



RAPPORT DE PROJET

LICENCE 3 INFORMATIQUE

Gestion du Parc Informatique de la Faculté des Sciences

Auteurs :

ALAOUI YOUSFI Khaoula
CHAMROUK Laila
EL OUAZZANI Soukaina
LAAFOU Ayoub
MAJDOUL Kaoutar

Sous la direction de :

M^{me} Anne-Muriel ARIGON
M. Michel LECLERE

ANNÉE UNIVERSITAIRE 2019-2020

Nous adressons nos remerciements à nos encadrants, M^{me} Anne-Muriel ARIGON et M. Michel LECLERE, pour nous avoir guidé tout au long de ce projet, ainsi que pour leur patience, leur présence et leurs précieux conseils. Nous les remercions également pour la révision ainsi que la correction de ce projet.

Table des matières

1	Introduction	1
2	Cahier des charges	2
2.1	Présentation du sujet et analyse du contexte	2
2.1.1	Analyse du sujet et de son contexte	2
2.1.2	Analyse de l'existant	2
2.1.3	Analyse de l'environnement dans lequel le logiciel va être utilisé	2
2.2	Analyse des besoins fonctionnels	2
2.3	Analyse des besoins non-fonctionnels	3
3	Conception	4
3.1	Modélisation de données	4
3.1.1	Dictionnaire de données	4
3.1.2	Modèle Entité-Association	6
3.1.3	Modèle logique	8
3.2	Cas d'utilisation	9
3.2.1	Entrée inventaire	9
3.2.2	Modification inventaire	9
3.2.3	Édition caractéristiques	10
3.2.4	Changement d'état	10
3.2.5	Sortie inventaire FDS dans UM	10
3.2.6	Sortie inventaire UM	11
3.3	Interface graphique	11
3.4	Diagrammes d'état-transition	13
3.5	Choix de l'architecture	14
4	Réalisation	15
4.1	Choix technologiques	15
4.1.1	Choix du SGBD	15
4.1.2	Choix des langages	15
4.1.3	Choix de l'hébergeur	15

4.2	Implémentation de la base de données	15
4.2.1	La création des tables	16
4.2.2	Les triggers	16
4.2.3	Les inserts	16
4.2.4	Les requêtes	17
4.3	Implémentation du site web	18
4.3.1	Gestion du panier	18
	Sessions	18
	Cookies	18
	Implémentation	18
4.3.2	Insertion d'un ordinateur	21
4.3.3	Édition des fiches inventaires	21
4.4	Test/Validation	22
5	Résultats	24
5.1	Installation	24
5.2	Manuel d'utilisation	25
5.2.1	Actions indépendantes du panier	25
	Se connecter	25
	Ajouter un ordinateur	25
	Effectuer un ajout dans le panier	25
	Éditer une fiche inventaire	25
	Afficher les ordinateurs à renouveler	25
	Afficher les ordinateurs en panne ou en réparation	25
5.2.2	Actions dépendantes du panier	25
	Effectuer une modification inventaire	25
	Effectuer un changement d'état	25
	Effectuer une sortie FDS	26
	Effectuer une sortie UM	26
	Effectuer une édition des caractéristiques	26
6	Gestion de projet	27
6.1	Méthodes de travail	27
6.2	Diagramme de Gantt	27
6.3	Répartitions des tâches	27
6.4	Outils de productivité	28
6.4.1	Outils de communication	28

6.4.2 Outils de collaboration	28
7 Conclusion	29
7.1 Perspectives	30
8 Annexes	I
8.1 Dictionnaire de données	I
8.2 Maquette - Page recherche	IV
8.3 Maquette - Page sélection	V
8.4 Maquette - Page panier	V
8.5 Diagramme de navigation	VI
8.6 Triggers	VII
8.7 Requêtes	X
8.8 Code - Ajout dans le panier	XII
8.9 Test - Ajout ordinateur manuellement	XIII
8.10 Test - Ajout ordinateur CSV	XIV
8.11 Test - Sortie FDS	XV
8.12 Test - Recherche	XVI
8.13 Test - Répétition Panier	XVII
8.14 Visuels de l'application web	XVIII
8.15 Product Backlog	XXIII
8.16 Exemple de compte rendu	XXVII
8.17 Diagramme de GanttXXVIII
8.18 Trello	XXIX
8.19 Diagrammes d'état-transition	XXX
8.19.1 Ajout ordinateur	XXX
8.19.2 Modification inventaire	XXXI
8.19.3 Édition caractéristiques	XXXII
8.19.4 Changement d'état	XXXIII
8.19.5 Sortie inventaire FDS	XXXIV
8.19.6 Sortie inventaire UM	XXXV

Table des figures

1	Exemple de l'attribut InfoFicheInventaire	5
2	Modèle Entité-Association	6
3	Maquette - Squelette du site	12
4	Architecture MVC	14
5	Création de la table Ordinateur	16
6	Implémentation des clés étrangères pour la table Ordinateur	16
7	L'insertion d'un ordinateur	17
8	Gestion durée session	18
9	Gestion des cookies	18
10	Code - Pré-ajout d'ordinateur dans le panier	19
11	Code - Enregistrement des ordinateurs sélectionnés	19
12	Algorithme pseudo-code - Ajout dans le panier	20
13	Paramétrage des cookies sur Firefox	24

1 Introduction

Actuellement, la Faculté des Sciences (FdS) possède un grand nombre d'ordinateurs à travers le campus. Cependant, le suivi du cycle de vie de ce parc de plus de 700 machines n'est pas optimal. En effet la traçabilité des ordinateurs se fait dans différents fichiers dispersés dont le format n'est pas identique d'un document à un autre. De ce fait, lorsque l'on veut avoir des informations sur, entre autres, la localisation d'un ordinateur, son état ou la personne chargée de sa maintenance, la recherche devient chronophage et compliquée. Cela rend également ardue l'édition des différentes fiches inventaire qui sont des fiches à produire à chaque achat, changement de lieu ou de responsable et sortie d'un matériel nécessitant de nombreuses informations présentes dans différents fichiers et ajoute du travail au personnel en charge de leur rédaction.

La FdS souhaiterait disposer d'une application permettant à son service logistique de suivre plus aisément le cycle de vie des machines de la faculté des sciences. Il s'agit donc de créer un système d'information avec une interface web ergonomique lié à une base de données. Ce système devra permettre, dans les grandes lignes, l'ajout d'ordinateur, la recherche de machines et leur sélection ainsi que différentes actions de modification et d'édition d'inventaires et de consultation d'ordinateurs selon leur état ou statut.

Dans le cadre du module de travail d'étude et de recherche, se déroulant tout le long du second semestre de notre troisième année de licence, nous avons été chargé de la conception, du développement et des tests de l'application web visant à gérer le parc informatique de la FdS. Ce travail a été effectué par notre groupe composé de cinq personnes : ALAOUI YOUSFI Khaoula, CHAMROUK Laïla, EL OUAZZANI Soukaina, LAAFOU Ayoub et MAJDOUL Kaoutar, ceci sous la direction de M^{me} ARIGON et M. LECLERE pour une durée de 15 semaines, soit de la mi-janvier à avril 2020.

Tout au long de ce rapport nous présenterons, dans un premier temps, le sujet du projet en détail, l'analyse que nous en avons fait ainsi que le cahier des charges que nous avons défini. Suite à cela, nous analyserons la partie conception de ce projet. Il en découlera les réalisations effectuées au niveau de l'application web et de la base de données ainsi que leurs résultats, leurs tests ainsi que la procédure à suivre pour l'installation du site. Enfin nous détaillerons l'organisation adoptée tout au long du projet avant de présenter la conclusion finale de notre travail et les différentes améliorations possibles.

2 Cahier des charges

2.1 Présentation du sujet et analyse du contexte

2.1.1 Analyse du sujet et de son contexte

La gestion du parc informatique est fondamentale au sein de la FdS. Cependant, à l'heure actuelle, aucun logiciel ou application web permettant le recensement et la gestion des machines n'est utilisé. En effet, l'inventaire est géré par différentes personnes sur différents fichiers Excel qui sont ensuite partagés. En parallèle des fichiers Excel, des fiches inventaires en format papier sont remplies à chaque entrée, modification ou sortie d'inventaire. De ce fait, il devient difficile de suivre les changements effectués lors de la modification des caractéristiques de certaines machines comme, par exemple, leur état ou l'historique de leurs déplacements. Cela entraîne des pertes d'informations. L'éparpillement des fichiers et des données augmente ainsi le travail à effectuer pour retrouver les informations recherchées.

Dans le contexte actuel, la solution à proposer devra permettre l'automatisation du traitement des données et d'avoir toutes les informations sur les machines de la FdS au même endroit. Cette centralisation des fonctionnalités permettra un gain de temps et une optimisation de l'organisation de la gestion du parc informatique.

2.1.2 Analyse de l'existant

Dans le marché informatique actuel, il existe des outils d'inventaire et de gestion de parcs informatique sous Windows ou Linux tels que *GLPI*, *OCS Inventory NG*, *H-Inventory* ou encore *OpenNMS*. Cependant, ils ne sont pas tous gratuits et ne permettent pas l'organisation précise que demande la FdS au niveau des sorties ou modifications d'inventaire. Nous avons donc été chargés de concevoir et d'implémenter une base de données ainsi qu'une application web répondant aux attentes de la FdS.

2.1.3 Analyse de l'environnement dans lequel le logiciel va être utilisé

L'application web sera utilisée par les personnes s'occupant de l'inventaire du parc informatique, et éventuellement, les techniciens réalisant ou constatant un changement sur l'une des machines. Étant donné que nous développons une application web destinée à être utilisée sur l'intranet de la FdS, tout utilisateur aura besoin d'un ordinateur contenant un navigateur web et une connexion internet.

2.2 Analyse des besoins fonctionnels

Au vu du grand nombre d'ordinateurs répartis à différents endroits de la FdS, il est plus que nécessaire de trouver une solution durable pour la gestion et la traçabilité de toutes les machines. Afin de gérer l'ensemble des ordinateurs, la FdS a souhaité la mise en place d'une application web associée à une base de données.

Premièrement, l'application web doit permettre l'intégration des ordinateurs dans la base de données de façon manuelle par un formulaire ou via un fichier. Elle doit également permettre la visualisation et la sélection de ces ordinateurs selon différents filtres choisis ainsi que la modification de leurs emplacements et caractéristiques. De plus, l'édition des fiches d'entrée, de modification et de sortie d'inventaire à destination de la comptabilité de l'Université de Montpellier doit pouvoir être générée depuis l'interface proposée.

Nous devons permettre à l'utilisateur d'ajouter des ordinateurs dans la base de données de manière simple et rapide. Nous devons également lui permettre d'effectuer des recherches d'ordinateurs par numéro de série, localisation, état, statut, garantie et s'ils possèdent ou non un numéro d'inventaire et d'immobilisation. Suite à cela nous devons mettre en place un système de panier permettant à l'utilisateur de stocker, au fur et à mesure, les sélections faites après une recherche. Puis nous devons implémenter les actions dépendantes de celui-ci qu'il pourra effectuer telles que la modification inventaire, le changement d'état, une sortie de l'inventaire FdS ou UM ou encore une édition des caractéristiques.

En parallèle, nous devons mettre à disposition l'accès aux fiches inventaire pour une entrée, une modification ou encore une sortie d'inventaire pour une période choisie ainsi que la possibilité de télécharger les résultats. De plus, l'utilisateur doit pouvoir visualiser les ordinateurs dont la date de garantie se termine avant une date choisie afin de procéder à leur renouvellement et visualiser l'ensemble de ordinateurs en panne ou en réparation.

Deuxièmement, au niveau de la base de données, nous devons proposer un moyen de conserver les données sur les ordinateurs de manière pérenne et permettre d'y accéder facilement dans le but de faire des ajouts de machine, des modifications ainsi que des mises à jour aux niveau des différentes données. Elle devra également permettre l'historisation des mouvement des ordinateurs ou encore l'édition des fiches inventaire.

En somme, cette application de gestion du parc informatique de la FdS devra proposer une interface web permettant l'ajout d'ordinateur, la consultation des ordinateurs selon certains filtres, la sélection d'ordinateurs afin d'éditer leurs informations, et la génération des différentes fiches inventaire. Le tout sera relié à une base de données pouvant stocker toutes les informations, vérifiées en amont, nécessaires au bon fonctionnement de l'application.

2.3 Analyse des besoins non-fonctionnels

Les besoins non fonctionnels permettent de visualiser l'ensemble des objectifs liés aux performances de l'application web ainsi que les contraintes de son environnement. De ce fait, au niveau des objectifs non fonctionnels de cette applications, nous devons assurer une cohérence des informations stockées dans la base de données. Nous devons également proposer des interfaces simples et ergonomiques. De plus, nous devons garantir que le site soit accessible et performant et nous devons faire en sorte que le code fourni soit clair afin de permettre son renouvellement si besoin. Enfin, le site doit être codé en HTML/CSS PHP et la base de données en SQL.

3 Conception

3.1 Modélisation de données

3.1.1 Dictionnaire de données

Un dictionnaire de données est le résultat de la phase de recueil d'information aussi appelée collection des données. Il regroupe toutes les données que nous devons conserver dans notre base. C'est une étape nécessaire à la conception de la base de données relationnelle. Pour recenser les informations, nous avons étudié tous les documents fournis par nos encadrants telles que les différents modèle de fiches inventaires, les tableaux Excel de redistribution d'ordinateur ou encore le fichier concernant la procédure inventaire, physique et comptable mis à disposition.

De ce fait, le dictionnaire de données présent dans l'annexe 8.1, indique pour chaque donnée :

- Le nom de la rubrique : il s'agit d'un champ décrivant ce à quoi la donnée correspond (par exemple «Numéro de série d'un ordinateur»)
- Le type de donnée pouvant avoir comme valeur :
 - Alphabétique : donnée composée seulement de caractères alphabétiques (de 'A' à 'Z' et de 'a' à 'z')
 - Numérique : donnée composée seulement de nombres (entiers ou réels)
 - AN (Alphanumérique) : donnée composée à la fois de caractères alphabétiques et numériques
 - Date : donnée sous la forme d'une date (AAAA-MM-JJ)
 - Booléen : Vrai ou Faux
- La longueur : décrivant le nombre de caractères ou de chiffres au maximum.
- La nature de la donnée pouvant être :
 - Élémentaire : l'information ne peut pas être décomposée, elle se confond avec la valeur prise par la donnée. Par exemple, un nom, un titre d'un livre etc...
 - Calculées : données obtenues par le calcul à partir de données élémentaires.
 - Composées : données obtenues par la concaténation de données élémentaires.
- La règle d'intégrité : représentant les contraintes et règles de calcul, par exemple si une donnée est strictement supérieure à 0, si elle est le résultat d'un calcul, NOT NULL, UNIQUE ect...

Concernant les informations collectées lors de notre phase de recueil d'informations, nous avons :

- pour **un ordinateur** : le numéro de série unique, le type, le modèle, l'état (i.e fonctionnel ou en panne ou bien en réparation), le numéro d'immobilisation unique, un champ destiné aux remarques pour un ordinateur (exemple : manque souris ou clavier...), le numéro d'inventaire unique, le nom du fabricant, le statut (i.e Dans inventaire FDS, Hors FDS dans inventaire UM ou bien Hors inventaire UM), le détail concernant la sortie de l'inventaire de l'ordinateur et enfin, une information sur le nombre d'année de garantie, sans oublier sa date de début gestion par un centre responsable.
- pour le **lieu** où sont installés les ordinateurs, on a : l'identifiant unique du lieu, le numéro de la salle, le nom de la salle, le niveau de l'étage, le nombre d'ordinateurs qu'il peut y avoir dans une salle représentant sa capacité, le numéro du bâtiment, le nom du bâtiment, le numéro du campus auquel appartient le bâtiment et le nom du campus.
- concernant la **commande** à laquelle un ordinateur est rattaché, on a : son identifiant et son numéro SIFAC uniques, la date début de garantie de l'ordinateur et le nom du fournisseur.

- à propos du **centre responsable** qui gère un/des ordinateurs et/ou un lieu, on a : l'identifiant unique de ce centre responsable, son nom, son code et son type de structure.
- concernant la **personne** qui gère le lieu où se trouve l'ordinateur et/ou qui dirige un centre responsable on a : l'identifiant unique de la personne, son nom, son prénom, son n° de téléphone, son email, son rôle dans l'UM et sa fonction.

Quelques attributs importants :

- **IdLieu** : est la concaténation des attributs suivants avec le séparateur " / " : NumSalle, NumBatiment, Num-Campus et NumNiveau. Il permet d'assurer de façon optimal l'unicité du lieu.

- **compteurModification** : représente le numéro de modification pour un ordinateur donné. Il permet de garder une trace du nombre de modifications effectuées sur un ordinateur.

- **typeFiche** : grâce à cet attribut qui est dans l'entité Gere_Par_Hist, on pourra distinguer les différents types de fiches d'inventaire. Il prendra l'une des trois valeurs suivantes : Entrée, Sortie ou Modification.

- **InfoFicheInventaire** : cet attribut permet d'avoir une concaténation de toutes les informations sous forme CSV utiles à l'édition des fiches d'entrée, de sortie et de modification d'inventaire.

Les informations à concaténer dans cet attribut sont : Le nom du responsable de la commande, le prénom du responsable de la commande, le numéro de téléphone du responsable de la commande, l'adresse mail du responsable de la commande, le centre responsable de la commande, le numéro de commande SIFAC, le modèle de l'ordinateur, type ainsi que son fabricant, fournisseur et numéro de série. Nous ajoutons à cela l'ancien numéro campus où il se trouvait, l'ancien numéro de bâtiment, l'ancien numéro d'étage ainsi que son ancien numéro de salle et son numéro campus, numéro de bâtiment, numéro d'étage et son numéro de salle actuel. La concaténation se poursuit avec la date de mise en service, le numéro d'immobilisation, le numéro d'inventaire, les détails de la sortie d'inventaire, la date de fin de gestion, le nom de l'ancien responsable de l'ordinateur, le prénom de l'ancien responsable de l'ordinateur, le numéro de téléphone de l'ancien responsable de l'ordinateur, l'adresse mail de l'ancien responsable de l'ordinateur, le nom et prénom du responsable de l'ordinateur ainsi que son numéro de téléphone et son adresse mail comme le montre la *figure 1*.

InfoFicheInventaire (l'ordre des informations dans cet attribut suit l'ordre des attributs cité ci-dessus)	
entrée d'inventaire	Dupont;Jean:0102030405;jd@email.com;Informatique; NULL ;Optiplex;Workstation;Dell;8GH5TRYX;XXXX;01/36/02/14;0,2020-01-01;numImm0123;UM005000;XXXXXX;Dupont;Jacques;0203040506;jd1@email.com
sortie d'inventaire	XXX ;Optiplex;Workstation;Dell;5THG67R;XXXX;01/36/02/14;0,X;numImm0567;UM005010;DON A UNE ASSOC;2020-02-02;XXXX;Dupont;Jeanne;0908070605;jdn@email.com
modification d'inventaire	XXX ;Optiplex;Workstation;Dell;4RTG68N;01/36/02/14;0;01/36/02/44;0,X;numImm0400;UM005222;K;2020-03-04;Dupont;Jack;060606060600;jdk@email.com;Dupond;Jane;0707070707;jdd@email.com

FIGURE 1 – Exemple de l'attribut InfoFicheInventaire

- Pour une **entrée d'inventaire**, les informations utiles ont été enregistrées dans l'attribut InfoFicheInventaire.

Cependant il y a des informations, pour lesquels des emplacements sont prévus dans cet attribut, qui sont inutiles pour une fiche d'entrée. Ces informations inutiles sont les suivantes : l'ancien lieu (n° campus, n° bâtiment, n° étage, n° salle), la date de fin de gestion et les informations de l'ancien responsable de la machine. Toutes les autres informations sont utiles et donc enregistrées.

Les emplacements prévus pour les informations non utiles dans l'attribut InfoFicheInventaire seront donc vides pour une entrée, ainsi pour mieux comprendre, les champs vides ont été marqués d'une croix X rouge pour l'exemple présent dans la *figure 1* et au niveau de la base de données, il y aura un champ vide ”.

Dans le cas où l'information serait utile, mais l'attribut récupéré dans la base serait égale à NULL (par exemple, le n° immobilisation, le n° commande SIFAC ou bien le n°inventaire), on notera cette information NULL dans l'attribut InfoFicheInventaire pour la distinguer des informations inutiles, qui ont été représentées par un champ vide.

- Pour une **modification d'inventaire**, les informations inutiles sont : les informations de la personne en charge de la commande, le fournisseur, dateDebGarantie et detailSortieInventaire.
- Pour une **sortie d'inventaire**, les informations inutiles sont : les informations de la personne en charge de la commande, le fournisseur, l'ancien lieu, la date de début de garantie et les informations sur l'ancien responsable de la machine.

3.1.2 Modèle Entité-Association

Au niveau conceptuel, on distingue les entités des associations qui représentent les liens entre ces entités. Dans notre modèle entité-association (figure 2), on représente les entités par des rectangles et les associations par des ovales. Chacune des deux peuvent avoir des attributs.

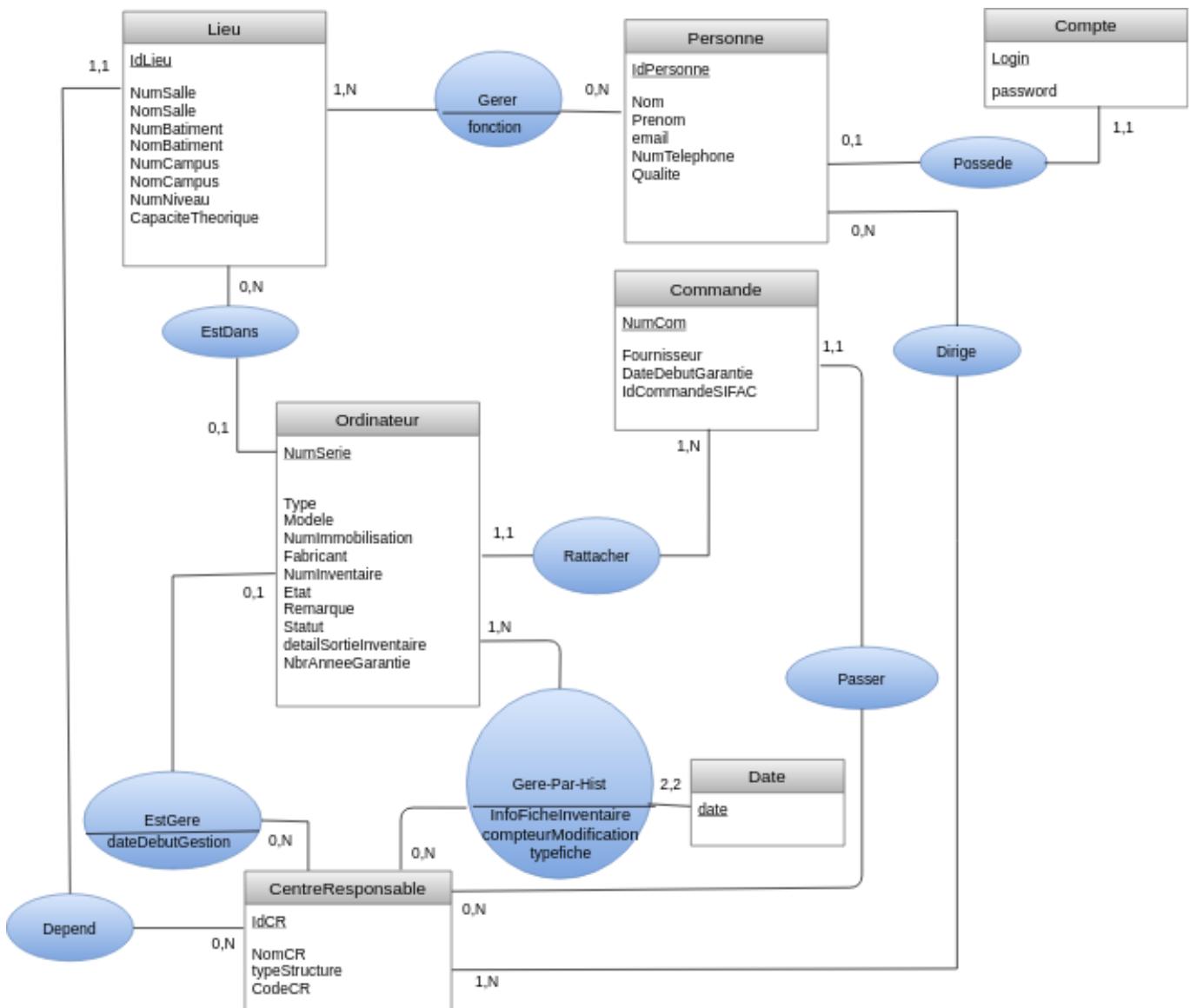


FIGURE 2 – Modèle Entité-Association

Analyse du modèle entité association :

- **Entité Ordinateur** : Un ordinateur donné a plusieurs caractéristiques. Pour cette raison, on a besoin de créer une entité ordinateur qui contiendra plusieurs informations : un numéro de série, un type, un modèle, un numéro d'immobilisation (donné par la personne s'occupant de la gestion des finances), le fabricant qui est la marque de l'ordinateur, un numéro d'inventaire (donné par la personne s'occupant de la gestion des finances), l'état d'un ordinateur (fonctionnel, en panne ou en réparation), le statut d'un ordinateur qui représente son positionnement au niveau des inventaires (dans inventaire FDS, hors inventaire FDS dans UM, hors inventaire UM), le détail de sortie de l'inventaire UM dans le cas où un ordinateur sort de l'inventaire UM, un attribut remarque et un nombre d'années de garantie nous permettant de savoir à quel moment il faudra le renouveler.
- **Entité CentreResponsable** : Pour l'entité CentreResponsable qui gère un/des ordinateurs et/ou un lieu, on a besoin d'un identifiant qui sera propre à chaque département, le nom du centre responsable, un type de structure représentant la structure de l'UFR et d'un codeCR qui est composé de 4 chiffres.
- **Entité Lieu** : Pour pouvoir préciser l'emplacement exact d'un ordinateur, on aura besoin d'une entité Lieu. Cette entité est composée de l'ensemble des attributs suivants : le numéro de la salle, du bâtiment, du campus et du niveau(étage), ainsi que le nom de la salle, du bâtiment et du campus. On a également un attribut précisant la capacité théorique (en ordinateurs) de la salle. Concernant l'identifiant de cette entité IdLieu, il sera représenté par la concaténation de quatre attributs parmi ceux cités ci-dessus : le numéro de salle, du bâtiment, du campus ainsi que le niveau.
- **Entité Commande** : Pour l'entité Commande à laquelle est rattachée un ordinateur, on a besoin d'un identifiant qui sera propre à chaque commande, d'un idCommandeSIFAC permettant d'identifier la commande par son numéro SIFAC, le nom du fournisseur et la date de début de garantie.
- **Entité Personne** : Pour une personne qui gère le lieu où se trouve l'ordinateur et/ou qui dirige un centre responsable, on aura besoin d'un identifiant, un nom, prénom, numéro de téléphone, une adresse mail et d'une qualité décrivant la fonction d'une personne au sein de l'université.
- **Entité Compte** : Pour qu'une personne puisse se connecter, elle aura besoin de créer un compte. Pour cela, on a besoin de définir une entité Compte qui contient son identifiant qui est un login, et d'un mot de passe.
- **Association Rattacher** Un ordinateur est rattaché à une seule et unique commande, une commande contient au moins un ordinateur.
- **Association Passer** Un CentreResponsable peut passer plusieurs Commandes (éventuellement aucun), et une Commande peut-être passée par un seul et unique CentreResponsable.
- **Association EstDans** Un ordinateur peut se trouver dans un seul lieu ou aucun (si sortie de l'inventaire UM). Mais il est toujours présent dans la base de données avec une éventuelle indication de sa situation dans detailSortieInventaireUM. Un lieu peut comporter plusieurs ordinateurs (éventuellement aucun).
- **Association Depend** Un lieu dépend d'un seul et unique centre responsable et un centre responsable administre plusieurs lieux (éventuellement aucun).
- **Association Gerer** Un lieu est géré par au moins une personne, une personne peut gérer plusieurs lieux (éventuellement aucun). Dans l'association, il y a un attribut *fonction* qui représente la fonction de la personne qui gère le lieu.

- **Association Dirige** Une personne gère plusieurs centres responsables (éventuellement aucun) et un centre responsable est dirigé par au moins une personne.
- **Association Possede** Une personne possède un seul compte ou aucun , un compte appartient à une seule et unique personne.
- **Association EstGere** Un ordinateur est géré par un seul centre responsable (éventuellement aucun), un centre responsable gère plusieurs ordinateurs (éventuellement aucun). Dans cette association, on trouve la date de début de gestion par le centre responsable qui est stockée dans l'attribut dateDebutGestionCR. On garde cette information pour pouvoir la réutiliser dans l'association Gere_Par_Hist, et ainsi, gérer l'historique.
- **Association Gere_Par_Hist** : Cette association va nous permettre de connaître la durée de gestion d'un ordinateur par un centre responsable. Elle a pour cardinalité 2 avec l'entité Date représentant la date de début et date de fin de la gestion par un centre responsable.
 - Un ordinateur est géré par un ou plusieurs centres responsables pendant des durées différentes (mais par un et un seul centre pour une durée donnée) et un centre responsable gère plusieurs ordinateurs pendant des durées différentes (éventuellement aucun).
 - Dans cette association, on a plusieurs attributs. L'attribut InfoFicheInventaire contient, sous forme concaténée, les informations nécessaires au remplissage des fiches inventaires d'entrée, de modification et de sortie pour un ordinateur donné. L'attribut typeFiche aura pour valeur soit "Entrée" ou "Modification" ou bien "Sortie". L'attribut compteurModification représente le numéro de modification pour un ordinateur donné. Il permet de garder une trace du nombre de modifications effectuées sur un ordinateur.

3.1.3 Modèle logique

Après la création du modèle conceptuel, nous sommes passés à la création du modèle logique relationnel avant de passer au modèle physique et à son implémentation.

Le modèle logique est le suivant :

- **Ordinateur**(NumSerie, Type, Modele, Fabricant, NumImmobilisation, NumInventaire, Etat, Remarque, Statut, detailSortieInventaire, NbrAnneeGarantie, dateDebutGestion, #IdCR, #IdLieu, #NumCommande)
- **Commande**(NumCommande, IdCommandeSIFAC, Fournisseur, DateDebutGarantie, #IdCR)
- **CentreResponsable**(IdCR, NomCR, typeStructure, CodeCR)
- **Personne**(IdPersonne, Nom, Prenom, email, NumTelephone, Qualite)
- **Lieu**(IdLieu, NumSalle, NomSalle, NumBatiment, NomBatiment, NumCampus, NomCampus, NumNiveau, CapaciteTheorique, #IdCR)
- **Compte**(Login, Password, #IdPersonne)
- **Gerer**(#IdPersonne, #IdLieu, fonction)
- **Gere_Par_Hist**(#IdCR, #NumSerie, #dateDebutGestion, dateFinGestion, InfoFicheInventaire, compteurModification, typefiche)
- **Dirige**(#IdPersonne, #IdCR)

Note :
clé primaire

#clé étrangère

3.2 Cas d'utilisation

3.2.1 Entrée inventaire

La base de données est essentielle pour enregistrer les informations des ordinateurs, ainsi que leur lieu et leur centre responsable. Elle doit être alimentée via l'application web, en remplissant un formulaire pour ajouter un ordinateur, ou bien via un fichier CSV téléchargeable, à pré remplir et à importer sur l'application web, dans lequel se trouve toutes les informations importantes d'un ou plusieurs ordinateurs. Cela automatisé l'enregistrement et facilite cette action pour l'utilisateur. Toutes les informations relatives à l'ordinateur et à la commande à laquelle il est rattaché seront ensuite récupérées sur le site web et envoyées à la base de données.

Voici les différentes étapes résumant l'ajout d'un ordinateur au niveau de la base de données :

- Insérer dans la table **Commande** un nouveau tuple en renseignant les valeurs pour les attributs suivants : l'idCR du centre responsable ayant passé la commande, IdCommandeSIFAC, le Fournisseur de la commande et DateDebutGarantie. L'attribut NumCommande est un attribut à valeur auto-incrémenté.
- Insérer dans la table **Ordinateur** un nouveau tuple en renseignant les valeurs pour les attributs suivants : numSerie, Type, Modele, Fabricant, numImmobilisation, numInventaire, nbrAnneeGarantie.
- Les attributs Etat et Statut dans la table **Ordinateur** prennent respectivement les valeurs "Fonctionnel" et "Dans inventaire FDS"
- Les attributs détailSortieInventaire et Remarque dans la table **Ordinateur** prennent la valeur NULL
- L'attribut dateDebutGestionCR dans la table **Ordinateur** aura pour valeur la date de validation du formulaire ou de l'importation (date du jour)
- Les attributs idCR et idLieu dans la table **Ordinateur** prennent les valeurs présentes dans les colonnes associées du fichier CSV ou dans les champs du formulaire sur l'application web.
- L'attribut NumCommande dans la table **Ordinateur** aura la même valeur que le NumCommande précédent dans la table **Commande**.
- Lors d'une entrée d'un nouvel ordinateur dans la base de données, on enregistre sa toute première ligne dans la table qui gère l'historique, **Gere_Par_Hist**. On enregistre l'idCR et le NumSerie de l'ordinateur. L'attribut dateDebutGestion prendra la valeur 1900-01-01 car on ne peut pas plus remonter dans le temps sur phpMyAdmin et la dateFinGestion aura pour valeur la date du jour. Cette période unique permet de préciser que nous n'avions pas d'informations sur l'ordinateur avant la date du jour. Il faudra également compléter l'attribut InfoFicheInventaire avec les informations nécessaires à l'édition de la fiche Entrée d'inventaire. Ces informations sont : les informations de la personne qui dirige le CentreResponsable qui a passé la commande (nom, prénom, téléphone, mail, UFR) ; IdCommandeSIFAC ; les informations sur l'ordinateur (type, fabricant, fournisseur, n° série, lieu, date de mise en service, n°immo, n°inventaire) ; les informations de la personne qui dirige le CentreResponsable auquel appartient l'ordinateur (nom, prénom, téléphone, mail). Enfin, l'attribut typeFiche prendra la valeur Entrée et l'attribut compteurModification prendra la valeur 1.

La deuxième action à effectuer sur l'inventaire est sa modification. Pour cela, nous avons distingués trois cas d'utilisation : modification d'inventaire, édition caractéristiques et changement d'état.

3.2.2 Modification inventaire

Une modification d'inventaire FDS est un changement de lieu et/ou de centre responsable dans la FDS. Ce changement doit générer un ajout dans l'historique de l'ordinateur sous le type "Modification". Une fois le changement validé, le nouveau centre responsable et/ou le nouveau lieu sont récupérés dans l'application web et envoyés à la base de données.

Voici les différentes étapes résumant la modification d'un ordinateur au niveau de la base de données :

- Dans la table **Gere_Par_Hist**, on insère pour chaque ordinateur un tuple en renseignant les valeurs pour les attributs suivants : idCR, NumSerie et dateDebutGestion que l'on récupère dans la table **Ordinateur**. C'est la date de début de gestion de cet ordinateur par ce centre responsable.
- dateFinGestion aura pour valeur la date du jour. L'attribut typefiche aura pour valeur "Modification". L'attribut

3.2.5

compteurModification, qui représente le nombre de modifications pour chaque ordinateur, sera augmenté de 1 par rapport au compteur de la dernière modification pour le même NumSerie à l'aide d'un trigger.

- Dans l'attribut InfoFicheInventaire, on enregistre la concaténation des informations nécessaires à l'édition de la fiche inventaire Modification. Ces informations sont : les informations sur l'ordinateur (modele, type, fabricant, n° série, ancien lieu, nouveau lieu, n°immobilisation, n°inventaire), la date de modification (correspond à la date de fin de gestion), les informations de la personne qui dirige l'ancien CR auquel appartenait l'ordinateur (nom, prénom, téléphone, mail), les informations de la personne qui dirige le nouveau CR auquel appartient l'ordinateur (nom, prénom, téléphone, mail).
- Dans la table ordinateur, on remplace la DateDebutGestionCR par la date du jour, le IdCR par le nouvel IdCR et l'ancien IdLieu par le nouvel idLieu, selon les informations fournies par l'utilisateur.

3.2.3 Édition caractéristiques

L'édition de caractéristiques a été pensé pour pouvoir modifier les informations d'un ordinateur (Type, Modele, NumImmobilisation, Fabricant, NumInventaire, Remarque, NbrAnneeGarantie). Cette action n'a pas d'impact sur l'historique de l'ordinateur puisque son lieu ou son centre responsable n'a pas été modifié.

Au niveau de la base de données, il suffit de mettre à jour les champs modifiés (Type, Modele, NumImmobilisation, NumInventaire, Remarque, NbrAnneeGarantie) dans la table **Ordinateur**.

3.2.4 Changement d'état

La dernière modification d'un ordinateur à traiter est le changement d'état parmi les valeurs suivantes : Fonctionnel, En panne, En réparation. Cette modification d'ordinateur a été séparée du changement de caractéristiques par soucis d'ergonomie. En effet, nous avons décidé que le changement d'état devrait être un cas d'utilisation à part entière afin de faciliter cette action courante à l'utilisateur, en lui proposant une page web dédiée.

Dans la base de données, on met à jour l'état de l'ordinateur avec le nouvel état sélectionné.

Enfin, la dernière action a effectué sur un ordinateur est sa sortie de l'inventaire. Là encore, deux cas sont distingués : la sortie de l'inventaire FDS et la sortie de l'inventaire UM.

3.2.5 Sortie inventaire FDS dans UM

Un ordinateur peut être sorti de l'inventaire FDS tout en restant dans l'inventaire UM. Cette sortie s'apparente à une modification d'inventaire puisque qu'on change le lieu et/ou le centre responsable, à la seule différence que ces informations (lieu et centre responsable) sont en dehors de la FDS mais dans l'UM. Cette sortie génère également une nouvelle ligne dans l'historique de l'ordinateur sous le type "Modification".

Dans la base de données, on aura les modifications suivantes :

- Dans la table **Ordinateur**, on modifie la dateDebutGestionCR à la date du jour, le nouvel idCR ainsi que le nouvel idLieu selon les informations fournies par l'utilisateur, et on met à jour le statut des ordinateurs sélectionnés à "Hors inventaire FDS dans UM"
- Pour chaque ordinateur, créer un tuple dans **Gere_Par_Hist** en renseignant les valeurs pour les attributs suivants : idCR, numSerie, dateDebutGestion (récupéré dans la table **Ordinateur**), dateFinGestion (date du jour) et on incrémente le compteurModification.
- Enregistrer dans l'attribut InfoFicheInventaire la concaténation des informations nécessaires à l'édition de la fiche inventaire Modification. Ces informations sont : les informations sur l'ordinateur (modele, type, fabricant, n° série, ancien lieu, nouveau lieu, n°immobilisation, n°inventaire), la date de modification (correspondant à la

date de fin de gestion), les informations de la personne qui dirige l'ancien CR auquel appartenait l'ordinateur (nom, prénom, téléphone, mail), les informations de la personne qui dirige le nouveau CR auquel appartient l'ordinateur (nom, prénom, téléphone, mail).

- Spécifier dans l'attribut typeFiche "Modification"

3.2.6 Sortie inventaire UM

La sortie d'un ordinateur de l'inventaire UM implique que plus aucune action ne peut être effectué sur celui-ci. Cette sortie d'inventaire engendre une ligne d'historique -la dernière- de type "Sortie" pour cet ordinateur. L'utilisateur peut choisir de donner des détails, ou non, sur cette sortie, en remplissant un champ texte.

Voici les différentes étapes résumant la sortie UM d'un ordinateur au niveau de la base de données :

- Dans la table **Ordinateur**, pour chaque ordinateur, mettre à jour l'attribut Statut à "Hors inventaire UM".
- Dans la table **Ordinateur**, pour chaque ordinateur, enregistrer dans l'attribut detailSortieInventaireUM le contenu du champ texte rempli par l'utilisateur.
- Créer une ligne dans **Gere_Par_Hist** en renseignant les valeurs pour les attributs suivants : idCR, numSerie, dateDebutGestion (récupéré dans la table **Ordinateur**), dateFinGestion (date du jour) et incrémenter l'attribut compteurModification.
- Enregistrer dans l'attribut InfoFicheInventaire la concaténation des informations nécessaires à l'édition de la fiche inventaire Sortie. Ces informations sont : les informations sur l'ordinateur (modele, type, fabricant, n° série, lieu, n° immobilisation, n° inventaire, detailSortieInventaire), la date de sortie (correspondant à la dateFinGestion) et les informations de la personne qui dirige le CR auquel appartenait l'ordinateur (nom, prénom, téléphone, mail, UFR). - Initialiser l'attribut typeFiche à "Sortie"

3.3 Interface graphique

Cette application web a pour objectif de faciliter la gestion des ordinateurs ainsi que leur traçabilité au sein de la faculté en proposant toutes les fonctionnalités nécessaires dans un seul et même endroit, notre interface. Nous avons créer pour cette application une première page de connexion qui demandera un identifiant et un mot de passe permettant de bénéficier de tous les droits sur toutes les différentes actions mises en place.

Avant de présenter la maquette de chaque page, nous allons commencer par décrire la maquette générale de l'application présente dans la *figure 3* puisque toutes les pages sont conçues suivant le même squelette. Tout d'abord, l'en tête de notre interface comporte le logo de l'université à gauche ainsi que le lien vers la page de recherche et le panier à l'autre extrémité. De plus, nous avons placé le menu à gauche de l'écran. La première partie du menu représente les actions indépendantes du panier, c'est à dire que quelque soit son état, plein ou vide, nous pouvons effectuer les actions présentes dans cette partie du menu telles que l'ajout d'ordinateur, l'édition des fiches inventaires et la consultation des ordinateurs à renouveler et en panne ou en réparation. Enfin, les actions dépendantes du panier se trouvent dans la deuxième partie du menu. Ces actions obligent l'utilisateur à avoir au minimum un ordinateur dans son panier pour effectuer une des actions parmi la modification d'inventaire, le changement d'état, la sortie de l'inventaire de la FDS, la sortie de l'inventaire l'inventaire de l'UM ou encore une édition des caractéristiques des ordinateurs.

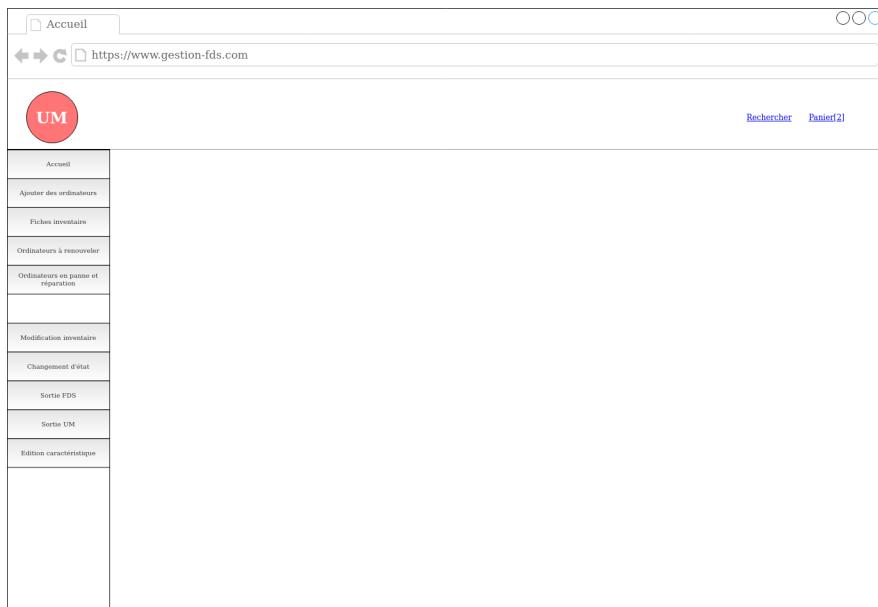


FIGURE 3 – Maquette - Squelette du site

Au niveau des fonctionnalités, nous avons décidé de placer au centre de chaque page le visuel de l'action choisie dans le menu par l'utilisateur. De plus, chaque fois que celui-ci choisira une action, son bouton dans le menu changera de couleur pour lui rappeler où il se trouve dans le site web.

Premièrement, nous devons permettre à l'utilisateur d'ajouter un ou plusieurs ordinateurs dans la base de données. Cela peut se faire manuellement et individuellement par l'entrée des informations via le formulaire où chaque information doit être insérer dans le champ approprié. Parmi ces champs, certains sont facultatifs et d'autres requis pour pouvoir valider le formulaire et insérer le nouvel ordinateur en cliquant sur le bouton d'ajout. L'ajout d'un ordinateur peut également se faire par le biais d'un fichier CSV dont la trame est téléchargeable afin de faciliter l'insertion à l'utilisateur et d'assurer la complétude des informations présentes et nécessaires telles que les précisions concernant l'ordinateur, son emplacement, son centre responsable, ainsi que sa garantie. Sachant que le numéro d'étiquette (i.e. le numéro d'inventaire) n'est pas forcément connu au moment de la saisie, nous permettons sa saisie de manière différée via la page d'édition caractéristiques. Il n'est donc pas obligatoire de le renseigner dans le formulaire sur l'interface ou le fichier CSV. Il en est de même pour le numéro d'immobilisation. Il est important de noter que la première ligne du fichier CSV est destiné à la trame, elle ne sera donc pas ajoutée à la base de données. Il faut donc impérativement entrer les informations des ordinateurs dès la deuxième ligne.

De plus, il est important de permettre la consultation des fiches recensant les changements de centres responsables ainsi que les changements de lieu d'un ordinateur dans le temps. Afin de consulter cet historique, l'utilisateur doit remplir un formulaire en précisant le type de fiches inventaires qu'il souhaite consulter (entrée, modification ou sortie d'inventaire), ainsi que deux dates constituant la période souhaitée. Une fois le formulaire validé, une nouvelle page se charge affichant un tableau avec les informations sur les ordinateurs correspondant aux critères du formulaire. Un fichier CSV récapitulatif est également mis à disposition pour être téléchargé.

Enfin, au niveau des actions indépendantes du panier, nous avons également la possibilité de consulter les ordinateurs à renouveler et ceux en panne et en réparation. Dans le cas où nous voulons visualiser les ordinateurs à renouveler pour une date de choisie nous devons renseigner cette date dans le champs associé et valider la recherche, une fois validée nous obtenons le tableau contenant l'ensemble des ordinateurs à renouveler ainsi que leurs informations. Si on veut visualiser l'ensemble des ordinateurs en panne et/ou en réparation nous devons préciser l'état voulu et nous obtenons le tableau de leurs informations.

Avant de pouvoir effectuer les actions dépendantes du panier il faut remplir celui-ci. Pour ceci il faut

se rendre dans la page de recherche, dont la maquette est présente dans l'annexe 8.2, et remplir les champs nécessaires, parmi ceux-ci le numéro de série, le lieu, le numéro de bâtiment, l'état, la garantie, le statut ou le fait qu'ils aient un numéro d'inventaire ou d'immobilisation. Une fois la recherche validée nous obtenons le tableau de tous les ordinateurs sélectionnables comme dans l'annexe 8.3. Une fois les ordinateurs voulus sélectionnés nous pouvons les ajouter au panier qui se mettra à jour avec les nouvelles machines comme dans l'annexe 8.4. Lorsqu'on se trouve dans le panier on peut choisir de supprimer tout son contenu ou un ordinateur en particulier grâce au bouton en fin de chaque ligne.

En ce qui concerne les actions dépendantes du panier, une fois le panier rempli, nous pouvons accéder à la page effectuant les modifications inventaire qui contient des listes déroulantes permettant de choisir parmi le nouveau lieu et/ou le nouveau centre responsable dans la FdS avant de valider la modification pour l'ensemble du panier. Il en est de même pour la sortie inventaire FDS où le contenu des listes déroulantes est différent puisqu'il contient les lieux et les centres responsables hors FDS mais dans UM. Au niveau du changement d'état il suffit de se rendre dans la page associée et de choisir le nouvel état à attribuer au panier. Pour sortir les ordinateurs sélectionnés définitivement de l'inventaire UM, il faut préciser le détail de la sortie dans le champ texte proposé et valider la sortie. Enfin, pour modifier les informations d'un ordinateur, il faut se rendre dans la page d'édition caractéristiques et remplir les champs voulus afin de mettre à jour les données des ordinateurs.

Au niveau de la navigation dans le site web, nous avons détaillé les différents renvois possible à partir de chaque page à l'aide du diagramme de navigation présent dans l'annexe 8.5. Par soucis de visibilité, le diagramme a été découpé en deux parties ce qui signifie que la deuxième image est la suite de la première.

Premièrement, les actions dépendantes du panier se trouvent dans la partie supérieur du diagramme, avec les flèches pointant vers le haut, quant à la partie inférieur elle concerne les actions indépendantes du panier. Ensuite, l'accès au menu est possible depuis toute page de l'application du moment que nous sommes connecté. De plus, chaque page contenant un formulaire possède un bouton permettant de valider son choix ou de l'annuler.

Par conséquent, il y a des formulaires renvoyant un résultat dans une nouvelle page comme la recherche ou l'historique des modifications, entre autres, et des formulaires tels que l'ajout d'ordinateur ou la modification d'inventaire qui, après validation, renvoient sur la même page avec un message détaillant si le formulaire a été soumis avec succès ou si une erreur s'est produite.

Nous avons également décidé de désactivé le "back", la fonction retour du navigateur, pour empêcher des pertes d'informations, pour plus de sécurité et pour ne pas créer d'erreur dans la navigation. Lorsque le bouton de retour est donc sollicité, il ramène vers la page de recherche. Enfin, ce diagramme nous a permis de schématiser les différentes actions et redirections entre chaque page web ainsi que l'impact de chaque clic sur les différents boutons à implémenter.

3.4 Diagrammes d'état-transition

Pour décrire le comportement interne de notre application, nous avons utilisé les diagrammes d'état-transition qui sont les seuls, de la norme UML, à offrir une vision complète et non ambiguë de l'ensemble des comportements de l'élément auquel il est attaché. Concrètement, ce diagramme est un graphe qui représente les séquences possibles d'états et d'actions qu'une instance peut traiter au cours de son cycle de vie.

Un état est un visuel associé à un ensemble d'actions possibles et on peut passer d'un état à un autre par un déclencheur (événement/action), en fonction d'une condition avec un certain effet. C'est ce qu'on a appellé une transition.

L'objectif est d'avoir un visuel homogène sur toutes les états de l'application avec des zones définies communes et de voir quelles actions sont possibles et permet d'accéder à quel état.

Par la suite, nous présenterons les états, les actions et les termes utilisés pour réaliser nos diagrammes disponibles en annexe 8.19.

Représentation d'une transition : évènement(paramètres)[conditions]/action.

- Liste des états : E_RSR (recherche sans résultat - état initial), E_EC (édition des caractéristiques), E_CE (changement d'état) , E_MI (modification inventaire), E_SF (sortie fds), E_SU (sortie um), E_AO (ajout ordi), E_FI (fiche inventaire), E_RAR (recherche avec résultat), E_EP (édition du panier), E_C (connexion), E_AI (aide), E_AN (A propos), E_RN-PR (ordinateurs à renouveler et ordinateurs en panne/en réparation)

- Liste des actions générales : EC (édition caractéristiques), CE (changement état), MI (modification inventaire), SF (sortie FdS), SU (sortie UM), AO (ajouter ordinateurs), FI (fiches inventaire), RN (ordinateurs à renouveler), PR (ordinateurs en panne/ en réparation), RC (recherche), EP (édition panier), DX (déconnexion), AI (aide), AN (à propos), RetNavig (requête http précédente de l'historique), AvNavig (requête http suivante de l'historique)

- Liste des actionsTab : TS (tout sélectionner), EX (exporter)

- Liste des actions formulaires : Go_XX (clic sur un bouton), AP_XX (ajout panier), SP (supprimer sélection panier)

- Liste des zones de l'écran : TA (accueille un titre), ZF (accueille un formulaire), ZB (accueille des boutons de sélections et d'actions), ZR (accueille un tableau de résultats avec noms de colonne et possibilité de sélect/deselect chaque ligne), QnP (accueille un entier, le nbre d'items du panier)

Les diagrammes présentés reposent tous sur plusieurs principes communs. Dans le cas où l'utilisateur souhaite effectuer une action alors que son panier est vide un message d'erreur lui est affiché. Si l'ordinateur, sur lequel on veut effectuer l'action, a déjà été modifié par la même action avec les mêmes valeurs nous précisons à l'utilisateur que cette modification a déjà été effectuée sinon, si aucune erreur n'est détectée le message de modification effectuée est affiché. Enfin, à partir de n'importe quel état nous pouvons aller sur n'importe quel autre état. En effet il est possible de passer d'une page à une autre avec le même panier avant de réaliser une l'action voulue.

3.5 Choix de l'architecture

Pour ce projet, nous avons décidé de choisir l'architecture MVC, qui est un acronyme pour Model-Vue-Controller. Cette architecture, utilisée par bon nombre de professionnels, permet d'assurer la maintenance du code et son évolution. Elle permet aussi de faciliter le travail en groupe. Comme l'indique son titre, le pattern MVC est divisé en trois parties : le modèle, la vue et le contrôleur. La vue se contente de traiter les affichages et ne contient pratiquement que du code HTML (à l'exception de certaines fonctions PHP d'affichage). Le modèle gère les données de l'application. C'est lui qui fait appel à la base de données et qui contient les fonctions récupérant les données souhaitées par des requêtes SQL. Ces données sont ensuite traitées par le contrôleur. Le contrôleur est l'intermédiaire entre les deux parties précédentes. Il effectue les décisions en fonction des données demandées au modèle. Il les analysent puis renvoie du code HTML à afficher sur la vue. L'architecture MVC se résume donc par la figure 4 suivante :

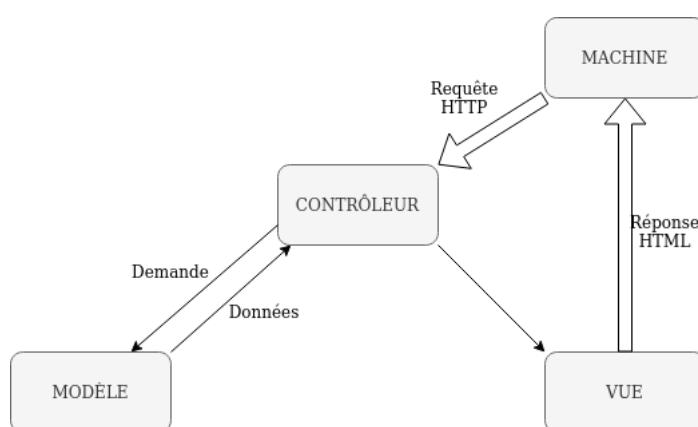


FIGURE 4 – Architecture MVC

4 Réalisation

4.1 Choix technologiques

4.1.1 Choix du SGBD

Lors de notre apprentissage en base de données, nous avons été en premier lieu introduit à Oracle, Cependant, ce dernier est une application de gestion de base de données payante, ce qui n'en fait pas un outil idéal pour un groupe d'étudiants. De part notre expérience avec l'application phpMyadmin, qui est une application web de gestion pour les systèmes de gestion de base de données (SGBD) MySQL, nous nous sommes donc tournés vers MySQL couplé du paquet phpMyAdmin.

4.1.2 Choix des langages

Au cours de notre formation en licence, nous avons appris à maîtriser de nombreux langages, en particulier HTML/CSS/PHP, qui ont été largement étudiés dans la création d'application web. Il nous a donc paru naturel de choisir ces langages, principalement utilisé pour produire des pages web dynamique liées à une base de données. Nous avons également introduit du JQuery, qui est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter la sélection d'éléments ainsi que la manipulation du code HTML d'affichage et la liaison avec les informations présentes dans la base de données. Par soucis d'ergonomie, nous avons choisi d'utiliser l'extension PDO (PHP Data Objects) qui est une extension PHP qui définit une interface d'accès à une base de données, pour effectuer les traitements en lien avec la base de données sur l'application. Le principal avantage du PDO est qu'il permet une abstraction pour l'accès aux données. C'est-à-dire que les fonctions pour exécuter des requêtes et pour récupérer des données sont les mêmes, quelque-soit le serveur SQL utilisé (MySQL, PostgreSQL, ...). En effet, si la base de données venait à changer (vers Oracle), l'application web n'en sera pas impacté puisque PDO est utilisable sur les deux SGBD.

4.1.3 Choix de l'hébergeur

Étant donné que la FdS ne disposait pas d'un hébergeur pour notre site web, nous avons dû trouver une solution alternative afin de pouvoir travailler collaborativement. De ce fait, nous avons recherché un hébergeur gratuit, ergonomique et rapide à mettre en place afin d'y placer nos fichiers et de visualiser l'avancée de nos travaux.

Nous avons fait le choix d'utiliser la plate-forme 000webhost car elle propose une utilisation gratuite et suffisante pour notre projet. Elle est sécurisée et permet l'intégration d'une base de données grâce à l'application phpMyadmin ainsi qu'une sauvegarde de tout le travail à tout moment et le transfert de fichiers FTP. De plus, elle propose les dernières versions de PHP avec toutes les bibliothèques à jour et MySQL ce qui garantit le bon fonctionnement des fonctions implémentées. Enfin, la plate-forme propose de bonnes performances au niveau du serveur ce qui nous a permis d'avoir un temps de réponse optimal entre toutes les pages du site web.

4.2 Implémentation de la base de données

Code d'implémentation de la base de données disponible sur le GitLab :

https://gitlab.info-ufr.univ-montp2.fr/BD_Parc_Info_TER_S6/TER_S6/tree/master/CODE_BD

Après avoir fait le modèle logique, le modèle physique a du être créé. Il représente l'implémentation de notre base de données dans un SGBD en utilisant un langage de définition de données SQL.

4.2.1 La création des tables

Nous avons tout d'abord commencé par la création des tables en conservant la structure du modèle logique sans oublier de préciser pour chaque table sa clé primaire et ses clés étrangères comme le montre par exemple les figures 5 et 6 pour la table Ordinateur.

```
CREATE TABLE Ordinateur (
    NumSerie varchar(30) ,
    Typee varchar(100) NOT NULL,
    Modele varchar(50) NOT NULL,
    NumImmobilisation varchar(50) UNIQUE,
    Fabricant varchar(50) NOT NULL,
    NumInventaire varchar(50) UNIQUE,
    Etat varchar(50) NOT NULL,
    Statut varchar(100) NOT NULL,
    detailSortieInventaire varchar(100) DEFAULT NULL,
    Remarque varchar(200) DEFAULT NULL,
    NbrAnneGarantie NUMERIC(2) NOT NULL DEFAULT 0,
    dateDebutGestion DATETIME NOT NULL ,
    IdCR INT(15) NOT NULL ,
    NumCommande INT(10) NOT NULL ,
    IdLieu varchar(25) NOT NULL,
    CONSTRAINT PK_Oordinateur PRIMARY Key (NumSerie)
);
```

FIGURE 5 – Création de la table Ordinateur

```
ALTER TABLE Ordinateur ADD CONSTRAINT fk1_Oordinateur FOREIGN KEY (IdCR) REFERENCES CentreResponsable (IdCR) ON DELETE CASCADE;
ALTER TABLE Ordinateur ADD CONSTRAINT fk2_Oordinateur FOREIGN KEY (IdLieu) REFERENCES Lieu (IdLieu) ON DELETE CASCADE;
ALTER TABLE Ordinateur ADD CONSTRAINT fk3_Oordinateur FOREIGN KEY (NumCommande) REFERENCES Commande (NumCommande) ON DELETE CASCADE;
```

FIGURE 6 – Implémentation des clés étrangères pour la table Ordinateur

Cet exemple est la traduction en SQL de la table citée précédemment tel que l'attribut NumSerie est la clé primaire et les attributs IdCR, IdLieu, NumCommande sont ses clés étrangères.

4.2.2 Les triggers

Grâce aux triggers, nous avons pu vérifier des contraintes tels que l'existence d'une information, mais aussi assurer l'intégrité de données, comme le fait d'imposer un ensemble fixe de valeur pour un attribut. Ils nous ont également permis de faire des mis à jours d'informations qui dépendent d'autres informations, comme c'est le cas pour table Gerer_Par_Hist qui est totalement rempli grâce à des déclenchements de triggers. En effet, après chaque insertion dans la table Ordinateur, un tuple représentant l'entrée dans l'inventaire de l'ordinateur sera inséré dans la table Gerer_Par_Hist à l'aide d'un trigger. Dans le cas d'une modification dans la table Ordinateur, un autre tuple sera inséré. Pour cela, on distingue deux cas : une modification du statut d'un ordinateur ou bien une modification du centre responsable et/ou du lieu. Pour le premier cas, ce tuple représentera une sortie d'inventaire dans le cas où le statut aura pour valeur "Hors inventaire UM". En revanche, pour le deuxième, ce tuple représentera une modification d'inventaire. L'insertion automatique de ces tuples respectent le format de l'attribut dédié à la gestion de l'historique d'inventaire, qui est InfoFicheInventaire tel que : pour chaque type d'inventaire, le trigger responsable sélectionnera uniquement les données nécessaires pour ce type (processus et autres exemples de triggers en annexe 8.6).

4.2.3 Les inserts

Après avoir implémenter notre base de données et créé tout ces triggers, nous avons effectué des insertions (exemple présent dans la figure 7) afin de pouvoir s'assurer du bon fonctionnement de notre base de données.

4.2.4

```
INSERT INTO Ordinateur(NumSerie, Typee, Modele, NumImmobilisation, Fabricant, NumInventaire, Etat, Statut, detailSortieInventaire, Remarque, NbrAnneGarantie, dateDebutGestion, IdCR, NumCommande, IdLieu) VALUES ('8CRMH5J', 'Workstation', 'OptiPlex 790', '57M4HG12', 'Dell', 'UM000001', 'Fonctionnel', 'Dans inventaire FDS', NULL, 'Manque souris', 2, '2016-02-12 00:00:00', 1, 1, '01/36/02/36.204');
```

FIGURE 7 – L'insertion d'un ordinateur

Exemple :

4.2.4 Les requêtes

Les requêtes permettent d'interroger une base de données et d'obtenir comme réponse des éléments de cette base vérifiant certaines conditions indiquées dans la requête exécutée. Ainsi, elles sont utilisées pour implémenter des fonctionnalités de l'application.

Quelques exemples de requêtes :

- La requête 1 suivante permet d'afficher les ordinateurs à renouveler (i.e en fin de garantie ou arrivant en fin de garantie avant l'année prochaine) :

```
SELECT NumSerie  
FROM Ordinateur O1, Commande  
WHERE O1.NumCommande=Commande.NumCommande  
AND SYSDATE()>=(SELECT ADDDATE(DateDebutGarantie, INTERVAL NbrAnneGarantie-1 YEAR) as DateFinGarantie  
FROM Ordinateur O2, Commande  
WHERE Commande.NumCommande=O2.NumCommande  
AND O2.NumSerie=O1.NumSerie);
```

Les résultats rendus par cette requête sont obtenus en sélectionnant les ordinateurs en fin de garantie ou arrivant en fin de garantie l'année prochaine. Pour ce faire, on additionne le nombre d'années de garantie (table Ordinateur) avec l'année de la date début de garantie (table Commande), pour obtenir la date fin de garantie, qui sera ensuite comparée avec la date du jour (SYSDATE) ou bien avec la date du jour de l'année prochaine (exemple requête 1 donné en annexe 8.7).

- La requête 2 suivante permet d'afficher les ordinateurs qui sont sous garantie :

```
SELECT NumSerie  
FROM Ordinateur O1, Commande  
WHERE O1.NumCommande=Commande.NumCommande  
AND SYSDATE()<(SELECT ADDDATE(DateDebutGarantie, INTERVAL NbrAnneGarantie YEAR) as DateFinGarantie  
FROM Ordinateur O2, Commande  
WHERE Commande.NumCommande=O2.NumCommande  
AND O2.NumSerie=O1.NumSerie);
```

Les résultats rendus par cette requête sont obtenus en sélectionnant les ordinateurs qui sont encore sous garantie le jour de la recherche faite par l'utilisateur dans l'application (exemple requête 2 en annexe 8.7).

- La requête 3 suivante permet d'afficher les ordinateurs qui sont en panne ou en réparation :

```
SELECT NumSerie
FROM Ordinateur
WHERE Ordinateur.Etat='En panne' OR Ordinateur.Etat='En reparation';
```

Les résultats rendus par cette requête sont obtenus en sélectionnant les ordinateurs tels que leur état est soit en panne ou en réparation (exemple requête 3 en annexe 8.7).

4.3 Implémentation du site web

4.3.1 Gestion du panier

La gestion du panier représente sans doute la plus grosse partie de l'implémentation de l'application. Nous avons dû mettre en place un système de sessions permettant l'enregistrement des ordinateurs sélectionnés afin de permettre à l'utilisateur d'appliquer une action dessus. Pour cela nous avons utilisé l'architecture MVC et Ajax ainsi que la bibliothèque JavaScript jQuery.

Sessions Une session est une période délimitée pendant laquelle un appareil informatique est en communication et réalise des opérations au service d'un client. Au niveau des sessions, PHP génère un ID de session qui est transmis automatiquement de page en page en utilisant un cookie, ce dernier est l'équivalent d'un fichier texte de petite taille, stocké sur le terminal de l'internaute. Il permet aux développeurs de sites web de conserver des données utilisateur afin de faciliter la navigation et de permettre certaines fonctionnalités. La durée de cette session est en général de 24 minutes et peut être modifier dans le fichier *php.ini* présent dans le dossier *etc/-php.ini* cependant sur notre hébergeur (000webhost) la modification de cette variable n'est pas permise. Afin de modifier la durée de la session il faut modifier la variable *session.gc_maxlifetime* en lui donnant une nouvelle valeur, représentée en seconde, pour la durée voulue. Par défaut le fichier comporte la valeur présentée par la figure 8.

```
; After this number of seconds, stored data will be seen as 'garbage' and
; cleaned up by the garbage collection process.
; http://php.net/session.gc-maxlifetime
session.gc_maxlifetime = 1440
```

FIGURE 8 – Gestion durée session

Cookies Les cookies sont un mécanisme d'enregistrement d'informations sur le client, et de lecture de ces informations. Ce système permet d'identifier et de suivre les visiteurs. Dans le fichier *php.ini* et sous *Chrome* cette session est valide tant que la fenêtre n'est pas fermée et que le temps n'est pas écoulé contrairement à *Firefox*, qui lui, garde la session active quelque soit le temps, du moment que l'ordinateur n'est pas redémarré. De plus, nous pouvons modifier la variable *session.cookie_lifetime* afin de lui donner une nouvelle valeur en secondes. Par défaut, la valeur présente dans le fichier *php.ini* est montrée dans la figure 9.

```
; Lifetime in seconds of cookie or, if 0, until browser is restarted.
; http://php.net/session.cookie-lifetime
session.cookie_lifetime = 0
```

FIGURE 9 – Gestion des cookies

Implémentation Dans un premier temps, dans la vue *selection.php* nous avons la liste des résultats obtenus suite à une recherche. Ces résultats sont sélectionnables à l'aide d'une checkbox. Lorsque la sélection est terminée et que l'utilisateur la valide, chaque ordinateur est ajouté au panier. Du côté serveur nous créons une

variable qui prend l'aspect d'un tableau vide pour chaque attribut à stocker, comme le montre la *figure 10* afin de pourvoir les récupérer et de les afficher dans leurs colonnes respectives. Ce choix de création d'un tableau vide pour chaque attribut plutôt qu'un seul tableau contenant l'ensemble des informations nous a permis de plus facilement récupérer et tester chaque attribut avant l'ajout dans le panier. Enfin nous précision le type d'action, qui est ici *add* pour l'ajout, afin de pouvoir récupérer cette variable dans le controller *action.php* qui se charge de la liaison entre chaque attribut et sa valeur présente dans la base de données comme le montre le code présent dans l'annexe 8.13.

```
$('#add_to_cart').click(function () {
    var product_id = [];
    var product_num = [];
    var product_type = [];
    var product_modele = [];
    var product_etat = [];
    var product_statut = [];
    var product_garantie = [];
    var product_lieu = [];
    var product_remarque = [];
    var product_inventaire = [];
    var product_immobilisation = [];
    var product_idcr = [];
    var product_nomcr = [];
    var action = "add";
```

FIGURE 10 – Code - Pré-ajout d'ordinateur dans le panier

Suite à cela, pour chaque ordinateur ajouté nous récupérons et stockons les valeurs dans les variables définies précédemment comme le montre la *figure 11*.

```
$('.select_product').each(function () {
    if ($(this).prop('checked') == true) {
        product_id.push($(this).data('product_id'));
        product_num.push($(this).data('product_num'));
        product_type.push($(this).data('product_type'));
        product_modele.push($(this).data('product_modele'));
        product_etat.push($(this).data('product_etat'));
        product_statut.push($(this).data('product_statut'));
        product_garantie.push($(this).data('product_garantie'));
        product_lieu.push($(this).data('product_lieu'));
        product_remarque.push($(this).data('product_remarque'));
        product_inventaire.push($(this).data('product_inventaire'));
        product_immobilisation.push($(this).data('product_immobilisation'));
        product_idcr.push($(this).data('product_idcr'));
        product_nomcr.push($(this).data('product_nomcr'));
    }
});
```

FIGURE 11 – Code - Enregistrement des ordinateurs sélectionnés

Une fois l'ensemble des données stockées dans leur variable respective, on vérifie que le numéro de série n'est pas vide. Puis, à l'aide d'Ajax, on appelle le controller *action.php* qui se charge de la création des variables de session pour l'ensemble des ordinateurs ajoutés au panier sous forme de tableau. Pour cela, nous devons passer ces données à la page en question (*action.php*) pour qu'elle puisse les intégrer à l'affichage. De ce fait, nous utilisons la méthode *POST* pour l'envoi des informations présentent dans le paramètre *data*. Dans le cas où chaque produit a été ajouté avec succès, on appelle la fonction *load_cart_data()*, qui rempli le panier de la session en cours dans le controller *fetch_cart.php*, et on affiche un message précisant le statut de l'ajout avant d'être redirigé vers la page de recherche à partir de laquelle on pourra choisir de nouveaux ordinateurs à sélectionner. Dans le cas où aucun ordinateur n'a été sélectionné on affiche un message indiquant qu'il est nécessaire d'ajouter un ordinateur avant de continuer.

Enfin, lorsque le panier n'est pas vide, la vue *panier.php* affiche le tableau de tous les ordinateurs ajoutés. Ce panier peut être vidé dans son ensemble ou nous pouvons choisir de supprimer un ordinateur individuellement. Ces explications sont liées à l'annexe 8.8 et l'algorithme présent dans la figure 12 représente un résumé des actions faites.

```

Algo ajout au panier :

Pour chaque ordinateur :
    Si numSerie != NULL :
        Creation tableau pour toutes les variables ;
        Affectation de tous les attributs présent dans la base de données à leur variable respective ;
        Récupération de tous les attributs présent dans les variables ;
        Remplissage du tableau avec les attributs récupérés ;
    fin si
fin pour
fin algo

```

FIGURE 12 – Algorithme pseudo-code - Ajout dans le panier

4.3.2 Insertion d'un ordinateur

Une autre partie importante de cette application est l'insertion d'un ou plusieurs ordinateurs. En effet, il a fallu traiter l'insertion par fichier CSV.

L'ajout d'un ordinateur via le formulaire HTML ou par importation d'un fichier CSV est assez similaire. C'est un exemple classique de l'architecture MVC. Dans la vue *ajouter_ordis.php* se trouve le code HTML de la page Ajouter des ordinateurs, avec deux formulaires : un formulaire HTML contenant plusieurs champs à remplir concernant la commande et l'ordinateur, et un autre permettant d'importer des ordinateurs et les informations sur leur commande par fichier CSV. Lorsque l'utilisateur a terminé de remplir ce formulaire ou d'importer le fichier CSV (l'un ou l'autre, pas les deux en même temps), il clique respectivement sur le bouton ajouter ou importer.

Cette action a pour effet d'envoyer les informations du formulaire HTML au contrôleur *ajouter_ordis.php* grâce à la méthode *POST* ou d'envoyer les informations du fichier CSV au contrôleur *ajouter_ordis_csv.php*. Ces données sont ensuite traitées par le contrôleur (qui vérifie qu'aucune information cruciale ne manque) avant de faire appel au modèle *model.php*, et plus particulièrement à ses fonctions. Ces données sont utilisées dans les fonctions *addOrdi*, qui permet d'ajouter un ordinateur, et *addCommande*, qui permet d'ajouter une commande, dont voici leurs signatures :

```
addOrdi($NumSerie, $Typee, $Modele, $NumImmobilisation, $Fabricant,
$NumInventaire, $Etat, $Statut, $detailSortieInventaire, $Remarque,
$NbrAnneGarantie, $dateDebutGestion, $IdCR, $NumCommande, $IdLieu)
```

```
addCommande($IdCommandeSIFAC, $Fournisseur, $DateDebutGarantie, $IdCR)
```

Si les données ont bien été rajoutées dans la base de données (via les fonctions du model qui contiennent des requêtes SQL d'ajout), le contrôleur renvoie vers la vue *ajouter_ordis.php* avec un message de succès dans l'URL comme ceci :

```
ajouter_ordis.php?status=ajout_reussi
```

La vue *ajouter_ordis.php* récupère ce message dans l'URL via la méthode GET et affiche un message de succès en HTML et stylisé en CSS sur la page d'ajout d'ordinateurs. Si les données n'ont pas pu être enregistrer dans la base de données pour une raison ou une autre, le contrôleur renvoie vers la vue avec un message d'erreur.

4.3.3 Édition des fiches inventaires

Une dernière partie intéressante de ce projet a été la gestion de l'historique des déplacements des ordinateurs, et en particulier le suivi de ceux-ci via les fiches inventaires. L'application web ne propose pour l'instant qu'un fichier CSV téléchargeable résumant les informations nécessaires à l'édition de ces fiches.

Pour accéder à ces fichiers CSV, l'utilisateur doit en premier lieu remplir le formulaire présent sur la vue *historique.php* : il choisit parmi les trois différents types de fiches (entrée, modification, sortie) et une période constituée de deux dates. Une fois le formulaire validé, les données sont envoyées à la vue *historique_result.php* qui affichera les ordinateurs résultant de la recherche effectuée par l'utilisateur. Cette dernière fait appel à une fonction d'affichage du modèle *model.php* qui est la suivante :

```
printHist($array);
```

Le tableau (\$array) que la fonction prend en paramètre est un tableau d'historique renvoyé par la fonction suivante, qui récupère les données envoyées par le formulaire :

```
getHist($dateDebut, $dateFin, $typeFiche);
```

Cette fonction renvoie le tableau résultant de la recherche effectuée en utilisant les données envoyées par la vue *historique.php*. Cette fonction contient une requête en MySQL qui fait appel à la table *Gere_Par_Hist*. Grâce à la

date de début, la date de fin et le type de fiche récupérés en paramètre, nous pouvons obtenir les lignes d'historique qui nous intéresse. L'attribut qui nous intéresse en particulier ici est l'attribut InfoFicheInventaire. Comme expliqué précédemment, c'est un attribut concaténé qui contient toutes les informations utiles à l'édition des fiches inventaire. Pour obtenir les informations concaténées une à une, il suffit de détacher les valeurs de cet attribut qui sont séparées par des points-virgules. Étant donné que l'ordre de cet attribut est bien défini, il suffit ensuite de placer ces valeurs séparées dans un tableau HTML. La fonction *printHist* permet donc de rendre ce tableau d'historique affichable en HTML.

Le fichier CSV téléchargeable mis à disposition sur la vue *historique_result.php* est rempli à partir de ce tableau. Les données de ce dernier sont récupérées et rentrées ligne par ligne dans un fichier CSV accessible en écriture dans le dossier view. Le fichier est alimenté à l'aide des fonctions suivantes :

```
getHistArray($array);  
  
historiqueCsv($array2, $typeFiche);
```

La première fonction, *getHistArray*, permet de transformer le tableau précédent (\$array) en un tableau (\$array2) correspondant aux normes CSV (les données sont séparées par des point-virgules). Enfin la fonction *historiqueCsv* prend ce tableau (\$array2) en paramètre et le type de fiche, puis rempli le fichier CSV correspondant au type de fiche : soit *entree.csv*, *modification.csv* ou *sortie.csv*. Les informations utiles à chacune des fiches (voir 3.2 Cas d'utilisation) étant différentes, nous avons séparé ces données. Le fichier est ainsi personnalisé selon le type de fiche.

4.4 Test/Validation

Tout au long de l'implémentation du site web nous avons mis en place différents tests avant toutes modifications de la base de données ou affichage sur l'interface afin de garantir la crédibilité des informations. Ces tests ont été faits avec des données réelles et nous avons fait en sorte de provoquer les erreurs afin de vérifier que l'application web se comporte correctement face à celles-ci.

Tout d'abord, les premiers tests ont été effectués via phpMyAdmin pour s'assurer du bon fonctionnement des requêtes et triggers comme le montre les quelques tests présent dans l'annexe 8.6 et 8.7. Après avoir créé quelques insertions dans la base de données nous avons essayé, entre autres, de changer de lieu et de centre responsable afin de vérifier la bonne mise à jour des informations entre les clés étrangères. Puis nous avons mis en place des triggers permettant de bloquer certains comportements et de mettre à jour certains attributs automatiquement. Parmi ceux-ci, nous avons implémenté un trigger se déclenchant avant l'insertion d'un ordinateur afin de vérifier que les données nécessaires sont remplies et qu'il n'y a pas de fautes au niveau de l'état et du statut. Si une erreur est détectée, le trigger la précise à l'utilisateur. Nous avons également testé l'ajout du tuple nécessaire dans la table *Gere-Par-Hist* après chaque ajout d'ordinateur afin de lui créer un nouvel historique grâce à un trigger en vérifiant que la nouvelle date de début de gestion est bien supérieure à la dernière date de fin de gestion par un centre responsable. De plus, étant donné que nous proposons une interface de connexion, nous avons également testé que les attributs contenant l'identifiant et le mot de passe ne soient pas nuls pour un compte créé. Nous avons également testé, toujours à l'aide d'un trigger, que l'email présent dans la table *Personne* respecte le format *text@umontpellier* et que les attributs comportant le nom et le prénom ne soient pas vides, dans le cas contraire un message d'erreur est affiché. Par la suite, nous avons testé les informations relatives au lieu. De ce fait nous vérifions, avant l'insertion de chaque lieu, que la capacité de la salle n'est pas nulle et que le numéro de salle, de campus et de niveau ne sont pas nuls également car la clé primaire d'un lieu est la concaténation de ces éléments. Enfin nous vérifions également que, lors d'un ajout d'un nouveau centre responsable, son nom n'est pas nul et lorsqu'on souhaite créer une nouvelle commande que ni son fournisseur ni la date de commande ne soient nuls. Pour l'ensemble des modifications liées à la base de données, ces tests sont également effectués lors de la mise à jour de celle-ci.

À présent, au niveau de l'interface web, lors de l'ajout d'ordinateur, nous avons testé au niveau de l'unicité de certains champs remplis par l'utilisateur. En effet, pour chaque ajout, nous vérifions que le numéro de série, d'immobilisation et d'inventaire sont uniques pour un ordinateur comme le montre l'annexe 8.9. Dans le cas où l'un des numéros existe déjà dans la base de données nous affichons un message d'erreur et nous interrompons

l'ajout sinon l'ajout se fait et nous affichons un message le précisant. En parallèle, si l'utilisateur souhaite effectuer un ajout via le fichier CSV proposé, nous devons également nous assurer de la crédibilité des données fournies. Pour cela, comme indiqué dans l'annexe 8.10, nous paramétrons les valeurs par défaut pour chaque nouvel ordinateur telles que son état, son statut et sa date de début de gestion qui est la date du jour ainsi que les attributs concernant les remarques et détails de sortie de l'inventaire UM qui sont NULL. Suite à cela nous nous assurons que la première ligne, contenant la trame, est lue sans y faire d'action puis nous vérifions que les colonnes contenant les informations obligatoires sont remplies. Si ce n'est pas le cas, nous le précisons avec un message d'erreur et ce message est également affiché dans le cas où le numéro de série, d'inventaire ou d'immobilisation est déjà existant dans la base de données. Si l'importation du fichier ne produit aucune erreur, alors l'ensemble des ordinateurs est ajouté.

De plus, pour une sortie d'inventaire FDS, une sortie de l'inventaire UM, un changement d'état ou une édition des caractéristiques qui nécessitent que le panier ne soit pas vide, nous testons dans un premier temps que le panier contient bien au minimum un ordinateur sinon un message d'erreur est affiché comme le montre l'annexe 8.11. Si l'ordinateur est hors de l'inventaire UM, il est impossible de le sortir de l'inventaire de la FDS. C'est pour cela que nous testons aussi ce cas et que nous affichons un message d'erreur si cela ce produit. De plus, si une erreur est rencontré au niveau de la base de données, nous précisons à l'utilisateur que sa modification a échoué et enfin, si aucun problème n'est rencontré, la modification se fait avec succès.

Il fut également important de tester la fonctionnalité de recherche de notre site web afin de nous assurer que les résultats renvoyés sont corrects. Pour cela, comme le montre l'annexe 8.12 nous vérifions dans un premier temps qu'au moins un champs du formulaire a été précisé et si ce n'est pas le cas nous affichons un message à l'utilisateur lui précisant que sa recherche est vide. Ensuite nous vérifions si les cases permettant la recherche par numéro d'immobilisation ou d'inventaire vides sont cochées afin d'intégrer ce critère aux requêtes. Puis nous testons individuellement si le numéro de série est renseigné car il est unique et/ou et enfin nous testons de manière groupée les champs relatifs au lieu, à l'état, à la garantie et au statut. Pour chacun de ces test nous vérifions en amont si le compteur de résultats obtenus vaut zéro afin de préciser à l'utilisateur que ça recherche n'a retourner aucun résultat sinon nous affichons le tableau des ordinateurs retournés dans la page de sélection.

Au niveau du panier, sachant que l'utilisateur peut sélectionner plusieurs fois le même ordinateur, nous avons dû mettre en place un test permettant d'éviter la répétition des ordinateurs identiques lors de l'affichage du panier. Afin d'éviter ceci, comme le montre l'annexe 8.13, après avoir attribuer la valeur de chaque colonne à sa variable respective nous vérifions si le numéro de série est déjà présent dans le panier et si c'est le cas nous ne faisons aucune action ce qui permet d'éviter son ajout. Sinon, s'il n'est pas déjà présent, nous ajoutons chaque nouvelles machines au panier.

Tout au long du projet, nous nous sommes assurés que l'ensemble des fonctionnalités développés entraînaient la bonne action au niveau de la base de données et du site web. De ce fait, nous avons vérifier que les requêtes et triggers concernant l'ajout d'ordinateur, la modification d'inventaire, le changement d'état, les différentes sortie d'inventaire ainsi que l'édition des caractéristiques étaient bien fonctionnels. De plus, nous avons vérifier et validé tout ce qui concerne la gestion du panier en mettant un point d'honneur à nous assurer que celui-ci concatène tous les ordinateurs sélectionné et ne commet pas de doublons. Enfin, les actions indépendantes à celui-ci tels que la consultation des ordinateurs en panne, en réparation ou à renouveler ainsi que les fiches inventaires ont été testées par comparaison du résultat de la base de données et celui sur l'interface web associée.

5 Résultats

5.1 Installation

Le site réalisé est disponible à l'adresse suivante :<https://parcinfofds.000webhostapp.com/>, par défaut le nom d'utilisateur est *admin* et le mot de passe aussi. Pour installer l'application web, il faut dans un premier temps une connexion internet et un navigateur internet. Il faut également avoir à sa disposition plusieurs programmes : Apache (serveur web), PHP version 5.4 minimum (plug-in pour Apache permettant de visualiser des pages web dynamiques) et MySQL (système de gestion de base de données) en version 14.14 au minimum. Ces programmes sont gratuits, et étant donné que cette combinaison (Apache+PHP+MySQL) est la plus courante sur les serveurs web, des paquets de cette combinaison ont été créés. Sous Windows, le paquet s'appelle WAMP, téléchargeable à l'adresse wampserver.aviaTechno.net/. Sous Mac OS X, le paquet se nomme MAMP qui peut être téléchargé ici

www.mamp.info/en/downloads/. Enfin, sous Linux, qui est le système sous lequel nous avons travaillé, il est plus courant d'installer ces paquets séparément, mais il est possible de télécharger la combinaison XAMPP disponible à l'adresse www.apachefriends.org/fr/index.html.environnement

Une fois l'environnement de travail préparé, vous pouvez récupérer l'archive de notre projet sur le GitLab de l'Université de Montpellier à l'adresse suivante gitlab.info-ufr.univ-montp2.fr/BD_Parc_Info_TER_S6/TER_S6 et le placer dans le dossier C :\wamp\www si vous êtes sous Windows, /opt/lampp/htdocs si vous êtes sous Linux. Sous Mac OS, vous devez placer le projet dans le dossier choisi lors de l'installation de MAMP (par exemple /Users/pseudo/Sites). Une fois le dossier transféré, il suffit de se rendre sur votre navigateur et de taper l'adresse localhost/nomDuDossier si vous êtes sous Windows ou Linux, ou localhost :8888/nomDuDossier si vous êtes sous Mac OS. nomDuDossier correspond au nom que vous avez attribué au dossier contenant le projet. Le site s'affiche enfin mais il faut également créer la base de données. Pour cela, il suffit de se rendre sur phpMyAdmin et d'importer le fichier qui se trouve dans le dossier que vous avez téléchargé, sous public/bdd. Si le nom de la base de données venait à changer, il faudrait modifier le fichier model.php se trouvant dans le dossier model. À la ligne 9 se trouve la commande permettant de se connecter à la base de données. En paramètre de la fonction PDO, il faudra changer ce qui se trouve après `dbname=...;` et placer le nom de la base de données à la place des trois petits points.

Dans le dossier include se trouve un fichier path.php. Ce fichier inclut les chemins des différents dossiers que nous avons réutilisé tout au long de notre projet. Si l'arborescence du projet venait à changer, il faudrait changer le fichier path.php afin de correspondre à la nouvelle arborescence.

Enfin, par défaut sur notre site web, les cookies sont nécessaires. En poursuivant la connexion, les cookies sont comptés comme accepté. Dans le cas où une erreur par rapport à ceux-ci est rencontré, il faut s'assurer que le paramétrage du navigateur est correct et les accepte. Pour ceci, comme le montre la figure 13, sous Firefox, dans l'onglet Vie privée et sécurité nous pouvons gérer les exceptions et en ajoutant localhost ou le nom du site web hébergé on autorise la création des cookies.

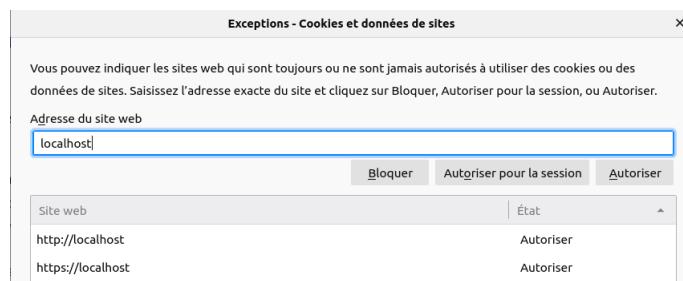


FIGURE 13 – Paramétrage des cookies sur Firefox

5.2 Manuel d'utilisation

Pour plus de lisibilité, les visuels de l'application web ont été placés dans l'annexe 8.14.

5.2.1 Actions indépendantes du panier

Se connecter Afin d'accéder au site web, il faut au préalable se connecter. Pour cela, il faut entrer un identifiant et un mot de passe présent dans la base de données. Seul l'administrateur peut ajouter un compte directement dans la base de données et, par défaut, nous avons mis la valeur "admin" comme identifiant et mot de passe. Une fois connecté nous pouvons accéder à toutes les actions du site.

Ajouter un ordinateur Pour ajouter un ordinateur dans la base de données, il faut compléter l'un des deux formulaires présents dans la page *Ajouter des ordinateurs*. Une fois sur cette page, nous avons le choix d'ajouter plusieurs d'ordinateurs via un fichier CSV téléchargé, dont la trame est téléchargeable. Il est également possible d'ajouter chaque ordinateur individuellement grâce au formulaire HTML où il est nécessaire de remplir au minimum tous les champs obligatoires marqués d'une astérisque rouge.

Effectuer un ajout dans le panier Avant d'ajouter des ordinateurs dans le panier, il faut d'abord se rendre sur la page *Rechercher* et renseigner les filtres choisis. Une fois le formulaire de recherche validé, nous arrivons sur la page *Sélection*, cette page affiche un tableau contenant tous les ordinateurs résultant de la recherche. A partir de là, nous pouvons sélectionner les ordinateurs qui nous intéressent avant de les ajouter au panier. Suite à la validation de la sélection les ordinateurs sont automatiquement ajoutés dans le panier de l'utilisateur.

Éditer une fiche inventaire Pour obtenir les informations nécessaires à l'édition d'une entrée inventaire, modification inventaire ou sortie inventaire, il faut se rendre sur la page *Fiches inventaires*, et choisir sur l'interface web laquelle de ces fiches nous souhaitons et pour quelle période. Suite à cela, nous obtenons un tableau contenant toutes les données nécessaires au remplissage de la fiche. Nous avons également la possibilité de télécharger le tableau récapitulatif dans un fichier CSV.

Afficher les ordinateurs à renouveler En se rendant sur la page *Ordinateurs à renouveler* nous obtenons l'ensemble des ordinateurs hors garantie à ce jour. Il est également possible de choisir une date à laquelle nous souhaitons obtenir tous les ordinateurs qui seront hors garantie à partir de celle-ci.

Afficher les ordinateurs en panne ou en réparation Les ordinateurs en panne et/ou en réparation sont consultables depuis la page *Ordinateurs en panne et en réparation*. En choisissant l'état souhaité nous obtenons le tableau de tous les ordinateurs possédant cet état ainsi que des informations sur leur localisation et leurs caractéristiques.

5.2.2 Actions dépendantes du panier

Effectuer une modification inventaire Après avoir ajouté les ordinateurs souhaités et présents dans l'inventaire FDS dans le panier, on se rend sur la page *Modification inventaire* présente dans le menu à gauche. Une fois sur cette page, on choisi dans les listes déroulantes le nouveau lieu et/ou le nouveau centre responsable auxquels les ordinateurs vont être affectés. Suite à la validation des changements, qui se font automatiquement dans la base de données, un message de confirmation est affiché à l'utilisateur.

Effectuer un changement d'état Avant de pouvoir effectuer un changement d'état sur un ou plusieurs ordinateurs, il faut d'abord sélectionner les ordinateurs voulus et les ajouter au panier. Puis, on se rend sur la page *Changement d'état* qui nous propose de choisir le nouvel état que l'on souhaite donner aux ordinateurs. En validant notre choix, les modifications seront effectuées sur la base de données et un message de confirmation des changements effectués sera affiché.

Effectuer une sortie FDS Afin d'effectuer une sortie de l'inventaire FDS, nous devons, au préalable, ajouter des ordinateurs présents dans l'inventaire FDS dans le panier. Suite à cela, nous devons nous rendre sur la page *Sortie FDS* où nous devons choisir à partir des listes déroulantes le nouveau lieu hors FDS ainsi que le nouveau centre responsable, hors FDS également. Une fois le changement validé un message indiquant le succès de la modification est affiché.

Effectuer une sortie UM Lorsqu'on veut sortir définitivement un ou des ordinateurs de l'inventaire UM, on doit se rendre sur la page *Sortie UM* avec les ordinateurs voulus ajoutés dans le panier en amont. Sur cette page nous pouvons détaillé au maximum le motif de cette sortie, son nouvel emplacement ainsi qu'un éventuel contact dans le champ texte associé.

Effectuer une édition des caractéristiques Une fois sur la page *Édition caractéristiques*, nous obtenons un formulaire permettant la modification, pour tous les ordinateurs du panier, des champs concernant le type, le modèle, le nombre d'années de garantie ainsi que les remarques. De plus, sous contrainte de n'avoir qu'un seul et unique ordinateur dans le panier, il est également possible de lui renseigner son numéro d'inventaire et/ou d'immobilisation.

6 Gestion de projet

6.1 Méthodes de travail

Dans le cadre du TER S6, nous devions réaliser une application web pour la gestion du parc informatique de la Faculté des Sciences en groupe de 5 personnes. Dès lors que le groupe a été formé, un travail de structuration de l'équipe ainsi que de définition du processus de gestion de projet ont été définis. De ce fait, nous avons choisi les méthodes Agiles afin de mener à bien notre projet.

Les méthodes Agiles reposent sur un cycle de développement itératif, incrémental et adaptatif, entrant dans la réalisation d'un projet. Elles privilégient le dialogue entre toutes les parties prenante du projet et garantissent une souplesse en cours de réalisation et une capacité à modifier un planning à tout moment afin de garantir des livraisons rapides et fonctionnelles. L'objectif de l'utilisation de ces méthodes est d'avoir une organisation solide ainsi qu'une flexibilité en ce qui concerne les besoins exprimés et un retour récurrent de nos tuteurs. Cette flexibilité permet de pouvoir gérer les changements à tout moment sans trop impacter le travail déjà fourni et en respectant au maximum le diagramme de Gantt établi. Par principe, ces méthodes reposent tout d'abord sur l'élaboration du product backlog, présent dans l'annexe 8.15, qui est le cahier des charges principal du produit. Suite à cela, ce cahier des charges est divisé, par ordre de priorité des tâches, en plusieurs sprint contenant les objectifs à réaliser avant la réunion suivante où nous faisons la rétrospective du travail effectuée avec l'ensemble de l'équipe et du client avant de planifier le prochain sprint. De ce fait, ces réunions régulières permettent un contact et un retour récurrent.

Par conséquent, nous avons mis en place des réunions hebdomadaires, en présentes ou sur Skype et Discord, avec nos tuteurs de projet et l'ensemble du groupe afin d'élaborer les objectifs du prochain sprint. Ce sprint, d'une durée d'une semaine, consiste à réaliser les tâches définies et réparties entre chaque membre du groupe afin de rendre un livrable lors de la réunion suivante. Le travail individuel et commun de chacun nous a permis d'avoir un rendu à présenter chaque semaine permettant l'avancée du projet. À la fin de chaque réunion nous redigions un compte rendu, comme dans l'annexe 8.16, de celle-ci, que nous envoyions, à nos tuteurs pour assurer la bonne compréhension des besoins exprimés ou la correction de ceux-ci si besoin. Ainsi chaque semaine nous réalisions une liste des tâches présente dans le product backlog qui contient la liste des besoins fonctionnels par ordre de priorité. Pour conclure, le travail collaboratif, l'adaptabilité au changement entraînant une coopération permanente entre les membres de l'équipe et les tuteurs de projet et la responsabilisation de chaque membre du groupe nous ont permis d'avancer à un rythme soutenable et constant tout en gardant en tête l'objectif à atteindre au niveau hebdomadaire comme général.

6.2 Diagramme de Gantt

Le diagramme de Gantt est un outil utilisé en ordonnancement et en gestion de projet et permettant de visualiser dans le temps les diverses phases composant un projet. Pour ce projet, le Gantt présent en annexe 8.17, a été découpé en cinq parties distinctes. Premièrement nous avions prévu un temps pour la compréhension du sujet ainsi que la définition des objectifs et suite à cela une phase de conception. Après la conception, nous avions prévu de démarrer le développement de l'application web et de la base de données puis d'effectuer les tests et validations tout cela en parallèle de la rédaction du rapport. À cela, nous avons ajouté une marge de risque ainsi que les dates clés, telles que la fin de projet, le rendu du rapport et de la présentation puis la date de la soutenance finale.

6.3 Répartitions des tâches

Tout au long du projet, nous avons mis en place les tâches à réaliser pour chaque sprint. Au départ, lors du début la phase de conception du projet, nous avons beaucoup travaillé en commun afin de nous assurer que chacun ait la même vision du travail à faire et du but à atteindre. Une fois ce stade atteint, où la vision du projet et de sa réalisation était commune, nous nous sommes répartis les tâches de façon homogène. Cette répartition s'est basée sur les compétences de chacun afin d'optimiser le travail de l'équipe. En effet, certains membre que l'équipe avaient un savoir-faire plus développé en programmation web qu'en base de données et vice-versa.

Cependant, même si nous avons cherché à maximiser les aptitudes de chacun, nous avons également fait en sorte que chaque membre de l'équipe touche à tous les aspects de ce projet. Ainsi, nous avons allié optimisation des compétences de chacun et diversification des tâches afin d'obtenir un résultat respectant les objectifs fixés tout en sortant de notre zone de confort et en élargissant nos connaissances à tous les niveaux.

6.4 Outils de productivité

6.4.1 Outils de communication

Au sein d'un projet la communication est primordial. Dans le but de partager nos interrogations, nos avancées et recueillir les différents avis avant l'implémentation d'une tâche, nous nous sommes servis de différents moyens de communication. Parmi ceux-ci, la communication plus naturelle reste la communication formelle lors des réunions en groupe dans lesquelles nous énumérions nos idées et partagions nos points de vues sur les différents travaux à réaliser. En parallèle, nous avons également utilisé des moyens de communication à distance comme WhatsApp afin de pouvoir échanger des informations en continu avec notre équipe ou l'envoi de mail aux professeurs en cas de questions. Enfin, étant donné la situation sanitaire actuelle, nos réunions hebdomadaires avec nos tuteurs se sont déroulées sur Discord et Skype.

6.4.2 Outils de collaboration

Tout d'abord nous avons commencé à stocker l'ensemble de notre production sur Google Drive, partagé avec les membres de l'équipe ainsi qu'avec nos tuteurs de projet. Par la suite nous avons basculer nos différents travaux, dit finaux, vers Gitlab, tout en gardant Google Drive comme dépôt.

De plus, étant donné l'utilisation des méthodes agiles, l'outil Trello nous a permis de dynamiser le suivi des tâches, l'assignation des responsables pour chacune d'entre elles et de rajouter des commentaires ou questions ainsi que des explications de ce qui a été finalement réalisé. À chaque fin de réunion hebdomadaire, un nouveau tableau Trello, appelé sprint, a été créé avec les cartes dédiées à chaque tâche pour le sprint à venir. Ces différentes cartes avaient comme titre : Tâches à faire / Urgent / En cours / Questions / Terminé / et nous y déplaçons chaque tâche selon son état d'avancement comme le montre l'exemple présent dans l'annexe 8.18.

7 Conclusion

En raison de la taille de ce projet, nous avons naturellement été confronté à de nombreuses difficultés car un travail majeur de conception était nécessaire en amont. De plus, une grande quantité de documents mis à disposition fut à trier afin d'y faire ressortir les informations importantes et nécessaires aux développement de la base de données et de l'application web. Dans le temps imparti certaines fonctionnalités ont pu être terminées dans leur globalité et d'autres non.

Premièrement nous avons conçu, implémenté et testé l'intégration des ordinateurs dans la base de données de façon manuelle, simple et rapide par un formulaire ou via un fichier CSV. Mais également la visualisation, la sélection d'ordinateur selon différents filtres choisis et leur enregistrement dans le panier ainsi que la modification de leurs emplacements et caractéristiques. De plus, l'édition des fiches d'entrée, de modification et de sortie d'inventaire peuvent être générée depuis l'interface proposée. Nous avons également conçu, implémenté et testé le système du panier permettant à l'utilisateur de stocker, au fur et à mesure, les sélections faites après une recherche, ainsi que les actions dépendantes de celui-ci qu'il pourra effectuer telles que la modification inventaire, le changement d'état, une sortie de l'inventaire FdS ou UM ou encore une édition des caractéristiques. Il pourra également visualiser les ordinateurs dont la date de garantie se termine avant une date choisie afin de procéder à leur renouvellement, ainsi que l'ensemble de ordinateurs. Cependant, la possibilité d'éditer les caractéristiques pour un ordinateur en particulier parmi plusieurs présentes dans le panier a seulement été conçue mais pas implémentée. De plus, par manque de temps, nous n'avons pas pu terminer l'implémentation de la fonctionnalité permettant la visualisation de la page de sélection obtenue sous le formulaire de recherche.

Deuxièmement, au niveau de la base de données, nous avons conçu son modèle entité-association, son modèle logique et implémenté la base de données. Ainsi nous pouvons conserver les données sur les ordinateurs de manière pérenne et permettre d'y accéder facilement dans le but de faire des ajouts de machine, des modifications ainsi que des mises à jour aux niveau des différentes données.

Par ailleurs, ce projet nous a aussi permis de réaliser à quel point toutes les facettes menant à la réalisation du produit final étaient importantes. Bien que la phase de conception du cahier des charges fut longue, son importance reste indéniable car sans elle de nombreuses questions seraient restées sans réponses entraînant une évidente perte de temps. De plus la communication au sein de l'équipe fut l'une des qualités nous ayant permis de gérer la planification mais également de renforcer la cohésion du groupe et son autonomie afin de rendre un projet répondant au maximum au besoins du client tout en respectant les différentes contraintes imposées.

Il est également indispensable de mettre en évidence les aspects positifs de l'accomplissement de ce projet. En effet celui-ci nous a non seulement fournit une approche concrète du futur métier que nous voudrions exercer mais il a également été une opportunité de mettre en pratique les différentes compétences théoriques et pratiques acquises lors de notre parcours. Il est clair que la gestion de projet, le travail en équipe, l'autonomie et le respect des délais sont des éléments essentiels dans le monde professionnel. De même, le travail fourni au niveau de la recherche, de synthèse de l'information et de résolution de problèmes nous a permis de consolider nos connaissances et stimuler notre créativité.

La communication entre le groupe et les tuteurs de ce projet a été un élément favorable au bon déroulement de ce dernier. Ce qui nous mène aujourd'hui à un résultat plutôt satisfaisant. En effet, il répond en grande partie aux exigences principales du cahier des charges mais il peut également être amélioré. Il est clair que ce projet a apporté à chacun d'entre nous une approche plus technique et professionnel autant au niveau conceptuel qu'au niveau de la gestion de projet de la programmation.

Pour conclure, durant ce projet, nous avons pu nous mettre dans la peau d'un développeur mais également dans celle d'un concepteur, face à un client. Cette approche, inédite pour certain d'entre nous, nous a permis d'avoir une idée de la vie professionnelle. Ainsi, notre expérience et nos compétences sont renforcées. Les acquis obtenus et les erreurs commises font que nous sortons grandis de ce projet.

7.1 Perspectives

Au cours de ce projet, notre groupe a fourni un travail conséquent de conception et d'implémentation. Outre la maturité et les différentes qualités acquises et appliquées lors de ces 15 semaines, notre projet peut être améliorable. En effet, par manque de temps, nous n'avons pas pu pousser le développement de l'application web à son maximum afin d'améliorer l'expérience de l'utilisateur final. Nous aurions pu implémenter la possibilité de télécharger l'ensemble du panier ou la recherche dans un fichier CSV. Nous aurions également pu améliorer l'édition des caractéristiques des ordinateurs en permettant la modification individuelle de chacun des ordinateurs présents afin de modifier chacun de leurs attributs séparément. De plus, toujours dans un objectif d'amélioration de l'expérience de l'utilisateur, nous aurions pu mettre en place une interface administrateur afin de ne pas avoir à effectuer des changements directement via la base de données, comme c'est le cas pour l'ajout de lieux ou de centres responsables. Enfin, à plus haut niveau, il aurait été intéressant de mettre en place un système remplissant automatiquement les différentes fiches inventaires pour chaque ordinateur avec les informations présentes dans les fichiers CSV téléchargeables, et ainsi mettre à disposition des fichiers PDF à télécharger pour chaque ordinateur.

8 Annexes

8.1 Dictionnaire de données

Dictionnaire de données	Code	Nom Rubrique	Type	Longeur	Nature	Règle d'intégrité
NumSerie	Numéro de série d'un ordinateur	AN	30	Élémentaire	UNIQUE , NOT NULL	
Type	Type d'un ordinateur	AN	100	Élémentaire		
Modele	Modèle de l'ordinateur	AN	50	Élémentaire		
Etat	l'état d'un ordinateur	Alphabétique	50	Élémentaire	NOT NULL : - fonctionnel -en panne -en réparation	
NumImmobilisation	Numéro d'immobilisation d'un ordinateur	AN	50	Élémentaire		
Remarque	Champs destiné aux remarques pour un ordinateur (manque de souris ou clavier...)	AN	1000	Élémentaire		
NumInventaire	Numéro d'inventaire de l'ordinateur	AN	80	Élémentaire		
Fabricant	Nom du fabricant d'un ordinateur	AN	50	Élémentaire		
Statut	Statut d'un ordinateur	Alphabétique	100	Élémentaire	NOT NULL : -Dans inventaire FDS -Hors FDS dans inventaire UM -Hors inventaire UM	
detailSortieInventaire	Détail concernant la sortie de l'inventaire de l'ordinateur	AN	1000	Élémentaire		
NbrAnneeGarantie	Champs donnant le nombre d'année de garantie d'un ordinateur	Numérique	2	Élémentaire	NOT NULL	
IdLieu	identifiant du lieu	AN	50	Élémentaire	UNIQUE NOT NULL	

Continued on next page

TABLE 1 – *Continued from previous page*

Code	Nom Rubrique	Type	Longeur	Nature	Règle d'intégrité
NumSalle	le numéro de la salle	AN	10	Élémentaire	NOT NULL
NomSalle	le nom de la salle	AN	50	Élémentaire	
NumNiveau	Le niveau de l'étage	AN	2	Élémentaire	NOT NULL
CapaciteTheorique	Le nombre d'ordinateur qu'il peut y avoir dans une salle	Numérique	3	Élémentaire	NOT NULL
NumBatiment	le numéro du bâtiment	Numérique	2	Élémentaire	NOT NULL
NomBatiment	le nom du bâtiment	Alphabétique	50	Élémentaire	
NumCampus	le numéro du campus auquel appartient le bâtiment	Numérique	50	Élémentaire	NOT NULL
NomCampus	le numéro du campus auquel appartient le bâtiment	Alphabétique	50	Élémentaire	
NumCom	Numéro d'une commande	Numérique	10	Élémentaire	AUTO-INCREMENTE
IdCommandeSIFAC	Identifiant d'une commande	AN	10	Élémentaire	UNIQUE
DateDebutGarantie	Date debut de garantie d'un ordinateur	Date	10	Élémentaire	aaaa-mm-jj
Fournisseur	Nom du fournisseur	Alphabétique	50	Elementaire	
IdCR	Identifiant d'un centre responsable	Numérique	50	Élémentaire	AUTOINCREMENT
NomCR	Nom du centre responsable	AN	50	Élémentaire	NOT NULL
CodeCR	Code du centre responsable	Numérique	4	Élémentaire	NOT NULL
typeStructure	Le type de structure d'un centre responsable	Alphabétique	4	Élémentaire	NOT NULL
IdPersonne	l'identifiant d'une personne	Numérique	10	Élémentaire	AUTOINCREMENT

Continued on next page

TABLE 1 – *Continued from previous page*

Code	Nom Rubrique	Type	Longeur	Nature	Règle d'intégrité
Nom	Nom d'une Personne	Alphabétique	50	Élémentaire	NOT NULL
Prenom	Prénom d'une Personne	Alphabétique	50	Élémentaire	NOT NULL
NumTelephone	n° de téléphone d'une personne	Numérique	10	Élémentaire	
email	l'email d'une personne	AN	10	Élémentaire	UNIQUE NOT NULL : text@umontpellier.fr
Qualite	Le rôle d'une Personne dans l'UM	Alphabétique	80	Élémentaire	NOT NULL
login	Login d'un compte	AN	50	Élémentaire	UNIQUE , NOT NULL
Password	mot de passe d'un compte	AN	50	Élémentaire	UNIQUE ,NOT NULL
dateDebutGestion	date et heure exact de début de gestion d'un ordinateur par un centre responsable	DateTime	10	Élémentaire	NOT NULL : aaaa-mm-jj HH :MM :SS
dateFinGestion	date et heure exact de fin de gestion d'un ordinateur par un centre responsable	DateTime	10	Élémentaire	NOT NULL : aaaa-mm-jj HH :MM :SS
InfoFicheInventaire	les informations concernant la fiche inventaire de modification pour un ordinateur	AN	1000	Élémentaire	NOT NULL
compteurModification	le nombre de modifications d'inventaire pour un ordinateur	Numérique	100	Élémentaire	AUTO-INCREmente
typefiche	type d'inventaire d'un ordinateur	Alphabétique	100	Élémentaire	NOT NULL : -entrée -sortie -modification
fonction	la fonction de la personne qui gère le lieu	Alphabétique	50	Élémentaire	

8.2 Maquette - Page recherche

Accueil

https://www.gestion-fds.com

UM

Rechercher Panier [0]

RECHERCHER

n°Série

n°Bat

Code lieu

Sous Garantie

En panne

Fonctionnel

En réparation

Dans inventaire UM

Hors inventaire UM dans FDS

Hors UM

à quelle date

Champs libre

Nouvel ordinateur sans étiquette

Rechercher

[ne remplir que la ou les parties nécessaires à votre recherche]

8.3 Maquette - Page sélection

Accueil

Ajouter des ordinateurs

Fiches inventaire

Ordinateurs à renouveler

Ordinateurs en panne et réparation

Modification inventaire

Changement d'état

Sortie FDS

Sortie UM

Édition caractéristique

SELECTION

Tout	Numéro de série	Type	Model	Etat	Statut	Durée garantie	Contact	+ email, téléphone
<input type="checkbox"/>	SFVFZR183K15J	Ordinateur fixe	OptiPlex 780	Fonctionnel	Dans inventaire FDS	5	Paul DUPONT	
<input type="checkbox"/>	KCPR45KFRJ34Y	Ordinateur fixe	OptiPlex 780	Fonctionnel	Dans inventaire FDS	5	Pierre MARTIN	
<input checked="" type="checkbox"/>	HRIFE47K23KDT	Ordinateur fixe	OptiPlex 780	Fonctionnel	Dans inventaire FDS	5	Paul DUPONT	
<input type="checkbox"/>	GRJI453DKD23R	Ordinateur fixe	OptiPlex 780	Fonctionnel	Dans inventaire FDS	5	Pierre MARTIN	
<input type="checkbox"/>	ADF345JL345FG	Ordinateur fixe	OptiPlex 780	Fonctionnel	Dans inventaire FDS	5	Pierre MARTIN	
<input checked="" type="checkbox"/>	RER347K894JIZ	Ordinateur fixe	OptiPlex 780	Fonctionnel	Dans inventaire FDS	5	Paul DUPONT	
<input type="checkbox"/>	PDLM34L5KOR4	Ordinateur fixe	OptiPlex 780	Fonctionnel	Dans inventaire FDS	5		

Ajouter au panier  **Annuler**

8.4 Maquette - Page panier

Accueil

Ajouter des ordinateurs

Fiches inventaire

Ordinateurs à renouveler

Ordinateurs en panne et réparation

Modification inventaire

Changement d'état

Sortie FDS

Sortie UM

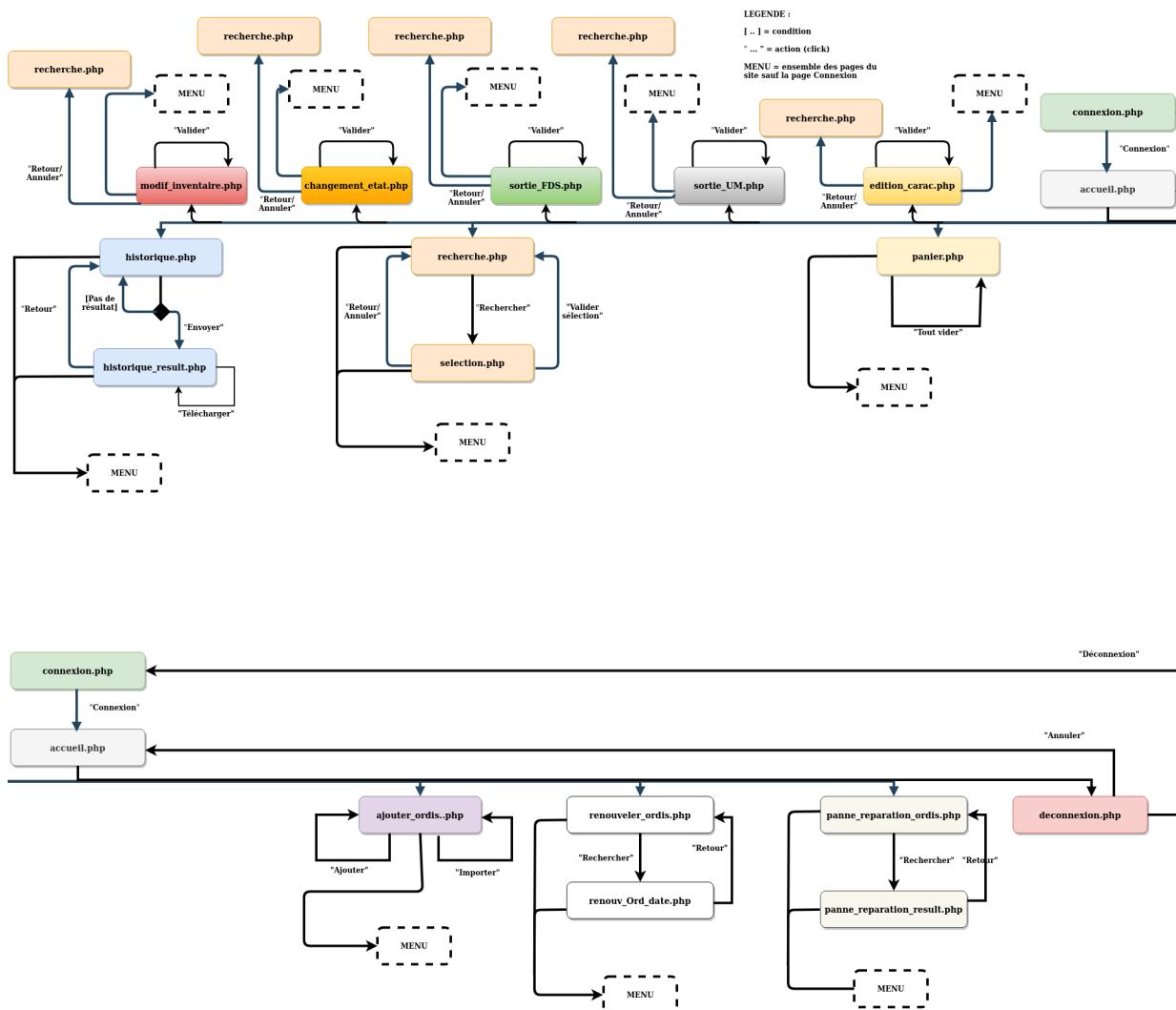
Édition caractéristique

PANIER

Lieu	Numéro de série	Type	Model	Etat	Statut	Durée garantie	Contact	+ num Etiquette, num Immo, type structure
01/36/04/46.0	HRIFE47K23KDT	Ordinateur fixe	OptiPlex 780	Fonctionnel	Dans inventaire FDS	5	Paul DUPONT	
01/36/04/46.0	RER347K894JIZ	Ordinateur fixe	OptiPlex 780	Fonctionnel	Dans inventaire FDS	5	Paul DUPONT	

Tout supprimer **Annuler**

8.5 Diagramme de navigation



8.6 Triggers

Trigger before_insert_Ordinateur : Ce trigger se déclenche avant chaque insertion dans la table *Ordinateur* et vérifiera que :

1. l'état est soit "En panne", "Fonctionnel" ou "En réparation".
2. le statut est soit "Dans inventaire FDS", "Hors FDS dans inventaire UM" ou bien "Hors inventaire UM"
3. le modèle, le type, le fabricant, l'état, le statut et la dateDebutGestion ne sont pas NULL.

Test du trigger :

Si on insère un ordinateur tels que :

```
INSERT INTO Ordinateur(NumSerie, Typee, Modele, NumImmobilisation, Fabricant, NumInventaire, Etat, Statut, detailSortieInventaire, Remarque, NbrAnneGarantie, dateDebutGestion, IdCR, NumCommande, IdLieu) VALUES ('1FRMH5K', 'Workstation', 'OptiPlex 790', '57M4HG12', 'Dell', 'UM000001', 'Fonctionnel', 'Dans inventaire Paris8', NULL, NULL, 2, '2016-02-12 00:00:00', 1, 1, '01/36/02/36.204');
```

on aura le message d'erreur suivant :

MySQL a répondu :

```
#1644 - ERREUR : Le statut doit être soit "Dans inventaire FDS" ou "Hors FDS dans inventaire UM" ou bien "Hors inventaire UM"
```

Trigger after_insert_Ordinateur : Ce trigger se déclenche après chaque insertion dans la table *Ordinateur* pour :

1. Insérer dans la table *Gere_Par_Hist* un tuple représentant l'entrée dans l'inventaire de l'ordinateur qui vient d'être insérer.
2. Mettre dans l'attribut *InfoFicheInventaire* les informations nécessaires pour la fiche entrée inventaire (récupérer à l'aide de requêtes).
3. Mettre dans l'attribut *typefiche* "Entrée".

Si on modifie le statut de l'insertion précédente tels que :

1 ligne insérée. (traitement en 0.0021 seconde(s).)

```
INSERT INTO Ordinateur(NumSerie, Typee, Modele, NumImmobilisation, Fabricant, NumInventaire, Etat, Statut, detailSortieInventaire, Remarque, NbrAnneGarantie, dateDebutGestion, IdCR, NumCommande, IdLieu) VALUES ('1FRMH5K', 'Workstation', 'OptiPlex 790', '57M4HG12', 'Dell', 'UM000001', 'Fonctionnel', 'Dans inventaire FDS', NULL, NULL, 2, '2016-02-12 00:00:00', 1, 1, '01/36/02/36.204')
```

Ce trigger insérera dans la table *Gere_Par_Hist* un tuple représentant l'entrée de cet ordinateur dans l'inventaire.

IdCR	NumSerie	dateDebutGestion	dateFinGestion	InfoFicheInventaire	compteurModification	typefiche
1	1FRMH5K	1900-01-01 00:00:00	2020-05-03 20:43:09	Gall-Borut;Pascale;777072536;Pascale.Gall-Borut@um...	1	Entrée

Trigger before_update_Ordinateur : Ce trigger se déclenche avant chaque modification du centre responsable et/ou du lieu d'un ordinateur dans la table *Ordinateur* tels que :

1. Il insérera dans la table Gere_par_hist le numéro de série de cet ordinateur, la date de début de gestion récupérée dans la table Ordinateur, la date du jour comme date de fin de gestion , incrémentera le compteur de modification, initialisera typefiche à "Modification" et mettra dans l'attribut infoFicheInventaire les informations concernant la fiche d'une modification d'inventaire. Dans le cas où la valeur de l'attribut Statut est remplacé par "Hors inventaire UM", le trigger sera déclenché pour faire la même chose sauf que l'attribut typefiche sera initialisé à "Sortie" et mettra dans l'attribut infoFicheInventaire les informations concernant la fiche d'une Sortie d'inventaire.
2. Modifiera la dateDebutGestion par la date du jour de la modification dans la table Ordinateur.
3. Vérifie que l'état est soit "En panne", "Fonctionnel" ou "En réparation".
4. Vérifie que le statut est soit "Dans inventaire FDS", "Hors FDS dans inventaire UM" ou bien "Hors inventaire UM".
5. Vérifie que le modèle, le type, le fabricant, l'état, le statut et la dateDebutGestion ne sont pas NULL.

Si on change le lieu de l'ordinateur précédent tels que :

```
✓ 1 ligne affectée. (traitement en 0.0022 seconde(s).)
UPDATE Ordinateur SET IdLieu='01/6/00/6.02' WHERE NumSerie='1FRMH5K'
```

Ce trigger insérera dans la table Gere_Par_Hist un tuple représentant la modification d'inventaire de cet ordinateur.

IdCR	NumSerie	dateDebutGestion	dateFinGestion	InfoFicheInventaire	compteurModification	typefiche
1	1FRMH5K	1900-01-01 00:00:00	2020-05-03 21:01:13	Gall-Borut;Pascale;777072536;Pascale.Gall-Borut@um...	1	Entrée
1	1FRMH5K	2016-02-12 00:00:00	2020-05-03 21:03:55OptiPlex 790;Workstation;Dell;;1FRMH5K;01;36...	2	Modification

Par contre en changeant le statut tels que :

```
✓ 1 ligne affectée. (traitement en 0.0025 seconde(s).)
UPDATE Ordinateur SET Statut='Hors inventaire UM' WHERE NumSerie='1FRMH5K'
```

Ce trigger insérera dans la table Gere_Par_Hist un tuple représentant la sortie d'inventaire de cet ordinateur.

IdCR	NumSerie	dateDebutGestion	dateFinGestion	InfoFicheInventaire	compteurModification	typefiche
1	1FRMH5K	1900-01-01 00:00:00	2020-05-03 21:01:13	Gall-Borut;Pascale;777072536;Pascale.Gall-Borut@um...	1	Entrée
1	1FRMH5K	2016-02-12 00:00:00	2020-05-03 21:03:55	::::OptiPlex 790;Workstation;Dell;;1FRMH5K;01;36...	2	Modification
1	1FRMH5K	2020-05-03 21:03:55	2020-05-03 21:35:35	::::OptiPlex 790;Workstation;Dell;;1FRMH5K;01;6;...	3	Sortie

Trigger before_insert_Personne : Ce trigger se déclenche avant chaque insertion dans la table *Personne* pour :

1. Vérifier que le mail est valide (i.e sous la forme : text@umontpellier.fr).
2. Le nom et le prénom ne sont pas NULL.

On a aussi créé un autre trigger qui sera déclenché avant toute modification vérifiant les mêmes conditions que le trigger *before_insert_Personne* et qui a comme signature : CREATE TRIGGER **before_update_Personne** BEFORE UPDATE ON *personne* (...).

Trigger before_insert_Compte : Ce trigger se déclenche avant chaque insertion dans la table *Compte* pour :

1. Vérifier que le mot de passe et le login ne sont pas NULL.

On a aussi créé un autre trigger qui sera déclenché avant toute modification vérifiant les mêmes conditions que le trigger *before_insert_Compte* et qui a comme signature : CREATE TRIGGER **before_update_Compte** BEFORE UPDATE ON *Compte* (...).

Trigger before_insert_Lieu : Ce trigger se déclenche avant chaque insertion dans la table *Lieu* pour :

1. Vérifier que la capacite d'une salle n'est pas negative ou NULL.
2. Vérifier que NumSalle, NumBatiment, NumCampus et NumNiveau ne sont pas NULL, car la clé primaire est la concaténation de ces quatre attributs.

On a aussi créé un autre trigger qui sera déclenché avant toute modification vérifiant les mêmes conditions que le trigger *before_insert_Lieu* et qui a comme signature : CREATE TRIGGER **before_update_Lieu** BEFORE UPDATE ON *Lieu* (...).

Trigger before_insert_Gere_Par_Hist : Ce trigger se déclenche avant chaque insertion dans la table *Gere_Par_Hist* pour :

1. Vérifier que la date est valide (i.e dateDebutGestion > dateFinGestion).
2. Vérifier que le typefiche n'est pas NULL.

-
3. Vérifier le typefiche est soit "Entrée", "Modification" ou bien "Sortie".

Trigger before_insert_CentreResponsable : Ce trigger se déclenche avant chaque insertion dans la table *CentreResponsable* pour :

1. Vérifier que le NomCR n'est pas NULL.

On a aussi créé un autre trigger qui sera déclenché avant toute modification vérifiant les mêmes conditions que le trigger before_insert_CentreResponsable et qui a comme signature :

`CREATE TRIGGER before_update_CentreResponsable BEFORE UPDATE ON CentreResponsable (...).`

Trigger before_insert_Commande : Ce trigger se déclenche avant chaque insertion dans la table *Commande* pour :

1. Vérifier que le fournisseur et la date début de gestion ne sont pas NULL.

On a aussi créé un autre trigger qui sera déclenché avant toute modification vérifiant les mêmes conditions que le trigger before_insert_Commande et qui a comme signature :

`CREATE TRIGGER before_update_Commande BEFORE UPDATE ON Commande (...).`

8.7 Requêtes

Exemple requête 1 :

Dans la table Ordinateur :

NumSerie	NumCommande	NbrAnneGarantie	...ect
8CRMH5J	1	2	
FCRMH5J	2	3	
4FRMH5J	3	2	

Dans la table Commande :

NumCommande	DateDebutGarantie	...ect
1	2016-01-07	
2	2017-04-24	
3	2019-09-08	

Résultat de la requête :

NumSerie
8CRMH5J
FCRMH5J

Exemple requête 2 :

Dans la table Ordinateur :

NumSerie	NumCommande	NbrAnneGarantie	...ect
8CRMH5J	1	2	
FCRMH5J	2	3	
4FRMH5J	3	2	
1FRMH5K	1	2	
3DRMH5J	4	5	
87DF5RY1	3	4	
11AZ45ER	3	4	

Dans la table Commande :

NumCommande	DateDebutGarantie	...ect
1	2016-01-07	
2	2017-04-24	
3	2019-09-08	
4	2017-12-04	

Résultat de la requête :

NumSerie
8CRMH5J
11AZ45ER
4FRMH5J
87DF5RY1
3DRMH5J

Exemple requête 3 :

Dans la table Ordinateur :

NumSerie	Etat	...ect
1FRMH5K	Fonctionnel	
8CRMH5J	En panne	
4FRMH5J	En panne	
3DRMH5J	Fonctionnel	
87DF5RY1	En reparation	
FCRMH5J	En panne	
11AZ45ER	Fonctionnel	

Résultat de la requête :

NumSerie
4FRMH5J
87DF5RY1
8CRMH5J
FCRMH5J

8.8 Code - Ajout dans le panier

```
if (product_id.length > 0) {
    $.ajax({
        url: "../controller/action.php",
        method: "POST",
        data: {
            product_id: product_id, product_num: product_num, product_type: product_type,
            product_modele: product_modele, product_etat: product_etat,
            product_statut: product_statut, product_garantie: product_garantie,
            product_lieu: product_lieu, product_remarque: product_remarque,
            product_inventaire: product_inventaire, product_immobilisation: product_immobilisation,
            product_idcr: product_idcr, product_nomcr: product_nomcr,
            action: action
        },
        success: function (data) {
            $('.select_product').each(function () {
                if ($(this).prop('checked') == true) {
                    $(this).attr('checked', false);
                    var temp_product_id = $(this).data('product_id');
                    $('#product_' + temp_product_id).css('background-color', 'transparent');
                    $('#product_' + temp_product_id).css('border-color', '#ccc');
                }
            });

            load_cart_data();
            alert("L'ajout a été effectué avec succès ! ");
            window.location.href = "recherche.php";
        }
    });
}
else {
    alert('Veuillez sélectionner au moins un ordinateur');
}

});
```

8.9 Test - Ajout ordinateur manuellement

```
if(isset($_POST['ajouter'])) {
    /* VÉRIFICATION DE L'UNICITÉ DU NUMSERIE, NUMINVENTAIRE ET NUMIMMOBILISATION */
    $nbr_serie = sameNumSerie($_POST['numserie']);

    if($nbr_serie != 0){
        header('Location: ' . $view_path . 'ajouter_ordis.php?status=serie_existe');
    }

    $numimmo = $_POST['numimmobilisation'];
    if($_POST['numimmobilisation'] == ''){
        $numimmo = NULL;
    }else {
        $nbr_immo = sameNumImmo($_POST['numimmobilisation']);
        if($nbr_immo != 0){
            header('Location: ' . $view_path . 'ajouter_ordis.php?status=immo_existe');
        }
    }

    $numinv = $_POST['numinventaire'];
    if($_POST['numinventaire'] == ''){
        $numinv = NULL;
    }else {
        $nbr_inv = sameNumInv($_POST['numinventaire']);
        if($nbr_inv != 0){
            header('Location: ' . $view_path . 'ajouter_ordis.php?status=inv_existe');
        }
    }
/* FIN VÉRIFICATION */
```

8.10 Test - Ajout ordinateur CSV

```

if (isset($_POST["import"])) {

    $fileName = $_FILES["file"]["tmp_name"];
    $Etat = 'Fonctionnel';
    $Statut = 'Dans inventaire FDS';
    $detailSortieInventaireUM = NULL;
    $Remarque = NULL;
    $dateDebutGestion = date('Y-m-j');

    $cpt=1;
    $verif='not ok';
    if ($_FILES["file"]["size"] > 0) {
        $file = fopen($fileName, "r");
        fgetcsv($file, 10000, ","); //lit la premiere ligne (pour rien) et permet de sauter la premiere ligne du fichier
        //contenant la trame
        while ((($column = fgetcsv($file, 10000, ",")) != FALSE) {
            /* VERIFICATION QU'AUCUNE INFORMATION NE MANQUE + LE LIEU EST BON */
            if($column[0]=='' || $column[1]=='' || $column[2]==''
            || $column[4]=='' || $column[6]=='' || $column[7]==''
            || $column[8]=='' || $column[10]=='' || $column[11]=='' || $column[12]==''){
                header('Location: ' . $view_path . 'ajouter_ordis.php?status=champ_obligatoire&cpt='.$cpt);
                exit();
            }
            if(sameLieu($column[8]) == 0){
                header('Location: ' . $view_path . 'ajouter_ordis.php?status=wrong_place&cpt='.$cpt);
                exit();
            }
            /* FIN VERIFICATION */
            /* VERIFICATION DE L'UNICITÉ DU NUMSERIE, NUMINVENTAIRE ET NUMIMMOBILISATION */

            $nbr_serie = sameNumSerie($column[0]);

            if($nbr_serie != 0){
                header('Location: ' . $view_path . 'ajouter_ordis.php?status=serie_existe&cpt='.$cpt);
            }

            $numimmo = $column[3];
            if($column[3] == ''){
                $numimmo = NULL;
            }else {
                $nbr_immo = sameNumImmo($column[3]);
                if($nbr_immo != 0){
                    header('Location: ' . $view_path . 'ajouter_ordis.php?status=immo_existe&cpt='.$cpt);
                }
            }

            $numinv = $column[5];
            if($column[5] == ''){
                $numinv = NULL;
            }else {
                $nbr_inv = sameNumInv($column[5]);
                if($nbr_inv != 0){
                    header('Location: ' . $view_path . 'ajouter_ordis.php?status=inv_existe&cpt='.$cpt);
                }
            }
            /* FIN VERIFICATION */
        }
    }
}

```

8.11 Test - Sortie FDS

```
<!-- MESSAGES D'ALERTE -->
<?php if (isset($_GET['status']) && $_GET['status'] == "panierVide" && $_SESSION['totalPanier']== 0): ?>
|   <div class="alert">
|       <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
|       Le panier est vide !
|   </div>
<?php endif ?>

<?php if (isset($_GET['status']) && $_GET['status'] == "reussi" && $_SESSION['totalPanier'] != 0 ) : ?>
|   <div class="alert">
|       <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
|       Modification effectuée !
|   </div>
<?php endif ?>
<?php if (isset($_GET['status']) && $_GET['status'] == "err") : ?>
|   <div class="alert">
|       <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
|       La modification a échouée.
|   </div>
<?php endif ?>
<?php if (isset($_GET['status']) && $_GET['status'] == "horsinv") : ?>
|   <div class="alert">
|       <span class="closebtn" onclick="this.parentElement.style.display='none';">&times;</span>
|       La modification a échouée, l'ordinateur est hors de l'inventaire UM.
|   </div>
<?php endif ?>

<!-- FIN MESSAGES D'ALERTE -->
```

8.12 Test - Recherche

```

if (isset($_POST['submit']))
{
    if (empty($_POST['NumSerie']) && empty($_POST['IdLieu']) && empty($_POST['numBat'])
        && empty($_POST['etat']) && empty($_POST['garantie']) && empty($_POST['statut'])
        && empty($_POST['sans_num_inv']) && empty($_POST['sans_num_immo']))
    {
        echo '<script language="Javascript"> document.location.replace("'" . $view_path . 'recherche.php?status=vide"); </script>';
        exit();
    }
    else
    {
        $numInv = "non_vide";
        $numImmo = "non_vide";

        if(empty($_POST['sans_num_inv'])){
            $numInv = "vide";
        }
        if(empty($_POST['sans_num_immo'])){
            $numImmo = "vide";
        }
        if(isset($_POST['NumSerie']) && $_POST['NumSerie']!="")
        {
            if(count_rech_numSerie($_POST['NumSerie']) == 0)
            {
                echo '<script language="Javascript"> document.location.replace("'" . $view_path . 'recherche.php?status=vide"); </script>';
                exit();
            }
            else
            {
                debut_selection();
                rech_numSerie($_POST['NumSerie']);
                fin_selection();
            }
        }
        else
        {
            if(isset($_POST['IdLieu']) && $_POST['IdLieu']!="")
            {
                if(count_rech_idLieu($_POST['IdLieu'], $_POST['etat'], $_POST['garantie'], $_POST['statut'], $numInv, $numImmo) == 0)
                {
                    echo '<script language="Javascript"> document.location.replace("'" . $view_path . 'recherche.php?status=vide"); </script>';
                    exit();
                }
                else{
                    debut_selection();
                    rech_idLieu($_POST['IdLieu'], $_POST['etat'], $_POST['garantie'], $_POST['statut'], $numInv, $numImmo);
                    fin_selection();
                }
            }
            else
            {
                if(isset($_POST['numBat']) && $_POST['numBat']!="")
                {
                    if(count_rech_numBat($_POST['numBat'], $_POST['etat'], $_POST['garantie'], $_POST['statut'], $numInv, $numImmo) == 0) {
                        echo '<script language="Javascript"> document.location.replace("'" . $view_path . 'recherche.php?status=vide"); </script>';
                        exit();
                    }
                    else{
                        debut_selection();
                        rech_numBat($_POST['numBat'], $_POST['etat'], $_POST['garantie'], $_POST['statut'], $numInv, $numImmo);
                        fin_selection();
                    }
                }
                else
                {
                    if(count_rech_reste($_POST['etat'], $_POST['garantie'], $_POST['statut'], $numInv, $numImmo) == 0) {
                        echo '<script language="Javascript"> document.location.replace("'" . $view_path . 'recherche.php?status=vide"); </script>';
                        exit();
                    }
                    else{
                        debut_selection();
                        rech_reste($_POST['etat'], $_POST['garantie'], $_POST['statut'], $numInv, $numImmo);
                        fin_selection();
                    }
                }
            }
        }
    }
}

```

8.13 Test - Répétition Panier

```

if(isset($_POST["action"]))
{
    if($_POST["action"] == "add")
    {
        $product_id = $_POST["product_id"];
        $product_num = $_POST["product_num"];
        $product_type = $_POST["product_type"];
        $product_modele = $_POST["product_modele"];
        $product_etat = $_POST["product_etat"];
        $product_statut = $_POST["product_statut"];
        $product_garantie = $_POST["product_garantie"];
        $product_lieu = $_POST["product_lieu"];
        $product_idcr = $_POST["product_idcr"];
        $product_nomcr = $_POST["product_nomcr"];
        $product_remarque = $_POST["product_remarque"];
        $product_inventaire = $_POST["product_inventaire"];
        $product_immobilisation = $_POST["product_immobilisation"];
        for($count = 0; $count < count($product_id); $count++)
        {
            //test de la répétition des ordinateurs dans le panier
            $cart_product_id = array_keys($_SESSION["shopping_cart"]);
            if(in_array($product_id[$count], $cart_product_id))
            {

            }
            else
            {
                $item_array = array(
                    'product_id'          => $product_id[$count],
                    'product_num'         => $product_num[$count],
                    'product_type'        => $product_type[$count],
                    'product_modele'      => $product_modele[$count],
                    'product_etat'        => $product_etat[$count],
                    'product_statut'      => $product_statut[$count],
                    'product_garantie'    => $product_garantie[$count],
                    'product_lieu'         => $product_lieu[$count],
                    'product_idcr'         => $product_idcr[$count],
                    'product_nomcr'        => $product_nomcr[$count],
                    'product_remarque'     => $product_remarque[$count],
                    'product_inventaire'   => $product_inventaire[$count],
                    'product_immobilisation' => $product_immobilisation[$count],
                );
                $_SESSION["shopping_cart"][$product_num[$count]] = $item_array;
            }
        }
        header("Location:../view/panier.php");
    }
}

```

8.14 Visuels de l'application web



AJOUTER DES ORDINATEURS

[Rechercher](#) [Panier \[2\]](#)

Accueil
Ajouter des ordinateurs
Fiches inventaire
Ordinateurs à renouveler
Ordinateurs en panne et en réparation
Actions dépendantes du panier
Modification inventaire
Changement état
Sortie FDS
Sortie UM
Édition caractéristiques

Ajouter des ordinateurs via un tableau

Fichier Aucun fichier sélectionné.

[Télécharger le tableau avec la trame incluse](#)

N.B. : La première ligne du fichier CSV est destinée à la trame. Il faut donc impérativement entrer les informations des ordinateurs dès la deuxième ligne.

Ajouter un ordinateur manuellement

Commande passée par * : LIRMM N° Série * : ex:4RTSYU Type * : ex:Workstation

Fabricant * : ex:Dell Modèle * : ex:Optiplex 7440 AIO Nombre d'années de garantie * :

N° Commande SIFAC : Fournisseur * : ex:Dell

Date de début de garantie (=date d'expédition) * : jj / mm /aaaa

Centre responsable de l'ordinateur * : FDS Salle * : 01/36/02/36.204 N° Inventaire :

N° Immobilisation :

* OBLIGATOIRE



FICHES INVENTAIRE

[Rechercher](#) [Panier \[2\]](#)

Accueil
Ajouter des ordinateurs
Fiches inventaire
Ordinateurs à renouveler
Ordinateurs en panne et en réparation
Actions dépendantes du panier
Modification inventaire
Changement état
Sortie FDS
Sortie UM
Édition caractéristiques

Génération des fiches inventaire

- Entrée d'inventaire
- Modification d'inventaire
- Sortie d'inventaire

Période : de jj / mm /aaaa à jj / mm /aaaa

© footer

[Déconnexion](#)



ORDINATEURS À RENOUVELER

[Rechercher](#) [Panier \[2\]](#)

Accueil
Ajouter des ordinateurs
Fiches inventaire
Ordinateurs à renouveler
Ordinateurs en panne et en réparation
Actions dépendantes du panier
Modification inventaire
Changement état
Sortie FDS
Sortie UM
Édition caractéristiques

Date de renouvellement :

 jj / mm / aaaa

Il y a 3 ordinateur(s) à renouveler.

Localisation	n° Série	Type	Model	Etat	Date de début de garantie	Date de fin de garantie	Durée de garantie
01/6/00/6.02	1FCRMH5K	Workstation	OptiPlex 790	Fonctionnel	2016-01-07	2018-01-07	2
01/6/00/6.02	FCRMH5J	work	OptiPlex 790	En panne	2017-04-24	2020-04-24	3
/36.204	TGIJ542J3	Workstation	Optiplex 70	Fonctionnel	2018-02-28	2019-02-28	1

© footer

[Déconnexion](#)


ORDINATEURS EN PANNE / EN REPARATION

[Rechercher](#) [Panier \[2\]](#)

Accueil
Ajouter des ordinateurs
Fiches inventaire
Ordinateurs à renouveler
Ordinateurs en panne et en réparation
Actions dépendantes du panier
Modification inventaire
Changement état
Sortie FDS
Sortie UM
Édition caractéristiques

Ordinateurs en panne / en réparation

Etat :

En panne
En réparation
En panne/En réparation

[Déconnexion](#)


RECHERCHE

[Rechercher](#) [Panier \[2\]](#)

Accueil
Ajouter des ordinateurs
Fiches inventaire
Ordinateurs à renouveler
Ordinateurs en panne et en réparation
Actions dépendantes du panier
Modification inventaire
Changement état
Sortie FDS
Sortie UM
Édition caractéristiques

[Ne remplissez que ce qui est nécessaire à votre recherche]

Numéro de série : ex: 8D9VJX2 Numéro du Batiment : Lieu : Etat : Garantie :

Statut : Ordinateur sans n° inventaire Ordinateur sans n° d'immobilisation

Sous garantie

Hors garantie

[Déconnexion](#)



SÉLECTION

[Rechercher](#) [Panier \[2\]](#)

Accueil
Ajouter des ordinateurs
Fiches inventaire
Ordinateurs à renouveler
Ordinateurs en panne et en réparation
Actions dépendantes du panier
Modification inventaire
Changement état
Sortie FDS
Sortie UM
Édition caractéristiques

Tout sélectionner

Choix	N° Série	Type	Modèle	État	Statut	Date de fin de garantie	Lieu	Centre Responsable	Remarque	N° Inventaire	N° Immobilisation
<input checked="" type="checkbox"/>	8CRMH5J	work	optiplex	En panne	Hors inventaire UM	2021-01-07	01/6/00/6.02	LIRMM	manque	UM000002	57M4HG13
<input checked="" type="checkbox"/>	11AZ45ER	Workstation	Optiplex	Fonctionnel	Hors inventaire UM	2023-09-08	01/6/00/6.02	LMGC			
<input type="checkbox"/>	3DRMH5J	workstation	optiplex	Fonctionnel	Dans inventaire FDS	2022-12-04	01/6/00/6.03	LMGC	manque	UM005000	89SQW032
<input type="checkbox"/>	4FRMH5J	Workstation	OptiPlex 790	En panne	Hors inventaire UM	2021-09-08	01/6/00/6.03	FDS	remarque	UM004200	35D7PE45
<input type="checkbox"/>	AZ12FG56	Workstation	Optiplex	Fonctionnel	Hors inventaire UM	2023-04-09	01/6/00/6.03	FDS			

[Déconnexion](#)


PANIER

[Rechercher](#) [Panier \[2\]](#)

Accueil
Ajouter des ordinateurs
Fiches inventaire
Ordinateurs à renouveler
Ordinateurs en panne et en réparation
Actions dépendantes du panier
Modification inventaire
Changement état
Sortie FDS
Sortie UM
Édition caractéristiques

Le contenu du panier :

N° Série	Type	Modèle	État	Statut	Date de fin de garantie	Lieu	Centre Responsable	Remarque	N° Inventaire	N° Immobilisation	
8CRMH5J	work	optiplex	En panne	Hors inventaire UM	2021-01-07	01/6/00/6.02	LIRMM	manque	UM000002	57M4HG13	Retirer
11AZ45ER	Workstation	Optiplex	Fonctionnel	Hors inventaire UM	2023-09-08	01/6/00/6.02	LMGC				Retirer

Il y a 2 ordinateur(s) sélectionné(s).

[Vider le panier](#)
[Déconnexion](#)


MODIFICATION INVENTAIRE

[Rechercher](#) [Panier \[2\]](#)

Accueil
Ajouter des ordinateurs
Fiches inventaire
Ordinateurs à renouveler
Ordinateurs en panne et en réparation
Actions dépendantes du panier
Modification inventaire
Changement état
Sortie FDS
Sortie UM
Édition caractéristiques

Modification inventaire

Nouvel emplacement

Nouveau lieu :

Nouveau centre responsable :

[Valider](#)

Panier

N° Série	Type	Modèle	État	Statut	Date de fin de garantie	Lieu	Centre Responsable	Remarque	N° Inventaire	N° Immobilisation	
8CRMH5J	work	optiplex	En panne	Hors inventaire UM	2021-01-07	01/6/00/6.02	LIRMM	manque	UM000002	57M4HG13	Retirer
11AZ45ER	Workstation	Optiplex	Fonctionnel	Hors inventaire UM	2023-09-08	01/6/00/6.02	LMGC				Retirer

Il y a 2 ordinateur(s) sélectionné(s).

[Déconnexion](#)

© footer



CHANGEMENT D'ETAT

[Rechercher](#) [Panier \[2\]](#)

Accueil

Ajouter des ordinateurs

Fiches inventaire

Ordinateurs à renouveler

Ordinateurs en panne et en réparation

Actions dépendantes du panier

Modification inventaire

Changement état

Sortie FDS

Sortie UM

Édition caractéristiques

Changeement d'état

Nouvel état :

- En panne
- Fonctionnel
- En réparation

Panier

N° Série	Type	Modèle	Etat	Statut	Date de fin de garantie	Lieu	Centre Responsable	Remarque	N° Inventaire	N° Immobilisation	
8CRMH5J	work	optiplex	En panne	Hors inventaire UM	2021-01-07	01/6/00/6.02	LIRMM	manque	UM000002	57M4HG13	<input type="button" value="Retirer"/>
11AZ45ER	Workstation	Optiplex	Fonctionnel	Hors inventaire UM	2023-09-08	01/6/00/6.02	LMGC				<input type="button" value="Retirer"/>

Il y a 2 ordinateur(s) sélectionné(s).

[Déconnexion](#)


SORTIE INVENTAIRE FDS DANS UM

[Rechercher](#) [Panier \[2\]](#)

Accueil

Ajouter des ordinateurs

Fiches inventaire

Ordinateurs à renouveler

Ordinateurs en panne et en réparation

Actions dépendantes du panier

Modification inventaire

Changement état

Sortie FDS

Sortie UM

Édition caractéristiques

Sortie inventaire FDS dans UM

Nouvel emplacement

Nouveau lieu :

Nouveau centre responsable :

Panier

N° Série	Type	Modèle	Etat	Statut	Date de fin de garantie	Lieu	Centre Responsable	Remarque	N° Inventaire	N° Immobilisation	
8CRMH5J	work	optiplex	En panne	Hors inventaire UM	2021-01-07	01/6/00/6.02	LIRMM	manque	UM000002	57M4HG13	<input type="button" value="Retirer"/>
11AZ45ER	Workstation	Optiplex	Fonctionnel	Hors inventaire UM	2023-09-08	01/6/00/6.02	LMGC				<input type="button" value="Retirer"/>

Il y a 2 ordinateur(s) sélectionné(s).

[Déconnexion](#)

© footer



SORTIE UM

[Rechercher](#) [Panier \[2\]](#)

Accueil

Ajouter des ordinateurs

Fiches inventaire

Ordinateurs à renouveler

Ordinateurs en panne et en réparation

Actions dépendantes du panier

Modification inventaire

Changement état

Sortie FDS

Sortie UM

Édition caractéristiques

Sortir de l'inventaire UM

Détails sortie :

Votre commentaire ici

Panier

N° Série	Type	Modèle	Etat	Statut	Date de fin de garantie	Lieu	Centre Responsable	Remarque	N° Inventaire	N° Immobilisation	
8CRMH5J	work	optiplex	En panne	Hors inventaire UM	2021-01-07	01/6/00/6.02	LIRMM	manque	UM000002	57M4HG13	<input type="button" value="Retirer"/>
11AZ45ER	Workstation	Optiplex	Fonctionnel	Hors inventaire UM	2023-09-08	01/6/00/6.02	LMGC				<input type="button" value="Retirer"/>

Il y a 2 ordinateur(s) sélectionné(s).

[Déconnexion](#)



EDITION CARACTERISTIQUES

[Rechercher](#) [Panier \[2\]](#)

Accueil
Ajouter des ordinateurs
Fiches inventaire
Ordinateurs à renouveler
Ordinateurs en panne et en réparation
 Actions dépendantes du panier
Modification inventaire
Changement état
Sortie FDS
Sortie UM
Édition caractéristiques

Edition des caractéristiques :

Remplissez les champs que vous voulez mettre à jour :

Type : <input type="text" value="ex: Workstation"/>	N° d'immobilisation : <input type="text" value="ex: 10LFH472"/>
Modèle: <input type="text" value="ex: Optiplex 7440 AIO"/>	N° inventaire : <input type="text" value="ex: UM004000"/>
Remarque : <input type="text" value="ex: Manque souris"/>	Nombre d'années de garantie : <input type="text" value=""/>

Panier

N° Série	Type	Modèle	État	Statut	Date de fin de garantie	Lieu	Centre Responsable	Remarque	N° Inventaire	N° Immobilisation	
8CRMH5J	work	optiplex	En panne	Hors inventaire UM	2021-01-07	01/6/00/6.02	LIRMM	manque	UM000002	57M4HG13	<input type="button" value="Retirer"/>
11AZ45ER	Workstation	Optiplex	Fonctionne	Hors inventaire UM	2023-09-08	01/6/00/6.02	LMGC				<input type="button" value="Retirer"/>

Il y a 2 ordinateur(s) sélectionné(s).

[Déconnexion](#)

8.15 Product Backlog

Product Backlog

Estimation : Par taille des t-shirts (*Small - Simple, Medium, Large, XLarge - Très difficile*)

Statut : Trello (*To do, Doing, Done*)

Priorité : De 1 (faible) à 5 (forte)

id	Titre	Scénario	Niveau utilisateur	Estimation	Statut	Priorité
COMPTE						
1	S'identifier	En tant qu'utilisateur, je souhaite m'identifier, afin d'accéder à mon compte.	Entre login et mot-de-passe.	S	To do	1
2	Se déconnecter	En tant qu'utilisateur, je souhaite me déconnecter, afin de limiter l'accès à mon compte.	Clique sur le bouton déconnecter.	S	To do	1
3	Afficher les informations du profil	En tant qu'utilisateur, je souhaite pouvoir afficher les informations de mon profil, afin d'accéder et modifier mes informations.	Affiche les informations du profil.	S	To do	1
4	Avoir un compte admin pour la sécurité	En tant qu'administrateur, je souhaite avoir tous les droits, afin d'assurer la sécurité de l'application.	Création d'un profil administrateur et d'un profil utilisateur.	S	To do	1
FICHE ENTRÉE						
5	Ajouter une fiche d'entrée	En tant qu'utilisateur, je souhaite ajouter une fiche d'entrée à un ordinateur afin d'ajouter un ordinateur du parc.	Ajoute une fiche inventaire d'entrée.	L	To do	5
SOUS SCÉNARIO						
	Modifier l'état	En tant qu'utilisateur,	Sélectionne un	S	To do	5

		ce dernier.				
8	Effectuer une sortie de l'inventaire UM	En tant qu'utilisateur, je souhaite effectuer une sortie d'inventaire afin de supprimer un ordinateur du parc.	Effectue une sortie de l'inventaire UM	M	To do	5
SOUS SCÉNARIO						
	Faire une modification inventaire FDS	En tant qu'utilisateur, je souhaite modifier le lieu d'un ordinateur et/ ou son CR afin de mettre à jour les informations de ce dernier.	Remplace le lieu existant par le nouveau lieu et/ou CR de l'ordinateur.	M	To do	5
FILTRES						
9	Rechercher un ordinateur suivant des filtres puis les sélectionnés	En tant qu'utilisateur, je souhaite voir la liste des ordinateurs avec des filtres choisis, afin de les sélectionner	Liste les ordinateurs, et les trier selon la salle, le département, la garantie, le statut, ou encore l'état.	L	To do	5
10	Afficher les ordinateurs en panne et réparation	En tant qu'utilisateur, je souhaite lister la l'état des ordinateurs par salle afin d'avoir plus de renseignement lisible.	Liste les ordinateurs triés par salle, avec pour chacun son état	M	To do	4
STATUT ORDINATEUR						
11	Modifier le statut d'un ordinateur.	En tant qu'utilisateur, je souhaite modifier le statut d'un ordinateur, afin de mettre à jour ces informations.	Sélectionne un des statuts suivants : -Rebut -don -achat -déplacement	S	To do	5
FIN DE GARANTIE						
12	Afficher le bilan de renouvellement	En tant qu'utilisateur, je souhaite afficher le bilan de	Affiche le bilan de renouvellement	M	To do	5

	pour une année.	renouvellement d'une année, afin d'avoir plus de renseignement sur la garantie à renouveler des ordinateurs	d'une année.			
Sous Scénario						
ASSISTANCE						
13	Afficher la personne à contacter en cas de problème.	En tant qu'utilisateur, je souhaite avoir les coordonnées d'une personne à contacter afin d'avoir une assistance qui puisse me dépanner.	Affiche les coordonnées de la personne à contacter en cas de problème.	S	To do	4
HISTORIQUE						
14	Afficher l'historique des centres responsables pour chaque ordinateur ainsi que les dates des changements.	En tant qu'utilisateur, je souhaite lister l'historique des centres responsables pour chaque ordinateur ainsi que les dates des changements, afin de pouvoir afficher l'historique de chaque ordinateur.	Liste l'historique des centres responsables pour chaque ordinateur avec les dates des changements.	XL	To do	5
EXPORT DE FICHIER						
15	Générer automatiquement les bons de sortie.	En tant qu'utilisateur, je souhaite recevoir un bon de sortie afin de ne pas perdre de temps à le faire manuellement.	Génère un bon de sortie ou modification.	L	To do	5
Sous Scénario						
	Générer fiche sortie FDS	En tant qu'utilisateur, je souhaite générer un bon de sortie FDS, afin d'avoir toutes les	Génère une fiche de sortie FDS , avec toutes les informations.	L	To do	4

		informations pour remplir la fiche.				
16	Générer automatiquement les bons d'entrée inventaire	En tant qu'utilisateur, je souhaite recevoir l'ensemble des informations à fournir pour une fiche inventaire.	Génère une entrée inventaire avec toutes les informations.	L	To do	5
IMPORT DE FICHIER						
17	Importer les fiches inventaire au format csv	En tant qu'utilisateur, je souhaite pouvoir importer une fiche au format csv, afin de générer les fiches associées	Importe sur son ordinateur une fiche en csv.	L	To do	5

8.16 Exemple de compte rendu

Compte rendu 8 du 26 Mars 2020

TER - BD PARC INFO

ALAOUI YOUSFI

CHAMROUK

EL OUAZZANI

LAAFOU

MAJDOL

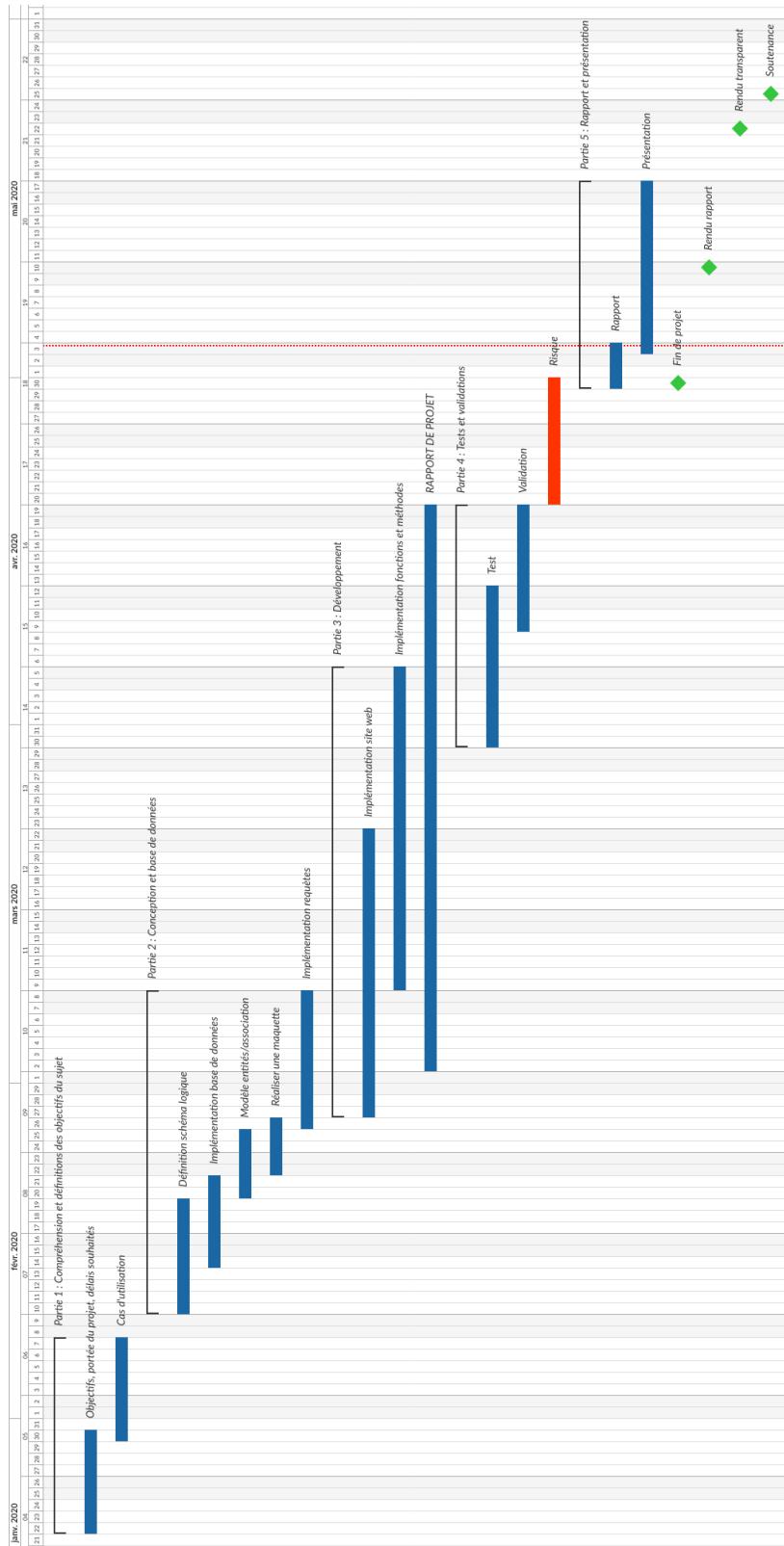
Réunion :

- Mettre des étoiles sur les champs à remplir obligatoirement
- Dans l'exemple du fournisseur, mettre Dell (pas FDS)
- Ajouter message pour ajout ordinateur : "Votre ordinateur a bien été ajouté"
- S'il y a plusieurs contacts pour un centre responsable, sélectionner le premier
- Modifier affichage des tableaux (différencier titre / données)
- Rassembler Sélection recherche et Édition de caractéristiques dans un seul onglet
- Les actions dans selectionRecherche : Modification inventaire FDS, Sortie inventaire FDS, Sortie inventaire UM, édition de caractéristiques, changements d'état
- L'attribut InfoFicheInventaire doit contenir toutes les informations pour permettre l'édition des fiches entrée / modification/ sortie
- Ajouter un attribut **typefiche** dans l'association Géré_Par_Hist qui va prendre soit entrée, sortie ou modification (si c'est entrée la dateDebutGestion sera 0000-00-00 et la dateFinGestionCR sera la date d'entrée dans l'inventaire)
- Ajouter dans l'attribut infoficheinventaire qui est dans la table Géré_Par_Hist l'attribut detailSortieInventaireUM ,ancienContact, nouveauContact.
- On garde l'IdCR d'une entrée inventaire (non NULL, c'est l'idCR entré par l'utilisateur lors de l'ajout d'un ordinateur), il représente le CR qui gère l'ordinateur dès son entrée.
- Renommer l'onglet Sélection recherche par Panier
- Mettre l'onglet Historique plus haut
- Rassembler les onglets Ordinateur en panne et Ordinateur en réparation sous un même onglet

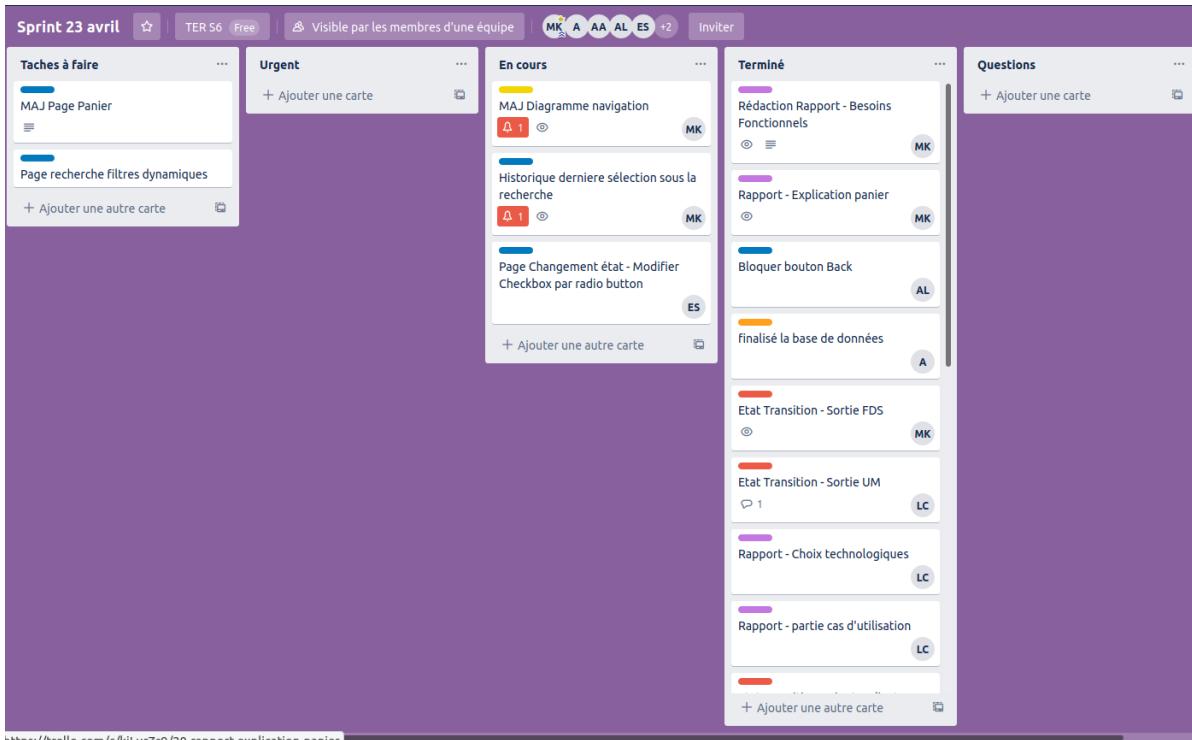
À faire :

- Mettre à jour le modèle entité association , modèle logique et dictionnaire de données.
- Mettre à jour la base de données sur PhpMyAdmin en MySQL.
- Mettre à jour l'attribut infoFicheInventaire.
- Se renseigner sur les session et les cookies (possibilité de garder la sélection).
- Ajouter les requêtes et les triggers pour les nouveaux changements.
- Faire les Diagrammes de navigation pour les maquettes.
- Mettre à jour les maquettes.
- Mettre à jour tous les scénarios .
- Mettre à jour l'application web.

8.17 Diagramme de Gantt

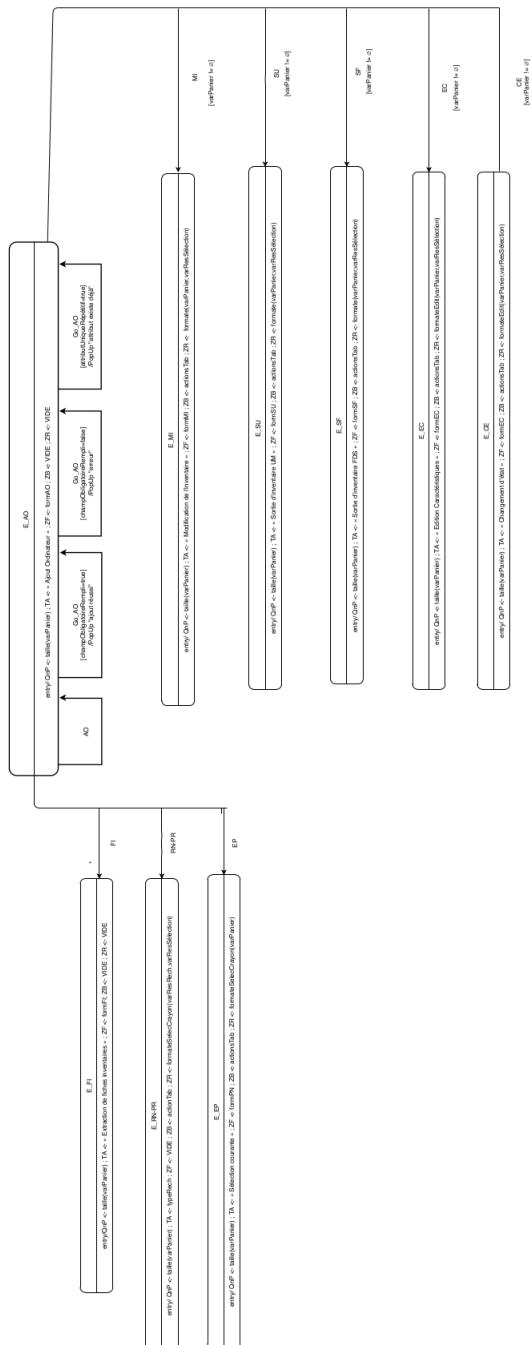


8.18 Trello

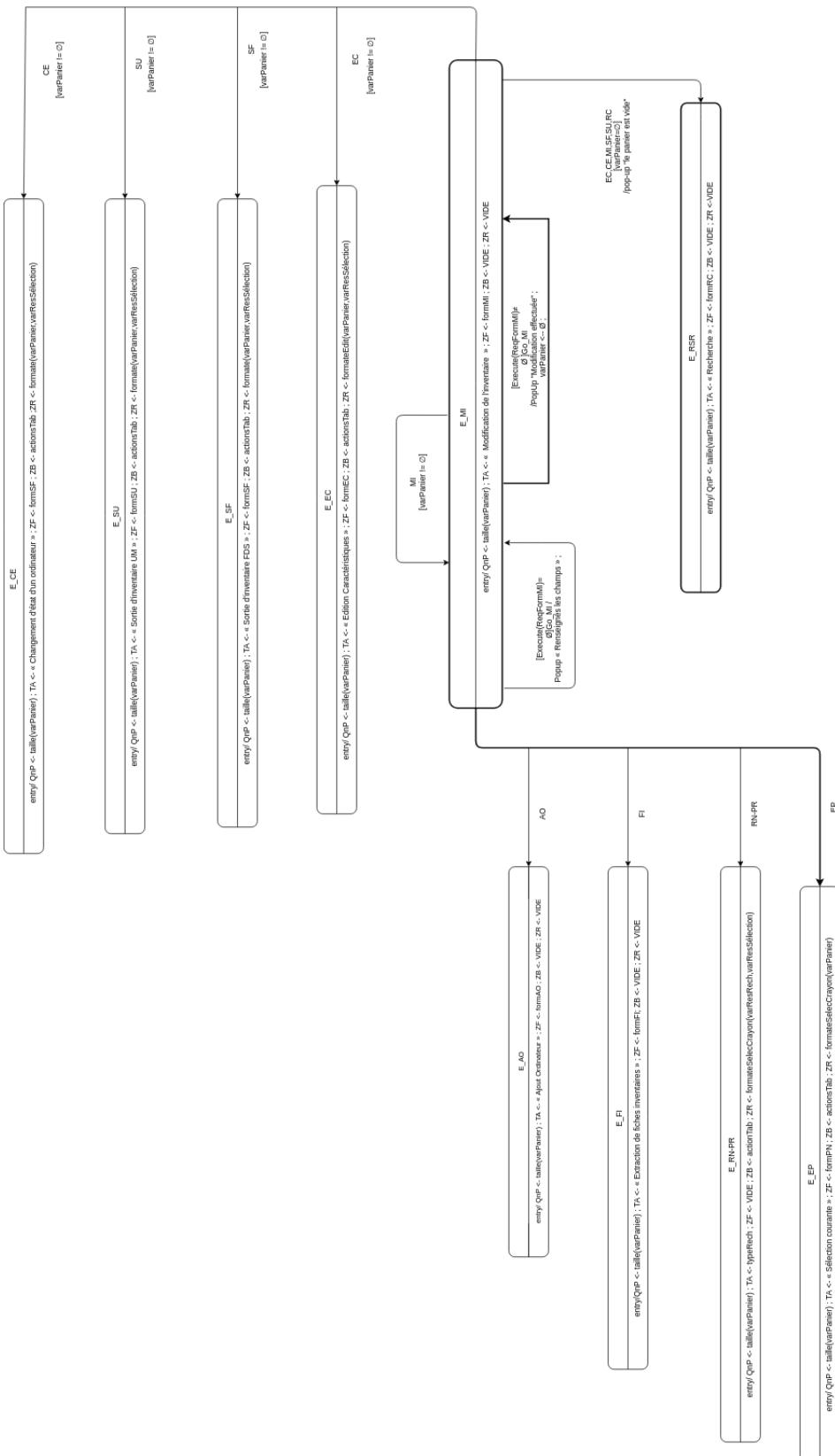


8.19 Diagrammes d'état-transition

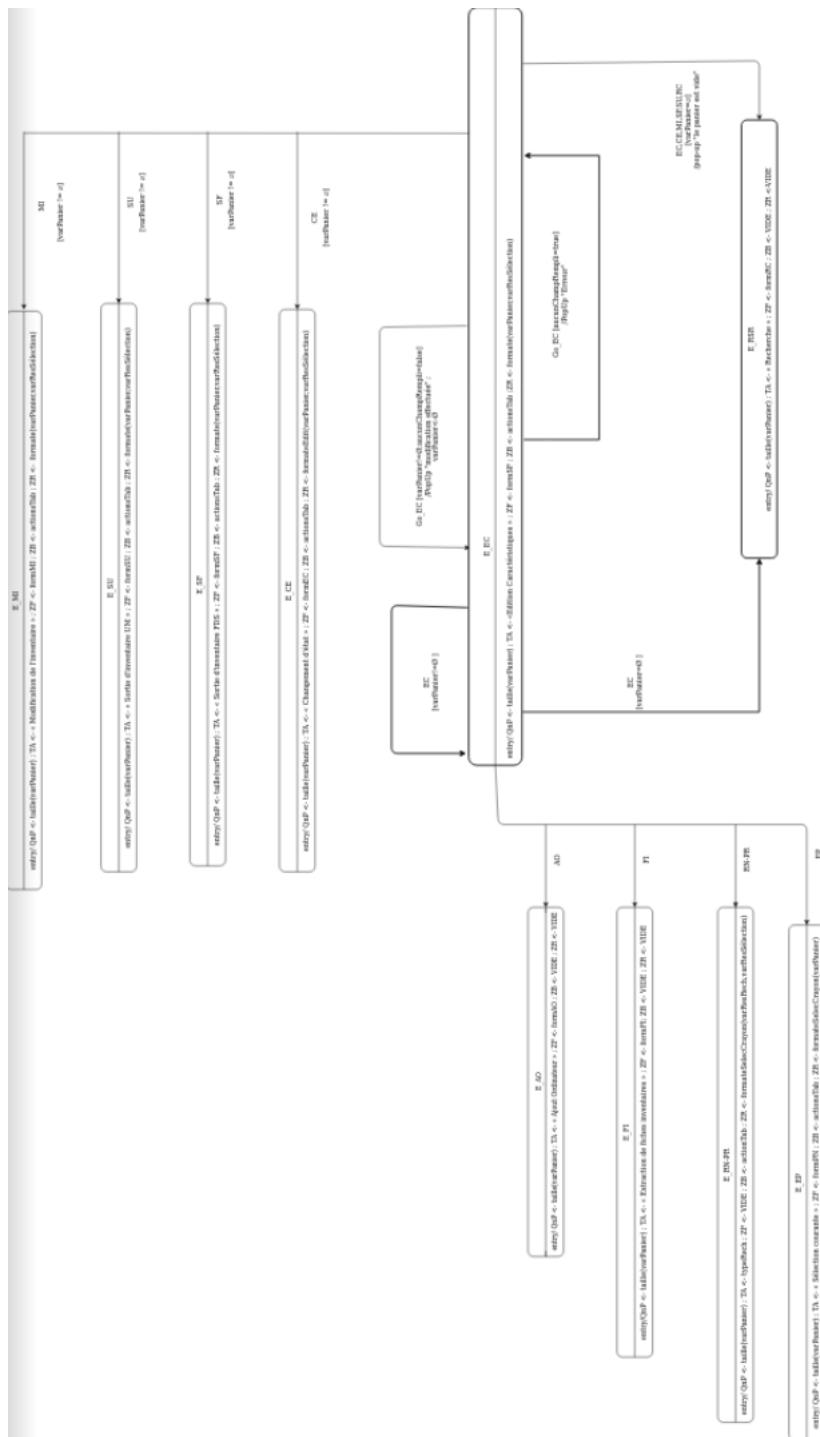
8.19.1 Ajout ordinateur



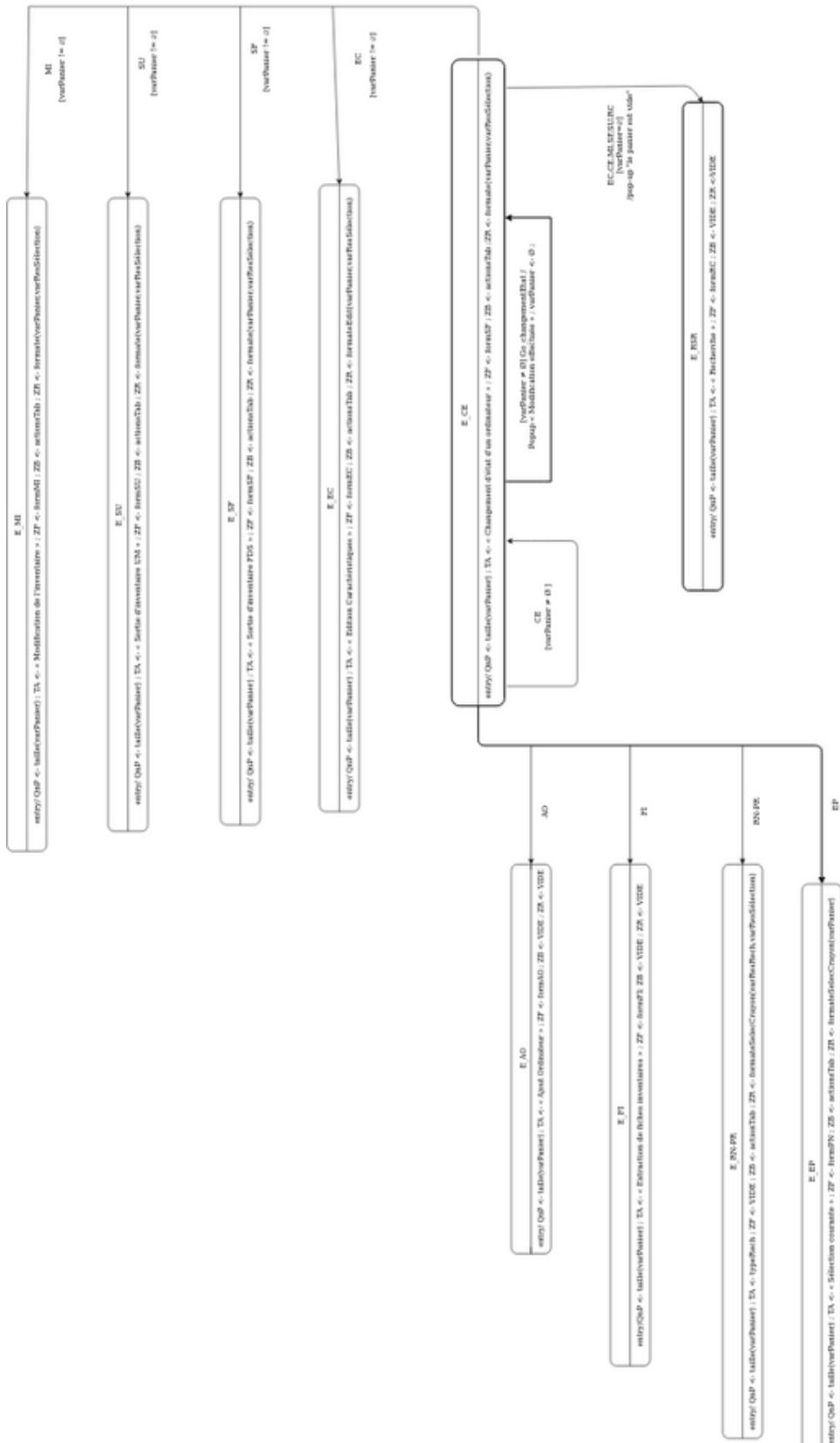
8.19.2 Modification inventaire



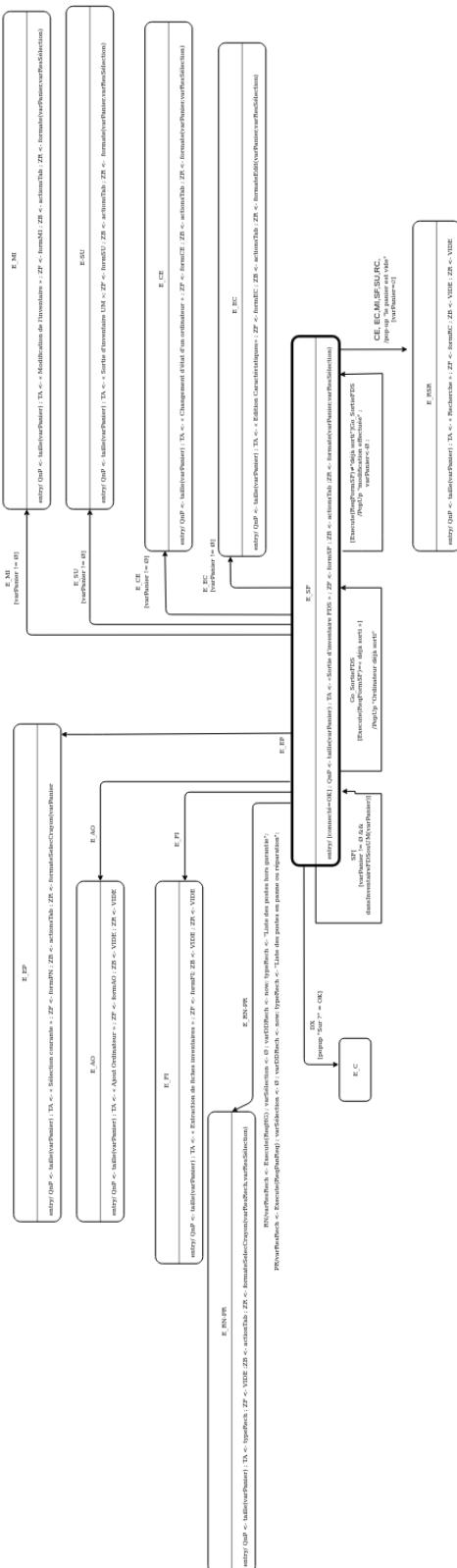
8.19.3 Édition caractéristiques



8.19.4 Changement d'état



8.19.5 Sortie inventaire FDS



8.19.6 Sortie inventaire UM

