**Republic of Cameroon**

**Peace – Work – Fatherland**

**------------------------------------------------------------**

**University of Yaounde I**

**Sapienta – Collativia – Cognitio**

**------------------------------------------------------------**

**National Advanced School of Engineering**

**------------------------------------------------------------**

**Department of Computer Engineering**

**------------------------------------------------------------**

**République du Cameroun**

**Paix – Travail – Patrie**

**------------------------------------------------------------**

**Université de Yaoundé I**

**Sapienta – Collativia – Cognitio**

**------------------------------------------------------------**

**Ecole Nationale Supérieure Polytechnique**

**------------------------------------------------------------**

**Département de Génie Informatique**

------------------------------------------------------------

# 

**Encadreur** :

Dr BATCHAKUI/ M. ARMEL NGATCHUI

**RAPPORT PORTANT SUR LE PROJET COMPUCLINIC : REPORTING**

**Reporting : Assurer la gestion des Sauvegardes, Historiques et Statistiques diverses**

**Classe** : 4GI

**Membres du groupe** :

* KAMENI NGANGUE Elie Alex
* TIKENG NOTSAWO Pascal Junior
* TEDONZE NICK JOFREIN

**Année scolaire** : 2019-2020

# SOMMAIRE

[0](#_Toc30254891)

[SOMMAIRE 1](#_Toc30254892)

[Introduction 2](#_Toc30254893)

[I- Problématique 2](#_Toc30254894)

[II- Rappel des objectifs à atteindre 2](#_Toc30254895)

[III- Rappel des fondamentaux 3](#_Toc30254896)

[IV- Présentation de la solution envisagée 3](#_Toc30254897)

[V- Diagrammes d’analyses, de conceptions et de prototypages de notre module 4](#_Toc30254898)

[VI- Implémentation 10](#_Toc30254899)

[VII- Présentation des résultats de l’implémentation 11](#_Toc30254900)

[VIII- Déploiement de notre module 13](#_Toc30254901)

[IX- Difficultés rencontrées 13](#_Toc30254902)

[X- Améliorations sur le projet 14](#_Toc30254903)

[Conclusion 14](#_Toc30254904)

[WEBOGRAPHIE - BIBLIOGRAPHIE 14](#_Toc30254905)

# Introduction

Fondamentalement, le reporting consiste à collecter des données à partir de différentes sources et les présenter de manière compréhensible afin qu’elles soient prêtes à être analysées. Il consiste aussi à présenter des données et des informations, de sorte qu’il inclut également une analyse, c’est-à-dire que l’audience finale puisse à la fois voir et comprendre les données, et surtout prendre des décisions d’action en conséquence. Il est question pour nous dans ce travail de mettre en place le reporting dans le cadre du projet COMPUCLINIC, ceci à travers la gestion des sauvegardes, des historiques et des statistiques diverses. Pour ce faire, nous présenterons d'abord nos spécifications fonctionnelles, ensuite nous analyserons et concevons notre solution (Uses Cases, Scénarios de base, Diagramme de séquence, prototypage des interfaces, listing des objets identifiés, diagramme de classe, schéma logique des données…) et nous implémenterons notre solution.

# Problématique

Le Dans le cadre du projet notre outil doit être capable :

* D’effectuer la sauvegarde des données des différents objets de notre application
* De gérer les l’historique sur les sauvegardes des données de notre application.
* De gérer les statistiques sur les sauvegardes des données dans notre outil historique.
* De garder une trace (sauvegarde) de l’historique des modifications des données

# Rappel des objectifs à atteindre

Le but visé ici est de produire un module permettant d’effectuer la sauvegarde des données de notre application, de garder une trace des différentes données modifiées pendant l’utilisation de notre application et enfin de faire des statistiques sur les variations de notre base de données

# Rappel des fondamentaux

Yii est un framework d'applications Web PHP pour MVC open source, orienté objet et basé sur des composants. Yii se prononce comme "Yee” et en chinois cela signifie "simple et évolutif" et il peut s'agir d'un acronyme pour "Yes It Is!".

Le framework Yii utilise également les dernières fonctionnalités de PHP, telles que les traits et les espaces de noms.

La structure Yii a une architecture à base de composants et une prise en charge complète de la mise en cache. Par conséquent, il convient à la construction de tout type d’applications Web : forums, portails, systèmes de gestion de contenu, services RESTful, sites Web de commerce électronique, etc.

# Présentation de la solution envisagée

La sauvegarde par défaut est assurée par la classe **yii\db\ActiveRecord**.

Cette classe nous permet d’effectuer les opérations comme :

* Insert (Inserer)
* Delete (Supprimer)
* Update (Mise à Jour)

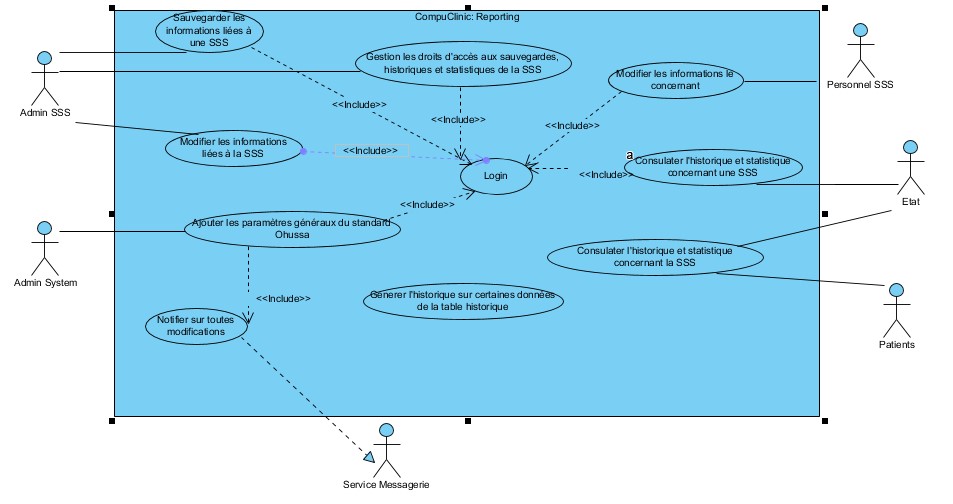
Sauf que dans le cadre de notre projet, toutes les opérations définies plus haut doivent s’accompagner d’un ajout d’une nouvelle ligne dans la table historique.

Nous avons donc crée la classe **CompluClinicRecord** qui elle va étendre la classe **ActiveRecord** et faciler les opérations spécifiées dans les spécifications.

En ce qui concerne l’historique, nous avons ajouté la classe **Historique** permettant la sauvegarde des variations de la base de données.

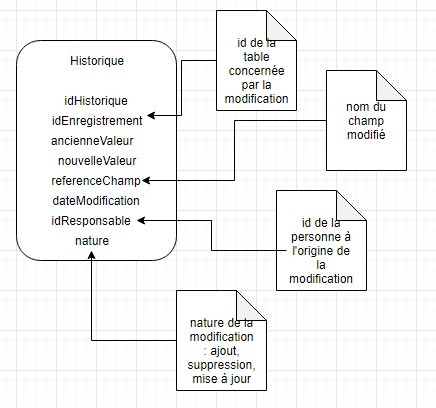
# Diagrammes d’analyses, de conceptions et de prototypages de notre module

* 1. **Analyse**
     1. **Listing des objets identifiés**
* Les codes de fonctions (racines, préfixes et suffixes) et les redicielles fonctionnelles
* Les codes de poste et les codes de badge
* Les profils critiques ainsi que les fonctions et postes associées
* Utilisateurs et groupes d'utilisateur
* Socle standard
* Paramétrage et renseignements de la SSS
  + 1. **Scénarios de base**
* Consulter l’historique d’un des objets identifiés.
* Faire des modifications (ajouts, suppressions, mise à jour)
* Consulter les statistiques des mois/semaine/jours antérieurs
* Configuration des droits d'accès aux informations sauvegardées
  + 1. **Uses cases**



* + 1. **Schéma logique de données**
* Gestion de l’historique sur les données

Nous écoutons les évènements qui surviennent lorsque nous tentons de faire une sauvegarde.



* Gestion des sauvegardes

En ce qui concerne les sauvegardes nous allons utiliser la classe CompuCliniRecord.

Cette classe étant la classe **ActiveRecord.**

* Gestion des statistiques

Une classification du personnel en fonction des compétences, des postes de travail, de la grille salariale... Cette statistique s’effectue sur la table personnel (user) et génère un histogramme sur les utilisateurs.

Une comparaison des modifications du mois/semaine actuel par rapport aux autres (ce qui a été supprimé, ajouté, modifié ou consulté).

Cette statistique s’effectue sur la table des historiques. Permet de visualiser sous forme de schéma plus explicite les différentes modifications les données.

Une évaluation du niveau infrastructurel des SSS

C’est une classification des SSS selon le standard OHUSSA. Ce diagramme nous permettra de visualiser les différentes structures en classe (classe 1 à 3 et pour chaque classe une représentation en sous classe) selon une zone de couverture.

Evaluation des stocks d’alerte

Nous présenterons dans cette partie les différentes variations des entrées et des sorties pour une SSS donnée (Les médicaments critiques, le nombre de décès…)

* 1. **Conception**
     1. **Diagrammes de classe**

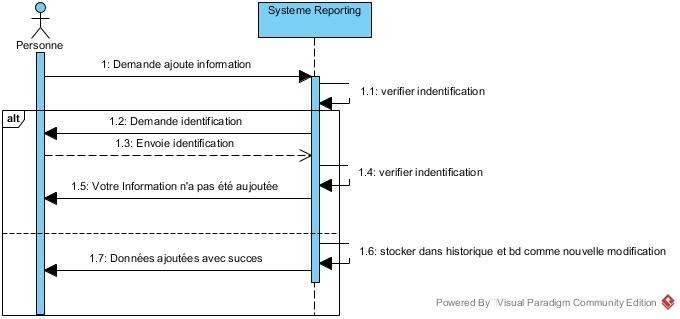
En ce qui concerne les statistiques, nous parlerons plutôt de service. Ce sont des classes qui interagissent sur les données présentent dans la base de données afin de générer les différents diagrammes nécessaires.

Nous utilisons le Framework **kool-report (**voir implémentation plus bas dans le document**)** pour la visualisation de nos différentes statistiques.

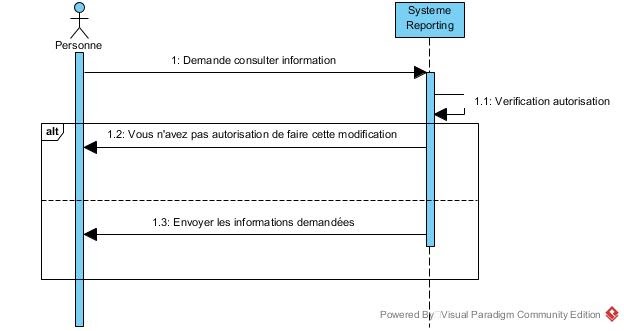
En ce qui concerne l’historique, nous utiliserons la classe historique présentée plus haut.



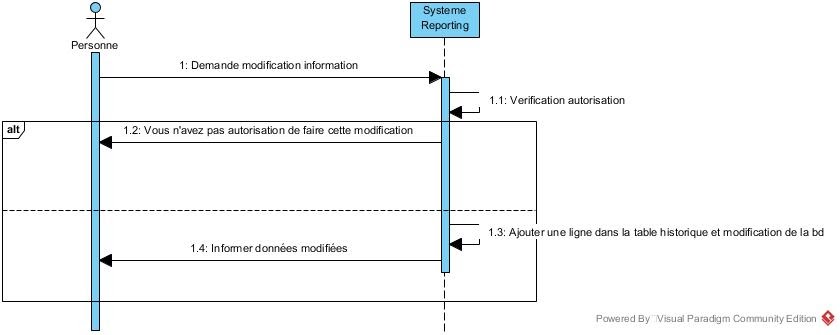
* + 1. **Diagrammes de séquences**
* **Ajouter des informations**

****

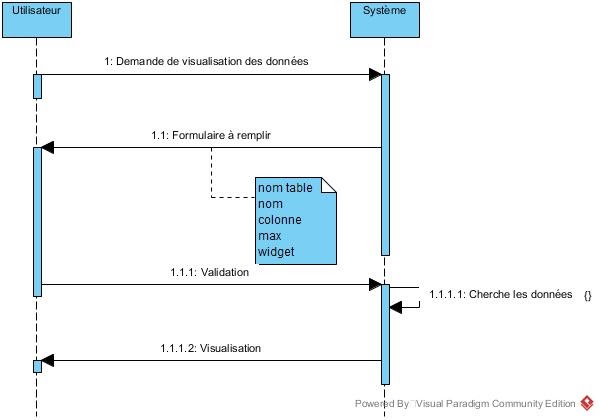
* **Consulter une information**

****

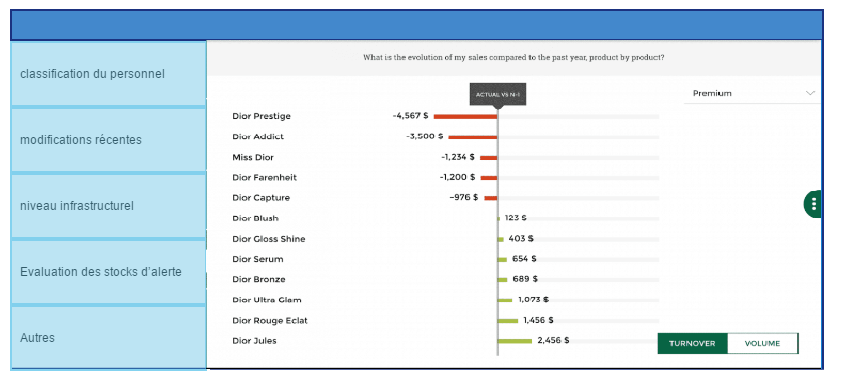
* **Modifier une information**

****

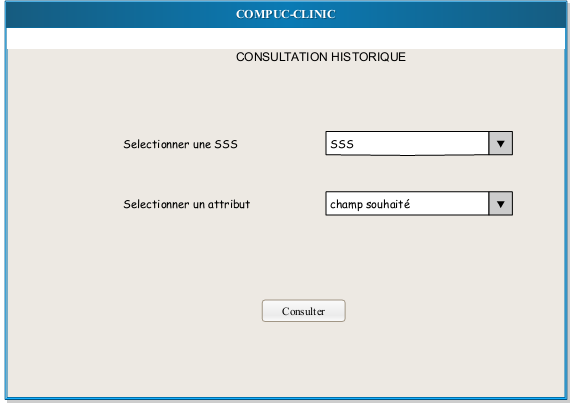
* **Visualiser les données**

****

* **Visualisation statistique**

****

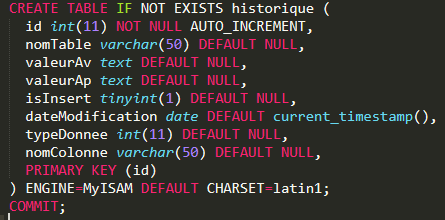
* **Consultation historique**

****

# Implémentation

1. **Module de de gestion des sauvegardes et de l’historique**

* Dans la phase d’implémentation pour assurer la sauvegarde des données nous nous sommes principalement servir de la classe ActiveRecord qui gère l'insertion(insert), la suppression(delete) et la mise à jour(update) des données dans notre application. Ces opérations fournies par le framework Yii répond à nos attentes en ce qui concerne les sauvegardes.
* En ce qui concerne l’historique, il est question de garder une trace dans la table historique décrite plus haut des modifications effectuées dans la base de données. Le code de création de cette table donne :



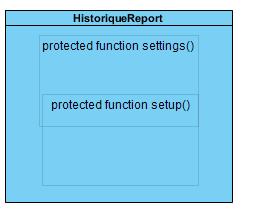
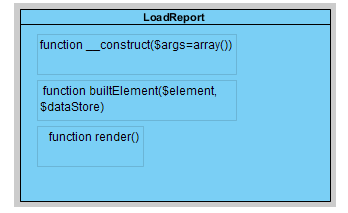
Pour assurer les fonctions liées à la gestion de l’historique nous avons créés la classe CompuClinicRecord qui étant la classe **ActiveRecord** cela ne nous dérange pas car nous n'envisageons pas effectuer l’historique de la table historique. L’interface de notre classe redéfinie les méthodes et utilise les objets suivants :

* Aftersave Méthode appelée juste après une insertion ou la mise à jour a été faite avec un comme paramètre
* **$insert :** Permet de savoir si c'est une insertion ou une mise à jour
* **$changedAttributes :** c’est un tableau contenant la liste des colonnes modifiées avec leurs valeurs (valeurs avant la modification)
* **getOldAttributes:** Elle nous permet d'avoir les attributs et valeurs de la table étendant *CompuClinicRecord*
* **AfterDelete** : Méthode appelée juste après une suppression de données dans la bd.

**On peut consulter le code associé à cette classe depuis le lien github fourni avec ce document y compris la description plus détaillée des différents codes contenue dans le fichier readme.md. Outre ces méthodes propres aux spécifications nous avons jugé nécessaire d’y ajouter un service de messagerie visant à aviser l’administrateur s’il avérait que des données sensibles soient modifiées. Cela se traduit par l’ajout de la méthode sendMail qui lit les paramètres mail de l’administrateur.**

1. **Module de reporting et des statistiques**

Dans la phase d’implémentation de ce module nous avons utilisés la bibliothèque koolreport qui est sert à faire du reporting php. Nos rendus graphiques s'appuient uniquement sur la table historique qui rend compte des modifications effectué sur la base de donnée. Le diagramme de classe est le suivant :



# Présentation des résultats de l’implémentation

Page d’acceuil pour faire du reporting.



Historique sur les structures de santé en abscisse et des différetes entrées et sorties pour chaque structure.



Reporting sur l’état de santé des différents patients sur la somme totale pour chaque cas après une consultation



# Déploiement de notre module

Pour pouvoir utiliser notre module, il faut :

* Il installer SGBD MySql, installer composer et puis Yii sur votre machine
* Configurer l’accès à la base de données depuis le fichier

**-chemin\_du\_projet-/config/db.php**

* Exécuter la commande

**php yii serve**

* Cliquer sur le lien de la page d’accueil et commencer la navigation.

# Difficultés rencontrées

Les principales difficultés que nous avons rencontrées sont :

* Les incompréhensions des spécifications du projet
* Absence de communication entre les différents modules du projet, ce qui a rendu le test de l’application avec les données concrètes plus difficiles.

# Améliorations sur le projet

Afin d’améliorer notre projet, nous pouvons par exemple :

* Intégrer la possibilité de notifications de l’administrateur, des utilisateurs de l’application
* Limiter l’accès à certaines tables de notre base de données.

# Conclusion

Pour la réalisation de notre projet où il était de question de concevoir et de développer les composants pouvant assurer la sauvegarde, la gestion de l’historique et du reporting des données dans le cadre du projet COMPUCLINIC; nous avons tout d’abord décrire et formaliser les différentes spécifications propres à chaque sous module puis nous avons effectués une analyse conceptuelle d’une solution aux spécifications qui en découlent et enfin nous avons réalisés des prototypes de nos interfaces ainsi qu’un compte rendu de leur implémentations. Notre outil n’est certes pas un élément far du projet mais reste un atout majeur pour l’exploitation et l’analyse des données récoltées par CompuClinic

# WEBOGRAPHIE - BIBLIOGRAPHIE

[1] [https://www.yiiframework.com/doc/guide/2.0/fr](https://www.researchgate.net/publication/275343154_Les_bases_de_donnees_dans_les_reseaux_de_capteurs_sans_fil)

[2] [**https://www.koolreport.com/**](https://www.koolreport.com/)

[3] [**https://www.jinfonet.com/resources/bi-defined/php-reporting/**](https://www.jinfonet.com/resources/bi-defined/php-reporting/)

[4] <https://www.php.net/docs.php>

[5] <https://getbootstrap.com/docs/4.4/getting-started/introduction/>