

Cahier des charges

Gestion et exploitation d'une base de données contenant des informations sur les convertisseurs hautes-performances

BERTHOME Valentin	BOUR Pascal	GONDET Romain
LABORDE Camille	MAGNY Clément	MAZEREAU Christelle
	ROBERT Maxime	

11 février 2008

Table des matières

1	Introduction	3
2	Besoins fonctionnels	4
2.1	Concernant la base de donnée	4
2.1.1	Les éléments à enregistrer	4
2.1.2	Les caractéristiques des CNA/CAN	4
2.2	Concernant l'utilisation de la base de données	6
2.2.1	Accès aux fichiers	6
2.2.2	Les différents acteurs	6
2.2.3	Représentation graphique	6
2.2.4	La sélection des composants	7
2.2.5	Exportation des données	7
2.2.6	Exportation des graphiques	7
3	Besoins non fonctionnels	8
3.1	Architecture	8
3.2	Serveur	8
3.3	Gestion des conflits multi-utilisateurs	8
3.4	Gestion des documents	9
3.5	Langages de programmation	9
4	Modèle conceptuel	10
4.1	Diagramme de classe	10
4.2	Choix relatifs au diagramme de classe	11
4.2.1	La classe composant	11
4.2.2	La classe acteur	11
4.2.3	La géographie	11
4.2.4	Les documents	11
4.2.5	La relation acteur/composant	11
4.3	Cas d'utilisation	12
4.3.1	Utilisateur	12
4.3.2	Administrateur	13
5	Matériel utilisé	15
5.1	Matériel	15
5.2	Langages de programmation	15
6	Sous ensembles et priorités d'implémentation	16
7	Informations de maintenance	17
7.1	Passage de l'intranet à l'internet	17
7.1.1	Description du changement	17
7.1.2	Modifications à apporter	17
7.2	Evolution de la base de données	17
7.2.1	Description des changements	17
7.2.2	Modifications à apporter	17

Chapitre 1

Introduction

Présentation de la problématique

Le produit final est une application disponible sur un réseau intranet, utilisé par un ou plusieurs utilisateurs, permettant de stocker dans une base de donnée des caractéristiques de composants électroniques, et d'exploiter ces informations.

L'application permet d'enregistrer deux types de composants : des Convertisseurs Analogique Numérique (CAN) et des Convertisseurs Numérique Analogique (CNA). La liste des informations qui contient la base de données sera développée par la suite. Dans la mesure du possible, on prendra soin de vérifier le typage des données, et de proposer à l'utilisateur des choix prédéfinis. Il sera possible d'associer des liens ou des fichiers avec les composants. Ces liens seront soumis à un système de verrou afin de déterminer les droits des autres utilisateurs face à ce lien.

Les utilisateurs disposeront d'un ensemble de possibilités pour interagir avec l'application, comme entrer un nouveau composant dans la base, consulter des composants existants, sélectionner un ensemble de composants et exploiter les données à travers des graphiques.

L'application doit pouvoir générer plusieurs types de graphique en fonction d'une sélection d'information issue de la base de données. Le choix du type de graphique et des informations à afficher est laissé à l'utilisateur. Ces informations peuvent être issues de traitements à partir du contenu dans la base de données. Les paramétrages des graphiques seront laissés le plus libre possible. Il est également prévu de permettre d'exporter les graphiques sous différents formats afin de pouvoir être réutilisés (le format CSV et un format compatible avec Matlab et Gnuplot).

La connexion à l'application est soumise à condition : il est nécessaire d'avoir un compte actif. Deux types de comptes sont disponibles : un compte utilisateur et un compte administrateur. L'administrateur peut activer des comptes utilisateurs, ajouter des technologies et des matériaux, et se transformer en n'importe quel utilisateur enregistré afin de gérer la base de données. Un utilisateur classique peut consulter la base, y entrer un composant, supprimer ou modifier les composants dont il est l'auteur, et créer des graphiques. C'est également lui qui définit les verrous qu'il désire mettre sur les fichiers et liens associés à ses composants.

Chapitre 2

Besoins fonctionnels

L'application sera réalisée en PHP (et autres technologies web), et utilisera une base de données gérée par MySQL.

2.1 Concernant la base de donnée

2.1.1 Les éléments à enregistrer

On doit pouvoir enregistrer un ensemble de composants électroniques. Pour l'instant, on a 2 types de composants électronique :

- Les Convertisseur Numérique-Analogique (CNA). Ce sont des composants électroniques dont la fonction est de générer à partir d'une valeur numérique (codée sur plusieurs bits) une valeur analogique proportionnelle à la valeur numérique codée. Le plus souvent il s'agira de tensions électriques.
- Les Convertisseur Analogique-Numérique (CAN). Il s'agit d'un montage électronique dont la fonction est de générer à partir d'une valeur analogique, une valeur numérique (codée sur plusieurs bits), proportionnelle à la valeur analogique entrée. Le plus souvent il s'agira de tensions électriques.

2.1.2 Les caractéristiques des CNA/CAN

Un ensemble de caractéristiques doivent être enregistrées sur les CNA et CAN. Ces caractéristiques sont issues d'un document fournit par le fabricant. Chaque composant est entré par un utilisateur, et visible par tous les utilisateurs. Il est modifiable et supprimable uniquement par son propriétaire et l'administrateur du système.

A chaque composant est associé un nom d'utilisateur (son propriétaire) et une référence.

Caractéristiques idéales

Les CAN et les CNA ont des caractéristiques communes idéales, qui sont fournies par le constructeur.

Nom de la caractéristique	Type
Résolution (ou nombre de bits)	entier positif
Fréquence d'échantillonnage	réel positif
Quantum	réel positif
Bande Passante Analogique	intervalle de réel
Dynamique de sortie pour le CNA	réel positif
Dynamique d'entrée pour le CAN	réel positif
Température minimale	réel
Température maximale	réel

Caractéristiques mesurées

Les caractéristiques idéales sont globalement des valeurs théoriques. Outre ses caractéristiques, on utilise des valeurs mesurées.

Communes aux deux convertisseurs

Nom de la caractéristique	Type
Non linéarité intégrale	réel
Non linéarité Différentielle	réel
Nombre de bits effectifs	réel positif
SNR	réel positif
SFDR	réel positif
THD	réel négatif
Puissance consommée	réel positif
SINAD	réel positif

Spécifiques au CNA

Certaines mesures sont spécifiques au CNA.

Nom de la caractéristique	Type
SR	réel positif
Tn	réel positif
Te	réel positif

Caractéristiques annexes

En tant que composant électronique (qu'il s'agisse de CNA ou de CAN), on souhaite enregistrer d'autres types de caractéristiques, liées à la fabrication de composant ou à son équipe de développement.

Nom de la caractéristique	Type
Référence	chaîne de caractère alpha-numérique
Nom	chaîne de caractère alpha-numérique
La technologie	élément de l'ensemble : CMOS, MOS, BiPolaire, BiCMOS
Les matériaux	élément de l'ensemble : Si, AsGa, Ge
L'architecture	chaîne de caractère alpha-numérique
Université + équipe de développement	chaîne de caractère alpha-numérique
Entreprise	chaîne de caractère alpha-numérique
Pays	chaîne de caractère alpha-numérique
URL de l'équipe de développement	chaîne de caractère alpha-numérique
Un ou des liens vers des documents	chaîne de caractère alpha-numérique
Année + Mois	format date

La référence correspond à l'identifiant donné par le fabriquant. Le nom d'un composant est celui utilisé par les utilisateurs.

Le composant électronique utilise une technologie bien précise. L'ensemble des technologies est un ensemble fini, et seul l'administrateur sera habilité à ajouter une nouvelle technologie dans cet ensemble. Cependant, un utilisateur peut envoyer une requête à l'administrateur pour ajouter une nouvelle technologie (si le composant qu'il souhaite ajouter l'utilise).

Un composant électronique est fabriqué à partir d'un matériau. L'ensemble des matériaux est un ensemble fini, et seul l'administrateur sera habilité à ajouter un nouveau matériau dans cet ensemble. Cependant, un utilisateur peut envoyer une requête à l'administrateur

pour ajouter une nouveau materiau (si le composant qu'il souhaite ajouter l'utilise).

Un composant électronique a été développé par une équipe universitaire, ou une entreprise. S'il s'agit d'une université, on doit connaître le nom de l'université, et les noms de l'équipe de développement. S'il s'agit d'une entreprise, le nom de l'entreprise suffit. Il est aussi important de connaître le pays où se situe l'université ou l'entreprise, ainsi que l'adresse de son site internet (s'il existe), et la date de mise à disposition du composant (l'année est obligatoire, le mois est facultatif).

Les caractéristiques d'un composant électronique sont issues d'un ou plusieurs documents. Ces documents peuvent être disponible dans un format électronique. L'utilisateur qui ajoute un nouveau composant est alors libre de lier son (ses) document(s) au composant.

2.2 Concernant l'utilisation de la base de données

2.2.1 Accès aux fichiers

Les documents enregistrés dans la base de données peuvent être confidentiels, et l'utilisateur ne souhaite peut-être pas que ses documents soient visibles ou téléchargeables par les autres utilisateurs. Il existe trois types de protections (ou verrous) applicables à un document :

- Publique : la référence au document est visible sur le composant, et tous les utilisateurs peuvent consulter le document.
- Protégé : la présence du document est indiquée à tous les utilisateurs mais il n'est consultable que par le propriétaire.
- Privé : aucune référence au document n'est visible, sauf pour l'utilisateur propriétaire du document.

Pour être sûr qu'un utilisateur n'enregistre pas de document illegal dans la base de données du logiciel, l'administrateur aura un droit de regard sur les documents stockés, malgré les verrous.

2.2.2 Les différents acteurs

Il existe plusieurs types d'acteurs, avec différentes actions possibles :

- Un administrateur unique, qui peut activer, désactiver ou supprimer des comptes, ajouter un nouveau type de technologie ou de matériaux, et peut se connecter en tant que n'importe quel utilisateur (sans nécessairement connaître ses identifiants de connexion).
- Des utilisateurs : ils ont la possibilité d'ajouter des composants dans la base de données, de modifier ou supprimer leurs composants. Ils peuvent poser des verrous sur les documents liés à leurs composants, envoyer des requêtes d'ajout de technologies ou de matériaux. A partir de la base de donnée, ils ont la possibilité d'utiliser celle-ci pour générer divers graphiques, à partir des caractéristiques de composant.

Il est nécessaire d'être identifié en tant qu'utilisateur ou administrateur pour pouvoir utiliser le logiciel.

Un nouvel acteur pourra faire sa demande d'activation de compte à l'administrateur. Il renseignera alors les données utiles à la création d'un compte (nom d'utilisateur, adresse e-mail, un mot de passe). Le compte sera alors dans un état désactivé, et l'utilisateur ne pourra pas encore accéder au logiciel. L'administrateur sera averti de cette demande d'activation, et pourra changer l'état du compte.

2.2.3 Représentation graphique

Un utilisateur identifié peut utiliser le logiciel pour générer un ou plusieurs graphiques.

Les données représentées

Ces graphiques seront des représentations de données, issues de la base de données du logiciel. L'utilisateur choisira les caractéristiques qu'il souhaite utiliser, et pourra appliquer diverses fonctions mathématiques de base entre ces caractéristiques. Ces caractéristiques, modifiées ou non, serviront ensuite à générer le graphique voulu.

Le type de graphique utilisé

Les types de graphiques disponibles seront uniquement limités par les possibilités de la bibliothèque graphique utilisée par le logiciel. Il s'agira principalement de courbes, d'histogrammes, de camemberts et de graphiques libres (chaque point est représenté par des coordonnées (x, y)).

2.2.4 La sélection des composants

Les données à représenter sont issues d'une sélection de composants dans la base de données du logiciel. Cette sélection sera le résultat d'une recherche à partir de critères précis, ou d'un choix individuel de l'utilisateur. On utilise la notion de "panier" pour visualiser la sélection de composants. L'utilisateur pourra ainsi voir les composants qu'il souhaite utiliser pour générer son graphique.

2.2.5 Exportation des données

Les informations sélectionnées dans la représentation graphique peuvent être exportées dans différents formats : le format CSV, et un format compatible avec le logiciel Matlab ou Gnuplot.

2.2.6 Exportation des graphiques

Les graphiques générés pourront être enregistrés dans un format "image" par l'utilisateur.

Chapitre 3

Besoins non fonctionnels

3.1 Architecture

Le programme nécessite une architecture trois tiers composée des entités suivantes :

- Un ou des poste(s) informatique(s) muni(s) d'un navigateur Web
- Un serveur Web faisant tourner Apache
- Un serveur de base de données faisant tourner MySQL

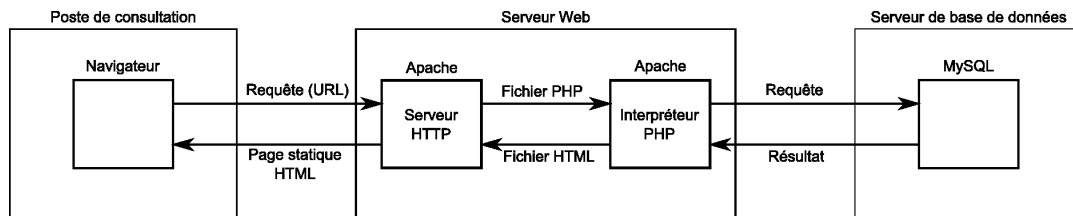


FIG. 3.1 – Architecture

3.2 Serveur

Le serveur Web et le serveur de base de données peuvent être installés sur la même machine. Les serveurs doivent disposer d'une plate-forme **LAMP** (Linux Apache MySQL PHP), **WAMP** (Windows Apache MySQL PHP) ou **MAMP** (Macintosh Apache MySQL PHP).

Dans le cas d'une utilisation mono-poste, il est envisageable que le serveur Web et le serveur de base de données soient installés sur l'unique poste de consultation.

3.3 Gestion des conflits multi-utilisateurs

Le nombre d'utilisateurs simultanés, bien que faible, peut poser des problèmes de conflit. Un composant appartenant à l'ensemble des utilisateurs (et non à son créateur) peut être modifié par plusieurs personnes en même temps. Pour résoudre ce problème, nous choisissons de verrouiller le composant dès qu'un utilisateur l'édite. Ainsi, pendant son édition, aucun autre utilisateur n'est autorisé à modifier ce composant.

De même, pour éviter que deux instances d'un même utilisateur puissent modifier simultanément un composant lui appartenant, un utilisateur ne peut pas ouvrir simultanément plusieurs sessions sur le programme. Ceci implique que lorsque l'administrateur utilise l'identité d'un des utilisateurs, l'utilisateur ne peut plus accéder au programme jusqu'à ce que l'administrateur quitte la session de l'utilisateur en question.

3.4 Gestion des documents

Les documents publics doivent être déposés sur le serveur Web afin qu'ils soient disponibles à la consultation pour tous les utilisateurs.

Les documents protégés et privés doivent être présents sur le disque dur de l'utilisateur afin qu'ils ne puissent en aucun cas être consultables par les autres utilisateurs.

Le document étant sur le disque dur de l'utilisateur, il ne sera consultable qu'à partir du poste sur lequel il est stocké. Ainsi, un changement de poste implique une non disponibilité du document. Cependant, cette solution apporte une sécurité accrue pour les documents confidentiels.

3.5 Langages de programmation

Les langages utilisés pour le programme sont :

- MySQL pour gérer la base de données
- HTML, CSS et JavaScript pour réaliser l'interface
- PHP pour réaliser le cœur du programme
- Artichow, une bibliothèque de PHP, pour construire les différents graphiques

Les logiciels devant être installés et configurés sur le serveur Web sont : Apache, PHP et MySQL. Les versions de ces logiciels gratuits et open source doivent être les plus récentes possibles et ne doivent pas être des versions bêta (versions de test) afin d'avoir une sécurité, des performances et une compatibilité optimales.

Les versions recommandées sont les suivantes (ou plus récentes) :

- Apache HTTP Server 2.2.8
- PHP 5.2.5
- MySQL 5.0.37 (avec le moteur MyISAM et InnoDB)

Chapitre 4

Modèle conceptuel

4.1 Diagramme de classe

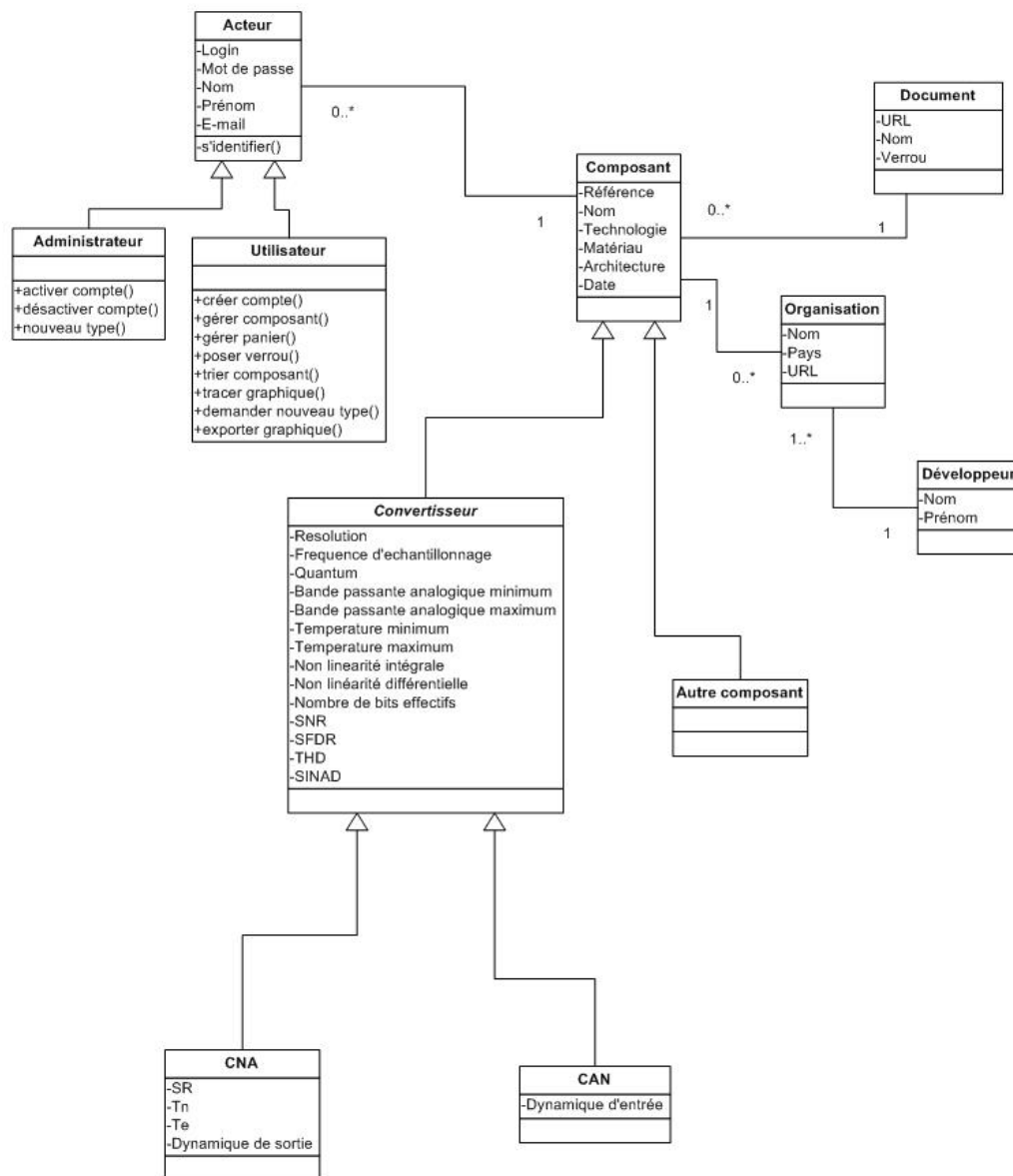


FIG. 4.1 – Diagramme de classe

4.2 Choix relatifs au diagramme de classe

Ce chapitre a pour but d'expliquer nos choix concernant le diagramme de classe ci-dessus.

4.2.1 La classe composant

Notre base de données stockera divers composants dont les convertisseurs. Néanmoins on retrouve certaines caractéristiques chez tous les composants telles que la référence, le nom ou encore la technologie utilisée. Pour représenter cela, nous utilisons la notion de hiérarchie, en définissant une classe de base composant, contenant les informations relatives à tous les composants, dont héritent deux sous-classes convertisseur et autre composant qui elles contiendront les informations relatives au type de composant qu'elles représentent. Ceci permet à la fois de factoriser l'information et de bien distinguer le type de chaque composant.

De la même façon nous définissons deux types de convertisseurs que sont le CAN et le CNA, qui représentent tout deux des sous-classes de convertisseurs et qui contiennent les informations spécifiques à leur type de convertisseur. Ainsi les informations relatives à un CNA seront stockées dans trois classes différentes : par exemple sa référence sera un attribut de la classe composant, sa résolution un attribut de convertisseur, et sa dynamique de sortie un attribut propre à sa classe.

4.2.2 La classe acteur

Deux types d'acteurs seront susceptibles d'interagir avec le système : l'administrateur et les utilisateurs. L'administrateur a seulement pour rôle de gérer les différents et de créer des nouveaux types, tandis qu'un utilisateur rentrera des informations et pourra entre autre tracer des graphiques. Bien qu'ayant des rôles différents, ces acteurs sont identifiés dans notre système de la même façon avec un login et un mot de passe ; ils ont par ailleurs les mêmes caractéristiques (nom, prénom, E-mail). Nous avons donc naturellement mis en place une nouvelle notion de hiérarchie en définissant les classes administrateur et utilisateur comme sous-classes d'acteur. Ainsi la classe acteur contiendra les informations relatives aux deux types d'acteur ainsi que la fonction "s'identifier", tandis que les sous-classes contiendront les fonctions concernant uniquement l'acteur qu'elles représentent.

4.2.3 La géographie

D'un point de vue géographique, un composant peut être lié à une entreprise ou à une université et une équipe de chercheurs. Nous avons par raison de commodité créé une classe organisation regroupant à la fois entreprise et université. En effet nous pouvons les représenter par les mêmes critères : un nom, un pays et une URL. Néanmoins lorsqu'il s'agit d'une université nous devons préciser l'équipe de chercheurs à l'origine du composant, c'est pourquoi nous avons créé la classe développeur qui contiendra leurs noms et prénoms. Cette classe est donc liée à organisation : une organisation peut compter de 1 à n développeur(s) tandis qu'un développeur n'appartient qu'à une seule organisation. La classe organisation est elle-même liée à composant : un composant ne peut appartenir qu'à une seule organisation tandis qu'une organisation peut être à l'origine de 0 à n composant(s).

4.2.4 Les documents

Nous avons vu que des documents peuvent être joints à certains composants. Etant donné qu'un composant pourrait être lié à plusieurs documents, nous avons dû créer une classe document avec ses attributs URL, nom et verrou. Un composant peut aussi n'être lié à aucun document, d'où la cardinalité 0 à n. Un document quand à lui ne référence qu'un seul composant.

4.2.5 La relation acteur/composant

Après avoir défini la hiérarchie des classes acteur et composant, nous devons nous interroger quant à la nature de la relation entre les deux classes de base. Un composant n'appartien-

dra qu'à un unique acteur qui sera celui qui l'aura référencé. En revanche un acteur pourra référencer autant de composant qu'il le souhaite, d'où la cardinalité 0 à n.

4.3 Cas d'utilisation

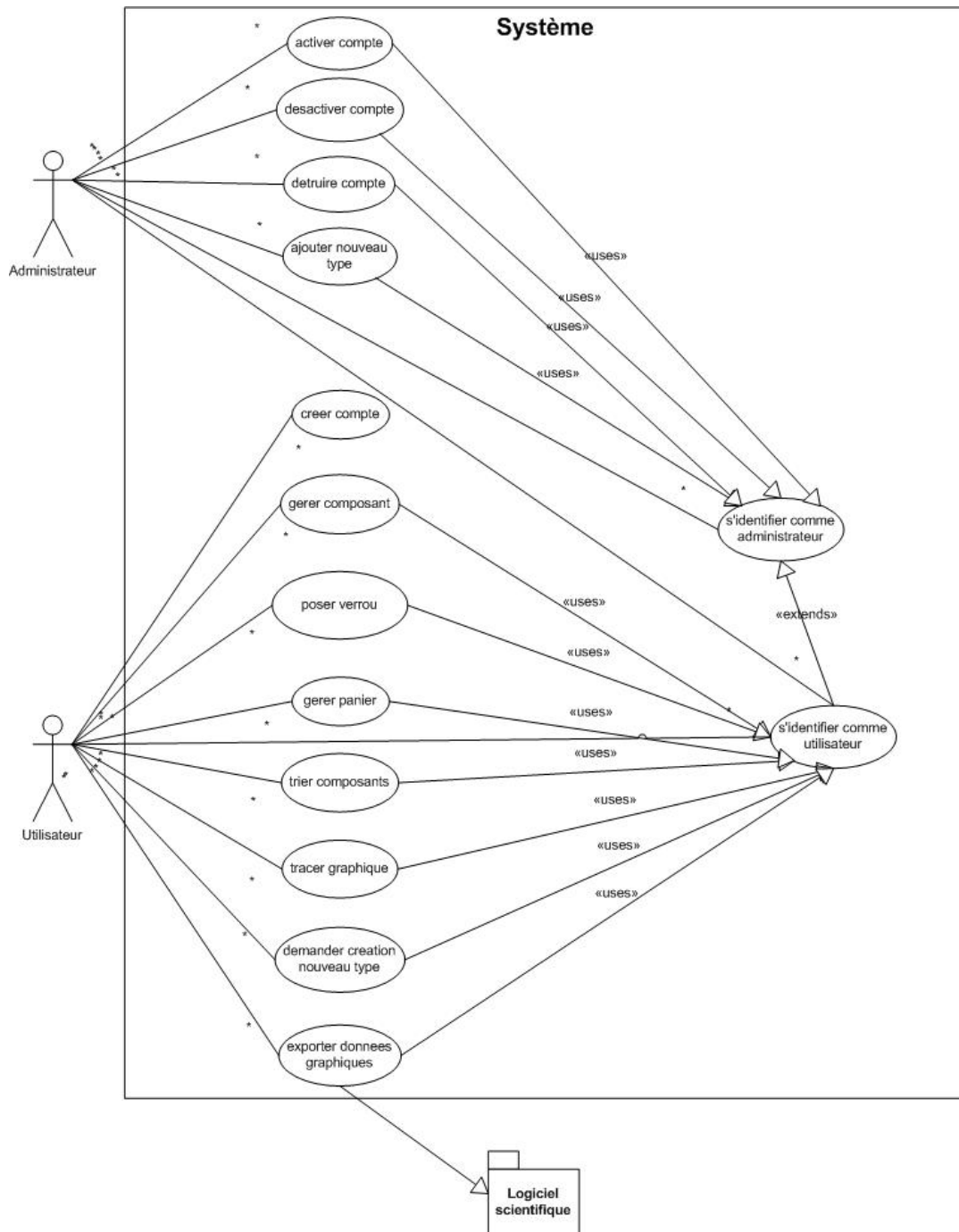


FIG. 4.2 – Cas d'utilisation

4.3.1 Utilisateur

créer compte : n'importe quelle personne accédant à la page d'accueil de l'application a la possibilité de créer son propre compte. Pour cela, l'utilisateur rentre dans les champs appropriés ses informations (nom, prénom, e-mail, login, mot de passe) et confirme. Le compte

est alors créé avec le statut désactivé et un mail est envoyé à l'administrateur : ce dernier peut alors choisir ou non d'accepter la création d'un compte par cette personne, et cela en choisissant soit d'activer le compte en question, soit de le détruire. L'utilisateur peut alors, à l'aide de son compte, accéder aux fonctionnalités du logiciel.

gérer composant : comprend la création, l'affichage, l'édition et la suppression de composants. L'affichage de tous les composants est possible, l'édition et la suppression n'est faisable que par le propriétaire du document (celui qui l'a créé).

poser verrou : permet de définir, pour le propriétaire d'un composant, la visibilité des documents attachés au composant en question. Il y a trois types de verrous : "public", définissant un document comme consultable et téléchargeable par les utilisateurs, "protégé", signalant simplement aux utilisateur l'existence du document, et enfin "privé", empêchant l'accès à toutes les informations relatives au fichier attaché, jusqu'à son existence. Bien évidemment, ces verrous ne s'appliquent pas au propriétaire du document, qui garde en toutes circonstances le contrôle sur les documents qu'il a ajouté.

gérer panier : La notion de "panier" dans notre application est la suivante : le panier représente un ensemble, en l'occurrence un ensemble de composants, qui peut être modifié, consulté et exploité à loisir par l'utilisateur. Les modifications consistent à ajouter un composant (ou un groupe de composants) que l'utilisateur choisit à partir de ceux qu'il visualise dans la zone de consultation de notre logiciel, ou encore à supprimer un élément du panier, dans la zone de consultation du panier. Le panier est personnel à l'utilisateur : il est le seul à disposer des fonctionnalités énoncées précédemment sur son panier. De plus, le contenu du panier a une durée de vie égale à la durée de la session de l'utilisateur : en d'autres termes, le panier d'un utilisateur est vidé lors de sa déconnexion. L'exploitation du panier consiste à tracer des graphiques avec pour échantillon de données le contenu du panier.

trier composants : permet d'organiser la liste des composants selon différents critères (critères encore à déterminer).

tracer graphique : permet de tracer un graphique représentatif des données présentes dans la base de données : l'utilisateur pourra choisir les données représentées en abscisse et en ordonnée ainsi que le type de graphique qu'il souhaite tracer. L'échantillon de données représenté sera sélectionné à l'aide du panier.

demander création nouveau type : lors de la création d'un nouveau composant, un utilisateur doit notamment remplir les champs "technologie" et "matériaux", et cela en choisissant une valeur parmi un ensemble de propositions. Seulement, un utilisateur peut parfois avoir besoin de rentrer une technologie ou un matériau qui ne lui est pas proposé : pour cela, lorsqu'il rentre un composant, il lui est proposé de demander la création d'un nouveau type, ce qui consiste en la rédaction d'un mail à l'administrateur afin de lui soumettre l'ajout d'une valeur. Ce dernier choisit alors d'ajouter ou non cette valeur.

exporter données graphiques : permet, à partir d'un graphique, d'exporter les données numériques s'y rapportant (coordonnées des points) afin de les traiter dans des logiciels scientifiques (Matlab, Scilab etc). On pourra notamment les exporter au format CSV et dans un format compatible avec Matlab ou Gnuplot.

4.3.2 Administrateur

activer / désactiver compte : l'administrateur a accès à un gestionnaire de compte, lui permettant de visualiser les comptes de tous les utilisateurs du logiciel. Dans ce gestionnaire, il a la possibilité d'activer / de désactiver n'importe quel compte, permettant ainsi à son propriétaire de pouvoir ou non accéder au logiciel.

détruire compte : dans un souci de cohérence des données de la base, on ne détruira jamais un compte utilisateur, et ce afin d'avoir toujours un contrôle (les droits nécessaires) sur toutes les données de la base. On fera cependant une exception, en proposant à l'administrateur la destruction d'un compte qui n'a jamais été activé, correspondant donc à un utilisateur qui n'a jamais entré de données dans la base. Cette fonction sert notamment lorsque l'administrateur souhaite rejeter la création d'un compte utilisateur.

ajouter nouveau type : permet à l'administrateur de proposer de nouvelles technologies et matériaux aux utilisateurs souhaitant créer un nouveau composant.

Chapitre 5

Matériel utilisé

5.1 Matériel

- un PC équipé
- un serveur

L'utilisateur doit posséder une machine munie d'un navigateur Web compatible avec les normes W3C. Voici une liste de quelques navigateurs compatibles :

- Firefox
- Internet Explorer
- Opera
- Safari

5.2 Langages de programmation

- MySQL sera utilisé pour gérer la base de données
- Nous utiliserons les langages HTML, CSS, PHP et JavaScript pour réaliser l'interface
- Nous utiliserons Artichow, une bibliothèque de PHP, pour construire les différents graphiques

Chapitre 6

Sous ensembles et priorités d'implémentation

Au cours du déroulement du projet, différentes versions du programme seront fournies. Les premières consisteront en une forme simple du logiciel, dans lequel toutes les fonctionnalités ne seront pas nécessairement implémentées.

De plus, le langage que nous utiliserons, à savoir le PHP, permet la programmation objet. Il est donc aisé de développer certaines classes avant les autres, pour lesquelles des classes faussaires pourront éventuellement être mises en place. Ainsi, une version sera vue en fonctionnement dans son ensemble, sans que toute l'application ait pour autant été développée.

Planning prévisionnel

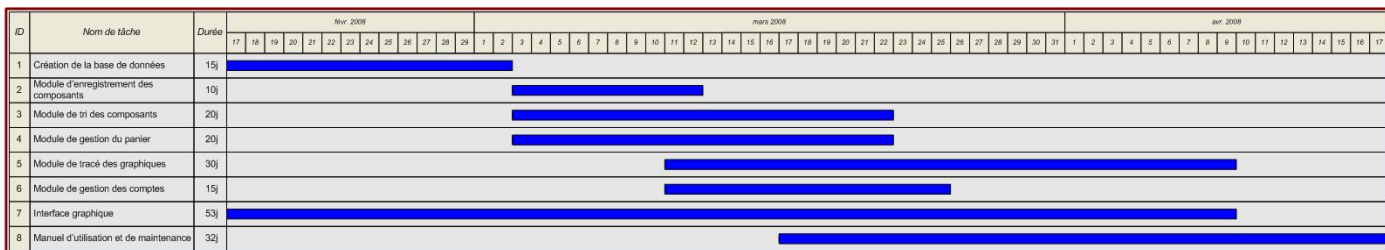


FIG. 6.1 – Diagramme de Gantt

Chapitre 7

Informations de maintenance

7.1 Passage de l'intranet à l'internet

7.1.1 Description du changement

Le logiciel est destiné à être utilisé sur un réseau interne non-visible de l'extérieur. Cependant, il est possible, que par la suite, les utilisateurs internes aient envie de pouvoir avoir un accès externe au logiciel afin de ne pas être obligé de se connecter de façon interne pour retracer un graphique ou travailler sur des données. Ainsi, il faut que l'utilisation du logiciel se fasse par des utilisateurs internes mais qu'il soit possible de le faire depuis l'extérieur.

7.1.2 Modifications à apporter

Notre système nécessite déjà obligatoirement d'être authentifié comme utilisateur pour accéder au contenu et il faut nécessairement passer par l'administrateur pour obtenir un compte, le problème est alors de n'autoriser la création d'un compte que pour des utilisateurs internes au réseau. En faisant cela on peut s'assurer que seuls les utilisateurs internes ont la possibilité d'utiliser le logiciel. Pour cela, il faudra interdire la création d'un compte pour des utilisateurs externes au réseau ce qui peut se faire par une vérification d'adresse IP. Ainsi, pour créer un compte il faut se situer dans l'entreprise mais une fois la création de compte effectuée un utilisateur a la possibilité d'y accéder depuis n'importe quel endroit et avec n'importe quel ordinateur connecté à internet.

7.2 Evolution de la base de données

7.2.1 Description des changements

La base de données doit pouvoir accueillir de nouveaux composants électroniques ormis les Convertisseurs Analogiques Numériques (CAN) et Convertisseurs Numériques Analogiques (CNA). la base de données doit pouvoir accueillir de nouveaux composants sans nécessiter un remaniement complet et le logiciel également.

7.2.2 Modifications à apporter

La base de données est prévu pour l'ajout de composants autres que le CNA et le CAN, les seules données en commun à tous les composants sont les données annexes. Il faut ajouter l'élément dans la base de données avec les caractéristiques voulues et un lien entre notre table de données annexes et ce composant. De plus, les formulaires doivent être prévu de manière à factoriser au maximum l'implémentation notamment entre les données obligatoires et les données facultatives. Il en est de même pour l'affichage des caractéristiques d'un composant.

Glossaire

- **LAMP** ("Linux Apache MySQL PHP") : architecture Web basée sur Apache, MySQL et PHP fonctionnant sur une plateforme Linux
- **MAMP** ("Macintosh Apache MySQL PHP") : architecture Web basée sur Apache, MySQL et PHP fonctionnant sur une plateforme Macintosh
- **WAMP** ("Windows Apache MySQL PHP") : architecture Web basée sur Apache, MySQL et PHP fonctionnant sur une plateforme Windows