



DOCUMENTATION TECHNIQUE

PC BUILDER

Loureiro Pinto Ronaldo. 05/2018. V.1.0



TABLE DES MATIÈRES

1	Introduction	5
2	Rappel du cahier des charges	6
2.1	Objectifs	6
2.2	Spécifications.....	6
2.3	Restrictions.....	6
2.4	Environnement.....	6
2.5	Organisation	7
2.6	Livrables	7
3	Analyse fonctionnelle	9
3.1	Fonctionnalités.....	9
3.1.1	Créer une configuration PC	9
3.1.2	Enregistrer une configuration	9
3.1.3	Modifier une configuration.....	9
3.1.4	Supprimer une configuration	10
3.1.5	S'inscrire à l'application.....	10
3.1.6	Se connecter à l'application.....	10
3.2	Interfaces.....	10
3.2.1	Barre de navigation.....	10
3.2.1.1	Utilisateur non-connecté.....	10
3.2.1.2	Utilisateur connecté.....	10
3.2.2	Page d'accueil	11
3.2.3	Page du configurateur	12
3.2.4	Page Mes configurations	13
3.2.5	Pop-up d'inscription	14
3.2.6	Pop-up de Connexion.....	14
3.3	Cas d'utilisation	15

3.3.1	Diagramme des cas d'utilisation	15
3.3.2	Utilisateur	15
4	Analyse organique	18
4.1	Diagramme de classes sans les associations	18
4.2	Base de données	19
4.2.1	Diagramme de la base de données	19
4.2.2	Détails des tables	20
4.2.2.1	Table « Brands »	20
4.2.2.2	Table « Processors »	20
4.2.2.3	Table « Sockets »	20
4.2.2.4	Table « Motherboards »	21
4.2.2.5	Table « Formats_Motherboard »	21
4.2.2.6	Table « Motherboards_Types_Storage »	21
4.2.2.7	Table « Rams »	21
4.2.2.8	Table « Types_Rams »	21
4.2.2.9	Table « Frequency_Rams »	22
4.2.2.10	Table « Graphics_Cards »	22
4.2.2.11	Table « Chipsets »	22
4.2.2.12	Table « Coolers »	22
4.2.2.13	Table « Coolers_Sockets »	22
4.2.2.14	Table « Cases »	23
4.2.2.15	Table « Cases_Formats_Motherboard »	23
4.2.2.16	Table « Cases_Formats_Power »	23
4.2.2.17	Table « Storages »	23
4.2.2.18	Table « Interfaces_Storage »	23
4.2.2.19	Table « Powers »	24
4.2.2.20	Table « Formats_Power »	24

4.2.2.21	Table « Configurations »	24
4.2.2.22	Table « Configurations_Storages »	24
4.2.2.23	Table « Users »	25
4.3	Architecture du code	25
4.3.1	Arborescence de fichier	25
4.3.2	Classe « ComponentManager »	25
4.3.3	Classe « ConfigurationManager »	28
4.3.4	Classe « UserManager »	30
4.3.5	Principales fonctions Javascript	30
4.4	Outils externes	32
4.4.1	Bootstrap 4	32
4.4.2	JQuery 3.3.1	32
4.4.3	Bitbucket	33
4.4.4	Font Awesome	33
5	Tests	34
5.1	Connexion & Inscription	34
5.2	Le configurateur	36
5.3	Page « Mes configurations »	39
6	Conclusion	41
6.1	Améliorations possibles	41
6.2	Bilan personnel	41
7	Bibliographie	42
7.1	Codes repris	42
7.2	Sites utilisés	42
7.3	Aides reçues	42
8	Planning	43
8.1	Planning prévisionnel	43
8.2	Planning effectif	44

9	Table des Illustrations.....	45
9.1	Figures	45
9.2	Tableaux	46
10	Annexes.....	47

1 INTRODUCTION

Cette documentation a pour but de détailler les étapes nécessaires à la réalisation de l'application web « PC Builder », réalisé dans le cadre du TPI (Travail Pratique Individuel).

Dans le cadre de ce travail, l'application que je souhaite réaliser permet aux utilisateurs de créer une configuration PC fonctionnelle. Le configurateur guidera l'utilisateur étape par étape et au fur et à mesure que les composants sont sélectionnés, le choix sera restreint pour assurer la compatibilité. L'utilisateur pourra appliquer des filtres afin que son choix soit plus facile à trouver. Il pourra également se connecter s'il souhaite enregistrer sa configuration et pourra par la suite la modifier ou la supprimer.

2 RAPPEL DU CAHIER DES CHARGES

2.1 OBJECTIFS

Durant les 10 jours de TPI, L'objectif est de créer une application WEB qui permet aux utilisateurs de créer une configuration PC avec des composants compatibles entre eux et de permettre aux utilisateurs connectés, d'enregistrer, de modifier et de supprimer une configuration.

2.2 SPÉCIFICATIONS

L'application permettra de :

- Créer une configuration PC qui contient que des composants compatibles entre eux
- Enregistrer une configuration
- Modifier une configuration enregistrée
- Supprimer une configuration enregistrée
- S'inscrire à l'application
- Se connecter à l'application

2.3 RESTRICTIONS

Seul la partie utilisateur sera développé

L'utilisateur non-connecté peut :

- Utiliser le configurateur PC
- Créer un compte (S'inscrire) et se connecter

L'utilisateur connecté peut :

- Utiliser le configurateur PC
- Enregistrer une configuration
- Modifier une configuration existante
- Supprimer une configuration existante

2.4 ENVIRONNEMENT

- PC sous Windows
- IDE à choix (PhpStorm / Visual Studio Code / NetBeans)

- Outil de versioning de code (GIT avec repository en ligne sur GitHub / Bitbucket / GitLab)
- Serveur web (EasyPhp / Apache)
- Navigateur web (Google Chrome / Mozilla Firefox)
- Outil de gestion de BD (Workbench / PhpMyAdmin)
- Outil bureautique à choix pour les documents (Google Docs, MSOffice, OpenOffice)
- Un logiciel pour créer les éventuels diagrammes, maquettes nécessaires pour l'analyse (p. ex. Visio, Gliffy, balsamiq, etc)

2.5 ORGANISATION

Élève :

- Ronaldo Loureiro Pinto, ronaldo.lrrpnt@eduge.ch

Maitre d'apprentissage :

- Jasmina Travnjak, jasmina.travnjak@edu.ge.ch

Experts :

- Yvan Poulin, yvpoulin@gmail.com
- Serge Murisier, serge@murisier.com

2.6 LIVRABLES

Pour la fin du TPI, le 24 mai 2018 au secrétariat de l'école

- 1 carnet de bord
- 1 exemplaire papier du mémoire de travail, avec en annexe
 - Documentation technique
 - Mode d'emploi
 - Code source

Par messagerie aux experts et au maître d'apprentissage

- PDFs du mémoire, de la documentation technique, du mode d'emploi et des sources imprimées du logiciel

Par messagerie ou autre support électronique au maître d'apprentissage

- Les sources (projet logiciel) et la version exécutable

Pour le jour de la présentation

- Les documents utilisés pour la présentation par messagerie au maître d'apprentissage

Pour la fin juin

- Résumé du TPI sur une à deux pages, en vue du concours suisse des TPI

3 ANALYSE FONCTIONNELLE

3.1 FONCTIONNALITÉS

3.1.1 CRÉER UNE CONFIGURATION PC

Cette fonctionnalité permet à un utilisateur de pouvoir créer une configuration PC. L'utilisateur aura le choix entre plusieurs composants. Le configurateur fera en sorte que tous les composants soient compatibles entre eux au fur et à mesure que l'utilisateur fera son choix. Pour faciliter la configuration d'un PC, le configurateur sera une sorte d'assistant avec des étapes à suivre (wizard). L'utilisateur aura aussi la possibilité de pouvoir filtrer les composants avec des filtres s'il sait ce qu'il veut.

L'utilisateur pourra configurer son PC en choisissant entre les composants suivants¹ :

- Des processeurs (Marque, Socket)
- Des carte-mères (Marque, Chipset, Format)
- Des barrettes de mémoires vives (Marque, Type)
- Des cartes-graphique (Marque, Chipset)
- Des systèmes de refroidissement (Marque, Type)
- Des boîtiers (Marque)
- Des périphériques de stockage (Marque, Type)
- Des alimentations (Marque)

3.1.2 ENREGISTRER UNE CONFIGURATION

Cette fonctionnalité permet à un utilisateur connecté de pouvoir enregistrer une configuration qu'il aura créée. Il pourra le faire lors de la dernière étape du configurateur où il devra donner un nom à sa configuration avant de pouvoir l'enregistrer. Pour enregistrer une configuration l'utilisateur doit être connecté à l'application.

3.1.3 MODIFIER UNE CONFIGURATION

Cette fonctionnalité permet à un utilisateur connecté de pouvoir modifier une configuration qu'il aurait précédemment enregistrée.

¹ Les filtres disponibles pour chaque composant sont écrits entre parenthèses à côté des composants

3.1.4 SUPPRIMER UNE CONFIGURATION

Cette fonctionnalité permet à un utilisateur connecté de pouvoir supprimer une configuration qu'il aurait précédemment enregistrée.

3.1.5 S'INSCRIRE À L'APPLICATION

Cette fonctionnalité permet à un utilisateur de pouvoir créer un compte en fournissant un pseudo, un mot de passe et un e-mail.

3.1.6 SE CONNECTER À L'APPLICATION

Cette fonctionnalité permet à un utilisateur qui possède un compte de pouvoir se connecter afin de pouvoir enregistrer, modifier et supprimer une configuration.

3.2 INTERFACES

3.2.1 BARRE DE NAVIGATION

3.2.1.1 UTILISATEUR NON-CONNECTÉ

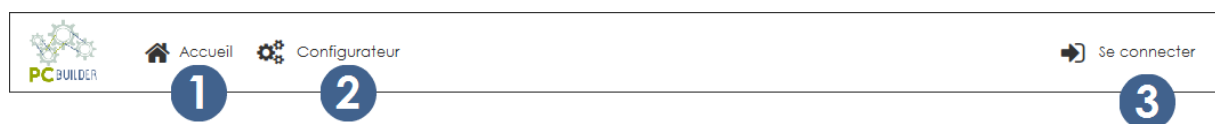


Fig. 1 - Barre de navigation d'un utilisateur non-connecté

- 1 - Le lien « Accueil » permet à l'utilisateur d'aller à la page d'accueil (voir Fig. 3 - Page d'accueil).
- 2 - Le lien « Configurateur » permet à l'utilisateur d'aller au configurateur PC (voir Fig. 4 - Page du configurateur).
- 3 - Le lien « Se connecter » permet à l'utilisateur d'ouvrir la pop-up qui permet de se connecter au site (voir Fig. 7 - Pop-up de connexion).

3.2.1.2 UTILISATEUR CONNECTÉ

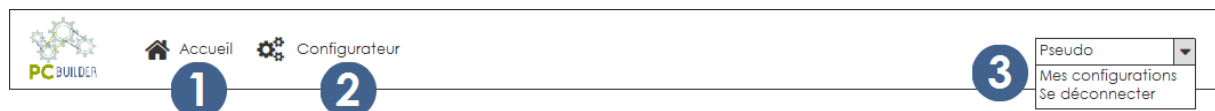


Fig. 2 - Barre de navigation d'un utilisateur connecté

- 1 - Le lien « Accueil » permet à l'utilisateur d'aller à la page d'accueil (voir Fig. 3 - Page d'accueil).

- 2 - Le lien « Configurateur » permet à l'utilisateur d'aller au configurateur PC (voir Fig. 4 - Page du configurateur).
- 3 - Le lien « Mes configurations » permet à l'utilisateur d'aller à la page des configurations qu'il a enregistré. Le lien « Se déconnecter » permet à l'utilisateur de se déconnecter du site (voir Fig. 5 - Page Mes configurations).

3.2.2 PAGE D'ACCUEIL



Fig. 3 - Page d'accueil

La page d'accueil accueille l'utilisateur avec un message de bienvenue et une phrase d'accroche.

- 1 - Le lien « Configurateur » permet à l'utilisateur d'aller au configurateur PC (voir Fig. 4 - Page du configurateur).
- 2 - Le lien « Se connecter » permet à l'utilisateur d'ouvrir la pop-up qui permet de se connecter au site (voir Fig. 7 - Pop-up de connexion).

3.2.3 PAGE DU CONFIGURATEUR

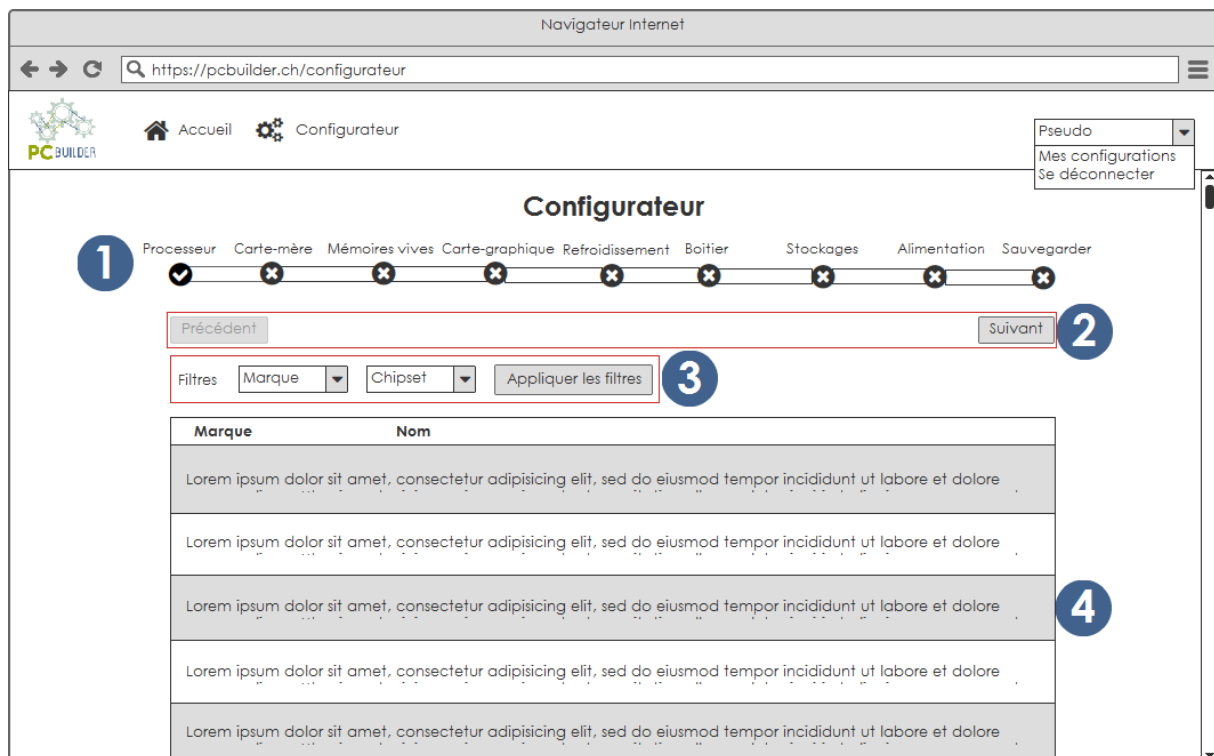


Fig. 4 - Page du configurateur

- 1 - La barre de progression indique à l'utilisateur à quelle étape il se trouve et ce qui manque à choisir pour terminer la configuration.
- 2 - Le bouton « Suivant » permet de passer à l'étape suivante et le bouton « Précédent » permet de revenir à l'étape précédente.
- 3 - Les filtres permettent de filtrer les composants qui sont affichés en n°4. Pour appliquer les filtres, il faut cliquer sur le bouton « Appliquer les filtres ».
- 4 - Les composants disponibles sont affichés sous forme d'un tableau.

3.2.4 PAGE MES CONFIGURATIONS

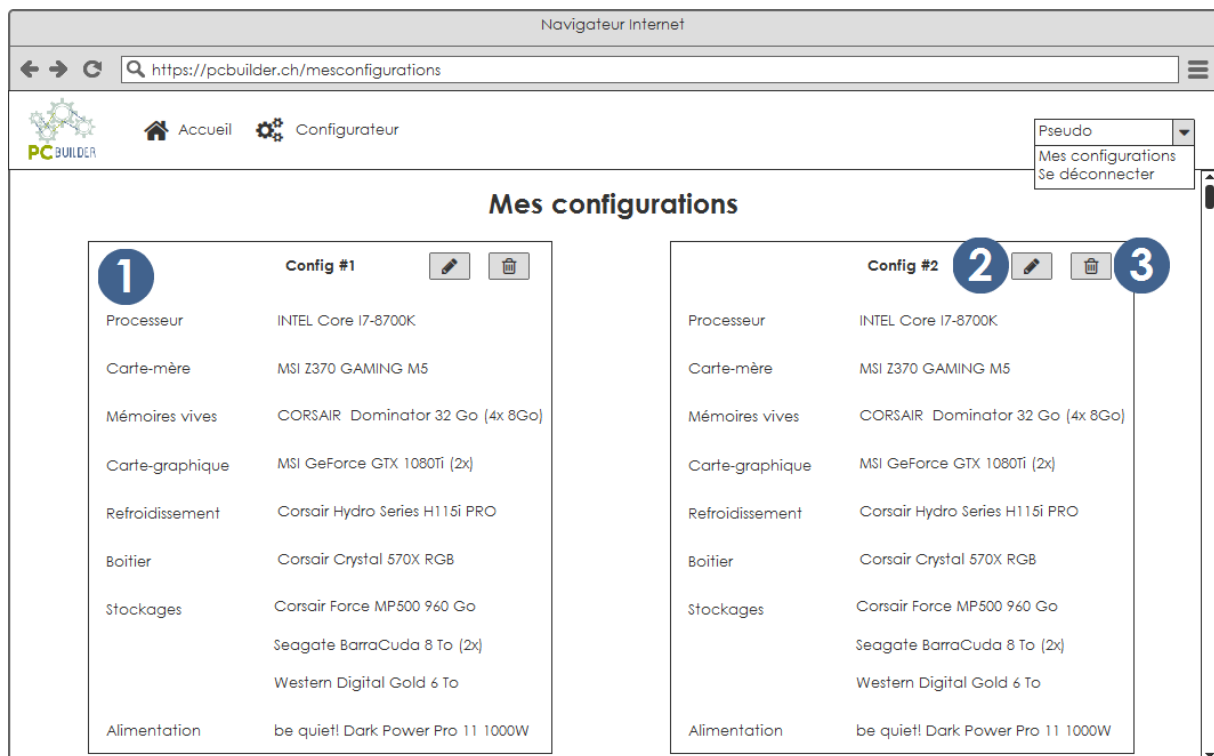
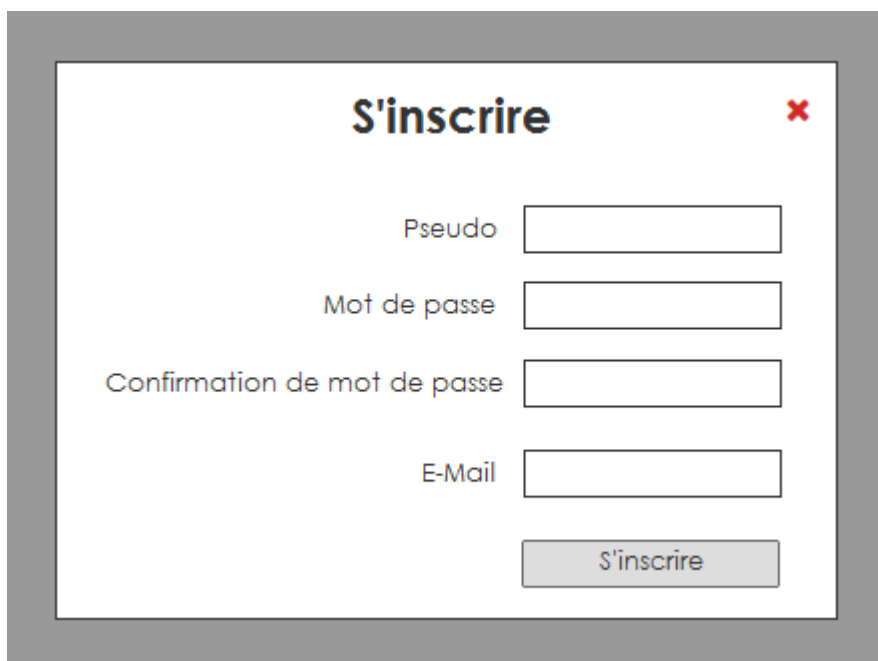


Fig. 5 - Page Mes configurations

Cette page est uniquement disponible si l'utilisateur est connecté au site.

- 1 - Les configurations enregistrées apparaissent sous formes de tableau avec les détails de la configuration.
- 2 - Le bouton « ✎ » permet à l'utilisateur de modifier la configuration. L'utilisateur sera redirigé vers le configurateur afin de modifier sa configuration.
- 3 - Le bouton « 🗑 » permet à l'utilisateur de supprimer la configuration. Une alerte demandant confirmation apparaîtra afin que l'utilisateur confirme son choix de suppression.

3.2.5 POP-UP D'INSCRIPTION



The image shows a pop-up window titled "S'inscrire" with a red close button (X) in the top right corner. The form contains four input fields: "Pseudo", "Mot de passe", "Confirmation de mot de passe", and "E-Mail". Below these fields is a button labeled "S'inscrire".

Fig. 6 - Pop-up d'inscription

La pop-up d'inscription permet à l'utilisateur de s'inscrire au site en indiquant un pseudo, un mot de passe et un e-mail. Le bouton « S'inscrire » permet à l'utilisateur d'envoyer le formulaire d'inscription.

3.2.6 POP-UP DE CONNEXION



The image shows a pop-up window titled "Se connecter" with a red close button (X) in the top right corner. The form contains two input fields: "Pseudo" and "Mot de passe". Below these fields is a link that says "Pas encore inscrit ? [Inscrivez-vous ici !](#)". At the bottom is a button labeled "Se connecter".

Fig. 7 - Pop-up de connexion

La pop-up de connexion permet à l'utilisateur de pouvoir se connecter au site en indiquant son pseudo et son mot de passe. Le lien « S'inscrire » permet d'ouvrir la pop-up d'inscription (voir Fig. 6 - Pop-up d'inscription).

3.3 CAS D'UTILISATION

3.3.1 DIAGRAMME DES CAS D'UTILISATION

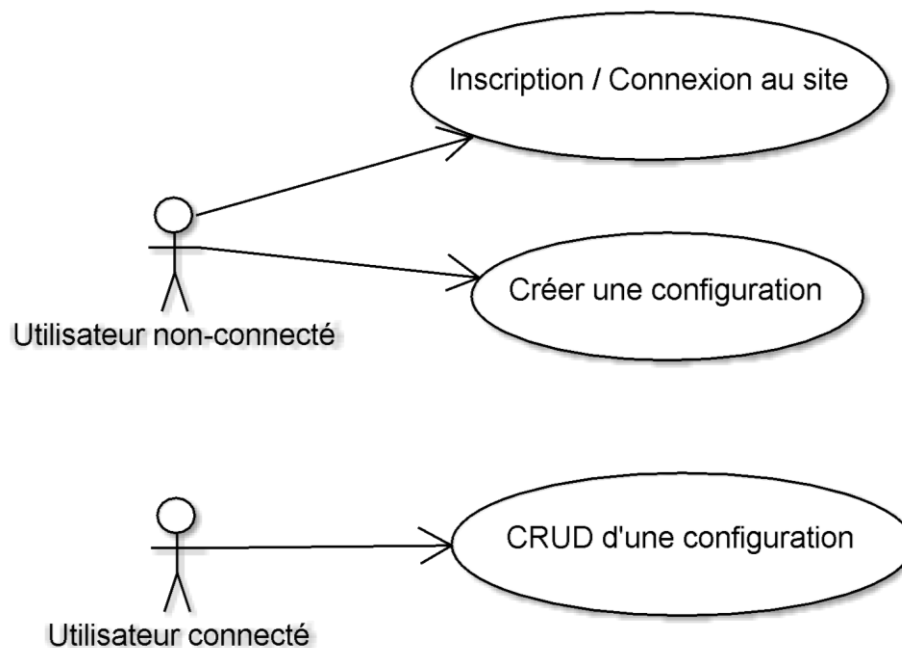


Fig. 8 - Diagramme des cas d'utilisation

3.3.2 UTILISATEUR

Nom UC : CRUD d'une configuration par un utilisateur connecté

Acteur principal : Utilisateur

Déclencheur : L'utilisateur accède au site.

Précondition : être connecté

Flot principal :

1. Le système affiche la page d'accueil.
2. L'utilisateur clique sur « Configurateur ».
3. Le système redirige l'utilisateur sur le configurateur et affiche les processeurs.
4. L'utilisateur applique des filtres (Marque, Socket).
5. Le système affiche les processeurs compatibles en fonction des filtres.

6. L'utilisateur sélectionne un processeur et clique sur « Suivant ».
7. Le système enregistre son choix et affiche les cartes-mères compatibles.
8. L'utilisateur applique des filtres (Marque, Chipset, Format).
9. Le système affiche les cartes-mères compatibles en fonction des filtres.
10. L'utilisateur sélectionne une carte-mère et clique sur « Suivant ».
11. Le système enregistre son choix et affiche les mémoires vives compatibles.
12. L'utilisateur applique des filtres (Marque, Type).
13. Le système affiche les mémoires vives compatibles en fonction des filtres.
14. L'utilisateur sélectionne une mémoire vive, indique la quantité qu'il souhaite et clique sur « Suivant ».
15. Le système enregistre son choix et affiche les cartes-graphiques compatibles.
16. L'utilisateur applique des filtres (Marque, Chipset).
17. Le système affiche les cartes-graphiques compatibles en fonction des filtres.
18. L'utilisateur sélectionne une carte-graphique, indique la quantité qu'il souhaite et clique sur « Suivant ».
19. Le système enregistre son choix et affiche les systèmes de refroidissements compatibles.
20. L'utilisateur applique des filtres (Marque, Type).
21. Le système affiche les systèmes de refroidissements compatibles en fonction des filtres.
22. L'utilisateur sélectionne un système de refroidissement et clique sur « Suivant ».
23. Le système enregistre son choix et affiche les boîtiers compatibles.
24. L'utilisateur applique un filtre (Marque).
25. Le système affiche les boîtiers compatibles en fonction du filtre.
26. L'utilisateur sélectionne un boîtier et clique sur « Suivant ».
27. Le système enregistre son choix et affiche les périphériques de stockage compatibles.
28. L'utilisateur applique des filtres (Marque, Type).
29. Le système affiche les périphériques de stockage compatible en fonction des filtres.

30. L'utilisateur sélectionne des périphériques de stockages, indique la quantité qu'il souhaite pour chaque périphériques et clique sur « Suivant ».
31. Le système enregistre ses choix et affiche les alimentations compatibles.
32. L'utilisateur applique des filtres (Marque).
33. Le système affiche les alimentations compatibles en fonction des filtres.
34. L'utilisateur sélectionne une alimentation et clique sur « Suivant ».
35. Le système enregistre son choix et affiche le formulaire de sauvegarde de la configuration.
36. L'utilisateur donne un nom à sa configuration et clique sur « Sauvegarder ».
37. Le système enregistre sa configuration et le redirige sur la page « Mes configurations ».

Flot alternatif :

- 2a. L'utilisateur clique sur « Mes configurations ».
 - 2a1. Le système redirige sur la page « Mes configurations » et affiche les configurations sauvegardées de l'utilisateur.
 - 2a2. L'utilisateur clique sur le bouton « ✎ » d'une configuration.
 - 2a3. Le système redirige l'utilisateur sur le configurateur en chargeant sa configuration et affiche les processeurs.
 - 2a4. Retour en 4.
- 2b. L'utilisateur clique sur « Mes configurations ».
 - 2b1. Le système redirige sur la page « Mes configurations » et affiche les configurations sauvegardées de l'utilisateur.
 - 2b2. L'utilisateur clique sur le bouton « 🗑 » d'une configuration.
 - 2b3. Le système affiche un message demandant à l'utilisateur de confirmer son geste.
 - 2b4. L'utilisateur confirme son geste.
 - 2b5. Le système supprime la configuration.
 - 2b6. Fin du UC.

4 ANALYSE ORGANIQUE

4.1 DIAGRAMME DE CLASSES SANS LES ASSOCIATIONS

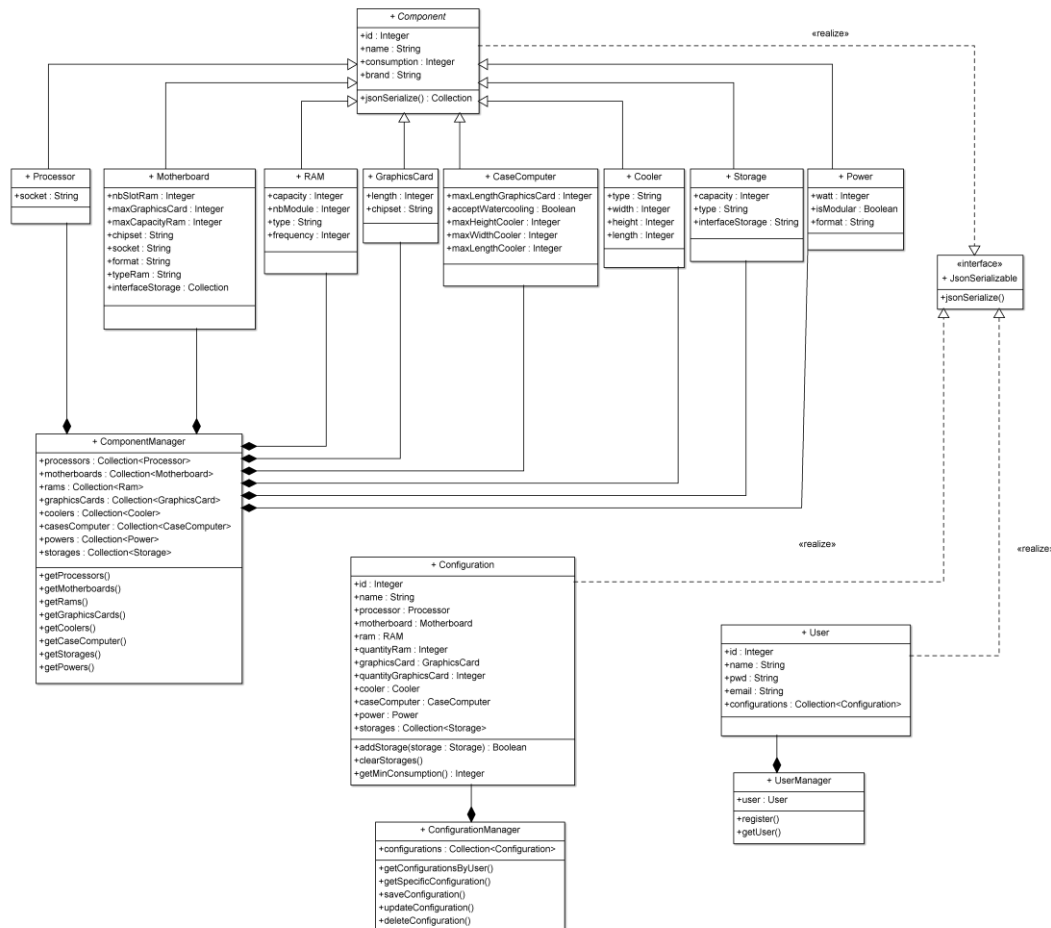


Fig. 9 - Diagramme de classes sans les liens d'association

Les associations n'ont pas été faites et les fonctions des classes n'ont pas été toutes mises pour plus de visibilité.

Classe Motherboard : interfaceStorage est un tableau qui contiendra le nom de l'interface en tant qu'index et la quantité de ports disponible en tant que valeur.

Classe Configuration : storages est un tableau qui stocke les périphériques de stockage qui a été choisi par l'utilisateur. La clé du tableau est composée de « Id_du périphérique-quantité_désiré » et la valeur contient l'objet « storage ».

4.2 BASE DE DONNÉES

4.2.1 DIAGRAMME DE LA BASE DE DONNÉES

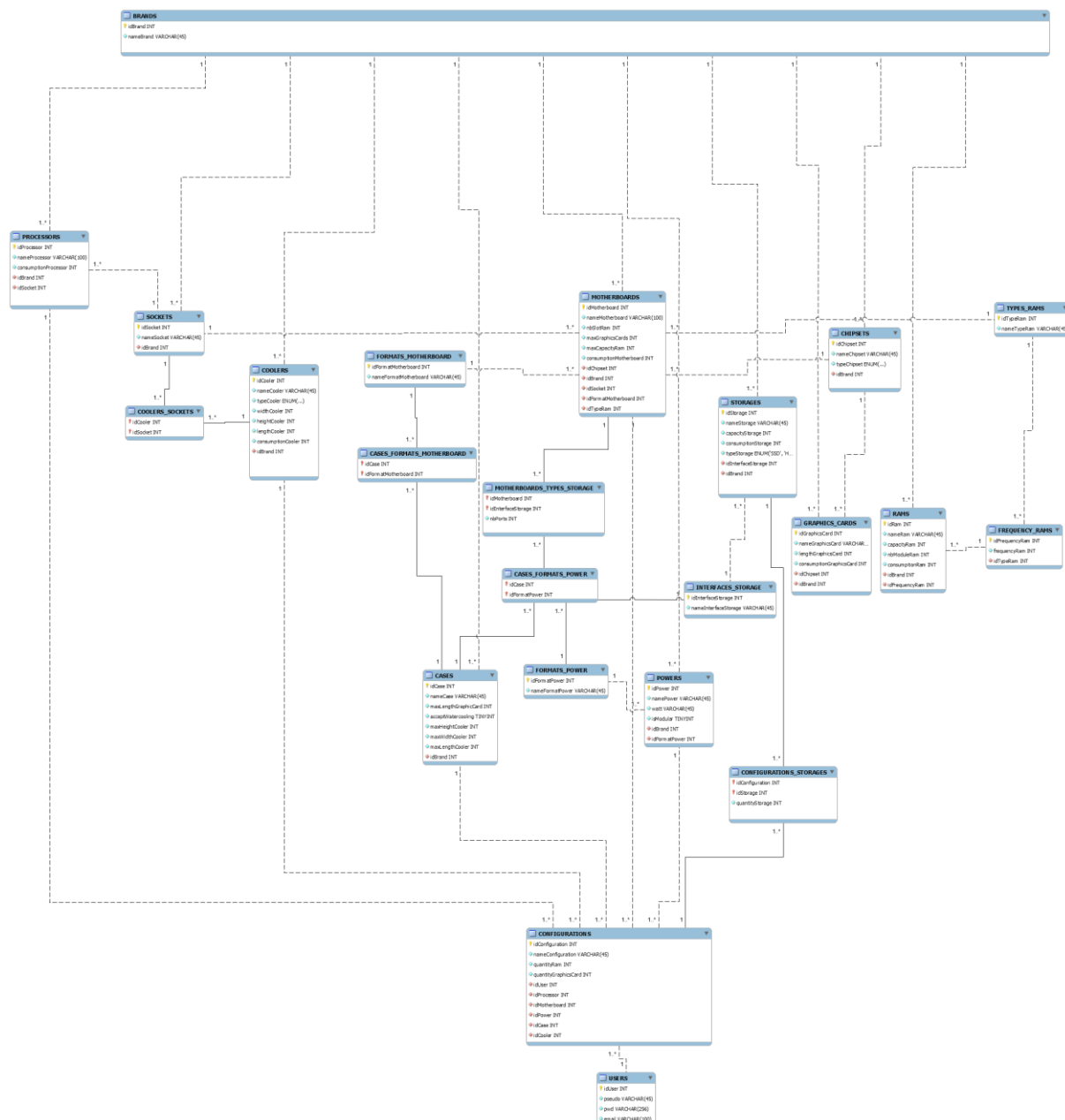


Fig. 10 - Digramme de base de données

Chaque type de composant possède une table avec les différentes informations du composant. Il y a également des tables de liaison afin de savoir les compatibilités entre les composants. Cela explique le nombre important de tables. Le détail des tables est affiché ci-dessous.

4.2.2 DÉTAILS DES TABLES

Abréviation utilisée	Signification
PK	Clé primaire
NN	Non Nul (Empêche la valeur NULL)
UQ	Unique (Empêche que la même valeur soit deux fois dans la table)
AI	Auto incrément

Tableau 1 - Légende des propriétés des attributs

4.2.2.1 TABLE « BRANDS »

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idBrand	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameBrand	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 11 - Table « BRANDS »

4.2.2.2 TABLE « PROCESSORS »






Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idProcessor	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameProcessor	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 consumptionProcessor	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idBrand	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idSocket	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 12 - Table « PROCESSORS »

4.2.2.3 TABLE « SOCKETS »




Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idSocket	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameSocket	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idBrand	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 13 - Table « SOCKETS »

4.2.2.4 TABLE « MOTHERBOARDS »












Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idMotherboard	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameMotherboard	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 nbSlotRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 maxGraphicsCards	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 maxCapacityRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 consumptionMotherboard	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idChipset	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idBrand	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idSocket	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idFormatMotherboard	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idTypeRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 14 - Table « MOTHERBOARDS »

4.2.2.5 TABLE « FORMATS_MOTHERBOARD »

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idFormatMotherboard	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameFormatMotherboard	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 15 - Table « FORMATS_MOTHERBOARD »

4.2.2.6 TABLE « MOTHERBOARDS_TYPES_STORAGE »




Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idMotherboard	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idInterfaceStorage	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 nbPorts	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 16 Table « MOTHERBOARDS_TYPES_STORAGE »

4.2.2.7 TABLE « RAMS »








Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idRam	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameRam	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 capacityRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 nbModuleRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 consumptionRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idBrand	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idFrequencyRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 17 - Table « RAMS »

4.2.2.8 TABLE « TYPES_RAM »


Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idTypeRam	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameTypeRam	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 18 - Table « TYPES_RAM »

4.2.2.9 TABLE « FREQUENCY_RAMS »




Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idFrequencyRam	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 frequencyRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idTypeRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 19 - Table « FREQUENCY_RAMS »

4.2.2.10 TABLE « GRAPHICS_CARDS »







Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idGraphicsCard	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameGraphicsCard	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 lengthGraphicsCard	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 consumptionGraphicsCard	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idChipset	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idBrand	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 20 - Table « GRAPHICS_CARDS »

4.2.2.11 TABLE « CHIPSETS »





Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idChipset	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameChipset	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 typeChipset	ENUM('Processor', 'Graphic')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idBrand	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 21 - Table « CHIPSETS »

4.2.2.12 TABLE « COOLERS »









Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idCooler	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameCooler	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 typeCooler	ENUM('Watercooling'...	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 widthCooler	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 heightCooler	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 lengthCooler	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 consumptionCooler	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idBrand	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 22 - Table « COOLERS »

4.2.2.13 TABLE « COOLERS_SOCKETS »

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idCooler	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idSocket	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 23 - Table « COOLERS_SOCKETS »

4.2.2.14 TABLE « CASES »









Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idCase	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameCase	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 maxLengthGraphicCard	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 acceptWatercooling	TINYINT(4)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 maxHeightCooler	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 maxWidthCooler	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 maxLengthCooler	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idBrand	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 24 - Table « CASES »

4.2.2.15 TABLE « CASES_FORMATS_MOTHERBOARD »

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idCase	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idFormatMotherboard	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 25 - Table « CASES_FORMATS_MOTHERBOARD »

4.2.2.16 TABLE « CASES_FORMATS_POWER »

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idCase	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idFormatPower	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 26 - Table « CASES_FORMATS_POWER »

4.2.2.17 TABLE « STORAGES »








Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idStorage	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameStorage	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 capacityStorage	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 consumptionStorage	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 typeStorage	ENUM('SSD', 'HDD')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idInterfaceStorage	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idBrand	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 27 - Table « STORAGES »

4.2.2.18 TABLE « INTERFACES_STORAGE »



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idInterfaceStorage	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameInterfaceStorage	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 28 - Table « INTERFACES_STORAGE »

4.2.2.19 TABLE « POWERS »







Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idPower	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 namePower	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 watt	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 isModular	TINYINT(4)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idBrand	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idFormatPower	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 29 - Table « POWERS »

4.2.2.20 TABLE « FORMATS_POWER »

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idFormatPower	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameFormatPower	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 30 - Table « FORMATS_POWER »

4.2.2.21 TABLE « CONFIGURATIONS »













Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idConfiguration	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 nameConfiguration	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 quantityRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 quantityGraphicsCard	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idUser	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idProcessor	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idMotherboard	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idRam	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idGraphicsCard	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idPower	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idCase	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idCooler	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 31 - Table « CONFIGURATIONS »

4.2.2.22 TABLE « CONFIGURATIONS_STORAGES »




Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
 idConfiguration	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 idStorage	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 quantityStorage	INT(11)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 32 - Table « CONFIGURATIONS_STORAGES »

4.2.2.23 TABLE « USERS »

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
idUser	INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
pseudo	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
pwd	VARCHAR(256)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
email	VARCHAR(100)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Fig. 33 - Table « USERS »

4.3 ARCHITECTURE DU CODE

4.3.1 ARBORESCENCE DE FICHIER

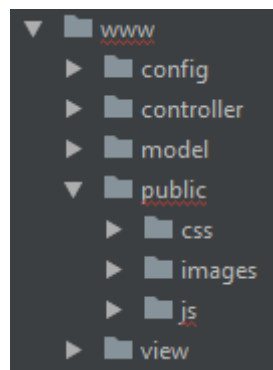


Fig. 34 - Arborescence de fichier

- « config » contient un fichier avec les variables de connexion à la base de données.
- « controller » contient tous les fichiers qui permettent à la vue de communiquer avec le model par des appels AJAX.
- « model » contient toutes les classes de l'application
- « public » contient les fichiers CSS et JS ainsi que les images utilisées.
- « view » contient les pages du site web

4.3.2 CLASSE « COMPONENTMANAGER »

Les compatibilités entre les composants sont gérées par les requêtes SQL en fonction des paramètres donnés aux fonction ci-dessous.

Nom de la fonction	Description
getInstance()	Garantie une seule instance de la classe et retourne une instance de ComponentManager
getProcessors(\$nameBrand = null, \$nameSocket = null)	Récupère les processors en fonction de la marque et/ou du socket, s'ils sont spécifiés et retourne un tableau contenant des objets Processors ou false
getSpecificProcessor(\$idProcessor)	Récupère un processeur en fonction de l'id et retourne l'objet Processor ou null ou false.
getMotherboards(\$nameSocket , \$nameBrand = null, \$nameChipset = null, \$nameFormatMotherboard = null)	Récupère les cartes-mère en fonction du socket. La marque, le chipset et le format peuvent être définis pour affiner le résultat. Retourne un tableau contenant des objets Motherboard ou false
getInterfaceStorageMotherboards()	Ajoute dans le tableau motherboards, le type d'interface des périphériques de stockage et le nombre de ports. En cas d'erreur, la fonction met le tableau à false.
getSpecificMotherboard(\$idMotherboard)	Récupère une carte-mère en fonction de l'id et retourne l'objet Motherboard ou null ou false.
getInterfaceStorageSpecificMotherboard(\$motherboard)	Ajoute dans un objet Motherboard, le type d'interface des périphériques de stockage compatible et le nombre de ports disponibles. Retourne l'objet Motherboard ou false
getRams(\$nameTypeRam , \$nbModuleRam ,	Récupère les rams en fonction du type de ram, du nombre du module et de la

\$maxCapacityRam, \$nameBrand = null)	capacité maximale. La marque peut être définie pour affiner le résultat. Retourne un tableau contenant des objets RAMS ou false
getSpecificRam(\$idRam)	Récupère une ram en fonction de l'id et retourne l'objet Motherboard ou null ou false.
getGraphicsCards(\$nameBrand = null, \$nameChipset = null)	Récupère les cartes-graphiques. La marque et le chipset peuvent être définis pour affiner le résultat. Retourne un tableau contenant des objets GraphicsCard ou false
getSpecificGraphicsCard(\$idGraphicsCard)	Récupère une carte-graphique en fonction de l'id et retourne l'objet GraphicsCard ou null ou false.
getCoolers(\$nameSocket, \$nameBrand = null, \$typeCooler = null)	Récupère les refroidisseurs en fonction du socket. La marque et le type du refroidisseur peuvent être défini pour affiner le résultat. Retourne un tableau contenant des objets Cooler ou false
getSpecificCooler(\$idCooler)	Récupère un refroidisseur en fonction de l'id et retourne l'objet Cooler ou null ou false.
getCases(\$maxLengthGraphicCard, \$maxWidthCooler, \$maxHeightCooler, \$maxLengthCooler, \$nameFormatMotherboard, \$acceptWatercooling = -1, \$nameBrand = null)	Récupère les boîtiers en fonction de la carte-graphique et de la taille du refroidisseur. La marque peut être définie pour affiner le résultat. Retourne un tableau contenant des objets CaseComputer ou false.

getSpecificCase(\$idCase)	Récupère un boîtier en fonction de l'id et retourne l'objet CaseComputer ou null ou false.
getStorages(\$idMotherboard , \$nameBrand = null, \$typeStorage = null)	Récupère les périphériques de stockages en fonction de la carte-mère. La marque peut être définie pour affiner le résultat. Retourne un tableau contenant des objets Storage ou false.
getSpecificStorage(\$idStorage)	Récupère un périphérique de stockage en fonction de l'id et retourne l'objet Storage ou null ou false.
getPowers(\$minConsumption , \$idCase , \$nameBrand = null)	Récupère les alimentations en fonction de la consommation et du boîtier. La marque peut être définie pour affiner le résultat. Retourne un tableau contenant des objets Power ou false.
getSpecificPower(\$idPower)	Récupère une alimentation en fonction de l'id et retourne l'objet Power ou null ou false.

Tableau 2 - Fonction de la classe « ComponentManager »

4.3.3 CLASSE « CONFIGURATIONMANAGER »

Nom de la fonction	Description
getInstance()	Garantie une seule instance de la classe. Retourne une instance de la classe.
getConfigurationsByUser(\$idUser)	Récupère les configurations d'un utilisateur spécifié. Retourne un tableau

	contenant des objets Configuration ou false.
<code>getSpecificConfiguration(\$idUser, \$idConfiguration)</code>	Récupère un configuration spécifique d'un utilisateur spécifique.
<code>getStorageInConfiguration(\$configuration)</code>	Récupère les périphériques de stockage d'une configuration spécifique. Retourne la configuration ou false.
<code>addConfiguration(\$idUser, \$config)</code>	Enregistre une configuration dans la base de données
<code>addStoragesInConfiguration(\$idConfiguration, \$storages)</code>	Enregistre les périphériques de stockage d'une configuration dans la base de données.
<code>deleteStoragesInConfiguration(\$idConfiguration)</code>	Supprime les périphériques de stockage d'une configuration dans la base de données.
<code>updateConfiguration(\$idUser, \$config)</code>	Mets à jour une configuration spécifique d'un utilisateur dans la base de données.
<code>deleteConfiguration(\$idUser, \$idConfiguration)</code>	Supprime une configuration spécifique de la base de données.

Tableau 3 - Fonctions de la classe « ConfigurationManager »

4.3.4 CLASSE « USERMANAGER »

Nom de la fonction	Description
getInstance()	Garantie une seule instance de la classe. Retourne une instance de la classe.
registerUser(\$pseudo, \$email, \$pwd)	Enregistre un utilisateur dans la base de données.
pseudoExist(\$pseudo)	Vérifie si le pseudo existe dans la base de données. Retourne true s'il existe sinon null ou false en cas d'erreur.
emailExist(\$email)	Vérifie si l'email existe dans la base de données. Retourne true s'il existe sinon null ou false en cas d'erreur.
loginUser(\$pseudo)	Récupère les informations d'un utilisateur en fonction du pseudo.

Tableau 4 - Fonction de la classe « UserManager »

4.3.5 PRINCIPALES FONCTIONS JAVASCRIPT

Nom de la fonction	Description
Fichier componentFunctionConfigurator.js	
displayComponent(data, table, arThead, columnName, callbackClick, callbackLoadCurrentConfig, componentName)	Affiche les composants reçus en paramètre dans une <table>. La fonction initialise aussi les événements clic sur les lignes du tableau.
loadCurrentConfig(table, componentName)	Sélectionne les composants qui avaient été choisis par l'utilisateur, au chargement du tableau.
componentClick(componentName, row, component)	Événement clic sur un composant. La fonction permet de sauvegarder le choix de l'utilisateur.

updateWizard()	Mets à jour la barre de progression du configurateur.
step()	Appel les bonnes fonctions en fonction de l'étape courante du configurateur.
saveComponent()	Appel la bonne méthode en fonction de l'étape courante du configurateur, afin de sauvegarder le choix de l'utilisateur sur le serveur.
resetFilter()	Réinitialise les filtres en supprimant les valeurs des filtres.
fillFilter(component)	Remplis les bons filtres en fonction de l'étape courante du configurateur.
Fichier function.js	
get_data(url , callback , params = {}, async , displayMessage = "header")	Effectue un appel ajax et exécute une fonction en cas de succès
getUrlVar(key)	Permet de récupérer des paramètres dans l'url.
errorMessage(message)	Retourne de l'html contenant le message d'erreur.
successMessage(message)	Retourne de l'html contenant le message de succès.
validForm(form)	Bloque le bouton d'envoi du formulaire spécifié tant que les champs requis ne sont pas remplis.
resetForm(form)	Supprime les valeurs des champs d'un formulaire spécifié.

<code>displayTableConfiguration(configuration)</code>	Affiche le détail d'une configuration sous la forme d'un tableau. Retourne le HTML du <table>
--	---

4.4 OUTILS EXTERNES

4.4.1 BOOTSTRAP 4

Bootstrap est une collection d'outils utile à la création du design (graphisme, animation et interactions avec la page dans le navigateur ... etc. ...) de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS, des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option².

Je l'ai utilisé dans mon projet afin d'avoir une interface agréable et responsive³ sans perdre trop de temps. Pour l'inclure, j'utilise les CDNs⁴ disponible sur le site de Bootstrap.

```
<!-- Bootstrap CDN -->
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/css/bootstrap.min.css"
integrity="sha384-9gVQ4dYFwwWSjIDZnLEWnxCjeSWEFphJiwGPXr1jddIhOegiu1FwO5qRGvFXOdJZ4"
crossorigin="anonymous">
<script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.0/umd/popper.min.js"
integrity="sha384-cs/chFZiN24E4KMATldqvsezGxaGsi4hLGOzlXwp5UZB1LY//20VyM2taTB4QvJ"
crossorigin="anonymous"></script>
<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.0/js/bootstrap.min.js"
integrity="sha384-uefMccjFJAIV6A+rW+L4AHf99KvxDjWSu1z9VI8SKNVmz4sk7buKt/6v9KI65qnm"
crossorigin="anonymous"></script>
```

4.4.2 JQUERY 3.3.1

JQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web⁵.

Je l'ai utilisé dans mon projet afin de pouvoir faire des requêtes AJAX et de modifier l'affichage facilement. Pour inclure JQuery j'utilise également le CDN disponible sur le site de JQuery.

² Source Wikipédia : [https://fr.wikipedia.org/wiki/Bootstrap_\(framework\)](https://fr.wikipedia.org/wiki/Bootstrap_(framework))

³ **Responsive** : Le site web s'adapte à tous les types supports afin de garder le confort visuel pour le visiteur

⁴ **CDN (réseau de diffusion de contenu) en français** : Ce sont des serveurs qui proposent du contenu (exemple : Bootstrap) afin d'améliorer la bande passante de notre serveur car le contenu serait téléchargé depuis le CDN.

⁵ Source Wikipédia : <https://fr.wikipedia.org/wiki/JQuery>

```
<!-- JQuery -->
<script src="https://code.jquery.com/jquery-3.3.1.js" integrity="sha256-
2Kok7MbOyxpgUVvAk/HJ2jigOSYS2auK4Pfbzm7uH60=" crossorigin="anonymous"></script>
```

4.4.3 BITBUCKET

Bitbucket est un service web d'hébergement et de gestion de développement logiciel utilisant les logiciels de gestion de versions Git et Mercurial⁶.

Je l'ai utilisé afin d'enregistrer chaque jour mon TPI. En cas de soucis, je peux aussi récupérer ce que j'ai fait les autres jours.

4.4.4 FONT AWESOME

Font Awesome est une police d'écriture et un outil d'icônes qui se base sur CSS, LESS et SASS. Elle a été créée par Dave Gandy afin d'être utilisée avec Bootstrap⁷. Avant la sortie de Bootstrap 4, Font Awesome était intégré à Bootstrap mais il a été enlevé de la V4 de Bootstrap.

J'ai utilisé Font Awesome pour afficher des icônes sur le site. Pour inclure Font Awesome j'utilise le CDN disponible sur le site de Font Awesome.

```
<!-- Font Awesome CDN -->
<link href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-
awesome.min.css" rel="stylesheet" integrity="sha384-
wvfXpqpZVQQGK6TAh5PVlGOfQNHSoD2xbE+QkPxCAFlNEevoEH3Sl0sibVcOQVnN"
crossorigin="anonymous">
<script defer src="https://use.fontawesome.com/releases/v5.0.6/js/all.js"></script>
```

⁶ Source Wikipédia : <https://fr.wikipedia.org/wiki/Bitbucket>

⁷ Source Wikipédia : https://fr.wikipedia.org/wiki/Font_Awesome

5 TESTS

5.1 CONNEXION & INSCRIPTION

N°	A tester	Comment ?	Résultat attendu ?	Résultat ?	Bug ?	Solution si problème ?
1	Inscription au site – Vérifier que la confirmation du mot de passe soit correcte	Dans le formulaire d'inscription site, insérer dans la confirmation du mot de passe, un mot de passe différent.	Une erreur doit apparaître	Ok	∅	∅
2	Inscription au site	Remplir le formulaire d'inscription	Un message « Inscription réussie ! » doit apparaître	Ok. La pop-up de connexion apparaît également	∅	∅
3	Pseudo unique	Remplir le formulaire d'inscription avec un pseudo déjà existant	Un message d'erreur « Le pseudo existe déjà » est affiché	Ok	∅	∅

4	Email unique	Remplir le formulaire d'inscription avec un email déjà existant	Un message d'erreur « L'email est déjà utilisé par un autre compte » est affiché	Ok	∅	∅
5	Connexion au site	Remplir le formulaire de connexion correctement	La page est rechargée et le pseudo apparaît en haut à droite de la barre de navigation.	Ok	∅	∅
6	Connexion au site	Remplir le formulaire de connexion correctement en étant dans le configurateur avec l'étape « Mémoires-vives »	La page est rechargée et le configurateur affiche l'étape « Mémoires-vives »	Déconnecte automatique après une déconnexion	Oui	Nettoyage des variables l'url après la déconnexion
7	La déconnexion	Cliquer sur le lien « déconnexion »	La barre de navigation affiche « Se connecter » à droite	Ok	∅	∅

Tableau 5 - Tests de connexion & inscription

5.2 LE CONFIGURATEUR

N°	A tester	Comment ?	Résultat attendu ?	Résultat ?	Bug ?	Solution si problème ?
8	Le bouton « Suivant »	Sélectionner un composant et cliquer sur le bouton « Suivant »	Le configurateur passe à l'étape suivante	Ok	∅	∅
9	Le bouton « Précédent »	Compléter une étape et cliquer sur le bouton « Précédent »	L'étape précédente s'affiche et le composant choisi est sélectionné	Ok	∅	∅
10	Le configurateur	Aller jusqu'à la dernière étape « Sauvegarder » du configurateur	La dernière étape s'affiche (un formulaire avec un champ texte et un bouton)	Ok	∅	∅
11	La configuration est sauvegardée correctement	Aller jusqu'à la dernière étape	Vérifier que à chaque étape, les choix faits	Ok	∅	∅

		« Sauvegarder » du configurateur et rafraîchir la page.	auparavant sont correct			
12	La barre de progression	Faire les étapes du configurateur	La barre progression se met à jour en affichant la bonne étape	Ok	∅	∅
13	Les filtres	Sélectionner des valeurs dans le filtres de chaque étapes	Vérifier que les composants affichés correspondent aux valeurs des filtres	Ok mais si aucun composant n'est trouvé un message d'erreur s'affiche	∅	∅
14	Le bouton « Réinitialiser les filtres »	Sélectionner des valeurs dans les filtres et cliquer ensuite sur le bouton « Réinitialiser les filtres »	Les valeurs dans les filtres sont réinitialisées et tous les composants sont réaffichés comme au début	Ok	∅	∅

15	Le bouton « Réinitialiser »	Faire une configuration complète et cliquer sur le bouton « Réinitialiser »	Le configurateur revient à la première étape et il n'affiche pas les choix de la dernière configuration	Ok	∅	∅
16	Les champs « Quantité »	Sélectionner un composant, changer la valeur du champs au maximum possible et cliquer sur « Suivant »	Aucune erreur ne doit apparaître	Une erreur apparaît en mettant la quantité au maximum	Oui	Correction dans la condition de validation de la RAM (setRam.php) car avant je multipliais la capacité des rams par le nombre de barrettes mais il faut multiplier par la quantité.
17	La sauvegarde d'une configuration	Faire toutes les étapes du configurateur et cliquer sur le bouton « Sauvegarder la configuration »	L'utilisateur est redirigé sur la page « Mes configurations »	Ok	∅	∅

18	Sauvegarder les modifications d'une configuration	Faire toutes les étapes du configurateur et aller sur la page « Mes configurations »	La configuration est affichée avec les modifications	Ok	∅	∅
----	---	--	--	----	---	---

Tableau 6 - Tests du configurateur

5.3 PAGE « MES CONFIGURATIONS »

N°	A tester	Comment ?	Résultat attendu ?	Résultat ?	Bug ?	Solution si problème ?
19	La page « Mes configurations »	Accéder à la page sans être connecté	Une redirection sur la page d'accueil doit avoir lieu	Ok	∅	∅
20	Affichage des configurations sauvegarder	Enregistrer plusieurs configurations et aller ensuite sur la page « Mes configurations »	Les configurations sauvegardées sont affichées	Ok	∅	∅
21	Modifier une configuration	Cliquer sur le bouton « Modifier d'une configuration »	Une redirection est faite sur le configurateur	Ok	∅	∅

22	Supprimer une configuration	Cliquer sur le bouton « Supprimer » d'une configuration et mettre « Oui » sur le message qui apparaît	La configuration est supprimée de la page	Ok	∅	∅
----	-----------------------------	---	---	----	---	---

Tableau 7 - Tests des configurations sauvegardées

6 CONCLUSION

6.1 AMÉLIORATIONS POSSIBLES

Il existe plusieurs améliorations possibles. Tout d'abord, je pense que la première amélioration serait de faire la partie « Administrateur » afin de pouvoir gérer les différents composants.

Ensuite, il serait intéressant d'intégrer un forum afin de pouvoir partager une configuration. Les autres utilisateurs pourraient donner leur avis sur la configuration.

Pour finir, on pourrait créer une IA⁸ qui après plusieurs questions posées à l'utilisateur créerait une configuration qui comblerait ses besoins.

6.2 BILAN PERSONNEL

Je suis content du résultat final au bout de ces 10 jours de TPI. Il y a eu quelques difficultés pour prévoir le temps qu'il me faudrait pour accomplir les tâches et du coup j'ai eu du retard sur quelques tâches et de l'avance sur d'autres mais malgré cela, il a été intéressant de pouvoir organiser mon temps de travail et de ne plus avoir la routine des cours. Je suis aussi satisfait d'avoir pu montrer les compétences acquises durant ses quatre dernières années.

⁸ IA (Intelligence artificielle)

7 BIBLIOGRAPHIE

7.1 CODES REPRIS

- Classe EDatabase, pris des projets École Entreprise de la 3^{ème} année
- HTML & CSS de la barre de progression du configurateur, pris du site : <https://bootsnipp.com/snippets/featured/form-process-steps>
- Fonction JS get_data(), pris des projets École Entreprise de la 3^{ème} année
- Fonction JS getUrlVar(), pris du site : <https://gist.github.com/varemenos/2531765>

7.2 SITES UTILISÉS

- Documentation JQuery : <https://jquery.com/>
- Documentation Bootstrap : <https://getbootstrap.com/>
- Site stackoverflow : <https://stackoverflow.com/>
- Site W3Schools : <https://www.w3schools.com/>
- Documentation Php : <http://www.php.net/>
- Documentation SQL : <http://sql.sh/>

7.3 AIDES REÇUES

- Mme Travnjak
- Dario Genga
- Jean-Daniel Küenzi
- Dylan Peiry

8 PLANNING

8.1 PLANNING PRÉVISIONNEL

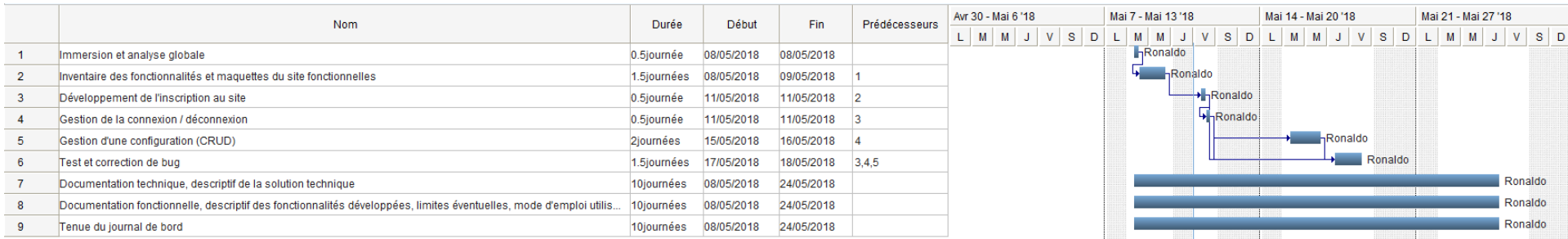


Fig. 35 - Planning prévisionnel

8.2 PLANNING EFFECTIF

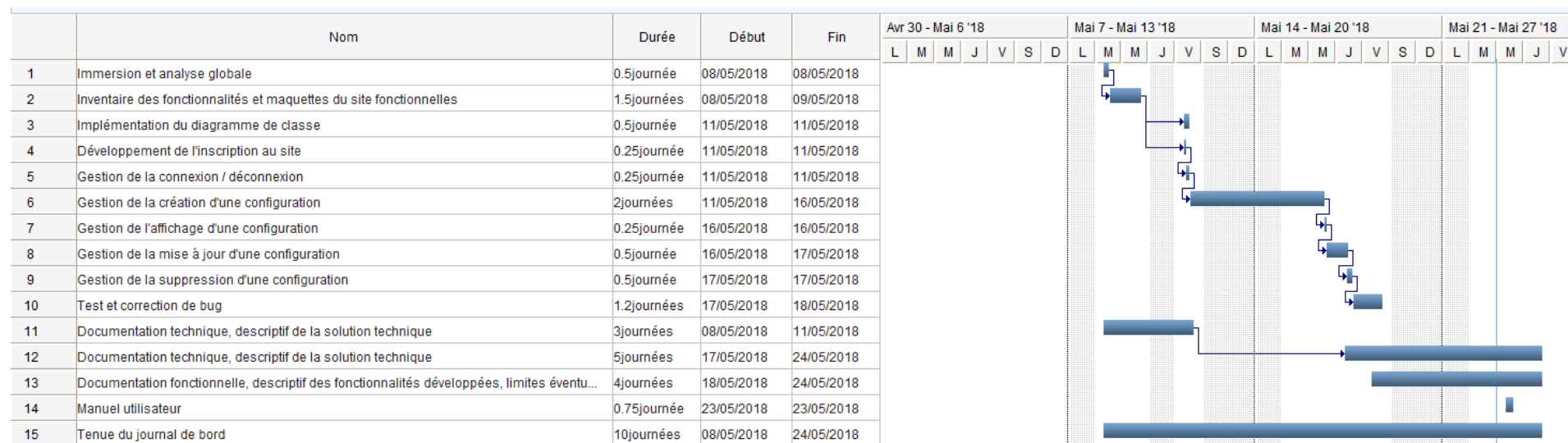


Fig. 36 - Planning effectif

Tout d'abord on peut voir que mon planning prévisionnel est moins détaillé que le planning effectif. On remarque aussi que j'avais prévu de faire de la documentation tous les jours mais en réalité j'en ai fait beaucoup au début et à la fin et que durant le développement j'en ai pas fait. J'ai aussi pris du retard sur la fonctionnalité « Gestion de la création d'une configuration », au lieu de prendre 1 jour pour la faire, j'ai pris 2 jours. J'ai pu récupérer le retard sur d'autres tâches où j'ai pris moins de temps à les faire.

9 TABLE DES ILLUSTRATIONS

9.1 FIGURES

Fig. 1 - Barre de navigation d'un utilisateur non-connecté	10
Fig. 2 - Barre de navigation d'un utilisateur connecté	10
Fig. 3 - Page d'accueil	11
Fig. 4 - Page du configurateur	12
Fig. 5 - Page Mes configurations.....	13
Fig. 6 - Pop-up d'inscription.....	14
Fig. 7 - Pop-up de connexion.....	14
Fig. 8 - Diagramme des cas d'utilisation	15
Fig. 9 - Diagramme de classes sans les liens d'association.....	18
Fig. 10 - Digramme de base de données.....	19
Fig. 11 - Table « BRANDS »	20
Fig. 12 - Table « PROCESSORS »	20
Fig. 13 - Table « SOCKETS »	20
Fig. 14 - Table « MOTHERBOARDS »	21
Fig. 15 - Table « FORMATS_MOTHERBOARD »	21
Fig. 16 Table « MOTHERBOARDS_TYPES_STORAGE »	21
Fig. 17 - Table « RAMS ».....	21
Fig. 18 - Table « TYPES_RAM »	21
Fig. 19 - Table « FREQUENCY_RAM »	22
Fig. 20 - Table « GRAPHICS_CARDS »	22
Fig. 21 - Table « CHIPSETS »	22
Fig. 22 - Table « COOLERS ».....	22
Fig. 23 - Table « COOLERS_SOCKETS »	22
Fig. 24 - Table « CASES »	23
Fig. 25 - Table « CASES_FORMATS_MOTHERBOARD »	23
Fig. 26 - Table « CASES_FORMATS_POWER ».....	23
Fig. 27 - Table « STORAGES »	23
Fig. 28 - Table « INTERFACES_STORAGE ».....	23
Fig. 29 - Table « POWERS ».....	24
Fig. 30 - Table « FORMATS_POWER »	24
Fig. 31 - Table « CONFIGURATIONS ».....	24

Fig. 32 - Table « CONFIGURATIONS_STORAGES »	24
Fig. 33 - Table « USERS »	25
Fig. 34 - Arborescence de fichier.....	25
Fig. 35 - Planning prévisionnel	43
Fig. 36 - Planning effectif.....	44

9.2 TABLEAUX

Tableau 1 - Légende des propriétés des attributs	20
Tableau 2 - Fonction de la classe « ComponentManager »	28
Tableau 3 - Fonctions de la classe « ConfigurationManager »	29
Tableau 4 - Fonction de la classe « UserManager ».....	30
Tableau 5 - Tests de connexion & inscription	36
Tableau 6 - Tests du configurateur	39
Tableau 7 - Tests des configurations sauvegardées	40

10 ANNEXES

- Code sources
- Manuel utilisateur
- Journal de bord manuscrit