

## Chapitre I : Etude de l'existant

### I.1 Préambule

Dans de nombreux domaines de la vie professionnelle, on se trouve confronter au problème de la gestion des ressources et d'événements ou la planification des ressources. Dans les usines, des pièces doivent être cheminées à travers plusieurs machines, la gestion de ce trafic doit répondre à certaines espérances, tel que la maximisation de la production ou l'exploitation maximale des machines. Dans les hôpitaux, un certain nombre de personnels constitués d'infirmiers et de médecins doivent être attribués aux postes de travail de manière à obéir à certaines règles de gestion des hôpitaux. Donc cela nécessite souvent une élaboration périodique et stratégique de planning de travail de personnel et les raisons de cette élaboration sont multiples. Elles vont de la réduction des coûts de productivité à l'amélioration de la qualité de service aux clients, en passant par l'assurance d'une bonne qualité de vie aux employés.

Dans le domaine de gestion des ressources et d'événements plusieurs applications ont été développées, soit des applications monoposte ou multi postes (sous réseau). Dans le tableau suivant on résume quelques applications.

<b>Application</b>	<b>Editeur</b>	<b>réseau</b>	<b>Fonction</b>
RAPLA [1]	Cristophe Dokhlass	Non	Gestion de l'emploi du temps
Vista plan [2]	Lazard Alexandre	Non	//
Le Planning[3]	Patrick Pelisson	Non	//
EDT [4]	index-éducation	Non	//
Edt-soft [5]	2C Informatique	Non	//
Visual Time Tabling [6]	Sylvain Piechowiak	Non	//
FET [7]	Liviu lalescu	Non	//
HYPERPLANNING [8]	index-éducation	Oui	//
ADE [9]	Jean-Luc Dagn	Oui	//
Gantt [10]	I.Veenpere, R. Kivimaa, I. Kuldkeep	Non	Gestion des activités de l'entreprise
Gesroom [11]	Safeware	Non	//
Algem [12]	Jean Marc Globat	Non	//
E-Optim [13]	Horizontal Software	Non	//
WorkForce Scheduler [14]	Kronos	Non	//
Kelio Planification [15]	Bodet	Non	//
AIGLe [16]	Dominique CARON	Oui	//
GRR [17]	Demove	Oui	//
Rooming'it [18]	Develop'it	Oui	//
MCS-Room [19]	2ai Software	Oui	//
POMCLASS [20]	Stéphane Cabanis	Oui	//

## Chapitre I : Etude de l'existant

---

PlanningPME [21]	TARGET SKILL	Oui	//
VISUAL PLANNING [22]	STILOG	Oui	//

Tableau 01 : Quelques travaux dans le domaine de gestion des ressources et d'événements.

### **I.2 Présentation de FSEI**

La faculté des sciences exactes et d'informatique est une filiale de l'université d'Abdelhamid Ibn Badis Mostaganem. Cette faculté comporte les trois départements suivants : Mathématiques et Informatique, Physique et Chimie, elle est dotée de cinq blocs suivants :

1. Bloc administratif (Doyenné).
2. Bloc pédagogique.
3. Bloc des amphis (quatre amphis).
4. Bibliothèque.
5. Bloc de laboratoires.

# Chapitre I : Etude de l'existant

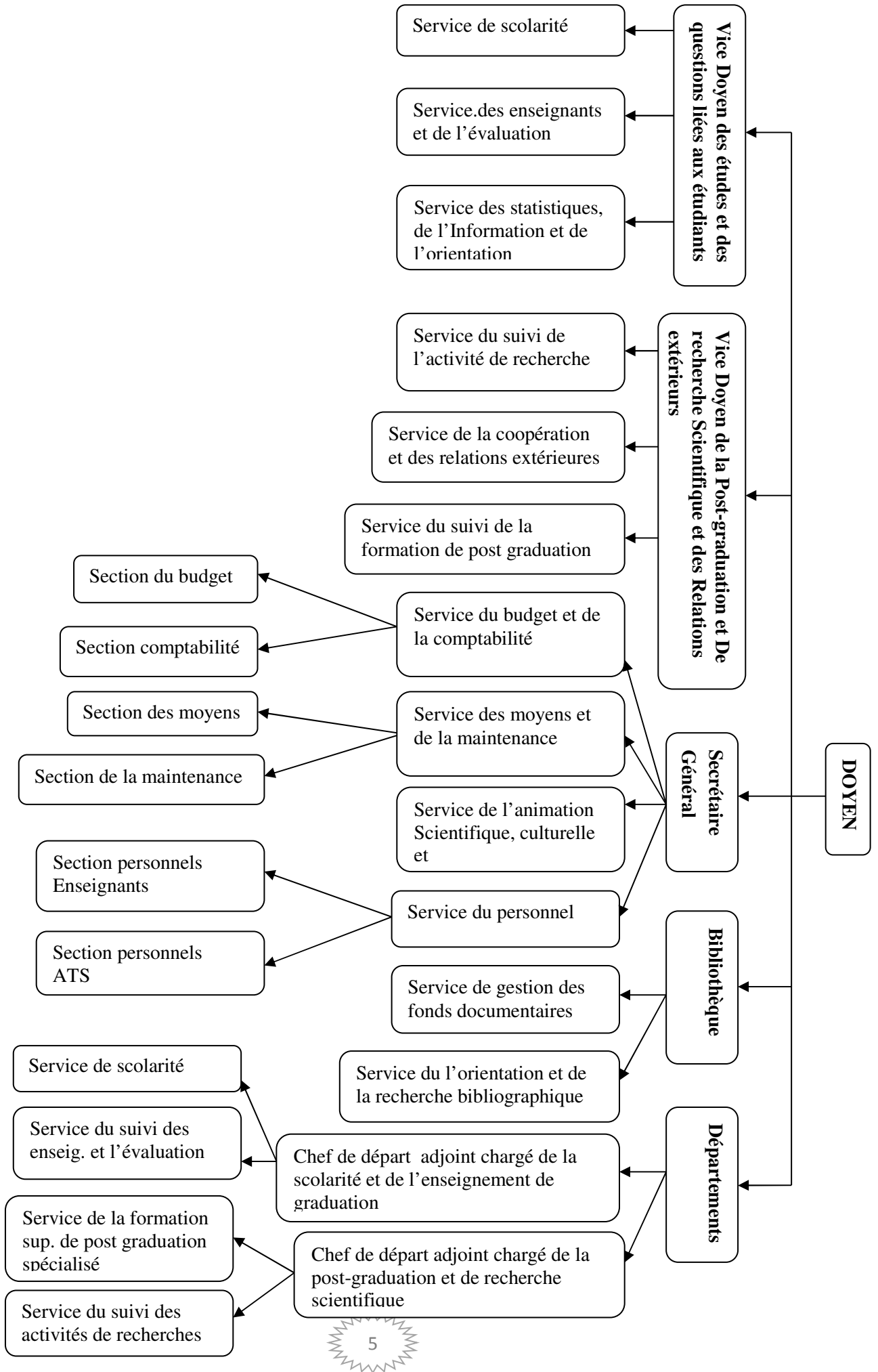


Figure 01 : Organigramme de FSEI

## I.3 Ressources

Plusieurs définitions peuvent être attribuées aux ressources :

### I.3.1 Définition 1

Ensemble des outils dont dispose un sujet pour satisfaire aux exigences des tâches auxquelles il est confronté [23].

### I.3.2 Définition 2

Le concept de ressources représente le terme le plus générique pour désigner ce que le sujet a à sa disposition pour agir, c'est-à-dire en didactique, pour atteindre un but préétabli (accomplir une tâche) [23].

### I.3.3 Types des ressources

Selon les domaines, on peut diviser les ressources aux catégories suivantes [24] :

- **Ressource naturelle** : matière première ou source d'énergie naturelle permettant de subvenir aux besoins d'un être vivant, d'une espèce ou de la société humaine. Selon leurs origines et leurs compositions, les ressources naturelles peuvent être classées comme suit :
  - **Ressources géologiques** :
    - organiques fossiles : pétrole, charbon, gaz naturel, tourbe.
    - minérales : minerais, roches, eau.
  - **Ressources biologiques** :
    - organiques : viandes, bois.
    - minérales : calcium extrait de coquilles.
- **Ressource (gestion de projet)** : ressources économiques utilisées dans la planification des tâches.
- **Ressource documentaire** : en particulier les ressources continues.
- **Ressource (informatique)** incluant :
  - **Ressource (Web)** : toute entité identifiée par une URI (Unified Resource Identifier).
  - **Ressource (Macintosh)** : données associées à un fichier Mac OS.
  - **Ressource (Windows)** : données embarquées dans des fichiers exécutables (.Exe) et lien (.dll).
  - **Ressource (Java)** : application data.
- **Ressources (entreprise et organisation)** incluant :
  - les **ressources humaines (RH)** : ou capital humain (gestionnaires et employés).
  - les **ressources informationnelles** : information et technologies d'information).
  - les **ressources matérielles** : équipements, outils, bâtiments.
  - les **ressources financières** : budget, liquidité, capital-action.

# Chapitre I : Etude de l'existant

---

- **Ressources en sport** : ensemble de potentialités dont dispose un athlète et qu'il peut améliorer par le biais de l'entraînement.

## I.3.4 Ressources de la faculté

Les ressources considérées sont les entités physiques nécessaires à l'élaboration des emplois du temps. Il s'agit des salles, des enseignants, des groupes et des étudiants. Les ressources sont caractérisées par des données abstraites et des données spécifiques. Les données abstraites caractérisent chaque ressource et sont constituées d'un code, qui permet de la différencier des autres ressources, son calendrier qui précise quels sont les jours de disponibilité et d'indisponibilité et sa description. En plus de ces données abstraites, chaque ressource possède des caractéristiques spécifiques qui dépendent du type de la ressource. L'intérêt de distinguer ces deux types de caractéristiques est que l'outil peut très facilement évoluer pour prendre en compte de nouveaux types de ressources.

Dans la suite nous décrivons les caractéristiques spécifiques des ressources considérées dans le projet :

- **Les ressources de type « salle »** : Une salle est un lieu dans lequel sont assurés des enseignements. Le type d'une salle est une indication sur le type d'enseignement qu'on peut y faire [25].

- **Les ressources de type « enseignant »** : L'enseignant désigne une personne pouvant assurer des enseignements. Chaque enseignant est caractérisé par : son nom et son prénom, son grade, sa résidence administrative, sa spécialité. La spécialité renseigne sur les matières que peut enseigner un enseignant [25].

- **Les ressources de type « groupe »** : Un groupe est un ensemble d'étudiants. Il se compose d'autres groupes qui peuvent être réduits à un seul étudiant (un binôme ou monôme). Cette modélisation ne fait pas de distinction entre groupe et étudiant. En général les étudiants d'une filière sont décomposés en section et chaque section est décomposée en plusieurs groupes [25].

## I.4 Evénements

L'événement principal sur lequel nous allons nous baser dans la réalisation de notre projet est l'emploi du temps :

### I.4.1 Emploi du temps

L'emploi du temps est un plan représentatif définissant le temps nécessaire pour élaborer des tâches et des objectifs fixés préalablement sous forme de créneaux horaires. Il consiste à gérer les charges de travail dans le temps tout en prenant compte des ressources humaines et matérielles disponibles.

## Chapitre I : Etude de l'existant

---

Donc on peut définir le problème de l'emploi du temps comme chercher à gérer les ressources d'une manière satisfaisante en respectant au maximum que possible les contraintes de temps [26].

Parmi plusieurs classifications possibles de l'emploi du temps nous avons choisi une classification de l'emploi de temps en deux catégories principales :

### a) Emploi du temps des cours :

Etablir le programme hebdomadaire pour tous les cours d'un ensemble de faculté en réduisant au minimum que possible les chevauchements des cours ayant les étudiants communs [26].

### b) Emploi de temps des examens :

Etablir un programme pour les examens d'un ensemble d'étudiants. L'emploi du temps d'un examen est un peu différent de l'emploi du temps des cours car il impose certaines contraintes (écarter les examens pour les étudiants autant que possibles, éviter le chevauchement des examens des cours ayant les étudiants communs) qui ne sont pas prises en compte dans celui des cours [26].

### I.4.1.1 Notion du temps

Les analyses menées avec les utilisateurs mettent en avant différentes perceptions du temps. Pour représenter le temps, deux notions de base sont nécessaires : l'*instant* et la *durée*. De plus, pour le problème des emplois du temps on rajoute la notion de *fréquence*. Cette notion de fréquence n'est pas indispensable puisqu'elle peut s'exprimer à partir de celles d'instant et de durée. Dans les emplois du temps, on dira qu'un enseignement a lieu tous les 15 jours (*fréquence*) si la durée qui sépare deux séances consécutives de l'enseignement est d'au moins 15 jours (*durée*). Par exemple, dire « qu'il y a cours de mathématiques tous les quinze jours » c'est-à-dire que la durée qui sépare deux séances consécutives de ce cours est de 15 jours. Ainsi on distingue la durée de chaque séance et de la durée qui sépare les séances [27].

#### - Les entités temporelles

Pour modéliser le temps, les entités : *date*, *heure*, *durée*, *créneau* et *calendrier* sont définies. Une *date* désigne un instant défini par un triplet (jour, mois, année). A partir de ce triplet, on détermine la valeur qui lui est associée sur l'axe des jours. Pour avoir un grain plus fin, la notion d'heure est utilisée. Il s'agit d'un nombre entier compris entre la valeur minimale  $H_{Min}$  et la valeur maximale  $H_{Max}$ . Ces deux nombres correspondent à des heures par rapport à une date. Par exemple, dans notre cas, nous avons choisi  $H_{Min}=8h30$  et  $H_{Max}=17h30$  par défaut. Une *durée* est un nombre compris entre  $D_{Min}$  et  $D_{max} = H_{max} - H_{Min}$ .  $D_{Min}$  représente la plus petite unité temporelle disponible [27].

#### - Les séances et les réservations

Une *séance* correspond à une instance temporelle d'un enseignement à une date donnée, pendant un créneau précis. L'ensemble des séances d'une ressource est appelé *planning* de cette ressource. Il s'agit de l'ensemble des séances dans lesquelles apparaît la ressource. Ainsi, il est possible de considérer le planning d'une salle au même titre que celui d'un

## Chapitre I : Etude de l'existant

---

enseignant. La notion de planning apparaît également au niveau des enseignements : cela permet ainsi de manipuler l'évolution dans le temps d'un enseignement particulier en vue. Une *réserve* correspond à un créneau ajouté au planning d'une ressource. Elle correspond à une option posée sur l'occupation de cette ressource. Par rapport à une séance, une réservation n'est pas associée à un enseignement. Les réservations apportent de la souplesse dans l'utilisation de l'outil notamment lors de la phase de création des emplois du temps par les différents responsables [27].

### **I.4.1.2 Filières**

La notion de filière permet de limiter le nombre des ressources à considérer. En effet, le responsable d'une filière n'a besoin de connaître que le planning des ressources qu'il utilise. Cependant, certains conflits ne peuvent être détectés que lorsque les différentes filières sont rassemblées dans une base de données commune. Il s'agit surtout des conflits d'enseignants et de salles puisqu'il est fréquent qu'un enseignant intervienne dans plusieurs filières ou qu'une salle soit partagée par plusieurs filières [27].

### **I.4.1.3 Contraintes**

L'analyse sur le terrain montre que les données gérées doivent vérifier certaines contraintes pour garantir leur cohérence. De manière abstraite, les contraintes à respecter peuvent être classées en deux groupes : les contraintes physiques et les contraintes pédagogiques [27].

#### **- Les contraintes physiques**

Ces contraintes ne doivent pas être violées sinon cela conduirait à des situations conflictuelles. On dira qu'il y a un « conflit physique de ressource » entre deux séances  $s_1$  et  $s_2$  si ces deux séances ont une ressource en commun pendant une durée non nulle. Voici les contraintes physiques:

- une ressource ne peut pas être occupée en même temps dans deux séances différentes.
- dès qu'une ressource  $r$  est occupée par une séance  $s$ , toutes ses ressources filles  $r_f$  sont également occupées par la même séance et cela récursivement (il s'agit d'une contrainte liée à la hiérarchisation).
- on ne peut pas mettre plus d'étudiants qu'il n'y a de places dans une salle
- le volume horaire total des séances d'un enseignement ne peut pas dépasser le volume prévu.
- les calendriers sont respectés pour toutes les ressources.

#### **- Les contraintes pédagogiques**

Lors des interviews, des contraintes pédagogiques ont pu être identifiées. Elles se différencient des contraintes physiques par le fait qu'elles peuvent éventuellement être violées, dans ce cas on obtient des emplois du temps de moins bonne qualité d'un point de vue pédagogique. Typiquement ces contraintes sont utilisées pour exprimer ce que doit être un « bon » emploi du temps. Voici quelques exemples de ces contraintes:

- homogénéiser la durée des séances d'un enseignement.
- éviter les séances creuses dans l'emploi du temps.
- éviter de placer plusieurs séances d'un même enseignement dans une journée.

## Chapitre I : Etude de l'existant

---

- éviter de finir après 17h30.
- éviter de placer des cours l'après midi.
- favoriser la régularité dans les emplois du temps.

On peut remarquer que dans l'expression de ces contraintes pédagogiques les termes utilisés ne sont pas précis (éviter, devraient, si possible ...). Ces contraintes sont plus difficiles à formaliser que les contraintes physiques et leur traitement est plus délicat. Il peut y avoir des degrés d'importance entre les contraintes et une même contrainte peut avoir différents « degrés » d'importance. Par exemple, la contrainte « éviter les séances creuses dans l'emploi du temps » représente l'ensemble de toutes les contraintes « éviter les séances creuses d'une durée  $D$  dans l'emploi du temps », où  $D \in [30\text{min}, 1\text{h}, 1\text{h}30, \dots]$ . Mais il est moins important d'éviter les séances creuses de 60 minutes que d'éviter les trous de 10 h.



## **Conclusion du chapitre I**

Dans ce chapitre, nous avons présenté la faculté des sciences exactes et d'informatique, ainsi que nous avons défini les ressources et les événements, et nous avons étudié en général le problème sur lequel notre projet tourne, c'est le problème de l'emploi du temps.

Nous avons conclu que la génération de l'emploi du temps ne consiste pas seulement à l'affectation des ressources à des événements, mais il y a des contraintes que le créateur de l'emploi du temps doit respecter pour garantir la génération des emplois du temps fiables et de bonne qualité.

Le chapitre qui suit est dédié au réseau et sécurité de l'information tel que nous allons aborder des généralités de ces deux termes qui sont aussi des mots clés dans notre projet.



### **Chapitre II : Réseaux et Sécurité de l'information**

#### **II.1 Préambule**

Les réseaux informatiques qui permettaient à leur origine de relier des terminaux passifs à de gros ordinateurs centraux autorisent à l'heure actuelle l'interconnexion de gros types d'ordinateurs que ce soit de gros serveurs, des stations de travail, des ordinateurs personnels ou de simples terminaux graphiques. Les services qu'ils offrent font partie de la vie courante des entreprises et administrations (banques, gestion, commerce, bases de données, recherche, etc.) et des particuliers (messagerie, loisirs, services d'informations par minitel et internet etc.) [28].

La sécurité des systèmes d'informations est une discipline de première importance car le système d'information est pour toute entreprise un élément absolument vital. Puisque le système d'information est vital tout ce qui le menace est mortel, cela semble couler de source. Les menaces contre le système d'information entrent dans l'une des catégories suivante : atteinte à la disponibilité des systèmes et des données, destruction des données, corruption ou falsification des données, vol ou espionnage des données, usage illicite d'un système ou d'un réseau, usage d'un système compromis pour attaquer d'autres cibles [29].

#### **II.2 Réseau**

##### **II.2.1 Définitions**

Un réseau est un ensemble d'ordinateurs (ou périphériques) autonomes connectés entre eux et qui sont situés dans un certains domaines géographiques.

Les réseaux informatiques sont nés du besoin de faire communiquer des terminaux distants avec un site central puis des ordinateurs entre eux.

Dans un premier temps ces communications étaient juste destinées aux transports de données informatiques alors qu'aujourd'hui on se dirige plutôt vers des réseaux qui intègrent à la fois des données mais en plus, la parole, et la vidéo [30].

##### **II.2.2 Topologies de réseau**

La topologie est une organisation physique et logique d'un réseau. L'organisation physique concerne la façon dont les machines sont connectées (Bus, Anneau, Etoile ...). La topologie logique montre comment les informations circulent sur les réseaux (Diffusion ou point à point) [30].

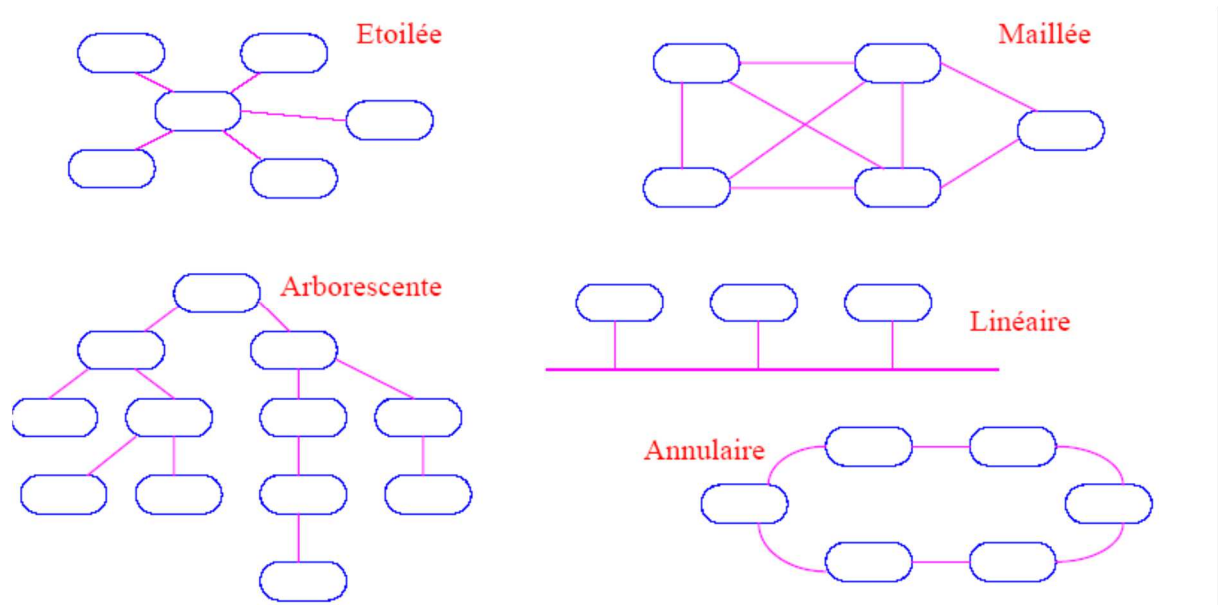


Figure 02 : quelques topologies d'interconnexion [31].

### II.2.3 Types des réseaux

On distingue différents types de réseaux selon leur taille (en terme de nombre de machines), leur vitesse de transfert des données ainsi que leur étendue.

On fait généralement trois catégories de réseaux [32] :

- LAN (local area network).
- MAN (metropolitan area network).
- WAN (wide area network).

#### II.2.3.1 LAN

LAN signifie Local Area Network (en français Réseau Local). Il s'agit d'un ensemble d'ordinateurs appartenant à une même organisation et reliés entre eux dans une petite aire géographique par un réseau, souvent à l'aide d'une même technologie (la plus répandue étant Ethernet).

Un réseau local est donc un réseau sous sa forme la plus simple. La vitesse de transfert de données d'un réseau local peut s'échelonner entre 10 Mbps (pour un réseau Ethernet par exemple) et 1 Gbps (en FDDI ou Gigabit Ethernet par exemple). La taille d'un réseau local peut atteindre jusqu'à 100 voire 1000 utilisateurs.

En élargissant le contexte de la définition aux services qu'apporte le réseau local, il est possible de distinguer deux modes de fonctionnement :

- dans un environnement "égal à égal" (en anglais peer to peer), dans lequel il n'y a pas d'ordinateur central et chaque ordinateur a un rôle similaire.
- dans un environnement "client/serveur", dans lequel un ordinateur central fournit des services réseau aux utilisateurs [32].

#### II.2.3.2 MAN

Les MAN (Metropolitan Area Network) interconnectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de km) à des débits importants. Ainsi un MAN permet à deux nœuds distants de communiquer comme si ils faisaient partie d'un même réseau local.

## Chapitre II : Réseaux et sécurité de l'information

Un MAN est formé de commutateurs ou de routeurs interconnectés par des liens hauts débits (en général en fibre optique) [32].

### II.2.3.3 WAN

Un WAN (Wide Area Network ou réseau étendu) interconnecte plusieurs LANs à travers de grandes distances géographiques.

Les débits disponibles sur un WAN résultent d'un arbitrage avec le coût des liaisons (qui augmente avec la distance) et peuvent être faibles. Les WAN fonctionnent grâce à des routeurs qui permettent de "choisir" le trajet le plus approprié pour atteindre un nœud du réseau. Le plus connu des WAN est Internet [32].

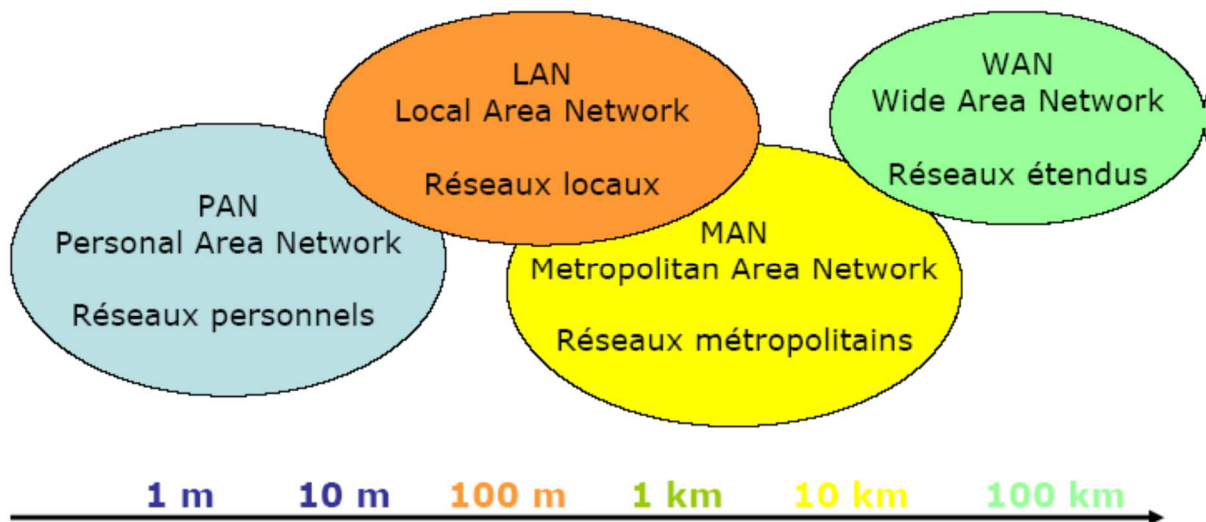


Figure 03 : Classification des réseaux selon leur taille [32].

### II.2.3.4 Similitudes entre types de réseaux

Les différents types de réseaux ont généralement les points suivants en commun [32] :

- Serveurs : ordinateurs qui fournissent des ressources partagées aux utilisateurs par un serveur de réseau.
- Clients : ordinateurs qui accèdent aux ressources partagées fournies par un serveur de réseau.
- Support de connexion : conditionne la façon dont les ordinateurs sont reliés entre eux.
- Imprimantes et autres périphériques partagés : fichiers, imprimantes ou autres éléments utilisés par les usagers du réseau.
- Ressources diverses : autres ressources fournies par le serveur.

## II.2.4 Architecture client/serveur

### II.2.4.1 Présentation de l'architecture client/serveur

De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui

leur fournit des services. Ces services sont des programmes fournissant des données telles que l'heure, des fichiers, une connexion, etc.

Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes. On parle ainsi de client (client FTP, client de messagerie, etc.) lorsque l'on désigne un programme tournant sur une machine cliente, capable de traiter des informations qu'il récupère auprès d'un serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client de messagerie il s'agit de courrier électronique) [32].

### II.2.4.2 Fonctionnement d'un système client/serveur

Un système client/serveur fonctionne selon le schéma suivant :

Le client émet une requête vers le serveur grâce à son adresse IP et le port, qui désigne un service particulier du serveur.

Le serveur reçoit la demande et répond à l'aide de l'adresse de la machine cliente et son port.

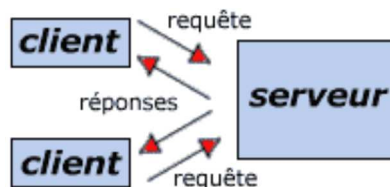


Figure 04 : Fonctionnement d'un système client/serveur [32].

### II.2.4.3 Avantages et inconvénients de l'architecture client/serveur

#### • Avantages [32]

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

**Des ressources centralisées :** étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction.

**Une meilleure sécurité :** car le nombre de points d'entrée permettant l'accès aux données est moins important.

**Une administration au niveau serveur :** les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés.

**Un réseau évolutif :** grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure.

#### • Inconvénients [32]

L'architecture client/serveur a tout de même quelques lacunes parmi lesquelles :

Un coût élevé dû à la technicité du serveur.

Un maillon faible : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui. Heureusement, le serveur a une grande tolérance aux pannes (notamment grâce au système RAID (Redundant Array of Independent Disks)).

### II.3 La sécurité informatique

#### II.3.1 Définition

La sécurité des systèmes d'information est l'ensemble des moyens techniques, organisationnels, juridiques et humaines nécessaires et mis en place pour conserver, rétablir et garantir la sécurité du système d'information. Assurer la sécurité du système d'information est une activité du management du système d'information [33].

#### II.3.2 Types de sécurité

On distingue trois types de sécurité [33] :

- **Sécurité humaine** : Un centre regroupant un ou plusieurs serveurs hébergeant les informations sensibles d'une entreprise est une cible de choix pour des êtres mal intentionnés. C'est pourquoi la sécurité des accès doit être surveillée au mieux. Pour une sécurité des informations maximale, la présence humaine continue (gardiennage 24/24) peut être renforcée par un système de vidéo surveillance, de régulation d'entrées/sorties par badges et d'alerte anti-intrusion déployé dans l'ensemble de ce bâtiment.
- **Sécurité physique et matérielle** : Pour éviter la destruction des informations en cas de destruction d'un serveur, les équipements déclarés comme sensibles doivent être systématiquement redondés, c'est-à-dire dupliqués en un ou plusieurs endroits. Il existe aujourd'hui des systèmes et des techniques de sauvegarde de données régulières afin d'éviter les pertes de données imprévues (par exemple le fonctionnement de deux disques miroirs via RAID1 (Redundant Array of Independent Disks) ou l'utilisation de scripts de sauvegarde automatique).  
Pour résister à une coupure (criminelle ou non) de ressources, un centre de serveurs se doit aussi de bénéficier d'une autonomie électrique régulièrement testée en cas de coupure électrique. Il peut aussi bénéficier d'un système de sécurité incendie ainsi que d'un groupe de refroidissement autonome adapté pour réguler au mieux la température des serveurs et en protéger le matériel.
- **Sécurité logicielle** : Pour bien protéger son système d'information, il est nécessaire de protéger son système de toutes les attaques logicielles.

#### II.3.3 Aspects de la sécurité

Pour considérer efficacement les besoins de sécurité d'une organisation et évaluer et choisir les nombreux produits et politiques de sécurité, le responsable de la sécurité a besoin de moyens systématiques de définition des exigences de sécurité et de caractérisation des approches qui satisfont ces exigences. Une approche possible est de considérer trois aspects de la sécurité de l'information : services de sécurité, mécanismes de sécurité et attaques de sécurité [34].

##### II.3.3.1 Services de la sécurité

Une classification utile des services de sécurité est la suivante [34] :

- **Confidentialité** : La confidentialité est la protection contre les attaques passives des données transmises. Plusieurs niveaux de protection de la confidentialité sont envisageables. Le service le plus général protège toutes les données transmises entre deux utilisateurs pendant une période donnée. Des formes restreintes de ce service

peuvent également être définies, incluant la protection d'un message élémentaire ou même de champs spécifiques à l'intérieur d'un message. Un autre aspect de la confidentialité est la protection du flot de trafic contre l'analyse. Cela requiert qu'un attaquant ne puisse observer les sources et destinations, les fréquences, longueurs ou autres caractéristiques du trafic existant sur un équipement de communication.

- **Authentification** : Le service d'authentification permet évidemment d'assurer l'authenticité d'une communication. Dans le cas d'un message élémentaire, tel un signal d'avertissement, d'alarme, ou un ordre de tir, la fonction du service d'authentification est d'assurer le destinataire que le message a bien pour origine la source dont il prétend être issu. Dans le cas d'une interaction suivie, telle une connexion d'un terminal à un serveur, deux aspects sont concernés. En premier lieu, lors de l'initialisation de la connexion, il assure que les deux entités sont authentiques (c'est-à-dire, que chaque entité est celle qu'elle dit être). Ensuite, le service doit assurer que la connexion n'est pas perturbée par une tierce partie qui pourrait se faire passer pour une des deux entités légitimes à des fins de transmissions ou de réceptions non autorisées.
- **Intégrité** : À l'instar de la confidentialité, l'intégrité s'applique à un flux de messages, un seul message, ou à certains champs à l'intérieur d'un message. Là encore, la meilleure approche est une protection totale du flux. Un service d'intégrité orienté connexion, traitant un flot de messages, assure que les messages sont reçus aussitôt qu'envoyés, sans duplication, insertion, modification, réorganisation ou répétition. La destruction de données est également traitée par ce service. Ainsi, un service d'intégrité orienté connexion concerne à la fois la modification de flux de messages et le refus de service. D'un autre côté, un service d'intégrité non orienté connexion, traitant des messages individuels sans regard sur un contexte plus large, fournit généralement une protection contre la seule modification de message.
- **Non-répudiation** : La non-répudiation empêche tant l'expéditeur que le receveur de nier avoir transmis ou reçu un message. Ainsi, lorsqu'un message est envoyé, le receveur peut prouver que le message a bien été envoyé par l'expéditeur prétendu. De même, lorsqu'un message est reçu, l'expéditeur peut prouver que le message a bien été reçu par le receveur prétendu.
- **Contrôle d'accès** : Dans le contexte de la sécurité des réseaux, le contrôle d'accès est la faculté de limiter et de contrôler l'accès à des systèmes et des applications via des maillons de communications. Pour accomplir ce contrôle, chaque entité essayant d'obtenir un accès doit d'abord être authentifiée, de telle sorte que les droits d'accès puissent être adaptés à son cas.
- **Disponibilité** : De nombreuses attaques peuvent résulter en une perte ou une réduction de la disponibilité d'un service ou d'un système. Certaines de ces attaques sont susceptibles d'être l'objet de contre-mesures automatiques, telle que l'authentification et le chiffrement, alors que d'autres exigent une action humaine pour prévenir ou se rétablir de la perte de disponibilité des éléments d'un système.

### II.3.3.2 Mécanismes de la sécurité

Un seul mécanisme ne peut fournir tous les services de sécurité. On peut noter qu'un élément particulier sous-tend la plupart des mécanismes de sécurité en usage : les techniques



cryptographiques. Le chiffrement ou des transformations similaires de l'information est le moyen le plus courant pour fournir une sécurité [34].

### II.3.3.3 Attaques de la sécurité

Les attaques portées à la sécurité d'un ordinateur ou d'un réseau sont mieux caractérisées en considérant le système en tant que fournisseur d'information. En général, il existe un flot d'information issu d'une source (un fichier ou une zone de la mémoire centrale), vers une destination - un autre fichier ou utilisateur. Il existe quatre catégories d'attaques : interruption, interception, modification, fabrication [34].

- **Interruption** : Un atout du système est détruit ou devient indisponible ou inutilisable. C'est une attaque portée à la disponibilité. La destruction d'une pièce matérielle (tel un disque dur), la coupure d'une ligne de communication, ou la mise hors service d'un système de gestion des fichiers en sont des exemples.
- **Interception** : Une tierce partie non autorisée obtient un accès à un atout. C'est une attaque portée à la confidentialité. Il peut s'agir d'une personne, d'un programme ou d'un ordinateur. Une écoute téléphonique dans le but de capturer des données sur un réseau, ou la copie non autorisée de fichiers ou de programmes en sont des exemples.
- **Modification** : Une tierce partie non autorisée obtient accès à un atout et le modifie de façon (presque) indétectable. Il s'agit d'une attaque portée à l'intégrité. Changer des valeurs dans un fichier de données, altérer un programme de façon à bouleverser son comportement ou modifier le contenu de messages transmis sur un réseau sont des exemples de telles attaques.
- **Fabrication** : Une tierce partie non autorisée insère des contrefaçons dans le système. C'est une attaque portée à l'authenticité. Il peut s'agir de l'insertion de faux messages dans un réseau ou l'ajout d'enregistrements à un fichier.

#### II.3.3.3.1 Exemples d'attaque

Il existe plusieurs attaques logicielles connues à ce jour parmi lesquelles [35] :

- **Virus** : Un virus est un segment de programme qui, lorsqu'il s'exécute, se reproduit en s'adjoignant à un autre programme (du système ou d'une application), et qui devient ainsi un cheval de Troie. Puis le virus peut ensuite se propager à d'autres ordinateurs (via un réseau) à l'aide du programme légitime sur lequel il s'est greffé. Il peut également avoir comme effets de nuire en perturbant plus ou moins gravement le fonctionnement de l'ordinateur infecté.
- **Cheval de Troie** : Un Cheval de Troie (trojan en anglais) est un programme effectuant une fonction illicite tout en donnant l'apparence d'effectuer une fonction légitime. La fonction illicite peut consister en la divulgation ou l'altération d'informations.
- **Ver** : Un ver est un programme autonome qui se reproduit et se propage à l'insu des utilisateurs. Contrairement aux virus, un ver n'a pas besoin d'un logiciel hôte pour se dupliquer. Le ver a habituellement un objectif malicieux, par exemple : espionner l'ordinateur dans lequel il réside, détruire des données sur l'ordinateur infecté.
- **Bombe logique** : Une Bombe logique est une partie d'un programme malveillant (virus, cheval de Troie, etc.) qui reste dormante dans le système hôte jusqu'à ce qu'un instant ou un événement survienne, ou encore que certaines conditions soient réunies, pour déclencher des effets dévastateurs en son sein.

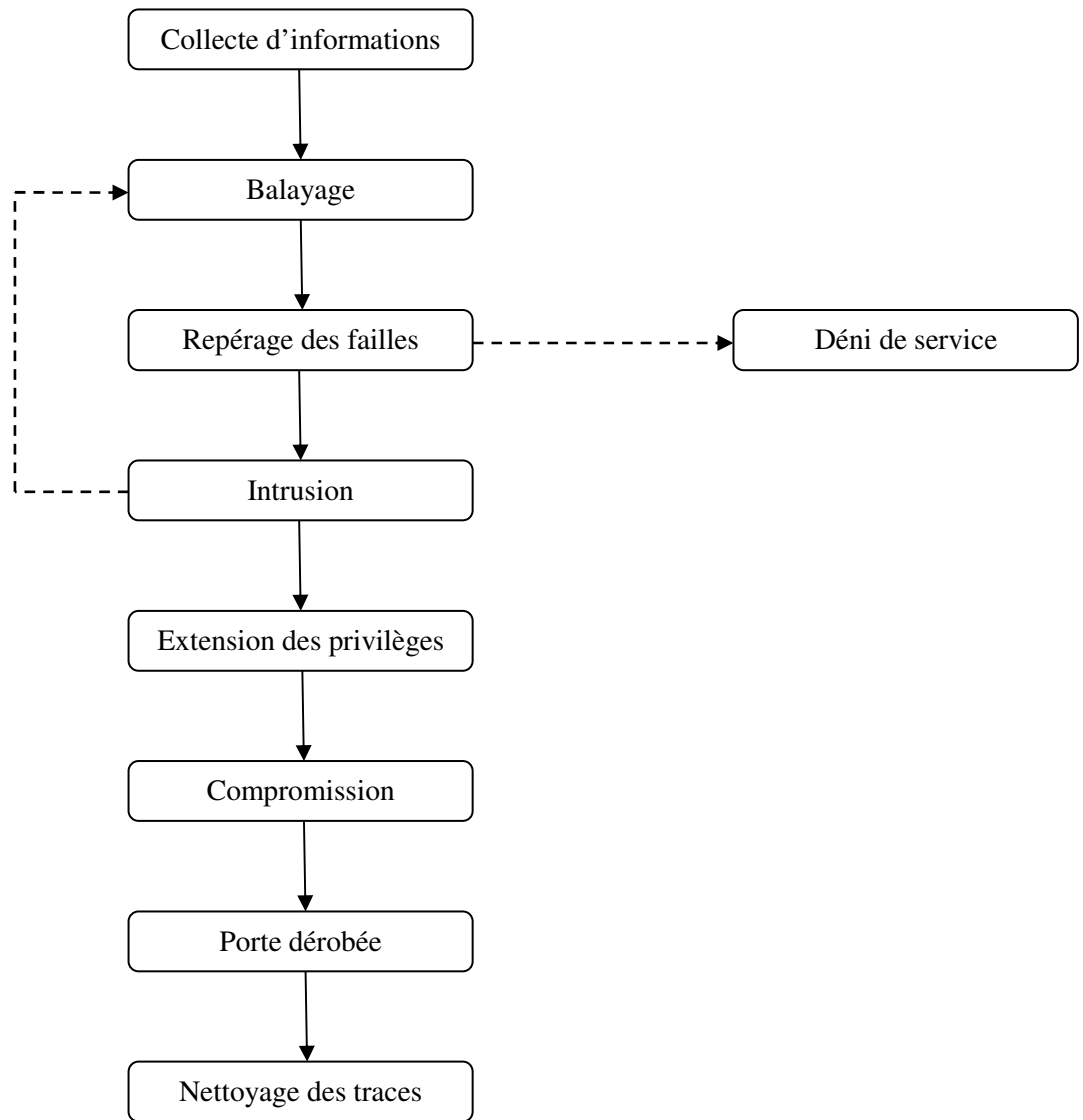


Figure 05 : Méthodologie d'attaque [36].

## **Conclusion du chapitre II**

Dans ce chapitre nous avons abordé quelques notions de base des réseaux comme les types et les topologies des réseaux ainsi que l'architecture client/serveur, c'est l'architecture la plus utilisée dans les réseaux et nous avons conclu que les réseaux sont des éléments importants que les dirigeants des organisations ont besoin afin de relier leurs différents équipements pour la gestion centralisée de l'entreprise. Le mot qui accompagne les réseaux c'est la sécurité de l'information, la sécurité informatique des systèmes d'information est un enjeu d'entreprise primordial actuellement. Nous avons vu qu'il existe de nombreux types d'attaques qui peuvent avoir des impacts très désastreux sur les finances et/ou l'image de marque des entreprises. Les entreprises se doivent de se mettre au norme en mettant en place des politiques de sécurité que cela soit au niveau physique, matérielle, humaine ou logicielle. Il en va de l'intégrité, de la confidentialité, disponibilité, non répudiation et imputation de données pouvant être très sensibles.

Dans le chapitre suivant nous allons parler des étapes utilisées dans le déroulement du projet.

### Chapitre III : Démarche à suivre

#### III.1 Préambule

Après un discours avec le vice doyen de la faculté des sciences exactes et d'informatique nous avons conclu que la problématique de la faculté réside dans la réservation des ressources (enseignants, salles et étudiants) pour les événements (cours, TD et TP), les acteurs principaux qui interviennent à ce processus sont les chefs des départements et le vice doyen, en effet il faut utiliser une application assurant les différentes opérations, et pour éviter de réserver une ressource par plusieurs départements pour des événements qui se font en même temps, il est nécessaire de mettre l'application sous réseau pour faciliter la communication et partager les ressources entre les différents acteurs, donc la réalisation du projet se déroulera comme il montre la figure ci-dessous :

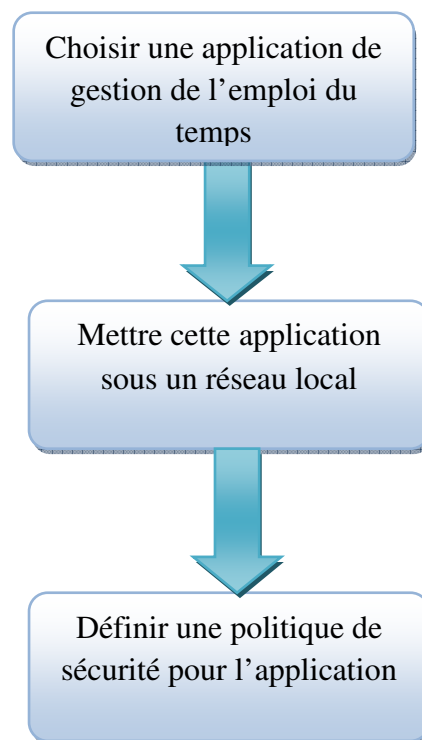


Figure 06 : Les étapes à suivre dans la réalisation du projet.

#### III.2 Choix de l'application

Pour notre projet nous avons décidé d'utiliser l'application RAPLA, c'est une application de gestion de l'emploi du temps universitaire. Ce choix est pour les raisons suivantes :

## Chapitre III : Démarche à suivre

- Cette application est gratuite.
- Les responsables de la faculté veulent l'implémenter.
- Il est possible de la mettre sous réseau.
- Cette application adopte une stratégie de sécurité conforme à notre stratégie

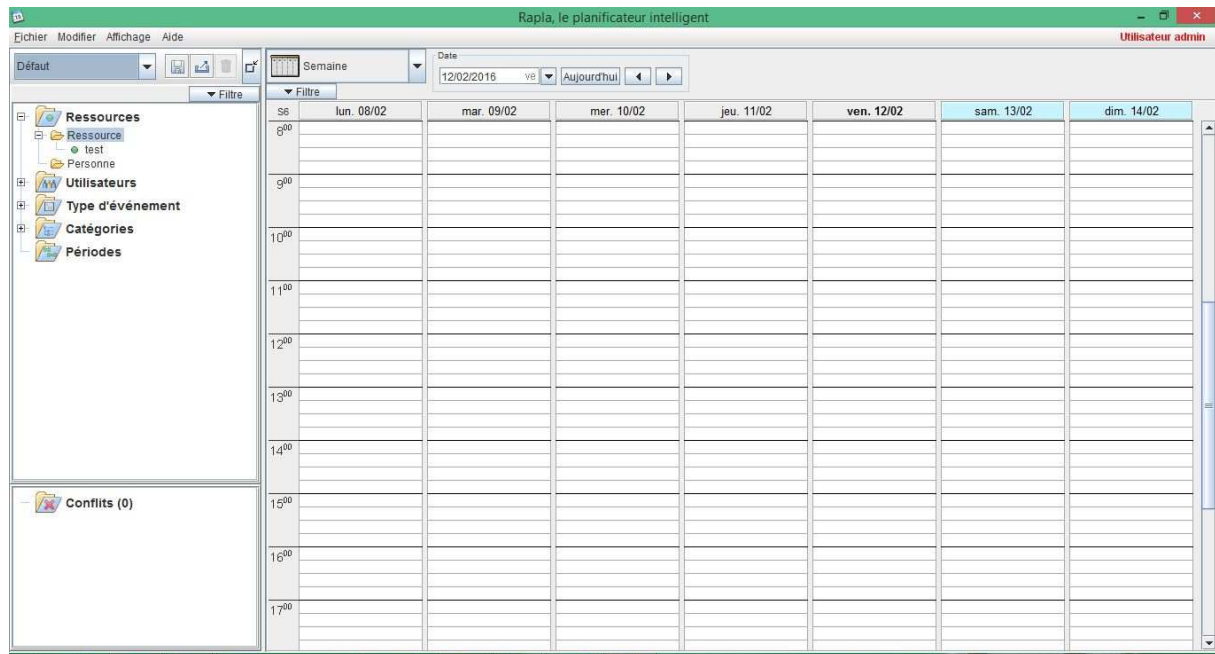


Figure 07 : Interface de l'application RAPLA.

### III.2.1 Caractéristiques de RAPLA

#### III.2.1.1 La base de données

L'application RAPLA utilise une base de données sous forme d'un fichier XML, en plus elle offre à son utilisateur d'autres types de base de données comme MySQL, Postgresql, Oracle [37].

# Chapitre III : Démarche à suivre

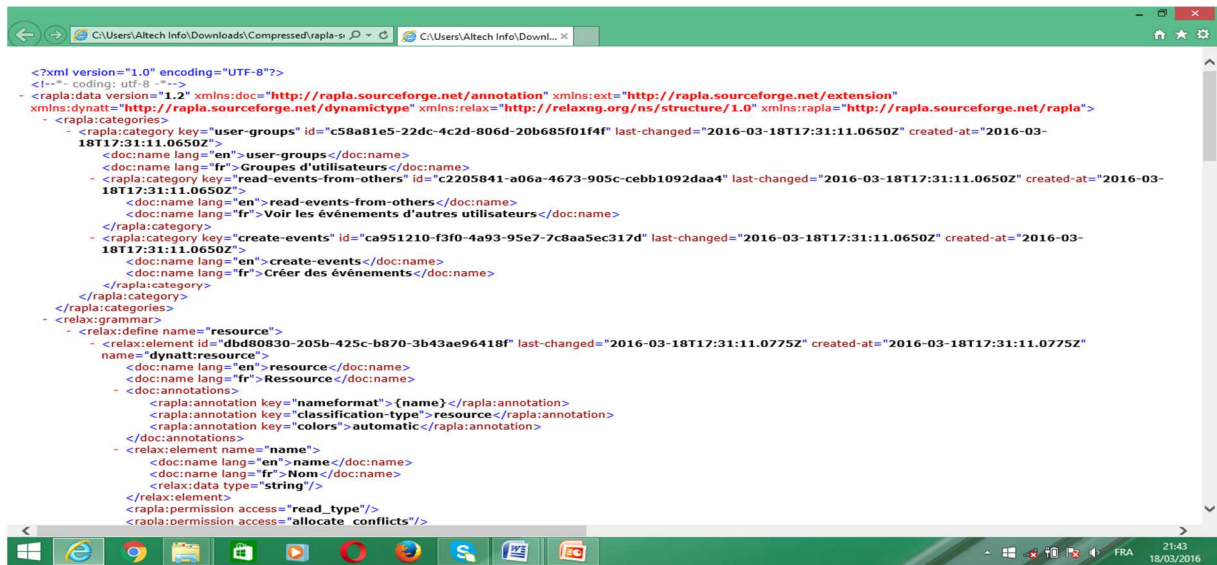


Figure 08 : Base de données de RAPLA.

## III.2.1.2 Gestion du conflit

Lorsqu'un utilisateur veut créer un nouvel événement, dans la fenêtre de choix des ressources RAPLA indique à cet utilisateur quelles ressources sont libres et quelles ressources sont réservées dans l'heure à laquelle il veut créer l'événement [37].

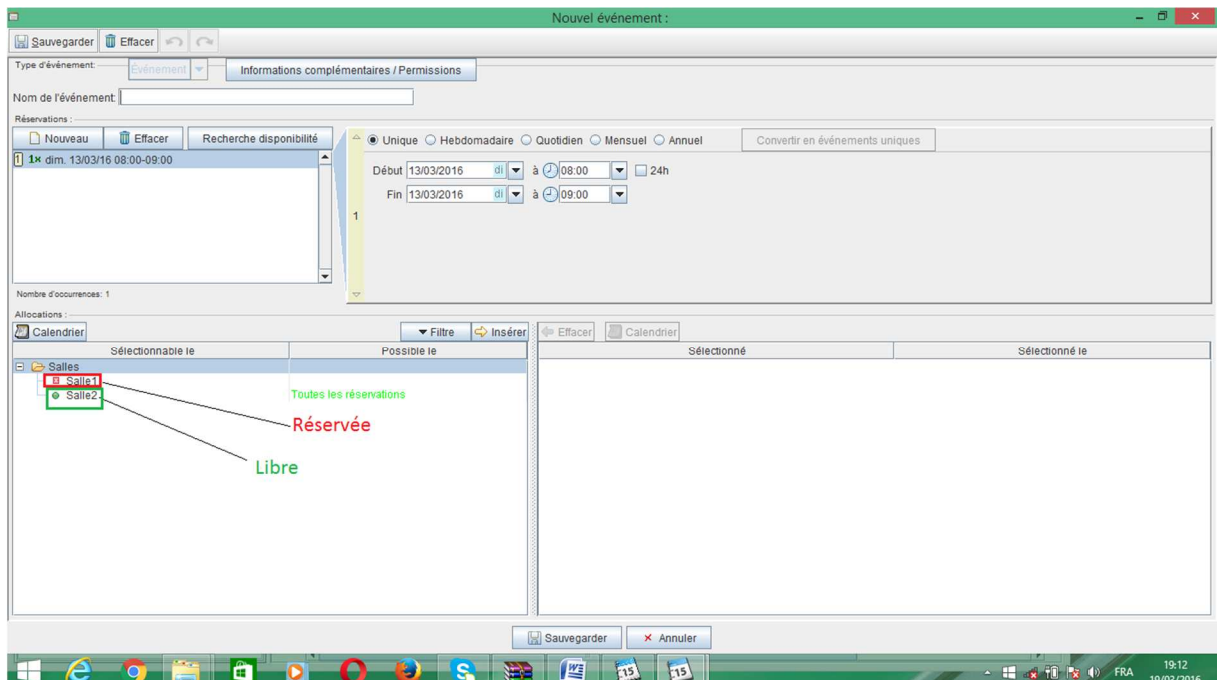


Figure 09 : fenêtre de création d'un événement.

# Chapitre III : Démarche à suivre

## III.2.1.3 Publication en HTML

Un utilisateur de RAPLA a la possibilité de publier un emploi du temps sous forme de HTML pour qu'un enseignant ou un étudiant puisse le consulter en ligne [ ].

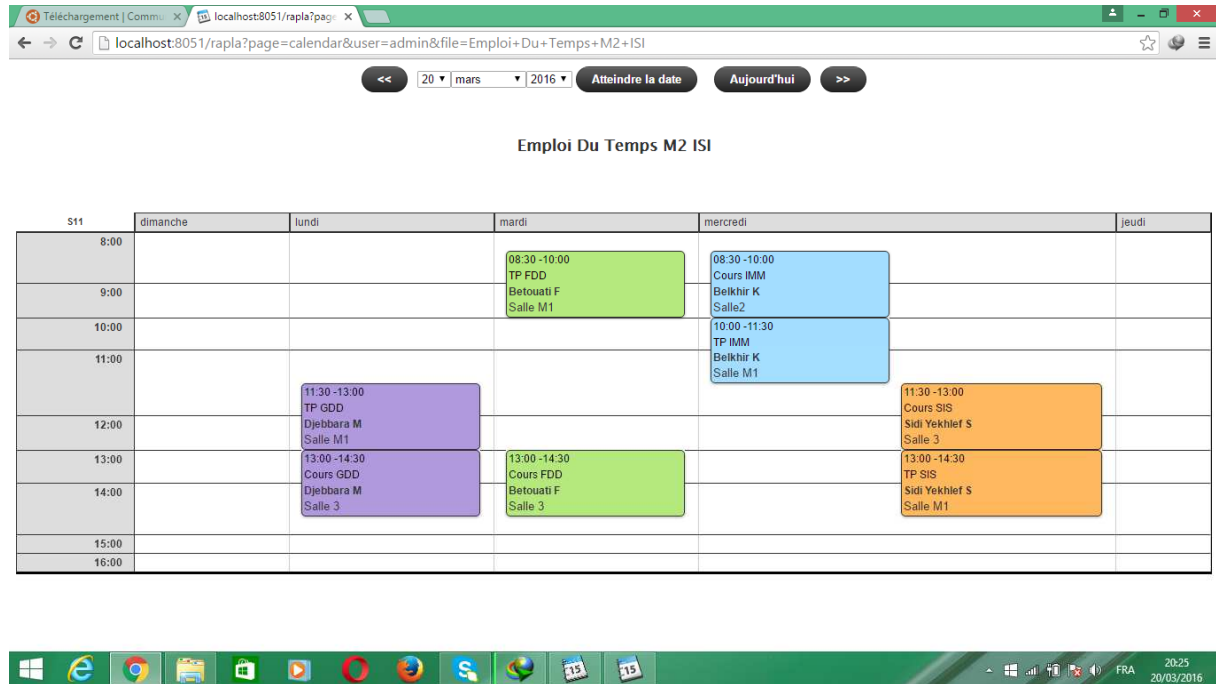


Figure 10 : Consultation de l'emploi du temps en ligne.

## III.2.1.4 Groupes d'utilisateurs

Lors de la création d'un utilisateur RAPLA propose à l'administrateur des groupes d'utilisateurs auxquels il veut inclure cet utilisateur, dans RAPLA il ya deux types de groupes d'utilisateurs le premier est voir les événements des autres utilisateurs, et le deuxième est créer des événements. L'administrateur peut choisir un des deux ou les deux [37].

## Chapitre III : Démarche à suivre

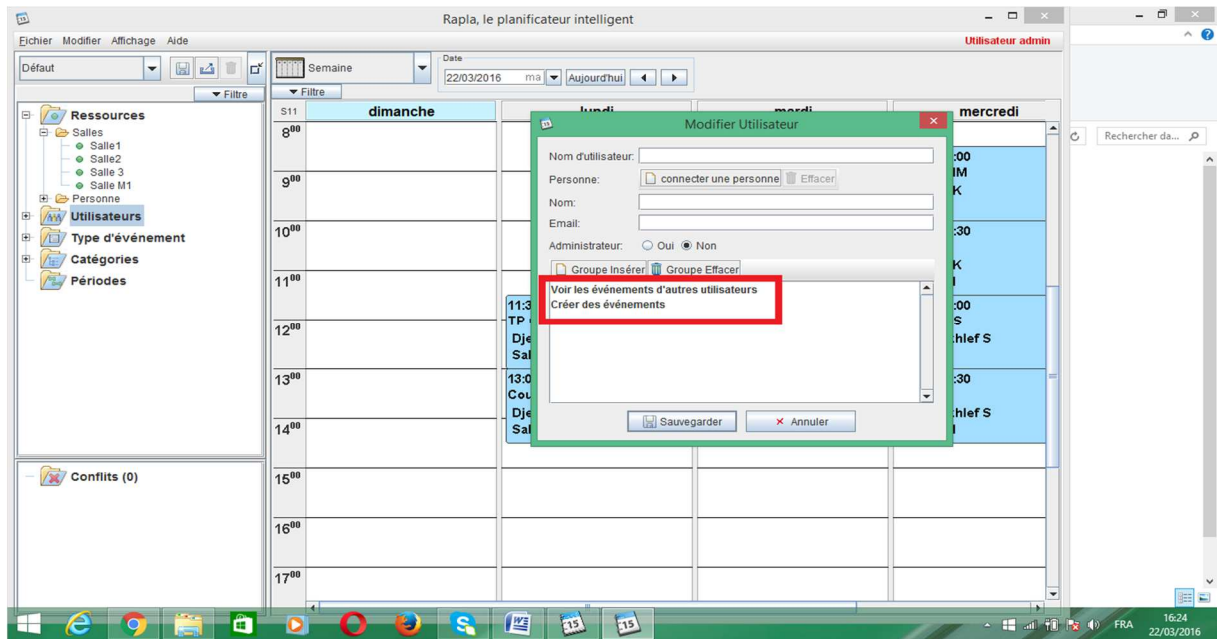
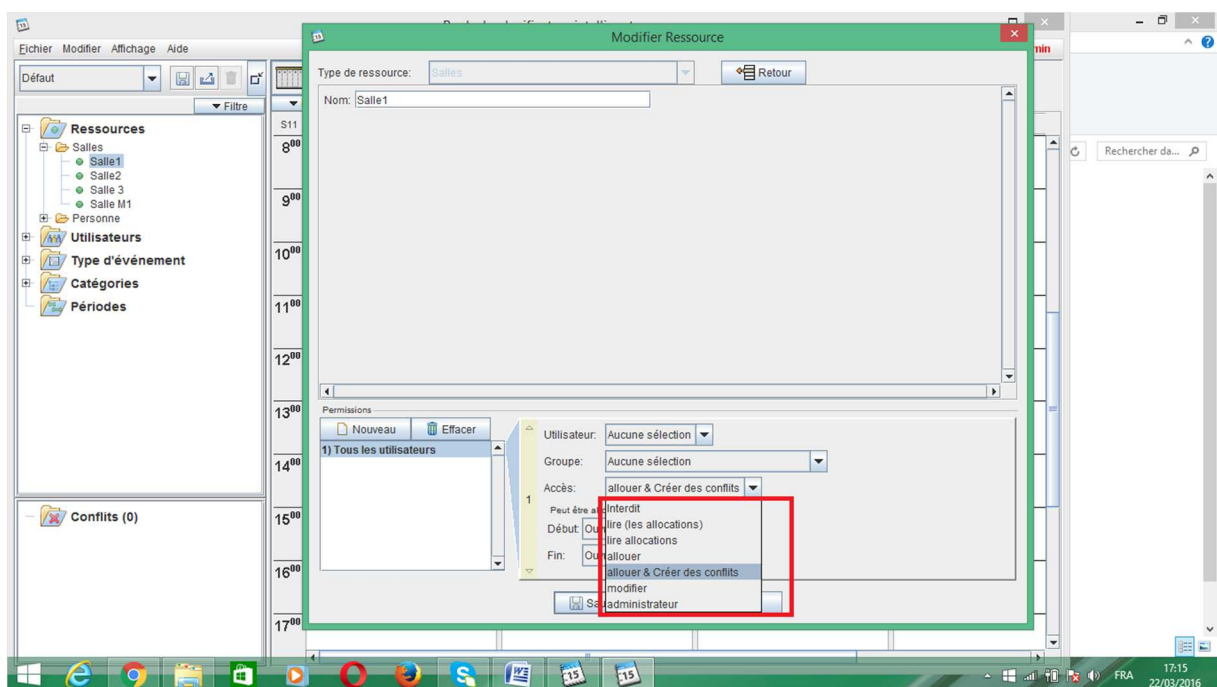


Figure 11 : choix des groupes d'utilisateurs.

### III.2.1.5 Les permissions

Avec RAPLA on des autorisations pour chaque ressource lors de sa création (ou après). Une autorisation à un utilisateur ou groupe, on peut également spécifier plus d'une autorisation par utilisateur ou groupe, si un utilisateur tente d'obtenir l'accès, toutes les autorisations seront vérifiées et si une autorisation correspond à l'accès spécifié dans cette autorisation sera accordée à l'utilisateur. Par exemple, si on refuse une autorisation d'accès pour l'utilisateur et une autre autorisation permet l'accès, l'utilisateur aura l'accès. Il existe différents niveaux d'accès, on peut définir pour chaque autorisation : refuser, peut lire, peut allouer, peut allouer & créer des conflits et peut modifier [37].





## Chapitre III : Démarche à suivre

Figure 12 : affectation des autorisations.

En plus de ces caractéristiques, il y a une possibilité d'accéder à RAPLA à partir d'une autre application java à travers RAPLA-API et la façade RAPLA [37].

### III.3 Mettre l'application sous un réseau

Le réseau utilisé est de type local qui interconnecte les machines des différents acteurs qui interviennent au processus de génération de l'emploi du temps qui sont principalement le vice doyen et les chefs de départements.

L'application RAPLA utilise une architecture client/serveur pour le mode multi utilisateurs mais en local host, on va profiter ce point et rendre l'appel du serveur à distance.

Il y a deux programmes RaplaClient et RaplaServer, le programme RaplaServeur va être hébergé dans un serveur et RaplaClient va être installé sur chaque machine des acteurs.

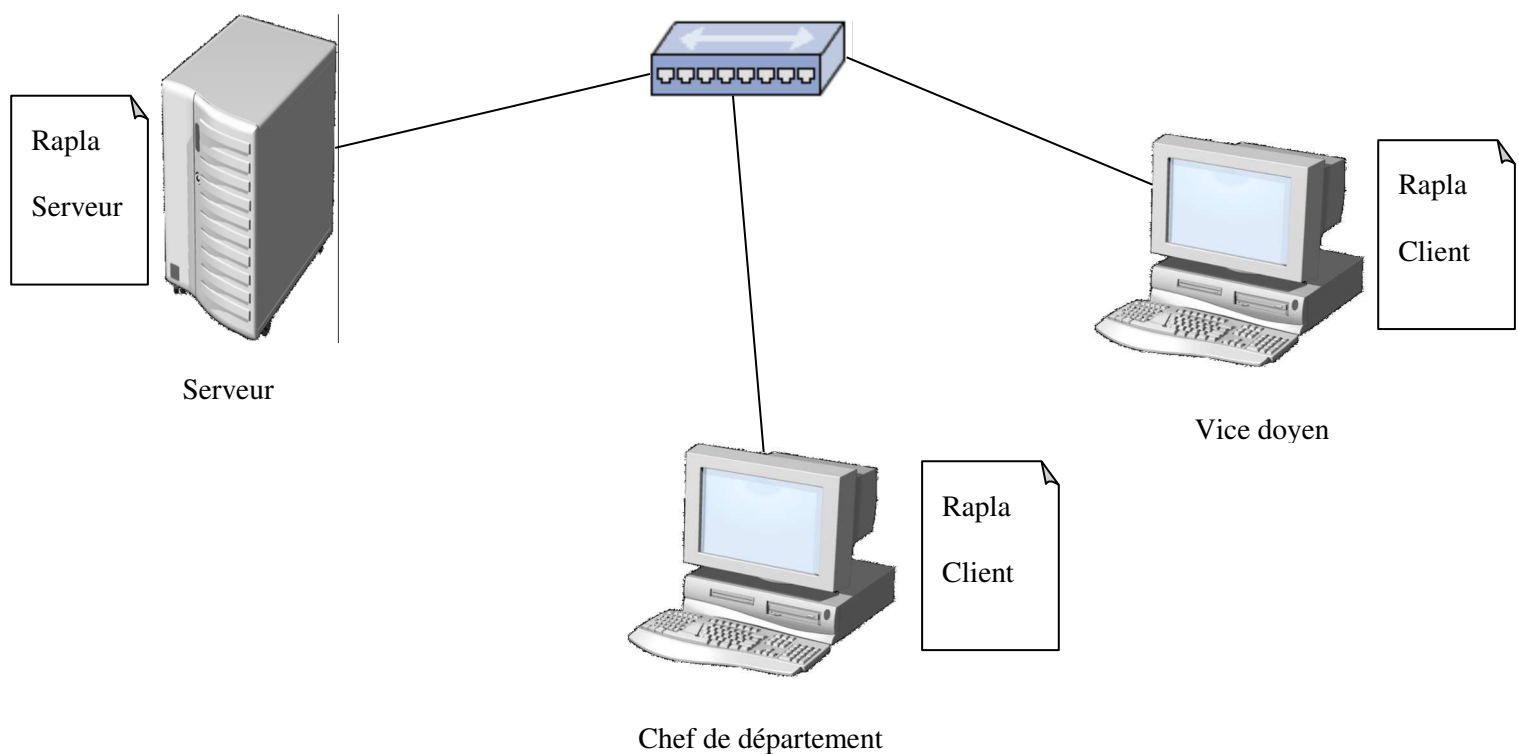


Figure 13 : La structure du réseau.

#### III.3.1 Intérêt du réseau

## Chapitre III : Démarche à suivre

Une fois l'application est mise sous un réseau les créateurs des emplois du temps vont partager les mêmes ressources et les emplois du temps générés vont être fiables et ne contenant aucun conflit sur les ressources, car le réseau aide le créateur de l'emploi du temps de connaître l'état de la ressource (réservée ou libre) avant qu'il la réserve, en effet ce créateur va être certain dans ses choix.

### III.4 Politique de sécurité

L'application RAPLA utilise la stratégie de sécurité suivante :

Le vice doyen accède en utilisant son nom d'utilisateur et son mot de passe comme administrateur, il peut ajouter, modifier ou supprimer des événements, il peut même lire les événements des autres utilisateurs et c'est lui qui génère les noms d'utilisateur et mots de passe des autres utilisateurs.

Le chef de département accède en utilisant son nom d'utilisateur et son mot de passe, il peut également ajouter des événements, mais il ne peut modifier, supprimer ou lire que ses événements.

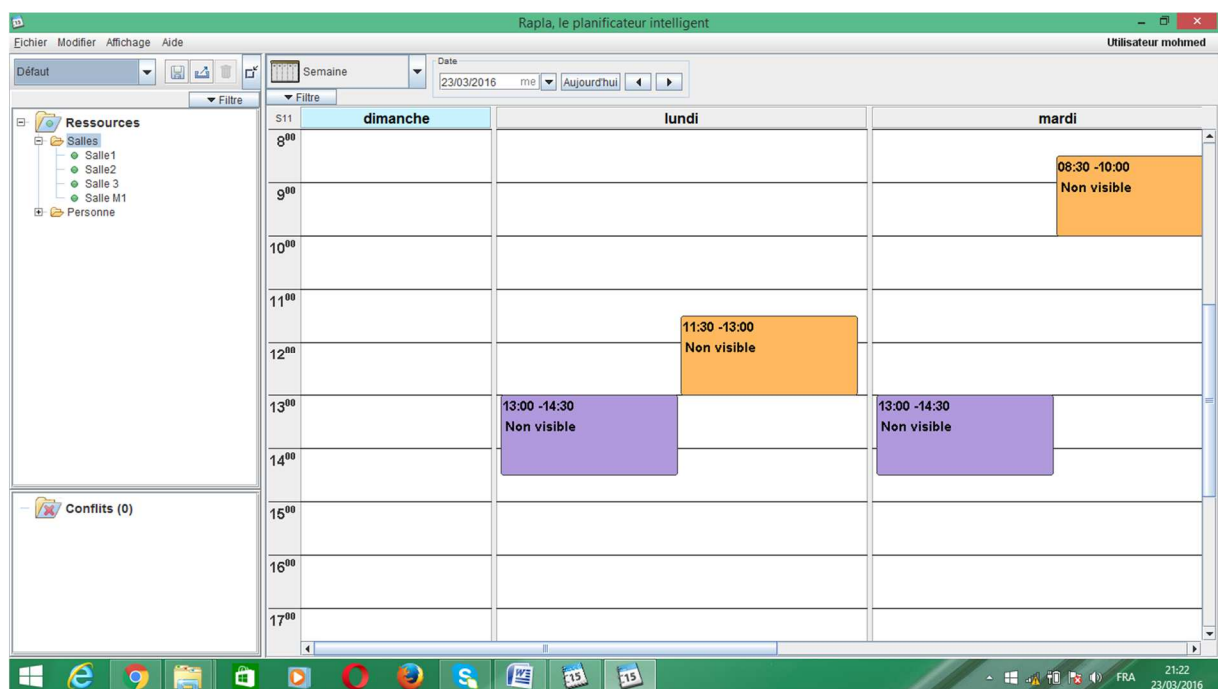


Figure 14 : Politique de sécurité de RAPLA.

### **Conclusion du chapitre III**

Dans ce chapitre nous avons dessiné une trajectoire par laquelle la réalisation du projet va passer, ainsi que on a étudié les caractéristiques de l'application RAPLA et comment cette application va être mise sous un réseau. On a étudié également la structure du réseau sur lequel l'application va être utilisée et la politique de sécurité que l'application adopte.

Nous avons conclu que l'application RAPLA est une meilleure solution pour résoudre le problème de l'emploi du temps car elle répond aux besoins des responsables de la faculté des sciences exactes et d'informatique.

Dans le chapitre suivant nous allons aborder l'implémentation de l'application. Nous allons parler de ce qu'on a modifié dans l'application.

## Chapitre IV : Implémentation de l'application

### IV.1 Préambule

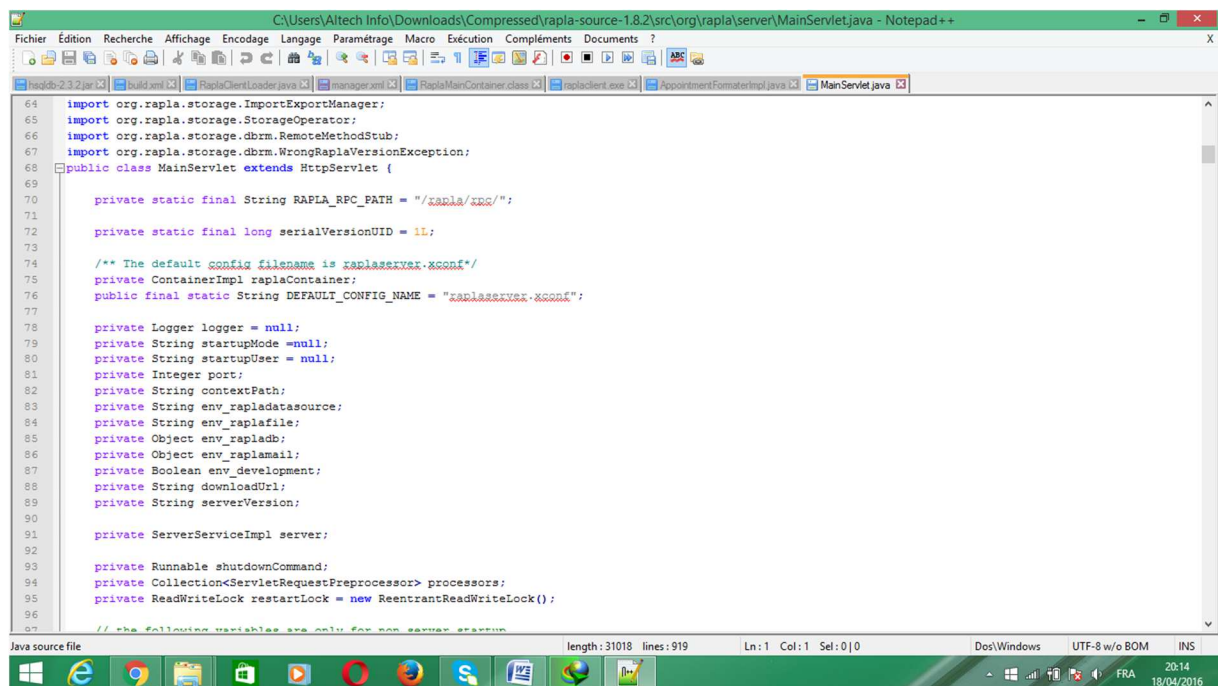
Comme nous l'avons déjà dit dans les chapitres précédents, nous avons choisi l'application RAPLA pour la gestion de l'emploi du temps de la faculté des sciences exactes et d'informatique, donc nous allons télécharger le code sources de cette application et faire les modifications nécessaires.

A travers ce dernier chapitre nous allons montrer les classes java modifiées, et puisque cette application va être sous un réseau on va expliquer les configurations nécessaires pour le serveur, ainsi que nous allons ajouter un guide pour l'utilisateur de RAPLA.

### IV.2 Modification de l'application

D'après les responsables de la faculté, ses problèmes avec RAPLA sont la mise de l'application sous un réseau et l'affichage, donc nous allons modifier les classes responsables de ça.

- **MainServlet.java** : cette classe faire la communication entre le client et le serveur.

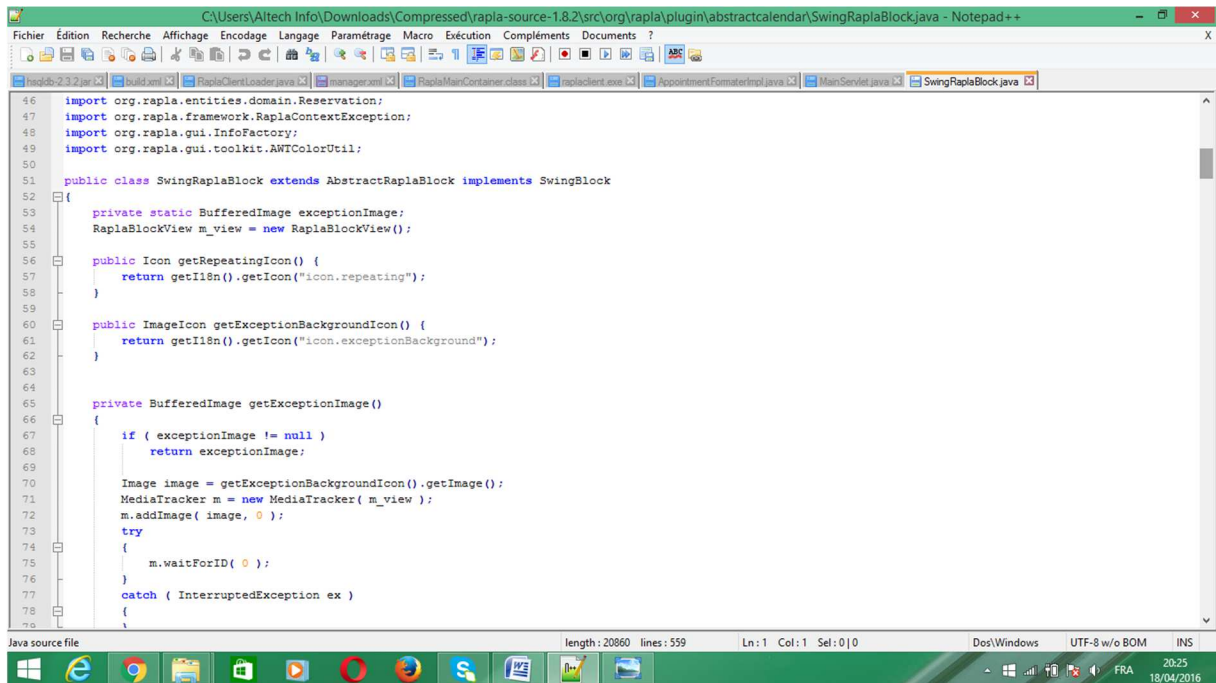


```
64 import org.rapla.storage.ImportExportManager;
65 import org.rapla.storage.StorageOperator;
66 import org.rapla.storage.dbm.RemoteMethodStub;
67 import org.rapla.storage.dbm.WrongRaplaVersionException;
68 public class MainServlet extends HttpServlet {
69
70     private static final String RAPLA_RPC_PATH = "/rapla/rpc/";
71
72     private static final long serialVersionUID = 1L;
73
74     /** The default config filename is raplaserver.xconf */
75     private ContainerImpl raplaContainer;
76     public final static String DEFAULT_CONFIG_NAME = "raplaserver.xconf";
77
78     private Logger logger = null;
79     private String startupMode = null;
80     private String startupUser = null;
81     private Integer port;
82     private String contextPath;
83     private String env_rapladataSource;
84     private String env_raplafile;
85     private Object env_rapladb;
86     private Object env_raplamail;
87     private Boolean env_development;
88     private String downloadUrl;
89     private String serverVersion;
90
91     private ServerServiceImpl server;
92
93     private Runnable shutdownCommand;
94     private Collection<ServletRequestPreprocessor> processors;
95     private ReadWriteLock restartLock = new ReentrantReadWriteLock();
96
97     // the following variables are only for non server startup
```

Figure 15: la classe MainServlet.java

- **SwingRaplaBlocks.java** : c'est la classe responsable de la forme du bloc qui contient les détails des événements (le temps de l'événement, nom de l'enseignant, nom de la salle, le module).

# Chapitre IV : Implémentation de l'application

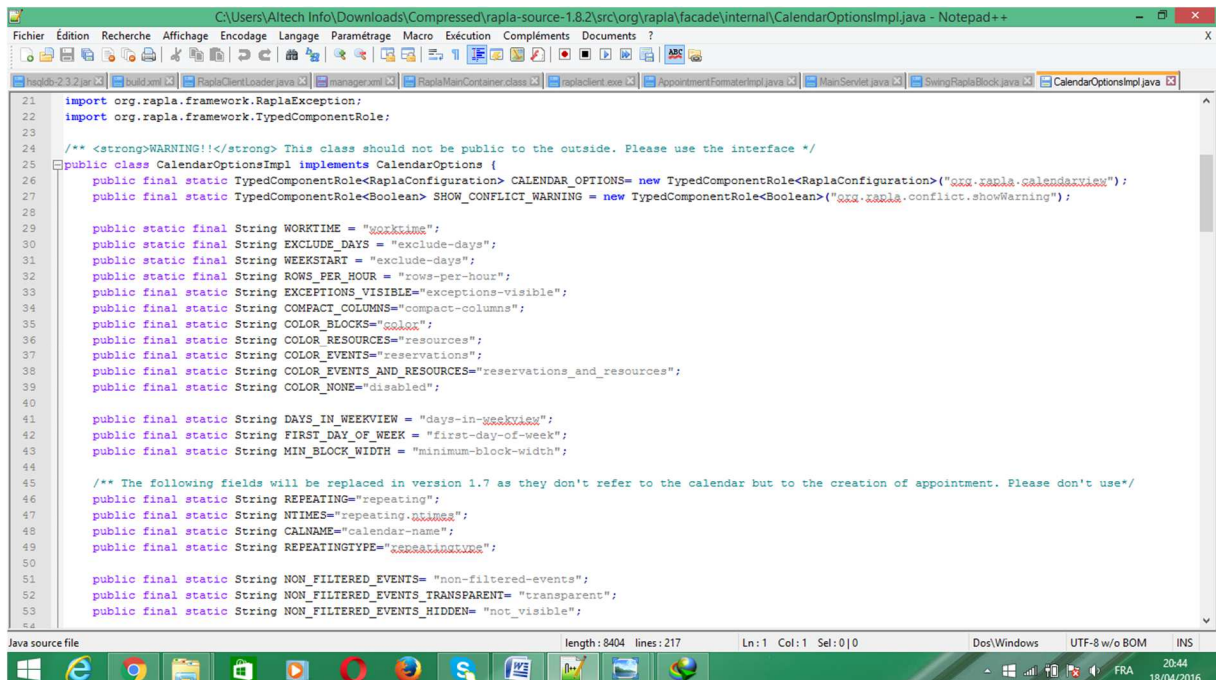


```
46 import org.rapla.entities.domain.Reservation;
47 import org.rapla.framework.RaplaContextException;
48 import org.rapla.gui.InfoFactory;
49 import org.rapla.gui.toolkit.AWTColorUtil;
50
51 public class SwingRaplaBlock extends AbstractRaplaBlock implements SwingBlock
52 {
53     private static BufferedImage exceptionImage;
54     RaplaBlockView m_view = new RaplaBlockView();
55
56     public Icon getRepeatingIcon() {
57         return getIi8n().getIcon("icon.repeating");
58     }
59
60     public ImageIcon getExceptionBackgroundIcon() {
61         return getIi8n().getIcon("icon.exceptionBackground");
62     }
63
64
65     private BufferedImage getExceptionImage()
66     {
67         if ( exceptionImage != null )
68             return exceptionImage;
69
70         Image image = getExceptionBackgroundIcon().getImage();
71         MediaTracker m = new MediaTracker( m_view );
72         m.addImage( image, 0 );
73         try
74         {
75             m.waitForID( 0 );
76         }
77         catch ( InterruptedException ex )
78         {
79
80         }
81     }
82 }
```

Figure 16: la classe SwingRaplaBlock.java.

Les classes qui suivent sont responsables de la forme du calendrier

- **CalendarOptionsImpl.java :**

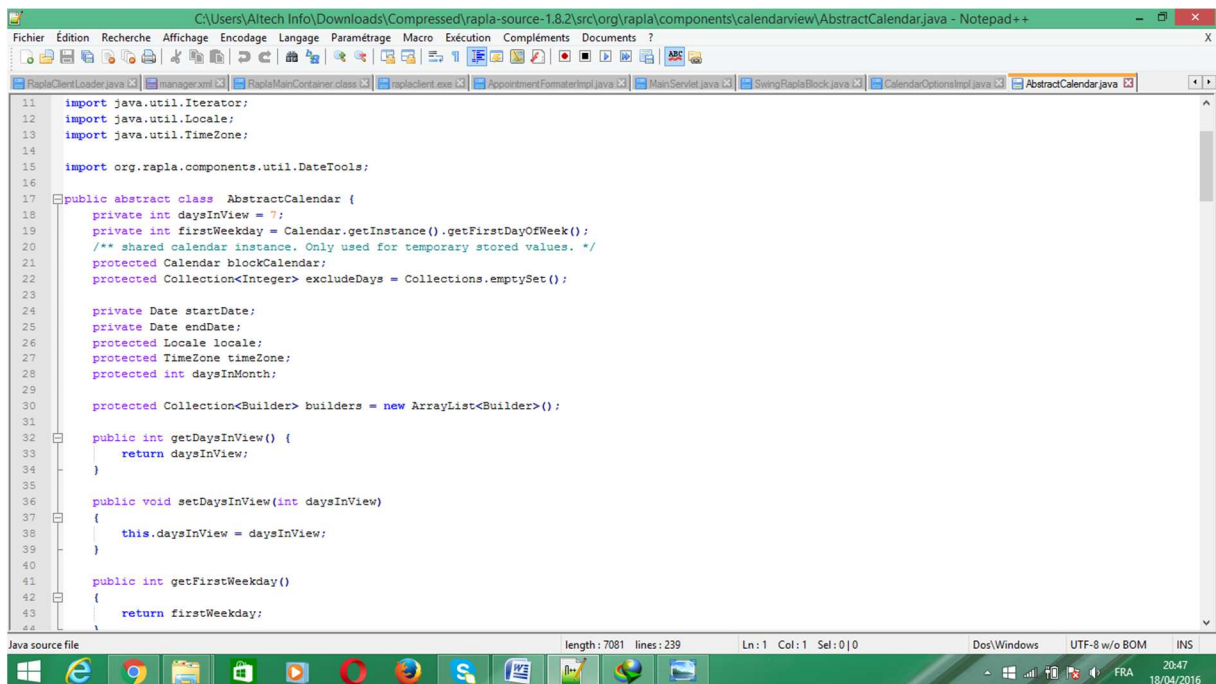


```
21 import org.rapla.framework.RaplaException;
22 import org.rapla.framework.TypedComponentRole;
23
24 /** <strong>WARNING!</strong> This class should not be public to the outside. Please use the interface */
25 public class CalendarOptionsImpl implements CalendarOptions {
26     public final static TypedComponentRole<RaplaConfiguration> CALENDAR_OPTIONS= new TypedComponentRole<RaplaConfiguration>("org.rapla.calendar.options");
27     public final static TypedComponentRole<Boolean> SHOW_CONFLICT_WARNING = new TypedComponentRole<Boolean>("org.rapla.conflict.showWarning");
28
29
30     public static final String WORKTIME = "worktime";
31     public static final String EXCLUDE_DAYS = "exclude-days";
32     public static final String WEEKSTART = "exclude-days";
33     public static final String ROWS_PER_HOUR = "rows-per-hour";
34     public static final String EXCEPTIONS_VISIBLE="exceptions-visible";
35     public static final String COMPACT_COLUMNS="compact-columns";
36     public static final String COLOR_BLOCKS="color";
37     public static final String COLOR_RESOURCES="resources";
38     public static final String COLOR_EVENTS="reservations";
39     public static final String COLOR_EVENTS_AND_RESOURCES="reservations_and_resources";
40     public static final String COLOR_NONE="disabled";
41
42     public static final String DAYS_IN_WEEKVIEW = "days-in-weekview";
43     public static final String FIRST_DAY_OF_WEEK = "first-day-of-week";
44     public static final String MIN_BLOCK_WIDTH = "minimum-block-width";
45
46     /** The following fields will be replaced in version 1.7 as they don't refer to the calendar but to the creation of appointment. Please don't use*/
47     public final static String REPEATING="repeating";
48     public final static String NTIMES="repeating-ntimes";
49     public final static String CALNAME="calendar-name";
50     public final static String REPEATINGTYPE="repeating-type";
51
52     public final static String NON_FILTERED_EVENTS= "non-filtered-events";
53     public final static String NON_FILTERED_EVENTS_TRANSPARENT= "transparent";
54     public final static String NON_FILTERED_EVENTS_HIDDEN= "not-visible";
55 }
```

Figure 17: la classe CalendarOptionsImpl.java.

## Chapitre IV : Implémentation de l'application

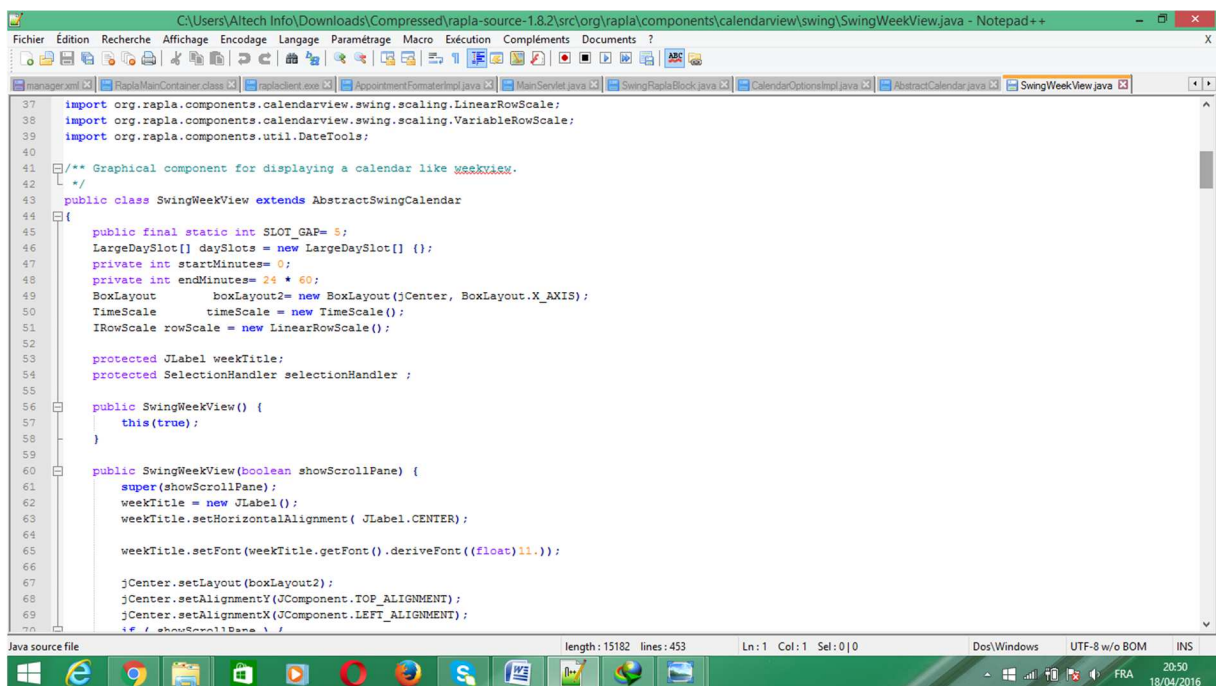
### • AbstractCalendar.java :



```
11 import java.util.Iterator;
12 import java.util.Locale;
13 import java.util.TimeZone;
14
15 import org.zapla.components.util.DateTools;
16
17 public abstract class AbstractCalendar {
18     private int daysInView = 7;
19     private int firstWeekday = Calendar.getInstance().getFirstDayOfWeek();
20     /** shared calendar instance. Only used for temporary stored values. */
21     protected Calendar blockCalendar;
22     protected Collection<Integer> excludeDays = Collections.emptySet();
23
24     private Date startDate;
25     private Date endDate;
26     protected Locale locale;
27     protected TimeZone timeZone;
28     protected int daysInMonth;
29
30     protected Collection<Builder> builders = new ArrayList<Builder>();
31
32     public int getDaysInView() {
33         return daysInView;
34     }
35
36     public void setDaysInView(int daysInView)
37     {
38         this.daysInView = daysInView;
39     }
40
41     public int getFirstWeekday()
42     {
43         return firstWeekday;
44     }
45 }
```

Figure 18: la classe AbstractCalendar.java.

### • SwingWeekView.java :

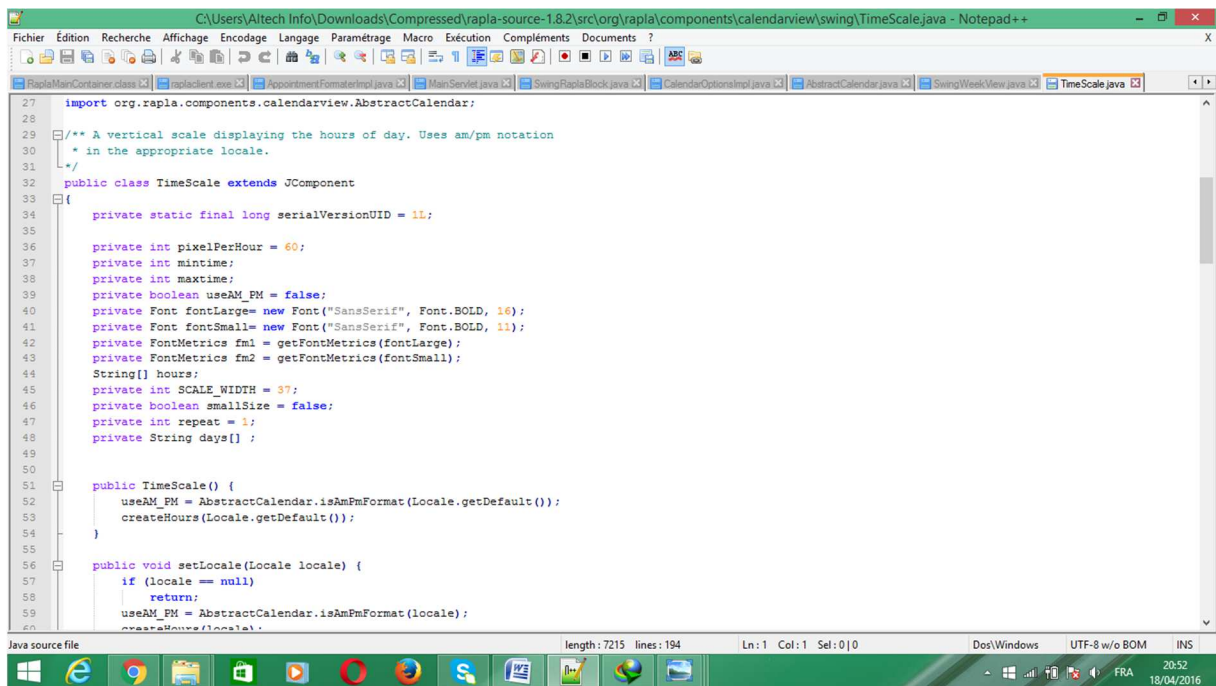


```
37 import org.zapla.components.calendarview.swing.scaling.LinearRowScale;
38 import org.zapla.components.calendarview.swing.scaling.VariableRowScale;
39 import org.zapla.components.util.DateTools;
40
41 /** Graphical component for displaying a calendar like weekView.
42 */
43 public class SwingWeekView extends AbstractSwingCalendar
44 {
45     public final static int SLOT_GAP= 5;
46     LargeDaySlot[] daysSlots = new LargeDaySlot[] {};
47     private int startMinutes= 0;
48     private int endMinutes= 24 * 60;
49     BorderLayout boxLayout2= new BorderLayout(JCenter, BorderLayout.X_AXIS);
50     TimeScale timeScale = new TimeScale();
51     IRowScale rowScale = new LinearRowScale();
52
53     protected JLabel weekTitle;
54     protected SelectionHandler selectionHandler ;
55
56     public SwingWeekView() {
57         this(true);
58     }
59
60     public SwingWeekView(boolean showScrollPane) {
61         super(showScrollPane);
62         weekTitle = new JLabel();
63         weekTitle.setHorizontalAlignment(JLabel.CENTER);
64
65         weekTitle.setFont(weekTitle.getFont().deriveFont((float)11.));
66
67         JCenter.setLayout(boxLayout2);
68         JCenter.setAlignmentY(JComponent.TOP_ALIGNMENT);
69         JCenter.setAlignmentX(JComponent.LEFT_ALIGNMENT);
70         if (showScrollPane) {
71             JScrollPane scrollPane = new JScrollPane(JCenter);
72             scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
73             scrollPane.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_NEVER);
74             scrollPane.setBorder(BorderFactory.createEmptyBorder());
75             scrollPane.getViewport().setView(JCenter);
76         }
77     }
78 }
```

Figure 19: la classe SwingWeekView.java.



### • TimeScale.java :



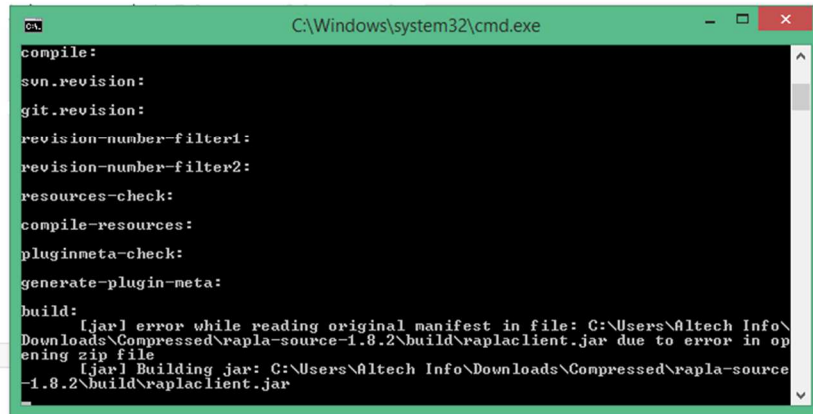
```
27 import org.rapla.components.calendarview.AbstractCalendar;
28
29 /** A vertical scale displaying the hours of day. Uses am/pm notation
30  * in the appropriate locale.
31  */
32 public class TimeScale extends JComponent
33 {
34     private static final long serialVersionUID = 1L;
35
36     private int pixelPerHour = 60;
37     private int mintime;
38     private int maxtime;
39     private boolean useAM_PM = false;
40     private Font fontLarge= new Font("SansSerif", Font.BOLD, 16);
41     private Font fontSmall= new Font("SansSerif", Font.BOLD, 11);
42     private FontMetrics fm1 = getFontMetrics(fontLarge);
43     private FontMetrics fm2 = getFontMetrics(fontSmall);
44     String[] hours;
45     private int SCALE_WIDTH = 37;
46     private boolean smallSize = false;
47     private int repeat = 1;
48     private String days[] ;
49
50
51     public TimeScale() {
52         useAM_PM = AbstractCalendar.isAmPmFormat(Locale.getDefault());
53         createHours(Locale.getDefault());
54     }
55
56     public void setLocale(Locale locale) {
57         if (locale == null)
58             return;
59         useAM_PM = AbstractCalendar.isAmPmFormat(locale);
60         createHours(locale);
61     }
62 }
```

Figure 20: la classe TimeScale.java.

## IV.3 Utilisation de l'application

### IV.3.1 Compilation de l'application

L'application RAPLA se trouve avec deux types, RAPLA-Binary c'est le logiciel exécutable qui peut être utilisé directement, et le deuxième est RAPLA-Source c'est le code source de l'application, pour les utilisateurs qui ont le deuxième type ils doivent faire la compilation avant l'utilisation. La compilation se fait par un fichier appelé build.bat, après l'exécution de ce fichier le logiciel exécutable va être créé c'est-à-dire RAPLA-Binary.



```
C:\Windows\system32\cmd.exe
compile:
sun.revision:
git.revision:
revision-number-filter1:
revision-number-filter2:
resources-check:
compile-resources:
pluginmeta-check:
generate-plugin-meta:
build:
[jar] error while reading original manifest in file: C:\Users\Altech Info\Downloads\Compressed\rapla-source-1.8.2\build\raplaclient.jar due to error in opening zip file
[jar] Building jar: C:\Users\Altech Info\Downloads\Compressed\rapla-source-1.8.2\build\raplaclient.jar
```

Figure 21: L'exécution du fichier build.bat.

### IV.3.2 Authentification

Tous Les utilisateurs soit l'administrateur ou les utilisateurs normaux connecte à RAPLA de la même manière en introduisant ses noms d'utilisateur et ses mots de passe.



Figure 22: Connexion à l'application RAPLA.

### IV.3.3 Vue principale

Une fois qu'on a connecté à Rapla, on arrivera à la fenêtre principale. La vue principale est divisée en trois parties: la zone de menu, la zone de sélection sur le côté gauche, et la grande région de calendrier dans le centre. Dans la zone de sélection de la fonction, on trouvera toutes les fonctions disponibles Rapla. Si on clique sur l'un d'entre eux l'aire de travail va changer à la vue appropriée.



## Chapitre IV : Implémentation de l'application

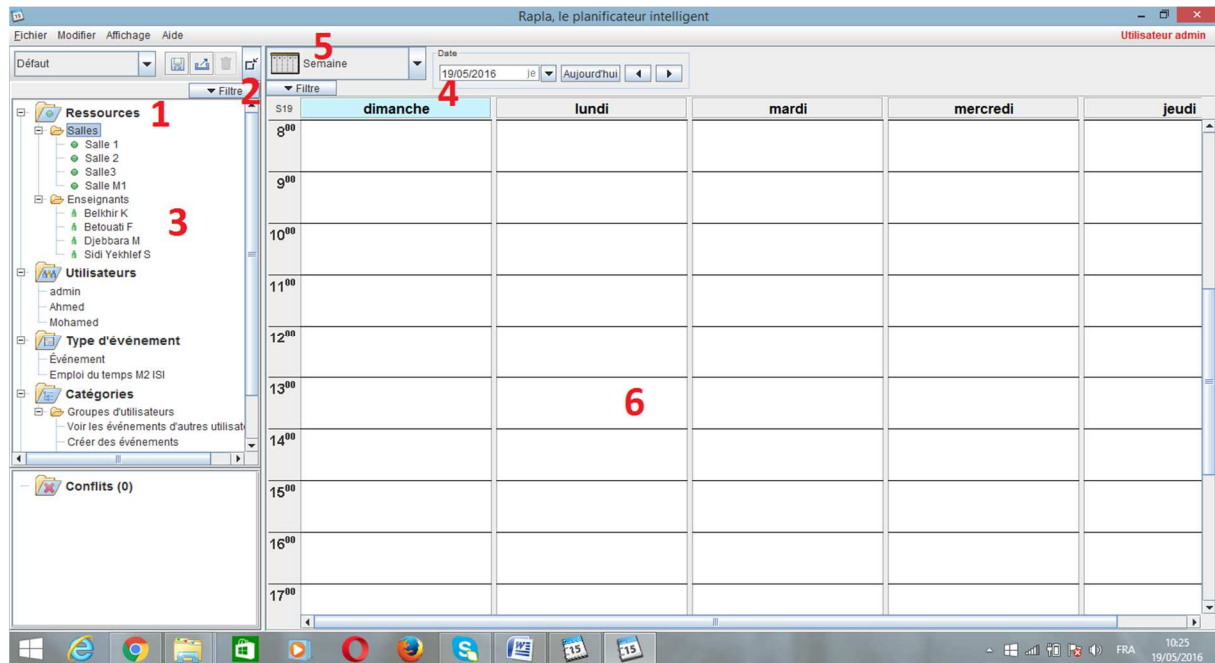


Figure 23: La vue principale de RAPLA.

(1) : montre dans l'arborescence de sélection toutes les ressources filtrées, En conséquence les événements filtrés.

(2) : ici, on peut spécifier, quelle ressources / personnes / événements sont affichés dans l'arborescence. On trouve plus de détails dans le cadre du filtre.

(3) : on choisit les objets à partir de l'arbre, qui devrait être visible dans le calendrier. Si on marque un nœud père (salles par exemple), tous les objets sous-jacents (dans ce cas, toutes les pièces) sont indiqués. Avec un clic-droit on active un menu contextuel avec des actions d'objet dépendant, comme modifier / supprimer ou créer de nouveaux.

(4) : Avec la sélection de date, on peut changer dans une autre semaine de calendrier, en entrant une nouvelle date ou en cliquant sur les flèches gauche ou droite.

(5) : La sélection de vue nous permet de basculer entre les différentes vues dans Rapla, comme le mois, semaine, jour, ressources / jour ou de tableau.

(6) : dans le domaine de calendrier, on peut créer de nouveaux événements, modifier ceux qui existent déjà et de modifier l'allocation des ressources.

Dans le coin en bas à gauche, on trouvera la vue des conflits. Les conflits sont quelque chose qu'on veut éviter, mais de temps en temps les conflits se développent comme lorsque deux personnes réservés la même pièce pour la même période. La vue des conflits offre des moyens de reconnaître et d'éliminer les conflits. Plus d'informations dans les conflits.

### IV.3.4 Gestion des utilisateurs

La création des utilisateurs et l'attribution de leurs mots est faite par l'administrateur, quand ce dernier clique droit sur le nœud utilisateur dans l'arborescence une boîte de dialogue s'ouvre en demandant la saisie d'un nom d'utilisateur, après la saisie du nom d'utilisateur et la confirmation une autre boîte de dialogue s'ouvre en demandant la saisie du mot de passe.

## Chapitre IV : Implémentation de l'application

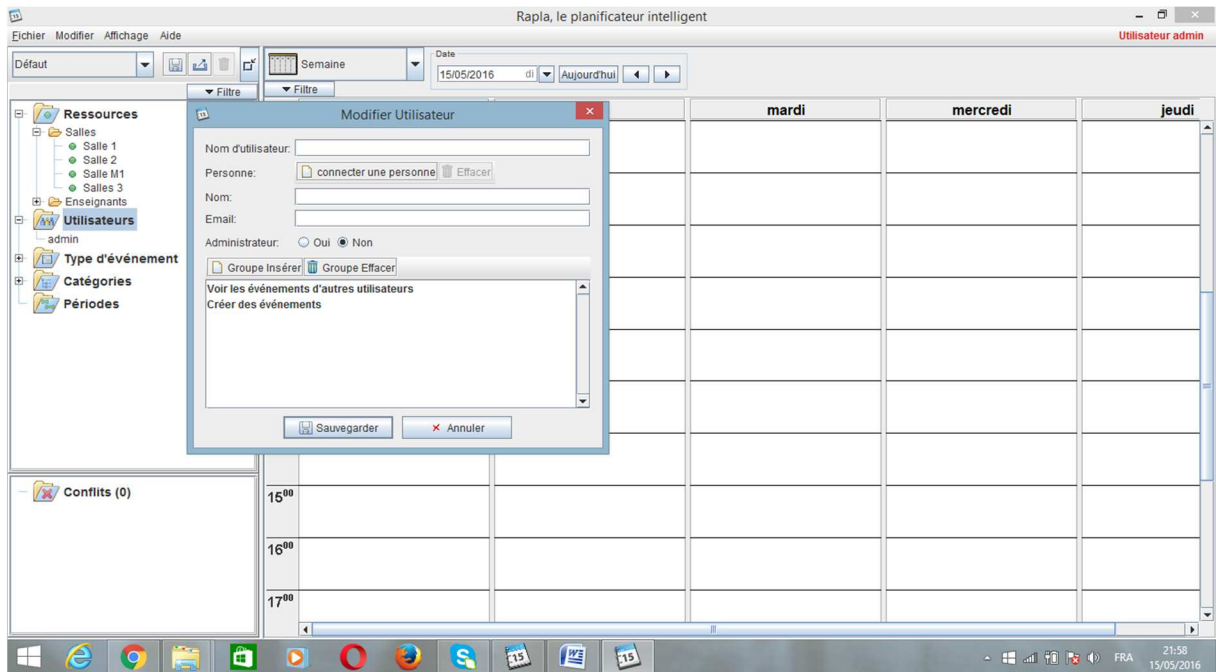


Figure 24: la saisie du nom d'utilisateur.

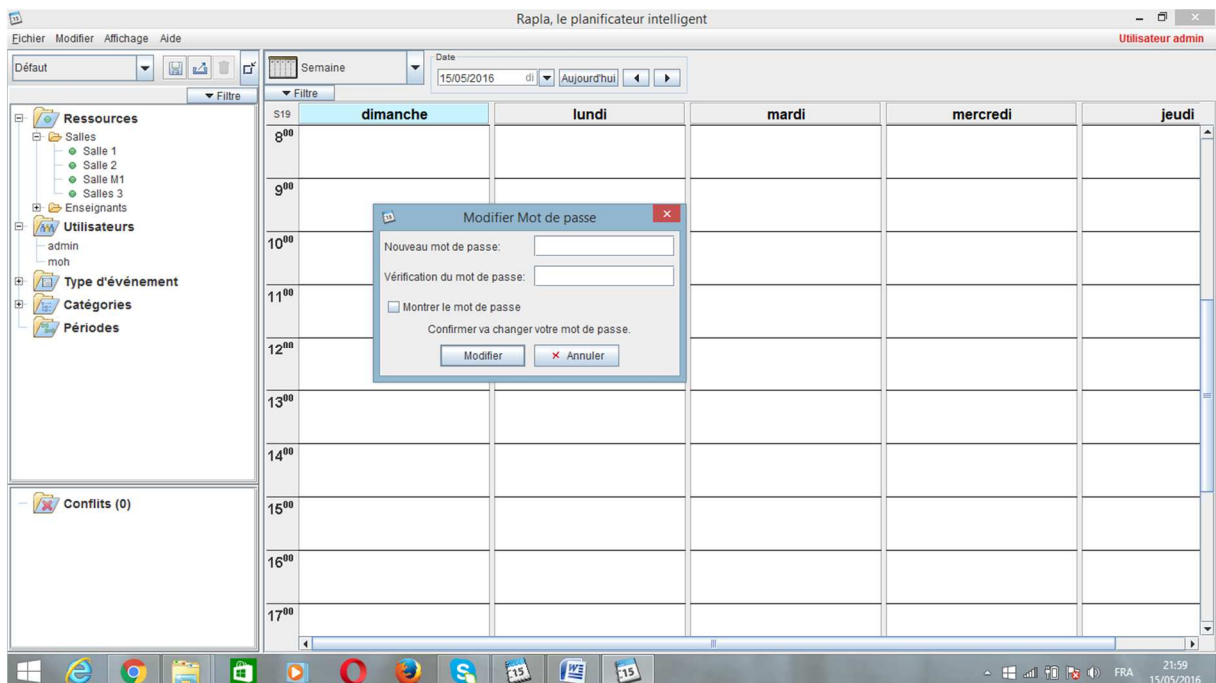


Figure 25: la saisie du mot de passe d'utilisateur.

### IV.3.5 Création des ressources

Cette opération est également faite par l'administrateur. En cliquant droit sur le nœud Ressources il peut créer des types des ressources (enseignants, salles), et sous types des ressources s'il veut organiser les ressources par catégories (par exemple pour les enseignants il crée les sous types : enseignants de département MI, enseignants de département physique).

## Chapitre IV : Implémentation de l'application

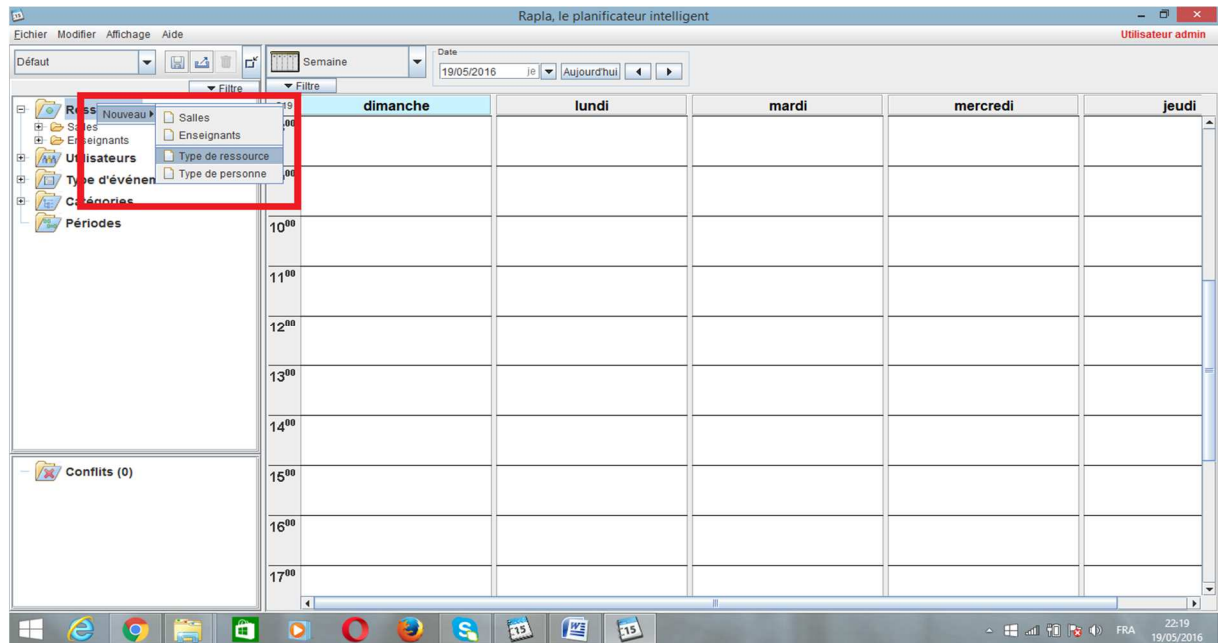


Figure 26: création des ressources.

### IV.3.6 Création des types d'événements

Si l'administrateur voulait organiser les événements par types, l'application RAPLA l'offre la possibilité de créer des types d'événements (dans notre cas chaque type représente un emploi du temps d'une promo) en cliquant droit sur le nœud types d'événements et choisissant nouveau type.

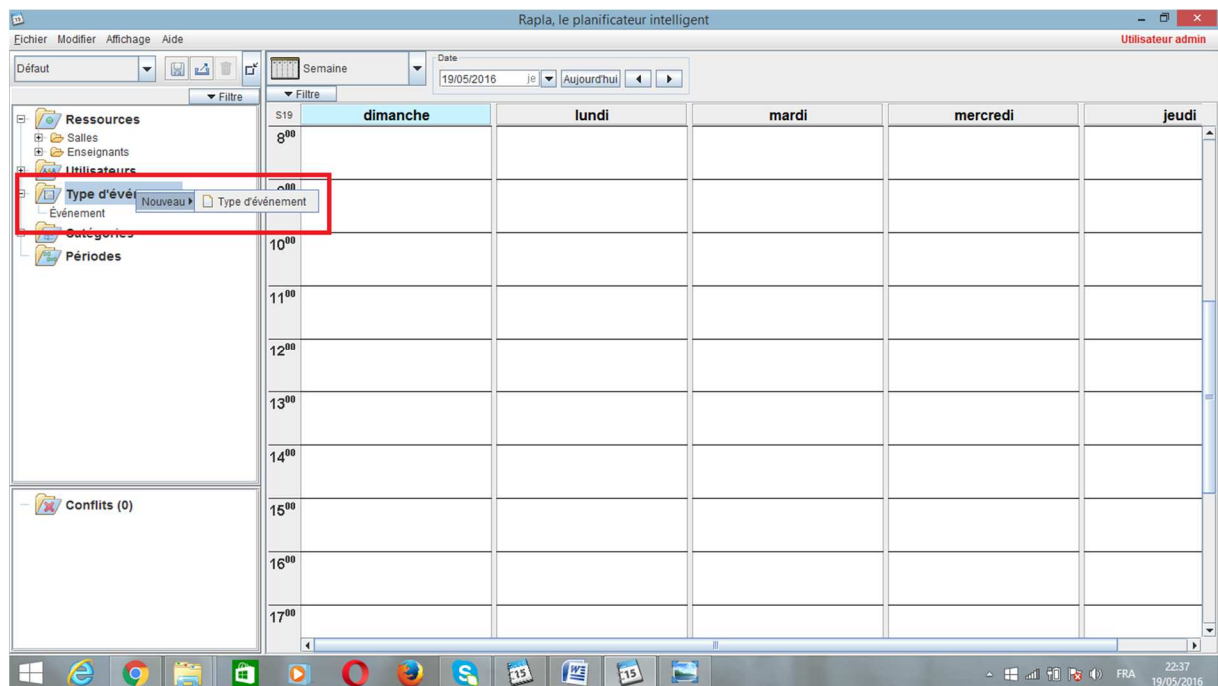


Figure 27: création des nouveaux types d'événements.

# Chapitre IV : Implémentation de l'application

## IV.3.7 Création des événements

Cette opération est faite par tous les utilisateurs, après un clic droit dans le calendrier et le choix du type d'événement une boîte de dialogue s'ouvre pour le choix des ressources et l'heure de l'événement.

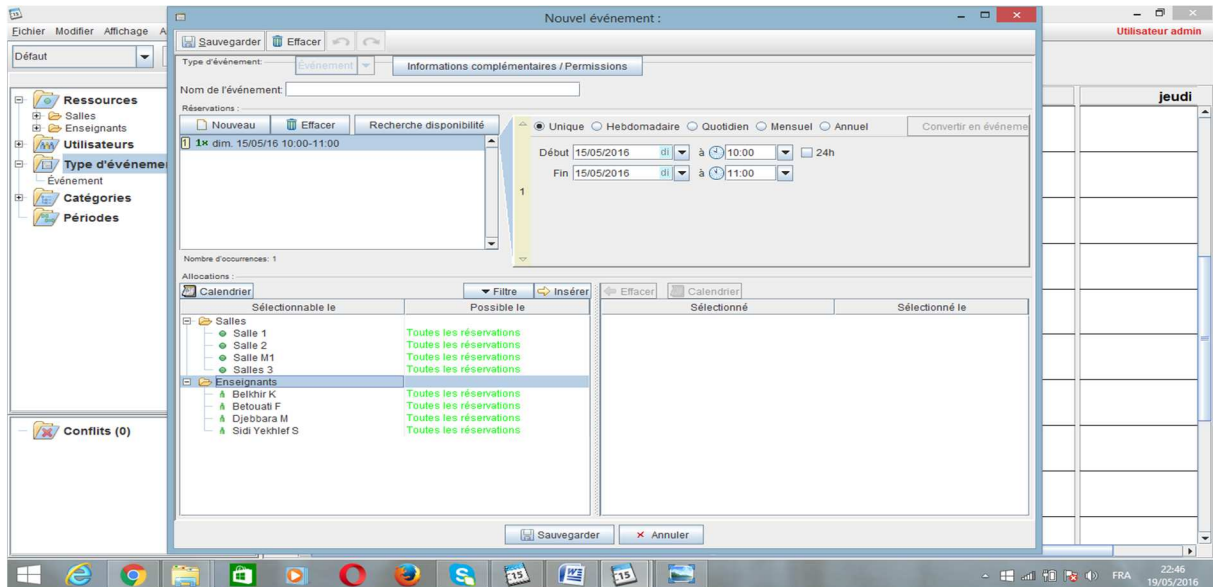


Figure 28: création des événements.

## IV.3.8 Conflits

Les conflits surviennent lorsque deux événements sont prévus pour la même date où en utilisant les mêmes ressources. Dans cet exemple, deux cours souhaitent en même temps la ressource de salle2, qui ne fonctionne pas naturellement.

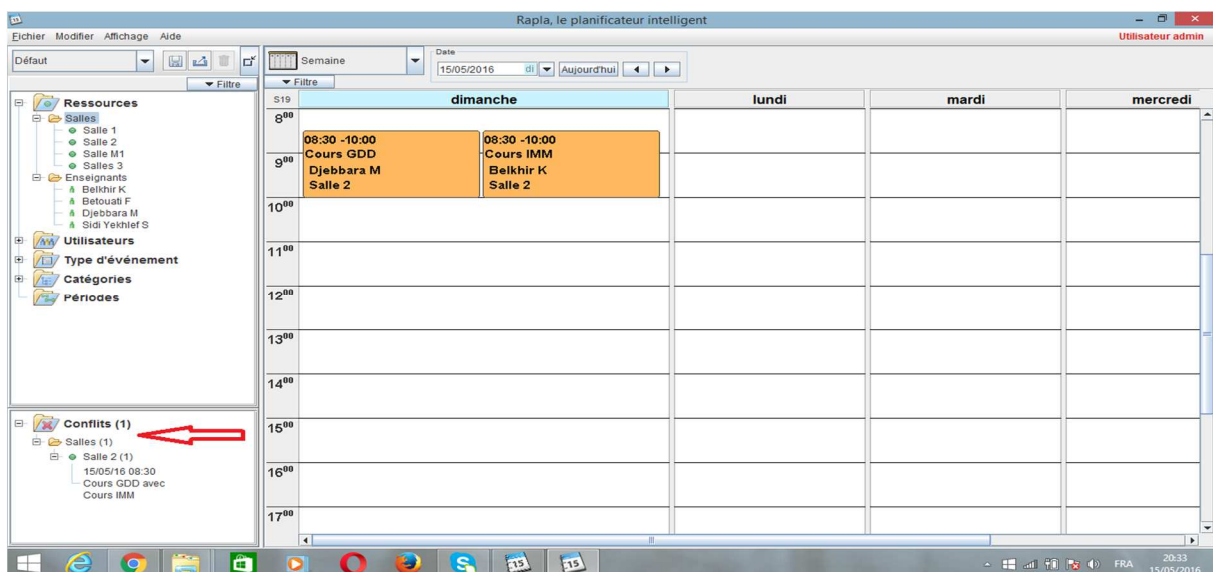


Figure 29: Le conflit dans RAPLA.

## Chapitre IV : Implémentation de l'application

Il faut noter que les deux conflits présentés ici se réfèrent à la même ressource / heure. C'est parce que l'utilisateur qui est connecté en possède les droits d'administrateur, et voit les conflits du point de vue de toutes les «parties au conflit». Un utilisateur normal ne verrait qu'une seule ligne.

On clique droit sur l'entrée des conflits et on choisit Modifier pour ouvrir une boîte de dialogue de travail approprié pour changer l'événement qui provoque le conflit. On peut également résoudre le conflit en faisant glisser l'événement dans la vue.

### IV.3.9 Filtrage

Un clic sur le bouton Filter au dessus de la liste des ressources montre le dialogue suivant:

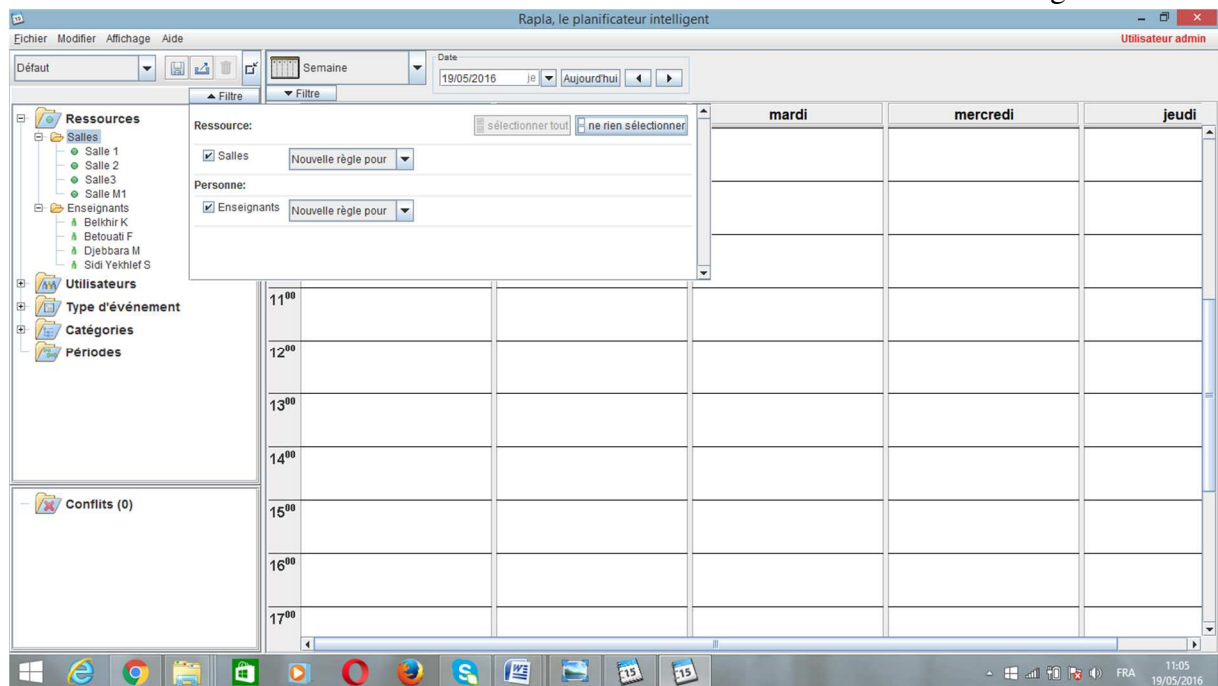


Figure 30: le filtrage des ressources dans RAPLA.

Par défaut, tous les types de ressources sont cochés, ce qui signifie: Chaque ressource est affichée et personne n'est filtré. On a deux possibilités pour la contrainte en outre la vue:

- Avec les cases, on peut marquer, quels types de ressources doivent passer à travers le filtre. Tous les types incontrôlés sont filtrés et donc toutes les ressources appartenant à ce type ne sont pas affichés dans la liste plus.
- Les ressources appartenant aux types de ressource vérifiés peuvent encore être contraints par des règles. Toutes les ressources d'un type de ressource, qui ne correspondent pas aux règles spécifiées droit à leur type de ressource sont confidentiels (Ils ne passent pas le filtre).

## **Conclusion du chapitre IV**

Dans ce chapitre, nous avons appliqué ce que nous avons dit dans les chapitres précédents, nous avons téléchargé l'application RAPLA et nous l'avons configuré selon les besoins des responsables de la faculté.

A la fin de l'implémentation de l'application, le résultat sera une application qui fait la génération des emplois du temps des différentes promotions et enseignants sans aucun conflit au niveau des séances. Cette application a permis de répondre aux besoins des administratifs par la résolution des problèmes de la gestion des emplois du temps ainsi que les conflits qui peuvent exister.