

Document d’Architecture Technique de l’ERP de gestion du centre d’appel

MEDIA CONTACT

*Document d’Architecture Technique*

***Version : 1.0***

|  |  |
| --- | --- |
| Par :  **Direction des programmes** | À  ***La Direction Générale*** |

Janvier 2022

Sommaire

[Introduction 3](#_Toc93572218)

[1. Gestion des Modules 4](#_Toc93572219)

[1.1. Spécifications fonctionnelles 4](#_Toc93572220)

[1.2. Spécifications techniques 5](#_Toc93572221)

[1.2.1. Couches applicatives 5](#_Toc93572222)

[1.2.2. Choix technologique 5](#_Toc93572223)

[1.3. Diagramme de classe 8](#_Toc93572224)

[1.4. Planning d’exécution 10](#_Toc93572225)

[2. Tableau de bord de la production (à venir) 11](#_Toc93572226)

[3. Module Zeus Rh (à venir) 11](#_Toc93572227)

[4. Planning (à venir) 11](#_Toc93572228)

# Introduction

Pionnier de la gestion des relations clients, le Groupe Media Contact est un centre qui facilite le contact entre les entreprises et leurs clients. Dans l’exercice de ses activités, l’entreprise utilise des solutions d’autres éditeurs et développe elle-même ses propres solutions.

A ce jour (mercredi 19 janvier 2022), l’entreprise s’est retrouvée avec une vingtaine d’applications dont certaines sont obsolètes. Dans le but de faire une refonte des applications existantes et de les regrouper en une seule plateforme, nous avons été contacté par la direction générale pour concevoir et développer une application extensible qui regroupe toutes les applications du groupe.

Il est question de développer une ERP de gestion de centre d’appel dont les fonctionnalités sont regroupées par module.

Ce document présente par module :

* Les spécifications fonctionnelles ;
* Les spécifications techniques ;
* Et le planning d’exécution

# Gestion des Modules

La gestion des modules constitue le cœur de l’ERP. Elle permet de gérer l’ajout de nouvelles fonctionnalités, la gestion des utilisateurs, et profiles. Cette section décrit l’architecture technique de la gestion des modules.

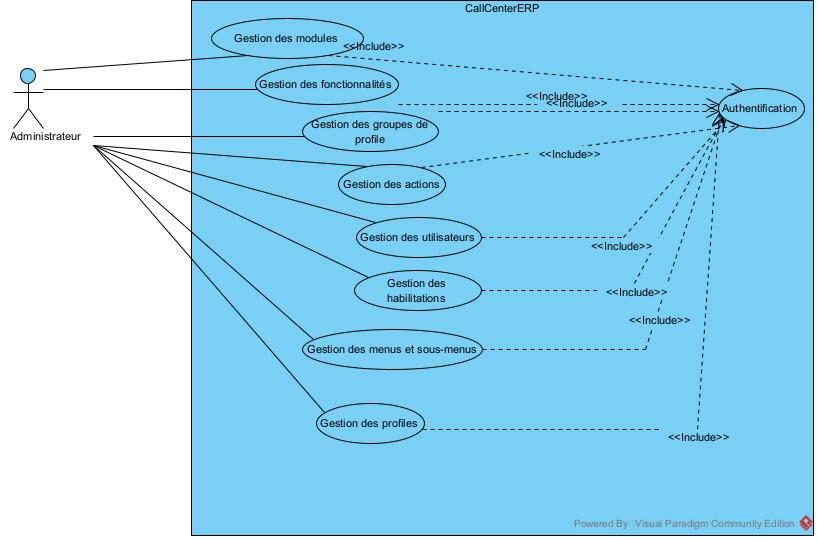
## Spécifications fonctionnelles

* + Identification des utilisateurs et des cas d’utilisations

|  |  |
| --- | --- |
| Acteurs | Cas d’utilisation |
| Administrateur | Gestion des modules (Ajouter, Consulter, Modifier, Supprimer, Restaurer)  Un module peut être : Paie, Ares, Morphe etc… |
| Gestion des Fonctionnalités, une fonctionnalité fait référence à un Controller applicatif et peut être UserControler, FinanceController. |
| Gestion des actions (Enregistrer, Modifier, Supprimer, Imprimer) |
| Gestion des profils d’utilisateurs |
| Gestion des utilisateurs |
| Gestion des habilitations : Une habilitation définie le droit d’un profile sur un module et précise le Controller et l’action auquel il a droit. |
| Gestion des menus et sous menus |
| Utilisateur | Utiliser un module selon ses habilitations |

* + Diagramme de cas d’utilisation

Le diagramme de cas d’utilisation relatif au module de gestion des modules se présente comme suit :



* + Contraintes applicatives

1. Chaque module doit rendre disponible un fichier présentant la liste de ses menus, sous-menus et Controllers ;
2. À l’enregistrement d’un module, tous ses composants (Menus, sous-menus et Controllers  sont automatiquement enregistrés à partir d’un fichier JSON ou texte.

## Spécifications techniques

### Couches applicatives

Le développement se fait sur trois couches :

* la couche le front-end
* la couche back end
* la couche base de données

### Choix technologique

Pour réussir chaque couche, plusieurs possibilités s’offrent. Une analyse de chacune de ces possibilités nous permet de faire faire un choix judicieux.

* La Couche front-end

|  |  |
| --- | --- |
| Sujet | *Contenu* |
| Décision d’architecture | *DA 0001 : Choix du Framework à utiliser au niveau de la vue* |
| Problématique | *Quelle technologie utilisée au niveau du front-end ?* |
| Hypothèse | *La couche vue, est détachée de la couche back end.* |
| Alternative | * *Option1 : Vue JS* * *Option2 : Angula JS* * *Option3 : React JS* |
| Décision | *Option1* |
| Justification | *Vue JS est léger, rapide et simple* |
| Implication | *La connaissance du Framework par le développeur.* |

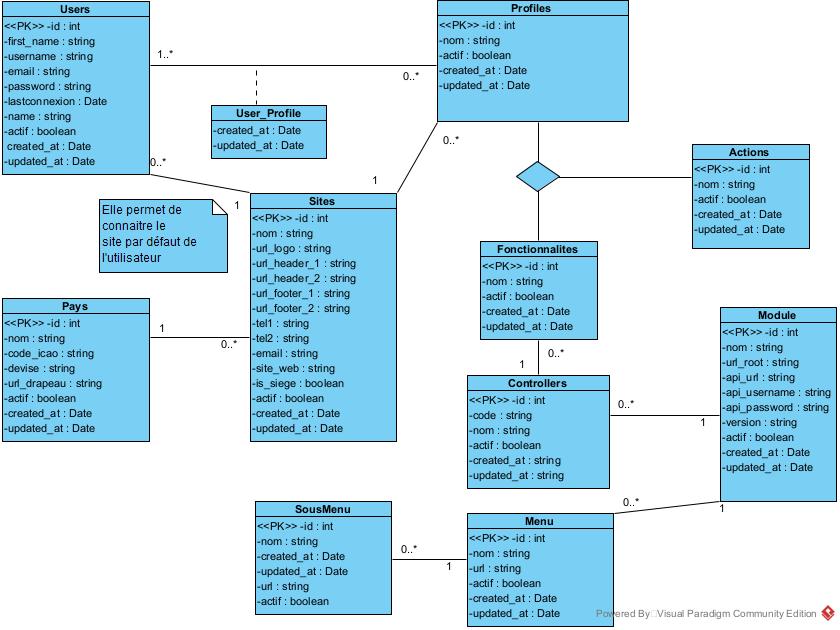
* La couche back end

|  |  |
| --- | --- |
| Sujet | *Contenu* |
| Décision d’architecture | *DA 0002 : Choix du Framework à utiliser au niveau back end* |
| Problématique | *Quelle technologie utilisée au niveau du back end ?* |
| Hypothèse | *La couche vue, est détachée de la couche back end.* |
| Alternative | * *Option1 : Laravel* * *Option2 : Node JS* * *Option3 : Laravel ET Node JS* |
| Décision | *Option3* |
| Justification | *En se basant sur le fait que Laravel permet un développement rapide et prenant en compte le fait qu’il est basé sur PHP qui ne permet pas une gestion Asynchrone, il sera appuyé par Node JS notamment au niveau des modules ou nous avons besoin d’une gestion asynchrone.* |
| Implication | *La connaissance du Framework par le développeur.* |

* La couche base de données

|  |  |
| --- | --- |
| Sujet | *Contenu* |
| Décision d’architecture | *DA 0003 : Choix du SGBD* |
| Problématique | *Quelle SGBD choisir ?* |
| Hypothèse | *L’entreprise s’inscrit dans une dynamique de réduction de coût en optant pour des solutions robustes et opens sources* |
| Alternative | * *Option1 : MySQL* * *Option2 : PostgreSQL* |
| Décision | *Option2* |
| Justification | *PostgreSQL est un SGBD qui supporte de l’objet comparativement à MySQL* |
| Implication | *Un effort effort de recherche* |

## Diagramme de classe

Le diagramme de classe relatif au module de gestion des applications se présente comme suit : 

## Planning d’exécution

Liste des taches avec les délais d’exécutions

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Jour | *Semaine 1* | | | | | *Semaine 2* | | | | | *Semaine 3* | | | | | *Semaine 4* | | | | |
| Taches | *1* | *2* | *3* | *4* | *5* | *1* | *2* | *3* | *4* | *5* | *1* | *2* | *3* | *4* | *5* | *1* | *2* | *3* | *4* | *5* |
| Analyse et Conception |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Front end  choix Template |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Configuration avec Git pour le travail collaboratif |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Développement Back end  Dev. des services (api) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Développement Front end  Dev. des services (api) |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Tests et Corrections |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# Tableau de bord de la production (à venir)

# Module Zeus Rh (à venir)

# Planning (à venir)

Conclusion

Ce document présente les spécifications fonctionnelles et techniques des différents modules nécessaires à un l’ERP de gestion d’un centre d’appel. Cette première version illustre la manière dont les modules interagissent avec les utilisateurs.