

UNIVERSITE SIDI MOHAMED BEN ABDELLAH
FACULTE DES SCIENCES ET TECHNIQUES FES
DEPARTEMENT D'INFORMATIQUE



Projet de Fin d'Etudes

Licence Sciences et Techniques Génie Informatique

Développement d'une application web Java EE pour le calcul de Headroom d'une combinaison



Lieu de stage : YAZAKI Morocco S.A, Tanger

Réalisé par :

EL MORABET Hicham
EL FALLAQY Youssef

Encadré par :

Pr. BENABBOU Abderrahim
M. ELADDALI Karim

Soutenu le 09/06/2015 devant le jury composé de :

Pr. MRABTI Fatiha
Pr. ZENKOUAR Khalid
Pr. BENABBOU Abderrahim

Année Universitaire 2015-2016

Remerciement

Après ALLAH, nous tenons à remercier :

En premier lieu nos parents qui nous ont soutenus moralement et matériellement tout au long de notre vie. Nous espérons avant tout que nous serons toujours votre fierté. Que DIEU vous bénisse.

Nos valeureux encadrants Monsieur Abderrahim BENABBO à la Faculté des Sciences et Techniques Fès et Monsieur Karim ELADDALI développeur software au sein du service IT à YAZAKI Tanger pour leurs conseils fructueux, leurs directives et leurs encouragements qui nous ont si généreusement aidés durant toute la période du stage.

Tout le personnel de YAZAKI spécialement Monsieur Karim MAHFOUD responsable COSEE sertissage à YAZAKI Europe et Monsieur Fouad FILALI Ingénieur au département Ingénierie et Process pour avoir nous accordé leur temps précieux, leur attention et leur énergie pour nous aider dans la réalisation de ce travail.

Enfin, toutes personnes qui ont contribué et collaboré de près ou de loin à la réalisation de ce modeste travail.

Table des matières

Résumé.....	8
Abstract.....	9
Introduction générale.....	10
Chapitre I. Présentation générale.....	11
I. Organisme d'accueil.....	12
1. YAZAKI.....	12
1.1. Historique du groupe YAZAKI.....	12
1.2. Activités de YAZAKI.....	13
1.3. Processus de délocalisation.....	13
2. YAZAKI Maroc.....	14
2.1. Création de YAZAKI Maroc.....	14
2.2. Activité de YAZAKI Maroc.....	15
2.3. Organigramme général.....	15
II. Présentation du projet.....	16
1. Problématique.....	16
2. Cahier de charge.....	17
3. Solution proposée.....	18
Chapitre II. Analyse des besoins et conception.....	19
I. Méthodologie du développement.....	20
1. Langage de modélisation.....	20
1.1. Définition.....	20
2. Méthodologie adoptée.....	21
2.1. Définition.....	21
2.2. Itératif et incrémental.....	22
2.3. Phases du RUP.....	22
II. Analyse.....	23
1. Identification des besoins.....	23

1.1. Besoins fonctionnels.....	23
1.2. Besoins non fonctionnels.....	24
2. Acteurs du système.....	25
3. Messages émis et reçu par le système.....	26
3.1. Messages envoyés par « l'Administrateur ».....	26
3.2. Messages envoyés par « l'Employé ».....	27
4. Cas d'utilisation par acteur.....	28
4.1. Diagramme associé à « l'Administrateur ».....	28
4.2. Diagramme associé à « l'Employé ».....	29
5. Description textuelle des cas d'utilisation.....	30
5.1. Description textuelle des cas d'utilisation de « l'Administrateur ».....	30
5.2. Description textuelle des cas d'utilisation de « l'Employé ».....	32
III. Conception.....	33
1. Diagramme de classes « modèle statique ».....	33
2. Diagrammes de séquences « modèle dynamique ».....	35
2.1. Diagramme de séquence « calculer le Headroom ».....	35
2.2. Diagramme de séquence « modifier une combinaison ».....	36
2.3. Diagramme de séquence « créer un compte ».....	37
2.4. Diagramme de séquence « modifier un compte ».....	38
3. La base de données de l'application.....	39
Chapitre III. Mise en place de l'application.....	40
I. Langages et technologies de développement.....	41
1. Plate-forme Java EE.....	41
1.1. Définition.....	41
1.2. Pourquoi Java EE.....	42
2. Framework JSF.....	43
3. HTML.....	43
4. CSS.....	43
5. JavaScript.....	44
6. JQuery.....	44
7. MySQL.....	44
II. Environnement de réalisation.....	45

1. Environnement matériel.....	45
2. Environnement logiciel.....	45
2.1. Apache TomEE.....	45
2.2. NetBeans IDE.....	46
2.3. Entreprise Architect.....	46
III. Présentation de l'application.....	46
1. Prototype de l'interface « Authentification ».....	46
2. Prototype de l'interface « calculer le Headroom ».....	48
3. Prototype de l'interface « gestion des combinaisons ».....	48
4. Prototype de l'interface « créer un compte ».....	51
5. Prototype de l'interface « gestion des comptes ».....	52
6. Prototype de l'interface « gestion des machines ».....	52
7. Prototype de l'interface « contacter un utilisateur ».....	53
8. Prototype de l'interface « contacter l'administrateur ».....	53
Conclusion générale.....	54
Annexes.....	55
Bibliographie.....	59
Webographie.....	59

Liste des figures

Figure 1 : logo de YAZAKI	12
Figure 2 : YAZAKI à travers le monde	13
Figure 3 : YAZAKI Maroc, Tanger	14
Figure 4 : câblage fini	15
Figure 5 : organigramme de la société	15
Figure 6 : Interface du fichier YEL-CE-S-211	16
Figure 7 : schéma de la méthode RUP	21
Figure 8 : les incréments du modèle incrémental itératif	22
Figure 9 : diagramme de cas d'utilisation associé à « l'Administrateur »	28
Figure 10 : diagramme de cas d'utilisation associé à « l'Employé »	29
Figure 11 : diagramme de classes	34
Figure 12 : diagramme de séquence « Calculer le Headroom »	35
Figure 13 : diagramme de séquence « Modifier une combinaison »	36
Figure 14 : diagramme de séquence « Créer un compte »	37
Figure 15 : diagramme de séquence « Modifier un compte »	38
Figure 16 : schéma de la base de données	39
Figure 17 : interface « authentification »	48
Figure 18 : interface « calculer le Headroom »	49
Figure 19 : interface « my combinations »	50

Figure 20 : interface « edit combination »	50
Figure 21 : interface « all combinations »	51
Figure 22 : rapport de combinaison (PDF)	51
Figure 23 : rapport de combinaison (Excel)	51
Figure 24 : interface « add user »	52
Figure 25 : interface « manage users »	53
Figure 26 : interface « manage machines »	53
Figure 27 : interface « contact users »	54
Figure 28 : interface « contact admin »	54

Liste des tableaux

Tableau 1 : messages envoyés au système par « l'Administrateur »	27
Tableau 2 : messages envoyés au système par « l'Employé »	28
Tableau 3 : description textuelle de cas d'utilisation « S'authentifier »	31
Tableau 4 : description textuelle de cas d'utilisation « Créer un compte »	32
Tableau 5 : description textuelle de cas d'utilisation « Modifier un compte »	32
Tableau 6 : description textuelle de cas d'utilisation « Contacter un utilisateur »	33
Tableau 7 : description textuelle de cas d'utilisation « Calculer le Headroom »	33
Tableau 8 : description textuelle de cas d'utilisation « Modifier une combinaison »	34
Tableau 9 : Caractéristique poste de travail	46

Résumé

Nous avons effectué notre stage de fin d'études au sein de YAZAKI Morocco qui est une Société de câblage d'automobile appartenant au groupe YAZAKI installée la zone franche de Tanger.

Ce stage était destiné à la réalisation d'une application web en Java EE pour le calcul de Headroom d'une combinaison. Nous avons réalisé l'application avec la plate-forme Java EE (Java Enterprise Edition) en utilisant MySQL comme un SGBD (Système Gestion de Base de Données).

Mots clés : Java EE,MySQL,combinaison, Headroom, calcul de Headroom.

Abstract

We have done my end of studies internship within YAZAKI Morocco which is an automotive wiring company belonging to the YAZAKI Group installed in Tangier free zone.

This training was intended for the realization of a Java EE web application to calculate the Headroom for a combination. We realized the application with the Java EE platform (Java Enterprise Edition) using MySQL as a DBMS (Database Management System).

Keywords : Java EE, MySQL, combination, Headroom, calculate the Headroom

Introduction générale

Au terme de nos études à la faculté des Sciences et Techniques Fès, et dans le cadre de l'obtention de la licence Science et Technique - Génie Informatique, nous sommes amenés à effectuer un stage de fin d'études de deux mois afin de parfaire nos connaissances acquises tout au long de cette année et de les mettre en application dans le milieu professionnel.

Ce rapport est le fruit d'un stage qui s'est déroulé au sein de la multinationale japonaise YAZAKI à Tanger de la période du 04 Avril au 04 Juin 2016. Notre travail au cours du stage porte sur la réalisation d'une application en Java EE et utilisant MySQL comme système de gestion de base de données. L'application a pour but de calculer l'indicateur « Headroom » pour une combinaison donnée.

Nous présentons à travers ce rapport trois chapitres principaux :

Le premier chapitre définit le cadre général de notre projet, il présente en premier lieu l'organisme d'accueil, puis une description du projet.

Le deuxième chapitre se résume dans l'analyse fonctionnelle du projet en spécifiant les fonctionnalités de l'application ainsi qu'une étude conceptuelle comprenant les différents diagrammes d'UML.

Le troisième chapitre décrit l'environnement technique du projet, les outils et les langages de programmation et de modélisation utilisés pour la réalisation et la présentation de l'application.

Chapitre I.

Présentation générale

Chapitre 1

Présentation générale

Introduction

Ce chapitre a pour objectif de situer le projet dans son contexte général. Pour ce faire, nous présentons dans un premier lieu l'organisme d'accueil, ensuite, nous décrivons minutieusement le projet en dégageant les problèmes en vue de préciser les objectifs.

I. Organisme d'accueil

1. YAZAKI

1.1. Historique du groupe YAZAKI

YAZAKI [1] est l'un des plus grands producteurs du monde de câblage automobile et un joueur dans la fabrication des systèmes de la distribution électrique et électroniques, l'instrumentation électronique et les composants pour les voitures. La date de succès de YAZAKI était en 1929, quand SADAMI YAZAKI a commencé à vendre les câblages pour les automobiles. En 1939, l'affaire pourrait être étendue et en 1941, YAZAKI Fil Electricque Industriel Co. Ltd a été établi avec approximativement 70 employés.



Figure 1 : logo de YAZAKI

1.2. Activités de YAZAKI

Le groupe japonais [1] dont le siège est basé à Tokyo, son activité principale est le câblage, la fabrication de composants électriques pour automobiles et instruments. Ses autres activités sont la fabrication des fils et câbles électriques, la fabrication des produits de gaz et la climatisation.

1.3. Processus de délocalisation

Le processus de délocalisation de la société a commencé en 1962 avec sa filiale, THAI YAZAKI ELECTRIC WIRE CO. LTD.

Le groupe [1] est localisé dans 45 pays, il compte à son actif :

- 173 sociétés.
- 478 unités réparties entre usines de production et centres de service au client et centres de Recherche & Développement.
- 284 200 d'employés.

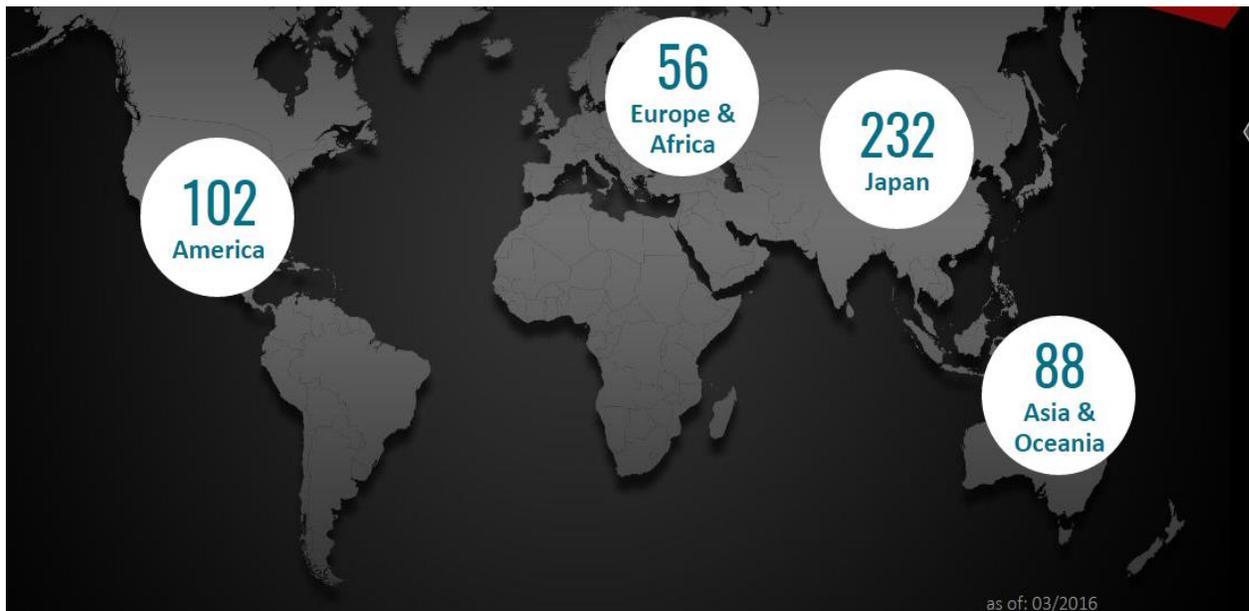


Figure 2 : YAZAKI à travers le monde [1]

2. YAZAKI Maroc

2.1. Création de YAZAKI Maroc

Le processus de délocalisation de la société s'est poursuivi par la création, en Octobre 2000, d'une unité de Production à Tanger au Maroc, sous la dénomination de « YAZAKI SALTANO DE PORTUGAL », Succursale MAROC.

En 2001, le Maroc a été le premier pays africain auquel Mr. YAZAKI a fait honneur, par l'inauguration de son site opérationnel YAZAKI MAROC (YMO) pour la production des câbles automobile, en présence de SM le ROI MOHAMMED VI. Ce n'est qu'en mai 2003 qu'elle acquiert son indépendance et devient YAZAKI MOROCCO. Avec un capital de 88 millions de DH, elle a accumulé un investissement de plus de 253 millions de DH.

YAZAKI MOROCCO a réalisé un chiffre d'affaires de 150 millions d'euros en 2010 pour un total d'emplois supérieur à 4900 postes. [2]



Figure 3 : YAZAKI Maroc, Tanger

2.2. Activité de YAZAKI Maroc

L'activité de YAZAKI Maroc est la conception des câbles électriques qui servent à connecter des différents éléments dans un système électromécanique et de fournir de l'énergie électrique et des signaux électronique à différents périphériques du système.

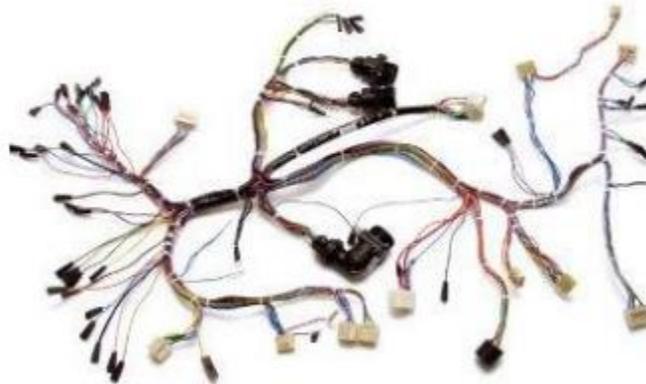


Figure 4 : câblage fini

2.3. Organigramme général

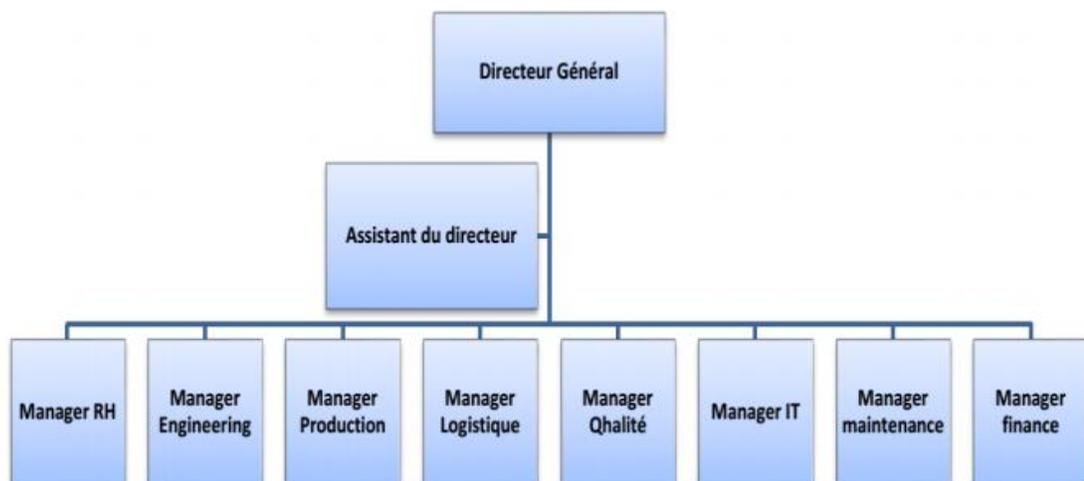


Figure 5 : organigramme de la société

II. Présentation du projet

1. Problématique

À chaque lancement de nouveau projet ou bien en relation avec les changements techniques des projets en cours, le service Crimping Process de chaque usine de YAZAKI Europe (19 usines) est toujours amené à faire un test de validation pour les nouvelles combinaisons par le calcul de leur indicateur Headroom.

Pour faire le calcul, les employés du service Crimping Process ont besoin de télécharger la dernière version du fichier YEL-CE-S-211 de E-Matrix et vérifier si la combinaison est déjà disponible dans le fichier (le Headroom de cette combinaison est déjà calculé dans une autre usine) les employés prennent seulement les résultats de calcul, sinon ils calculent le Headroom et enregistrent la combinaison avec ses résultats dans le même fichier.

Le fichier YEL-CE-S-211 est une application VBA (Visual Basic for Applications) faite en Excel.

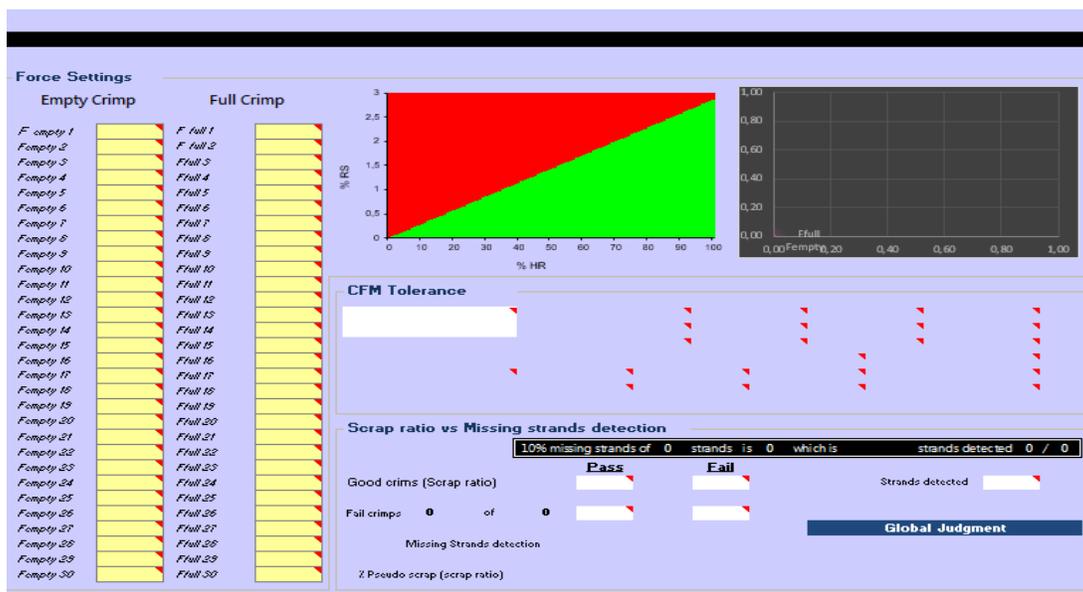


Figure 6 : Interface du fichier YEL-CE-S-211

Les employés ont envisagé plusieurs problèmes utilisant ce fichier pour le calcul et pour l'enregistrement des combinaisons et leurs informations :

- Chaque jour l'employé doit télécharger le fichier, faire les calculs et le charger, ce qui prend beaucoup de temps.
- Excel devient très lent avec l'ajout des centaines des combinaisons chaque jour.
- L'employé doit vérifier la liste des combinaison enregistrées dans le fichier avant le calcul de Headroom de chaque combinaison.
- Les informations enregistrées dans le fichier sont confidentielles et doivent être sécurisées, ce qui n'est pas le cas avec cette méthode.

2. Cahier de charge

Au but de calcul de Headroom d'une combinaison au sein du service Crimping Process :

« L'Employé » fait le calcul de Headroom en saisissant les valeurs de Crimping Full et Empty, ainsi que les informations de la combinaison, la machine, l'applicateur et CFM Tolerance.

Il peut en plus consulter tous combinaisons enregistrées dans base de système, ainsi la génération de rapport d'une combinaison (sous forme PDF ou Excel), comme il peut modifier et supprimer les combinaisons qu'il a enregistré lui-même.

« L'Administrateur » gère les comptes utilisateurs de l'application, il ajoute de nouveaux comptes, comme il peut consulter, activer, désactiver, modifier et supprimer les comptes existants.

En plus, il gère les machines (Press, CFM, Applicateurs ...) et les paramètres de combinaison (Wire Size et Strands).

3. Solution proposée

Nous avons proposé une application web Java EE en utilisant MySQL comme système de gestion de base de données.

L'application a pour but :

- ✓ Calculer l'indicateur « Headroom » pour une combinaison donnée
- ✓ Meilleure gestion des combinaisons
- ✓ Garantir la sécurité des données
- ✓ Communication entre les utilisateurs de l'application

Chapitre II.

Analyse des besoins et conception

Chapitre 2

Analyse des besoins et conception

Introduction

Dans ce chapitre nous allons présenter l'étude technique de notre projet. L'analyse des données sera présentée en premier lieu suivie d'une conception, vu que l'étape de la réalisation n'est qu'un fruit d'une bonne analyse et conception, ceux-ci s'avèrent donc primordiaux et fondamentaux dans la totalité du travail du projet.

I. Méthodologie du développement

1. Langage de modélisation

Nous avons choisi comme langage de modélisation UML (*Unified Modeling Language*).

1.1. Définition

UML [3] est une méthode de modélisation orientée objet développée en réponse à l'appel à propositions lancé par l'OMG (Object Management Group) dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets.



UML est utilisée pour spécifier un logiciel et/ou pour concevoir un logiciel. Dans la spécification, le modèle décrit les classes et les cas d'utilisation vus de l'utilisateur final du logiciel. Le modèle produit par une conception orientée objet est en général une extension du modèle issu de la spécification. Il enrichit ce dernier de classes, dites techniques, qui n'intéressent pas l'utilisateur final du logiciel mais seulement ses concepteurs. Il comprend

les modèles des classes, des états et d'interaction. UML est également utilisée dans les phases terminales du développement avec les modèles de réalisation et de déploiement.

2. Méthodologie adoptée

Parmi les méthodologies du développement qui existent et en coordination avec notre projet, nous avons trouvé que le RUP est la méthodologie de travail qui peut nous orienter tout au long du projet.

2.1. Définition

Le processus RUP (*Rational Unified Process*) est un processus de développement logiciel itératif et incrémental, centré sur l'architecteur, piloté par les cas d'utilisation et par les risques.

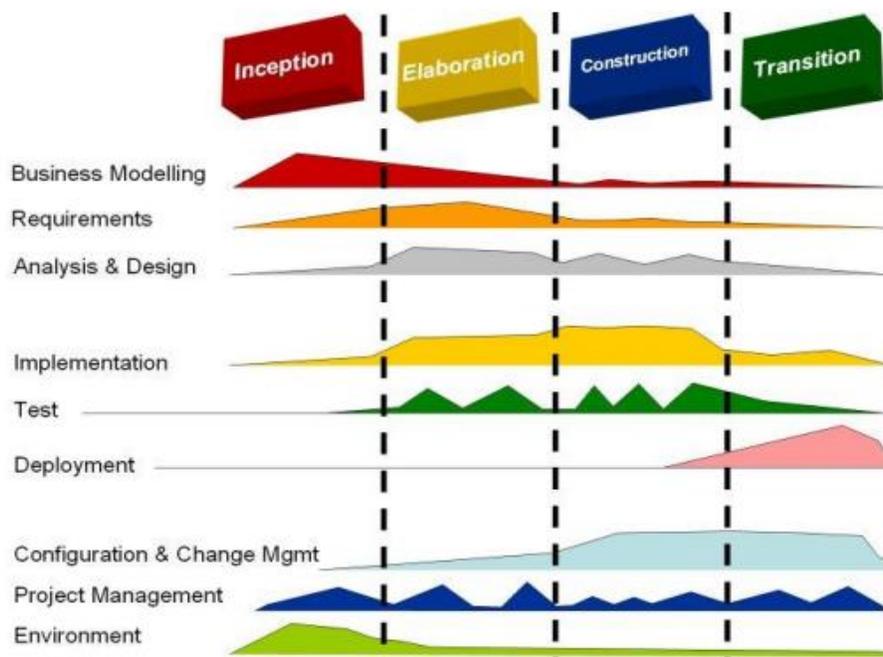


Figure 7 : schéma de la méthode RUP

2.2. Itératif et incrémental

Notre projet a été découpé en itérations de courte durée qui nous ont aidé à mieux suivre l'avancement de projet. À la fin de chaque itération une partie exécutable du système final est produit de façon incrémentale.

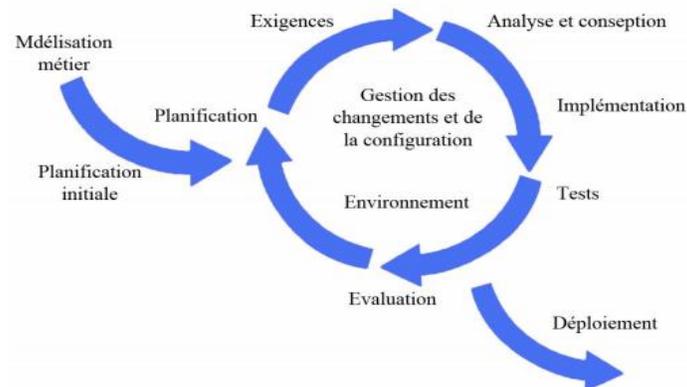


Figure 8 : les incréments du modèle incrémental itératif

2.3. Phases du RUP

Les quatre phases de cycle de vie du RUP :

- ✓ **Phase d'incubation** : Définir le champ d'action de projet (spécification des besoins et présenter l'architecteur de système).
- ✓ **Phase d'élaboration** : On développe de façon incrémentale l'architecture du noyau, préciser la plupart des cas d'utilisateurs afin de présenter l'architecture de système sous forme de vue de présentation pour chaque modèle.
- ✓ **Phase de construction** : C'est la phase de réalisation de produit, il contient en fin tous les cas d'utilisation.
- ✓ **Phase de transition** : Le produit final est livré à la disposition des utilisateurs. Les activités comme les formations et la correction des anomalies seront supposées dans cette phase.

II. Analyse

1. Identification des besoins

L'objectif principal d'un système logiciel est de rendre service à ses utilisateurs, il faut par conséquent bien comprendre les désirs et les besoins des futurs utilisateurs. Donc dans ce qui suit, nous commencerons par énumérer les besoins fonctionnels et non fonctionnels.

1.1. Besoins fonctionnels

Les besoins fonctionnels sont définis comme étant des services attendus par l'utilisateur de produit. Ils doivent constituer un ensemble complet et cohérent. Dans ce qui suit, nous allons présenter les besoins fonctionnels ébauchés durant la proposition de la solution.

- **Authentification** : pour l'accès aux services basé sur le profil de l'utilisateur.
- **Calcul de Headroom d'une combinaison** : le calcul d'indicateur « *Headroom* » pour la validation d'une combinaison.
- **Gestion des combinaisons** : la modification, la suppression et la consultation des combinaisons existants.
- **Génération des rapports** : l'utilisateur peut générer un rapport pour chaque combinaison avec la possibilité de le télécharger sous forme PDF ou Excel.
- **Gestion des comptes utilisateurs** : La création d'un nouveau compte utilisateur ainsi que l'activation, la désactivation, la modification, la suppression et la consultation des comptes existants ne sont faits que par l'administrateur système.
- **Gestion des machines** : l'administrateur système a la possibilité d'ajouter des nouvelles machines (machine Press, machine CFM et applicateur) ainsi que la suppression des machines existantes.

- **Gestion de paramètres de combinaison** : l'administrateur système a la possibilité d'ajouter des nouveaux paramètres de combinaison (Wire Size et Strands) ainsi que la suppression des paramètres existants.
- **Communication entre les utilisateurs** : les utilisateurs peuvent communiquer entre eux à travers l'application.
- **Contacteur l'administrateur** : les utilisateurs peuvent envoyer un email à l'administrateur à travers l'application.

1.2. Besoins non fonctionnels

Les besoins non fonctionnels se basent sur le respect des normes de l'ergonomie et des interactions homme/machine qu'on fournit à l'application. Notre application doit nécessairement assurer ces besoins :

- ✚ **La rapidité de traitement** : le système doit garantir la rapidité d'exécution des traitements.
- ✚ **La fiabilité** : le système doit exécuter correctement toutes ses structures, pour répondre convenablement aux besoins de l'utilisateur.
- ✚ **L'ergonomie** : les interfaces doivent être claires et bien structurées. A ce propos un thème sera choisi et utilisé au cours de développement de l'application pour assurer le bon choix de design.
- ✚ **L'intégrité** : les fonctionnalités offertes à chaque utilisateur doivent être restreintes à celles qui lui sont autorisées. L'information n'est modifiée que par les personnes y ayant droit. Dans ce cas, nous définissons pour chaque utilisateur ces droits d'accès au système.

- ✚ **La portabilité** : il s'agit de minimiser l'effort pour se faire transporter dans un autre environnement matériel et/ou logiciel.

- ✚ **L'extensibilité** : l'application devra être extensible, c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

- ✚ **La sécurité** : le système doit traiter les failles de sécurité d'où le besoin d'un login et d'un mot de passe pour accéder au système. Les messages d'erreurs doivent identifier tous les cas d'erreurs de saisie et leur source.

2. Acteurs du système

Nous parvenons à une étape clé du processus. C'est elle qui grâce à l'étude réalisée dans la partie précédente mettra en valeur le rôle de chaque acteur du système ainsi que les fonctionnalités présentées plus haut.

Dans cette partie nous allons énumérer les différents acteurs susceptibles d'interagir avec le système.

❖ **Administrateur** : ces rôles sont résumés comme décrit ci-dessous :

- S'authentifier
- Créer un compte
- Modifier un compte
- Supprimer un compte
- Activer/désactiver un compte
- Consulter compte
- Ajouter une machine
- Supprimer une machine

- Ajouter un paramètre de combinaison
- Supprimer un paramètre de combinaison
- Contacter un utilisateur

❖ **Employé** : (*Process Owner*) ces rôles sont résumés comme décrit ci-dessous :

- S'authentifier
- Calculer le Headroom d'une combinaison
- Modifier une combinaison
- Supprimer une combinaison
- Consulter une combinaison
- Générer le rapport d'une combinaison
- Contacter un utilisateur
- Contacter l'administrateur

3. Messages émis et reçus par le système

Nous allons détailler les différents messages permettant de décrire les interactions de plus haut niveau entre les acteurs et le système.

3.1. Messages envoyés par « l'Administrateur »

Messages envoyés au système	Messages émis par le système
Authentification	- Espace administrateur - Message d'erreur
Créer un compte	- Message succès - Message d'erreur
Modifier un compte	- Message succès - Message d'erreur

Supprimer un compte	- Confirmation de suppression
Consulter un compte	- Liste des comptes utilisateurs
Activer/désactiver un compte	- Pas de message
Ajouter une machine	- Message succès - Message d'erreur
Supprimer une machine	- Confirmation de suppression
Ajouter un paramètre de combinaison	- Message succès - Message d'erreur
Supprimer un paramètre de combinaison	- Confirmation de suppression
Contacteur un utilisateur	- Pas de message

Tableau 1 : messages envoyés au système par « l'Administrateur »

3.2. Messages envoyés par « l'Employé »

Messages envoyés au système	Messages émis par le système
Authentification	- Espace administrateur - Message d'erreur
Calculer le Headroom d'une combinaison	- Résultats
Modifier une combinaison	- Message succès - Message d'erreur
Supprimer une combinaison	- Confirmation de suppression
Consulter une combinaison	- Liste des combinaisons
Générer le rapport d'une combinaison	- Rapport d'une combinaison
Contacteur un utilisateur	- Pas de message
Contacteur l'administrateur	- Message succès - Message d'erreur

Tableau 2 : messages envoyés au système par « l'Employé »

4. Cas d'utilisation par acteur

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel du système logiciel.

4.1. Diagramme associé à « l'Administrateur »

Le diagramme ci-dessous présente le diagramme de cas d'utilisation décrivant les différentes fonctionnalités que le « l'Administrateur » peut les accomplir.

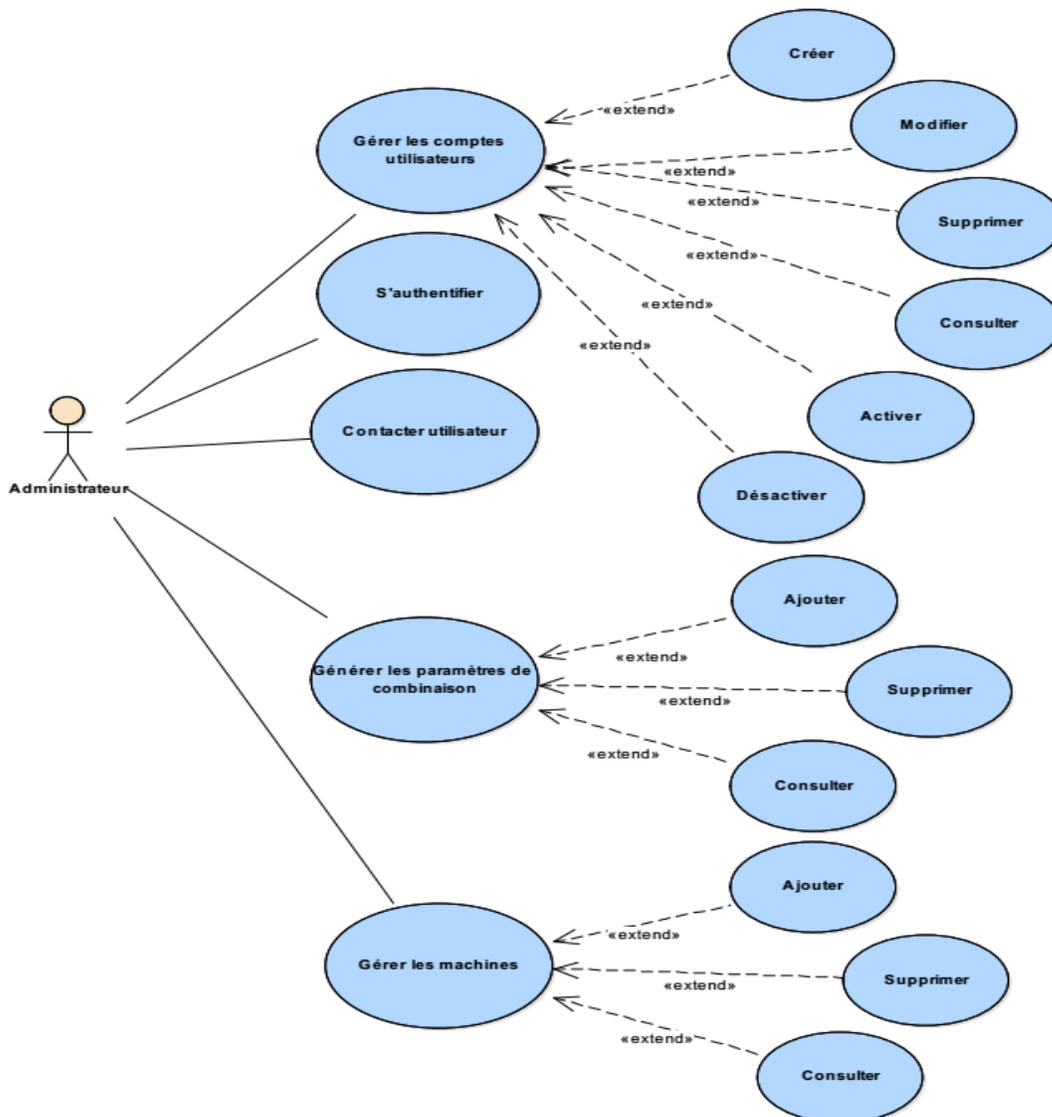


Figure 9 : diagramme de cas d'utilisation associé à « l'Administrateur »

4.2. Diagramme associé à « l'Employé »

Le diagramme ci-dessous présente le diagramme de cas d'utilisation décrivant les différentes fonctionnalités que le « l'Employé » peut les accomplir.

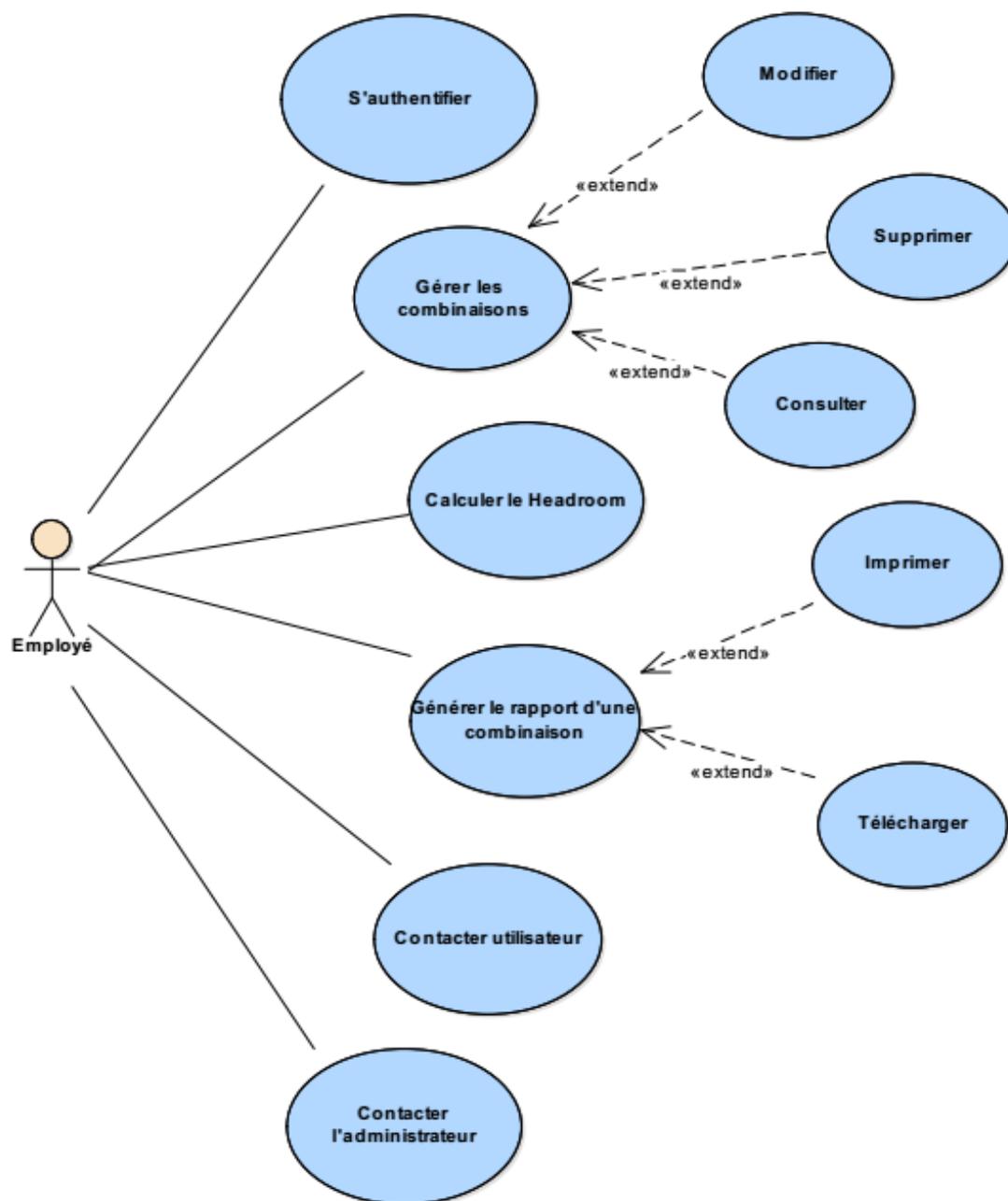


Figure 10 : diagramme de cas d'utilisation associé à « l'Employé »

5. Description textuelle des cas d'utilisation

La description textuelle de cas d'utilisation [4] consiste à recenser toutes les interactions entre le système et les acteurs de façon textuelle. Le cas d'utilisation doit avoir un début et une fin clairement identifiés. Il doit préciser quand ont lieu les interactions entre acteurs et système, et quels sont les messages échangés. Il faut également préciser les variantes possibles, telles que les différents cas nominaux, les cas alternatifs, les cas d'erreurs, tout en essayant d'ordonner séquentiellement les descriptions, afin d'améliorer leur lisibilité. Chaque unité de description de séquences d'actions est appelée enchaînement. Un scénario représente une succession particulière d'enchaînements, qui s'exécute du début à la fin du cas d'utilisation.

Dans cette partie nous allons limiter aux cas d'utilisation importantes.

5.1. Description textuelle des cas d'utilisation de « l'Administrateur »

❖ *Description textuelle* : s'authentifier (Administrateur, Employé)

Titre	Authentification	
Objectif	Être connue par le système	
Acteur	Administrateur / Employé	
Pré-condition		Post-condition
L'utilisateur doit être connu par le système		L'utilisateur avoir l'accès à son espace privé
Scénario nominal		
1. L'administrateur/l'employé lance l'application 2. Le système affiche un formulaire de connexion 3. L'administrateur/l'employé saisie le « E-Mail » et le « Password » 4. L'administrateur/l'employé demande la connexion 5. Le système vérifie les données saisies 6. Le système valide les données et permet l'accès		

Tableau 3 : description textuelle de cas d'utilisation « S'authentifier »

❖ *Description textuelle* : créer un compte

Titre	Créer un compte	
Objectif	Donner accès au système	
Acteur	Administrateur	
Précondition		Post-condition
L'administrateur doit être authentifié		Un nouveau compte est ajouté
Scénario nominal		
<ol style="list-style-type: none"> 1. L'administrateur accède au menu « Users » 2. L'administrateur choisit « Create users » dans le menu 3. Le système affiche un formulaire de création d'un compte 4. L'administrateur remplit le formulaire 5. Le système valide les données saisies 6. Le système fait les mis à jour et indique que le compte est ajouté avec succès 		

Tableau 4 : description textuelle de cas d'utilisation « Créer un compte »

❖ *Description textuelle* : modifier un compte

Titre	Modifier un compte	
Objectif	Modifier un profil utilisateur	
Acteur	Administrateur	
Pré-condition		Post-condition
L'administrateur doit être authentifié		Un profil utilisateur est modifié
Scénario nominal		
<ol style="list-style-type: none"> 1. L'administrateur accède au menu « Users » 2. L'administrateur choisit « Manage users » dans le menu 3. Le système affiche la liste des comptes des utilisateurs enregistrés 4. L'administrateur demande de modifier un compte 5. Le système affiche un formulaire avec les données enregistrées 6. L'administrateur modifie le formulaire 7. Le système valide les données saisies 8. Le système fait les mis à jour et indique que le compte est modifié avec succès 		

Tableau 5 : description textuelle de cas d'utilisation « Modifier un compte »

❖ *Description textuelle* : contacter un utilisateur (Administrateur, Employé)

Titre	Contacter un utilisateur	
Objectif	Envoyer un message à un utilisateur	
Acteur	Administrateur/Employé	
Pré-condition		Post-condition
L'utilisateur doit être authentifié		Le message est envoyé à l'administrateur
Scénario nominal		
<ol style="list-style-type: none"> 1. L'administrateur/l'employé accède au menu « Contact » 2. L'administrateur/l'employé choisit « Contact users » dans le menu 3. Le système affiche une boîte de message 4. L'administrateur/l'employé écrit le message 5. L'administrateur/l'employé demande d'envoyer le message 		

Tableau 6 : description textuelle de cas d'utilisation « Contacter un utilisateur »

5.2. Description textuelle des cas d'utilisation de « l'Employé »

❖ *Description textuelle* : calculer le Headroom d'une combinaison

Titre	Calculer le Headroom d'une combinaison	
Objectif	Calculer le Headroom d'une combinaison donnée	
Acteur	Employé	
Pré-condition		Post-condition
L'employé doit être authentifié		Le Headroom d'une combinaison est calculé
Scénario nominal		
<ol style="list-style-type: none"> 1. L'employé accède au menu « Headroom & RS » 2. Le système affiche des formulaires d'ajoute d'une combinaison 3. L'employé remplit les formulaires 4. Le système valide les données et affiche les résultats 5. L'employé demande d'enregistrer la combinaison 6. Le système fait les mis à jour et indique que la combinaison est ajoutée avec succès 		

Tableau 7 : description textuelle de cas d'utilisation « Calculer le Headroom d'une combinaison »

❖ *Description textuelle* : modifier une combinaison

Titre	Modifier une combinaison	
Objectif	Modifier les données d'une combinaison	
Acteur	Employé	
Pré-condition		Post-condition
L'employé doit être authentifié		Une combinaison est modifiée
Scénario nominal		
1. L'employé accède au menu « Headroom Register » 2. L'employé choisit « My combinations » dans le menu 3. Le système affiche la liste des combinaisons enregistrées par cet employé 4. L'employé demande de modifier une combinaison 5. Le système affiche un formulaire 6. L'employé remplit le formulaire 7. Le système valide les données saisies 8. Le système fait les mis à jour et indique que la combinaison est modifiée avec succès		

Tableau 8 : description textuelle de cas d'utilisation « Modifier une combinaison »

III. Conception

La conception est une étape primordiale pour la réalisation d'un produit informatique. Nous allons utiliser le diagramme de classes et le diagramme de séquence en vue de présenter le comportement statique et dynamique de notre modèle : le diagramme de classes exprime d'une manière générale la structure statique d'un système, en termes de classes et de relation entre classes et le diagramme de séquence illustre d'une manière dynamique le scénario de réalisation des fonctions du système.

1. Diagramme de classes « modèle statique »

Le diagramme de classes [5] est le point central dans le développement orienté objet, il représente la structure statique du système sous forme de classes et de relations entre classes. Les classes constituent la base pour la génération de code et pour la génération des schémas de bases de données.

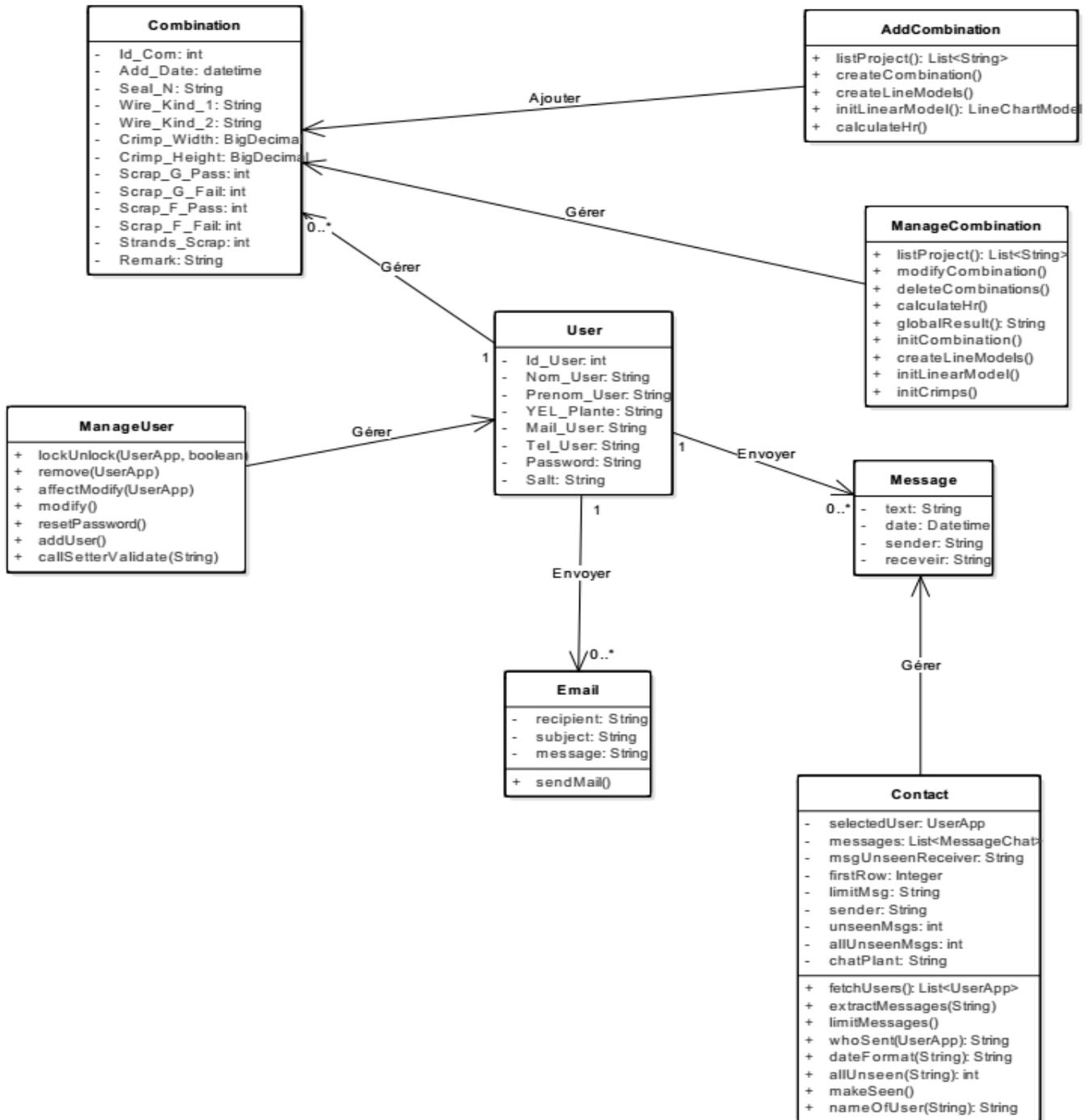


Tableau 11 : diagramme de classes

2. Diagrammes de séquences « modèle dynamique »

Les diagrammes de séquences [5] servent à illustrer les cas d'utilisation, ils permettent de représenter les interactions dans le temps entre les objets du système.

2.1. Diagramme de séquence « Calculer le Headroom »

L'Employé accède au menu « Headroom & RS », le système affiche des formulaires d'ajoute une combinaison. Ensuite l'Employé remplit les formulaires et valide, si ces données sont incorrectes le système lui renvoi un message d'erreur, sinon le système affiche les résultats. L'Employé demande d'enregistrer la combinaison calculée, le système fait les mis à jour et indique que la combinaison est ajoutée avec succès.

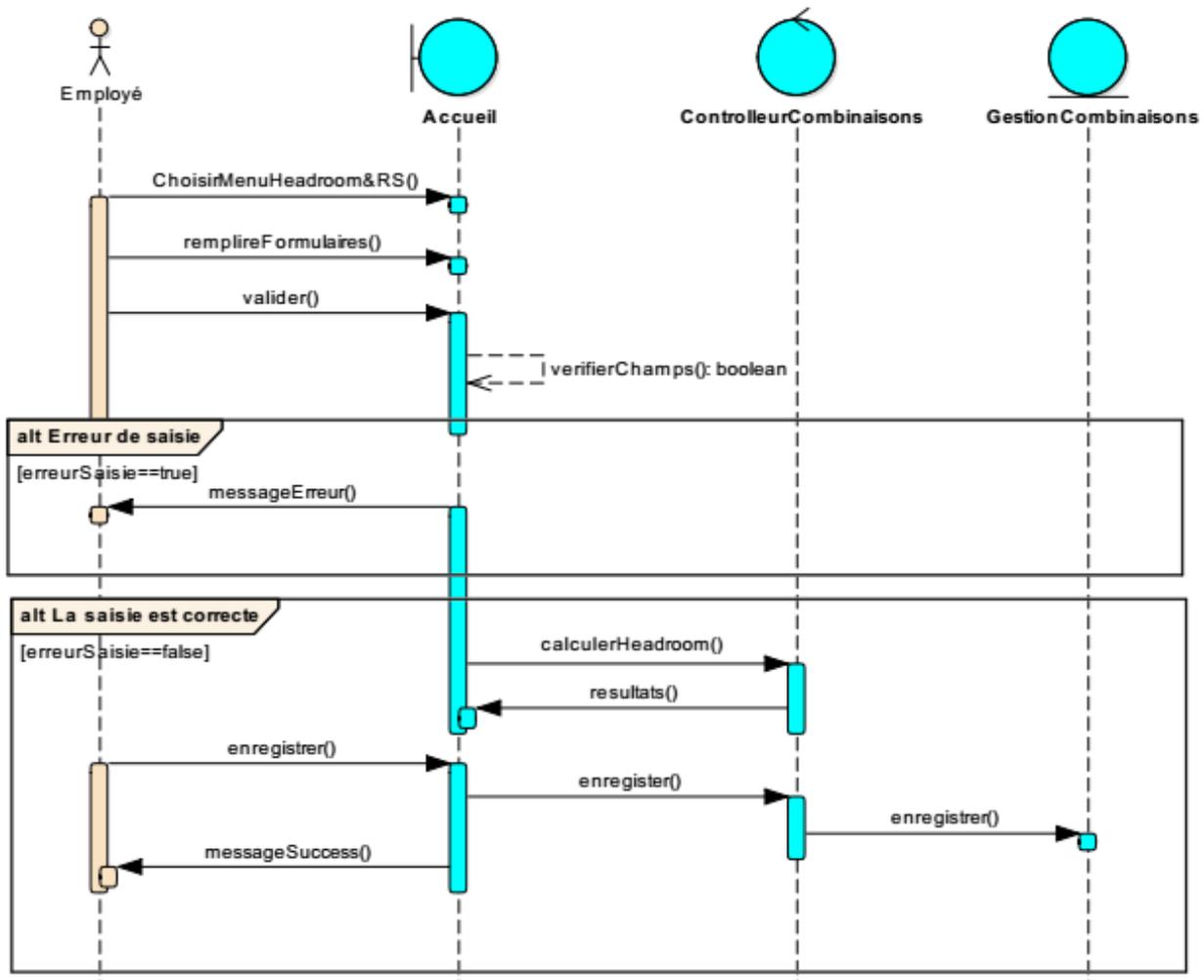


Figure 12 : diagramme de séquence « Calculer le Headroom »

2.2. Diagramme de séquence « Modifier une combinaison »

L'Employé choisit « My combinaisons » dans le menu « Headroom Register », le système affiche la liste des combinaisons enregistrées par cet employé. Ensuite l'Employé demande de modifier une combinaison, le système affiche un formulaire, l'Employé remplit le formulaire et valide, si ces données sont incorrectes le système lui renvoi un message d'erreur, sinon le système fait les mis à jour et indique que la combinaison est modifiée avec succès.

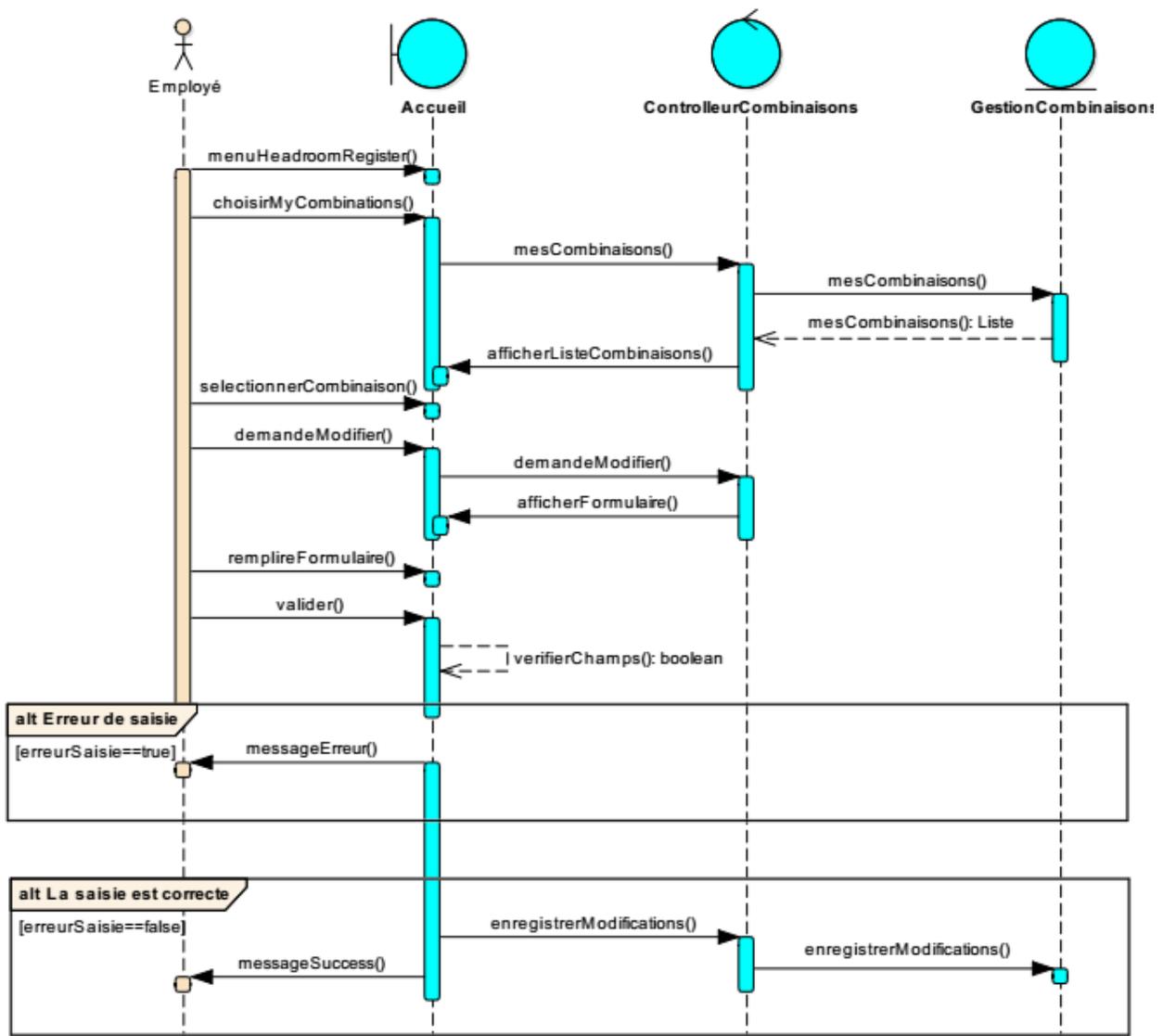


Figure 13 : diagramme de séquence « Modifier une combinaison »

2.3. Diagramme de séquence « Créer un compte »

L'Administrateur choisit « Create users » dans le menu « Users », le système affiche un formulaire de création d'un compte. Ensuite l'Administrateur remplit le formulaire et valide, si ces données sont incorrectes le système lui renvoi un message d'erreur, sinon le système fait les mis à jour et indique que le compte est ajouté avec succès.

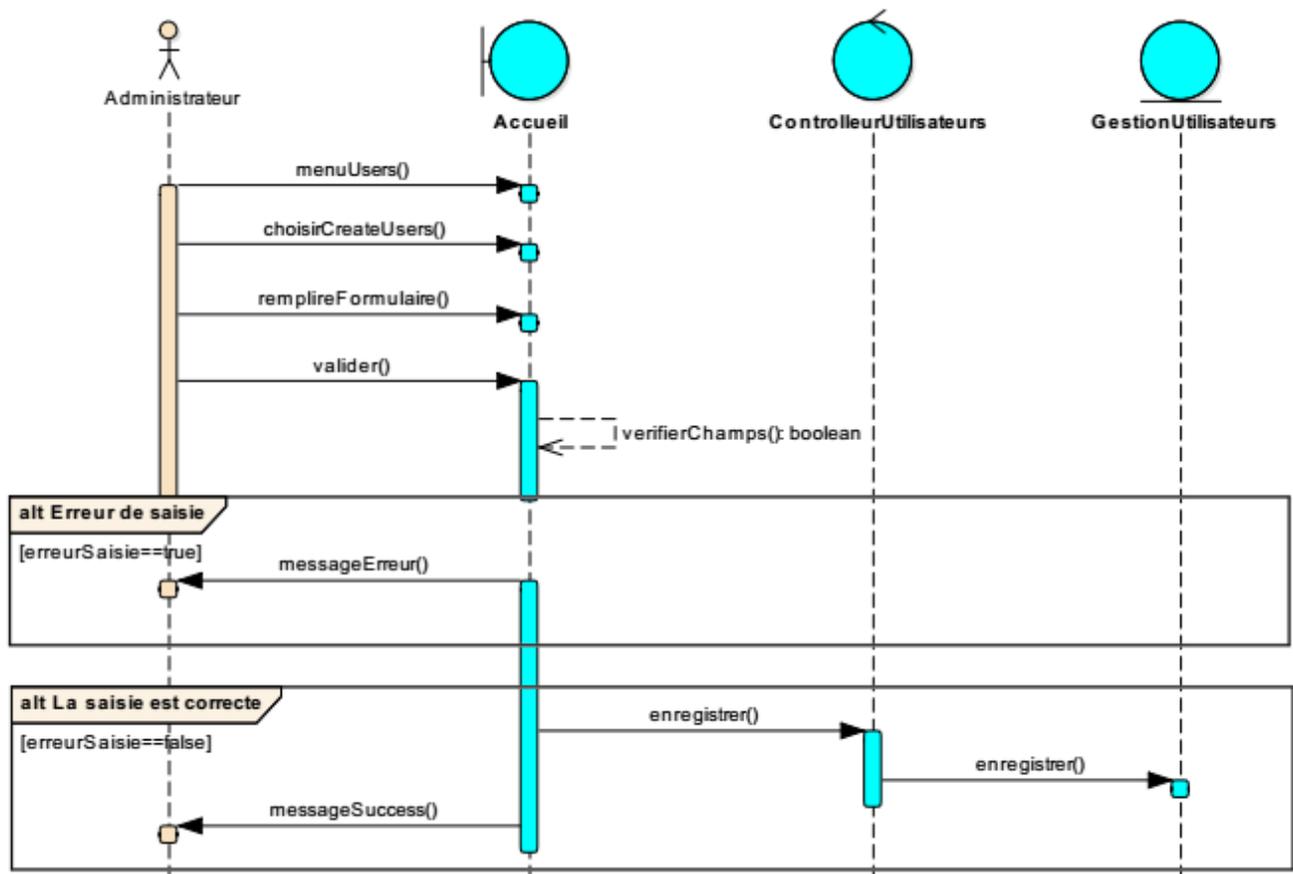


Figure 14 : diagramme de séquence « Créer un compte »

2.4. Diagramme de séquence « Modifier un compte »

L'Administrateur choisit « Manage users » dans le menu « Users », le système affiche la liste des comptes des utilisateurs enregistrés. Ensuite l'Administrateur demande de modifier un compte, le système affiche un formulaire, l'Administrateur remplit le formulaire et valide, si ces données sont incorrectes le système lui renvoi un message d'erreur, sinon le système fait les mis à jour et indique que le compte est modifié avec succès.

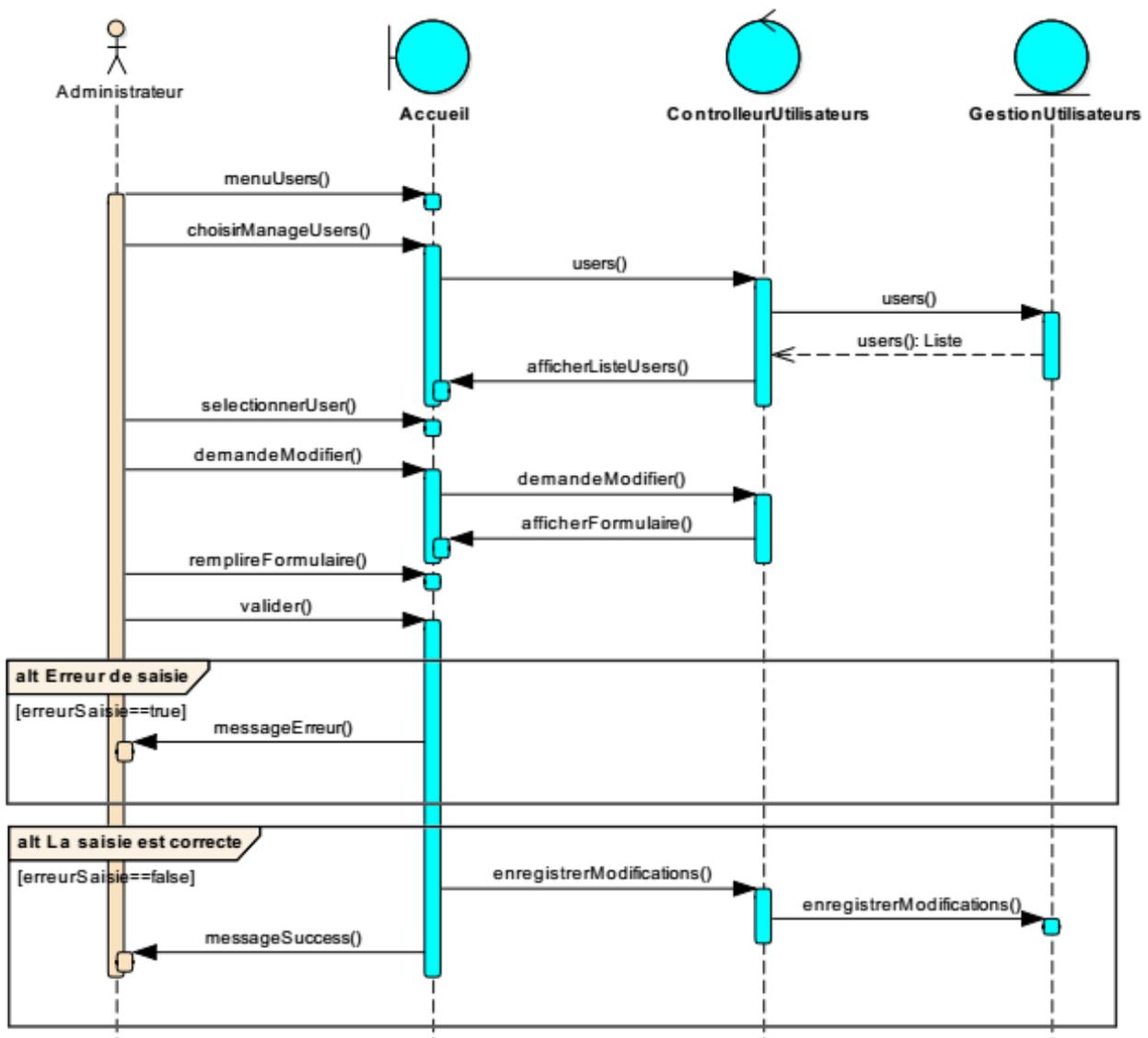


Figure 15 : diagramme de séquence « Modifier un compte »

3. La base de données de l'application

La figure ci-dessous représente le modèle relationnel de données de notre application.

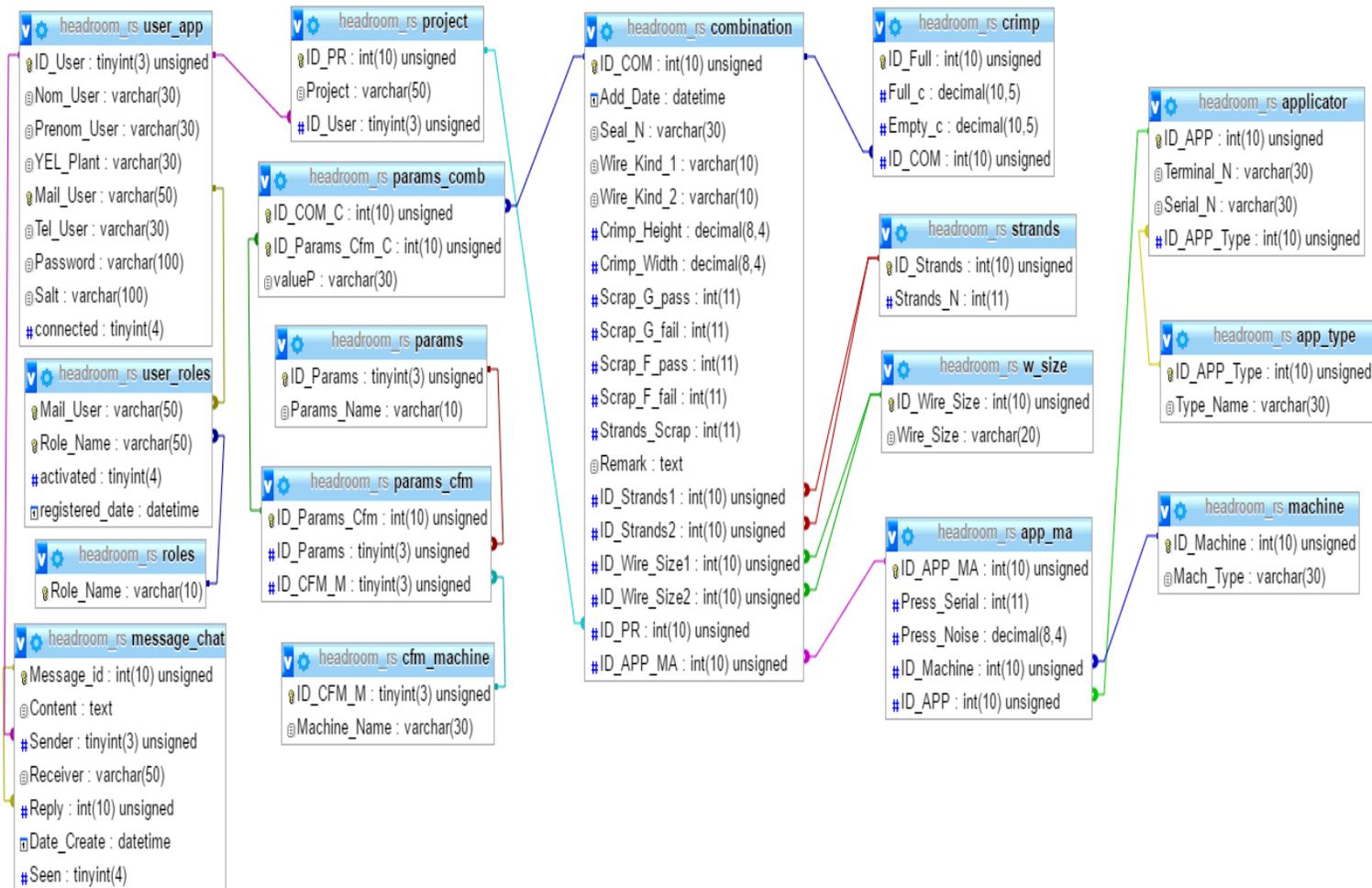


Figure 16 : schéma de la base de données

Conclusion :

Dans ce chapitre nous avons détaillé les différentes vues conceptuelles de l'application à réaliser à travers les modèles UML nécessaires. Cette conception est essentielle pour la phase de réalisation qui constitue l'objet du chapitre suivant.

Chapitre III.

Mise en place de l'application

Chapitre 3

Mise en place de l'application

Introduction :

Dans ce chapitre, nous aborderons la partie réalisation du projet. Plus précisément nous parleront des outils et des langages de développement, puis nous allons entamer les étapes de la réalisation et quelques captures d'écran pour expliquer de fonctionnement du système.

I. Langages et technologies de développement

1. Plate-forme Java EE

1.1. Définition

Sun (le concepteur de Java) a donc mis en place un ensemble de technologies pour réaliser des applications Web. Ces technologies sont regroupées sous le nom J2EE (Java 2 Enterprise Edition), désormais Java EE.

La plateforme Java EE [6] s'appuie entièrement sur le langage Java. Java EE est donc une norme, qui permet à des développeurs, entreprises et SSII de développer leur propre application qui implémente en totalité ou



partiellement les spécifications de SUN. En simplifiant, il est possible de représenter Java EE comme un ensemble de spécifications d'API, une architecture, une méthode de packaging et de déploiement d'applications et la gestion d'applications déployées sur un serveur compatible Java.

1.2. Pourquoi Java EE

Il existe actuellement beaucoup d'autres plates-formes de développement qui sont basées sur d'autres langages (C#, PHP5, .NET...). Les principaux avantages d'utiliser Java EE (et donc Java) sont la portabilité, l'indépendance, la sécurité et la multitude de bibliothèques proposées.

Le développement d'applications d'entreprise nécessite la mise en œuvre d'une infrastructure importante. Beaucoup de fonctionnalités sont utilisées et développées, le but étant de produire des applications sûres, robustes et faciles à maintenir. Certains services sont d'ailleurs récurrents comme : l'accès aux bases de données, l'envoi de mails, les transactions, la gestion de fichiers, la gestion d'images, le téléchargement, le chargement, la supervision du système...

C'est pour cela que l'architecture Java EE est intéressante car tous les éléments fondamentaux sont déjà en place. Pas besoin de concevoir une architecture, des bibliothèques et des outils spécialement adaptés. Cela nécessiterait un temps et un investissement considérables.

Enfin, la plateforme Java EE est basée sur des spécifications, ce qui signifie que les projets sont portables sur n'importe quel serveur d'applications conforme (Tomcat, JBoss, WebSphere...) à ces spécifications. Cette implémentation est gratuite et permet de bénéficier de la totalité de l'API sans investissement. La plateforme Java EE est la plus riche des plates-formes Java et offre un environnement standard de développement et d'exécution d'applications d'entreprise multi-tiers. [6]

2. Framework JSF

JSF [7] (Java Server Faces) est une technologie dont le but est de proposer un framework qui facilite et standardise le développement d'applications web avec Java. Son développement a tenu compte des différentes expériences acquises lors de l'utilisation des technologies standard pour le développement d'applications web (servlet, JSP, JSTL) et de différents frameworks (Struts, ...).



3. HTML

HTML [8] L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte, d'où son nom. Langage de balisagehypertexte Permet de structurer et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des éléments programmables.



4. CSS

Le terme CSS [9] est l'acronyme anglais de « Cascading Style Sheets » qui peut se traduire par « feuilles de style en cascade ». Le CSS est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML.



5. JavaScript

JavaScript [10], souvent abrégé en JS, est le langage de script développé par Netscape utilisé dans des millions de pages web et d'applications serveur dans le monde entier. Le JavaScript de Netscape est une extension du langage de script standard ECMA-262 Edition 3 (ECMAScript), ne différant que légèrement des standards publiés. JavaScript est un langage léger, interprété, orienté objet (les fonctions étant des objets à part entière).



JavaScript

6. JQuery

JQuery [11] est une bibliothèque JavaScript libre qui porte sur l'interaction entre JavaScript (comprenant Ajax) et HTML, et a pour but de simplifier des commandes communes de JavaScript. La première version date de janvier 2006.



7. MySQL

MySQL [12] est un Système de Gestion de Base de Données (SGBD) parmi les plus populaires au monde, C'est est un serveur de base de données relationnelles SQL qui fonctionne sur de nombreux systèmes d'exploitation (dont Linux, Mac OS X, Windows, Solaris, FreeBSD...) et qui est accessible en écriture par de nombreux langages de programmation, incluant notamment PHP, Java, Ruby, C, C++, .NET, Python ...



II. Environnement de réalisation

Dans cette partie avons présentés l'environnement matériel et logiciel de la réalisation de notre projet.

1. Environnement matériel

L'application a été développée sur un ordinateur portable avec les caractéristiques suivantes :

<i>Marque</i>	Dell Inspiron 3543
<i>Processeur</i>	Intel® Core™ i5-5200U CPU @ 2.20GHz (4CPU),
<i>Mémoire vive</i>	4Go
<i>Disque dur</i>	500Go
<i>Type du système</i>	64 bits
<i>Système d'exploitation</i>	Windows 10

Tableau 9 : Caractéristique poste de travail

2. Environnement logiciel

Lors de développement de notre application, nous avons exploité plusieurs logiciels, ça ce qu'on va présenter dans cette partie.

2.1. Apache TomEE

Apache TomEE prononcé « tommy » est un serveur web combinant plusieurs projets d'entreprise JAVA tels que les Servlet, JSP, JSF, JTA, JPA, CDI, JAAS, JACC, JavaMail API, Bean Validation, Entreprise JavaBeans.



2.2. NetBeans IDE

L'EDI NetBeans [13] est un environnement de développement - un outil pour les programmeurs pour écrire, compiler, déboguer et déployer des programmes. Il est écrit en Java - mais peut supporter n'importe quel langage de programmation. Il y a également un grand nombre de modules pour étendre l'EDI NetBeans. L'EDI NetBeans est un produit gratuit, sans aucune restriction quant à son usage.



2.3. Enterprise Architect

Enterprise Architect [14] permet le développement d'applications selon le schéma d'architecture orienté modèle ainsi que le schéma d'Architecture orientée services. Enterprise Architect couvre tous les aspects du cycle de développement d'applications depuis la gestion des exigences, en passant par les phases de conception, la construction, tests et maintenance. Ces aspects sont appuyés par des fonctions de support tels que la traçabilité, la gestion de projet, ou encore le contrôle de version.



III. Présentation de l'application

1. Prototype de l'interface « Authentification »

Il est très important de noter que chaque opération ne peut être réalisée qu'après authentification. Seul « l'Administrateur » qui se charge de définir les login et mot de passe de chaque employé. Au lancement de l'application, la fenêtre d'authentification s'affiche automatiquement, elle permet à l'utilisateur de se connecter. S'il y a des problèmes lors de la connexion, un message d'erreur est affiché.

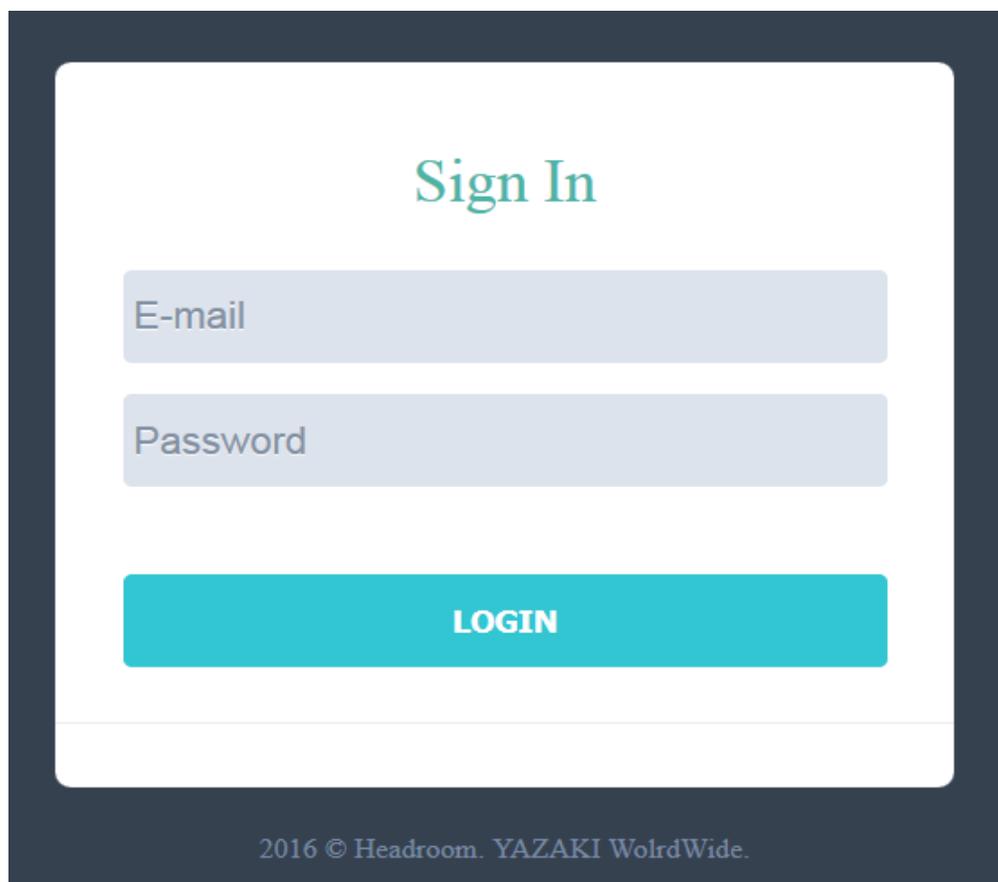
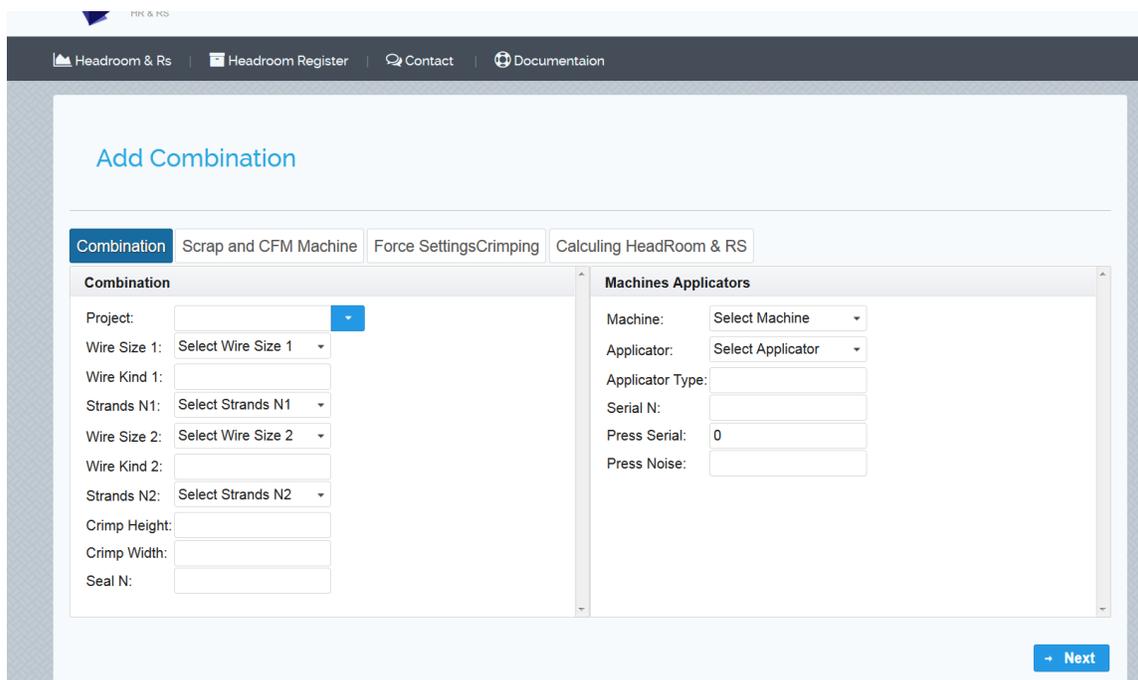


Figure 17 : interface « authentification »

2. Prototype de l'interface « Calculer le Headroom »

L'application permet à « l'Employé » de calculer le Headroom d'une combinaison en saisissant des informations correspondant à celle-ci



The screenshot shows a web application interface titled "Add Combination". At the top, there is a navigation bar with links for "Headroom & Rs", "Headroom Register", "Contact", and "Documentaion". Below the navigation bar, the main content area is divided into two panels: "Combination" and "Machines Applicators".

The "Combination" panel contains the following fields:

- Project:
- Wire Size 1:
- Wire Kind 1:
- Strands N1:
- Wire Size 2:
- Wire Kind 2:
- Strands N2:
- Crimp Height:
- Crimp Width:
- Seal N:

The "Machines Applicators" panel contains the following fields:

- Machine:
- Applicator:
- Applicator Type:
- Serial N:
- Press Serial:
- Press Noise:

At the bottom right of the form, there is a blue button labeled "Next".

Figure 18 : interface « calculer le Headroom »

3. Prototype de l'interface « Gestion des combinaison »

L'application permet à l'administrateur une gestion complète des combinaisons, où il la possibilité de consulter tous les combinaisons et la génération de leurs rapports (sous forme PDF ou Excel), ainsi que la modification et la suppression de ses combinaisons.

Headroom & Rs | Headroom Register | Contact | Documentaion

My Combinations

Excel | Report | Delete | Edit

My Combinations									
Date	Operator	YEL Plant	Terminal N	Wire Size 1	Wire Size 2	Wire Kind 1	Wire Kind 2	Seal N	
2016-06-03 11:33:03.0	alam yousef	Yazaki Meknes	7009-5873-02	8	10	tra	ttt5	0000-0000	

2016 © Headroom. YAZAKI WolrdWide.

Figure 19 : interface « my combinations »

Headroom & Rs | Headroom Register | Contact | Documentaion

Cancel | Report | Edit

Combination | Scrap and CFM Machine | Force Settings | Calculating HeadRoom & RS

Combination Informations

Project: Renault PM31

Wire Size 1: 8

Wire Kind 1: tra

Strands N1: 45

Wire Size 2: 10

Wire Kind 2: ttt5

Strands N2: 45

Crimp Height: 3.0000

Machines Applicators

Machine: AC90

Applicator: 7009-5873-02

Applicator Type: 87 side feed

Serial N: F24521

Press Serial: 103

Press Noise: 0.3000

Next

Figure 20 : interface « edit combination »

Headroom & Rs | Headroom Register | Contact | Documentaion

All Combinations

Excel | Report | Delete

All Combinations								
10 (1 of 1)								
	Date	Operator	YEL Plant	Terminal N	Wire Size 1	Wire Size 2	Wire Kind 1	Wire Kind 2
<input type="checkbox"/>	2016-06-03 11:33:03.0	alam youssef	Yazaki Meknes	7009-5873-02	8	10	tra	ttt5

Figure 21 : interface « all combinations »

Headroom_Relative Spread Scrap Ratio vs Missing Strands Detection

Terminal: 7009-5873-02 | Seal: 0000-0000

Information		Applicator Settings		Force Settings			
Date	2016-06-03 11:33:03.0	Date	F24521	Empty Crimp		Full Crimp	
Yel Plant	Yazaki Meknes	Yel Plant	87 side feed	FEmpty0	170.60000	FFull0	206.00000
Operator	alam youssef	Operator	0000-0000	FEmpty1	160.00000	FFull1	204.70000
Project	Renault PM31			FEmpty2	159.40000	FFull2	205.60000
Combination		Machine Settings		FEmpty3	160.40000	FFull3	203.40000
Wire Kind 1	tra	Machine Type	AC90	FEmpty4	160.20000	FFull4	205.60000
Wire Size 1	8	Press Serial N	103	FEmpty5	160.40000	FFull5	203.40000
Wire Kind 2	ttt5	Press Noise	0.3000	FEmpty6	158.10000	FFull6	207.10000
Wire Size 2	10			FEmpty7	162.40000	FFull7	208.30000
Crimp Height	8			FEmpty8	160.10000	FFull8	204.10000
Crimp Width	8			FEmpty9	159.30000	FFull9	205.20000
				FEmpty10	159.40000	FFull10	205.90000
				FEmpty11	161.20000	FFull11	204.20000

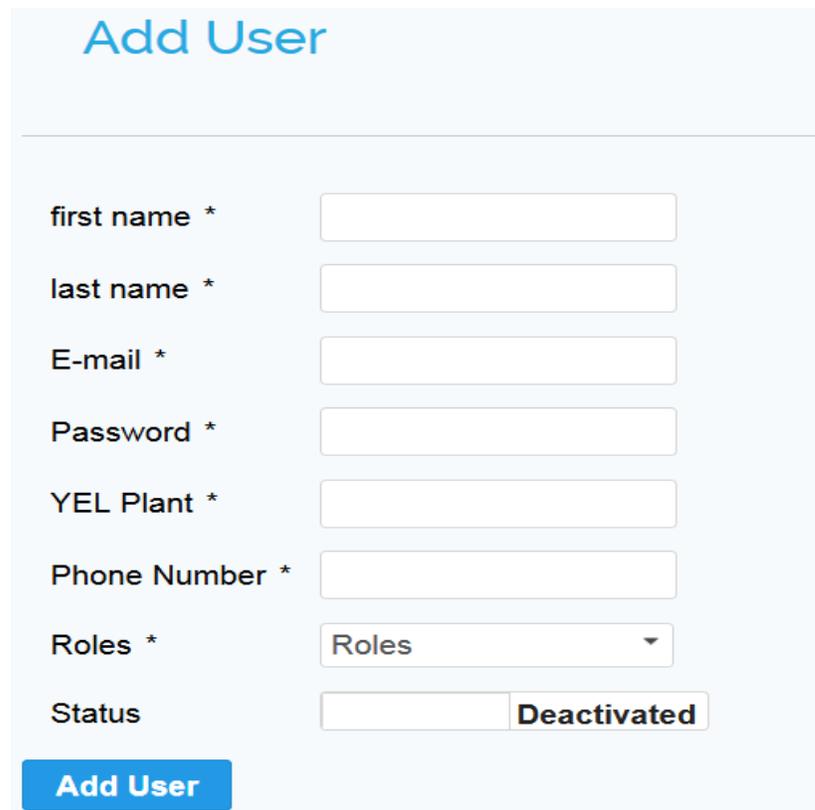
Figure 22 : rapport de combinaison (PDF)

Date	Operator	Yel Plant	Terminal N	Seal N	Wire Kind 1	Wire Kind 2	Wire Size 1	Wire Size 2	Press Noise	Crimping Height	Crimping Width	HR Min	HR Max	Average HR	RS
06/06/2016	alam youssef	YMO	7123-7231		2	3	12	23	0.4		2	3	23%	23%	12%

Figure 23 : rapport de combinaison (Excel)

4. Prototype de l'interface « Créer un compte »

Pour créer des comptes utilisateurs il est nécessaire à « l'Administrateur » de passer à l'interface de création des comptes comme illustre la figure suivante :



The image shows a web interface titled "Add User". It contains a form with the following fields and controls:

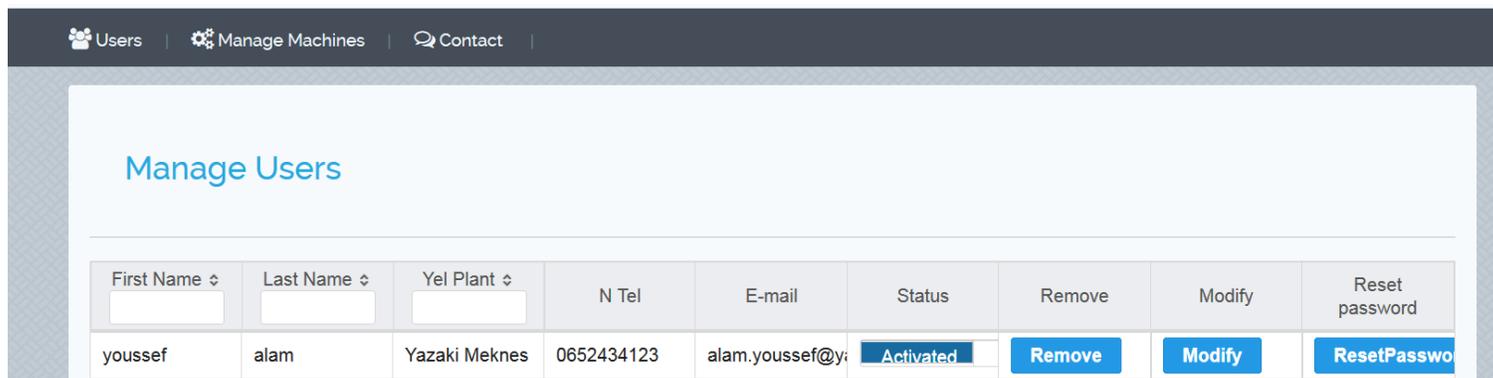
- first name *
- last name *
- E-mail *
- Password *
- YEL Plant *
- Phone Number *
- Roles *
- Status

At the bottom left of the form is a blue button labeled "Add User".

Figure 24 : interface « add user »

5. Prototype de l'interface « Gestion des comptes »

Pour consulter, modifier ou supprimer des comptes utilisateurs, il est nécessaire à « l'Administrateur » de passer à l'interface de gestion des comptes comme illustre la figure suivante :

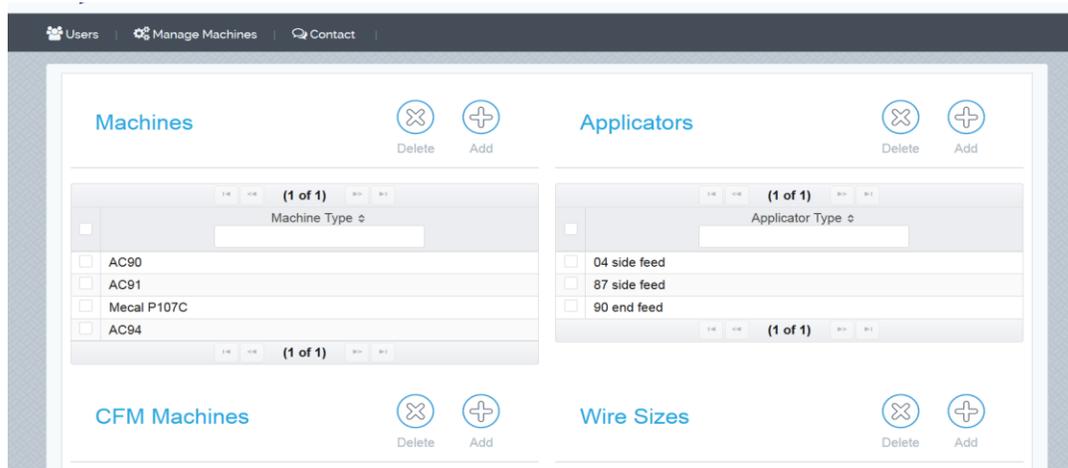


First Name	Last Name	Yel Plant	N Tel	E-mail	Status	Remove	Modify	Reset password
youssef	alam	Yazaki Meknes	0652434123	alam.youssef@y	Activated	Remove	Modify	ResetPasswo

Figure 25 : interface « manage users »

6. Prototype de l'interface « Gestion des machines »

L'application permet à l'administrateur une gestion complète des machines et des paramètres de combinaison, où il la possibilité de consulter, ajouter et supprimer.



The interface displays four main sections for machine management:

- Machines:** A list of machine types including AC90, AC91, Mecal P107C, and AC94.
- Applicators:** A list of applicator types including 04 side feed, 87 side feed, and 90 end feed.
- CFM Machines:** A section for managing CFM machines.
- Wire Sizes:** A section for managing wire sizes.

Figure 26 : interface « manage machines »

7. Prototype de l'interface « Contacter un utilisateur »

L'application permet la communication entre ses utilisateurs

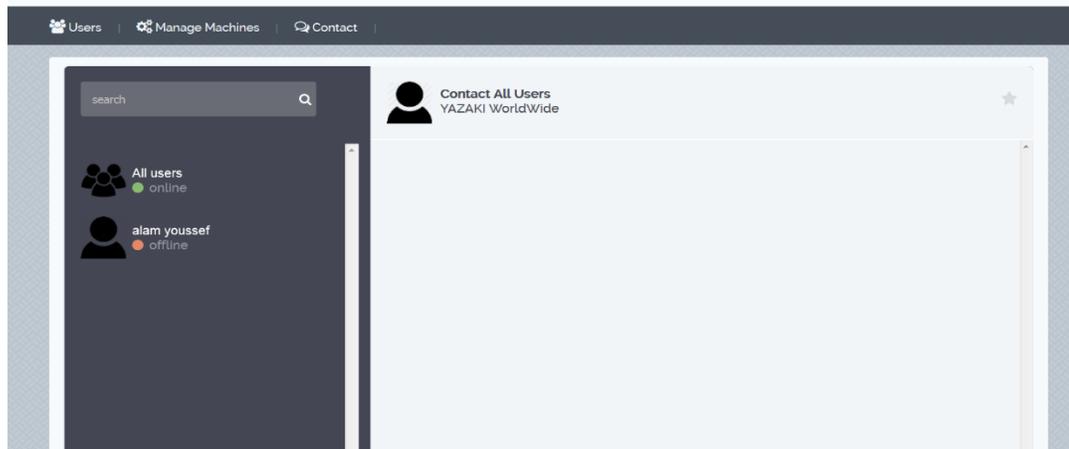


Figure 27 : interface « contact users »

8. Prototype de l'interface « Contacter l'administrateur »

L'application permet à ses utilisateurs d'envoyer un email à l'administrateur

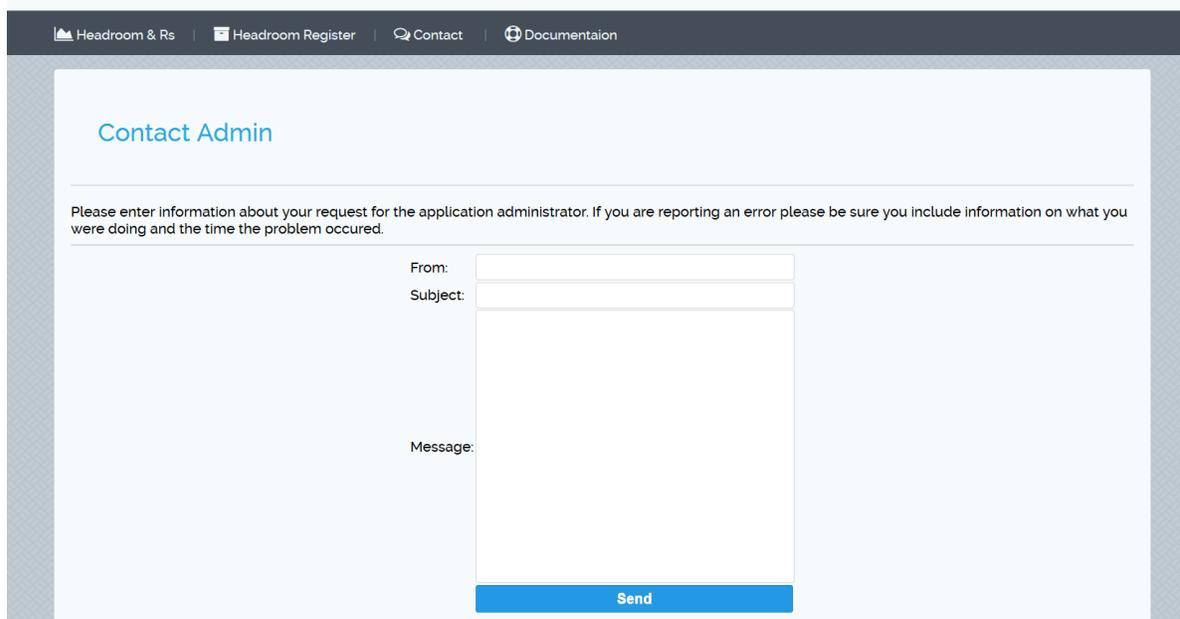


Figure 28 : interface « contact admin »

Conclusion générale

Ce stage a été l'occasion de mettre à profit nos connaissances acquises à la Faculté des Sciences et Techniques Fès notamment en JAVA, HTML, CSS, JavaScript et en MySQL pour pouvoir approfondir mes connaissances en JAVA et la pratique et la compréhension des outils MySQL. Nous avons également acquis de nouveaux outils de développement tels que la plate-forme JAVA EE et le Framework JSF.

Ce projet nous a donné l'opportunité de s'initier à la vie professionnelle dans un milieu réel et avoir au début d'expérience significative et il nous a appris comment dès le matin on peut prendre le parti de la gaieté, comment réussir de bonnes relations pour assurer un travail du groupe, comment compter sur soi pour résoudre les problèmes au cas où ils se présentent, comment être attentives aux indications de nos supérieurs.

Au cours de la réalisation de notre projet, nous avons été astreints par quelques limites notamment, la contrainte de temps qui était relativement un obstacle devant l'ajout de certaines autres fonctionnalités.

En définitive, ce stage a été une expérience déterminante dont je pourrais certainement profiter au niveau professionnel et de manière encore plus évidente au niveau personnel.

Annexes

- **Le code de hachage et de salage**

```
public class Hasher implements Serializable{
    public final Integer DEFAULT_ITERATIONS = 10000;
    public final String algorithm = "pbkdf2_sha256";
    private String salt;

    public Hasher() {
        salt = BCrypt.gensalt();
    }

    public String getEncodedHash(String password, int iterations) {
        // Returns only the last part of whole encoded password
        SecretKeyFactory keyFactory = null;
        try {
            keyFactory = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
        } catch (NoSuchAlgorithmException e) {
            System.err.println("Could NOT retrieve PBKDF2WithHmacSHA256
algorithm");
        }

       KeySpec keySpec = new PBEKeySpec(password.toCharArray(),
salt.getBytes(Charset.forName("UTF-8")), iterations, 256);

        SecretKey secret = null;
```

```
try {
    if(keyFactory != null)
    {
        secret = keyFactory.generateSecret(keySpec);
    }
} catch (InvalidKeySpecException e) {
    System.out.println("Could NOT generate secret key");
}
if(secret != null)
{
    byte[] rawHash = secret.getEncoded();
    byte[] hashBase64 = Base64.getEncoder().encode(rawHash);
    return new String(hashBase64);
}
return null;
}

public String encode(String password, int iterations) {
    // returns hashed password, along with algorithm, number of iterations and salt
    String hash = getEncodedHash(password, iterations);
    return hash;
}

public String encode(String password) {
    return this.encode(password, this.DEFAULT_ITERATIONS);
}

public boolean checkPassword(String password, String hashedPassword) {
```

```
String hash = encode(password, this.DEFAULT_ITERATIONS);

return hash.equals(hashAndPassword);
}

public String getSalt() {
    return salt;
}

public void setSalt(String salt) {
    this.salt = salt;
}
```

• Le code d'envoi d'un email

```
Transport transport = null;
try {
    MimeMessage mimeMessage = new MimeMessage(mailSession);

    mimeMessage.setFrom(new InternetAddress(FROM));
    mimeMessage.setSender(new InternetAddress(FROM));
    mimeMessage.setSubject(subject);
    mimeMessage.setContent(message, "text/plain");

    mimeMessage.addRecipient(Message.RecipientType.TO, new
    InternetAddress(recipient));
```

```
transport = mailSession.getTransport("smtp");
transport.connect(HOST, PORT, USER, PASSWORD);

transport.sendMessage(mimeMessage, mimeMessage.getRecipients(
Message.RecipientType.TO ));
transport.close();
} catch (MessagingException ex) {
    Logger.getLogger(Email.class.getName()).log(Level.SEVERE, null, ex);

} finally {
    if(transport != null && transport.isConnected()) try {
        transport.close();
    } catch (MessagingException ex) {
        Logger.getLogger(Email.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Bibliographie

- [3] : Unified Modeling Language (UML). Bruno BOUZY
- [4] : UML en action de l'analyse des besoins à la conception en Java. Pascal ROQUES
- [5] : Modélisation en UML. Pr. Abderrahim BENABBOU
- [6] : Java EE-Guide de développement d'applications web en Java. Jérôme LAFOSSE

Webographie

- [1] : [http:// www.yazaki-europe.com](http://www.yazaki-europe.com)
- [2] : <http://docslide.fr/documents/rapport-de-stage-qualite.html>
- [7] : <http://www.jmdoudoux.fr/java/dej/chap-jsf.htm>
- [8] : <http://slideplayer.fr/slide/4345759>
- [9] : <http://glossaire.infowebmaster.fr/css>
- [10] : https://developer.mozilla.org/fr/docs/Web/JavaScript/A_propos
- [11] : <http://distillari.es/javascript/jquery>
- [12] : <http://sql.sh/sghd/mysql>
- [13] : https://netbeans.org/index_fr.html
- [14] : http://republic.pink/enterprise-architect_1439556.html