



"Synchronisation multi-base"

Par

"Andrianarisaona Misaina Tsiory Tafita"

Mémoire présenté

en vue de l'obtention du grade de Licence

En Informatique

Option Développement

"Fevrier, 2021"

Jury :

Monsieur Rojo Rabenanahary, président

Monsieur Rakotondralambo, examinateur

Madame Ravakiniaina Andriamiadanary, encadreur professionnel

Table des matières

Liste des tableaux	ii
Liste des figures.....	iii
Glossaire	iv
Avant-propos	1
L'IT University.....	1
Le Bureau international de Consultance en Informatique	1
Remerciements	3
Introduction	4
1 Présentation du projet	5
1.1 Objectifs du projet	5
1.2 Planning de réalisation	5
1.3 Technologies utilisées	7
1.4 Architecture logicielle	10
2 Réalisation de l'application	11
2.1 Analyse et conception.....	11
2.1.1 Analyse de l'existant	11
2.1.2 Conception de l'application.....	12
2.2 Développement par fonctionnalité ou module.....	15
2.2.1 Gestion des bases de données et de projets	15
2.2.2 Gestion des serveurs distants	19
2.2.3 Synchronisation multi-base	21
2.3 Problèmes rencontrés et solutions	26
2.4 Budget du projet	26
3 Évaluation du projet et connaissances acquises	27
3.1 Bilan pour l'entreprise.....	27
3.2 Bilan personnel	27
3.3 Extension et évolution de l'application	27
Conclusion.....	28
Bibliographie	A

Liste des tableaux

Tableau 1 : Représentation des technologies utilisées	7
Tableau 2: Comparaison Java - PHP	9
Tableau 3: Comparaison MySQL - PostgreSQL	10
Tableau 4: Exemple base de données	16
Tableau 5: Explication fichier de configuration	24
Tableau 6: Budget du projet	26

Liste des figures

Figure 1: Diagramme de Gantt	6
Figure 2: Top 7 des meilleurs langages de programmation en 2020.....	8
Figure 3:Evolution de la popularité des bases de données	9
Figure 4: Architecture EJB	11
Figure 5 : Analyse de l'existant	12
Figure 6 : conception MPD Serveur	14
Figure 7: MPD Application à synchroniser	14
Figure 8:Page de connexion	15
Figure 9: Menu de l'application	15
Figure 10: Saisie de base de données	16
Figure 11:Liste base de données.....	17
Figure 12:Fiche base de données.....	17
Figure 13:Saisie de Projet.....	18
Figure 14:Liste Projet	18
Figure 15: Fiche Projet	18
Figure 16:Saisie Serveur.....	19
Figure 17:Liste Serveur	20
Figure 18:Fiche serveur	21
Figure 19:Script à exécuter.....	21
Figure 20: Interface de Radmin VPN	22
Figure 21: Fichier de configuration synchronisation.....	23
Figure 22:Fichiers de synchronisation exécutables	24
Figure 23: Déroulement de la synchronisation.....	24
Figure 24:Code de Synchronisation	25

Glossaire

API : Application Programming Interface

APJ : Application Personnalisée JRWES

Backend : Partie d'une application non visible

CSS : Cascading Style Sheet

Framework : ensemble de composants structurels logiciels

Frontend : Partie d'un système informatique ou d'une application avec laquelle l'utilisateur interagit directement

HTML : HyperText Markup Language

IEEE : Institute of Electrical and Electronics Engineers

IHM : Interface Homme-Machine

IP : Internet Protocol

JDBC : Java Database Connectivity

JEE : Java Enterprise Edition

JSP : JavaServer Pages

MPD : Modèle Physique de Données

MVC : Modèle-Vue-Contrôleur

PHP : Hypertext Preprocessor

SGBD : Système de Gestion de Base de Données

TIC : Technologie de l'Information et de la Communication

VPN : Réseau Privé Virtuel

Test fonctionnel. Test qui sert à tester automatiquement toutes les fonctionnalités de l'application, pour voir par exemple si un utilisateur peut bien se connecter.

Test unitaire. Procédure permettant de vérifier le bon fonctionnement d'une partie d'un logiciel ou d'une portion de programme, on l'utilise par exemple pour tester si l'algorithme qui calcule le nombre de jours entre deux dates retourne une valeur exacte.

Avant-propos

Le présent mémoire présente les résultats du travail effectué lors de notre stage de fin d'études de Licence en Informatique de l'IT University : stage effectué au Bureau International de Consultance en Informatique (BICI) durant 3 mois, du mois d'Octobre jusqu' au mois de Décembre 2020. Afin de poser clairement le contexte de ce mémoire, nous allons présenter de façon succincte de l'IT University et de notre entreprise d'accueil : le BICI.

L'IT University

Fondée en 2011, l'IT University est une université privée, spécialisée en informatique, formant les jeunes bacheliers scientifiques :

- En trois ans, pour l'obtention d'une Licence, option Développement d'applications, Réseaux et Bases de Données ou Web et Design
- En cinq ans, pour l'obtention d'un Master MBDS en coopération avec l'Université de Nice Sophia Antipolis – France

Étant une formation professionnalisante, l'ITU a tissé des liens forts avec ses partenaires industriels, dont l'opérateur convergent TELMA, le GOTICOM ¹ et la plupart des entreprises et institutions du secteur des TIC ². Ces partenaires participent effectivement à la formation par la fourniture de connexion Internet à haut débit, l'envoi de conférenciers ou encore par l'accueil des étudiants en stage. D'autre part, le corps enseignant de l'IT University est constitué intégralement de spécialistes de très haut niveau et obligatoirement actifs professionnellement dans leurs domaines respectifs. Enfin, l'IT University fait partie du programme Microsoft Imagine en plus d'être un Oracle Gold Partner

Le Bureau international de Consultance en Informatique

Le BICI ou Bureau International de Consultance en Informatique est une branche du groupe « Vidy Varotra », une société de Service en Ingénierie Informatique Malgache créée en 2003, sous l'appellation « JRews Informatique » à l'époque. Sa principale activité est l'assistance informatique et BICI offre aujourd'hui les services suivants :

- Ingénierie logicielle : développement spécifique, mobile et site web

¹ GOTICOM : Groupement des Opérateurs des Technologies de l'Information et de la Communication

² TIC : Technologies de l'Information et de la Communication

- Ingénierie de performance et sécurité
- Réseau, communication et système d'exploitation
- Solution matérielle : dépannage, location et restauration de matériel
- Audit Informatique
- Formation (en partenariat avec l'IT University)

Remerciements

En premier lieu, je ne saurais ne pas remercier Le Tout Puissant pour toute la bénédiction dont j'ai été témoin tout au long de mon existence et particulièrement durant l'accomplissement de tous les projets que j'ai entrepris, dont celui-ci.

J'exprime mes sincères gratitude au Consulat Générale Honoraire de Monaco à Madagascar de m'avoir accordé ma bourse d'étude.

Je tiens à remercier également, toute l'équipe pédagogique de l'IT University responsable de la formation développement d'application de m'avoir transmis les savoirs durant ces trois (3) ans jusqu'à l'aboutissement de ce présent mémoire.

Plus particulièrement, je remercie Monsieur Rojo Rabenanahary d'avoir accepté de m'encadrer dans mon stage. De même, je tiens à remercier vivement les membres du jury pour l'intérêt qu'ils ont porté pour l'examen de ce travail.

Je remercie également la société BICI Madagascar de m'avoir accueilli pour que je puisse effectuer mon stage de troisième année en vue de l'obtention de ma Licence et surtout Madame Ravakiniaina Andriamiadanary mon encadreur professionnelle et toute son équipe, pour l'aide et les conseils qu'il m'a apporté lors des différents suivis.

Enfin, je tiens également à remercier ma famille, mes amis et mes collègues de la onzième promotion de l'IT University pour leur aimable soutien.

A vous tous, mes sincères remerciements.

Introduction

Ces dernières années, le monde numérique prend de plus en plus d'ampleur dans le quotidien. Il ne cesse de s'améliorer et de se renouveler. La mise en place des systèmes d'informations est de plus fréquente, et les données des entreprises sont stockées sur divers serveurs de base de données. Non seulement ce genre de système permet le bon fonctionnement d'une entreprise mais il permet aussi de fournir des paramètres décisionnels qui leur permettront d'être plus performant et d'améliorer leur entreprise. De plus, les données sont au cœur des discussions actuelles, leur collecte, leur traitement, leur utilisation. Elles présentent un enjeu pour les grandes entreprises pour pouvoir produire de meilleures prévision et analyse.

Les données de plusieurs entreprises sont actuellement éparpillées dans divers serveurs car ces derniers sont délocalisés ou se situent dans des régions différentes. Pourtant, synchroniser les données dans un système d'information est primordiale car les données sont des indicateurs clés et améliorent de manière considérables la stratégies omnicanal dans une entreprise. Elles aident dans la prise de décision et pour la création de valeur. Cependant, il existe divers logiciels et applications afin de synchroniser les données des entreprises. Mais ces applications ont certains failles et limites. En effet, il existe aussi les services cloud afin de stocker les données mais ils sont chers et il y a une risque au niveau de la sécurité vue la multitude de piratage. De ce fait, le BICI a fait lui-même la synchronisation des serveurs des systèmes d'informations qu'on a développés. Mais hélas, des failles et des risques se présentaient en grande majorité. La synchronisation était encore vulnérable mais aussi lent et ne pouvait prendre ne compte qu'un seul type de base de données. Cette réflexion mène alors, à l'aide du choix intitulé "Synchronisation multi-base ».

Ainsi, un problème se pose à notre intellect, Comment synchroniser les bases de données d'une application de façon optimal tout en minimisant les risques ?

Afin de mieux répondre à notre problème, c'est dans un sens logique que nous évalueront les trois grandes parties du présent mémoire à savoir : la présentation du projet, sa réalisation et son évaluation avec les connaissances acquises.

1 Présentation du projet

1.1 Objectifs du projet

Pour une boîte informatique comme BICI, la synchronisation des bases de données de leurs clients est un besoin vital. Comme on le sait déjà, les données sont les plus importantes dans un système d'information car c'est un élément majeur afin de prendre des décisions mais surtout pour la création de valeur. Le projet a surtout pour objectif d'accélérer et faciliter la synchronisation des serveurs base de données au sein de chaque projet. Pour ce faire, une plateforme web facilitera la gestion des serveurs de bases de données mais aussi le suivi de chaque serveur.

Notre application a donc pour objectif de remédier à tout cela et à développer une certaine efficacité dans la gestion des serveurs des projets développés au sein du BICI. Elle va faciliter l'ajout d'un nouveau serveur et permettre aux données d'être les mêmes à tout moment.

1.2 Planning de réalisation

Lors de la réalisation d'un projet, faire un planning est très important. Il permet non seulement à partager la vision des travaux que l'on va effectuer mais aussi aide beaucoup dans le suivi de la réalisation d'un projet. Durant ces trois mois de stage, on a planifié les tâches via le diagramme de Gantt, accompagné des listes de tâches quotidiennes afin d'éviter les pertes de temps. Le projet étant composé de plusieurs parties plus ou moins indépendantes qu'on verra plus tard ; les étapes de conception, de développement, de test et de mise en production se sont répétées pour chaque partie du projet :

- Avant-projet :

Au début du stage, on a étudié le Framework APJ, en réalisant divers projets au sein de l'entreprise, y compris la synchronisation inter-Oracle pour un projet en production.

- Conception :

Après l'étude du projet en production, notre travail consistait à définir la structure de la base de données, mais aussi à concevoir les composants techniques appropriés pour atteindre les objectifs fixés par le projet.

- Développement :

C'est la partie où la conception prend forme. C'est à partir des conceptions faites que le projet se concrétise avec la création des tables et la mise en œuvre des différentes fonctionnalités

- Test et fin du projet :

Tout au long de l'élaboration de chaque partie du projet, plusieurs tests sont effectués pour s'assurer du bon fonctionnement du système. On a fait cinq (5) types de tests : les tests unitaires pour tous les modules, les tests d'intégration des différents modules entre eux, les tests fonctionnels afin de vérifier la conformité de l'application, les tests de performance et de montée en charge des serveurs d'exécution.

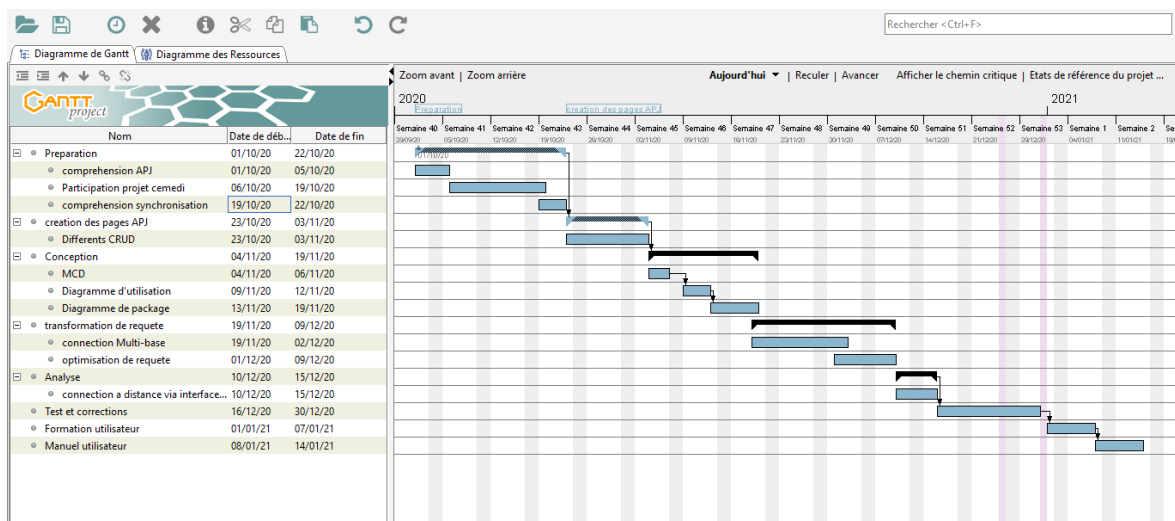


Figure 1: Diagramme de Gantt

1.3 Technologies utilisées

Partie	Technologies
Front-end	   
Back-end	
Base de données	  

Tableau 1 : Représentation des technologies utilisées

Le tableau ci-dessus représente succinctement les technologies dont nous avons utilisé pour la réalisation de ce projet.

a. Front-end :

Pour la partie front-end, nous avons choisi les pages dynamiques JSP, dont le squelette est soutenu par le trio infernal : HTML, CSS, JS. Le tout, rendu dynamique et alimenté par les données

b. Back-end

Pour la partie back-end, nous avons choisi le langage Java qui joue bien sa part de popularité en deuxième place dans le classement des meilleurs langages selon l'IEEE. Le langage java dispose d'API forts pratiques tel que la manipulation de chaine, d'image. « La rapidité de conception d'un programme, console, IHM, bref c'est simple et rapide. Le fait que des tas de classes, api

existent déjà vous font gagner un temps. Java est en constante évolution du fait de sa masse importante d'utilisateur »³

Comparé à PHP qui fait le succès du web pour sa simplicité et sa facilité d'accès, Java possède tout de même des critères bien plus intéressants.

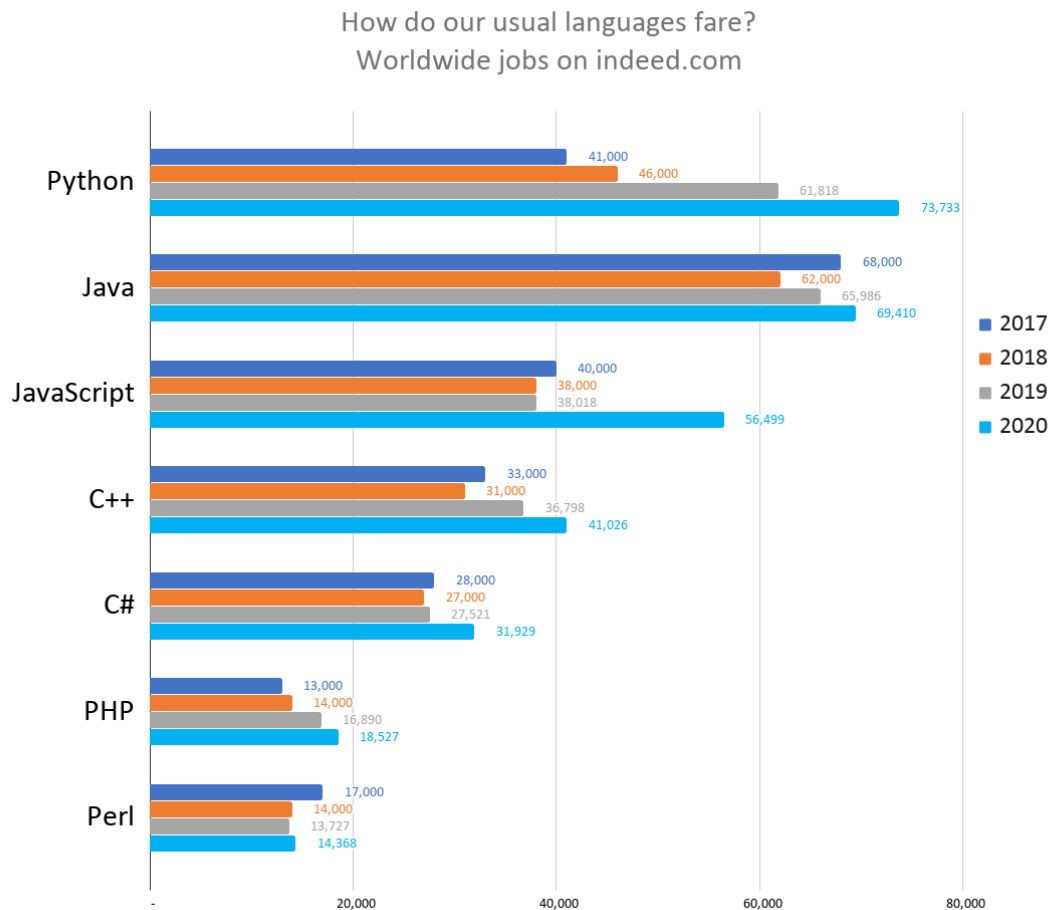


Figure 2: Top 7 des meilleurs langages de programmation en 2020⁴

Critère	Java	PHP
Paradigme	Orienté objet, structuré, impératif	Impératif, objet, fonctionnel, procédural, réflexif. Langage interprété
Typage	Statique, fort, sûr, nominatif	Dynamique, Faible

³ <https://www.security-helpzone.com/2016/10/22/pourquoi-le-java-est-il-si-utilise> (Consulté le 11/11/2020)

⁴ <https://www.243tech.com/meilleurs-langages-de-programmation> (Consulté le 08/12/2020)

Application cible	Tous(Write Everywhere)	Once	Run	Web
-------------------	---------------------------	------	-----	-----

Tableau 2: Comparaison Java - PHP

c. Base de données

Quant à la partie base de données, elle est assurée par PostgreSQL pour notre application Web qui est un SGBD ⁵adapté pour entreprise et dont la popularité est due à son caractère open source, lui permettant ainsi de bénéficier d'une constante amélioration par des développeurs expérimentés. PostgreSQL est un SGBD open source, qui prend en charge les données volumineuses et la gestion fine possible des droits d'accès.⁶

De plus, comparé à MySQL par exemple, PostgreSQL présente plusieurs avantages que nous pouvons constater dans le tableau ci-dessous

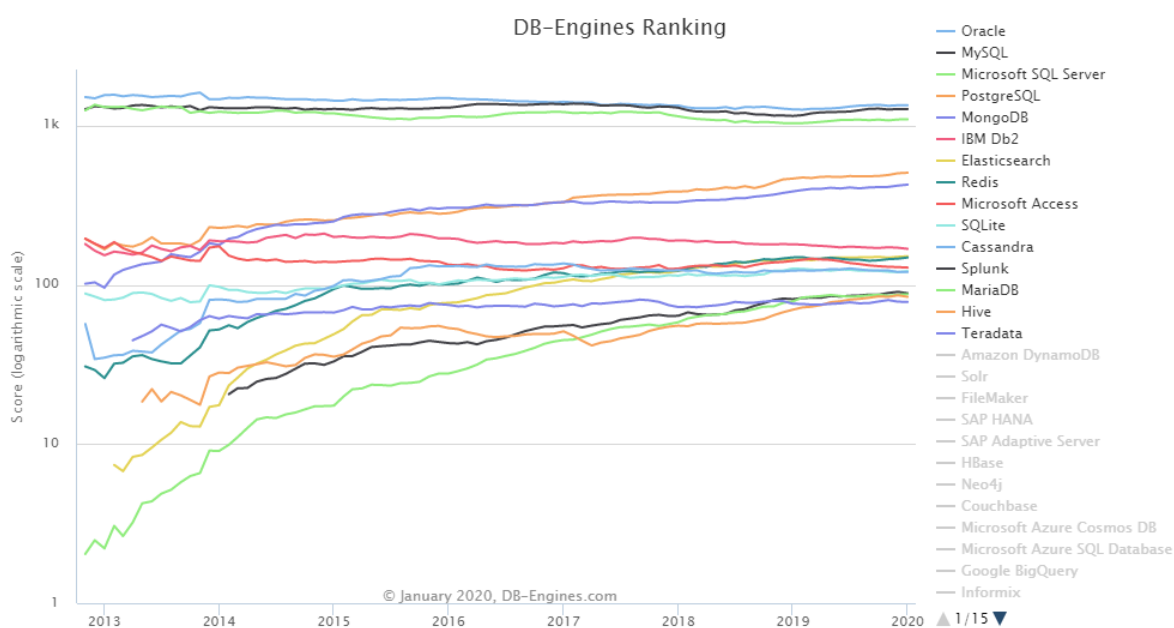


Figure 3: Evolution de la popularité des bases de données⁷

Critère	MySQL	PostgreSQL
Base	SGBD relationnelle	SGBD relationnelle-objet
Extension	N'est pas extensible	Hautement extensible

⁵ SGBD : Système de Gestion de Base de Données

⁶ https://data.sigea.educagri.fr/download/sigea/supports/PostGIS/distance/perfectionnement/M01_rappels_et_concepts_gen_web/co/10_avantages_inconvenients.html

(Consulté le 09/12/2020)

⁷ <https://www.digora.com/fr/blog/TOP-10-des-bases-de-donnees> (Consulté le 08/12/2020)

Sauvegarde	MySQL dump, et XtraBackup fournit une sauvegarde en MySQL	Sauvegarde en ligne
-------------------	---	---------------------

Tableau 3: Comparaison MySQL - PostgreSQL

On a également utilisé d'autres bases de données comme Oracle et MariaDB afin de réaliser l'adaptation à tout type de base de données au sein de la synchronisation.

1.4 Architecture logicielle

En premier lieu, il est nécessaire de préciser que le BICI fait usage de son propre Framework, dénommé APJ. Il consiste en gros à générer automatiquement des listes, des fiches, des pages d'insertion et de modification à partir d'une classe. Le choix explicite de ce Framework est surtout pour la sécurité de l'application et pour faciliter la maintenance.

Ce Framework est fondé sur l'architecture JEE, représenté par la figure ci-dessus, parfaitement adapté au milieu entreprise et suit tout naturellement le pattern Modèle Vue Contrôleur (MVC). Cette architecture se divise en 3 couches :

- Couche de présentation : elle correspond aux écrans et affichages, le dialogue avec l'utilisateur mais surtout l'interaction avec l'utilisateur.
- Couche Métier : elle correspond aux traitements des règles de gestions au sein de l'application
- Couche données : c'est la couche persistante qui est utilisée pour sauvegarder les données.

Pour ainsi élucider la figure, la partie métier assurée par un module EJB ira se connecter à la base de données puis les diffusera à la partie affichage, assurée par les Servlets et les JSP

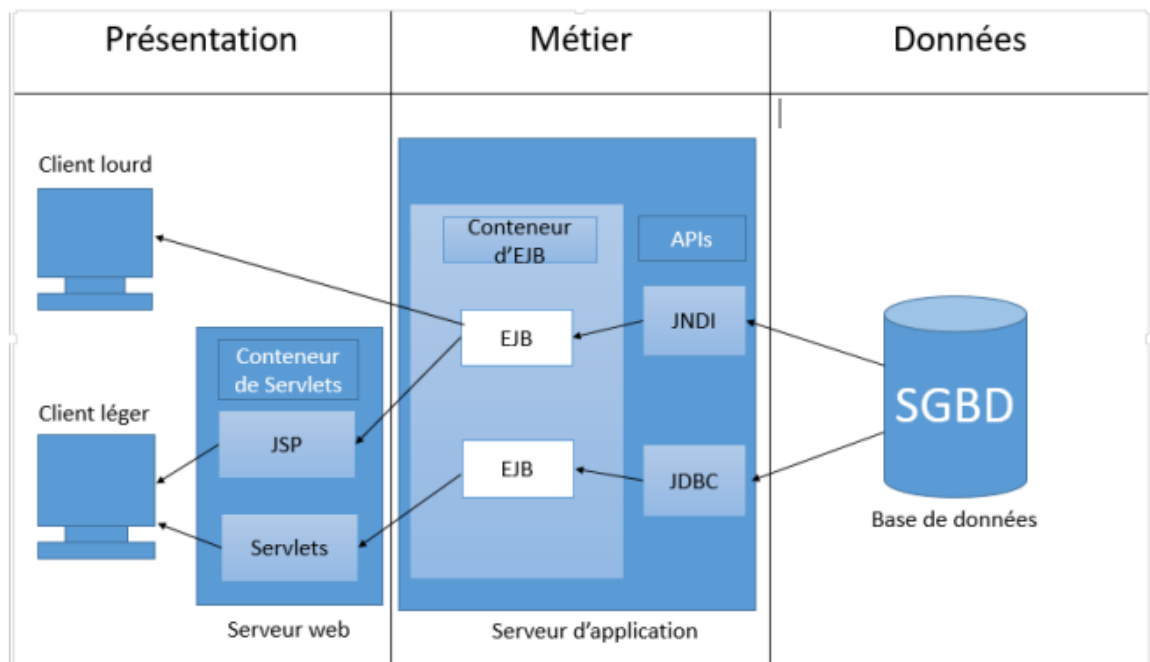


Figure 4: Architecture EJB

2 Réalisation de l'application

2.1 Analyse et conception

Pour pouvoir concrétiser un projet, il est toujours important d'analyser l'existant ou ce qui est déjà en place afin de voir les différents points à éviter et à améliorer dans la concrétisation du projet. Analyser et concevoir une application sous-entend alors une formalisation de toutes les étapes de développement en partant de la demande du client. Cela implique une liste de tâches détaillée suivie des estimations du délai nécessaire à leur accomplissement. Le résultat attendu doit impérativement convenir aux attentes du client en restant fidèle à sa demande. Il est aussi important d'effectuer une bonne conception afin d'analyser les stratégies et techniques à adopter durant la réalisation d'un projet ainsi que d'avoir une vue détaillée sur l'ensemble du projet.

2.1.1 Analyse de l'existant

Avant la réalisation de ce projet, l'entreprise BICI a déjà possédé un système de synchronisation de base de données. Mais ce système déjà en production présente encore des failles comme des problèmes de performances et une complication d'utilisation. Il n'y a pas d'interface pour gérer les serveurs au sein de chaque projet. Comme l'indique la figure

suivante, les développeurs et les responsables de base de données doivent se connecter directement sur le serveur et exécuter des scripts manuellement pour insérer des serveurs distants et configurer les anciens serveurs.

ID	NOM	ADRESSE_IP	UTILISATEUR	MDP	DRIVER	SERVICE_NAME	PORT
1	SERV01	68.26.227.152.245	cededi	cededi	oracle.jdbc.driver.OracleDriver	orcl	1521
2	(null)	(null)	(null)	(null)	(null)	(null)	(n...

Figure 5 : Analyse de l'existant

En effet, il existe de nombreux outils afin de synchroniser les bases de données au sein de deux serveurs distants mais aucun d'entre eux ne permettent de synchroniser deux ou plusieurs types de base de données. L'un d'entre eux est Shareplex. Shareplex est un logiciel de réplication de données qui assure une haute disponibilité. Mais ce logiciel est surtout utilisé pour la synchronisation inter-Oracle mais il n'est pas gratuit.

Ainsi, nous fûmes dans l'obligation de créer une interface afin de gérer les serveurs de chaque projet. Étant donné que le présent projet fut développé avec le Framework APJ, il nous a fallu une légère adaptation aux techniques de développement. A la fin du stage, le projet a fait l'objet de plusieurs améliorations, que nous détaillerons dans les pages suivantes.

2.1.2 Conception de l'application

Passer par la conception est une étape cruciale lors de la réalisation d'un projet car il nous permet d'évaluer tout le chemin à faire, allant des fonctionnalités les plus faciles au plus complexes. La conception représente alors la phase d'analyse du projet. Selon une citation canadienne, « Apprends à réfléchir avant d'agir » (Les proverbes et adages du Canada, 1956), avant la réalisation d'un acte, la réflexion doit toujours être mise en avant. En devinant naturellement notre liste des tâches, elle situe notre travail en nous permettant de faire la distinction entre ce qui a été fait et ce qui ne l'a pas été. Cela nous évite de travailler les yeux fermés et rend la partie développement très facile étant donné que le plus dur a été d'élaborer minutieusement les étapes à brûler. Il ne nous reste alors qu'à faire la traduction au moyen d'un langage de programmation. Bien concevoir permettra

donc une bonne réalisation, car elle permettra la conformité du système réalisé avec les demandes initiales.

Ainsi, pour la réalisation de ce projet nous avons eu recours à travailler avec :

- 9 Tables et 6 Vues
- 15 Classes
- 12 écrans

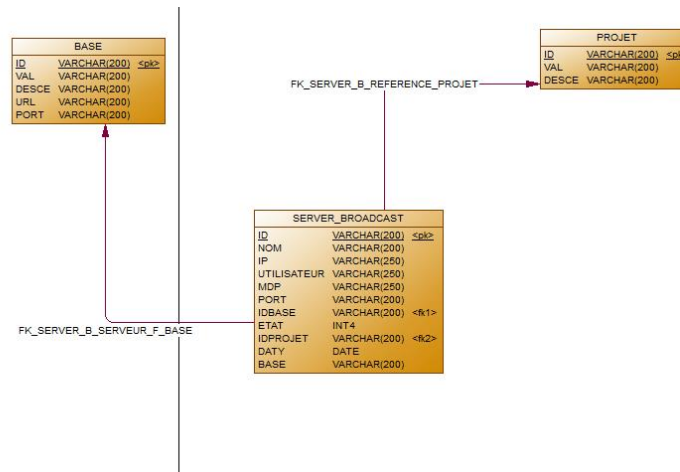


Figure 6 : conception MPD Serveur

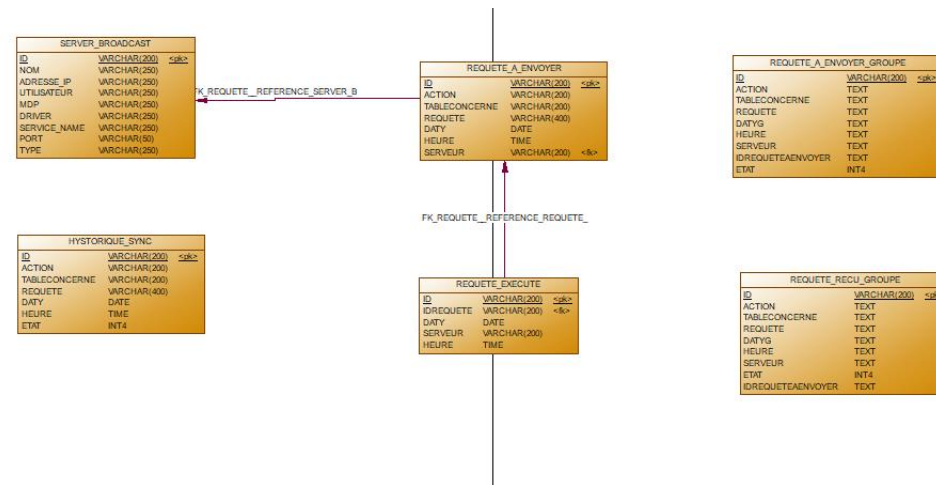


Figure 7: MPD Application à synchroniser

2.2 Développement par fonctionnalité ou module

Ainsi est-il important de tenir en compte des nouvelles modifications apportées. Le tout, résumé dans la figure ci-après, figure représentant le menu du logiciel.

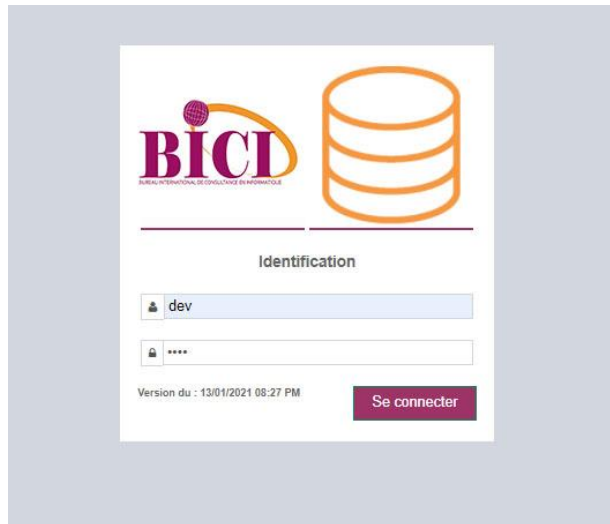


Figure 8:Page de connexion

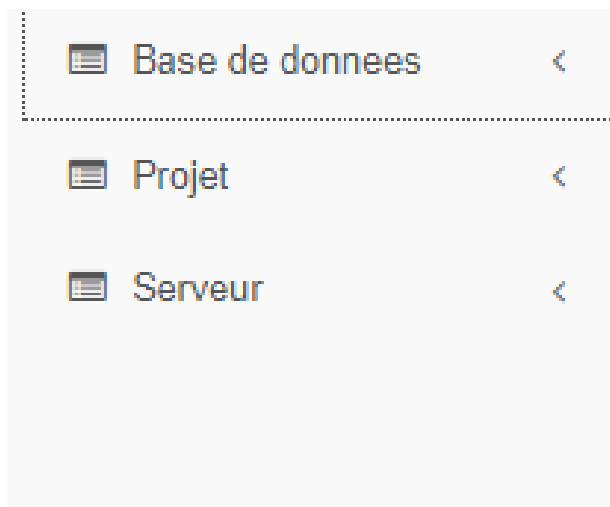


Figure 9: Menu de l'application

2.2.1 Gestion des bases de données et de projets

Cette fonctionnalité permet aux utilisateurs d'insérer un nouveau type de base de données et des projets selon les projets développés au sein du BICI.

2.2.1.1 Saisie et modification d'une base de données

Tout commence par la saisie de base de données. Dans cette partie, on insère les types de SGBD⁸. Par exemple, « PostgreSQL ». Dans la partie Driver se trouve le nom du driver supporte par JDBC⁹. Il est à note que le Driver est un composant logiciel permettant à une application d'interagir avec une base de données. Dans notre cas le driver est *org.postgresql.Driver*. Le port est le port par défaut utilise par le serveur de base de données.

The screenshot shows the BICI Synchronisation Multi-base web application. The interface has a purple header with the BICI logo and the text 'Synchronisation Multi-base'. Below the header is a navigation bar with 'Synchronisation' and a menu icon. On the left is a sidebar with a tree view containing 'Base de données', 'Liste', '+ Creation', 'Projet', and 'Serveur'. The main content area is titled 'Saisie Base de données' and contains a form with four input fields: 'Base', 'Driver', 'URL JDBC', and 'Port par défaut'. Below the form are two buttons: 'Réinitialiser' (grey) and 'Enregistrer' (green). At the bottom right of the page, there is a small copyright notice: '© BICI, 2020 Tous droits réservés'.

Figure 10: Saisie de base de données

Dans le tableau ci-dessous quelques exemples de types de base de données les plus utilise au sein du BICI.

Type Base	Driver	URL JDBC	Port par défaut
Oracle	<i>oracle.jdbc.driver.OracleDriver</i>	jdbc:oracle	1521
PostgreSQL	<i>org.postgresql.Driver</i>	jdbc:postgresql	5432
MySQL	<i>com.mysql.jdbc.Driver</i>	jdbc:mysql	3306

Tableau 4: Exemple base de données

⁸ SGBD : Système de Gestion de Base de données

⁹ JDBC : Java Database Connectivity

Après la saisie d'un type de base de données, l'utilisateur peut consulter ce dernier dans une liste et voir même sa fiche détaille et le modifier.

Liste Base de données

Recherche avancée

TOTAL

	Nombre
Total	2

+ Nouveau

LISTE

id	Base	Driver	Url Jdbc	Port par défaut
BDD000081	Postgres	org.postgresql.Driver	jdbc:postgresql	5432
BDD000101	Oracle	oracle.jdbc.driver.OracleDriver	jdbc:oracle	1521

Nombre de résultat : 2

page 1 sur 1

Exporter

Figure 11: Liste base de données

← FICHE Base de données BDD000081

Id	BDD000081
Base	Postgres
Driver	org.postgresql.Driver
Url Jdbc	jdbc:postgresql
Port par défaut	5432

Modifier

Figure 12: Fiche base de données

2.2.1.2 Saisie de projet

Dans cette partie, les responsables au sein du BICI peuvent inclure les projets développés dans la société. Il est à noter que l'évolution et les tâches de chaque projet sont déjà traitées par une autre application au sein de l'entreprise (« Team Task »). De ce fait ce module est juste à titre instructif afin de gérer les Serveurs distants.

Synchronisation Multi-base

Synchronisation

Base de données < | Projet < | + Creation | Liste | Serveur <

Saisie Projet

Projet:

Description:

Réinitialiser Enregistrer

© BICI, 2020 Tous droits réservés

Figure 13:Saisie de Projet

Après la saisie de projet, on peut consulter la liste des projets, la fiche d'un projet et modifier un projet.

Synchronisation Multi-base

Synchronisation

Base de données < | Projet < | + Creation | Liste | Serveur <

Liste projet

Recherche avancée

TOTAL	
	Nombre
Total	3

+ Nouveau

LISTE		
Id	Projet	Description
PRJ000001	Cemedi	tamatave
PRJ000021	Amadia	Amadia version final
PRJ000041	Spat	Capitaine Tamatave

Nombre de résultat : 3 page 1 sur 1

Exporter

Figure 14:Liste Projet

Synchronisation Multi-base

Synchronisation

Base de données < | Projet < | + Creation | Liste | Serveur <

FICHE Projet PRJ000021

Id	PRJ000021
Projet	Amadia
Description	Amadia version final

Modifier

Figure 15: Fiche Projet

2.2.2 Gestion des serveurs distants

Les serveurs de base de données d'une application peuvent se localiser dans des endroits différents. De ce fait, ce module est une interface pour gérer ces derniers. Afin que ces serveurs puissent se communiquer entre eux, le BICI utilise un VPN (RaminVPN). Ces serveurs sont appelés des « Server Broadcast ». Ce terme est employé pour designer des serveurs en transmission.

2.2.2.1 Saisie d'un Serveur

Dans cette partie, l'utilisateur peut créer un serveur distant selon un Projet. Il est à noter qu'un Projet peut avoir plusieurs Serveurs que l'on va Synchroniser à la suite.

Dans le formulaire, l'utilisateur donne l'adresse IP ¹⁰ du serveur (Son IP sur Radmin VPN), les identifiants de la base de données (utilisateur et mot de passe). Le Port de connexion et le type de base donnée. Au moment du choix de la base donnée, le port est défini par défaut selon le choix, mais on peut la changer selon la configuration du serveur. Le Serveur Broadcast est spécifique selon le projet traité au sein du BICI. Le nom de la base de données est aussi à spécifier au moment de l'ajout du serveur broadcast. Il est primordial de savoir que certaines bases de données n'ont qu'une seule base de données, par exemple, Oracle.

The screenshot displays the BICI web application interface. At the top, the BICI logo is on the left, and 'Synchronisation Multi-base' is centered. A purple navigation bar contains 'Synchronisation' and a menu icon on the left, and 'dev' and 'Déconnexion' on the right. A sidebar on the left lists 'Base de données', 'Projet', 'Serveur', '+ Creation', and 'Liste'. The main content area is titled 'Saisie Serveur Broadcast' and contains a form with the following fields: 'Nom Serveur', 'IP (Radmin VPN)', 'Utilisateur', 'Mot de passe', 'Port', 'Type Base de données' (with a dropdown arrow), 'projet' (with a dropdown arrow), 'Date' (pre-filled with '14/12/2020'), and 'Nom base'. At the bottom of the form are two buttons: 'Réinitialiser' and 'Enregistrer'.

Figure 16:Saisie Serveur

Au moment de saisie d'un Serveur, il est possible de faire des erreurs, pour ce faire, l'option de validation a été intégrée dans ce module. On peut alors consulter une liste avec les filtres

¹⁰ Adresse IP : Identifiant unique attribué à un périphérique relié à un réseau informatique.

selon les projets, et une fiche du serveur pour voir les caractéristiques. Il est à noter qu'au moment de validation d'un serveur, il est impossible de modifier ce dernier.

id	Serveur	IP (Radmin VPN)
Base de données	Projet	Date minimale
Date maximale	Cemedi	

Choix des colonnes à afficher

+

Tri

+

Afficher

Réinitialiser

+ Nouveau


Nombre de résultat : 2

page 1 sur 1

Figure 17:Liste Serveur

2.2.2.2 Validation de Serveur

Dans la fiche d'un serveur, afin de mettre le serveur parmi les serveurs à synchroniser dans un projet, il faut la valider. Dans cette fonctionnalité, dans chaque serveur valide du projet concerne, il y a connexion à distance grâce à RadminVPN, et insertion du serveur dans la table ServerBroadcast de chaque serveur distant. Dans la fiche suivante, se situe le bouton viser afin de valider le serveur, et dans le bout de code suivant se trouve le mode de connexion à distance.

 **FICHE Serveur Broadcast**
SERVBROD000141

Id	SERVBROD000141
Serveur	Oracle
IP (Radmin VPN)	26.227.152.245
Utilisateur	sysdba
Mot de Passe	root
Port	1521
Type Base	Oracle
Etat	CRÉÉ(E)
Projet	Cemedi
Date	06/12/2020
Base de données	orcl

Viser Modifier

Figure 18:Fiche serveur

2.2.3 Synchronisation multi-base

Ce module se situe au niveau de chaque application.

2.2.3.1 Préparation du serveur

Pour synchroniser les bases de données d'un serveur, il faut exécuter un script selon les bases de données.





Nom	Modifié le	Type	Taille
 mysql.sql	14/12/2020 13:46	Fichier SQL	2 Ko
 oracle.sql	14/12/2020 13:46	Fichier SQL	2 Ko
 postgresql.sql	14/12/2020 13:47	Fichier SQL	2 Ko
 SqlServer.sql	14/12/2020 13:47	Fichier SQL	2 Ko

Figure 19:Script à exécuter

Afin d'exécuter et de mettre en œuvre la synchronisation au sein de l'application, il faut que les serveurs soient en réseaux que ce soit en réseau local ou par intranet. De ce fait, l'utilisation d'un VPN est indispensable si les serveurs sont délocalisés.

Cependant, après l'exécution du Script, il faudra installer RadminVPN afin de mettre en réseau le nouveau serveur parmi les serveurs traités au sein du BICI. Un VPN est un système pour créer des relations entre des ordinateurs distants afin d'assurer les trafics et la communication entre ces derniers. Le VPN facilite alors la synchronisation des serveurs distants car il permet de créer un lien direct entre deux ordinateurs distants comme ils étaient sur le même serveur local. On peut également utiliser d'autres VPN car le VPN utilisé ici est surtout utilisé pour Windows. Mais on peut également utiliser d'autres VPN comme NordVPN ou ExpressVPN.



Figure 20: Interface de Radmin VPN

Au sein du projet « Synchronisationsplit », le projet qui se connecte avec le serveur de base de données. Dans le fichier « config.properties », mettre à jour les paramètres de configuration de se connecter avec le serveur base de données.

```
config.properties X
E: > stage > synchronisation > src > java > periodictask > config.properties
1  #Derniere modification: Tue Dec 15 14:07:48 MSK 2018
2  usernameSource=cemedi
3  pwdSource=cemedi
4  ipSource=26.227.152.245
5  serverType=orcl
6  refreshtime=00d:00h:01m:00s
7  driver=oracle.jdbc.driver.OracleDriver
8  serverPort=1521
9  database=oracle
10 urljdbc=jdbc:oracle
11
```

Figure 21: Fichier de configuration synchronisation

Ce fichier de configuration est utilisé pour Synchroniser le serveur de base de données. Dans le tableau suivant, se situe les explications des différentes configurations au sein du fichier.

Attribut	Explication
usernameSource	L'utilisateur pour se connecter avec la base de données
pwdSource	Le mot de passe de l'utilisateur
ipSource	IP ou se trouve le serveur (Adresse Radmin VPN)
serverType	Le nom de la base de données Pour oracle, il n'y a qu'une seule base de données
Refreshtime	Le temps de rafraichissement pour synchroniser les données au sein la base de données
Driver	Le nom du pilote JDBC
serverPort	Le port de connexion pour accéder à la base de données
Database	Le type de base de données (Oracle, PostgreSQL,Mysql,...)
Urljdbc	Le début de la fonction pour se connecter avec la base de données.

Tableau 5: Explication fichier de configuration

Après avoir configuré le serveur, dans le dossier se situent les exécutables suivants :





 sync_build.bat	26/11/2020 09:26	Fichier de comma...	1 Ko
 sync_build.sh	26/11/2020 09:26	Shell Script	1 Ko
 sync_run.bat	26/11/2020 09:26	Fichier de comma...	1 Ko
 sync_run.sh	26/11/2020 09:26	Shell Script	1 Ko

Figure 22: Fichiers de synchronisation exécutables

Ces fichiers sont utilisés pour faciliter le lancement de la synchronisation au sein d'un serveur. Pour ceux des serveurs avec un système d'exploitation Linux ou MacOS, les exécutables se terminent par un .sh et pour le système d'exploitation Windows, un .bat. Pour compiler le projet, on lance le sync_build et pour le lancer, sync_run. Il est à noter qu'on ne lance qu'une seule fois la synchronisation et on laisse le programme s'exécuter seul sans éteindre la fenêtre de la console en cours.

2.2.3.2 Déroulement de la synchronisation

La Synchronisation de base de données au sein d'un serveur se fait en 2 Étapes : le traitement de requête reçu et l'envoi de requête. L'envoi et le traitement des données a été optimiser de façon à minimiser le nombre de connexion à distance au sein de chaque serveur. Ci-dessous se situent le déroulement de la synchronisation et le bout du code pour synchroniser les bases de données.

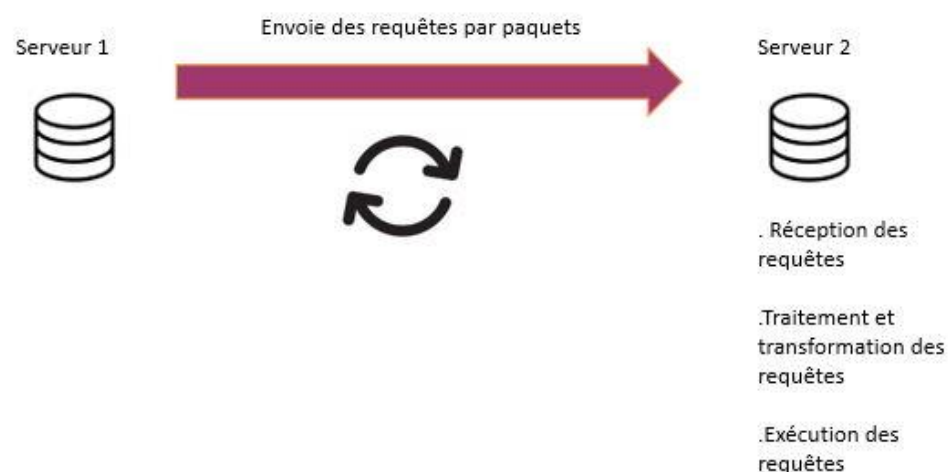


Figure 23: Déroulement de la synchronisation

```

public static void execute() throws Exception {
    Connection con = null;
    ArrayList<Server_Broadcast> servers = null;
    try {
        LogSynchro.println("[INFO]: Début de la synchronisation");
        con = (new UtilDB()).GetConn();
        LogSynchro.println("[INFO]: Reception requetes");
        new RequeteRecuGroupe().traitementRequeteRecu(con);
        ArrayList<HistoriqueSync> requestsToExecute = getRequeteAExecuter(con);
        LogSynchro.println("[INFO]: Reception des requetes recus");
        executeRequeteRecu(con, requestsToExecute);
        LogSynchro.println("[INFO]: Execution des requetes recus ");
        servers = SynchroBase.getServers(con);
        _serv = servers;
        LogSynchro.println("servers tonga"+servers.size());
        for (Server_Broadcast server : servers) {
            if (server != null) {
                Connection connection = null;
                try {
                    connection = server.GetConn();
                    LogSynchro.println("INFO: Connection established [" + server.getNom() + "]");
                    LogSynchro.println("INFO: Enregistrement des operations! Veuillez ne plus interrompre le programme");
                    if (connection != null) {
                        NamedServerConnection connectionRow = new NamedServerConnection(server);
                        connectionRow.setConnection(connection);
                        ArrayList<RequeteAEnvoyer> requests = null;
                        LogSynchro.println("INFO[" + connectionRow.getNom() + "]: recuperation des requetes a envoyer ... ");
                        requests = getRequeteAEnvoyer(con, server);
                        RequeteAenvoyerGroupe requeteAenvoyerGroupe = new RequeteAenvoyerGroupe(requests);
                        envoyerRequeteGroupe(connectionRow, requeteAenvoyerGroupe.grouperRequeteAenvoyer(con), con);
                        LogSynchro.println("INFO[" + connectionRow.getNom() + "]: Enregistrement OK!");
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                    continue;
                } finally {
                    if (connection != null) {
                        connection.close();
                    }
                }
            }
        }
    }
}

```

Figure 24:Code de Synchronisation

Tout d’abord, le traitement des requêtes reçu. Au sein du serveur, une table Hystorique_sync contient les requêtes à exécuter au sein des autres serveurs, c'est-à-dire, les données manquantes dans le serveur à synchroniser. Au début, les requêtes sont groupées par lot et concaténées par un « ; » que l’on va splitter à la suite. Après cela, chaque ligne de requête sera transformée de façon générique à ce que cette dernière s’adapte au type de base de données du serveur. Puis, la requête sera exécutée et l’état de l’hystorique_sync sera mis jour en fait.

En deuxième partie, l’envoi des requêtes, dans la table, REQUETE_EXECUTE se situe les requêtes à envoyer dans les serveurs distants. En premier lieu, on groupe ces requêtes par lot, afin d’optimiser le nombre de connexion à distance au sein des autres serveurs. Puis, pour chaque serveur distant, après connexion, les requêtes sont envoyées et la requête est inséré dans la table REQUETE_EXECUTE afin qu’elle ne soit plus envoyer dans la prochaine synchronisation.

Pour ce projet, le cas réel était un projet en cours au sein du BICI, le système d’information du Centre Médicale du Port Toamasina (CEMEDI).

2.3 Problèmes rencontrés et solutions

Nous avons rencontré différents problèmes durant le stage, il s'agit en premier lieu de problèmes techniques :

- **Manque d'expérience en Git** : quand nous avons commencé le développement nous avons eu du mal à comprendre et nous familiariser avec les bonnes pratiques de l'entreprise sur l'utilisation de Git, mais le problème a été réglé avec l'aide des autres collaborateurs.

Les problèmes techniques ont été réglés au fur et à mesure, mais on aussi rencontre des problèmes au niveau de la base de données :

- **Manque de pratique pour les données de grande taille** : les types utilisés pour de grandes données n'ont pas été traités au sein de l'entreprise, surtout par le moteur du Framework APJ. De ce fait, une adaptation et modification au sein du moteur a été faite.
- **Clé primaire identique au sein de 2 serveurs différents** : les séquences au sein de 2 serveurs peuvent être incrémentées de façon à ce que les clés primaires des tables soient les mêmes, cela posera problème au sein de la synchronisation. Mais le problème a été géré en asynchrone grâce à l'heure, la date et le serveur. La clé primaire a été alors modifiée selon ces paramètres.

2.4 Budget du projet

Afin de mettre en œuvre la Synchronisation au niveau d'une application, le tableau ci-dessous montre le budget nécessaire.

Désignation	Quantité	Prix Unitaire	Catégorie	Total
VPN	2	360 000,00	Logiciel	720 000,00
Serveur	2	2 500 000,00	Matériel	5 000 000,00
Connexion internet	1	249 000,00	Matériel	249 000,00
Total		5969000		

Tableau 6: Budget du projet

3 Évaluation du projet et connaissances acquises

3.1 Bilan pour l'entreprise

Le développement de faciliter la synchronisation multi-base a permis à BICI de synchroniser les bases de données des projets développés au sein de l'entreprise. La méthode utilisée consiste à synchroniser les n'importe quel type de base de données en envoyant les données par paquets afin de minimiser les risques. La synchronisation gère également l'asynchrone au sein des bases et devient plus performant.

3.2 Bilan personnel

Pour notre part, ces trois mois de stage n'ont pu qu'être bénéfiques. Non seulement nous avons eu la chance d'apprendre les méthodes de travail en milieu entreprise mais nous avons également pu découvrir le mode de vie à adopter. Nous n'avons pu qu'améliorer notre façon de développer et d'en faire presque un automatisme. La gestion du temps fut par ailleurs une autre notion que nous nous sommes forcés de maîtriser vu que parallèlement à notre stage, nous avons travaillé sur tous les projets courants au sein de l'entreprise (Gestion de facturation de l'entreprise AMADIA, Gestion de l'ordre du menu du site www.ketrika.com , Gestion de stock de CEMEDI (Centre Médicale du Port) Tamatave, Gestion de commande et de livraison de l'application de Phô-Resto, etc.). Et nous tenons particulièrement à préciser que nous avons eu la chance de développer, de présenter et même de déployer le module « Hospitalisation » pour AMADIA, module servant à faire le suivi de l'hospitalisation et des consultations au sein de l'AMADIA.

3.3 Extension et évolution de l'application

Toujours dans un esprit évolutif, et pour améliorer l'efficacité de l'application, un déploiement en ligne est envisageable, afin de ne pas dépendre d'un VPN mais en utilisant seulement des web services.

Conclusion

Ainsi, après avoir effectué un stage de fin d'études de Licence en Développement d'applications au sein de BICI Madagascar, nous avons pu mettre en pratique toutes les connaissances théoriques acquises durant la formation à l'IT University, tout en étant confronté aux difficultés réelles du monde du travail.

On avait pour objectif la gérance des serveurs et de la synchronisation de la base de données à partir d'une interface tout en gérant la synchronisation au sein des logiciels développés dans l'entreprise. A présent, les serveurs des applications des serveurs au sein du BICI sont gérés par une interface web. Tout d'abord, pour la gestion des bases de données, des projets et des serveurs, nous pouvons maintenant consulter leur fiche, leur liste, et même en insérer un nouveau à partir d'une interface. Puis, les serveurs sont mis en réseaux grâce à un VPN, dans le cas réel, nous avons utilisé RadminVPN. Après la validation d'un serveur distant, il faudra le configurer et lancer la synchronisation à l'aide d'un fichier exécutable. Cette synchronisation a été optimisée de façon à ce que les requêtes soient envoyées par paquets. Nous pouvons dire alors que les objectifs sont atteints et l'entreprise est satisfaite de notre travail.

Lors du développement d'un système d'information pour le Centre Médical du Port, une synchronisation a été nécessaire car ils utilisaient deux serveurs différents. Confrontés à des problèmes de performance, et de type de base de données, une synchronisation optimale et multi-base a été nécessaire. L'application a été utilisée dans ce cas réel, celle de l'application de CEMEDI (Centre Médical du Port) Tamatave.

Ce stage a été très enrichissant, car il a permis d'améliorer les compétences techniques, fonctionnelles et relationnelles. De plus, c'est très plaisant de travailler chez BICI Madagascar : je me suis bien intégré dans l'équipe, tout le monde se respecte tout en étant ouvert à chacun et l'entreprise tient vraiment à mon évolution en termes de compétences.

Toujours dans une vision évolutive, ce projet n'est pas à l'abri d'une constante amélioration dont l'installation d'un module pour les clients du BICI afin de gérer les services après-vente des applications.

Bibliographie

- <https://www.quest.com/fr-fr/products/shareplex>
Présentation de Shareplex. Consulté le 01/11/2020
- https://fr.wikipedia.org/wiki/Architecture_trois_tiers
Architecture trois tiers. Consulté le 24/11/2020
- <https://support.claris.com/s/article/Synchronisation-des-bases-de-donn%C3%A9es-un-aper%C3%A7u-des-approches-1503692967597?language=fr>
Synchronisation de base de données. Consulté le 12/12/2020
- http://www.journaldunet.com/solutions/0508/050826_synchronisation.shtml
5 Points clés de la synchronisation de données. Consulté le 07/01/2021
- <https://www.lemondeinformatique.fr/toute-l-actualite-marque-sur-postgresql-456.html>
Actualités sur PostgreSQL. Consulté le 25/01/2021
- <https://www.vocalcom.com/fr/blog/limportance-base-de-donnees>
Base de données : une nécessité pour l'entreprise. Consulté le 27/01/2021
- <https://openclassrooms.com/fr/courses/1734211-analysez-et-gerez-des-risques-si/6477076-identifiez-les-risques-si>
Identification des risques SI. Consulté le 01/02/2021