Ecole Supérieure Privée d'Ingénierie et de Technologies

Ecole Supérieure Privée d'ingénieure et de Technologies

Chapitre II

Module JEE

Carried Out by

Benjemaa Amor
Jarrar Jawhar
Jamel Mustapha
Guibane Najmeddine
Kalai Rania
Zarkouna Malek

University Tutor :

Academic Year : 2017-2018

Outline

1. Technical Branch 4
   a. Tools et Technologies 4
   b. Logical Architecture 6
   c. Physical Architecture 6
2. Conception préliminaire et détaillée 7
   a. Object sequence diagrams and design classes 7
   b. Component diagram 9
   c. Deployment diagram 9
3. Codage et test 10
   a. Interfaces of the application 10

Introduction

Java Platform, Enterprise Edition (Java EE) is the standard in community-driven enterprise software. Java EE is developed using the Java Community Process, with contributions from industry experts, commercial and open source organizations, Java User Groups, and countless individuals. Each release integrates new features that align with industry needs, improves application portability, and increases developer productivity.

Due to the enterprise software side of our Bartering application along the first module we implemented the specification in its 7th version. It offers a panoply of tools and APIs that enhances de development capabilities. Java EE led us to follow well-formed specifications such
as JPA and EJB in order to separate the core modules of our application from the client side.
Java EE is implemented by a variety of application servers. In our case, we used “WildFly 9.x”
which is an open source solution proposed by the Red Hat community.

In this project timebox, we tended not only to call the previously developed web services but also concretize the main functionalities of our application. This led us to the Design phase during which we introduced some related diagrams such as the Use Case Diagram and the Design Class diagram following the global architecture of our application.

1. Technical Branch

a. Tools et Technologies

Jboss Developer Studio

JBoss is a division of Red Hat that provides support for the JBoss open source application server program and related middleware services marketed under the JBoss Enterprise Middleware brand. JBoss is an open source alternative to commercial offerings from IBM WebSphere and SAP NetWeaver. JBoss developer Studio is an Integrated Development Environment which provides a variety of tools that enhances the developer capabilities. One of the most important tools integrated in the Jboss Developer Studio is the debugging mode. An advanced feature that helps the developer fix coding issues.

Maven

Apache 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.

MySql 5.6

MySQL is the world's most popular open source database. With its proven performance, reliability, and ease-of-use, MySQL has become the leading database choice for web-based applications, used by high profile web properties including Facebook, Twitter, YouTube, and all five of the top five websites. Additionally, it is an extremely popular choice as embedded database, distributed by thousands of ISVs and OEMs.

WildFly 9.x

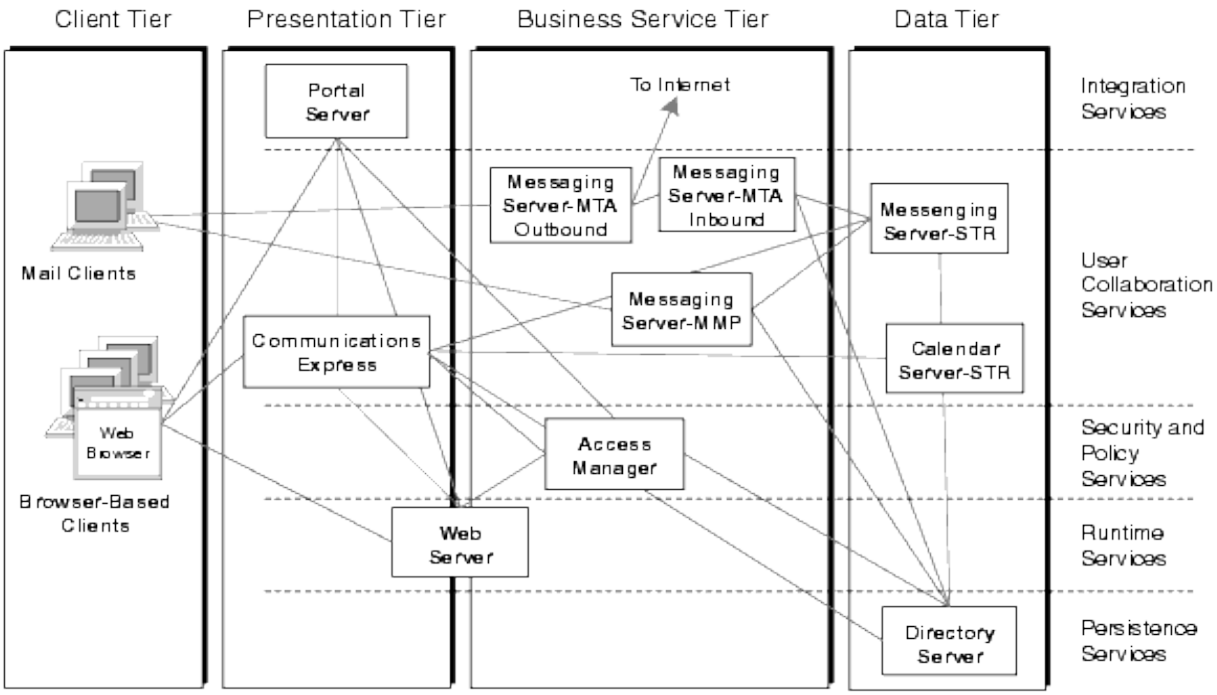
WildFly, formerly known as JBoss AS, or simply JBoss, is an application server authored by JBoss, now developed by Red Hat. WildFly is written in Java, and implements the Java Platform, Enterprise Edition (Java EE) specification. It runs on multiple platforms. WildFly is free and open-source software, subject to the requirements of the GNU Lesser General Public License (LGPL), version 2.1.

Hibernate

Hibernate ORM enables developers to more easily write applications whose data outlives the application process. As an Object/Relational Mapping (ORM) framework, hibernate is concerned with data persistence as it applies to relational databases (via JDBC).

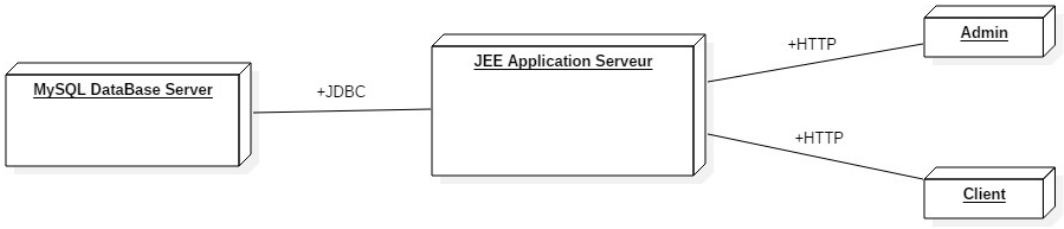
b. Logical Architecture

This section provides brief descriptions of the four logical tiers shown in the figure above The descriptions refer to application components implemented using the Java Enterprise Edition (JEE) component model. However, other distributed component models, such as CORBA, also support this architecture.



c. Physical Architecture

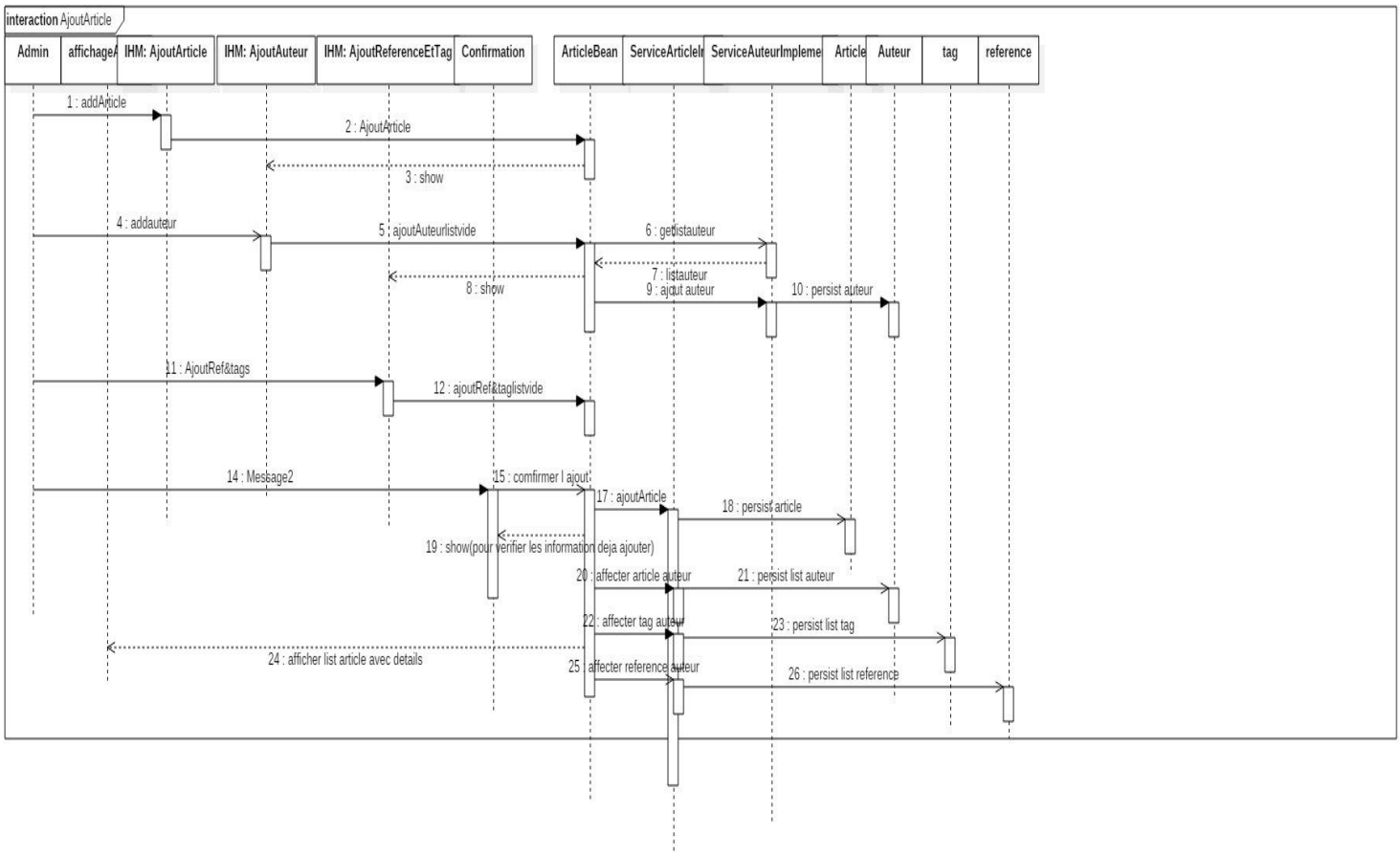
The physical architecture is a physical representation of important its interfaces and major system components. The physical architecture identifies the transportation systems and the information exchanges that support ITS. The figure below represents our own physical architecture that we designed.

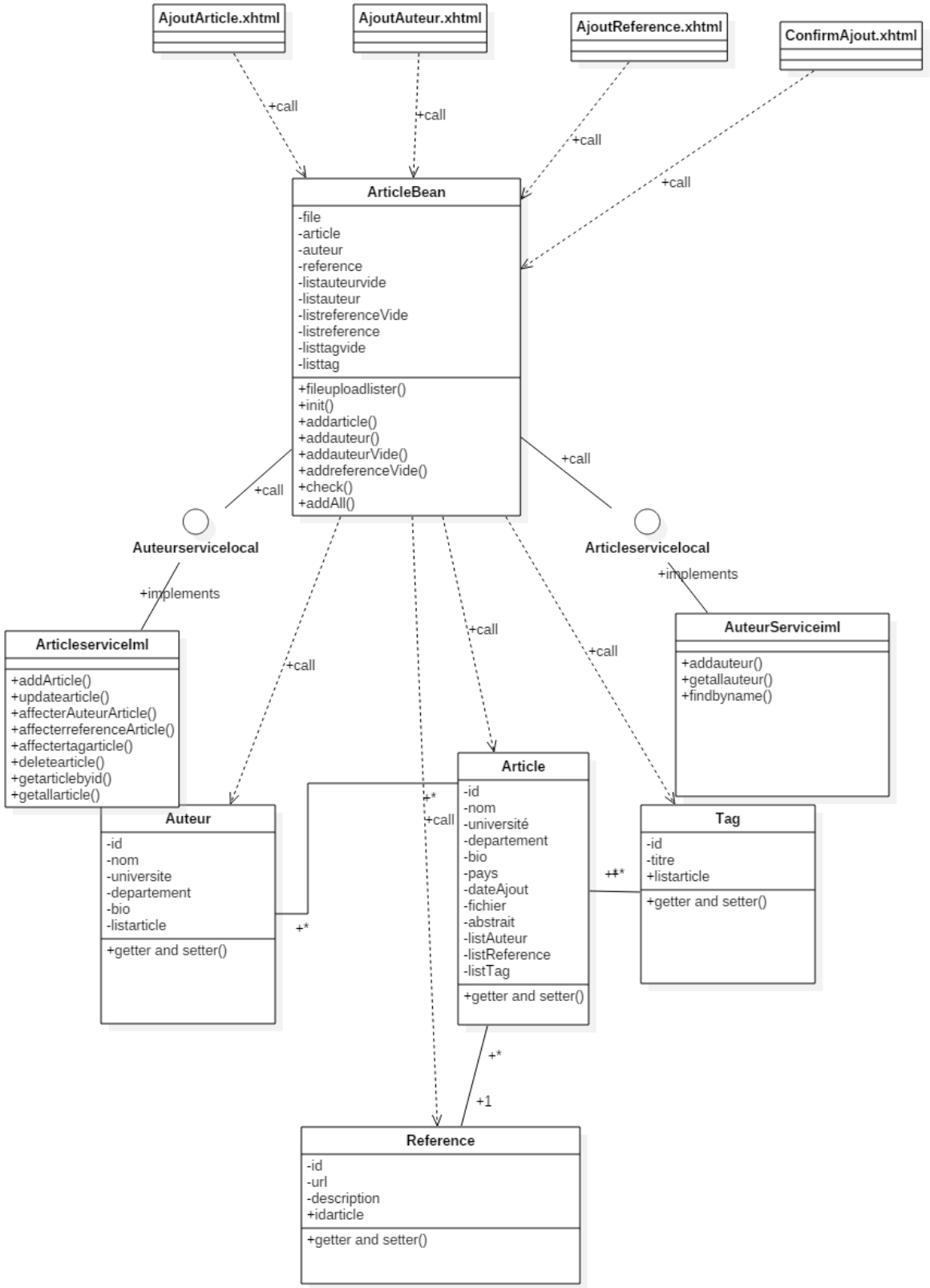


2. Conception préliminaire et détaillée

a. Object sequence diagrams and design classes

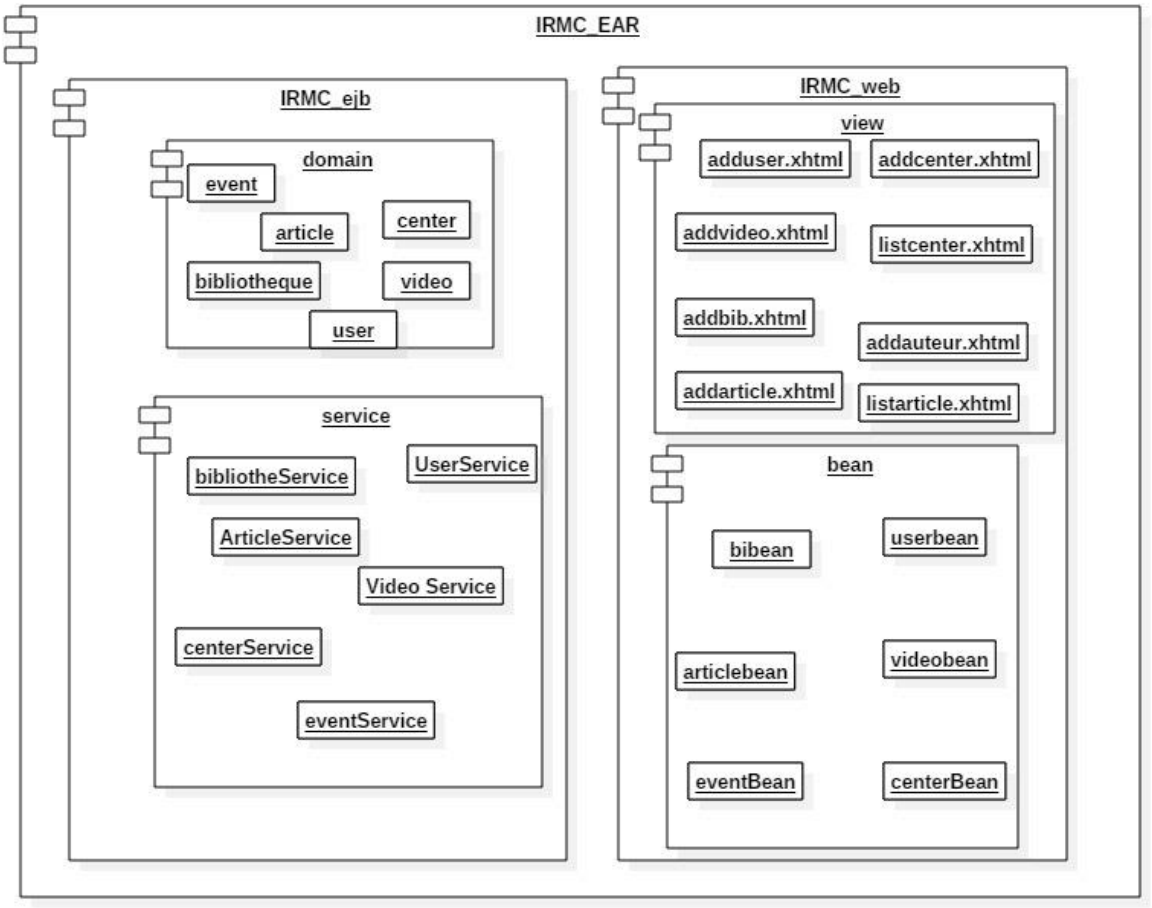
A sequence diagram has a collaboration with a superimposed interaction. In general a sequence focuses on one specific type of action that the description should be strengthened.




AjoutArticle.xhtml
AjoutAuteur.xhtml
AjoutReference.xhtml
ConfirmAjout.xhtml
+call
ArticleBean
-file
-article
-auteur
-reference
-listauteurvide
-listauteur
-listreferenceVide
-listreference
-listtagvide
-listtag
+fileuploadlister()
+init()
+addarticle()
+addauteur()
+addauteurVide()
+addreferenceVide()
+check()
+addAll()
Auteurservicelocal
Articleservicelocal
+implements
ArticleserviceIml
+addArticle()
+updatearticle()
+affecterAuteurArticle()
+affecterreferenceArticle()
+affectertagarticle()
+deletearticle()
+getarticlebyid()
+getallarticle()
AuteurServiceiml
+addauteur()
+getallauteur()
+findbyname()
Auteur
-id
-nom
-universite
-departement
-bio
-listarticle
+getter and setter()
Article
-id
-nom
-université
-departement
-bio
-pays
-dateAjout
-fichier
-abstrait
-listAuteur
-listReference
-listTag
+getter and setter()
Tag
-id
-titre
+listarticle
+getter and setter()
+*
+1
Reference
-id
-url
-description
+idarticle
+getter and setter()

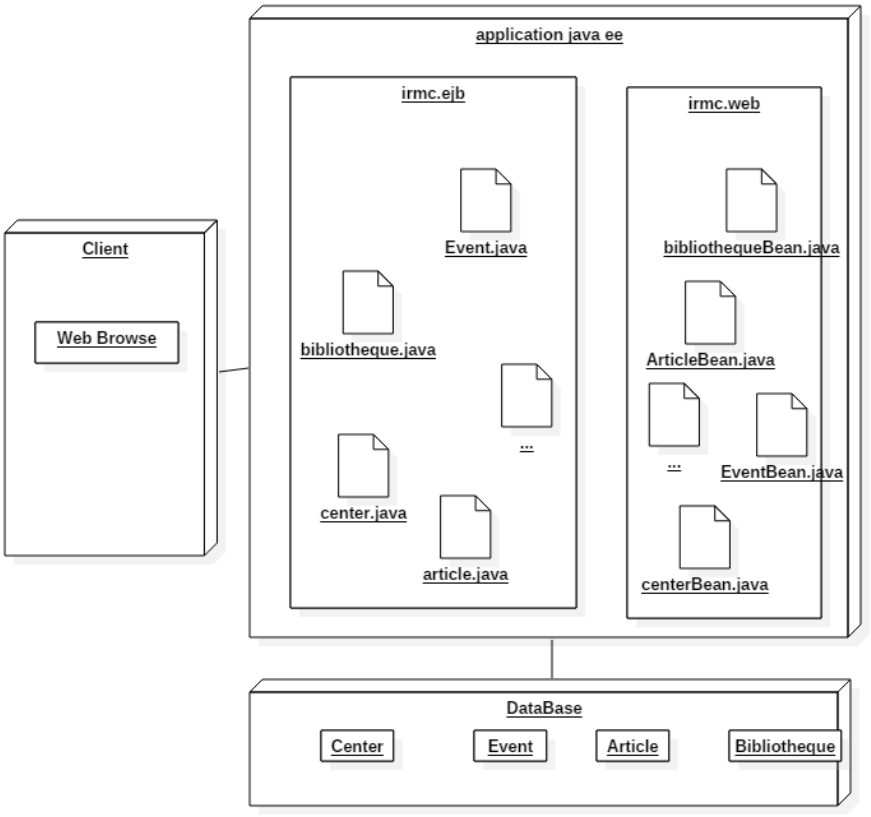
b. Component diagram

A component diagram depicts how components are wired together to form larger components or software systems. They are used to illustrate the structure of arbitrarily complex systems.



c. Deployment diagram

Deployment diagram is a structure diagram, which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. Artifacts represent concrete elements in the physical world that are the result of a development process.



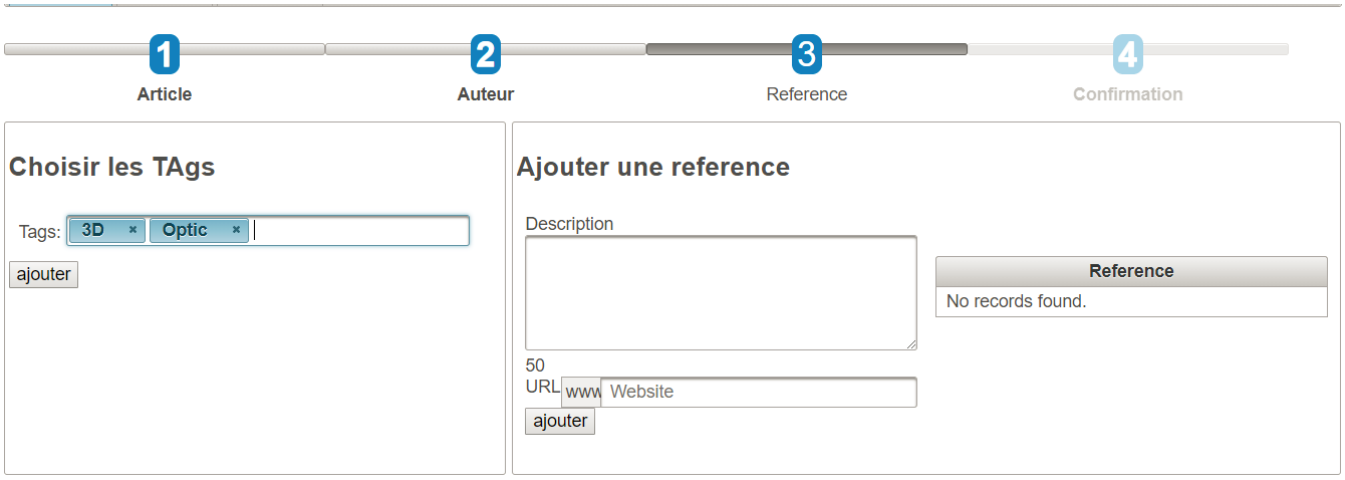
3. Codage et test

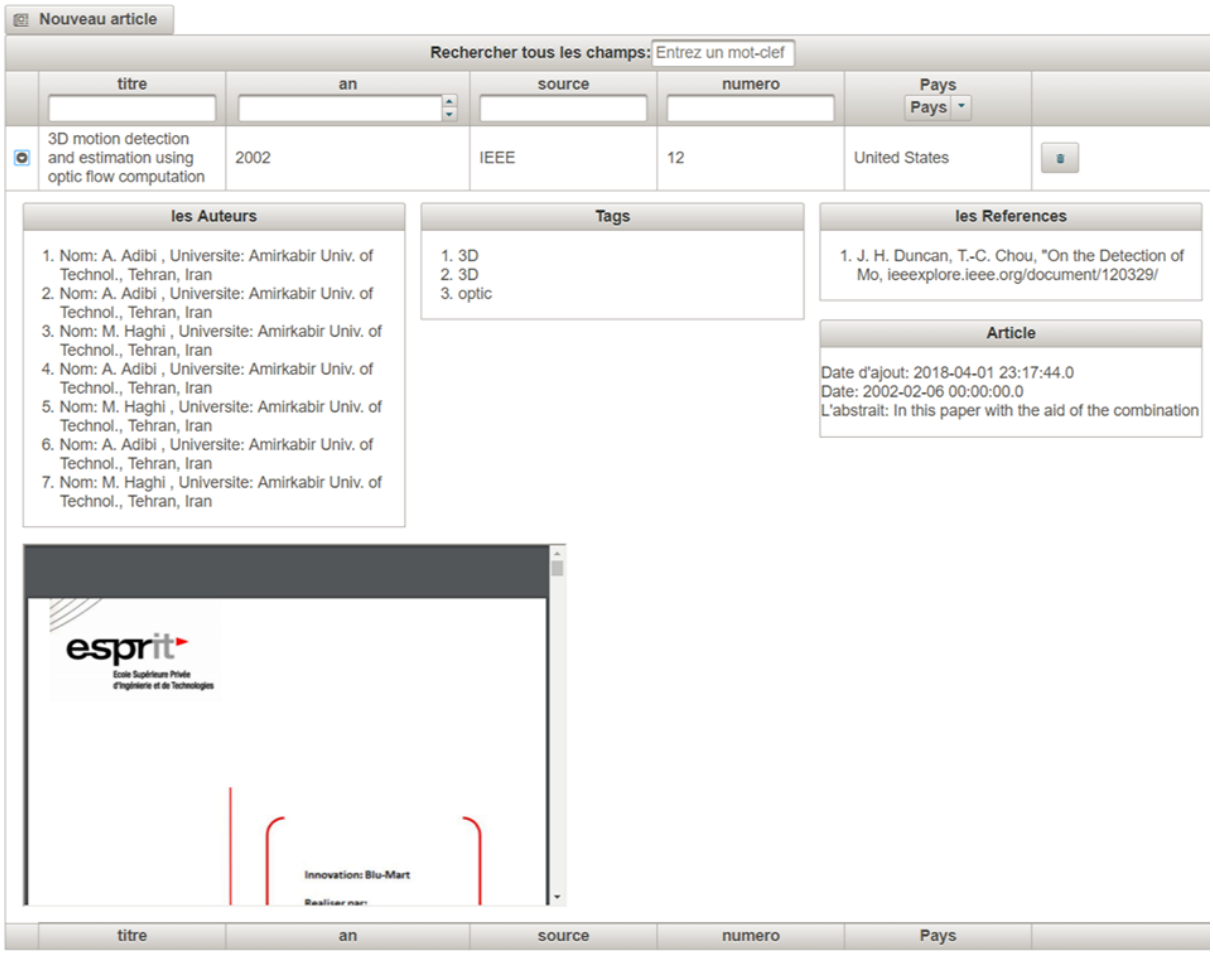
a. Interfaces of the application

Nouveau article

Rechercher tous les champs: Entrez un mot-clef

	titre	an	source	numero	Pays	
					Pays	
0	3D motion detection and estimation using optic flow computation	2002	IEEE	12	United States	
	titre	an	source	numero	Pays	





b. Implementation and integration tips

For the implementation we started with the creation of the entities and after we chose a template and integrated it so that everybody could start the development and using Git for ease of integration

c. Application Test

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.