



Capgemini

ENEDIS
L'ELECTRICITE EN RESEAU

**Internship Report for master degree in Communications and
Computer Networks Engineering**

Application Consultant: *Construction of both business and technical transformations
of projects using Java/J2EE, HTML, CSS and Angular 2 technologies for customers*

Student: Alain FOUOTSAP FOUKMENIOK

Capgemini Tutor: Andrea TODARO
andrea.todaro@capgemini.com

Academic Tutor: Giovanni MALNATI

Internship period: from July 3rd 2017 to January 2nd 2018

This report is restricted to Politecnico di Torino & Capgemini

Acknowledgements

Firstly, I would like to acknowledge and thank Philippe GOUX and Romain RAMBAUD, the two project managers whom I worked with. I appreciate the confidence you gave me.

I would also like to thank Thomas RAPILLY who proposed me to join the SYCOMORE project and Andrea TODARO for the attention you paid towards my work during the internship period.

My heartfelt gratitude also goes to Sylvie VIGNES my academic tutor at Telecom ParisTech for the advices you gave me before my internship and Giovanni MALNATI my academic tutor at Politecnico di Torino.

I cannot forget to thank the team members of “La Ruche” and SYCOMORE for their sympathy, their availability and their help which facilitated my integration. I have learned a lot working alongside you all.

Lastly, I thank my family and my friends for the unconditional support they rendered towards me during my whole academic career till date. I appreciate your wise counsel and attentive ear.

Table of contents

.....	0
Acknowledgements	1
Abstract	4
INTRODUCTION	5
CAPGEMINI PRESENTATION	6
ENEDIS PRESENTATION	7
1. LA RUCHE	8
Project context	8
1.1 CPAP	8
1.1.1 Used technologies on CPAP	9
1.1.2 Functional requirement of CPAP	9
1.1.3 My work on CPAP	10
1.2 RM Portal	14
1.2.1 Functional requirement	14
My Work on RM Portal	15
1.2.2 Technical design of RM Portal app	15
a. UML diagram	15
b. Back-end architecture	18
1.2.3 Tools of the development environment	19
a. DBMS (database management system)	19
b. ORM (Object Relational Mapping)	19
c. Maven	19
d. Spring	20
e. Application Server	20
1.2.4 Project creation and implementation	21
2. SYCOMORE	27
2.1 Project Context	27
2.2 Project challenges	28
2.2.1 Decision making tool	28
2.2.2 Internal structure	28
2.3 Internship context at SYCOMORE	29

a. Work methodology.....	30
b. Team	30
c. Scrum board	31
d. Product backlog.....	31
e. Sprint backlog.....	31
f. Item.....	31
2.4 Agile activities	32
a. Sprint	32
b. Sprint planning.....	32
c. Daily meeting.....	32
d. Backlog grooming	32
e. Demonstration.....	32
f. Retrospective.....	32
g. Kanban	33
2.5 Development practise.....	33
a. TDD Test-Driven Development.....	33
b. Pair Programming.....	33
c. Continuous integration.....	33
d. Delivery process.....	33
2.6 Development environment	33
a. Git	35
b. Angular 2.....	35
c. Jenkins	35
d. SonarQube.....	35
2.7 Integration in the team	35
a. Board organisation.....	35
b. Board 3	35
2.8 My work on SYCOMORE	36
Conclusion	40
Annexe	41
References	43
Glossary	44

Abstract

My work duties during my six months internship at Capgemini was to participate at the construction of customers projects; namely at the persistence, the business and the presentation layers.

During the first assignment, I worked on two applications of the "La Ruche". "La Ruche" is a talent incubator of Capgemini. The first application was Continuous Positive Airway Pressure (CPAP). It is an application that allows to monitor people that suffer of sleep apnoea. The second application was RM (Resource Manager) Portal, a mobile application that allows to resource managers to have a good visibility on their consultants. On both applications, I did numerous tasks that I can summarise as follows:

- Reverse engineering of the CPAP application writing the detailed technical architecture document (UML diagram).
- Design of the backend of the RM Portal application from scratch.
- Implementation of some functionalities using frameworks J2EE, spring, Hibernate and Angular2.

During the second assignment, I joined another project called SYCOMORE. SYCOMORE is a tool that Capgemini is developing for ENEDIS in order to optimize the access cost to electricity transport network. My mission on SYCOMORE was software development.

In the following pages, I will be glad to describe the different challenges encountered during the two projects and how I managed to achieve my two missions.

INTRODUCTION

I am enrolled at Politecnico di Torino in communications and computer networks engineering and Telecom ParisTech as part of an 18 months double degree agreement. I am thus in parallel pursuing a master degree in system and distributed software at Telecom ParisTech. The internship was a mandatory part of my study program.

I'm very interested by the software development and I want to improve my technical skills thus, I chose to do my six months final studies internship at Capgemini. This allows me to put in practice and in a real context what I studied at school, from requirement engineering to implementation using J2EE, SQL, Angular 2, HTML, and CSS technologies.

Fully integrated in the Capgemini team, my main mission was to satisfy the client requirements implementing good solutions using the above mentioned technologies.

During my internship, I participated to two projects: the first one being "La Ruche" and the second one, SYCOMORE.

The main benefits of this internship have been my complete integration in both teams taking responsibilities to follow up tasks, applying every day the Scrum agile framework. This allowed me to work closer to experimented developers. Thanks to knowledge sharing by my co-workers on the technologies and work methodologies, I quickly grew in competence. I also got the chance to discover the internal organisation and processes of a big company.

This experience gave me basic ideas of the computer science dedicated to the public sector and the energy sector.

My internship took place from July 3rd, 2017 to January 2nd, 2018. During the first month, I worked on the CPAP project of "La Ruche" where I started writing the technical document of the application and I implemented some functionalities using a Capgemini internal framework based on J2EE. During the second month, I built up the back end of RM Portal application of "La Ruche" from scratch. For the last four months, I worked on SYCOMORE project where I started by fixing bugs. This helped me to understand the project gradually. Then I also implemented some requirements of the application.

In the following, I am first of all going to give a brief description of the company, then continue with the presentation of my work during the two first months on the "La Ruche". After that, I am going to talk about my work on SYCOMORE and to finish, I will talk about the agility methodology used in both projects.

CAPGEMINI PRESENTATION

Founded in October 1st, 1967 in Grenoble by the businessman Serge KAMPF, Capgemini is today the first computer service company in France. Present in over 40 countries with more than 220 000 collaborators, the company is one of the leaders in the world in consulting, application services and outsourcing.

The group trades:

Consulting services

Capgemini Consulting is the group trade that is in charge of specific type of services which consist in identifying, developing and realizing the transformations required by customers in order to increase their competitiveness in the business world.

Integration systems services

Teams in charge of this part design and develop big transformation projects that covers the integration of complex systems and the development of applications.

IT proximity services

IT SOGETI (Société pour la Gestion de l'entreprise et du Traitement de l'Information) takes care of providing professionals to reply to local needs in term of infrastructures, applications, engineering, tests and operations.

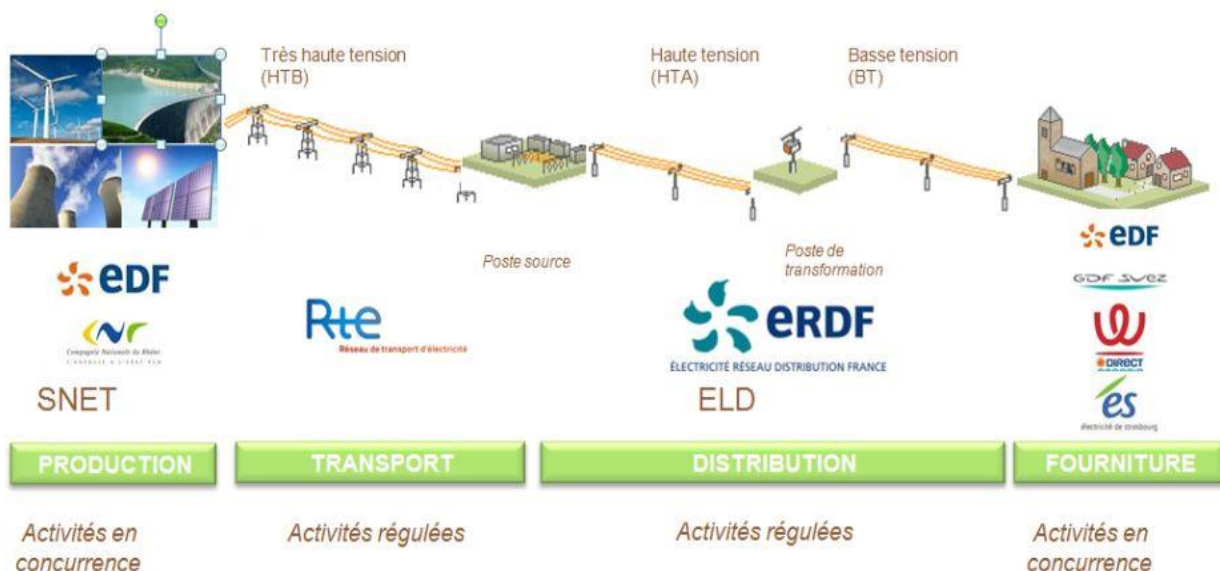
Outsourcing services

The group experts assist their customers and give them all the help they need in the management of their information system and related activities, and this for a period up to five years and above.

ENEDIS PRESENTATION

ENEDIS previously known as ERDF (Electricité Réseau Distribution France) was born on January 1st 2008. Enedis manages **95% of the electricity distribution network in continental France**. This network belongs to local authorities (French municipalities or groups of municipalities), who subcontract to Enedis as an operator through a public service delegation. Enedis thus has 2 major public service duties.

- **Service continuity and quality:** managing nearly 1.3 million km of electric lines, Enedis is responsible for continuous public electricity service. To fulfil this role, the company operates, maintains and develops the network. Enedis also invests in modernising and securing the network, particularly against extreme weather conditions.
- **Non-discriminatory access to the distribution network:** in compliance with regulations, Enedis ensures that users have transparent, objective and non-discriminatory access to the network. The company also guarantees the confidentiality of commercially sensitive information handled, having developed a **code of conduct** for this specific purpose. Adherence to this code is examined in an annual report submitted to the Commission de Régulation de l'Énergie, France's regulatory authority on energy.



Electricity transit

1. LA RUCHE

Project context

“La Ruche” is a talent incubator. Its aim is to let each newcomer integrating the company show all its technical potential and demonstrate how they interact with others. It also allows newcomers to know and learn the company’s culture, philosophy and processes.

In the “La Ruche”, there are many projects. I spent my two first months there working on two specific projects: CPAP (Continuous Positive Airway Pressure) and RM Portal.

1.1 CPAP

The aim of the project is to implement an application that allows doctors to monitor the sleep apnoea. Data we want to display come from CPAP devices connected to patients. Those data are characterised by:

- ✓ The date of the event
- ✓ The user session, a user could have many sessions in a single night.
- ✓ The event name (apnoea, obstructive, clear airway, sense awake, etc....).
- ✓ The name of the patient who is using the device.

Below is the global architecture.

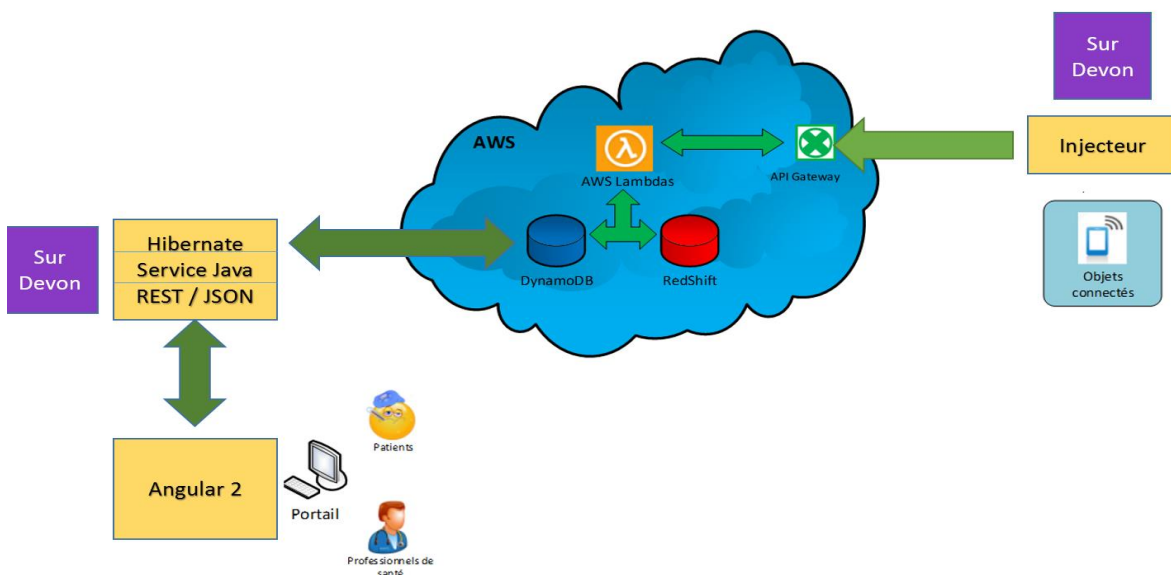


Figure 1 global architecture of CPAP

As we can see, in the first set of the global architecture, we have connected objects or sensors linked to the patient that send information in a cloud in order to have centralized information. That cloud contains databases and from there we can find the information we need. We do not care about how data are sent from the patient to the databases. We are only interested in information stored in the databases and how we can access them. However, before going to find the information in the cloud, we built our application using data stored in our local database

knowing that if the application works with a local database, it can also work with a remote database because we just have to change the URL of the database connection assuming that the data format is almost the same. We can also access to data of the cloud through a dedicated web service.

The second set of the global architecture is the cloud.

The third set is really our application. It is subdivided in two parts: the back end (where the logic of our application is implemented) and the front end (where users interact with the application).

1.1.1 used technologies on CPAP

At the server side, we used **DevOn** and at the client side, we used **Angular 2**. Let me give a brief presentation of **DevOn**. I am going to talk about **Angular 2** later on.

DevOn

DevOn is a Capgemini internal framework based on J2EE, spring and hibernate. The reason we used it is that it does many things automatically. Another reason is that it is an internal framework to the company, it allows to Capgemini to have uniform projects, so we had to know it since it is used in many other customer projects. Anyway, it simplifies the work to programmers. It is still in the development phase. At the creation of a new project, many things are already configured. For example, basic project dependencies and some class path files.

1.1.2 Functional requirements of the CPAP application

During the month I spent on the project, functionalities to implement with related IHM were:

- ✓ Register a doctor
Requirement: as a doctor, I would like to register, I click on "s'inscrire", I fill the form then I click on ok, then I receive a confirmation email and I can sign up.
- ✓ Register a patient
Requirement: as an authenticated doctor, I would like to register my new patient. I click on "inscription patient", I fill the patient information then I click on ok. The patient receives a confirmation email with its credentials.
- ✓ User authentication
Requirement: as a user (doctor or patient), I would like to sign up, I click on "se connecter", I fill my username and password, then I access my personal page.
- ✓ Get the list of patients of a doctor; a doctor should be able to see the list of all his patients
Requirement: as an authenticated doctor, I would like to visualize all my patients, I click on "liste patients" then I see the list.
- ✓ Show data of a patient; a doctor click on a specific patient to see his details: personal data (name, surname, security social number, etc.) and medical data (pressure, temperature, etc.).

1.1.3 My work on CPAP

The first thing I did on the CPAP project was to understand the functional requirements of the application. Then, I had to understand the technical architecture of the project. After browsing the back end, I designed the use case diagram and the class diagram. Thus, I did a kind of reverse engineering of the back end as we can see on pictures below.

The document I did became the DAT (technical architecture document) of the CPAP because I started it. New people that joined the project consulted it in order to understand the project and improved it gradually.

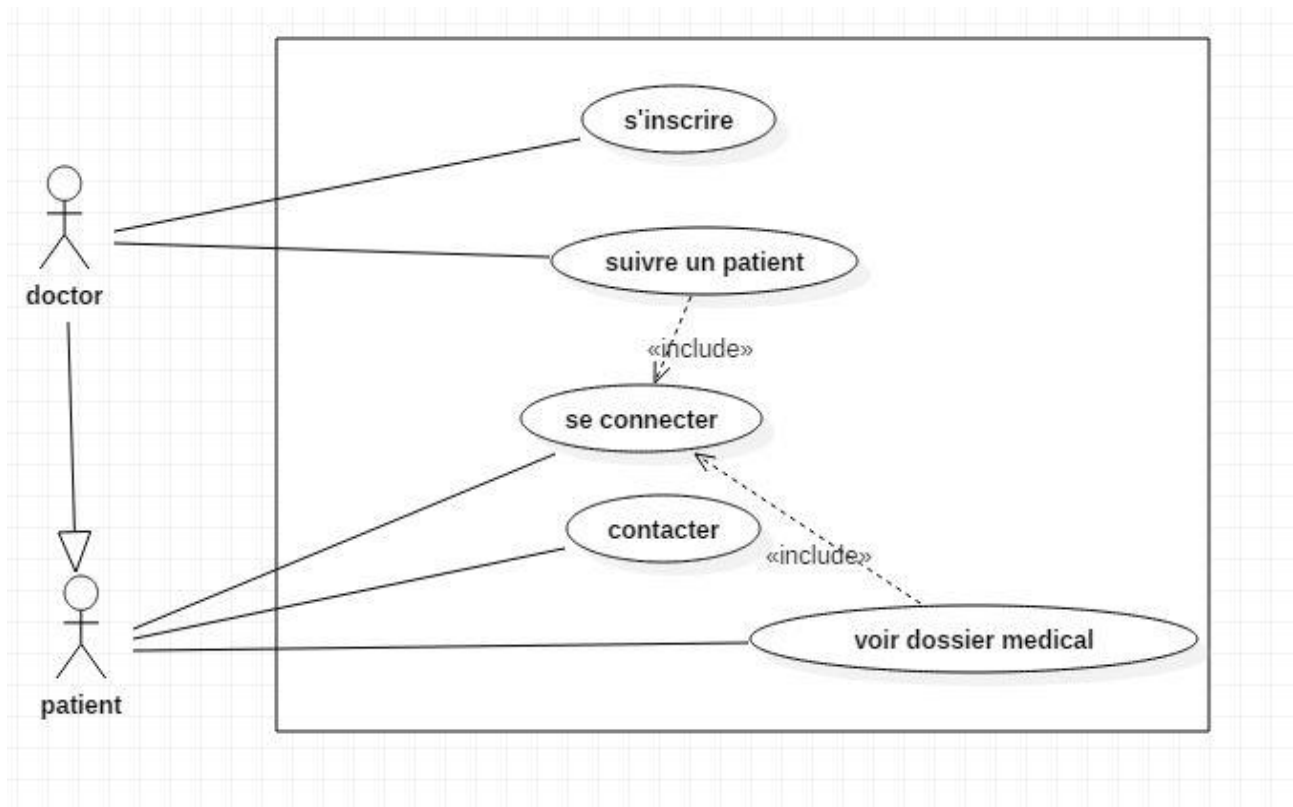


Figure 2 part of the use case diagram of CPAP

The picture above is a simple use case diagram. Here I only put use cases defined when I was working on the project. Therefore, the diagram for sure should evolve as soon as the project evolves. Let me remember that the goal of the application is to monitor the sleep apnoea, this is represented on the picture by the use case "suivre un patient". Before using the application, the user has to be registered first; this is traduced by the use case "s'inscrire". The logic requires that only doctors are allowed to register themselves. Other users (patients) are registered by doctors then they receive an email with their credentials. It is only after that they can possibly connect and access their personal page. From that moment, they can use the application accessing to offered functionalities.

After the use case diagram, I designed the class diagram. This class diagram contains only our main classes or business classes. Auxiliary classes are not present in the diagram. I did it like this in order to have a simple DAT that shows the minimum required to understand the project. It was also way to have a document easily maintainable and clear.

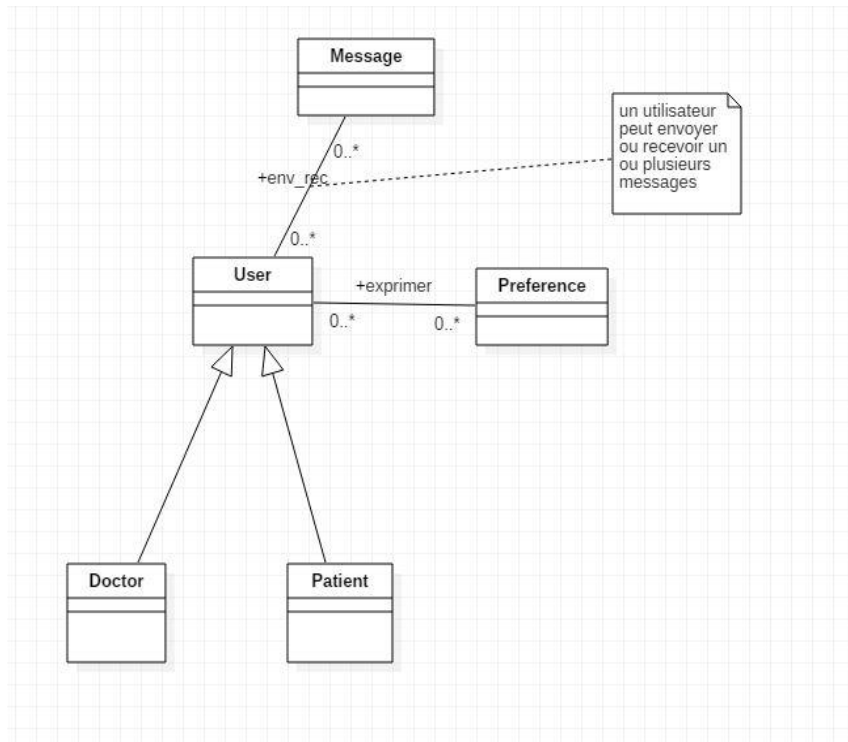


Figure 3 higher level of class diagram of CPAP

The above picture is the class diagram of a part of our application since we do not take into account medical data because like I said before, the application was at the very beginning and specifications were not completely defined. As we can see, we have a user, who can be a doctor or a patient. Then a user can send or receive more messages and can also express his preferences.

The above class diagram is very close to the MCD as we can see on the picture below. The choice to have the class diagram similar to schema of the database is justified by the framework used to implement the application. That framework gives good performances when the two models are similar because it does many things without us writing much codes.

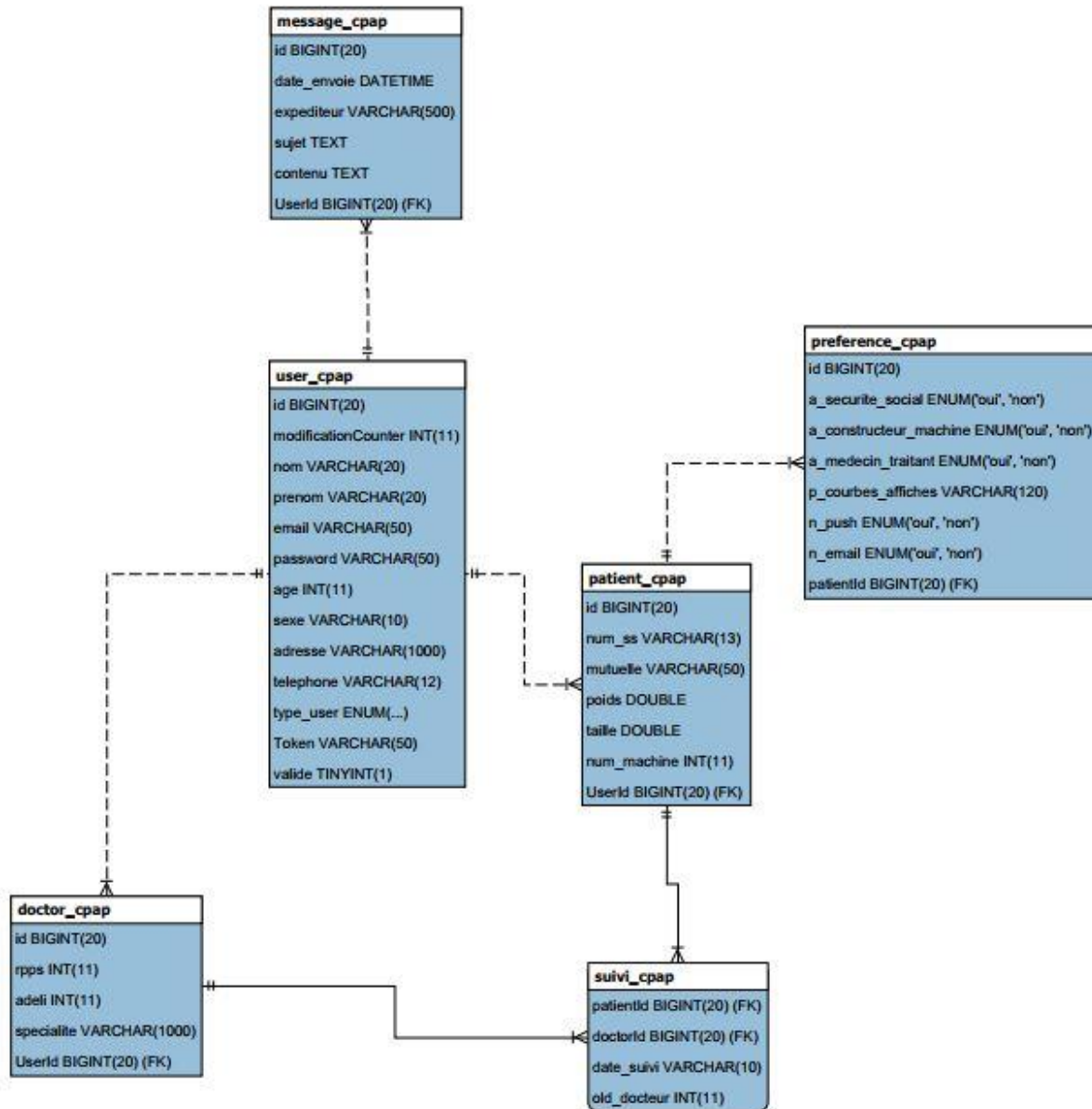


Figure 4 MCD of CPAP

The picture above is our schema in the database. In the table USER_CPAP, attributes that do not correspond to personal data of the user like modification counter, token, valid are useful in the authentication process.

Having the first specifications and designs ready, the implementation gave us what we can see on pictures below.

Results

During the month I spent on this project, I was focused on functionalities related to a patient. I implemented the CRUD operations of a patient Entity at the server side.

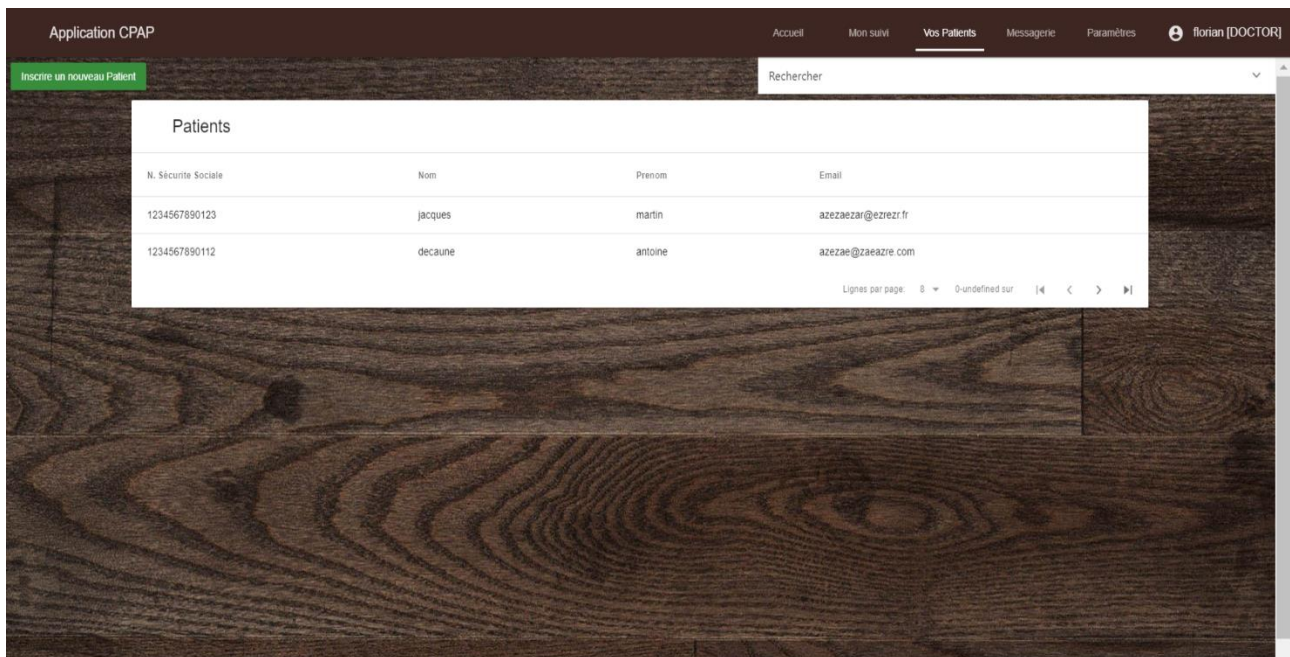


Figure 5 IHM list of patients of a given doctor

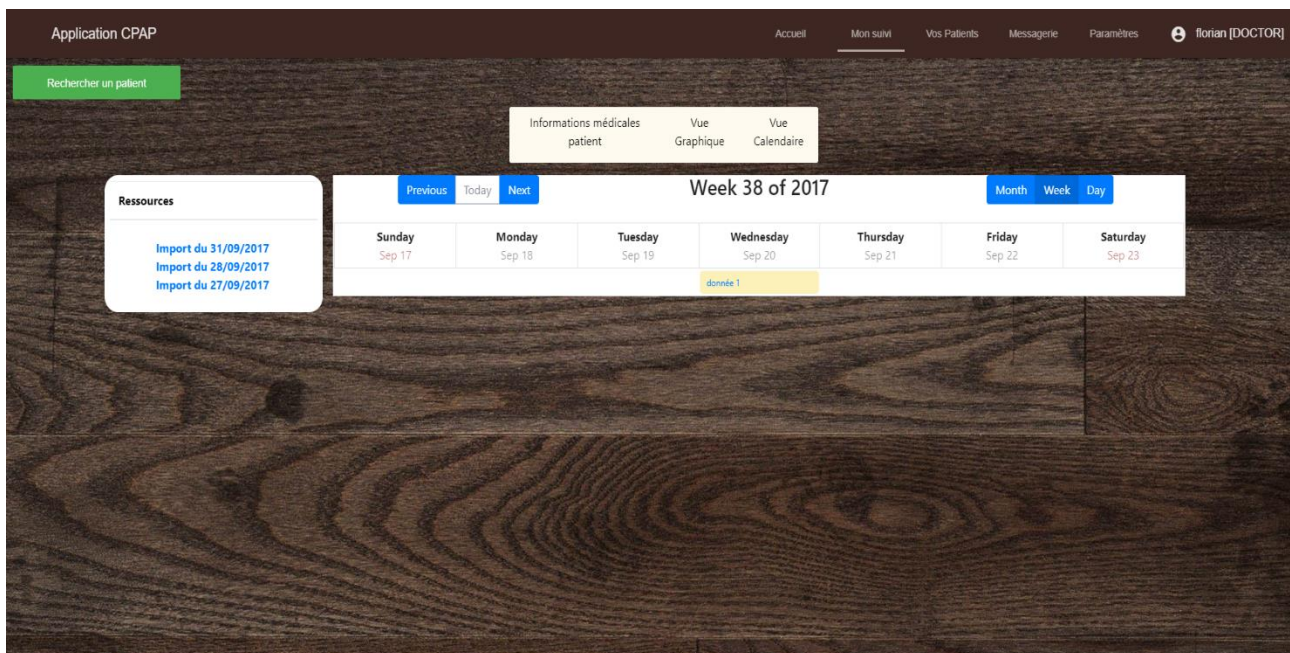


Figure 6 IHM planning of a doctor

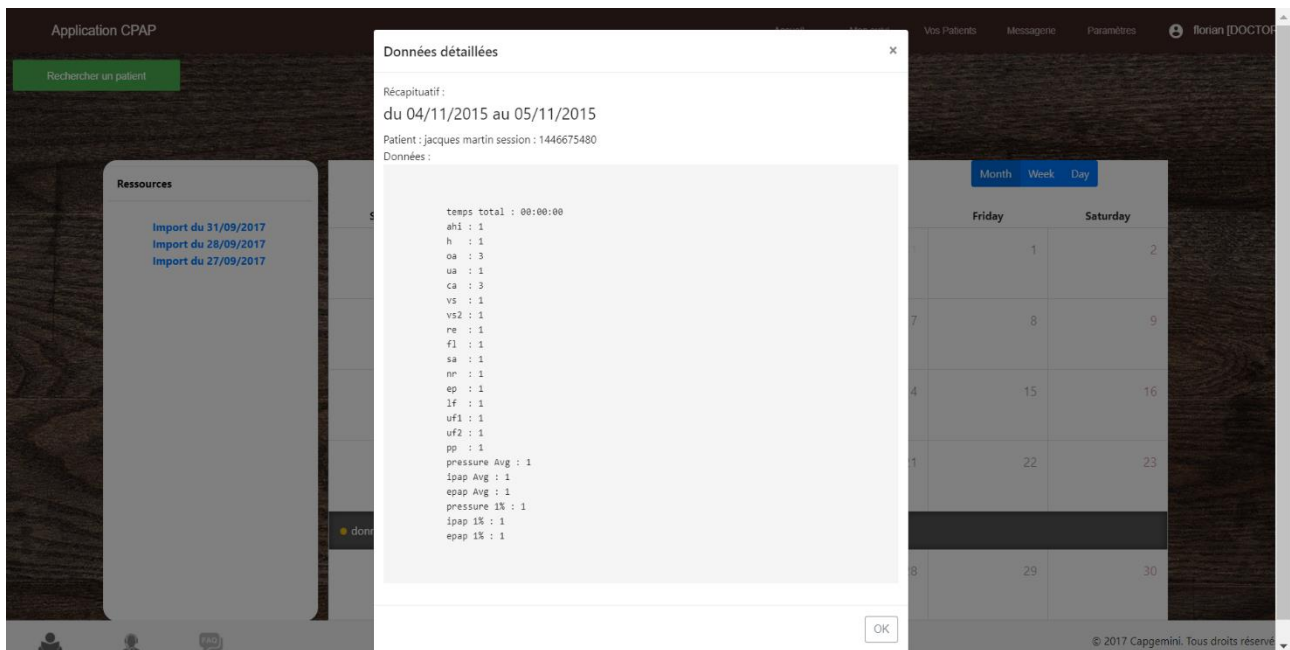


Figure 7 IHM detailed medical information of a patient

1.2 RM Portal

Resource Manager PORTAL is another project of “La Ruche” on which I worked. The goal of the project was to build an application that allows to resource managers to have a visibility on their consultants. It is a mobile application. Through the application, a resource manager can have a global view of his consultants like:

- ✓ Availability
- ✓ Skills
- ✓ Achievements and future missions or projects
- ✓ Hopes

In order to have a better understanding about everything, I am going to present the functional perimeter of the application.

1.2.1 Functional requirement

The application is divided in features as shown on the table below.

feature	title	requirement
fiche	Creation/update a consultant profile	in the app, a consultant select “edit profile”, the file appears with initial information, he edits and clicks on “OK”, information are modified
	Consultation of a resource manager profile or career manager profile	As a consultant, I click on RM, my RM or CM information appear. I can see the name, surname, email, practice, etc....

planning	a consultant can take an appointment with his RM	As a consultant, I select “take appointment with my RM”, then I can see the days and hours available of my RM. I select a free slot then I receive a confirmation by email. The planning is updated.
	a RM could update his time table or agenda	As a RM, I click on my planning I can change my availabilities.
activities		<ul style="list-style-type: none"> ✓ a consultant should be able to see activities of his PU (People Unit) ✓ update activities: a RM or PU should be able to add or remove activities ✓ a consultant should be able to visualize events of his PU
tchat		Users should be able to talk. For example, a RM could contact a consultant and vice versa.
general	A user should be able to login with username and password	As a user, I launch the app, an authentication pop-up opens, I can enter my username and password and click on “valider”. If my credentials are right, my home page appears otherwise an error message appears
	A user should be able to reinitialize his password	As a user, I am disconnected. I click on connect, then I click on “mot de pass oublié”, a new page appears, I enter my email then I click on send. I receive an email with a link to reinitialize my password. If the email is not valid, an error message appears.
search	a RM should be able to find a consultant by name or surname	I enter the name and/or surname I search; I see the list of results, I can choose the consultant that I am looking for.

search	a RM should be able to find consultants by skills	I enter skills, I search, the list of consultants who have such skills appears, I can select.
	a RM should be able to find consultant with many criteria	

When I joined this project, specifications only were defined. I participated a little bit in the specification phase but my main mission was to design and implement the back end from scratch. In the following, I am going to explain in details in the technical point of view what I did.

My Work on RM Portal app

1.2.2 Technical design of RM Portal app

Once having specifications and knowing what the application has to do, the first step was to model. I designed a high level of use case diagram. I also designed a class diagram like the ones we can see on the pictures below.

a. UML diagram

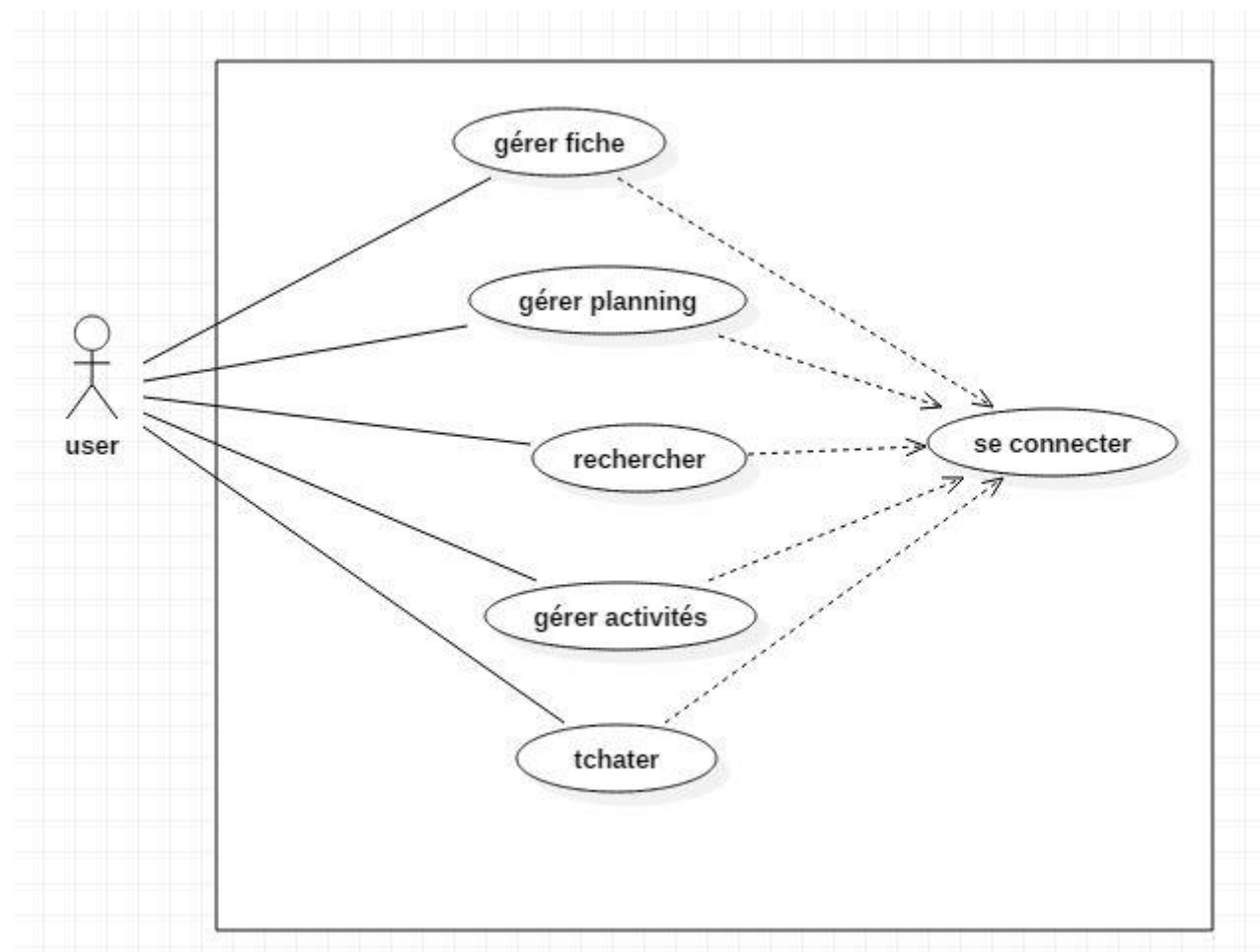


Figure 8 use case diagram of RM Portal

This diagram represents the general use case diagram. Each use case includes many other ones as shown on the example below. A user must authenticate first before executing any action. It is why all use cases depend on “se connecter”.

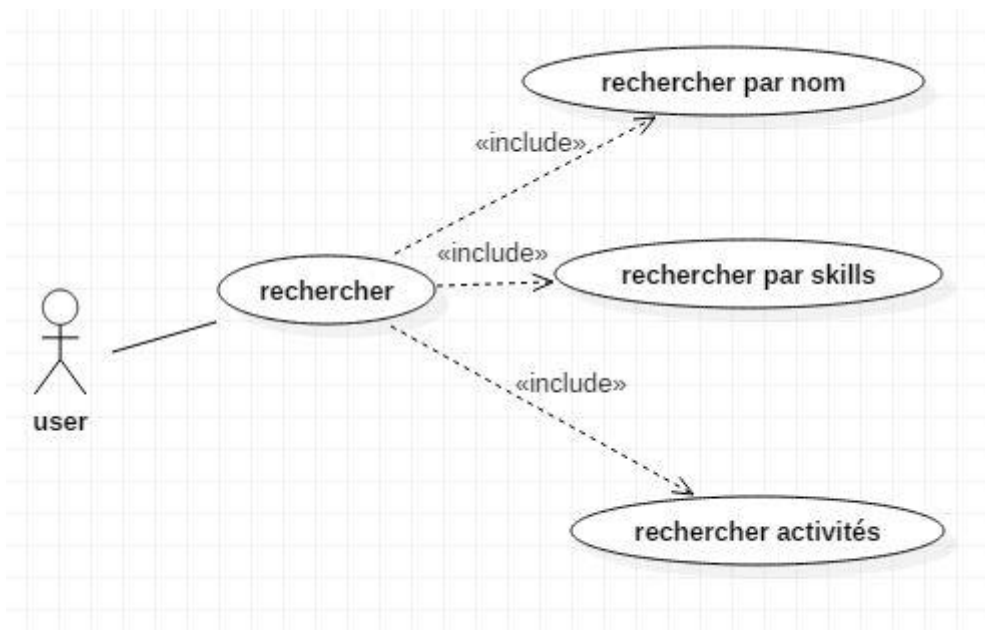


Figure 9 zoom of use case diagram “rechercher”

The picture above is a zoom of the user case “rechercher”, I did not include all sub use cases. As we can see, a user (consultant or RM) can find another user by name or by surname or by both. He can also find another one by skills. He can find activities related to his practice.

Below is the first iteration of the class diagram. Those classes are our business classes.

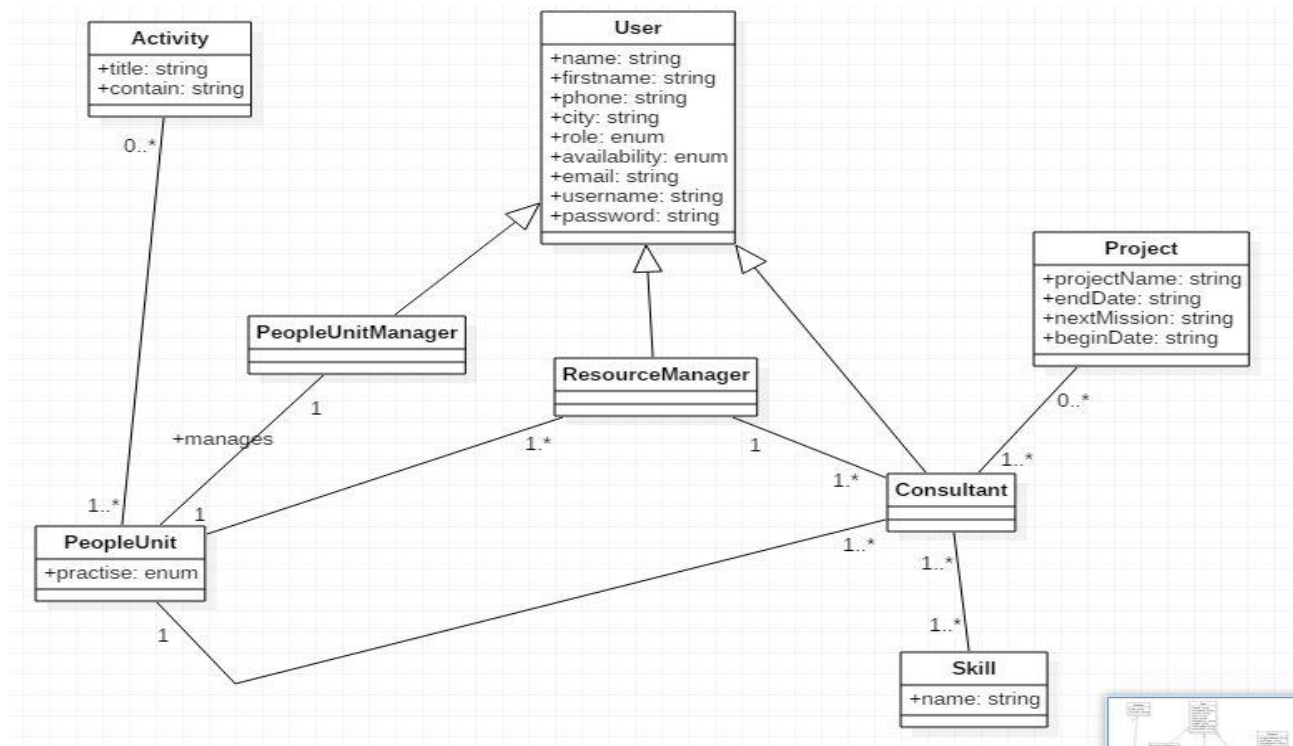


Figure 10-class diagram of RM Portal

b. Back-end architecture

As I said before, my mission on this project was to build from scratch the back end. Having the first modelling, I designed the global architecture of the server side.

Global architecture

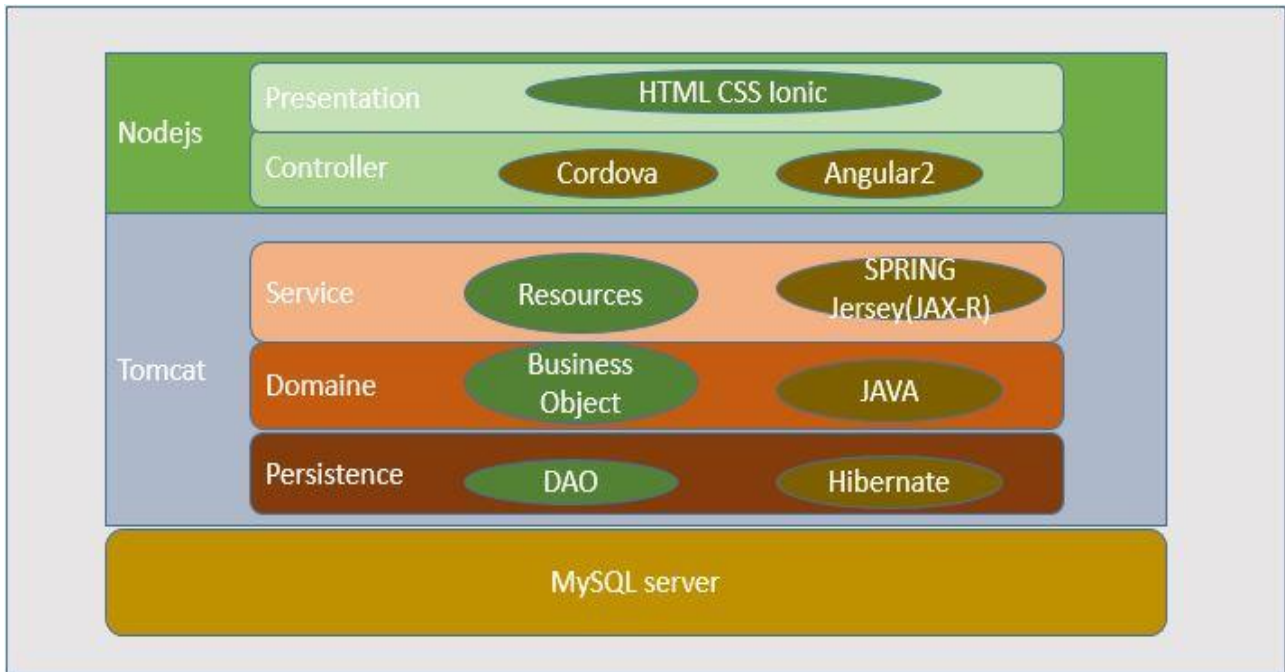


Figure 11 software architecture of RM Portal

I built an n-tiers application. Layers communicate between them through interfaces instead of concrete classes. This allows us to have more flexibility to do things in a better way. It also allows us to have weak coupling between classes.

Each layer has a well-defined task.

➤ Data Access Layer

It is in charge of transferring entities from/to database. It offers CRUD (Create, Retrieve, Update, and Delete) operations and many others if the logic of our application requires. We can write each of that method by ourselves or we can use ORM (Object Relational Mapping) frameworks.

➤ Domain Layer

It is our business logic; it is where our entities or classes are defined and the relation between them.

➤ Service Layer

It is the interface between the user and the application. This layer contains logic actions a user can do in our application. It is in charge of coordinate actions delegating other to do them. For example, it can call the DAO to perform one of the CRUD operations. Generally, classes present in this layer do not have attributes; it is why they are usually in the interface form.

➤ Presentation Layer

This layer transforms user requests in operations in the service layer then encapsulates results to show to the user. Each controller is mapped to an URL.

After that, I studied a little bit seriously tools that are used in the web application development in general and in particular when J2EE is the development platform in order to prepare the development environment.

1.2.3 Tools of the development environment

Many tools are useful in software development. I studied in details those tools before doing the choice. Let us start from bottom to top.

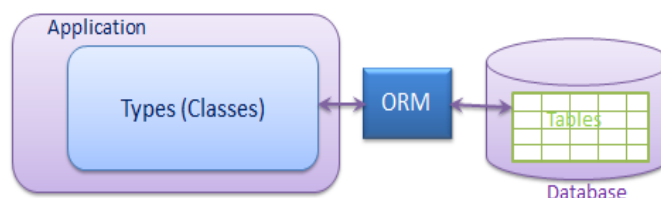
a. DBMS (database management system)

The database management system we used is MySQL. It is the suited one for our application because it is lightweight and fast in reading data but the main reason is that, we do not have to manage a huge amount of data in our application.



b. ORM (Object Relational Mapping)

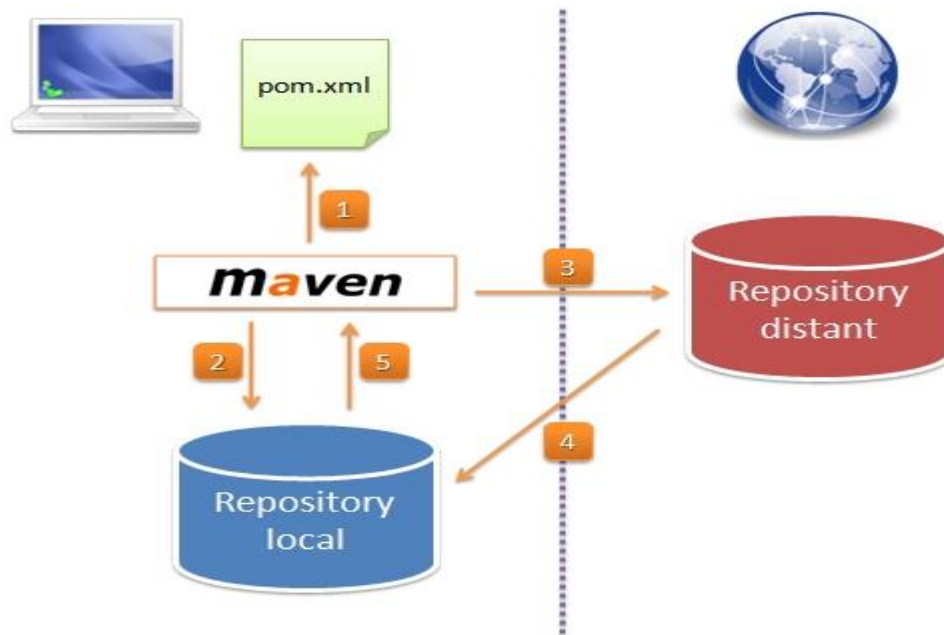
ORM are very useful because they avoid us to write a lot of SQL code. They do the mapping between classes and tables in the database. There are three main JAVA ORM: MyBatis, TopLink and Hibernate. For this project, I chose to use Hibernate, which is a good implementation of Java Persistence API. It is simple to configure, it is also suited for small project where we do not have big queries to do. The interrogation language is HQL (Hibernate Query Language). It is similar to SQL; the only difference is that Hibernate acts on entities while SQL queries act on tables.



The main drawback of using ORM is that, we can obtain a not well-designed database. We can also have not optimum queries since we do not know the internal implementation of ORM. I decided to use it because our project is small, so it is easy to keep control on what is happening.

c. Maven

Maven is a very useful tool in software development. It is a support tool for the compilation and the software maintenance. A project built with eclipse is not easily portable out of eclipse and if our project have dependencies, it becomes strongly maintainable. It is where Maven enters in action. It is a tool for an automatic compilation, it is a kind of an evolved make file. Maven dependencies are contained in a POM (Project Object Model) file. That POM file describes the structure of the project. There we can also find the project's name and version. So, at the compilation, Maven automatically download from its repository to our local repository all dependencies.



Some maven useful commands are:

- ✓ maven clean: delete all file generated by maven
- ✓ maven install: build the project
- ✓ maven update: update project files generated by maven

d. Spring

Another important framework we used in our project is spring. Spring is a framework that avoid us to have "spaghetti code" i.e. classes that slot each other or classes that have a strong coupling between them. Spring does what is called inversion of control. In basic words we do not want to instantiate our classes with "new" by ourselves, we use a spring annotation and when we need an instance of our class, spring does it for us. This process is called auto wiring i.e implicit dependency injection. So, we focus on the business logic code delegating technical code to Spring.

e. Application Server

Since our application is a web application, we need an application server to run it. There are many application servers such as WebLogic, jetty and tomcat. For our application, I chose tomcat because it lightweight and is fast in the deployment phase.



To run our application, we have many ways to do it. For example, we can create a war (web archive) and then deploy it in a servlet container or an application server. However, this requires to write by ourselves configuration files like web.xml file. We can avoid this by using Spring Boot.

➤ SPRING BOOT



Spring Boot is a framework that allows easily developing and running spring applications. We just have to configure the class path in general the application properties file. Then based on those configurations spring boot builds the environment (context). The maven plugin boot allows to generate a unique JAR from all JAR of the project and this is good because it makes things simpler. Thus, the main method becomes the entry point whatever the kind of application: standalone application or web application. In our case since it is a web application, using annotation **@SpringBootApplication** which is equivalent to **@Configuration**, **@ComponentScan**, **@EnableAutoConfiguration** together, spring boot understand, builds the suited context for our application and runs an embedded tomcat.

1.2.4 Project creation

I created a spring boot application because it avoided us to do manually a lot of configurations. To create a spring boot application, we create a maven project and we add right dependencies. There is also an easy way to create a spring boot application. It just consists in going on their official web site, select project dependencies and download the maven project generated already configured.

The image shows the Spring Initializr web interface. At the top, it says "SPRING INITIALIZR bootstrap your application now". Below this, there's a form to "Generate a" with a dropdown menu set to "Maven Project", "with" a dropdown menu set to "Java", and "and Spring Boot" with a dropdown menu set to "1.5.8". Underneath, there are two main sections: "Project Metadata" and "Dependencies". In "Project Metadata", there's a "Group" field with the value "com.cappgemini.hive.rmportal" and an "Artifact" field with the value "rmportal". In the "Dependencies" section, there's a "Search for dependencies" field with the text "Web, Security, JPA, Actuator, Devtools...". Below this, there's a "Selected Dependencies" section showing four green buttons: "Jersey (JAX-RS)", "Security", "JPA", and "DevTools", each with a small 'x' icon to its right. At the bottom center, there's a large green button that says "Generate Project" with a keyboard shortcut "alt + ⌘" next to it.

Figure 12 set up a spring boot project from the official web site

The picture above is an example for our application. I selected four dependencies at the beginning that for sure I was going to use. The spring security dependency since we perform the authentication in our application, Jersey for rest web services, JPA for the persistence. Let me specify that I gradually added other dependencies I needed when I was developing because at the beginning we do not know all dependencies we needed. It is also possible to remove some ones if we note that they are useless.

Package diagram

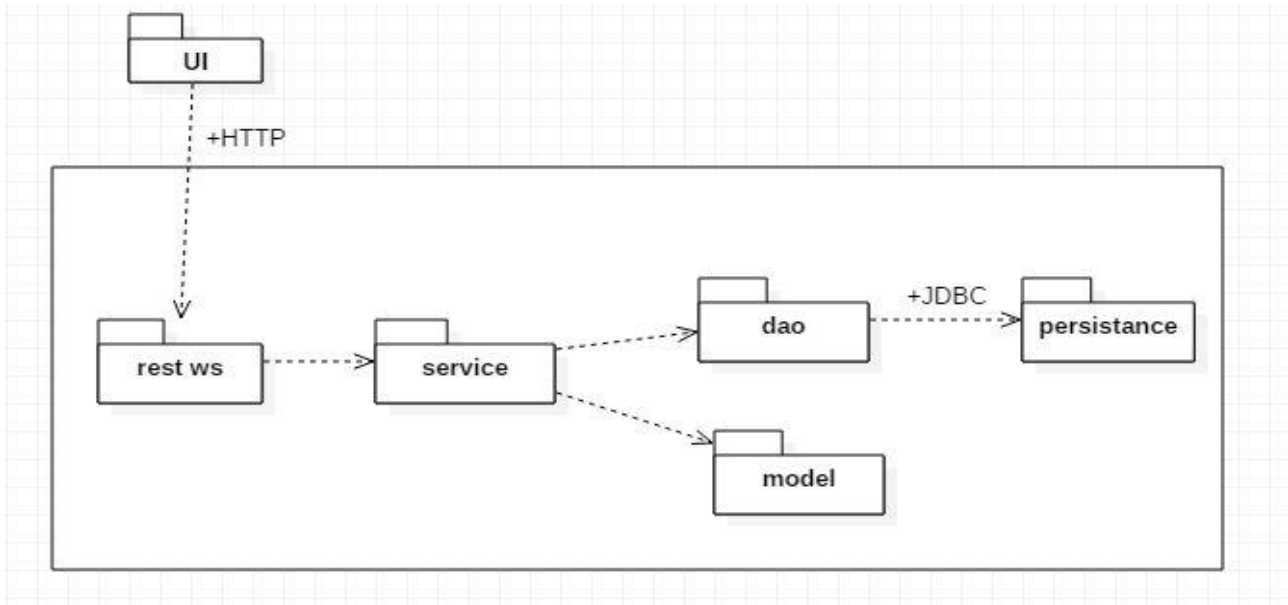


Figure 13-package diagram RM Portal

UI is the User Interface, it is the front end of our application, and it is a mobile application. I am going to talk a little bit about it later on.

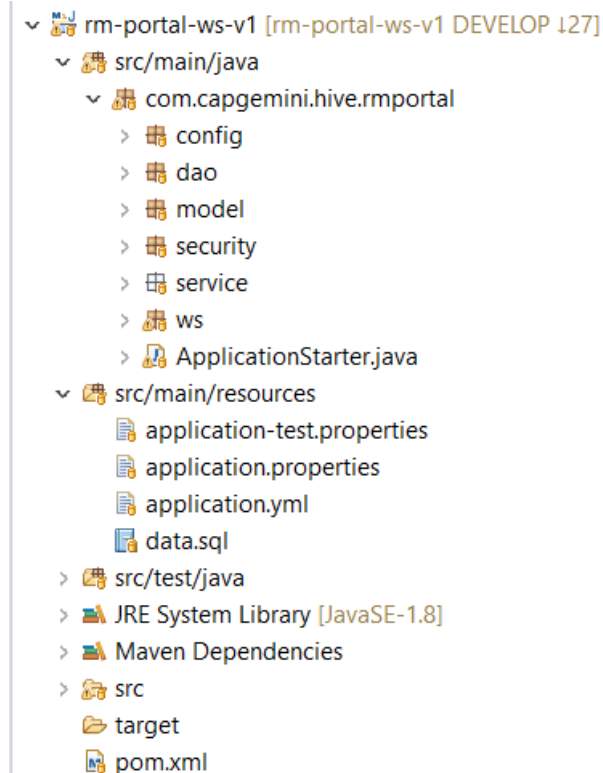


Figure 14 view of our project in eclipse

This picture is the architecture of our project in eclipse. In addition to packages shown on the package diagram, there are two more packages which are “config” and “security”. It is where I put some configurations and the security chain respectively. They contains also utilities classes. In resources, we find configurations of the connection to the database, the ORM properties and data we want to insert in our database contained in the file (data.sql).

Let us see in details what each package contains.

Package model

This package contains business classes of our application. Let us look at an example

```

20 * Copyright © 2017 Capgemini. All rights reserved.
7 package com.capgemini.hive.rmportal.model;
8 import java.io.Serializable;
25
26 /**
27  * @author afouotsa
28  * @since 25 août 2017
29  */
30 @Entity
31 @Table(name = "user")
32 public class User implements Serializable {
33     private static final long serialVersionUID = -8544566433314253358L;
34     @Id
35     @GeneratedValue(strategy = GenerationType.AUTO)
36     @Column(name = "user_id")
37     private Long id;
38
39     @Column(name = "user_lastname")
40     private String lastname;
41
42     @Column(name = "user_firstname")
43     private String firstname;
44
45     @Column(name = "user_phone")
46     private String phone;
47
48     @Column(name = "user_email")
49     private String email;
50
51     @Column(name = "username", unique = true)
52     private String username;
53     @Column(name = "password")
54     private String password;
55     @ManyToMany(fetch = FetchType.EAGER, cascade = CascadeType.ALL)
56     @JoinTable(name = "user_role", joinColumns = {
57         @JoinColumn(name = "user_id", referencedColumnName = "user_id") }, inverseJoinColumns = {
58             @JoinColumn(name = "role_id", referencedColumnName = "role_id") })
59     private List<Role> roles;

```

Figure 15 example of business class in the entity package

Each entity has to implement the interface serializable; it means that, an instance of this class can be persisted in a database. A class that must be persisted is annotated with **@Entity**. The annotation **@Table** allows to define the corresponding table in database of this entity. **@Column** defines the corresponding column name in the table of each attribute. We have a relation between a user and role that is bidirectional and expressed through a collection. It is why we use the annotation **@ManyToMany**; there is a juncture or association table between these two entities. We will use the field role (Consultant, RM, CM, PUM, etc.) to know which view we will

show. The ORM uses those annotations to generate tables we will use and thus, avoid us a lot of work (writing by ourselves scripts to generate tables of the database).

Fetch type allows to say if we want to fetch also user roles when we are fetching a user or not.

Cascade type defines if CRUD operations performed on a user entity have to be performed also on the corresponding role entity. We can select which CRUD operation has to be performed on both entities at the same time through the cascade property.

Package DAO

This package contains interfaces through, which we access to database.

```
2+ * Copyright © 2017 Capgemini. All rights reserved.
7 package com.capgemini.hive.rmportal.dao;
8
9+ import org.springframework.data.jpa.repository.JpaRepository;
12
13+ /**
14 * {#link UserDao}: The Interface UserDao.
15 *
16 * @since 25 août 2017
17 */
18 public interface UserDao extends JpaRepository<User, Long> {
19
20     User findByUsername(String username);
21 }
22
```

Figure 16 example of a Dao interface

We extend the **JpaRepository** that is a spring repository. It performs CRUD operations. It offers a big advantage, it avoids us to write query by ourselves unless it is complicated. In that case, we use the annotation **@Query (our query expression)**. It is able to automatically generate queries if we use an appropriate semantic. For example, it generates the query corresponding to find a user by his username when that method is called based only on the name of that method. It uses the camel case to split the name of the method and generates the corresponding query.

Package Service

Here we implement the business logic.

```

20 * Copyright © 2017 Capgemini. All rights reserved.
7 package com.capgemini.hive.rmportal.service.impl;
8
9 import java.util.List;
21
22 /**
23  * this class implements UserService interface {@link }: we still have to manage
24  * exceptions!!!! for the moment we assume every things is good.
25  *
26  * @author afouotsa
27  * @since 25 août 2017
28  */
29
30 @Service
31 @Transactional
32 public class UserServiceImpl implements IUserService {
33     private static final Logger LOGGER = LoggerFactory.getLogger(UserServiceImpl.class);
34
35     @Autowired
36     private UserDao userDao;
37

```

Classes contained here are annotated with **@Service** to say that, the class offers services. We annotate also with **@Transactional**. It means that, independently on how is the database, each transaction has to succeed (in this case we commit) or fail (in this case we rollback). Here we have an example of dependency injection. In fact, the user service class needs an instance of user Dao class, instead of instantiate in a constructor by ourselves, we use auto wiring trough the annotation **@Autowired**, spring will do it for us.

Package WS

This package contains resources that can be consumed by HTTP clients.

```

20 * Copyright © 2017 Capgemini. All rights reserved.
7 package com.capgemini.hive.rmportal.ws;
8
9 import java.util.List;
21
22 /**
23  * implementation of user's web services, we still have to manage exceptions
24  * {@link }:
25  *
26  * @author afouotsa
27  * @since 25 août 2017
28  */
29 @RestController
30 @CrossOrigin
31 @RequestMapping(value = "/rmApp/users")
32 public class UserWebService {
33
34     @Autowired
35     private IUserService userService;
36

```

Here we have another example of dependency injection since web services call offered services. We have another advantage of spring boot here that exposes in an easy way web services using the annotation **@RestController**. Therefore, we do not need to create a class where we declare our resources any more (resource config class to expose web services). **@RequestMapping** maps the resource on an URL. **@CrossOrigin** is for security reasons.

Package Security

In this package, there are classes used in the authentication process like filters, security configuration, user details service, etc. I am not going to enter in the details otherwise it becomes too long.

everything I presented above was used to implement the first functionality required: the user authentication because a user has to connect first before using the application. The authentication technology used is the JWT (JSON Web Token) authentication. The basic concept of JWT is that, the server generates a token for each connected client. At each Http request, the client associates that token to the request. When the server receives the request, it checks if the token is valid. If yes, it provides the resource to the client otherwise it refuses.

The advantage of using JWT is that, it is simple to implement. Another one is that, there is not overload at the server side since there is not the creation of the user session. Only the user has the charge to store the token in its local storage.

Let talk a little bit about the front end. I did not do anything on the client side; it was not a part of my job.

Front end

The application is a mobile application. We decided to use the ionic framework. Ionic is an HTML5 mobile app development framework targeted at building hybrid mobile apps. Hybrid apps are essentially small websites running in a browser shell in an app that have access to the native platform layer. Hybrid apps have many benefits over pure native apps, specifically in terms of platform support, speed of development, and access to 3rd party code.

Think of Ionic as the front-end UI framework that handles all of the look and feel and UI interactions your app needs in order to be compelling. Kind of like “Bootstrap for Native,” but with support for a broad range of common native mobile components, slick animations, and beautiful design.

Unlike a responsive framework, Ionic comes with very native-styled mobile UI elements and layouts that you would get with a native SDK on iOS or Android but did not really exist before on the web. Ionic also gives you some opinionated but powerful ways to build mobile applications that eclipse existing HTML5 development frameworks.

Since Ionic is an HTML5 framework, it needs a native wrapper like Cordova or Phone Gap in order to run as a native app. We strongly recommend using Cordova proper for your apps, and the Ionic tools will use Cordova underneath. When I finished the implementation of the authentication at

the server side, my teammates also finished the related IHM. Pictures below represent the demo.

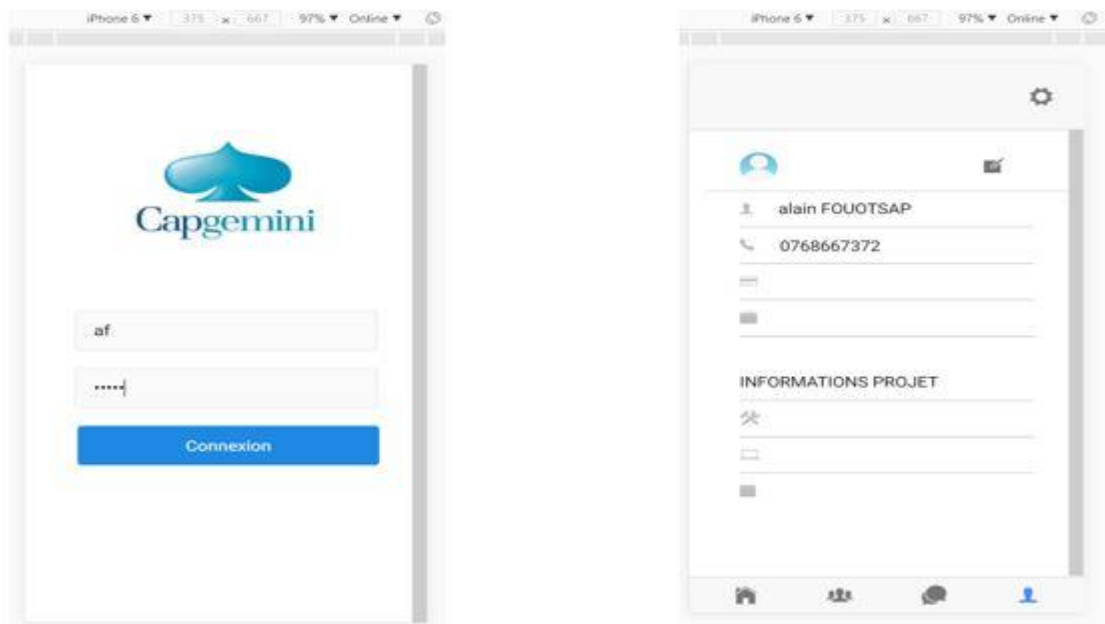


Figure 17-user authentication RM Portal

This concludes the two first months that I spent on “la ruche”.

After 2 months in intern at Capgemini, I grew up in competence on the technical point of view then I joined a big customer project called SYCOMORE (SYstème COntrole Mesures et Optimisation REseau).

2. SYCOMORE

2.1 Project context

ENEDIS has been created on July 1st 2008 after the splitting between EDF distribution activities and production, selling, transport activities of the electricity. ENEDIS is the new name of ERDF.

ENEDIS is the administrator of the distribution network in France. It has to guarantee the access to the network in an objective way, with transparency and without any discrimination. Its role is to transport electricity from the public network managed by RTE to consumers (customers, companies and local communities).

The value chain of the energy trade is shown on the picture below:

Les différents flux financiers

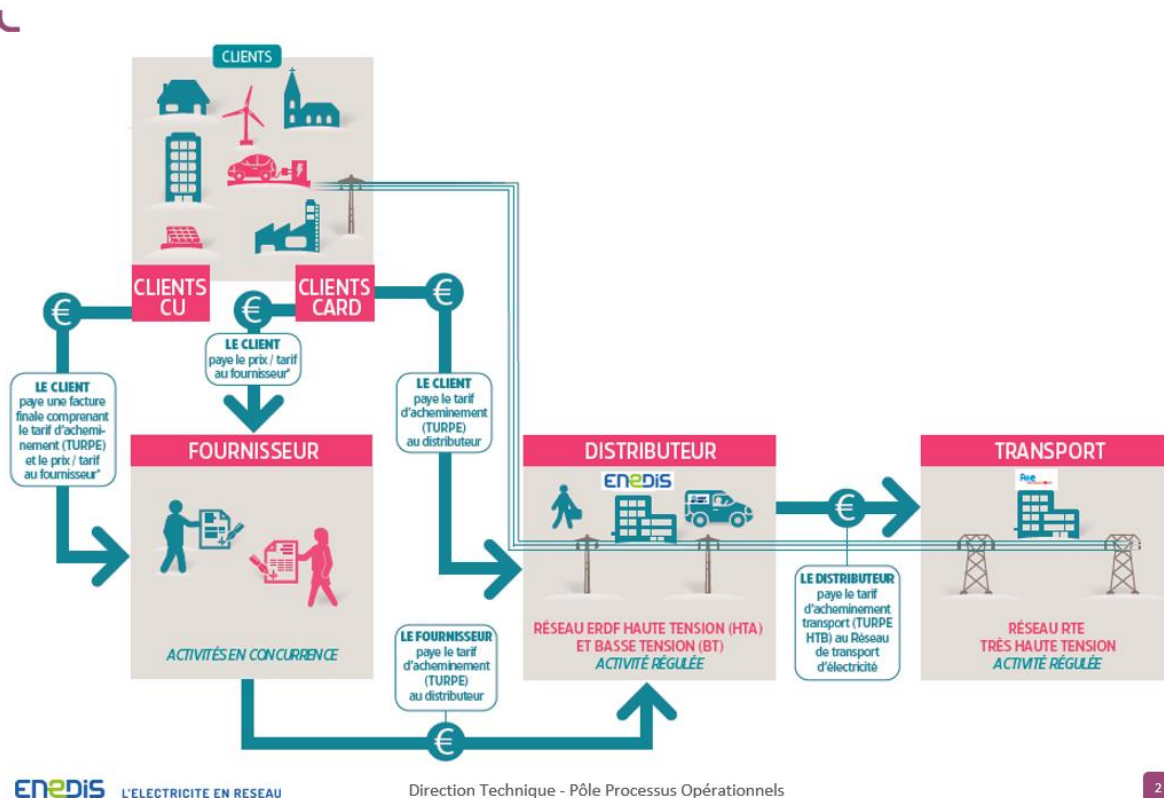


Figure 18-value chain of the electricity trade

2.2 Project challenges

The cost bound to RPT access represents an important spending source of money for ENEDIS (more than 3 billion euro per year). The control of that spending source is a big challenge for ENEDIS. It is why SYCOMORE was born. The goal is to allow to ENEDIS to control and to master the cost of the access to public transport network. In fact, the Energy Regulation Commission (CRE) defines those costs in Electricity Public Network Usage Price (TURPE). That money allows to pay network managers. The price setting between RTE and ENEDIS is done through the Access Contract to the Transport Network (CART).

In practice, RTE (transport network) charges ENEDIS (distribution network) at each interface point between the two networks. Those connection points correspond to source posts. Charges are calculated respect to the load that flows through each of those connection points and the power that has been subscribed at each point.

2.2.1 Decision making tool

The SYCOMORE application can be seen as a decision making tool. In fact, it performs calculations in order to find the best power subscription i.e. the optimum one. Thanks to indicators returned by the application, the ENEDIS agent makes his subscription choice.

Project modules

The project has been divided in 3 main modules. The goal is to cover the maximum functionalities in order to give to users the possibility to do all their operations on the same application.

- ✚ Optimisation of RPT access cost: to realize studies allowing to identify and to anticipate subscription elements. Those studies also allow ENEDIS to anticipate the right usage of the electricity public transport network.
- ✚ RPT access bill control: it allows to check the conformity between the RPT access bill given by RTE and the subscription elements done by ENEDIS, the consummation included. Bills are then injected into SYCOMORE information system for subsequent analysis.
- ✚ Network conduct: to product useful studies for the network exploitation aiming to optimize the network usage through the charge repartition between connexion points. Those studies can be done thanks to graphs provided by the application.

In additional, SYCOMORE is also the referential of RTE connection points and related load curves. Information about maintenance of the network are also consultable in SYCOMORE.

2.2.2 Internal structure

The project has been started in June 2013 in the classical development methodology called V cycle. The miscommunication and the disorganisation between the business, the MOA and the MOE conducted to a delivery not covering all requirements after 2 years.

In 2016, the DT and the DSI took a common decision to restart the project again this time using the agile methodology. This method consists in putting in the same team all stakeholders of the project. Thus the business, the MOA and the MOE directly communicate in a transparency way. Here we have a picture of the team.

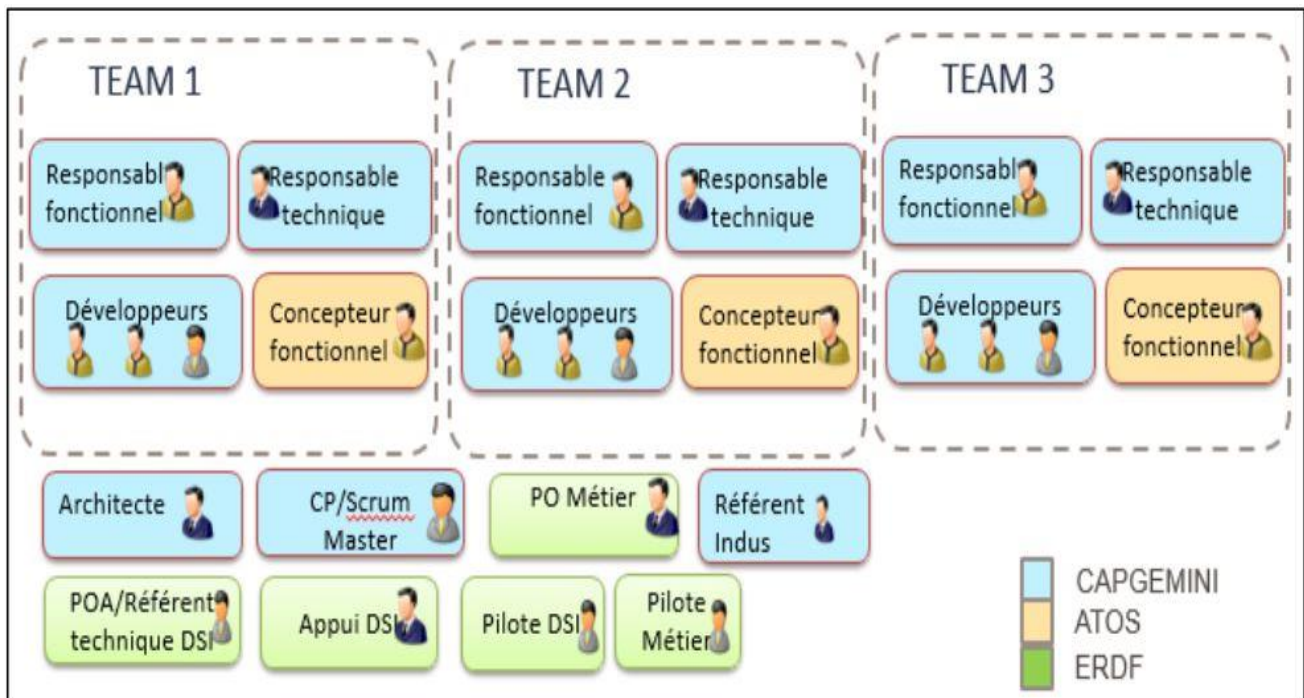


Figure 19 SYCOMORE team

The approach is to move gradually in small iterations. Business requirements are divided in unit tasks called User Story and prioritized in a backlog. The product owner defines a set of requirements to satisfy in one development cycle called sprint.

Before each iteration, tasks are challenged by all the team in order to find the best trade-off to solve them. During the sprint, the progress of tasks is followed daily to avoid blockings and failures.

The method is beneficial for both sides. From the business point of view, the PO (product owner) can understand the difficulty of each task; this helps him to prioritize requirements. Short cycles (sprint) allow to quickly see the results and to avoid gaps between what is produced and what should be produced. From the SI point of view, developers can quickly understand and have precisions about the business.

2.3 Internship context at SYCOMORE

I joined the technical team of SYCOMORE on September 11th as intern developer. I have not got a particular personal work. I integrated the third board that is in charge of the bill injection into SYCOMORE SI and curve module. I started following the directives of the scrum methodology.

The technical architecture of SYCOMORE is shown on the picture below:

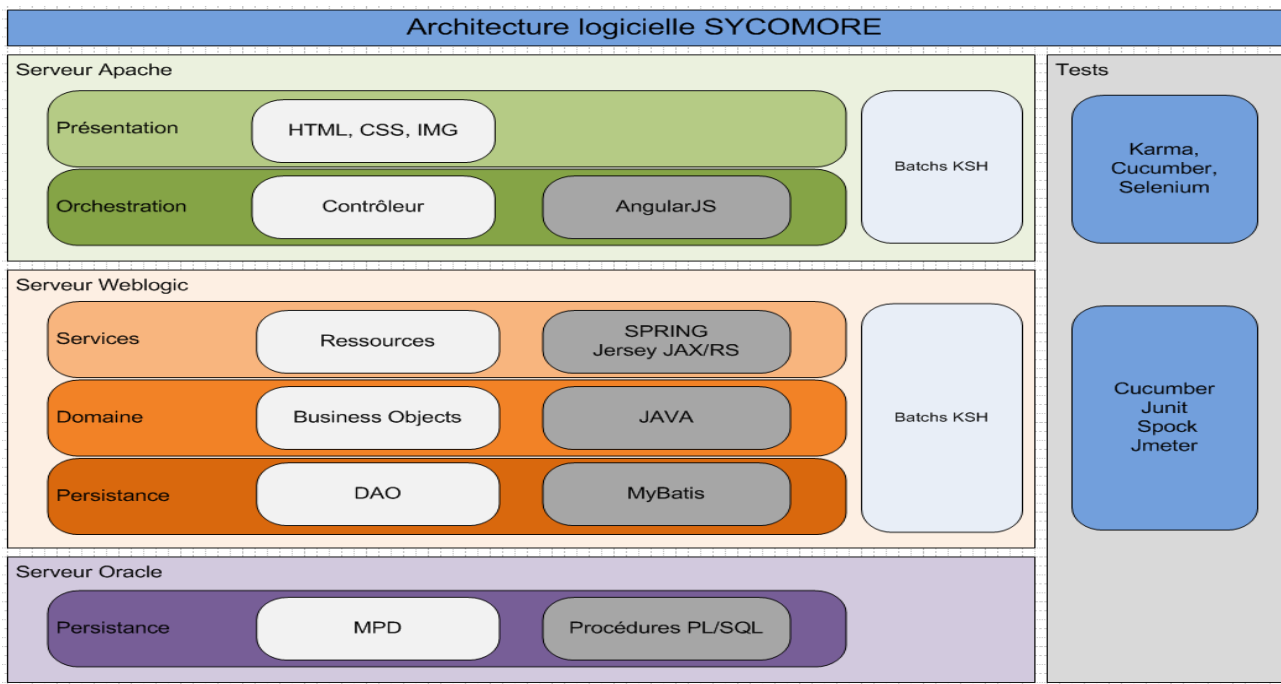


Figure 20 SYCOMORE architecture

Now we are progressively moving from angular JS to Angular 2 and in a near future, we are going to move from WebLogic server to Tomcat server.

a. Work methodology

Each team applies the agile methodology. It is a way to manage software development that aims to satisfy real functional requirements instead of respecting the contract of the development. It allows the adapting to changes and recommends five principles:

- ✓ The interaction between people
- ✓ A software that works, delivered by successive iterations
- ✓ The inclusion of the client in the project delivery
- ✓ The reactivity against incidents and changes
- ✓ The presence of the team at each level, from the design to the final production

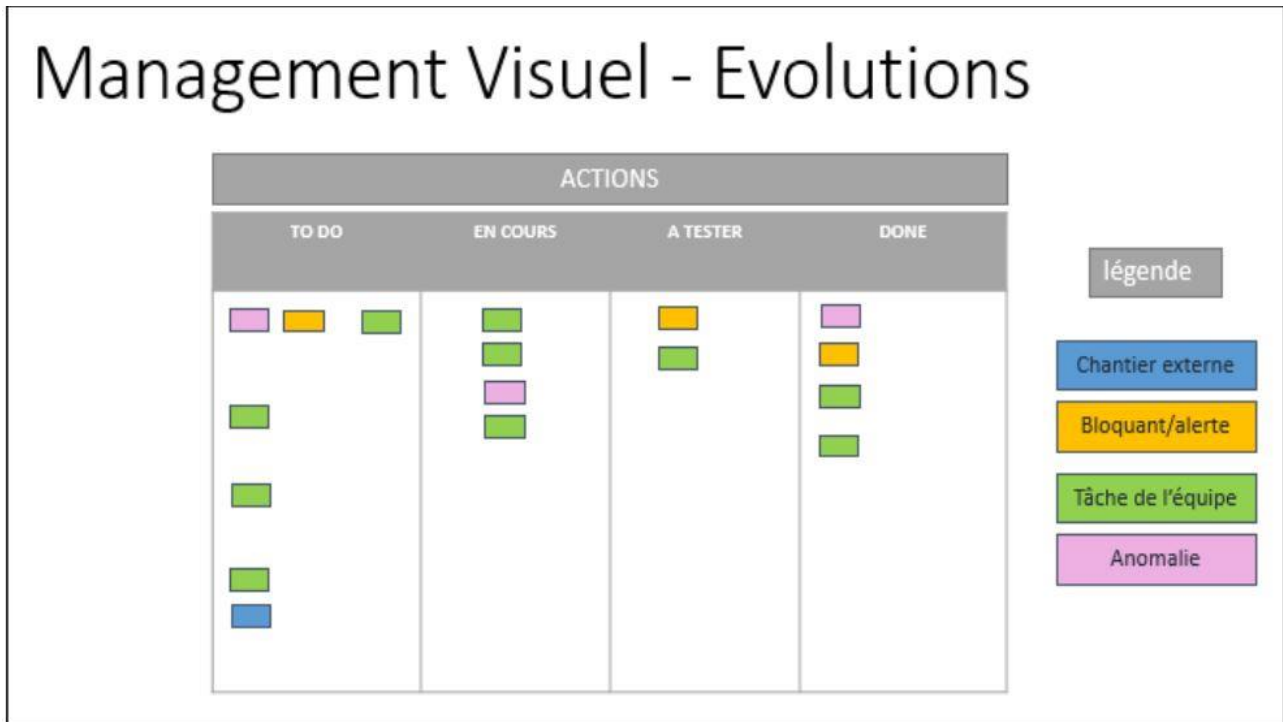
Agile methodologies used in SYCOMORE are Scrum, Kanban and XP. All three are complementary because Scrum and Kanban guide the organisation of the project while XP covers the technical aspect such as code quality.

b. Team

In the scrum methodology, a team is composed of developers, the product owner and the scrum master (he is the most experienced developer of the team). The product owner and the scrum master play a big role because they receive disturbing elements (requests of other external team, bugs in the production phase...) while developers are focused on the project. The product owner builds tasks and prioritize them in a backlog respect to their importance and their criticality. He must be available to answer eventual questions of the team.

c. Scrum board

The scrum board allows the team to have a global visibility of the project and its splitting in tasks. It is divided in four columns: to do, in progress, in testing and done. Those four columns form the path to follow by each task. Each developer assigns a task himself through a post-it, works and passes his task to done after verification and validation with the product owner.



Prioritization is done using the MoSCoW method

- ✓ M: must have this; it means what is vital.
- ✓ S: should have this if at all possible; it means it is essential in the version to deliver
- ✓ C: could have this if it does not affecting anything else; it could be in the version to deliver
- ✓ W: won't have this time but would like in the future

Blocking abnormalities are solved in the order of priority. The product owner decides the priority.

d. Product backlog

In the product backlog, we find the different functionalities the final product must have. Those functionalities are represented in "user stories" (a user story is a use case or a part of a use case).

e. Sprint backlog

Here we find functionalities that are going to be implemented during the sprint. It is different of the product backlog where we find the functionalities of the project in its totality.

f. Item

It is an element of the backlog and represents a functionality demonstrable of the project.

2.4 Agile activities

a. Sprint

Developers are focused on the implementation of functionalities defined during this period. It is long three weeks on SYCOMORE.

Scrum : Schéma global

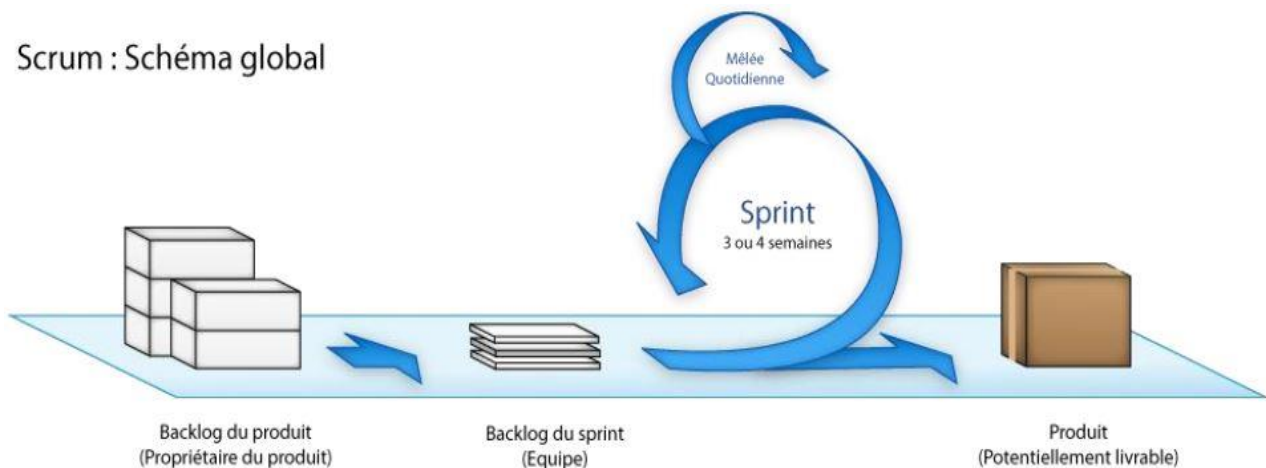


Figure 21 delivery cycle

b. Sprint planning

Sprint planning takes place every two weeks in order to plan tasks of the next sprint. The product owner presents tasks of the next iteration. The team evaluates the complexity of each task. The complexity of a task do not depend on the time the task takes since the time depends on the developer to whom the task is assigned. After that, the team defines the velocity to respect at the end of the sprint.

c. Daily meeting

A typical day starts with DSTUM (Daily Stand Up Meeting). It is a meeting in front of the board where the team discusses about the progress of the sprint. Each team member gives the progress of his task. This meeting has several advantages: it allows a sharing of information about the project; it allows to anticipate eventual problems and to unblock potentially some developers on their tasks.

d. Backlog grooming

It is a meeting where the product owner presents coming projects. Thus, the team can predict developments and problems eventually.

e. Demonstration

It is a meeting where the team presents functionalities developed during the sprint.

f. Retrospective

It is a meeting after the demonstration where each team member gives the strong points and the weak points encountered during the achieved sprint. The team sees also how to improve weak points.

g. Kanban

In SYCOMORE, we apply this methodology during the week that precedes the version to deliver. During that period, a part of the team correct bugs signalled by the validation team while another part of the team design the next version.

2.5 development practice

a. TDD Test-Driven Development

It is a software development technic that consists in writing unit tests before writing the source code. The main advantage of this technic is to ensure the code we write covers requirements. It also allows to write a simple and well structured code. It is composed of five steps:

- Write a first test
- Verify that it fails since the method is not implemented yet
- Write the minimal code that allows to succeed the test
- Verify that the test succeeds
- Refactor the code

b. Pair Programming

It is a technic that has two programs. The first one writes the code while the other one reads, then they change roles. This method allows to maximize the development time since two people think on the same task.

c. Continuous integration

Continuous integration allows to prepare the final delivery of the project. When the development is finished, a global build is done and test classes are launched. Thus, we are sure that a development does not create a regression of the whole project. If there are eventual problems, they are detected and corrected.

d. Delivery process

The SYCOMORE project passes through three environments before the delivery of the version to produce.

- ✓ The development environment where developers code
- ✓ The recipe environment where the software is tested before going to the production. Both the functional team and the integration team test the software here.
- ✓ The integration environment where the software is tested with other existing software in order to check and ensure the global working of the solution. Here functional tests, non-regression tests are more accurate. It is the last step before the production one.

Passing through these phases allow to ensure a more reliable software in terms of quality and in terms of functional requirements.

2.6 Development Environment

The development environment is managed in interne by the developer team. Each Thursday, there is a technical presentation where a team member presents something. It could be some good practices or a topic related to a programming language. For example the presentation of lambda expressions in Java 8 or observable in Angular 2.

a. Git

It is a versioning tool to manage software in a decentralized way. We use it in SYCOMORE to manage source code. It allows the continuous integration since each one can develop on his branch and then merge with the master. Thus, integration problems can be detected and corrected earlier.



b. Angular 2

Angular 2 is an upgrade of AngularJS. A google team has rewritten it; it is a new framework with its own concepts. Angular 2 is a framework for building client applications in HTML and either JavaScript or a language like Typescript that compiles to JavaScript. Angular 2 combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop.



c. Jenkins

It is a continuous integration tool. At each commit, a build is performed in background. That build compiles sources and launches all tests (unit tests and integration tests) developed on the project. If a build fails or a test fails, the team receives a notification indicating the failure encountered. The error can be corrected, we are sure to avoid regressions.



d. SonarQube

It is a free software that allows to measure continuously the quality of the code of the project.



2.7 Integration in the team

As I said before, I started my internship on July 3rd. I spent the two first months in Capgemini where I did my E-learning, I also took part to "La Ruche" project where I worked on CPAP application and RM portal application in order to grow up in competences on technologies (J2EE, Angular 2, HTML, etc.) and on the agile methodology (Scrum).

On September 11th, I joined the SYCOMORE project. During the first week, I read the beginner guideline that contains the technical and the functional description of the project. I had also some presentations with the functional responsible because I had to understand a little bit the project before working on its improvement. Therefore, after more than one week, I had a global view of the application even if it was not in deep.

The technologies used are J2EE, spring, MyBatis and Angular 2. I was familiar with J2EE and spring since I had the occasion to work with at school and on "La Ruche" project. I knew a little bit Angular2 but I was not familiar with it, while I discovered MyBatis for the first time. MyBatis is an ORM (Object Relational Mapping).

After setting up my working environment, I started doing some small annexe tasks in order to be familiar with the application. I started with improving some user interfaces. They were CSS and HTML improvements. Those tasks helped me grow up in competence. They helped me to gradually understand the SYCOMORE project and become a real team member i.e. at the support of the team not just like an observer anymore.

After that, I was able to do a user's story. I participated to poker planning; I developed and tested users stories during the next sprint.

Thanks to the developer team and the functional team, I succeeded in doing my users stories since they were always available to help me both on the technical and on the functional point of views. That helped me understand the application.

a. Board organisation

As I said before the SYCOMORE project is divided in three boards (sub team). The first board is in charge of the optimisation, the second one of the bill control and the third one of the bill injection and curves. I integrated the third board. Before presenting what I did, I am going to first of all talk about what our board does.

b. Board 3

Our board is in charge to inject bills and many other kinds of flow into the SYCOMORE SI. So, when ENEDIS receives the bill from RTE, it has to save it in its information system in order to check if it is consistent with the power subscription.

There are batches that go to find those files and then process them one by one. The processing phase consists of:

- Reading the file
- Checking if the file is at the right format
- Checking if mandatory fields are filled
- Checking if fields are at the right format
- Checking if values have any functional meaning

If these conditions are verified, values are inserted in the corresponding tables of the SYCOMORE database. Some errors are blocking and others are non-blocking i.e. they are simply alerts. Even errors are taken into account since there is a table dedicated to track errors.

Let us see a real example.

Insertion of regional reporting of periods of extreme cold

This is a user's story. The goal is to inject coefficient corresponding to extreme cold in the SYCOMORE SI in order to consider them during the optimisation phase. In fact, in SYCOMORE

there are periods called exceptional periods where some factors are considered in the optimisation and in the unit price rate.

- ✓ RDC (load postpone) it is a period during which there are works on high voltage line of RTE, thus a connection point is involved and is down (out of service). It can not carry energy anymore until the end of works. This means that, the part of energy that flowed through that connection point is transferred to another one. Since ENEDIS buys respect to the amount of energy flowed through a connection point, the price will be high because there will be overages. However, ENEDIS does not pay those overloads since works are done by RTE.
- ✓ DPP (programmed overages) it is the same thing like RDC but this time works are done by ENEDIS at a connection point. The energy is transferred to another connection point. Overages here are measured respect to threshold higher than usual one (the threshold in the normal usage when there is not energy transfer from a PDC to another one).
- ✓ GF (extreme cold) this corresponds to winter period, there are also many overages since the energy is also used for heating houses.

During those periods the unit price changes. In consequence fields of the bill change as well as the processing.

Users stories that inject exceptional periods of RDC and DPP where already done.

2.8 My work on SYCOMORE

I did the user story corresponding to the injection of *extreme cold periods* present in the regional reporting.

An example of file is shown in annexe (figure 1).

At the end of the processing, we must have values of these fields inserted in their corresponding tables in the database.

My second work was the injection of temperatures curves in the SYCOMORE SI. In fact, in the SYCOMORE project we use weather prevision to do some analysis. Temperatures are provided by another information system called SYPEL through a web service. What I did was:

- Call the SYPEL web service
- Receive the response in the JSON format
- Read data in the response and insert them in the corresponding tables in the database

An example of the web service response is shown in annexe (figure 3)

I also worked on the IHM of the graph module. The application provides curves of each connection point graphically. We have many kinds of curves. A search can take many parameters like those that we can see on the picture below

SYCOMORE [REFERENTIEL](#) [MAITRISE DES COUTS](#) **COURBES** [FACTURATION](#) [REPORTING](#) [SUPERVISION](#) [AIDE](#) SYC0001

COURBES DE CHARGE

▼ Recherche des courbes

Entre le et le

Maille de recherche

Type de sélection Valeur de sélection

Éléments de recherche

Type d'éléments ☒ PDC/PDR ☐ Equipement de comptage ☒ Producteur ☐ Maille de rattachement ☐ Station météo

Type de courbes ☒ EA ☒ ER ☒ Puissance souscrite ☐ Température

Nature ☒ Facturée ☒ Mesurée ☐ Simulée ☒ Prévisionnelle ☒ Prévisionnelle en étude

[Importer](#) [Opérations](#) [Versions](#) [Nouvelle Recherche](#) [Je cherche](#)

► Résultats et lancement d'opérations en masse

► Graphique

Figure 22 search curves module

To do a search of curve, we have to select:

- The time period
- The locality, in this example Paris
- The kind of element we want to visualize the graph. It can be a connection point if we want to see graphically the energy that flows through that PDC. It can be also a producer of energy if we want to see the amount of energy produced or a weather station if we want to see temperatures during the search period.
- The kind of energy we want to visualize. It can be the active energy i.e. the energy injected in the network or the energy subtracted from the network or temperatures.
- The nature of curve: measured, invoiced or predicted.

After the search, we obtain a list of curve I we can click to visualize the graph like on the picture below

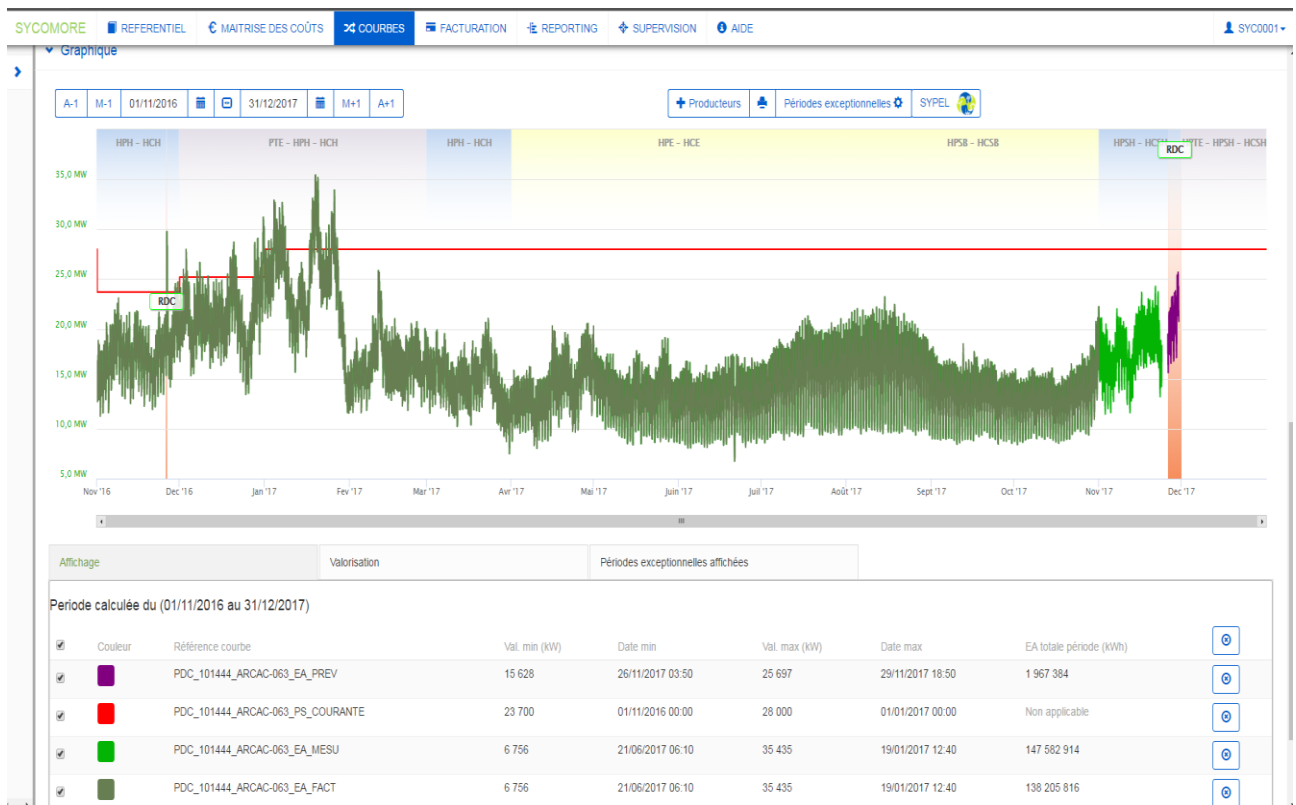


Figure 23 load curve of a PDC

This is the load curve of the connection point PDC_101444_ARCA. It is the concatenation of three kinds of curves. The first part is the billed curve, the second in green is the measured one and the last is the provisional curve. So users see graphically what happens, they have the full control of subscriptions. They are also aware of what happens on the network in the operational point of view. The red line is the subscribed power ENEDIS did nearby RTE.

Users can do many operations on the graph. They can move in the pass by months or by years. They can also move by months or by years in the future. They can zoom the graph in order to accurate their analysis. Those operations are traduced on the IHM by buttons A-1, M-1, A+1, M+1. One important thing they want to do after all operations I listed is to return at the initial graph in one click. I implemented that functionality which allows them to reset the graph at the initial point. That functionality is traduced on the graph by the button with the diamond form between dates.

I improved also the colour picker so that users have many choices of colours. I performed also the export (figure in annexe). In fact, it is possible to write data of the application on a file and then download them in order to process them manually.

Below there are some other improvements I performed at the client side.



SYCOMORE

Correction irritants et petites améliorations suite au retour utilisateur



Recherche des courbes

Entre le 01/10/2016

et le 31/10/2015

la date de début doit être antérieure à la date de fin

Contrôle des dates de recherche

Maille de recherche

Type de sélection

PDC / PDR

Amélioration de la liste Type de sélection

Ajout du Scrollbar

Affichage du nombre
des courbes
sélectionnées dans
Les formulaires
d'opérations en masse

Opération entre courbes

Récapitulatif des 165 courbes sélectionnées

+	+	Addition	PDC_102165_AGNEA-090_EA_PREV
+	+	Addition	PDC_102165_AGNEA-090_EA_FACT
+	+	Addition	PDC_102165_AGNEA-090_EA_MESU
+	+	Addition	PDC_102385_ALENC-090_EA_PREV
+	+	Addition	PDC_102385_ALENC-090_EA_FACT

Date de début de la
courbe simulée

01/10/2016

00

:

00

Date de fin de la
courbe simulée

31/10/2017

23

:

59

DSI/DONNEES/DOP

enedis
L'ELECTRICITE EN RESEAU

Sprint



SYCOMORE

Correction irritants et petites améliorations suite au retour utilisateur

Exporter

Récapitulatif des 2 courbes sélectionnées

Date de début

01/10/2016

00

:

00

Date de fin

31/12/2016

00

:

00

Type d'export

CSV

Format de l'export

☐ Un fichier par courbe
 ☒ Un fichier pour l'ensemble des courbes

Présentation

☐ 6 points/ligne
 ☒ 1 point/ligne

Annuler

Exporter

Remplacer BNPS et Classique par
6 points/ligne et 1 point/ligne

Graphique

A-1

M-1

01/10/2016

31/12/2016

M+1

A+1

+ Producteurs

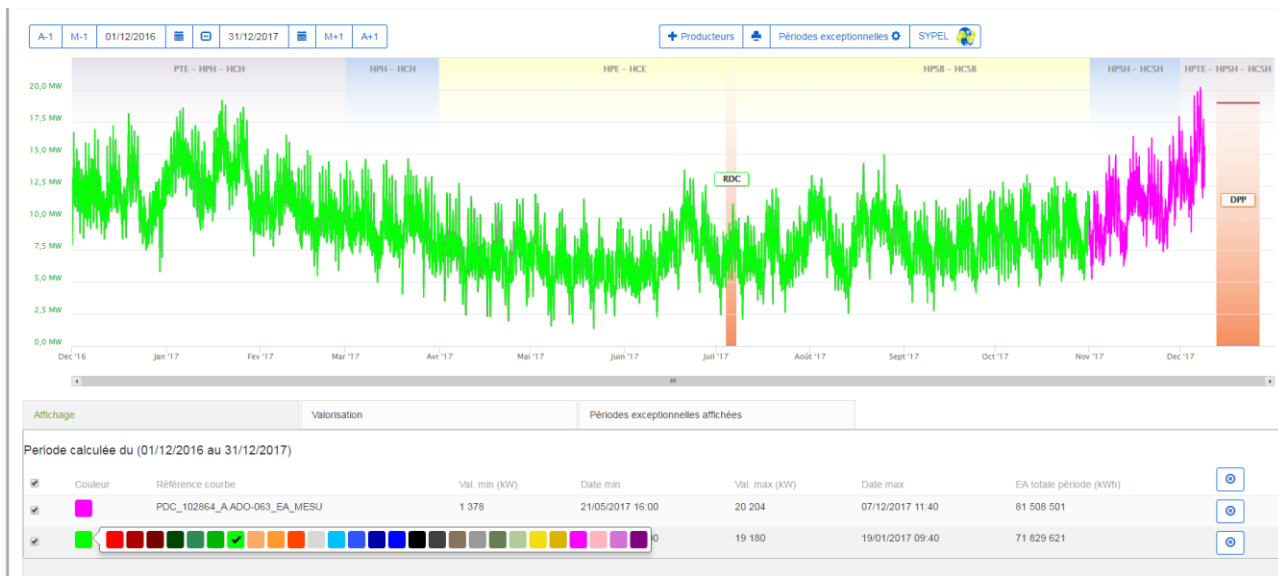
Périodes exceptionnelles

Résolution de l'anomalie du
bouton « Initialiser »

DSI/DONNEES/DOP

enedis
L'ELECTRICITE EN RESEAU

Sprint 9



Conclusion

This internship represents an enriching experience for me in both the technical and the personal point of view.

In fact, this internship allowed me to discover many new technologies and increase the knowledge I acquired during my studies. Obviously, my learning does not stop here because after my internship I am going to continue my training.

CPAP application allowed me to do my first steps in the business world while RM Portal allowed me to search and deeply understand how to build a small application. Thanks to RM Portal, I understood many software development concepts and many frameworks.

SYCOMORE on the other hand instead allowed me to be part to a big project. It allowed to have a real idea about what it is and how it feels to finally integrate the real professional world. I had the chance to be with wonderful people, I learned a lot alongside them.

At the beginning of my internship, it was difficult since it was my first time to have to do with the professional world. My integration was a little bit difficult; it was difficult to understand the work mechanism. Thanks to the support of my colleagues, I overcame those issues and fully integrated in both teams where I assigned.

In overall, I am satisfied by my internship assessment because I learned a lot of things. I succeeded in overcoming many issues and in a global way, I achieved my tasks. I succeeded in everything I was asked to do by my one and the grateful help of my colleagues.

I hope that through this report you have a global view of the proceeding of my internship. I also hope you clearly understood "La Ruche" and SYCOMORE projects.

Annexe

Période de facturation	N° Installation	Libellé ligne	Début période d'application	Fin période d'application	Type de ligne
01.12.2017	101583	Coeff k :	03.12.2017 03:00:00	03.12.2017 03:59:59	D

Figure 1 example of received file from RTE

Date de facturation PE_D_FACT	Element ID PE_K_ELM_ID	Seuil PE_SEUIL	Horodate début PE_HORODAT_DEB	Horodate fin PE_HORODAT_FIN	Nature PE_K_NAT_ID	Type PE_K_TPE_ID
01/12/2017	Elément ID du PDC 101583	0,9875	03/12/2017 03:00:00	03/12/2017 03:59:59	Id de la nature « Facturée » (3)	Id du type « Période GF » (6)

Figure 2 insertion of the file in the corresponding table in database

```
1/meteo/stations/realise_h
oraire/156/20171126/2017
```

GET https://sydel.ededis.fr/api/v1/meteo_raw/p6/station/s/510/temp/prev/2017121

▼ December 19

PUT http://localhost/rest/refere
ntiel/stationMeteo/meteo
Def?horodatePub=201711

PUT http://localhost/rest/refere
ntiel/stationMeteo/meteo
Def?horodatePub=201711

GET http://sydel.ededis.fr/api/v1/meteo/stations/realise_h
oraire/156/20171125/2017

GET https://sydel.ededis.fr/api/

```
1 {
2   "status": 200,
3   "error": false,
4   "msg": "Successful response",
5   "data": {
6     "omm": [
7       156,
8       156
9     ],
10    "horodate": [
11      "2017-11-26 00:00:00",
12      "2017-11-26 01:00:00"
13    ],
14    "valeur": [
15      4,
16      4,
17    ]
18  }
19 }
20 }
```

Figure 3 example of Sydel response web service

```

1 #language:fr
2
3 @RecherchePeriodesExceptionnelles
4 Fonctionnalité: Courbes- recherche des périodes exceptionnelles de grand froid à partir du module courbes
5
6 #Scénario1
7 @RecherchePeriodesExceptionnelles_GF_1
8 Scénario: récupération des PE de GF pour un PDC donné(recherche par défaut)
9
10 Etant donné le PDC fictif avec les paramètres suivants:
11 |refUsage | refRte |
12 |A.ATO-063 | 777779 |
13
14 Et la période exceptionnelle GF fictive avec les paramètres suivants:
15
16 |type | nature | dateDebut | dateFin | seuil | elementId |
17 | GF | Facturée | 01/11/2017 00:00:00 | 30/11/2017 23:59:59 | 0.9787 | 777779 |
18 | GF | Facturée | 01/10/2017 00:00:00 | 31/10/2017 23:59:59 | 0.1477 | 777779 |
19 | GF | Notifiée | 01/09/2017 00:00:00 | 30/09/2017 23:59:59 | 0.2354 | 777779 |
20
21 Quand je fais une recherche personnalisée des PE du PDC "A.ATO-063" sur la période du "01/09/2017" au "31/11/2017"
22 Et je choisis dans le formulaire des périodes exceptionnelles affichage "Personnalisé"
23 Et je sélectionne le type "Episode de grand froid"
24 Et je sélectionne la nature "Facturée"
25 Et je sélectionne la Conformité "Conforme"
26
27 Alors on récupère un tableau de périodes exceptionnelles avec les paramètres suivants:
28
29 | refUsage | type | dateDebut | dateFin | nature | seuil | elementId |
30 | A.ATO-063 | GF | 01/11/2017 00:00:00 | 30/11/2017 23:59:59 | Facturée | 0.9787 | 777779 |
31 | A.ATO-063 | GF | 01/10/2017 00:00:00 | 31/10/2017 23:59:59 | Facturée | 0.1477 | 777779 |

```

Figure 4 example of integration test of extreme cold

```

PDC_101444_ARCAC-063_EA_MESU_20637.csv X
1 Horodate;PDC_101444_ARCAC-063_EA_MESU
2 01/12/2016 00:00;18708
3 01/12/2016 00:10;19746
4 01/12/2016 00:20;19445
5 01/12/2016 00:30;19275
6 01/12/2016 00:40;18919

PDC_101444_ARCAC-063_EA_MESU_20640.csv X
1
2 01/12/2016 00h00;18708;19746;19445;19275;18919;18515
3 01/12/2016 01h00;18013;18684;18376;18026;17760;17385
4 01/12/2016 02h00;17337;18539;18279;17965;17669;17518
5 01/12/2016 03h00;17265;17192;17138;17023;16993;16757
6 01/12/2016 04h00;17096;17150;17247;17241;17216;17235
7 01/12/2016 05h00;17772;17899;18020;18116;18322;18316

```

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Horodate	PDC_101444_ARCAC-063_EA_MESU											
2	01/12/2016 00:00	18708											
3	01/12/2016 00:10	19746											
4	01/12/2016 00:20	19445											
5	01/12/2016 00:30	19275											
6	01/12/2016 00:40	18919											
7	01/12/2016 00:50	18515											

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	01/12/2016 00:00	18708		19746		19445		19275		18919		18515	
2	01/12/2016 01:00	18013		18684		18376		18026		17760		17385	
3	01/12/2016 02:00	17337		18539		18279		17965		17669		17518	
4	01/12/2016 03:00	17265		17192		17138		17023		16993		16757	
5	01/12/2016 04:00	17096		17150		17247		17241		17216		17235	
6	01/12/2016 05:00	17772		17899		18020		18116		18322		18316	
7	01/12/2016 06:00	19076		19433		19771		19982		20127		20520	

Pictures above are kind of export. We can export as CSV or Excel file. We can export as 1 point by row or 6 points by row.

References

<http://www.enedis.fr/company-profile>

<https://ionicframework.com/docs>

<https://start.spring.io/>

<https://www.concretepage.com/spring-boot/spring-boot-jersey-rest-jpa-hibernate-crud-example>

<https://docs.jboss.org/hibernate/orm/3.3/reference/fr-FR/html/queryhql.html>

<http://www.baeldung.com/jackson-bidirectional-relationships-and-infinite-recursion>

<https://angular.io/>

<https://jwt.io/introduction/>

<http://enicolashernandez.blogspot.fr/2010/03/modifier-le-build-path-de-votre-projet.html>

<http://blog.soat.fr/2014/10/a-la-decouverte-du-role-de-product-owner/>

Glossary

CPAP : Continuous Positive Airway Pressure

RM : Resource Manager

SYCOMORE : Système Contrôle des Mesures et Optimisation du Réseau de Transport

SYPEL : Système de Prévisions des Energies Locales, a pour but de mettre à disposition des clients Enedis l'ensemble des données liées à la prévision des flux énergétiques et météorologiques à la maille locale.

PDC : Point De Connexion (Connection point)

MCD: Model Conceptuel de Données (data design model)

ENEDIS

RPT Réseau Public de Transport (public transport network)

EDF Electricité De France (Electricity of France)

ERDF Electricité Réseau Distribution de France (Electricity Network Distribution of France)

RTE Réseau de Transport d'électricité (Electricity transport network)

TURPE Tarif Utilisation Réseaux Public d'Electricité

CART Contrat d'Accès au Réseau de Transport

CRE Commission de régulation Electrique

MOA Maîtrise d'ouvrage

MOE Maîtrise d'œuvre

PO Product Owner

RDC Report de charge

DPP Dépassement ponctuel programmé

GF Grand Froid

DT Direction Technique