

**CLIN Matthieu  
CZEKALA Hugo  
KOBAY Dylan  
TRIBOU Romain  
VECCHIO Quentin**

**Université de Metz  
DUT INFO  
2<sup>ème</sup> Année**

**Groupe 3**

# **Rapport de projet de synthèse**

## **Analyse**

### **Logiciel de gestion d'une bourse aux livres**



## SOMMAIRE

I. Introduction	5
II. Analyse du sujet	6
1) Diagramme de cas d'utilisation	6
2) Diagramme de séquence	8
3) Diagrammes d'état-transition	9
a) Enregistrement d'un parent	10
b) Dépôt de livres	11
c) Vente de livres	12
d) Facture	13
e) Stock disponible	13
f) Visualisation facture	14
g) Gestion des états	14
h) Gestion des livres	15
i) Gestion des opérateurs	15
j) Gestion des filières	16
j) Gestion des filières	16
III. Analyse des données	17
IV . Scénario	18

V. Analyse du design	21
VI. Etude du développement	22
1) Application Web	22
2) Logiciel multi-OS	23
3) Déploiement	24
VII. Gestion Projet	25
VIII. Conclusion	26

# I. Introduction

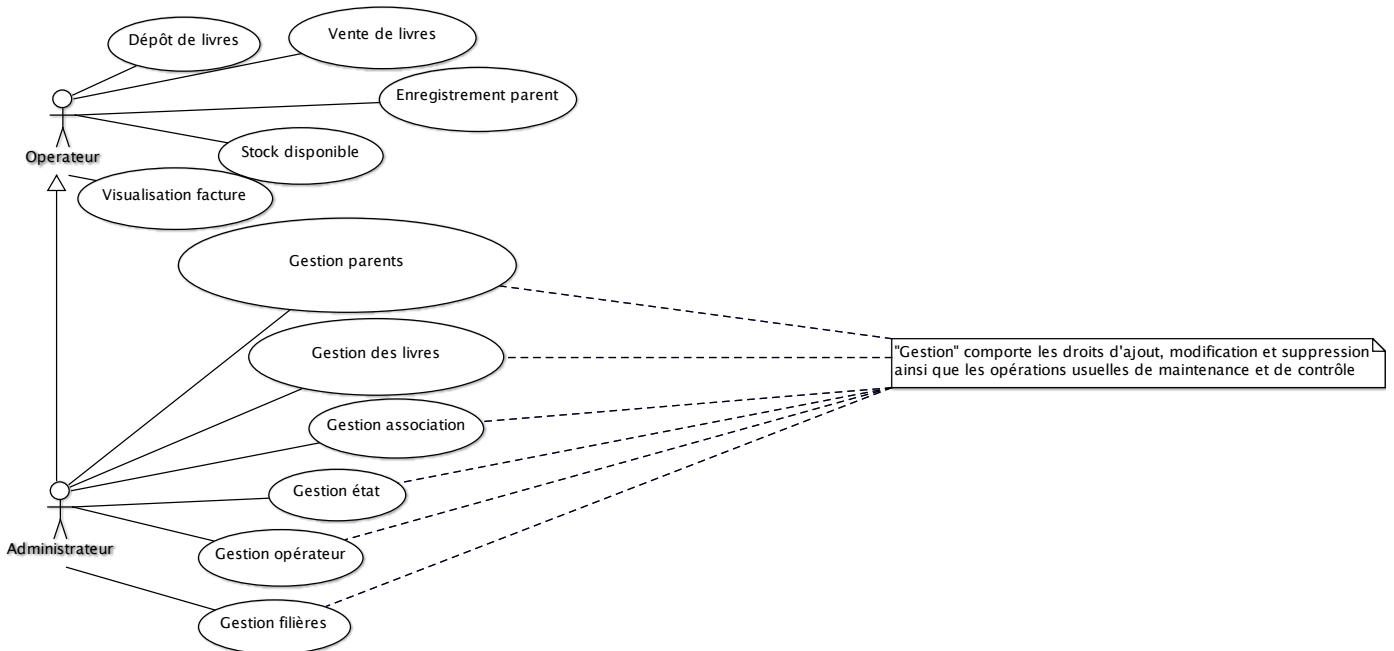
Dans le cadre de notre formation, nous devons réaliser un projet dit de synthèse par groupe de cinq personnes et avec la particularité d'y inclure un professionnel. Le projet que nous avons choisi est le site internet pour la gestion d'une bourse aux livres. En effet, la gestion manuelle faite par les associations durant ces bourses aux livres est difficile car des erreurs peuvent arriver, mais surtout aucun visuel rapide du stock n'est disponible. De plus, la gestion d'une telle bourse est souvent faite de manière non optimisée, par exemple on est obligé de gérer la bourse sur plusieurs jours afin de ne pas se perdre dans le stock de livre. C'est ici qu'entre en jeu notre application, car elle permettra de gérer de façon optimiser une bourse aux livres, grâce à la visualisation de stock en direct mais aussi grâce à un stockage des données via une base de données, ce qui évite la perte de documents et données importante.

Dans ce rapport nous tacherons de vous expliquer notre analyse du sujet. Pour cela, nous avons découpé notre analyse en différentes parties. Vous trouverez tout d'abord notre analyse complète concernant le développement. Ensuite notre analyse sur la base de données répondant aux besoins de notre solution. Nous expliquons également notre analyse et nos choix concernant le design de l'application et pour terminer, une partie consacrée à la gestion de ce projet.

## II. Analyse du sujet

Dans cette première, nous allons vous expliquer notre analyse du sujet grâce au langage de modélisation qu'est UML. Nous partirons d'un diagramme des cas d'utilisation de notre application pour expliquer chaque partie plus en détails à travers les diagrammes d'état transition. Nous utiliserons un diagramme de séquence pour expliquer plus en détails le déroulement d'une bourse aux livres.

### 1) Diagramme de cas d'utilisation



Ce diagramme de cas d'utilisation nous explique, comme son nom l'indique, tous les différents cas ou actions que chaque acteur pourra effectuer. Il est présent dans ce projet deux acteurs distincts qui sont l'opérateur et l'administrateur.

En premier lieu nous allons établir la liste des fonctionnalités accessibles à l'opérateur.

Ce dernier se charge d'enregistrer les dépôts ou ventes de livres, il saisit les données récoltées concernant ces actions dans la base de données; à la suite du dépôt ou de la vente, il enregistre la personne concernée si cela n'est pas déjà fait. L'opérateur peut également visualiser les stocks disponibles afin d'effectuer correctement les ventes et éditer et visualiser la facture concernant l'achat.

A présent nous allons lister les fonctionnalités de l'administrateur.

Cet acteur va donc s'assurer de correctes mises à jour des différents postes, c'est-à-dire de la gestion des livres à celle des parents.

Cette gestion comprend les possibilités d'ajout, de modification, de suppression ainsi que des fonctionnalités de maintenance et de contrôle, cette notion sera présente dans chacun des cas ci-dessous.

L'administrateur va gérer les livres, les filières (comprenant celle des classes et des matières), les états de livres (bon, moyen et médiocre), les opérateurs (login, mot de passe, ...), l'association du lycée, ainsi qu'une gestion des parents.

Toute cette gestion comme expliqué ci-dessus comporte plusieurs droits d'applications (ajout, modification, suppression,...). On peut également ajouter que l'administrateur hérite des fonctionnalités de l'opérateur, cela signifie qu'il pourra visualiser les différentes factures, le stock disponible et autres opérations propres à l'opérateur; ceci est représenté donc par une flèche d'héritage entre l'administrateur et l'opérateur.

## 2) Diagramme de séquence

Le diagramme de séquence nous permet d'observer comment la bourse aux livres doit se passer théoriquement. Une bourse est découpé en quatre parties, tout d'abord le début, la configuration de l'application et de la base de données, ensuite on commence une session de dépôt, arrive après la session de vente et enfin une session X où les parents ayant déposer des livres viennent chercher les invendus ou l'argent récolté grâce à la vente de leurs livres.

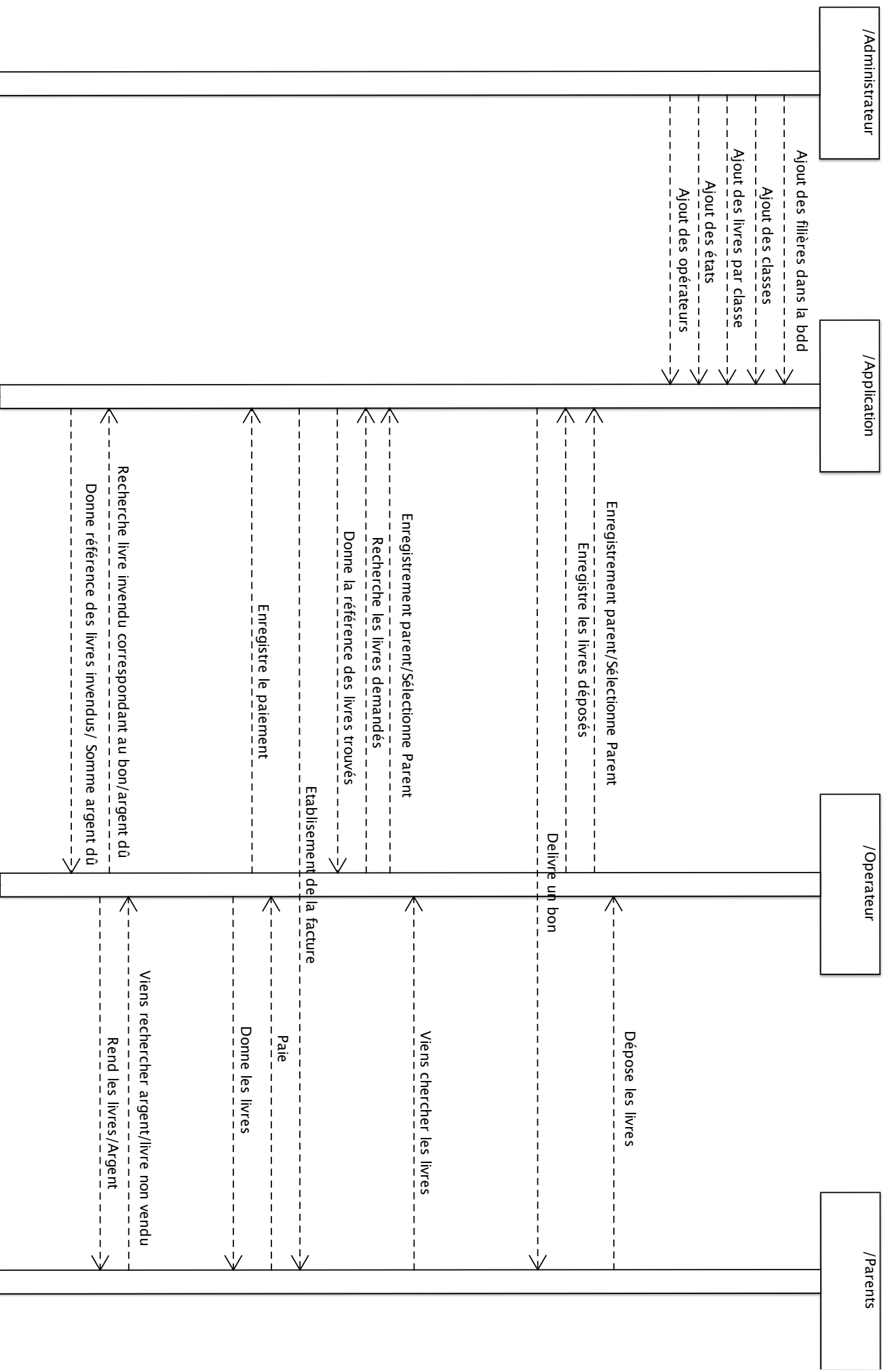
Au début, un administrateur configura l'application et la base de données. Il devra ajouter des filières (ex : SSI, SSVT, STG), ensuite des classes (ex : seconde, première et terminale) et enfin ajouter tous les livres (ex : Math 1èreS, ..). Il devra également créer des états pour les livres et fixer des pourcentages de réduction (ex : Médiocre : 45%). Il pourra aussi ajouter différentes associations auxquels les parents pourront adhérer. Pour finir la configuration de la bourse aux livres, l'administrateur ajoutera des opérateurs qui traiteront les dépôts et ventes.

Ensuite débute la session de dépôt, le traitement reste toujours le même pour l'opérateur. Le parent arrive, il dépose ses livres. L'opérateur lui demande son nom, si le parent est déjà enregistré alors on passe à l'étape suivante, sinon on l'enregistre. L'opérateur va ensuite alimenter la base de données avec les livres déposé par le parent. Enfin on génère un bon au parents, celui-ci permettra de venir à la fin de la bourse afin de récupérer l'argent de la vente ou bien récupérer les livres invendus.

La session de vente se déroule comme une session de dépôt, le parent arrive donne son nom, si il n'est pas enregistré alors on l'enregistre. Le parent va ensuite sélectionner les livres dont il a besoin. Une fois tous les livres sélectionnés, on génère une facture pour le parent. Le parent paye et l'opérateur lui donne ses livres.

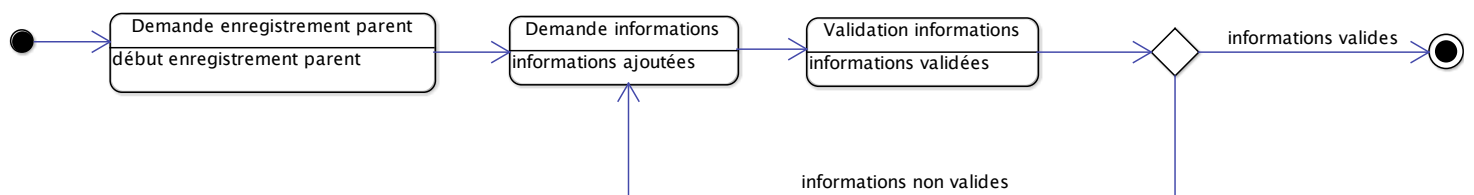
Enfin vient la session X, les parents qui ont déposé des livres lors de la session de dépôt viennent récupérer l'argent de la vente. Mais il se peut que certains livre reste invendu, dans ce cas l'opérateur remettra au parents tous les livres qu'il a déposé et qui ont été invendu.





### 3) Diagrammes d'état-transition

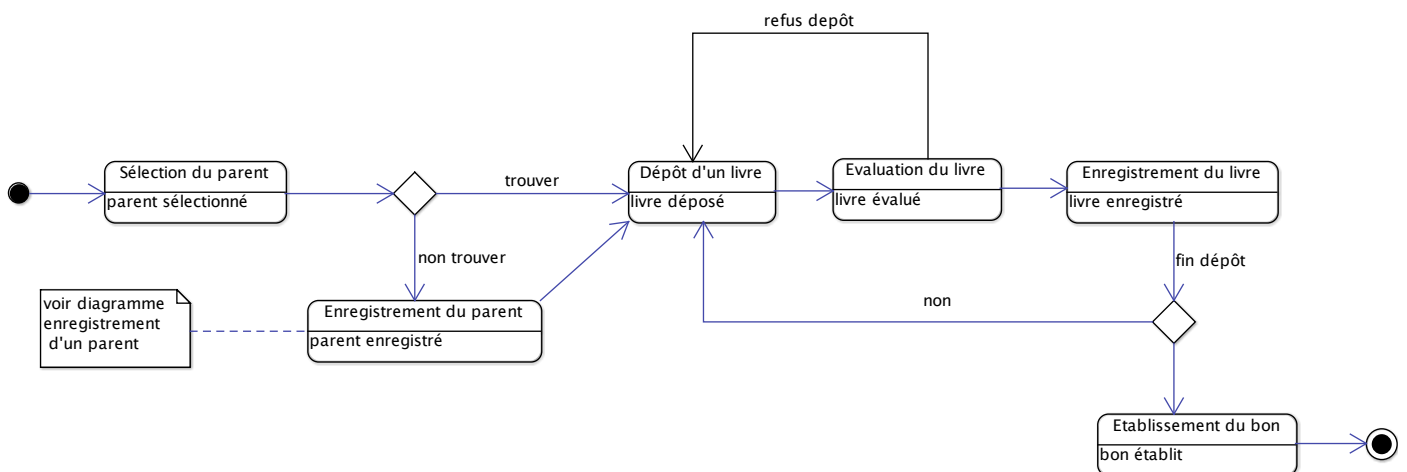
#### a) Enregistrement d'un parent



Lorsqu'un parent se présente à une bourse aux livres pour la première fois, il ne figure pas dans la base de données des années précédentes. Il est donc nécessaire que le parent s'enregistre pour pouvoir vendre et ou acheter des livres.

Des informations vont donc lui être demandées puis validées si et seulement si ses informations sont correctes sinon un retour par la case demande d'information sera nécessaire

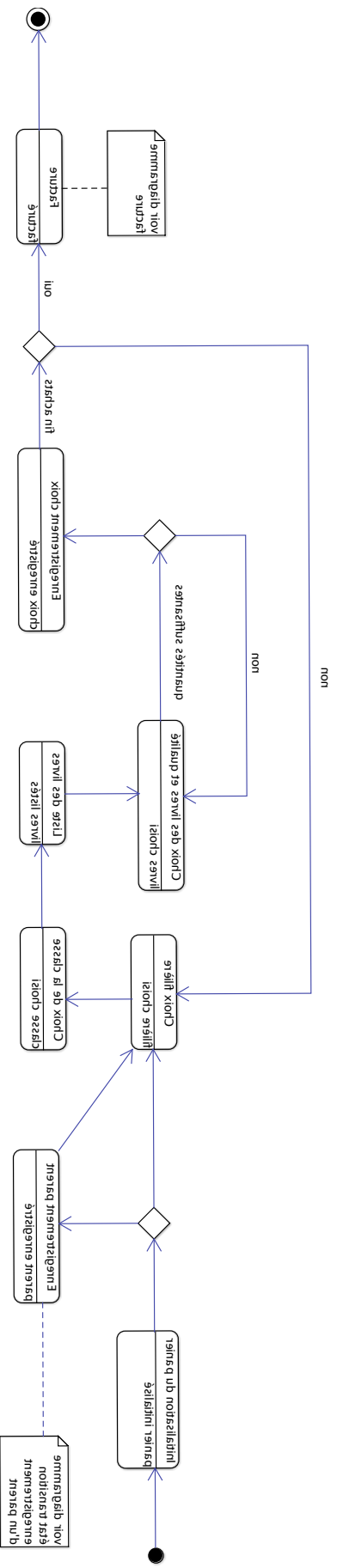
## b) Dépôt de livres



Pour pouvoir déposer un livre, un parent doit être présent dans la base de données. Dans le cas où le parent n'est pas trouvé il devra avant toutes choses s'enregistrer. Une fois le livre déposé, il va être évalué par l'opérateur, selon son état le dépôt du livre peut être accepté ou au contraire être refusé.

Si le dépôt du livre est accepté le livre est donc enregistré. Ensuite le parent peut mettre fin au dépôt ou au contraire il peut en déposer d'autres, si par exemple il a plusieurs enfants dans des classes différentes. Lorsque le parent met fin au dépôt un bon lui est remis afin qu'il puisse venir chercher l'argent récolté suite à la ventes de ses livres.

### c) Vente de livres

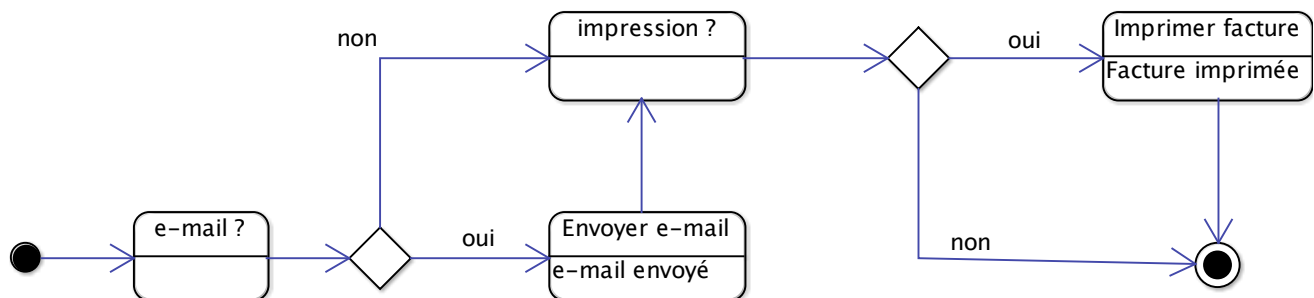


Pour pouvoir acheter un livre, un parent doit être présent dans la base de données. Dans le cas où le parent n'est pas trouvé il devra avant toutes choses s'enregistrer. Une fois le parent trouvé ou enregistré, la recherche d'un livre peut commencer.

Le parent indique la filière ainsi que la classe du livre recherché, ensuite les livres correspondant à sa recherche lui sont listés. Après avoir sélectionné le/les livres désiré(s), ainsi que la qualité et la quantité de chacun, son choix est enregistré si et seulement si la quantité de livre disponible en stock est suffisante.

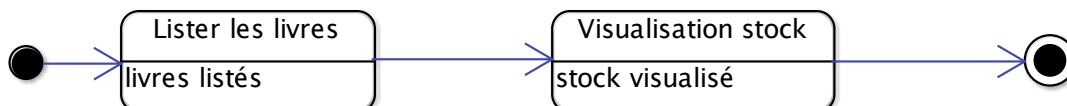
Après cela, il peut mettre fin à ses achats ou bien continuer en choisissant de nouveau livre. S'il choisit de mettre fin à ses achats il pourra visualiser sa facture.

## d) Facture



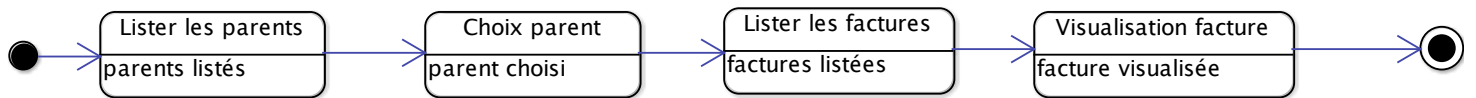
Suite à la visualisation de sa facture, le parent aura la possibilité de garder une trace de son achat. Soit par un envoi de la facture par mail ou encore par l'impression de celle-ci, il peut aussi faire abstraction de ces options.

## e) Stock disponible



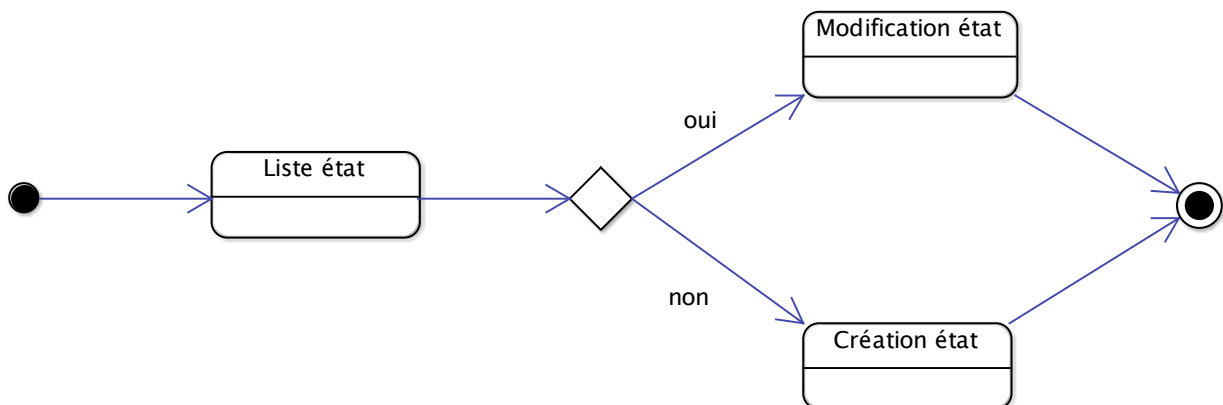
Si l'utilisateur veut visualiser le stock disponible, les livres en stock lui sont listés.

## f) Visualisation facture



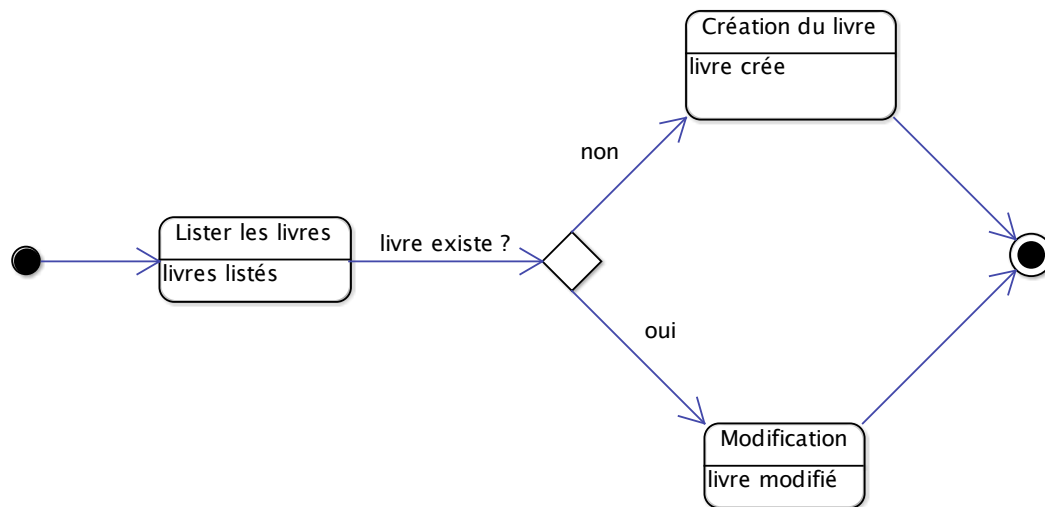
Pour la visualisation de la facture les parents sont listés, une fois le parent choisi les factures le concernant sont listées ce qui permet de visualiser les différentes factures associées à un parent.

## g) Gestion des états



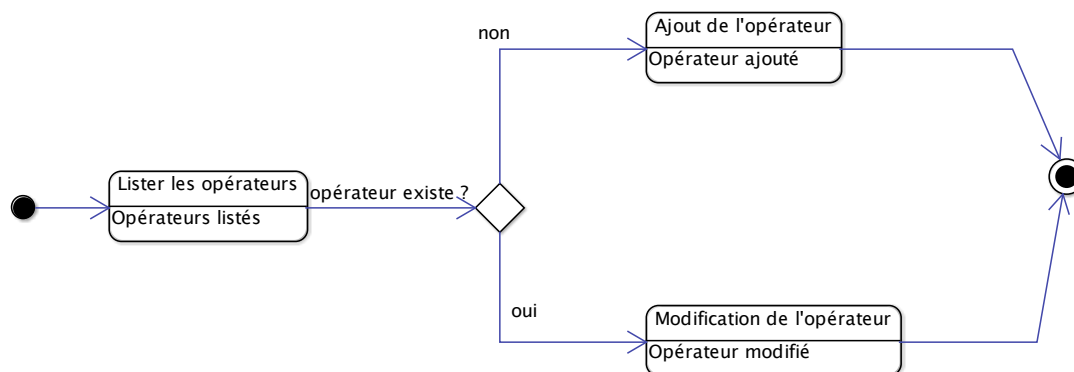
L'administrateur a la possibilité de gérer les états des livres. Il va les lister et si l'état qu'il recherche n'existe pas, il a alors la possibilité de l'ajouter. Sinon il peut le modifier.

## h) Gestion des livres



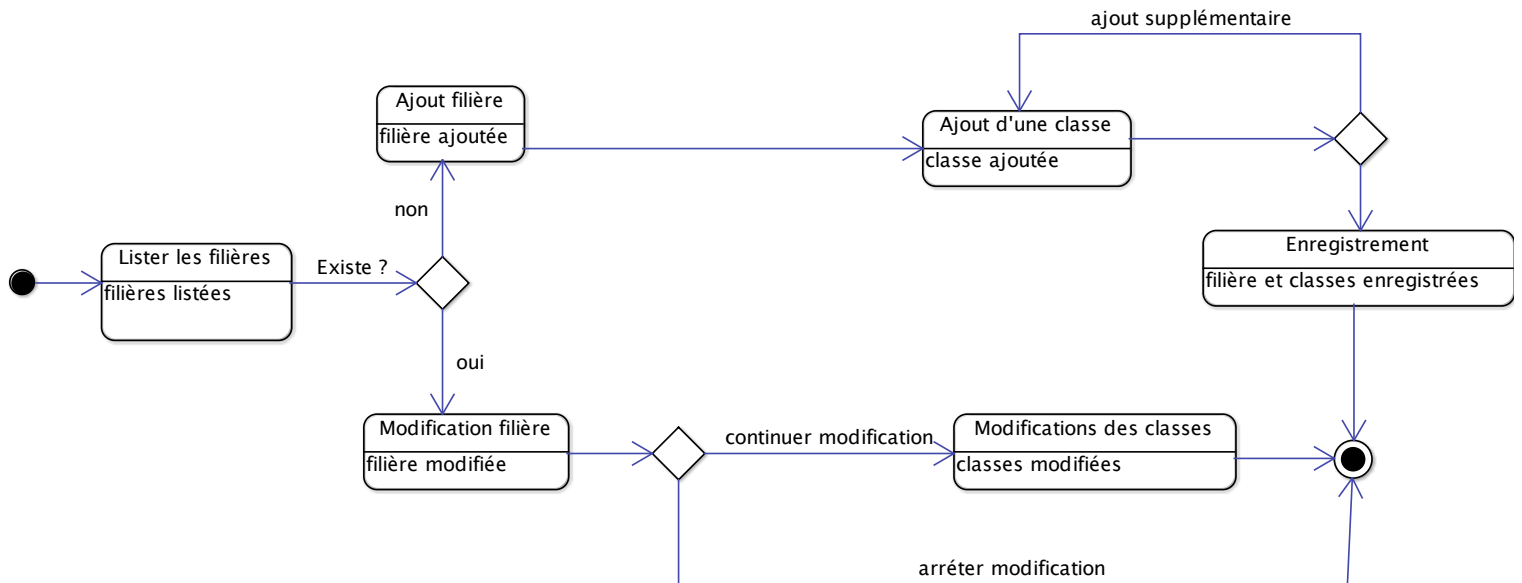
L'administrateur a la possibilité de gérer les livres. Il va les lister et si le livre qu'il recherche n'existe pas, il a alors la possibilité de l'ajouter. Sinon il peut le modifier.

## i) Gestion des opérateurs



L'administrateur a la possibilité de gérer les opérateurs. Il va les lister et si l'opérateur qu'il recherche n'existe pas, il a alors la possibilité de l'ajouter. Sinon il peut le modifier.

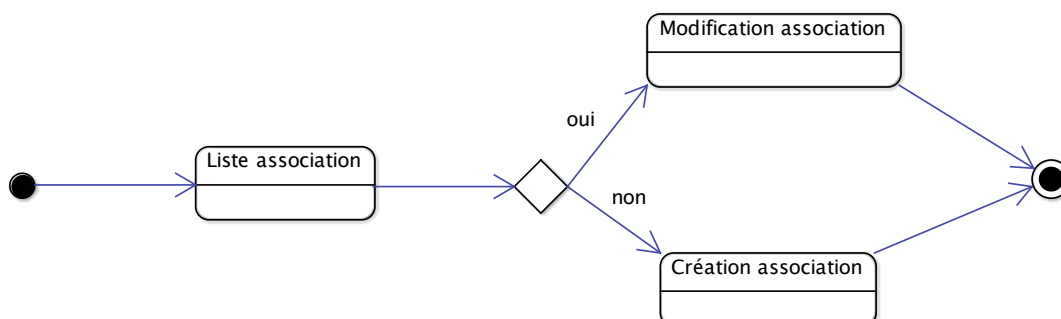
## j) Gestion des filières



L'administrateur a la possibilité de gérer les filières. Après avoir listées celle-ci, si la filière recherché n'existe pas, il a la possibilité de l'ajouter.

Ensuite il peut ajouter à cette filière une ou plusieurs classes, une fois la création terminée, la filière et ces classes sont enregistrées. Si la filière recherchée existe, il a la possibilité de la modifier, il peut choisir de modifier uniquement la filière ou continuer les modifications avec les classes associées à celle-ci.

## j) Gestion des filières



L'administrateur a la possibilité de gérer les associations. Il va les lister et si associations qu'il recherche n'existe pas, il a alors la possibilité de l'ajouter. Sinon il peut le modifier.



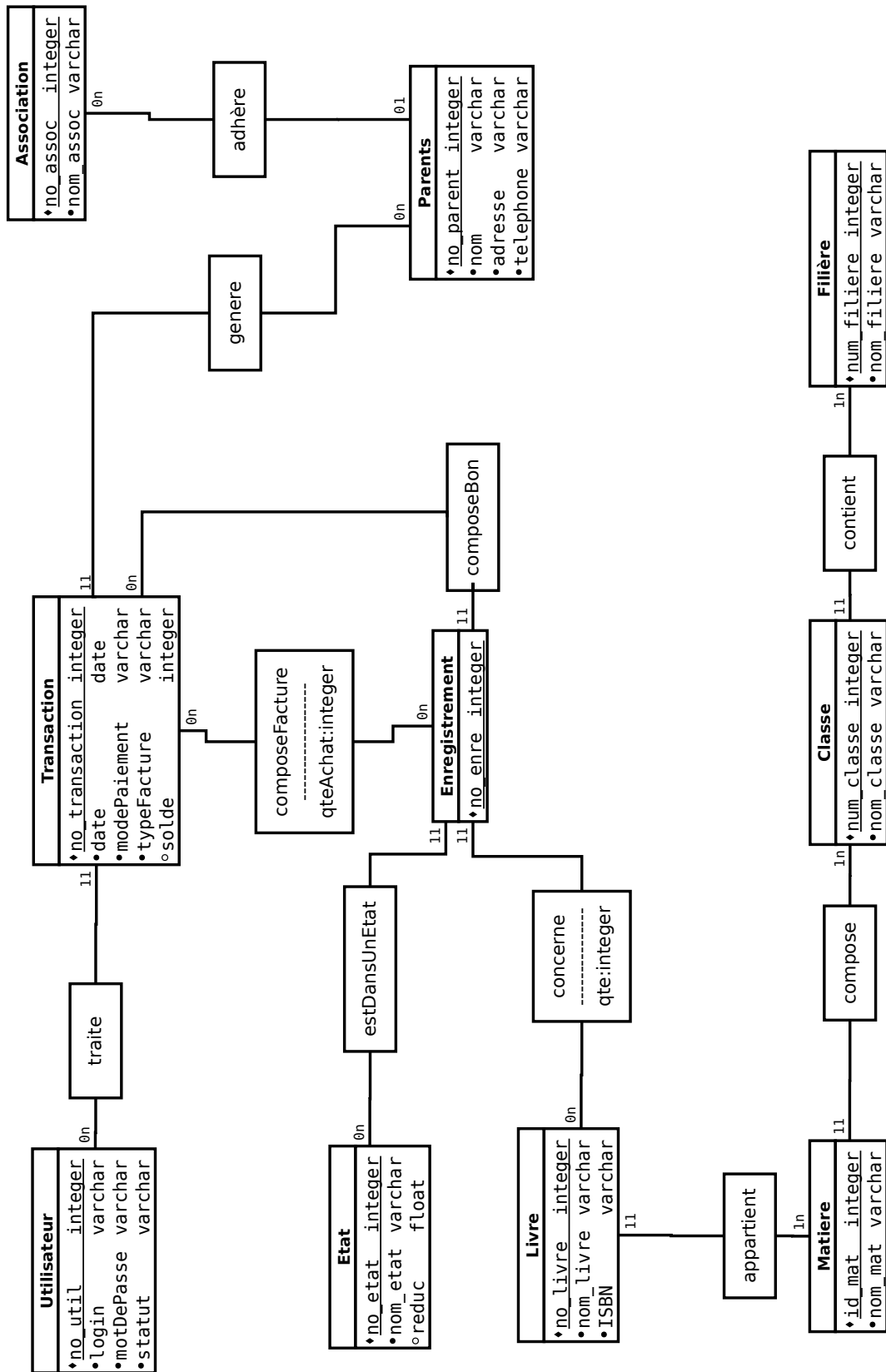
## III. Analyse des données

Pour la mise au point de l'application de gestion d'une bourse aux livres, nous avons dû analyser le déroulement de cette bourse à travers différents diagrammes afin d'automatiser un maximum de processus. A partir de ces diagrammes, nous avons pu en retirer les informations nécessaires au bon déroulement du procédé. Nous avons donc dégagé dans un premier temps un schéma entité/association.

Dans ce schéma E/A, nous retrouvons l'entité "Utilisateur". Cette entité contient toutes les personnes, administrateur et opérateur, qui interagissent avec l'application. Sur la partie de droite se trouve l'entité "Parent". Cette entité représente le parent qui vient acheter et/ou vendre des livres. Chaque parent, peut décider ou non d'adhérer à une association. Toutes la partie basse du schéma concerne les classes. Nous avons découpé chaque filière en classe puis en matière. A partir de là, chaque matière a des livres. La décomposition s'imposait d'elle même du fait que les livres différents selon les filières puis en fonction de la classe. D'autant plus que certaines filières ne sont que sur deux ans, telle que la filière S ou encore L, et d'autres sur trois ans, comme les BAC PRO.

Là où nous avons eu des difficultés, c'est pour la représentation des livres déposés et des livres vendus. Cette partie, bien que claire dans nos esprits au moment de l'analyse en aval, c'est trouvé être plus complexe que prévu. En effet, un parent peut déposer un ou plusieurs livres étant ou non dans le même état. De même un second parent peut venir acheter un ou plusieurs livres qui peuvent être dans des états différents et venir de parents différent. Nous avons éprouvé une difficulté afin de maintenir une traçabilité du livre. C'est à dire savoir à tout moment qui a déposé quel livre, quels livres sont vendus et surtout à qui appartiennent les livres non vendus.

Nous avons surmonté ce contre-temps en utilisant l'entité "Enregistrement", "Transaction" et l'association "ComposeFacture". L'entité "Transaction" représente l'action global, c'est à dire l'action de déposer ou d'acheter des livres. L'entité "Enregistrement" décrit une ligne de cette transaction lors du dépôt. Elle permet de savoir que le livre L a été déposé en quantité Q et l'état E. Un enregistrement concerne une et une seule transaction de type "dépôt". L'association "ComposeFacture", permet, quant à elle, d'enregistrer les livres vendus. Grâce a cette représentation, nous sommes capable de répondre à toutes les contraintes d'intégrité énoncés plus haut.



## IV . Scénario

Chaque année, un lycée organise une bourse aux livres, où des parents se présentent afin de déposer, dans l'objectif d'une vente, les livres de son ou ses enfants; ou bien d'en acheter.

Une vente est caractérisée par le dépôt d'un ou plusieurs livres auprès d'un opérateur chargé d'enregistrer les livres à la date du dépôt, il édite alors un bon de dépôt/reçu afin que le parent ait une trace de transaction pour venir tranquillement encaisser la somme perçue par la vente.

Lorsqu'un parent se présente pour la première fois, il est enregistré dans la base de données, puis on lui propose d'intégrer l'association du lycée afin d'obtenir des réductions ou divers autres offres.

Un achat de livres peut concerner plusieurs livres de plusieurs matières, dans des classes et filières différentes (livres de comptabilité 1ère STG, Spé SVT Terminale S,...); l'état du livre entre en compte dans le montant total de l'achat (ou de la vente), on distingue 3 états : bon, moyen et médiocre.

La parent effectue l'achat auprès d'un opérateur qui enregistre l'opération et délivre une facture au parent.

Un achat de livres peut concerner des provenances différentes, en effet on peut acheter des livres provenant de parents différents (3 livres de SVT, Maths et Philosophie de Terminale S; SVT et Maths provenant d'un parent et Philosophie d'un autre), bien entendu les livres achetés peuvent être dans différents états (2 livres en bon état et 1 en état médiocre).

Les livres achetés d'occasion profitent d'une réduction.

Livres			
id	name	ISBN	id_matiere
1	Mathématiques	...	1
2	Français	...	2
3	Allemand	...	3

Enregistrement_Ventes (2 personnes)						
id	id_livre	id_etat	id_transaction	qte	qte_prise	
1	1	1	1	2	2	1ere personne
2	2	1	1	1	1	
3	3	2	1	1	1	
4	2	1	2	1	1	2eme personne
5	3	1	2	1	1	

Transaction (3 personnes)			
id	id_enreg	qte	
3	1	1	3eme personne prenant 2 livres d'état différent
3	2	1	
4	3	1	4eme personne prenant 2 livres de même matière mais d'états différents
4	5	1	
5	1	1	5 eme personne

- Un parent souhaite acheter plusieurs livres, ceux-ci provenant de parents différents.
- Un parent se présente pour acheter/déposer des livres de plusieurs enfants.
- Un parent souhaite acheter un livre dans un certain état mais il n'est pas disponible pour le moment.
- Un parent souhaite acheter deux livres d'un état différents.

# V. Analyse du design

Le design d'une application est très important. En effet nous avons choisi d'être attentif au design lors de notre analyse, le design joue un rôle important car c'est la première chose que l'utilisateur verra, si le design lui plaît alors il sera ravi d'utiliser l'application. De plus, nous avons mis une priorité sur le fait d'allier design et convivialité. L'interface doit être simple d'utilisation et permettre à l'utilisateur une adaptation facile. Nous avons de plus choisi le framework bootstrap pour faire le design de l'application web, celui-ci propose des interfaces simples et épurées.

Nous avons misé sur des couleurs simples comme le blanc, le noir, le gris, ... Nous avons mis en avant aussi certains éléments en valeur avec des ombres. Chaque couleur a un rôle par exemple le vert signifie une action de validation, le rouge une action de suppression.

Nous voulions également mettre en place une barre de visualisation des étapes, en voici un croquis :

Cette barre rappelle celle présente dans les sites de ventes pour voir les différentes étapes comme enregistrement , paiement , validation.

# VI. Etude du développement

Nous vous expliquons dans cette partie nos choix concernant les méthodes de développement, vous trouverez plus de détails dans le rapport concernant exclusivement le développement.

## 1) Application Web

Le type de logiciel résultant de notre analyse est un logiciel Client-Serveur. En effet, le client doit avoir une interface simple et épuré, comme nous l'avons vu dans la partie analyse du design. Le serveur, quant à lui, doit juste contenir une base de données et gérer les accès à celle-ci.

Il existe plusieurs langages de programmation pour développer une telle application, mais nous avons choisi de créer une application exclusivement avec des langages de programmation orienté Web, car nous avons déjà de l'expérience dans ce domaine, de plus l'exportation d'une application web est réellement simpliste.

Voici une liste des langages utilisés :

- HTML5.
- CSS3.
- JavaScript.

Nous avons également fait le choix d'utiliser plusieurs frameworks puissants, qui nous simplifierions grandement le développement.

Voici une liste des frameworks que nous utiliserons :

- Bootstrap : Framework de twitter, il permet une gestion simple du design.
- AngularJS : Framework de Google, il permet de rendre dynamique une page web de manière simple et rapide, l'utilisateur aura l'impression d'être sur une vraie application et pas sur un site internet.
- CakePHP : CakePHP est défini par une convention stricte. Cette convention est le fruit d'année d'expérience des développeurs fondateurs de CakePHP et permet d'uniformiser les méthodes de développement. Cette convention, si elle est respectée, permet d'automatiser un grand nombre

d'opération telle que la connexion ou encore les jointures entre les tables. Pour cela, nous avons du traduire les termes du schéma relationnel en anglais et modifier la convention de nommage. Ainsi, toutes les clefs primaires des tables ont pour nous "id", les clefs étrangères suivent la forme tableOrigine\_id, les noms deviennent "name" etc. Ces conventions peuvent ne pas être respectées, mais cela expose à un coût de développement plus lourd.

- JQuery : Framework Javascript, il permet de faire des animations.

Concernant la base de données, nous avons fait le choix d'utiliser le langage MySQL car il s'adaptera à tous les SGBD.

## 2) Logiciel multi-OS

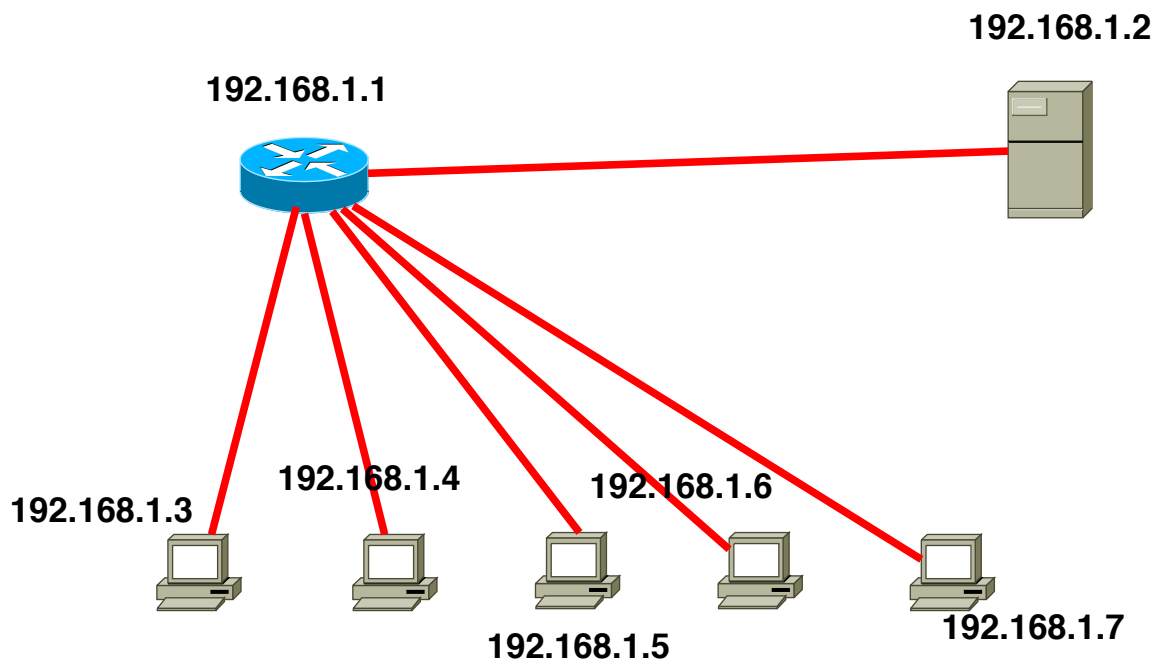
Une application web représente beaucoup d'avantages, mais aussi beaucoup d'inconvénient. En effet, le design d'une application web peut varier suivant les navigateurs, mais encore pire, si le navigateur est ancien (IE 6, IE7, IE8), il y aura de gros soucis de compatibilités, par exemple des éléments peuvent disparaître.

De plus, une application web peut être sujet à de nombreuses failles de sécurité. En effet, un site internet est à la portée de tout le monde, même si il est développé en local.

Notre analyse nous a permis de trouver une solution afin de résoudre plusieurs de ces problèmes. Le développement d'un logiciel multi-OS, qui contiendra notre application web, semble être la meilleure solution. En effet, le design sera adapté au logiciel, donc il n'y aura plus aucun soucis de compatibilité de style. De plus on pourra facilement gérer l'accès au site web en autorisant seulement les possesseurs du logiciels à accéder à l'application.

### 3) Déploiement

Voici un exemple de déploiement de notre application web sur un réseau local :



Cette exemple est bien entendu une solution possible, il en existe d'autres, cependant celle-ci reste la plus facile.

Le principe est simple, si le réseau et le serveur sont bien configurés, alors chaque opérateurs peut accéder au serveur web, et donc à l'application ,en passant par l'adresse 192.168.1.2.

Vous trouverez dans le rapport de développement plus d'informations concernant les plusieurs possibilités de déploiement.



## VII. Gestion Projet

Afin d'avancer dans notre projet de façon optimale, nous avons mis en place un planning de partage des tâches.

Ces tâches sont divisés en plusieurs parties distinctes:

- L'analyse.
- Le développement du logiciel.
- Le développement de l'application Web.
- Les simulations et tests.
- La rédaction de rapports.

Voici un tableau de répartition des tâches qui récapitule la gestion de notre projet.

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
Analyse														
Logiciel														
Application Web														
Simulations et tests														
Rapports														

Nous avons fait en sorte que toutes ces tâches soient équitablement répartie, afin que tout le monde puissent apprendre de nouvelle choses.

## VIII. Conclusion

Au cours de ce rapport, nous avons essayé de faire preuve de la plus grande clarté et précision possibles pour expliquer et justifier nos choix. Nous nous sommes appuyés sur les différents diagrammes de cas d'utilisation, d'état-transitions et de séquences afin de transmettre notre analyse du développement. Le diagramme de cas d'utilisation nous a permis de bien définir les actions de notre application finale. Ceux d'état-transitions et de séquences nous ont permis de mieux nous représenter le cheminement de l'application, du point de vue d'une action particulière puis relativement au temps. Le diagramme d'entités-association nous a permis de mettre en place une base de données cohérente et optimisée qui nous évite la redondance. L'analyse du design nous a également permis de nous faire une idée rapide de la futur application.

L'analyse du sujet nous a également permis de choisir les langages de programmation que nous utiliserons dans le développement de l'application, un rapport sera entièrement consacré à ces choix.

La modélisation de ces différents diagrammes nous aura permis d'avoir une analyse plus structurée et surtout de pouvoir revenir sur nos différents choix pour les modifier et ainsi nous rapprocher de la solution finale choisie. La réalisation de ce projet à l'aide d'UML et des diagrammes de modélisation des systèmes d'informations nous aura donc été très bénéfique. Nous avons pu nous rapprocher de méthodes d'analyse et de développement proches de ceux de l'entreprise et ainsi nous éloigner de cette mauvaise tendance que nous avons tous en commençant à programmer et ensuite réfléchir.