14	Upload application Add application Delete application View application Search application Comment application Delete comment	
	Agent tasks management	JEE 15 JEE 16 JEE 17 JEE 17 JEE 18	Add task Delete task Modify task Search task View tasks	
	About us	JEE 19 JEE 20	Streaming video Upload video	
	Newsletter /Contact us	JEE 21 JEE 22 <b>JEE 22.1</b>	Send mail (API mail) InBox (Internal mail box) <b>Send document by internal e-mail</b>	
	Documentation center management	JEE 23 JEE 24 JEE 25 JEE 26 JEE 27	Search mail Add documentation center Delete documentation center View documentation center Update documentation center	Nader KASRI
	Events management	JEE 28 JEE 29 JEE 30 JEE 31	Add event Delete event View event Update event	Khaled laboubi
	Documents management	JEE 32 JEE 33 JEE 34 JEE 35	Add document Delete document View document Update document	Sahar Mastoura

Figure 4.17: JAVA EE functional requirement

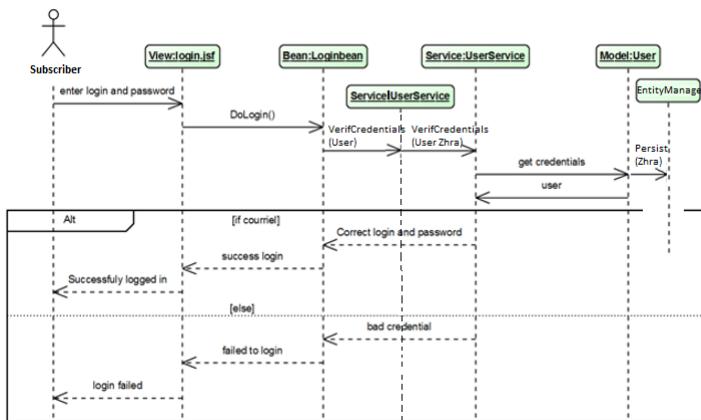
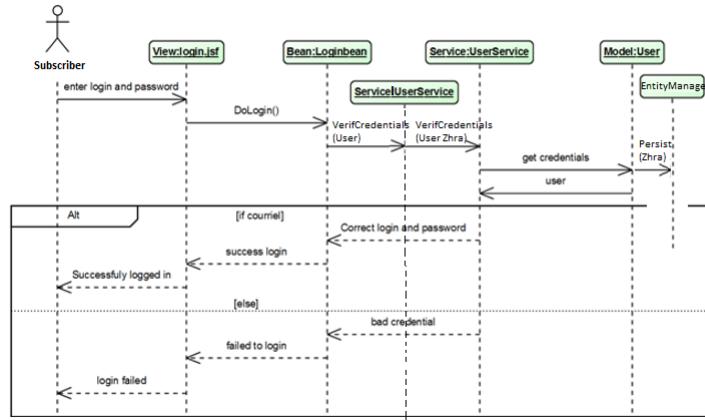


Figure 4.18: Global java EE deployment diagram

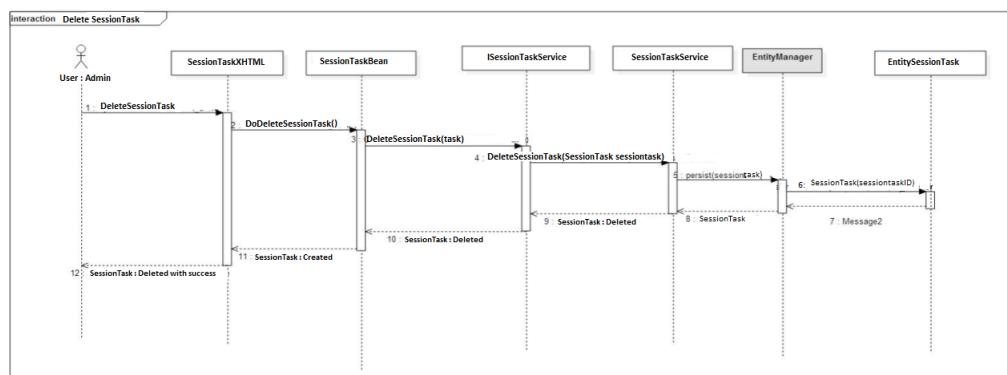


**Figure 4.19:** Object sequence diagram "Add SessionUser"

-Add user() and persist data

-Session created

#### 4.4.2.3 Object sequence diagram "Delete sessionTask"



**Figure 4.20:** Object sequence diagram "Delete sessionTask"

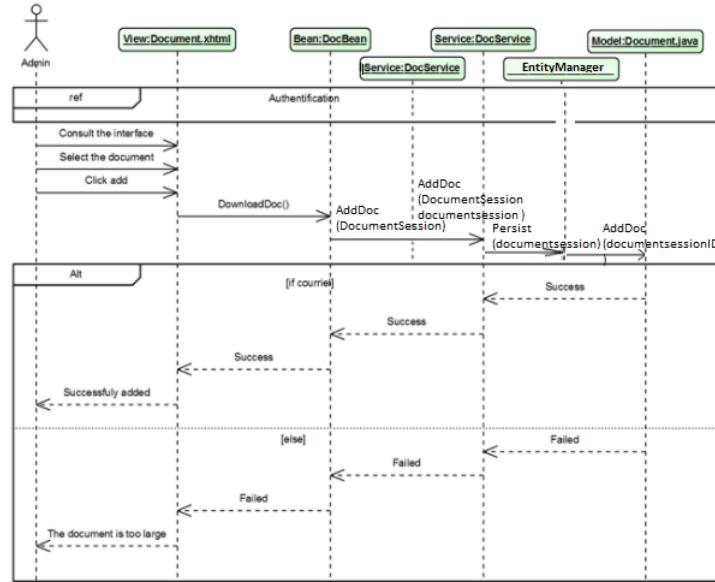
#### 4.4.2.4 Object sequence diagram "Upload applications and documents"

-In this diagram we try to resume the actions followed to,

upload any PDF document or application for the candidate.

We'll see in the next diagram the main functionality we can affect on our uploaded files:

-Just like giving feedback on the applications by comments or sending documents by an



**Figure 4.21:** Object sequence diagram "Upload applications and documents"

internal e-mail.

#### 4.4.2.5 Object sequence diagram "Send mails"

### 4.4.3 JEE Analysis class diagram

-Class diagrams describe the objects of the system:

Their inter-relationships,

The main persistence operations

Private attributes.

Public functionality

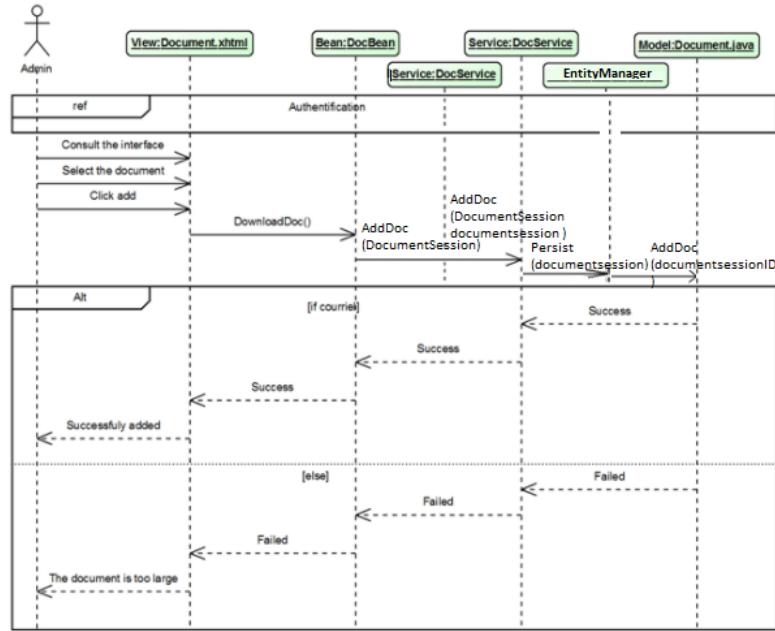
Cardinality

-To resume our diagram strength:

We tried to optimize our class diagram into 10 tables of objects.

We had two roles for the user and three files state as enumeration tables

We had to add some joined tables to optimise the attributes in the objects class.



**Figure 4.22:** Object sequence diagram "Send mails"

## 4.5 Design Class Diagram (JEE)

The design class diagram shows the most important transactions for the code of the application:

- We can see the Entity-Manager class that persist,merge and create-Queries for data treatments.
- The interfaces classes
- The service classes
- The Managed-Bean classes
- Finally,The main entity class

## 4.6 Design class diagram decomposed into packages (JEE)

The design class diagram aim to decompose the class diagram into detailed packages:

- Web package
- EJB package
- Persistence package

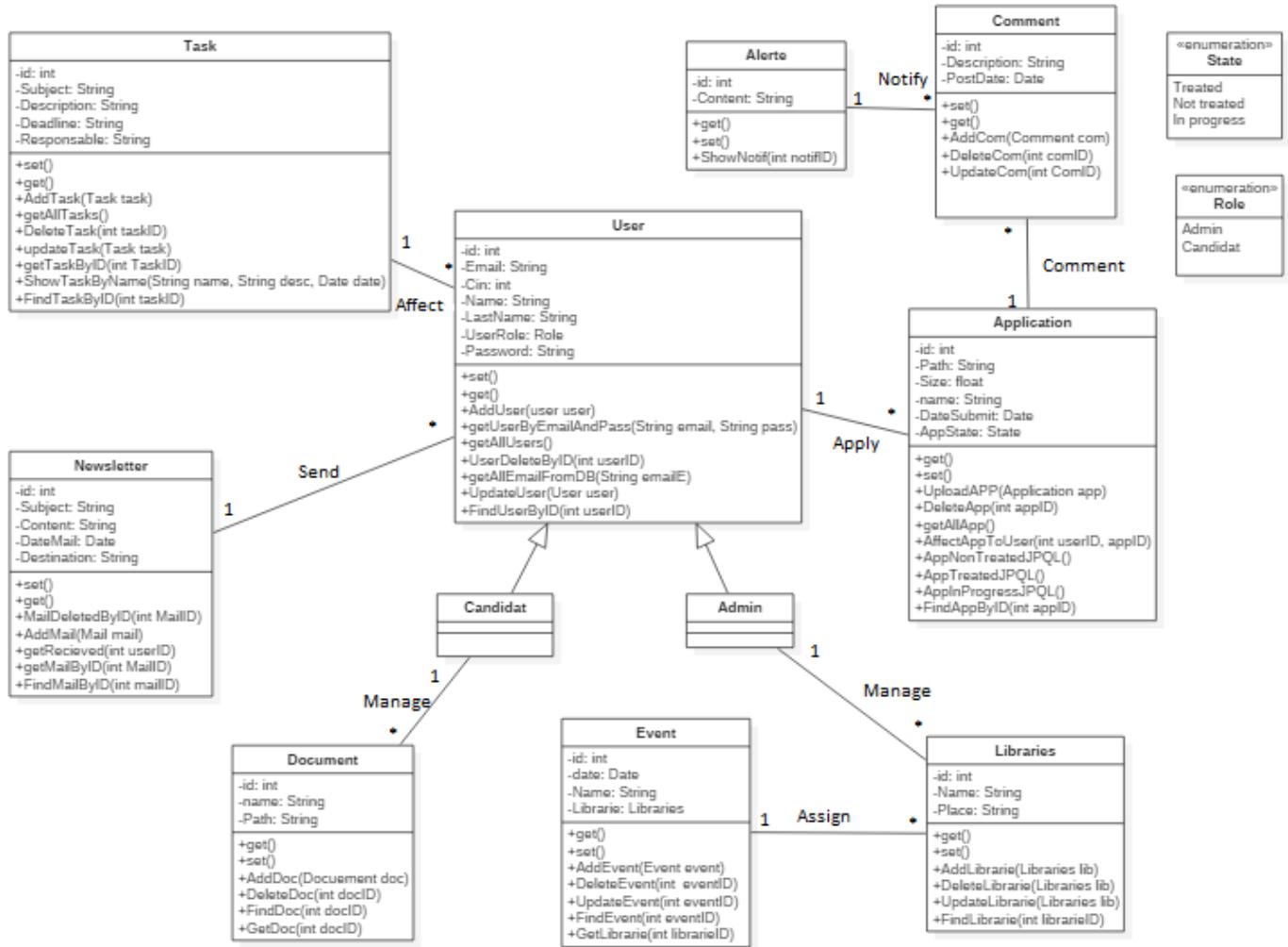


Figure 4.23: JEE Analysis class diagram

#### 4.6.1 Web package (JEE)

-The web package that present our M.V.C model:

-View Controller

-Managed-Bean

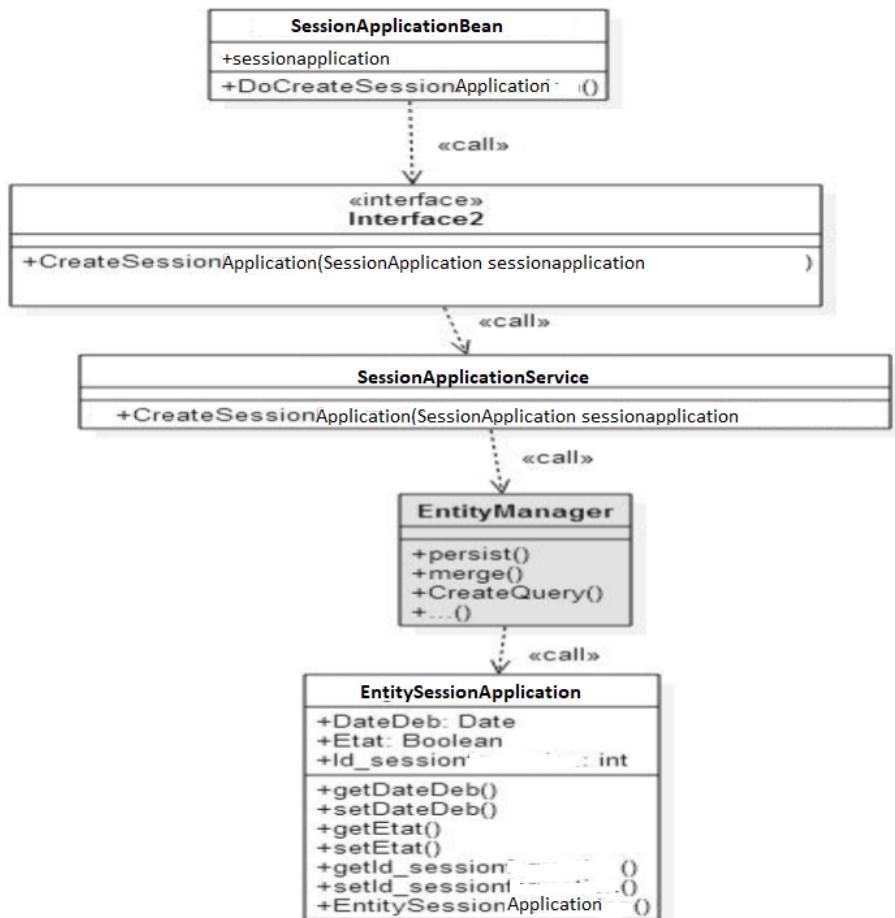


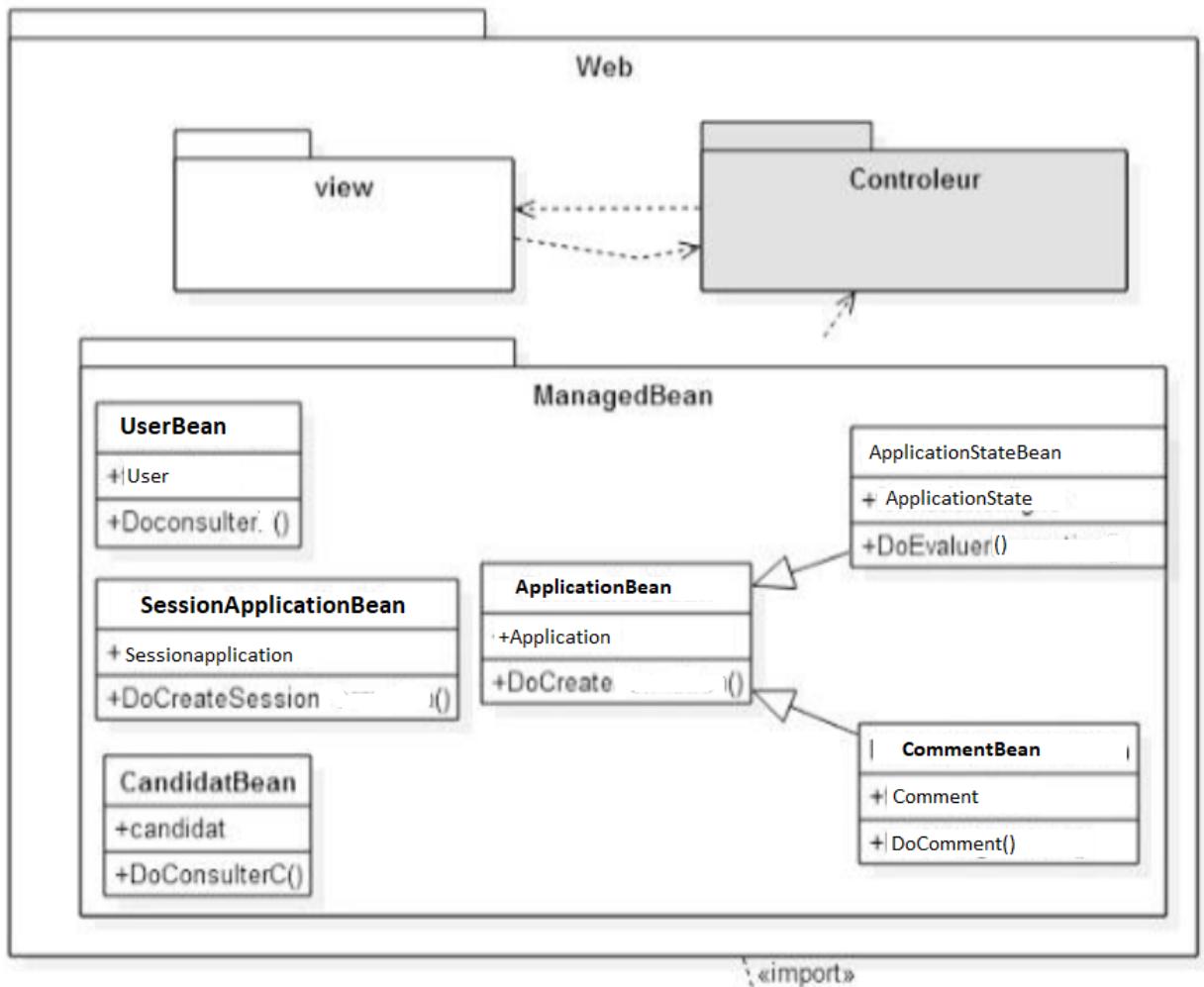
Figure 4.24: Design Class Diagram (JEE)

#### 4.6.2 EJB package (JEE)

- The EJB package that present our:
- Entities(Objects)
- Services
- I-Services(Interfaces)

It models our methods and their relation between our packages

-  
-  
-



**Figure 4.25:** Web package (JEE)

#### 4.6.3 Package Persistance (JEE)

The persistence package shows our entities and its deployment on the data base:

- One class that collects the persistence methods, The entity-manager.
  - The deployment of the attributes data

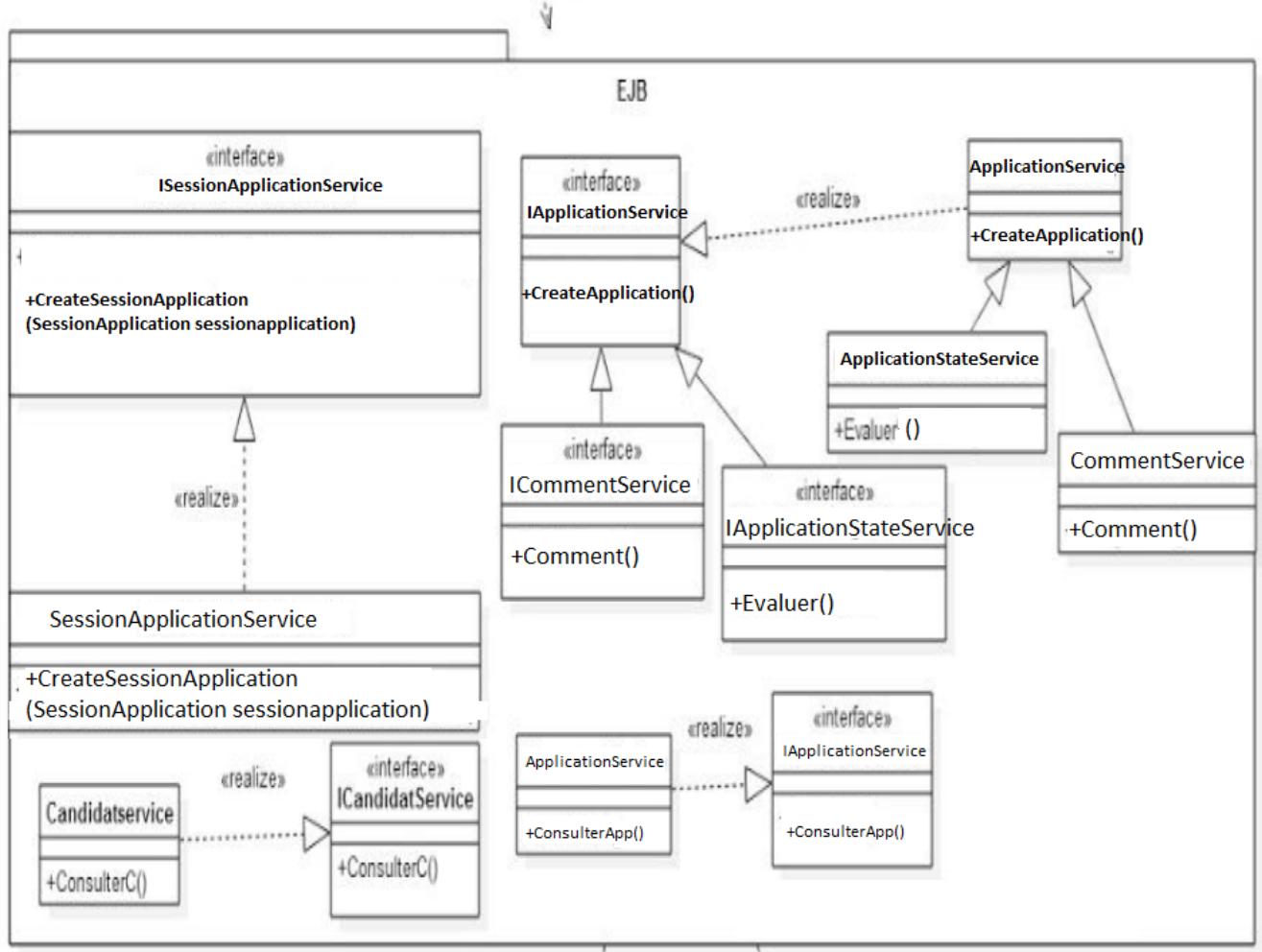


Figure 4.26: EJB package (JEE)

#### 4.6.4 Component Diagram (JEE)

This is the components diagrams which contains all the components that defines our EJB, Managed-Bean and our JSF, XHTML files.

- We can see the most three important layers of our application:
- Persistence layer
- EJB layer
- War layer, that contains:
  - Presentation layer with the JSF files.
  - The Managed-Bean

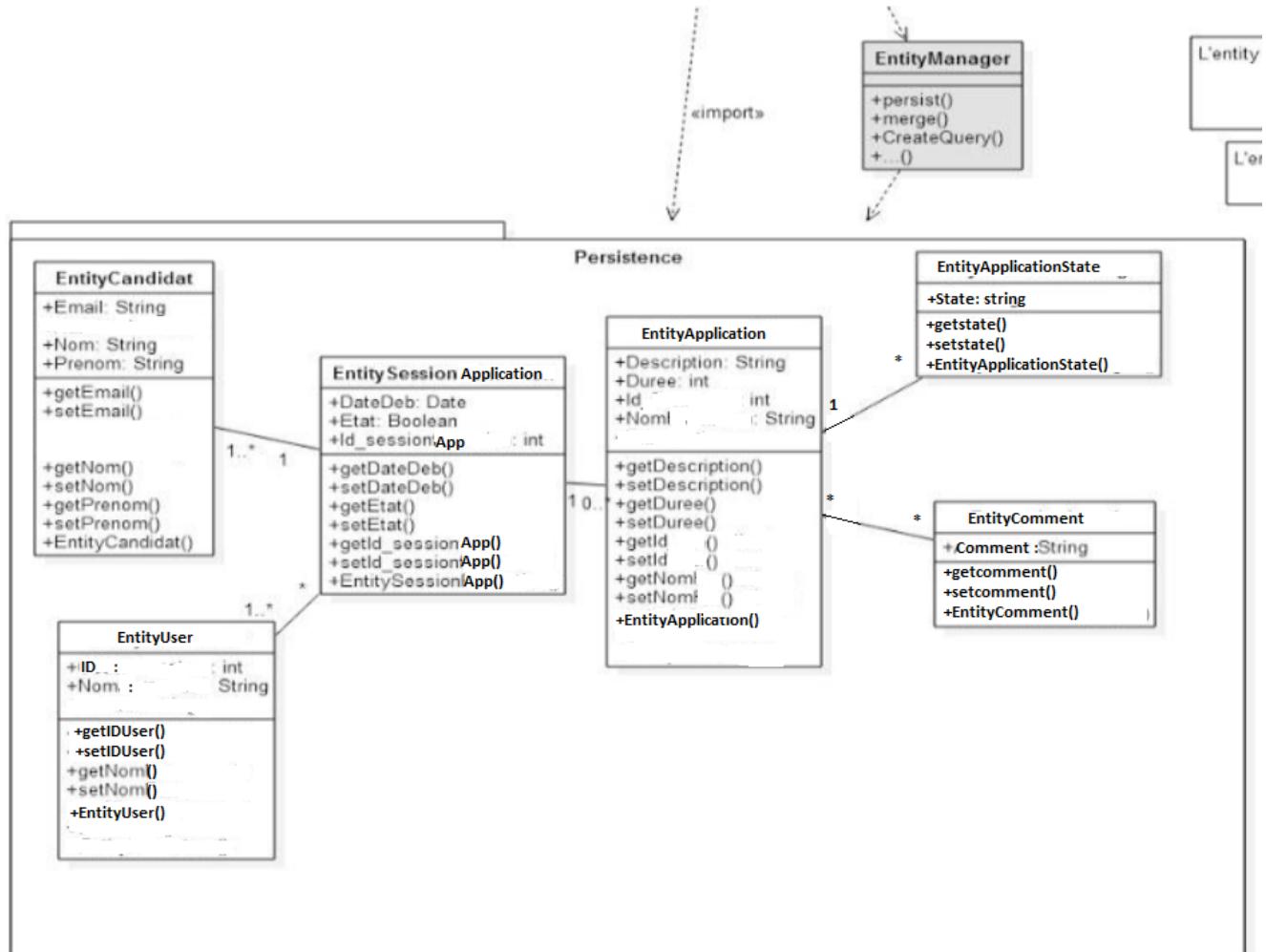


Figure 4.27: Package Persistence (JEE)

#### 4.6.5 Deployment Diagram (JEE)

-For the deployment diagram we can see three sides:

The client side for the presentation.

The ear side for the .jar and the .war packages.

The data base that deployed the .jar

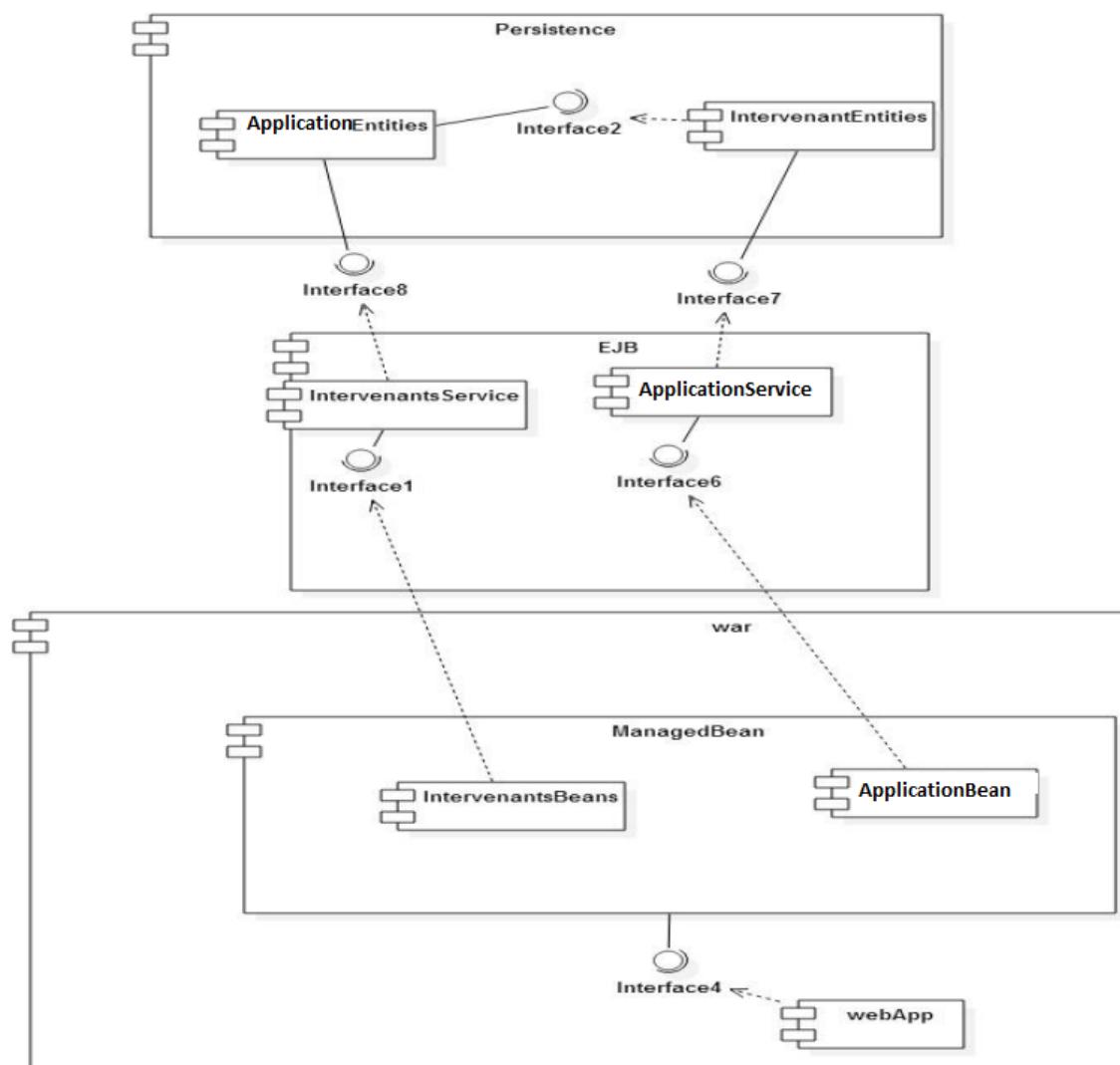


Figure 4.28: Component Diagram (JEE)

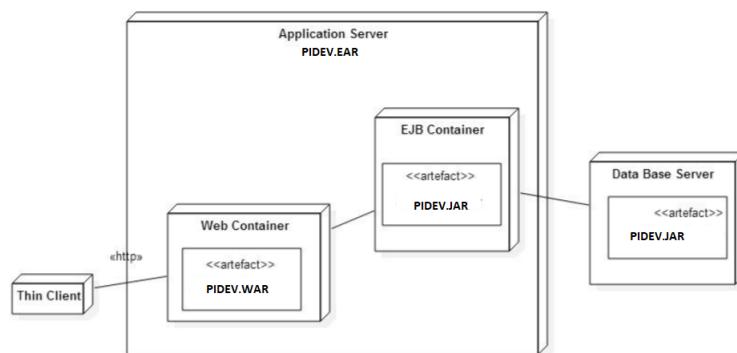


Figure 4.29: Deployment Diagram (JEE)

#### 4.6.6 Detailed deployment diagram(JEE)

This is the deployment diagram that describes our components diagram. In this module, we designed the conceptual diagrams which helped us understand how our application is supposed to work. Knowing how our classes interact with each other, we were able to develop the java ee part of our project management tools.

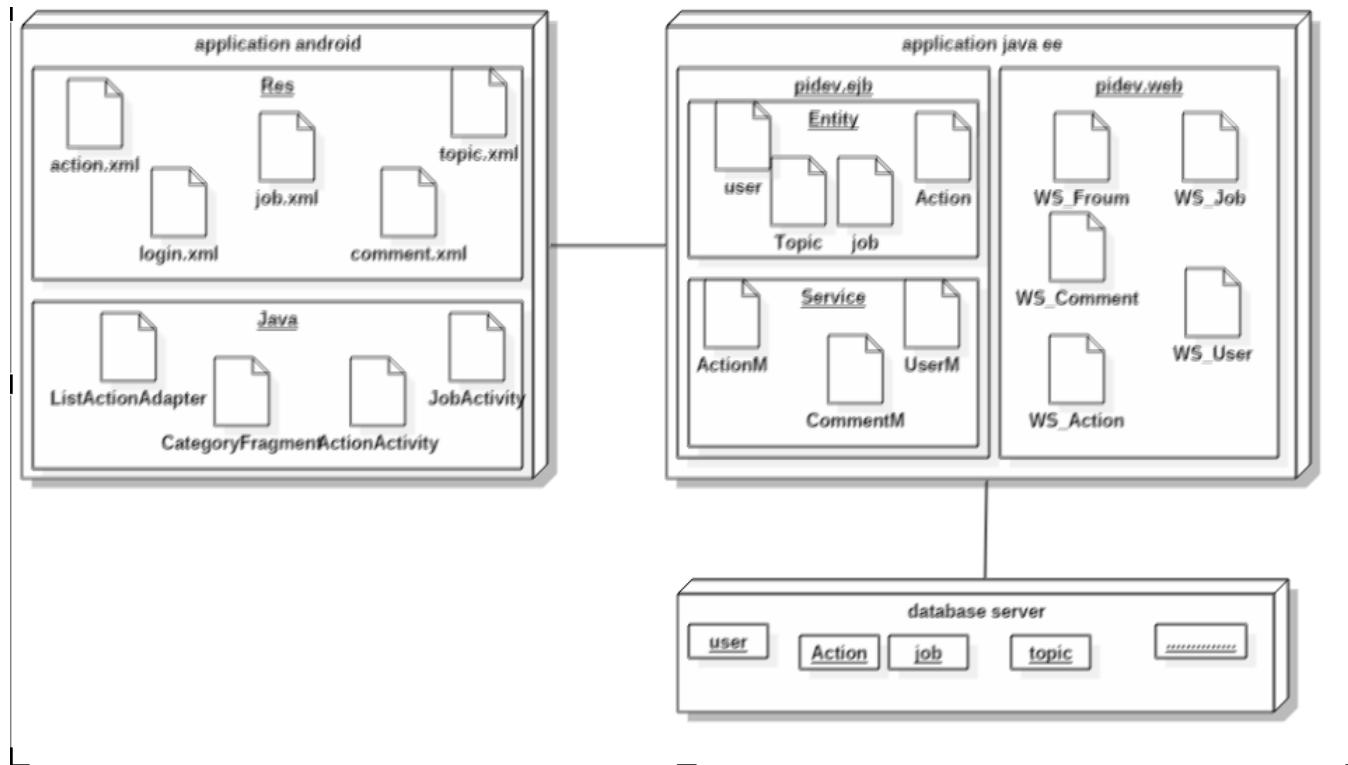


Figure 4.30: Deployment diagram(JEE)

## 4.7 Implementation and test

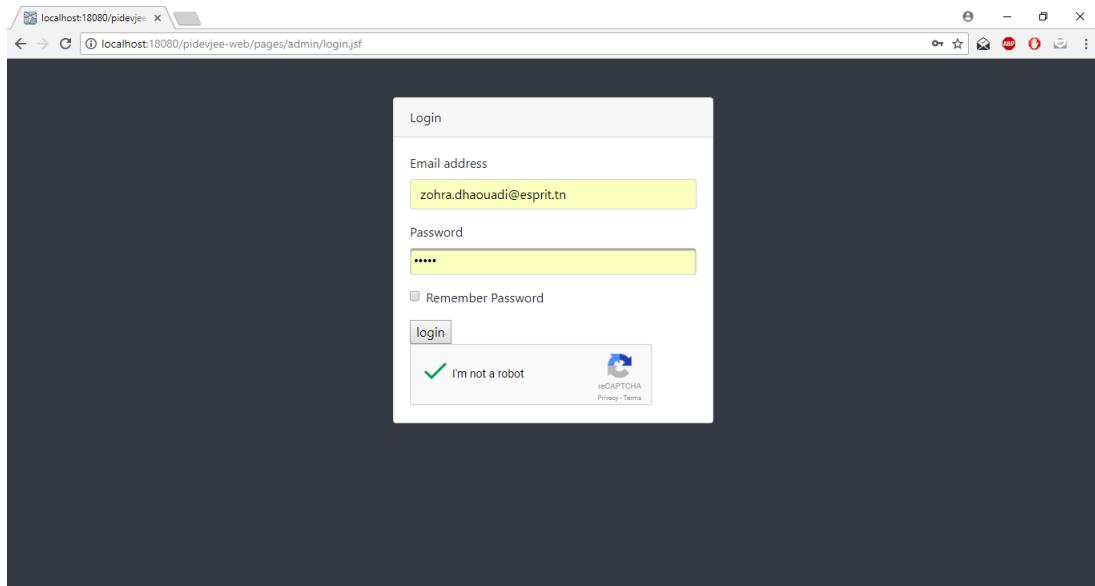
### 4.7.1 Application interface

#### 4.7.1.1 Sign in/Sign out

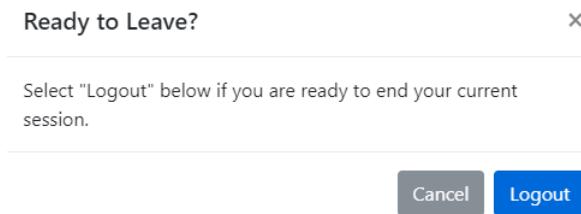
-This interfaces present our:

-Template integration

-Login method



**Figure 4.31:** Sign in



**Figure 4.32:** Sign out

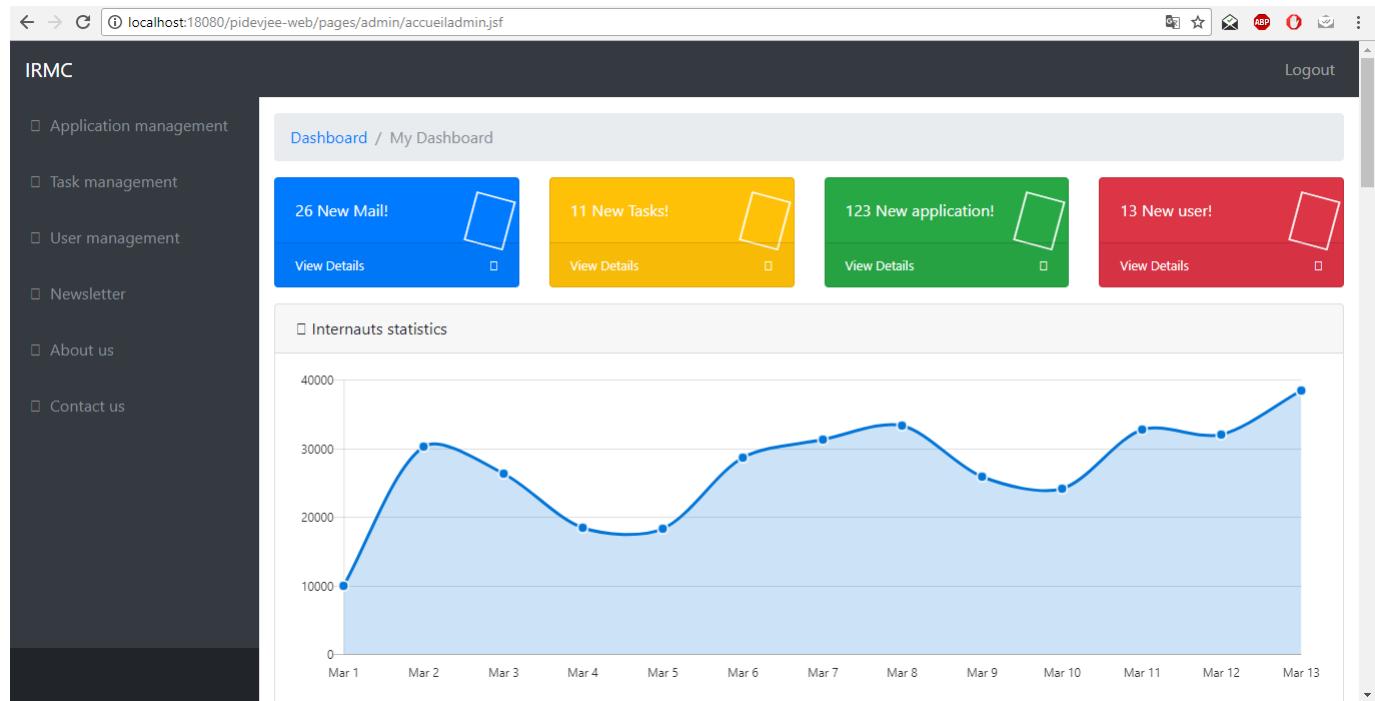
-Logout method

#### 4.7.2 Home page

- For the home page we show our most important functionnalities:
- It represent the statistic interface as a first view
- User management
- Application management
- Task management
- Newsletter:

Internal mail box

## Chapter 4. Technical branch



**Figure 4.33:** Home page

-Contact us:

External mail box

-About us:

Streaming videos

### 4.7.3 User management

This is where we find the user C.R.U.D methods:

-We can add an agent

-Find an agent

-Modify an agent data

-Show all our IRMC agents

## Chapter 4. Technical branch

The screenshot shows a web-based administration interface for the IRMC system. On the left, a sidebar lists navigation options: Application management, Users management, Task management, and Newsletter. The main content area has a header "Welcome zahra dhaouadi". Below this is a table showing four user entries:

login	password	cin	Nom	Prenom	Role	Sexe	Age	Salaire	actif	Action
le.dh@esprit.tn	hhh	1152364	hhhhhhhhhh	hhhhhhhhhh	administrateur	Femme	20	500.0	oui	<button>Modifier</button>
Medhaouadi@esprit.tn	blah	1152364	zaho	Dh	administrateur	Femme	0	2000.0	oui	<button>Modifier</button>
zhra.dhaouadi@esprit.tn	zahra	26612978	dh	zhra	administrateur	Femme	0	500.5	oui	<button>Modifier</button>
zohra.dhaouadi@esprit.tn	zahra	11375763	zahra	dhaouadi	administrateur	Femme	23	500.0	oui	<button>Modifier</button>

Below the table, it says "Showing 1 to 4 of 4 entries". To the right, there are buttons for "Previous" (disabled), "1", and "Next".

At the bottom, a modal window titled "Add user" is open, containing fields for login (email), password, cin, nom, Prenom, Salaire, Age, Role (with a dropdown menu "Selectionner le role"), Sexe (with a dropdown menu "Selectionner le sexe"), and Actif / inactif (checkbox). There are "Add" and "Update" buttons at the bottom of the modal.

At the very bottom of the page, it says "Copyright © Your Website 2018".

**Figure 4.34:** User management

### 4.7.4 Task management

This is where we find the users' tasks C.R.U.D methods:

- We can affect a task for a user
- Find users' tasks
- Modify any affected task
- Delete users' tasks
- Show all our IRMC agents tasks

### 4.7.5 Application management

This is where we find the users' applications C.R.U.D methods:

- A user can apply for an application
- the admin can find all the applications
- Comment any application
- Delete users' application

## Chapter 4. Technical branch

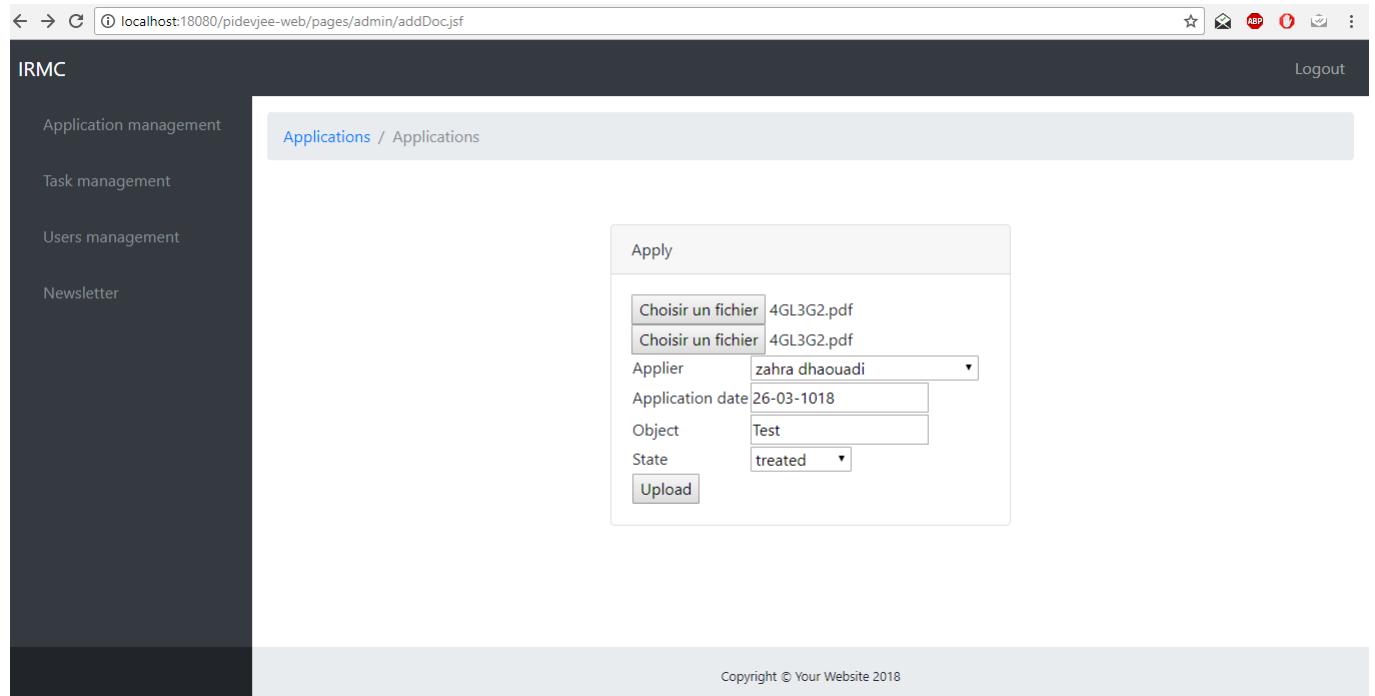
The screenshot shows a web application interface for task management. The left sidebar has links for Application management, Tasks management, User management, and Newsletter. The main content area shows a 'Tasks list' table with columns: Nom, Description, deadline, Responsable, Etat, and Action. Two entries are listed: 'zaza' with 'le' description, deadline '2018-04-21', and state 'En cours'. The second entry is identical. Action buttons 'Supprimer' and 'Modifier' are available for each row. Below the table, it says 'Showing 1 to 2 of 2 entries' and has 'Previous' and 'Next' buttons. A modal window titled 'Add' is open, containing fields for nom, description, deadline, and Responsable, along with 'Add' and 'Update' buttons. At the bottom, it says 'Updated yesterday at 11:59 PM' and 'Copyright © Your Website 2018'.

**Figure 4.35:** Task management

- Upload applications
- Give it a state as:
  - Treated,
  - Not treated,
  - In a progress.

### 4.7.5.1 Applications list

- This is the application list presentation:
  - We focus on the delete and comment button.



**Figure 4.36:** Application management

### 4.7.5.2 Application comment

- This interface suppose to look like the ratting for the applications by:
- The most recommended
- The least recommended
- As we can delete an affected feedback by comments on the applications.

### 4.7.6 Newsletter

In this interface we had developed an internal mail box.

Where we can send documents by email

- We can see the upload method of files
- The send method of the email

Reference	Nom	Date	Responsable de documents	Etat	Action
1	blah		hhhhhhhhhh	treated	[Supprimer] [Commenter]
3	me	2018-03-12 01:00:00.0	zahra	treated	[Supprimer] [Commenter]
5	me	2018-03-12 01:00:00.0	zahra	treated	[Supprimer] [Commenter]
6	me	2018-03-12 01:00:00.0	zahra	treated	[Supprimer] [Commenter]
8	mehhh	2018-08-11 02:00:00.0	zahra	notTreated	[Supprimer] [Commenter]
9	zf	2018-03-22 01:00:00.0	dh	treated	[Supprimer] [Commenter]
11	tf	2018-03-22 01:00:00.0	dh	treated	[Supprimer] [Commenter]
12	vdr	2017-02-12 01:00:00.0	zahra	treated	[Supprimer] [Commenter]

**Figure 4.37:** Applications list

### 4.7.7 Inbox

-On this stage we aimed to develop an internal mail box where mails will be stored on a local folder called "dsi".

### 4.7.8 Contact us

-This is where we implement an integration for the JAVA mail API .

#### 4.7.8.1 Mail reception

### 4.7.9 About us/Streaming

-For the streaming video,we had developed a streaming service on a Centos , virtual machine, where we have integrate it on our site.

## Chapter 4. Technical branch

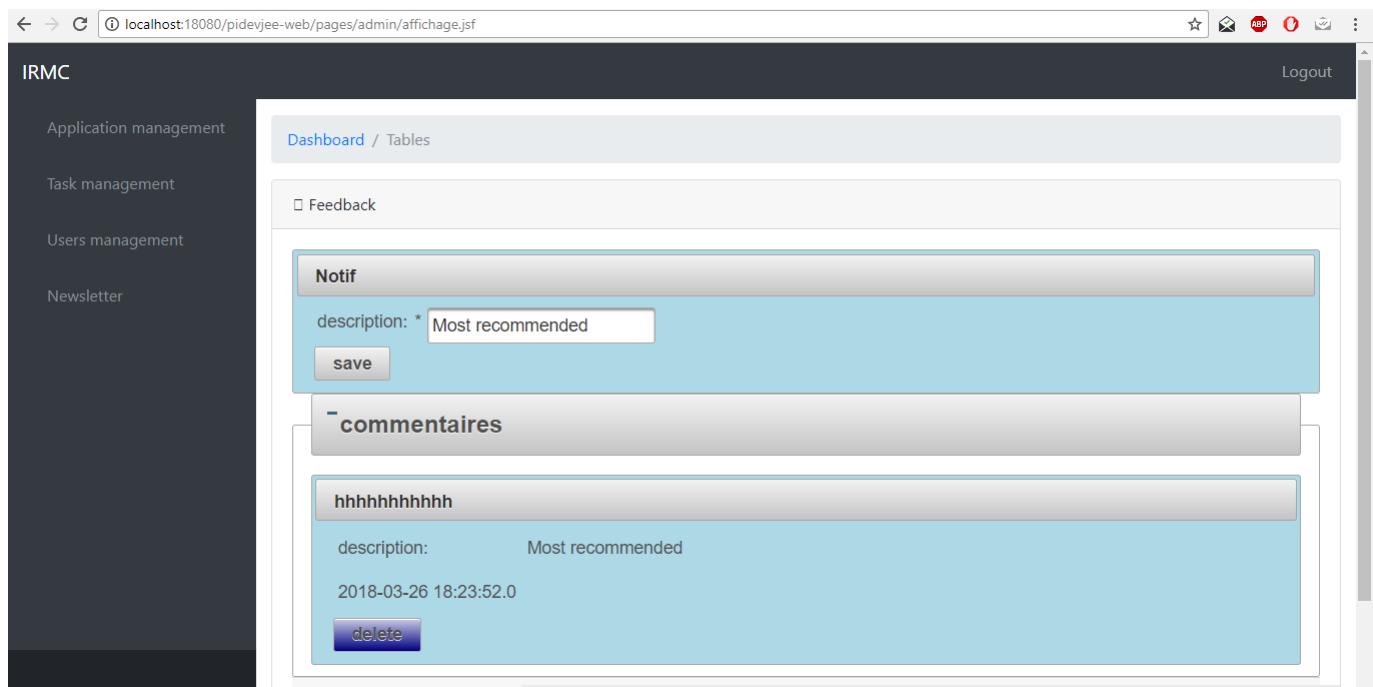


Figure 4.38: Application comment

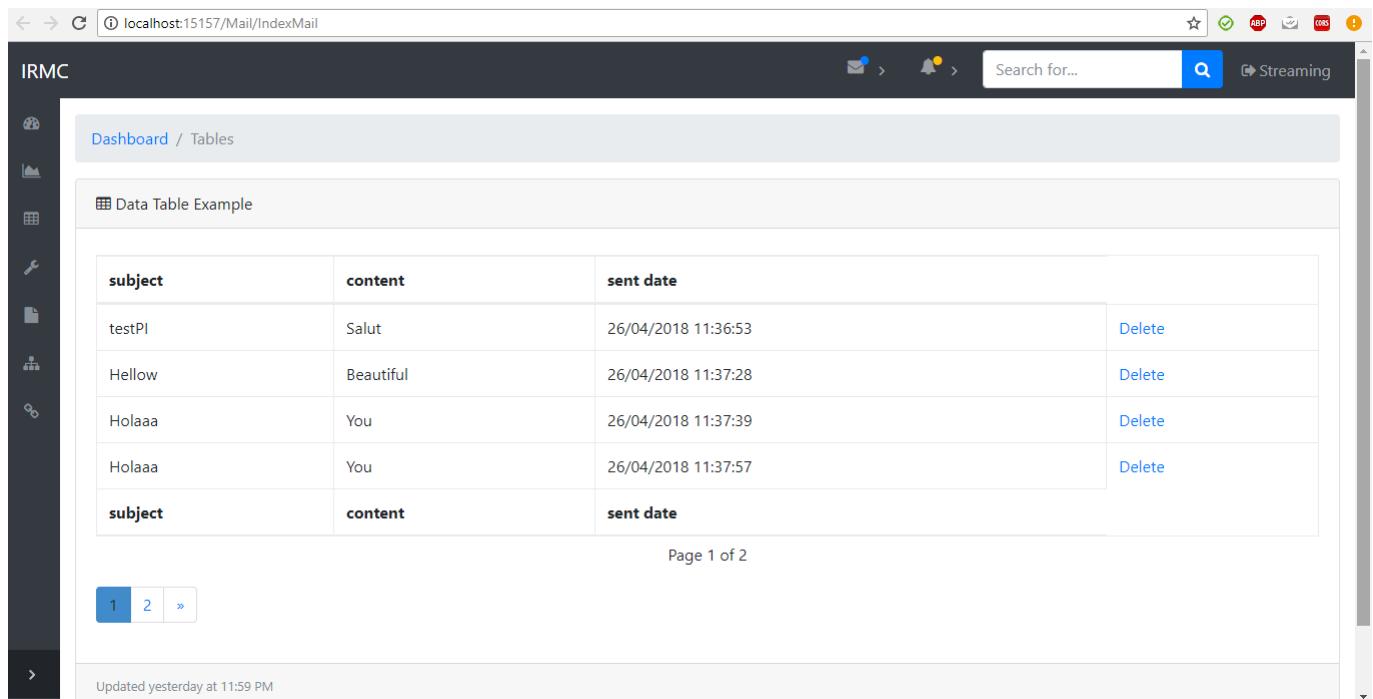
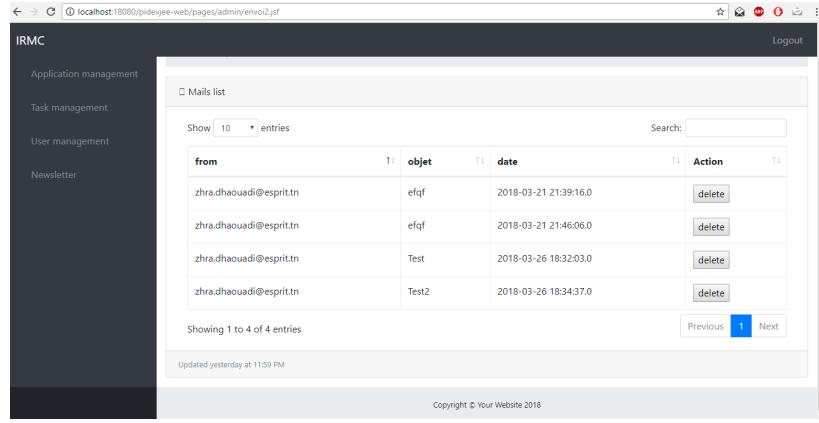


Figure 4.39: Newsletter



**Figure 4.40:** Inbox

## 4.8 Implementation tips and integration

For the implementation, we choose a HTML5 bootstrap template for the design phase and we started by deploying the DATABASE on the same server to manage the integration of our work .

For the integration, we used Gitlab as a technology to work in group, which is the most widely used modern version control system on a the worldwide.

Git has the functionality, performance, security and flexibility that most teams and individual developers needed.

We started by implementing our entities,generated our C.R.U.D then we integrated our template.

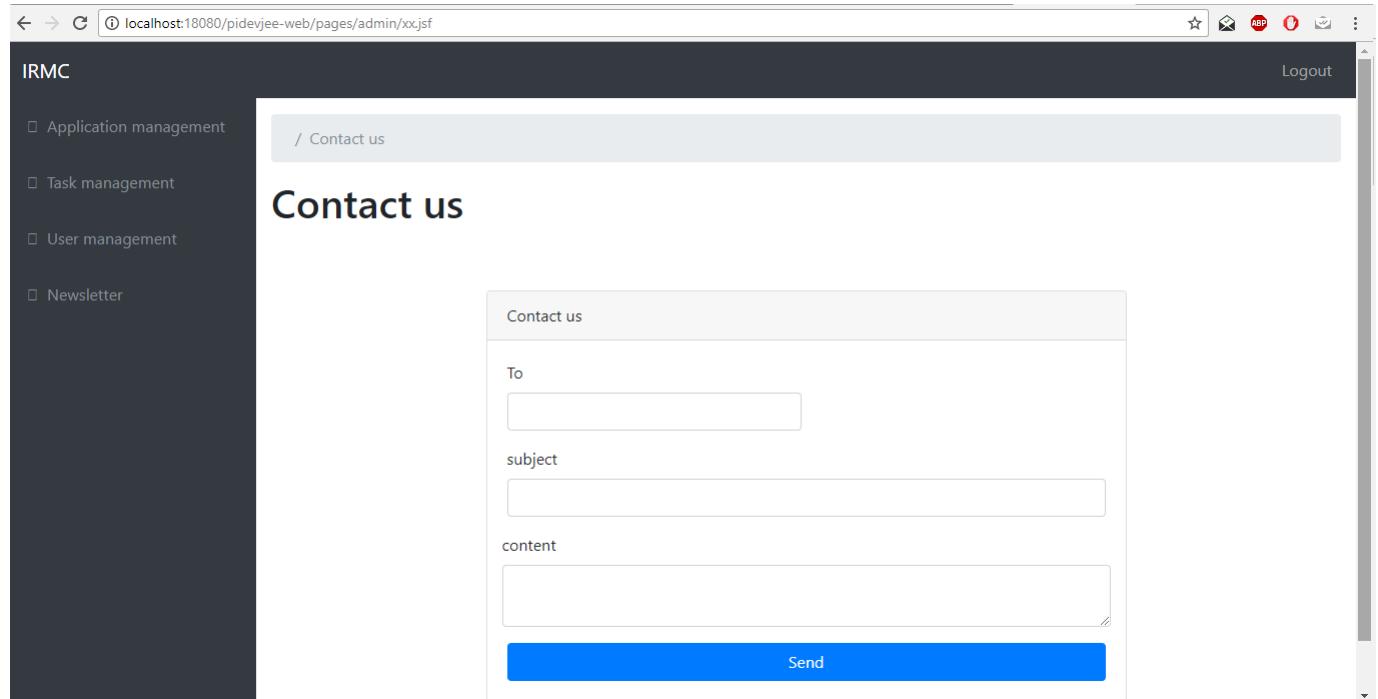
At the end we deployed the project on GIT ,so that everybody could start the development to ease the integration of the different project parts.

## 4.9 Application test

We used several types to check our work, Integration tests to validate the integration of the different modules between them and unit tests to validate the quality of the code and the performance.

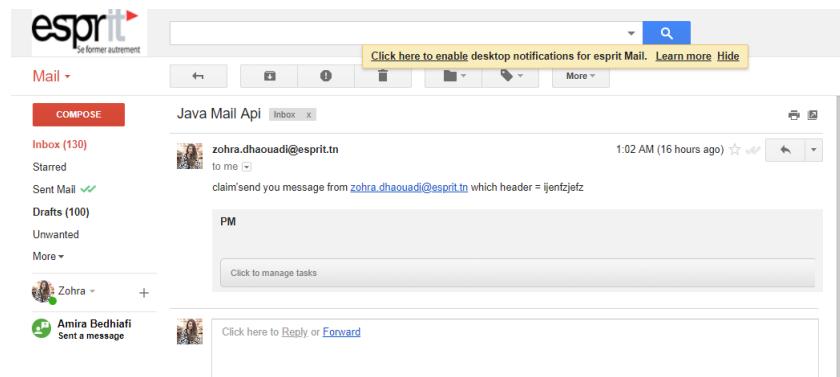
For the test of the application, we didn't find any important constraints. For that from

## Chapter 4. Technical branch

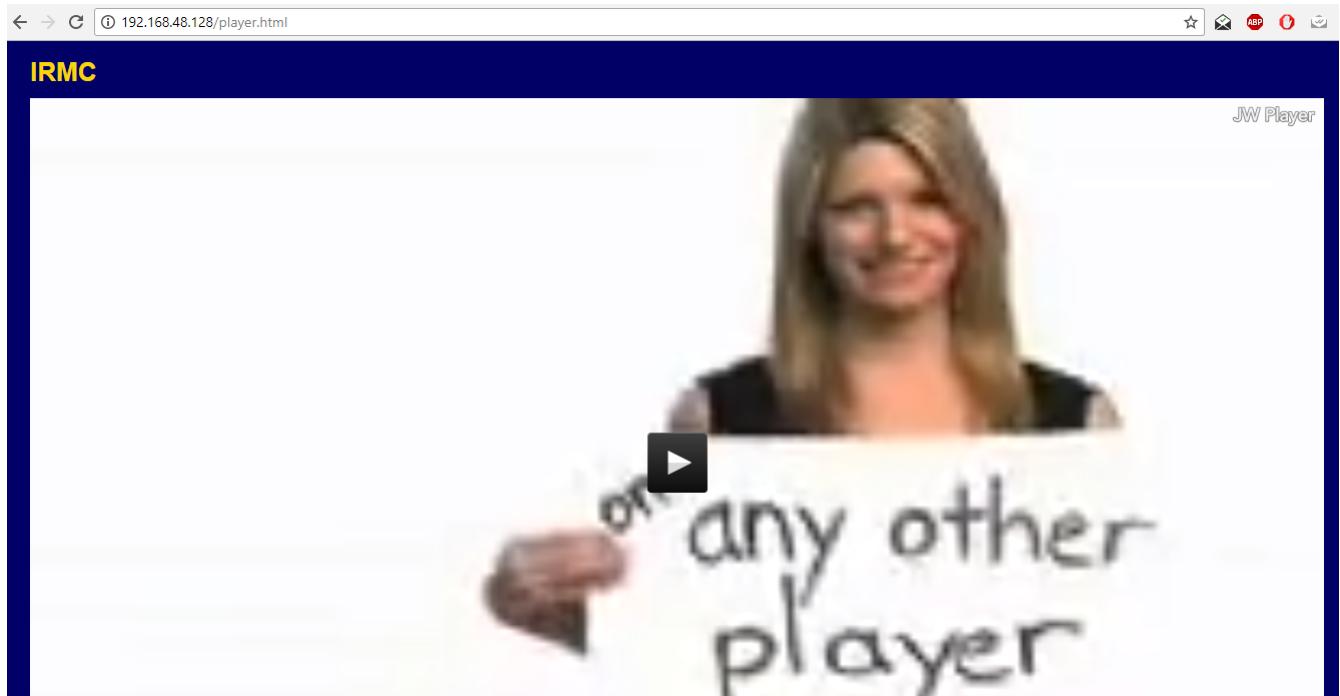


**Figure 4.41:** Contact us

At the beginning we shared the same template and we worked on the same DATABASE. We started by creating a user with administrator access and we log in with it. Now we can upload a document from the computer, affected to a destination and display all the details, we can also make comments on it. Moreover, we made tasks with a deadline and affected to a responsible, this last will validate his tasks before the deadline.



**Figure 4.42:** Mail reception



**Figure 4.43:** About us/Streaming

The next step was the test of sending mails between users and between in box mails, as an internal and external box.

## 4.10 Conclusion

After we had defined and explained our software development process. In this module, we designed the conceptual diagrams which helped us understand how our application is supposed to work. Knowing how the packages interacts with each other, we were able to develop the java EE part of our project management tools and its rest services. As a matter of fact, we are now able to interact with our database by adding, deleting, updating or displaying records through the interfaces. We are now ready to start working on our database with the .NET technologies and consume it with services rest in the mobile module.

# 5

## Mobile/Web Module

### Introduction

We have previously conducted the analysis phase in order to identify the actors, the functional needs of each, and non-functional requirements of any application while taking into consideration the physical system architecture. Thanks to this, we are now ready to start the conceptual phase and to work on its deployment in the application java EE. In this conceptual phase, we are able to explain better how our functionality work. This is a crucial step before starting developing our application. All the necessary diagrams help us code this part of our platform.

### 5.1 Functional requirements Mobile/W.S

The conceptual class diagram offers a much more detailed view of how our classes are supposed to interact with each other, as well as to indicate us the necessary attributes and methods we need to implement in our code. After designing the required diagrams, we are now ready to focus on developing our application.

Module	Sub-Module	Functionality code	Functionality	Responsible
Mobile	Comment management	Module 01 Module 02 Module 03 Module 04 Module 05 Module 06	Add comment Delete comment Modify comment Consult comments Send mail Consult mail	Dhaouadi Zohra
	Contact us	Module 07 Module 08	Generate QR code Scan QR code	
	QR code	Module 09 Module 10 Module 11 Module 12	Add library Consult library Delete library Modify library	Nader Kasri
	Documentary center management			

**Figure 5.1:** Technical branch

Web service	Authentication	Module 13	Authentication	Dhaouadi Zohra
	User management	Module 14 Module 15 Module 16 Module 17 Module 18	Token GET user Post user Delete user Put user	
	Mail management	Module 19 Module 20 Module 21 Module 22	Get mails Post mails Delete mails Put mails	
	Comment management	Module 23 Module 24 Module 25 Module 26	Get comments Delete comments Put comments Post comments	
	Task management	Module 27 Module 28 Module 29 Module 30	Get tasks Post tasks Delete tasks Put tasks	
	Document management	Module 31 Module 32 Module 33 Module 34	Get document Put document Post document Delete document	
	Library management	Module 35 Module 36 Module 37 Module 38	Get libraries Post library Put library Delete library	Nader Kasri

**Figure 5.2:** Web service functional requirements

## 5.2 Architecture and methodology

In our application we used the architecture M.V.C and different methodology like Retrofit and volley libraries:

This library makes downloading JSON or XML data from a web API fairly straightforward. Once the data is downloaded then it is parsed into a Plain Old Java Object (POJO) which must be defined for each "resource" in the response.

### 5.2.1 Retrofit2

Retrofit is a type-safe REST client for Android (or just Java) developed by Square. The library provides a powerful framework for authenticating and interacting with APIs and sending network requests with OkHttp. See this guide to understand how OkHttp works.



**Figure 5.3:** Retrofit2 android methodology

### 5.2.2 Volley

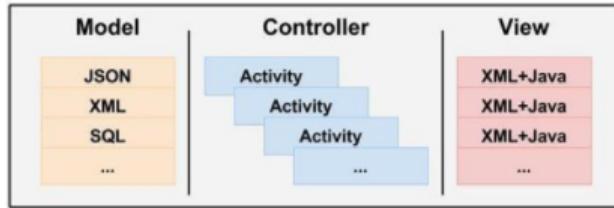
Volley is an HTTP library that makes networking for Android apps easier and most importantly, faster. Volley excels at RPC-type operations used to populate a UI, such as fetching a page of search results as structured data. It integrates easily with any protocol and comes out of the box with support for raw strings, images, and JSON. By providing built-in support for the features you need, Volley frees you from writing boilerplate code and allows you to concentrate on the logic that is specific to your app.



**Figure 5.4:** Volley android methodology

### 5.2.3 M.V.C modeling

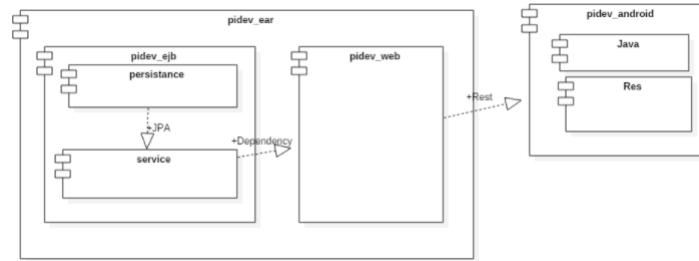
Model–View–Controller (M.V.C) is a software architectural pattern for implementing user interfaces on computers. It divides a given application into three interconnected parts. This is done to separate internal representations of information from the ways information is presented to, and accepted from, the user. The M.V.C design pattern decouples these major components allowing for efficient code reuse and parallel development.



**Figure 5.5:** M.V.C architecture

## 5.3 Preliminary and detailed conception

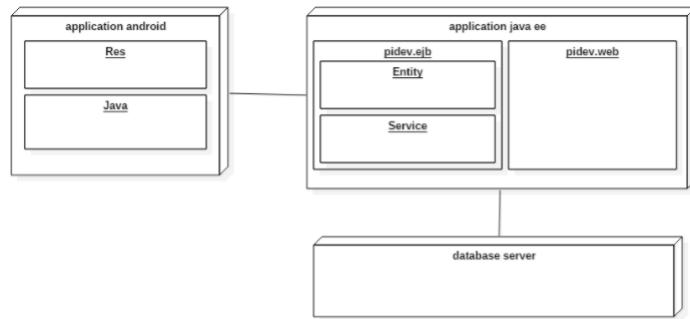
### 5.3.1 Logical architecture



**Figure 5.6:** Logical architecture

### 5.3.2 Physical architecture

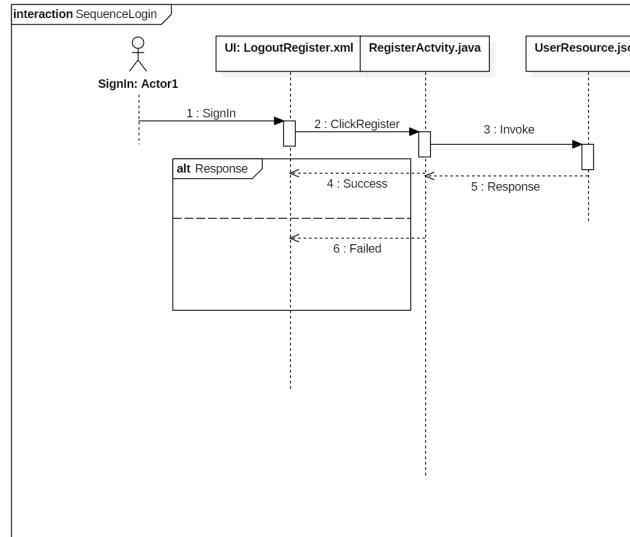
The physical architecture of our application is presented using U.M.L deployment diagram.



**Figure 5.7:** Physical architecture

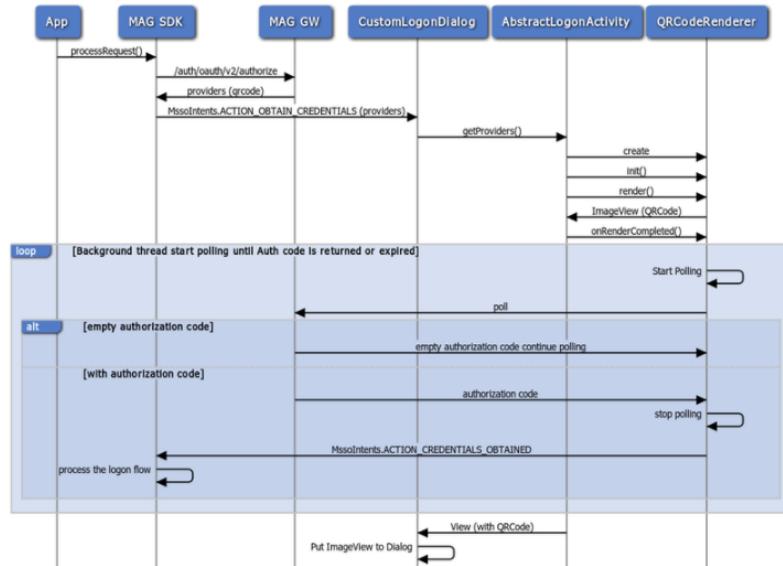
### 5.3.3 Object sequence diagram

#### 5.3.3.1 Authentication sequence diagram

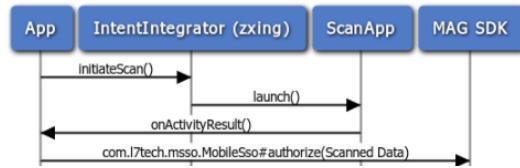


**Figure 5.8:** Authentication sequence diagram

#### 5.3.3.2 QRCode generator sequence diagram

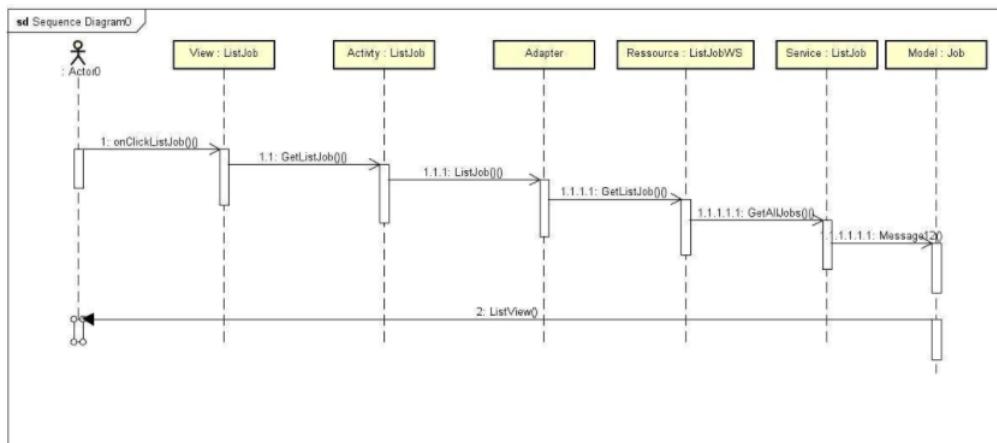


**Figure 5.9:** QRCode generator sequence diagram



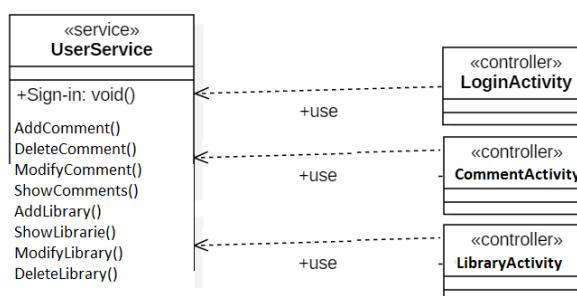
**Figure 5.10:** QrCode scan sequence diagram

### 5.3.3.3 Comment on jobs offer sequence diagram



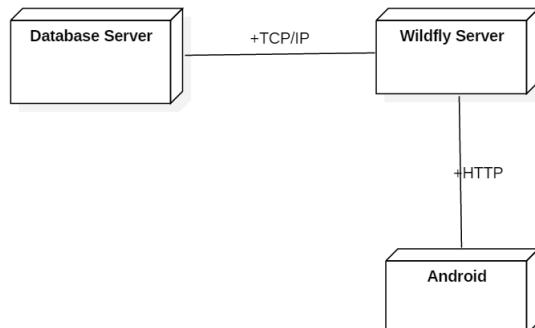
**Figure 5.11:** Comment on jobs offer sequence diagram

### 5.3.4 Class diagram



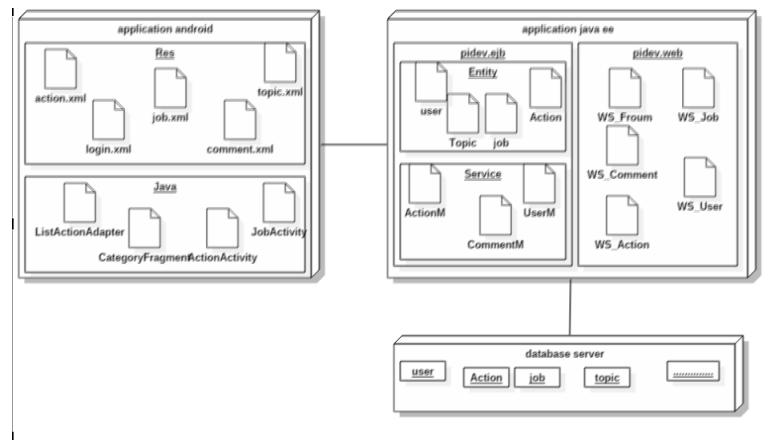
**Figure 5.12:** Class diagram

### 5.3.5 Deployment diagram



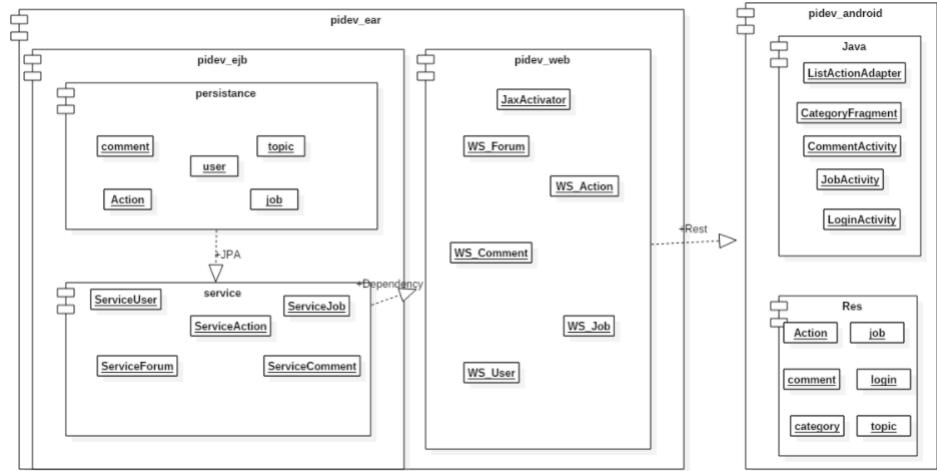
**Figure 5.13:** Deployment diagram

### 5.3.6 Detailed deployment diagram



**Figure 5.14:** Deployment diagram

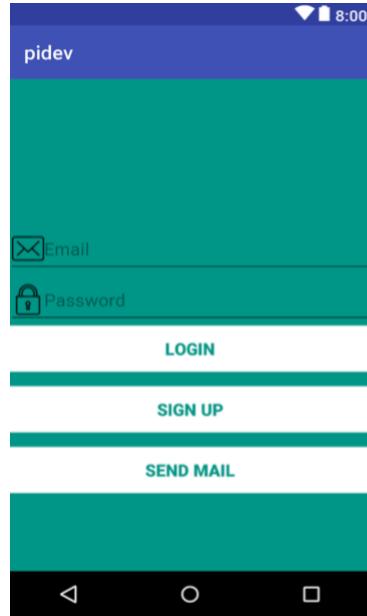
### 5.3.7 Component diagram



**Figure 5.15:** Component diagram

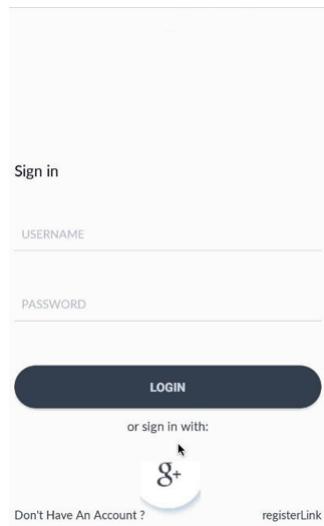
## 5.4 Coding and test

### 5.4.1 Login interface



**Figure 5.16:** Login interface

#### 5.4.2 SignIN interface



**Figure 5.17:** SignIN interface

#### 5.4.3 Library center localization interface



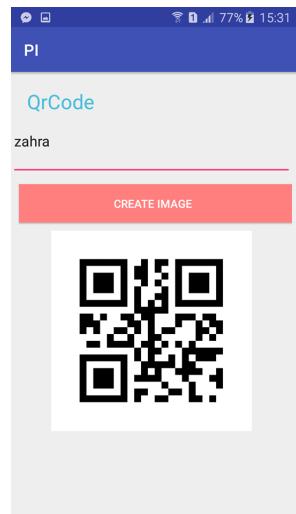
**Figure 5.18:** Library center localization interface

#### 5.4.4 Library center localization interface



**Figure 5.19:** Library center localization interface

#### 5.4.5 Generate QR



**Figure 5.20:** Generate QR

#### 5.4.6 ScanQR

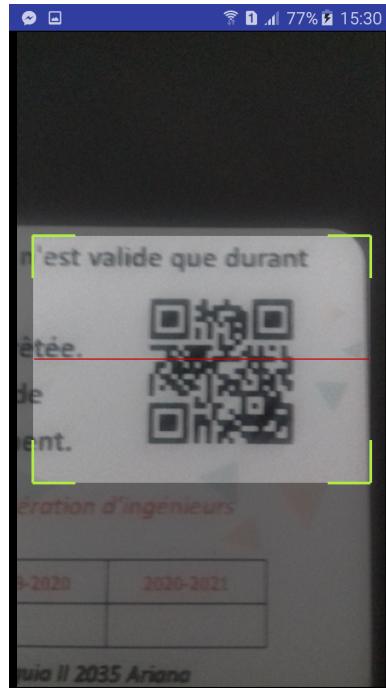


Figure 5.21: ScanQR

#### 5.4.7 Resultat scanQR console

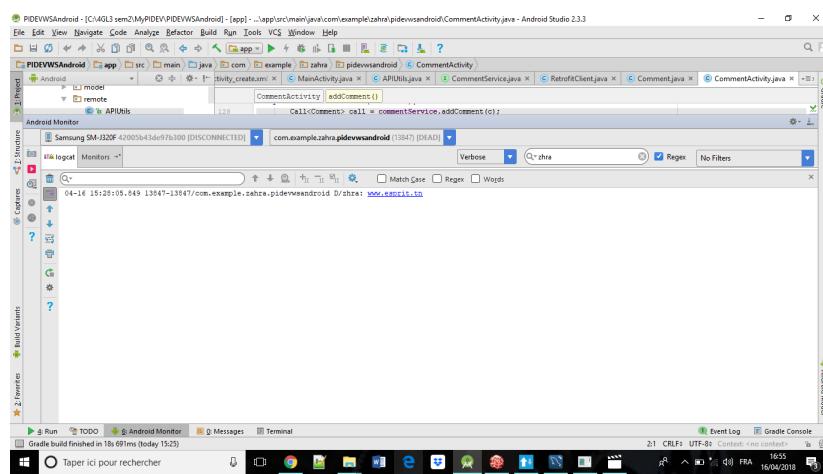


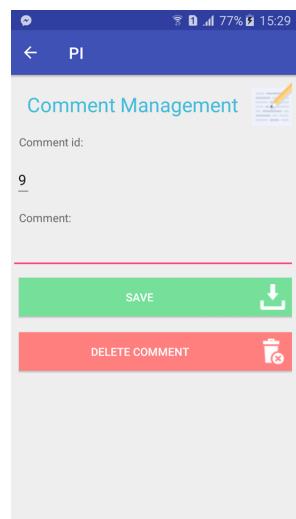
Figure 5.22: Resultat scanQR console

#### 5.4.8 Add comment interface



**Figure 5.23:** Add comment interface

#### 5.4.9 Delete/Modify comment interface



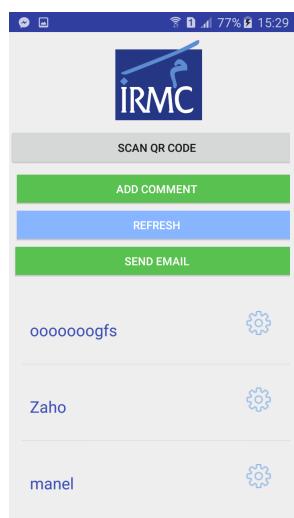
**Figure 5.24:** Delete/Modify comment interface

#### 5.4.10 Mail interface



**Figure 5.25:** Mail interface

#### 5.4.11 Main and get comments interface



**Figure 5.26:** Main and get comments interface

## 5.5 Tip of implementation and integration

The team need to work on their web services using the Rest technology to communicate with the server and they use the GitLab as a tool to integrate the different parts of the android application

## 5.6 Test the application

In this test part, we used several types to check our work. Integration tests to validate the integration of the different modules between them and unit tests to validate the quality of the code and the performance

## 5.7 Conclusion

In this module, we designed the conceptual diagrams which helped us understand how our application is supposed to work. Knowing how our classes interact with each other, we were able to develop the android part of our project management tools. As a matter of fact, we are now able to interact with our database by adding, deleting, updating or displaying records through the interfaces. We are now ready to start working on our database with the .NET technologies .

# 6

## Asp.NET

### Introduction

Through the last development module, we engaged ourselves into the use of an external technology such as the Framework .Net in order to enhance the inter-operable aspect of our application. Furthermore, we developed some of the main functionality of our final release following the .Net architecture requirements. Thus, we can take advantage of the various .Net proposals.

### Technical Branch

In order to develop our Project Management software, we needed various tools and programming languages.

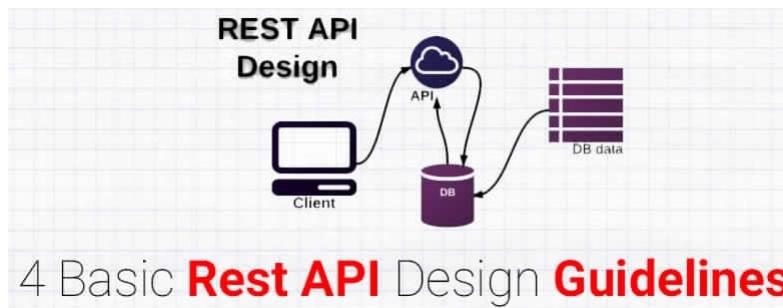
#### 6.1 Tools and technologies

The .Net Framework: During this module, we exploited multiple tools related to the “.Net Framework” in its 4.5.2 version which is a software framework developed by Microsoft that runs primarily on Microsoft Windows.

It includes a large class library known as Framework Class Library (FCL) and provides language inter-operable (each language can use code written in other languages) across several programming languages.

The application domain of the framework reaches desktop or web application through Forms or ASP.NET respectively. But its role doesn't stop here, as it gives developers the ability to host and consume SOAP or REST web services thorough the Web API. The bundle of functionality provided by the .NET framework is powerful as most of the basic functionality are handled automatically for the developers.

These feature makes the developers only think about the business logic which is the most important part. - Restful Web services through .Net Web API REST defines a set of archi-



**Figure 6.1:** Restful Web-API

tectural principles by which you can design Web services that focus on a system's resources, including how resource states are addressed and transferred over HTTP by a wide range of clients written in different languages.

Restful Web services have been the main development part of our second time box.

This would ensure a better inter-operable and a specific exploitation of every single resource. Furthermore, we adopted a concrete strategy according to the development of the main functionality related to each branch. In fact, we have chosen to develop our basic functionality using the .Net framework and there after exploiting those treatments into our final Java Enterprise Edition Application.

All in all, compared to the SOAP protocol the architectural Style Rest ensures a better and a lighter use of each resource developed in the first time box. It is in fact mostly designed to light web and mobile applications.

## 6.2 Language developing tools

### 6.2.1 C# language



**Figure 6.2:** Asp.net C

C is an object-oriented programming language from Microsoft that aims to combine the computing power of C++ with the programming ease of Visual Basic. C is based on C++ and contains features similar to those of Java.

### 6.2.2 Asp.net language



**Figure 6.3:** Asp.net developing tool

.Net is a programming infrastructure created by Microsoft for building, deploying, and running applications and services that use .NET technologies, such as desktop applications and Web services. The .NET Framework contains three major parts:

- The Common Language Run-time
- The Framework Class Library
- ASP.NET

### 6.2.3 Entity Framework

Is a set of technologies in ADO.NET that support the development of data-driven software applications. Architects and data-driven application developers are faced with the need to

achieve two very different objectives.



**Figure 6.4:** ADO.NET: Entity framework

## 6.3 Design developing tool and template

### 6.3.1 CSS3,HTML,Javascript



**Figure 6.5:** CSS3,HTML,Javascript for template design

Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language.

JavaScript, often abbreviated as JS, is a high-level, dynamic, weakly typed, prototype-based, multi-paradigm, and interpreted programming language

### 6.3.2 Razor CsHtml



**Figure 6.6:** Razor CsHtml

Razor is an ASP.NET programming syntax used to create dynamic web pages with the C or Visual Basic .NET programming languages.

### 6.3.3 jQuery

jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML. It is free, open-source software using the permissive MIT License.



**Figure 6.7:** jQuery

## 6.4 MYSQL Connector



**Figure 6.8:** MYSQL Connector

MySQL Connectors provides standards-based drivers for JDBC, ODBC, and .Net enabling developers to build database applications in their language of choice. In addition, a native C library allows developers to embed MySQL directly into their applications.

## Preliminary and detailed conception

## 6.5 .NET functional requirement

Module	Sub-Module	Functionality code	Functionality	Responsible
<a href="#">ASP.NET</a>	E-mails management	NET 01 NET 02	Get E-mails Delete received E-mails	Dhaouadi Zohra
	Agents leave management	NET 03 NET 04 NET 05 NET 06	Apply for a leave Delay a leave request Modify a leave request Consult leave requests	
	Administrator leave management	NET 07 NET 08 NET 09	Approve a leave request Disapprove a leave request Deny an agent request	
	Contact us	NET 10	Streaming service	
	Document centry management	NET 11 NET 12 NET 13	Add a library Delete a library Edit a library	Nader Kasri

Figure 6.9: .NET functional requirement

## 6.6 Logical architecture

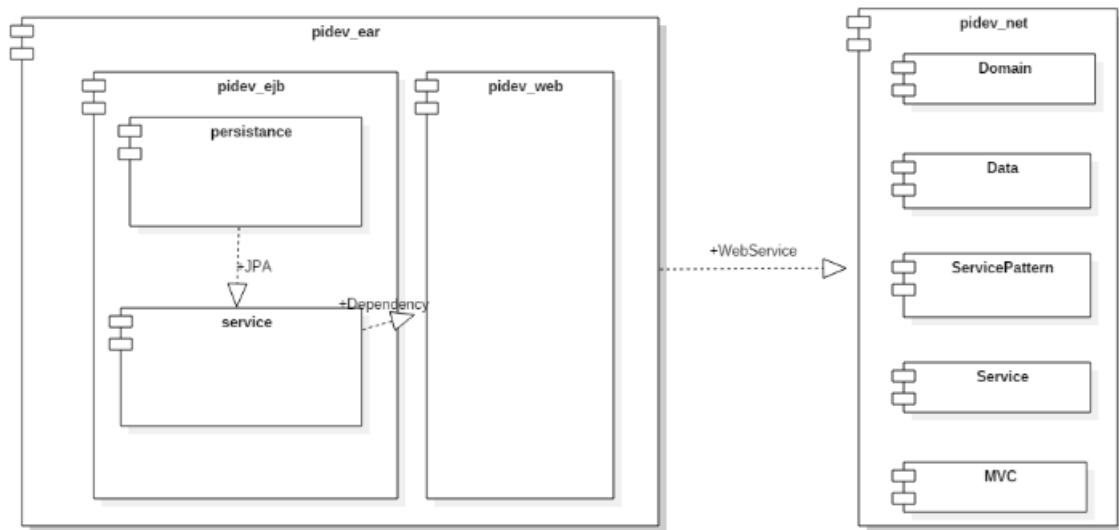
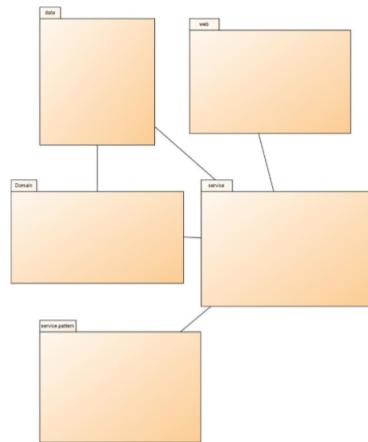


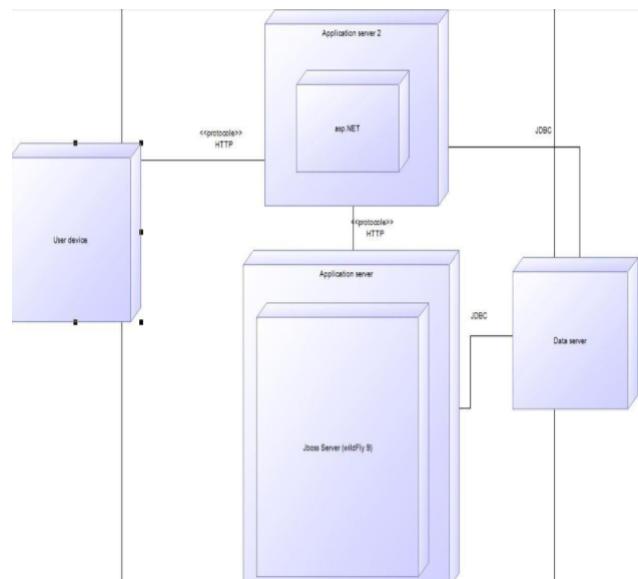
Figure 6.10: .NET functional requirement

### 6.6.1 .NET web reference logical architecture



**Figure 6.11:** .NET web logical architecture

## 6.7 Physical architecture



**Figure 6.12:** Physical architecture

## 6.8 Object Sequence Diagram

An object sequence diagram is an interaction diagram that shows how objects operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence.

### 6.8.1 Login sequence diagram

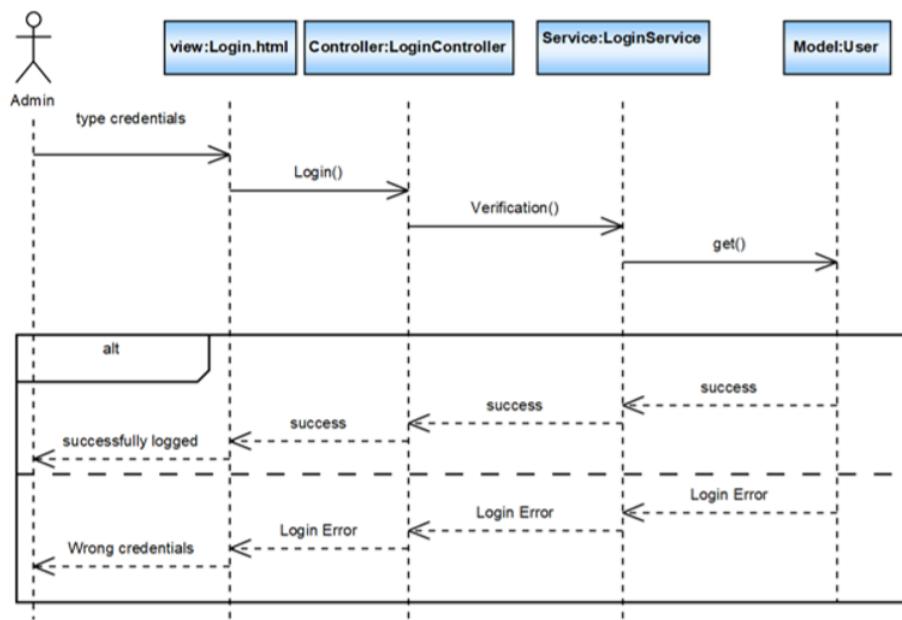


Figure 6.13: Login sequence diagram

### 6.8.2 Admin Leave sequence diagram

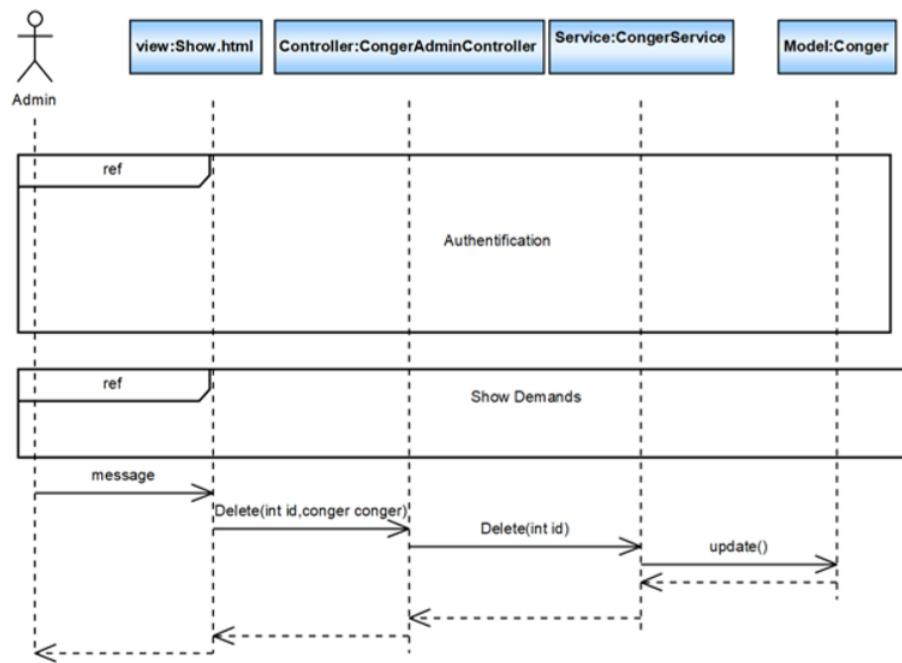


Figure 6.14: Admin Leave sequence diagram

### 6.8.3 Agent Leave sequence diagram

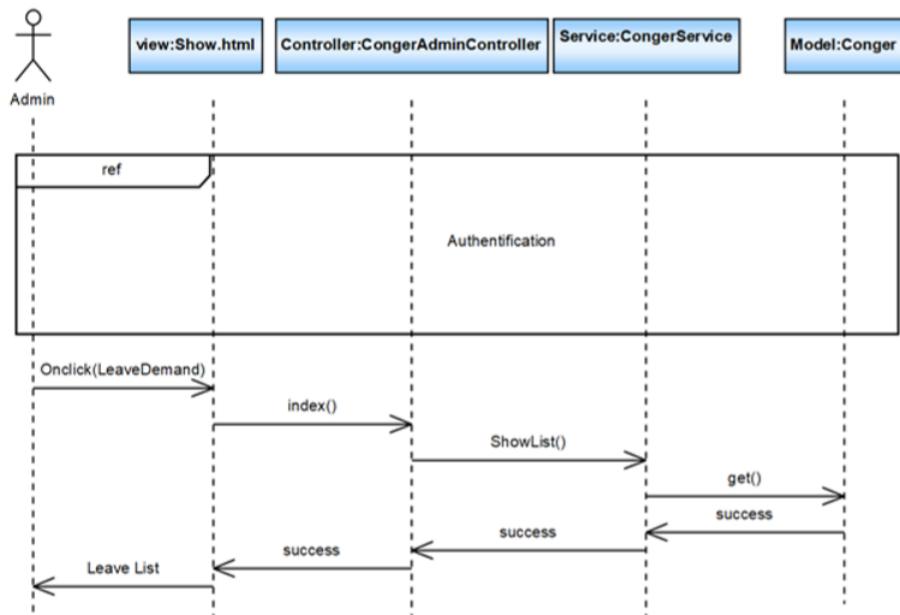


Figure 6.15: Agent Leave sequence diagram

## 6.9 Class diagram

Class diagrams show the classes of the system, their inter-relationships, and the operations and attributes of the classes .

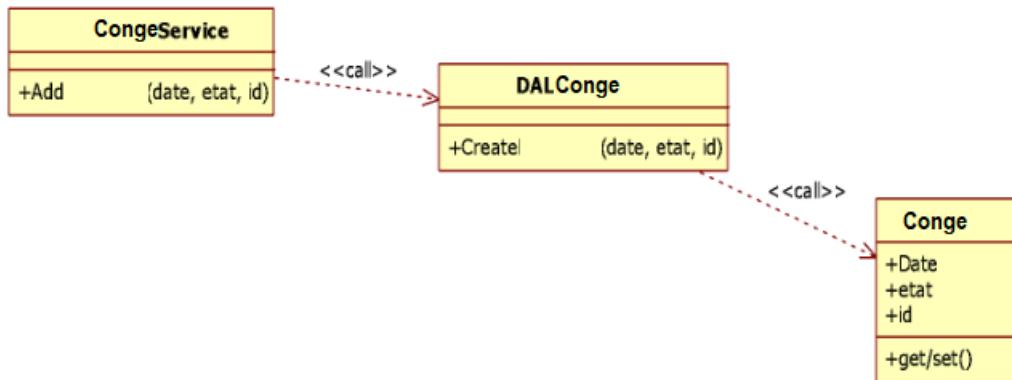


Figure 6.16: Class diagram

## 6.10 Component diagram

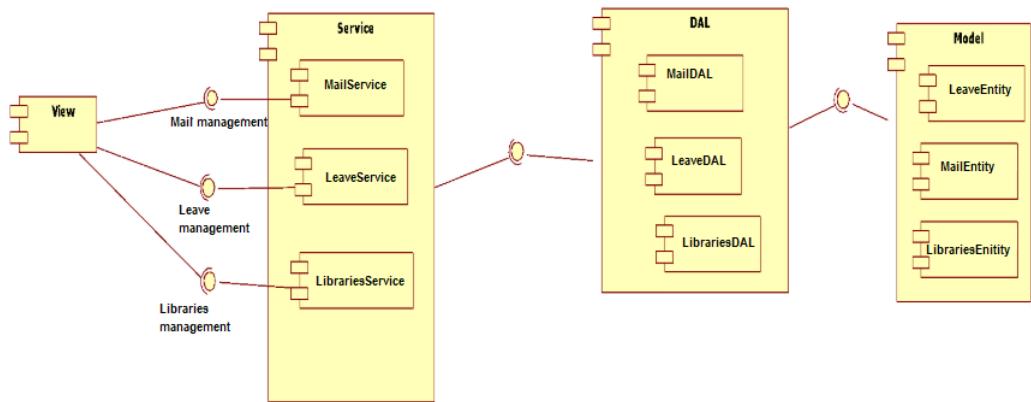


Figure 6.17: Component diagram

## 6.11 Deployment diagram

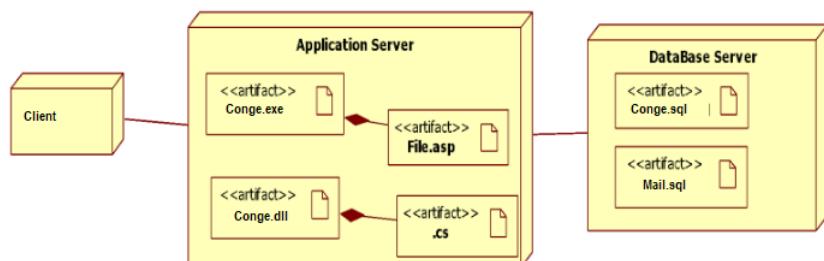


Figure 6.18: Deployment diagram

## 6.12 Package diagram

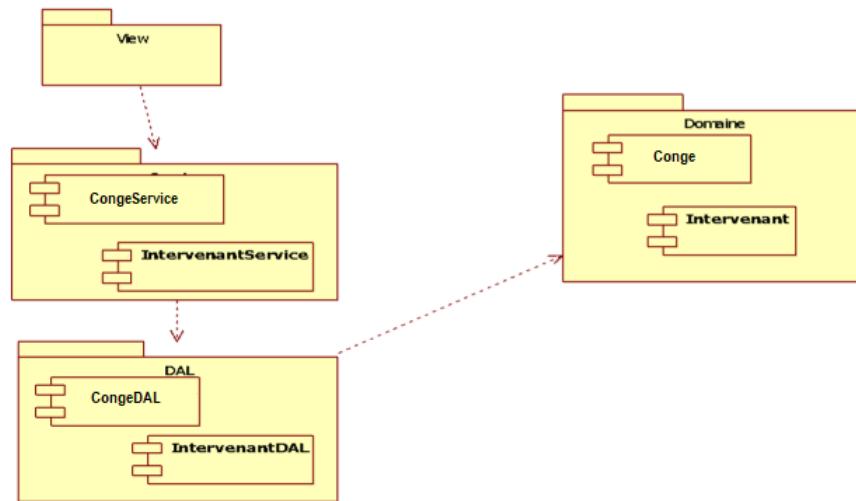


Figure 6.19: Package diagram

## Coding and test

### 6.13 Application interfaces

#### 6.13.1 Login interface

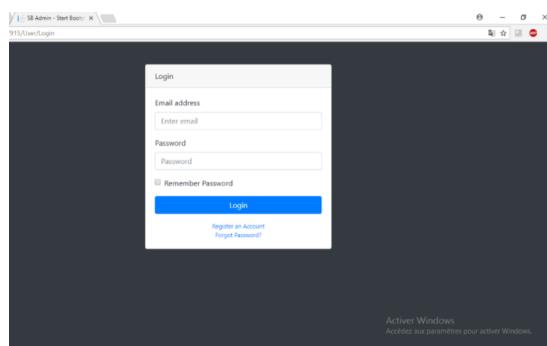
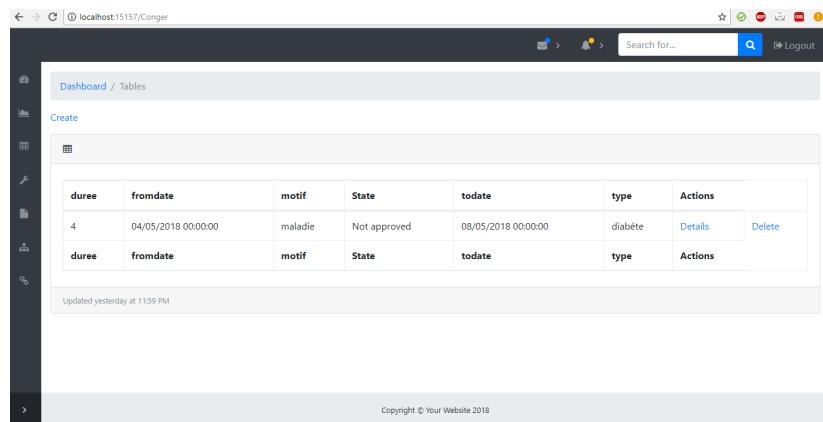


Figure 6.20: Login interface

### 6.13.2 Agent leave interface



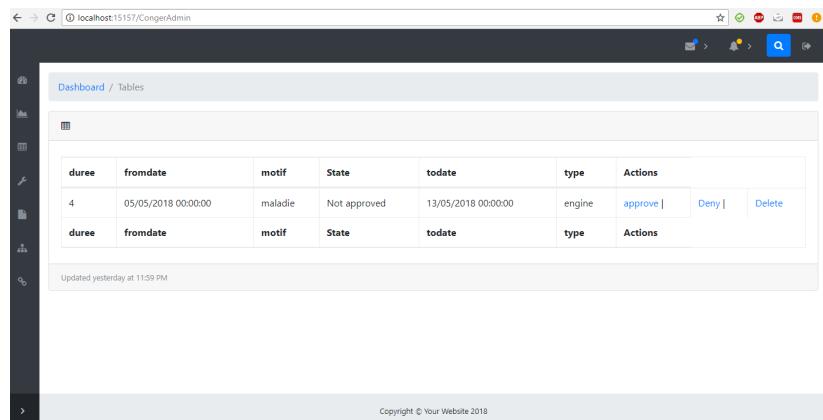
The screenshot shows a web application interface for managing agent leave requests. The URL is `localhost:15157/Conger`. The page title is "Dashboard / Tables". A sidebar on the left contains icons for Home, Create, Delete, and others. The main content area displays a table with the following columns: duree, fromdate, motif, State, todate, type, and Actions. There is one visible row:

duree	fromdate	motif	State	todate	type	Actions
4	04/05/2018 00:00:00	maladie	Not approved	08/05/2018 00:00:00	diabète	<a href="#">Details</a>   <a href="#">Delete</a>

A message at the bottom states "Updated yesterday at 11:59 PM". The footer includes "Copyright © Your Website 2018".

Figure 6.21: Agent leave request interface

### 6.13.3 Administrator leave management interface



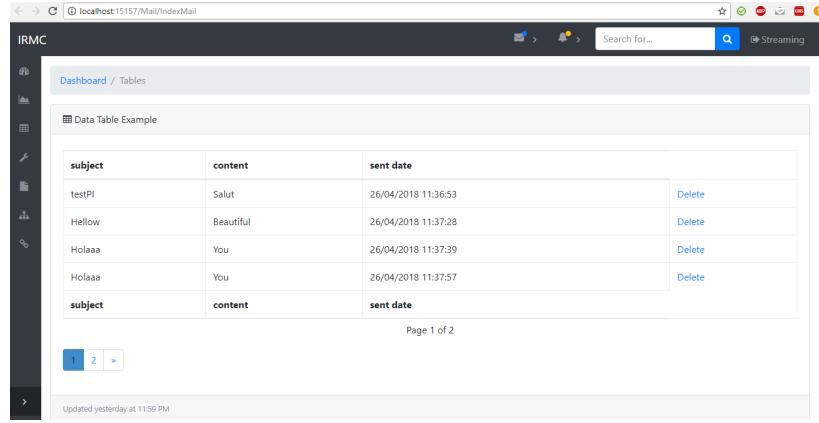
The screenshot shows a web application interface for managing administrator leave requests. The URL is `localhost:15157/CongerAdmin`. The page title is "Dashboard / Tables". A sidebar on the left contains icons for Home, Create, Delete, and others. The main content area displays a table with the following columns: duree, fromdate, motif, State, todate, type, and Actions. There is one visible row:

duree	fromdate	motif	State	todate	type	Actions
4	05/05/2018 00:00:00	maladie	Not approved	13/05/2018 00:00:00	engine	<a href="#">approve</a>   <a href="#">Deny</a>   <a href="#">Delete</a>

A message at the bottom states "Updated yesterday at 11:59 PM". The footer includes "Copyright © Your Website 2018".

Figure 6.22: Administrator leave management interface

### 6.13.4 Mail management interface



**Figure 6.23:** Mail management interface

## Tip of implementation and integration

For the implementation we started with a reverse engineering of the entities and after that we integrated the template, so now everyone could start the development using Gitlab to make the integration easier, where it's is the most widely used modern version control system in the world today. Git has the functionality, performance, security and flexibility that most teams and individual developers need. For more details, we choose : html5 bootstrap template and we used an existing database via the reverse engineering.

## 6.14 Test the application

At this test part, we used several types to check our work. Integration tests to validate the integration of the different modules between them and unit tests to validate the quality of the code and the performance. For the test of the application, we didn't find any obstacles because from the beginning we had the same template and we worked on the same DATABASE and we used the same package namespace for all our services and chose to use the same path for our jax-rs api (we created the same jaxrsactivator with a general path that all our members use).

We started by administrator access and we tried to log in with it. Now we can upload a document from the computer, affected to a destination and display all the details, we can also make comments on it. Moreover, we made tasks with a deadline and affected to a responsible, this last will validate his tasks before the deadline.

The next step was the test of sending mails between users and we implemented java mail-Api to allow us to send emails outside our application and we tested it and it works fine. The tests we did are Static and dynamic testing.

List of tests we did:

Integration testing.

Black-Box testing

Visual testing

Grey-Box testing

## Conclusion

During the last module, we have been led through the development of the basic functionality of our application. Thus, we mainly gained a huge experience through the handling of new technologies such as the .Net framework and its Web API tools. This experience enhanced our C knowledge and our .Net capabilities.

# 7

## Conclusions and perspectives

Since the documentation is one of the major problem here and the main reason that gives rise to such actions is the S.E.O (Search Engine Optimization), it is highly important to toughen up the representation layer of our application .

Moreover, inspections at our frontiers or high technology performance should be carried out on a wider scale and the agent of the IRMC should be better trained in order to efficiently detect and deter the waste shipments of some articles and its vulnerability. In connection with that, our goal is to work on giving e-lectures to customs researchers and to covers both, theoretical information and practical tutorials that come in handy during inspections.

Analyze information provided in the e-documents is the most important service, what things to pay attention to? what are most common researches for this issue, what to look for when

examining the uploaded or downloaded files and videos,serving for the internaut intention.

In addition, An e-environment of supervising of the whole management is aiming to be working on, as a future project that should shed some more light on the faced issues of uses and facilitate the work of various agents in the institute,including the people that work at our division and simple visitors.