PROJET DE FIN D’ÉTUDES

***Présenté en vue d’obtenir le***

DIPLÔME D’INGÉNIEUR D’ÉTAT

En GÉNIE INFORMATIQUE

Par *NAINIA Youness*

**Application Desktop d’un système de gestion de paie**

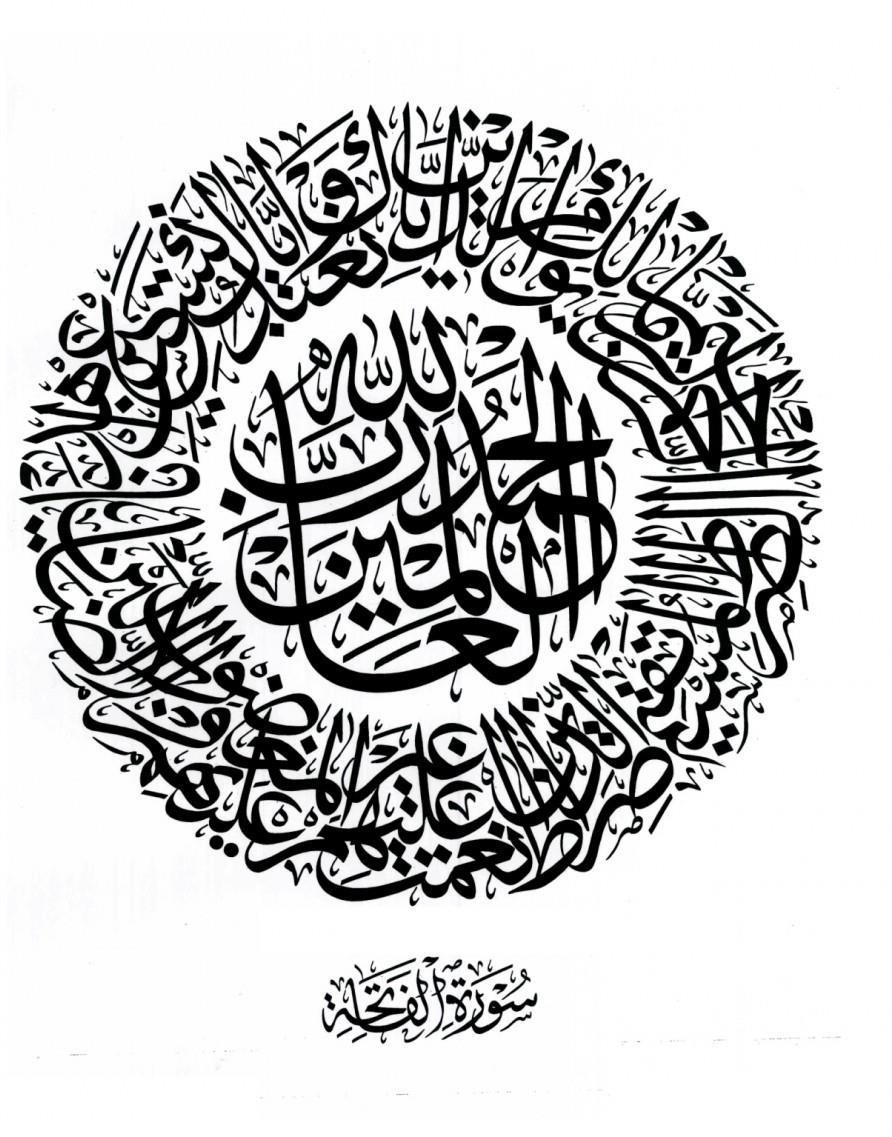
Les travaux relatifs au présent PFE ont été réalisés auprès de

**SAPRESS, Casablanca**

**Encadrant SAPRESS: Encadrant ENSAM:**

Mr. Ahmed ABRARAY Mr. ZAKRANI

2022-2023



## Dédicaces

***A ma mère****, qui m'a comblé de son soutien et m'a voué un amour inconditionnel. Tu es pour moi un exemple de courage et de sacrifice continu. Que cet humble travail témoigne mon affection, mon éternel attachement et qu'il appelle sur moi ta continuelle bénédiction.*

***A mon père****, Aucune dédicace ne saurait exprimer l’amour, l’estime, le dévouement et le respect que j’ai toujours eu pour vous. Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.*

***A mon très cher frère et mes chers cousins****, je vous remercie pour votre amour, votre soutien et vos encouragements*

***A toute ma famille****, pour leurs soutiens, leurs conseils partagés*

***Au staff professoral du Groupe Edito****, je serais vaniteux si je me devais énumérer en ces quelques lignes vos remarquables qualités humaines et professionnelles. Veuillez trouver ici l'expression et le témoignage de ma gratitude ressentie*

***A tous mes chers amis****, pour le soutien que vous m’aviez offert, je vous*

*dis MERCI*

## Remerciements

*Avant de commencer à rédiger mon rapport, je tiens à adresser mes sincères remerciements aux personnes qui m'ont permis de mener à bien mon travail par leurs sincères collaborations.*

## Liste des abréviations

|  |  |
| --- | --- |
| **PFE** | Projet de fin d’études |
| **PROD** | production |
| **ERP** | Enterprise Resource Planning |
| **AIP** | Accelerated Implementation Methodology |
| **PME** | Petites et Moyennes Entreprises |
| **SI** | Système d’Information |
| **OT** | Ordre de transport |
| **BL** | Bon de livraison |
| **BE** | Bon d’enlèvement |
| **TMS** | Transport Management System |
| **CRM** | Customer Relationship Management |
| **OTP** | One Time Password |
| **PDA** | Personal Digital Assistant |
| **SLA** | Service level Agreement |

## Mots clés

Cartographie ; Stock ; Processus ; Analyse ; Transport ; AIP ; Messagerie ; ERP ; Odoo

## Résumé

Ce rapport s’inscrit dans le cadre du projet de fin d’études à l’Université ENSAM Casablanca

Pour l’obtention du diplôme d’ingénieur d’état en INTELLIGENCE ARTIFICIEL ET GENIE INFORMATIQUE

Durant ces mois au sein du siège de la société SAPRESS Maroc à Casablanca, ma mission consistaient à développer une application Desktop de gestion de paie

## Key words:

Mapping; Stock; Process ; Analysis; Transportation; AIP; Messaging; ERP; Odoo

## Abstract

## Table des matières

[Dédicaces](#_bookmark0) [3](#_bookmark0)

[Remerciements](#_bookmark1) [4](#_bookmark1)

[Mots clés](#_bookmark2) [5](#_bookmark2)

[Résumé](#_bookmark3) [6](#_bookmark3)

[Key words](#_bookmark4) [6](#_bookmark4)

[Abstract](#_bookmark5) [7](#_bookmark5)

[Table des matières](#_bookmark6) [7](#_bookmark6)

[CHAPITRE 1 : CADRE GENERAL](#_bookmark7) DU PROJET [9](#_bookmark7)

1. [STRUCTURE ET ORGANISATION DU GROUPE EDITO](#_bookmark8) [11](#_bookmark8)
   1. [Présentation de l’organisme d’accueil Groupe Edito 11](#_bookmark9)
   2. [Organisme du Groupe 12](#_bookmark10)
   3. [L’entité SAPRESS 17](#_bookmark13)
2. [PRESENTATION](#_bookmark8) DU PROJET [11](#_bookmark8)
   1. [Etude de l’existant 30](#_bookmark25)
   2. [Problématique 44](#_bookmark34)
   3. [Le but à atteindre 46](#_bookmark37)
   4. [Travail demandé 46](#_bookmark37)
3. PLANIFICATION DU PROJET [11](#_bookmark8)

[CHAPITRE 2 : ANALYSE](#_bookmark23) DE L’APPLICATION ET SPECIFICATION DES BESOINS [29](#_bookmark23)

1. SPECIFICATION DES BESOINS [11](#_bookmark8)
2. ANALYSE DES CAS D’UTILISATIONS [52](#_bookmark40)
3. BACKLOG DU PRODUIT [52](#_bookmark40)

[CHAPITRE 3 : CONCEPTION](#_bookmark46) [56](#_bookmark46)

1. [CONCEPTION](#_bookmark47) GENERALE [57](#_bookmark47)
2. [CONCEPTION](#_bookmark53) DETAILLE [58](#_bookmark53)

[CHAPITRE 4: LES RÉALISATIONS DU PROJET](#_bookmark73) [70](#_bookmark73)

1. [ENVIRONNEMENT](#_bookmark74) DU PROJET [71](#_bookmark74)

[Conclusion](#_bookmark101) [83](#_bookmark101)

Bilan 35

[Annexes](#_bookmark103) [84](#_bookmark103)

Liste des figures 35

Liste des tableaux 35

[Bibliographie](#_bookmark104) [85](#_bookmark104)

**Introduction générale**

## 

## CHAPITRE 1 : « CADRE GENERALE DU PROJET »

# Introduction :

Ce chapitre a pour objectif de présenter en détails des généralités sur l’environnement de l’entreprise SAPRESS et ensuite de présenter le cadre général de notre projet.

### STRUCTURE ET ORGANISATION DU GROUPE EDITO

### Présentation de l’organisme d’accueil Groupe Edito

Le groupe EDITO VENTURES est le fruit d’une fusion depuis l’an 2019 de trois entreprises, SAPRESS créé en 1977 (Société Arabo-Africaine de distribution et d’éditions de presse), SOCHEPRESS crée le 1924 (Société Chérifienne de distribution et de presse) et de Warakpress. Chacune d’entre elles effectue des tâches bien précises et contribue ainsi au succès du groupe.

Ce groupe possède un réseau de plus de 5000 points de ventes repartie dans un total de 23 villes du royaume. Ce large réseau leurs permet de soutenir le développement de ses activités dans les services de la logistique pour le marché marocain et d’instaurer la culture et l’éducation avec leurs stratégie de distributions de livres et manuelles scolaires.

Ces trois Filiales de la société mère édito partage une vocation essentielle d’investir dans le secteur de la distribution, de la culture et de l’éducation et ont pour principal objectif de maintenir la disponibilité de la presse partout dans le royaume auprès des lecteurs dans des conditions économiques équilibrées, tout en donnant aux éditeurs l’opportunité de bénéficier de la synergie des deux réseaux de distribution en termes de couverture et de distribution numérique.



*Figure 1 – Logo du groupe Edito*

### Organisme du Groupe

### Le Groupe EDITO est constitué de trois entreprises, SAPRESS, SOCHEPRESS, et WARAKPRESS (warak trading)



*Figure 2 – Les trois entités du groupe EDITO*

###### Entité de distribution des livres

Cree en 1924, SochePress est l'entité s’ocuppe de la distribution des livres et fournitures scolaires ainsi que la presse nationale et internationale. Elle a commencé aussi ses premières éditions des Manuels scolaires. Lorsqu’on parle de SochePress, on fait référence à la partie administre les produits, les fournisseurs et les éditeurs.



*Figure 3 – Logo de l’entreprise Sochepress*

###### Entité de commercialisation

C'est l'entité qui commercialise les fournitures scolaires (BTS), les fournitures de bureau, papiers A4, toners, les cahiers et les manuels scolaire. Elle offre De plus un service de paiement mobile



*Figure 4 – Logo de l’entreprise Warak*

###### Entité logistique

Avec une expérience de plus de 95 ans et une présence Nationale avec 23 Agences sur les 12 régions du Royaume., Sapress se positionne comme étant le leader sur son domaine.

Son activité reste est principalement la distribution de la presse nationale et internationale

Et la logistique en général. 

*Figure 5 – Logo de l’entreprise SAPRESS*

**Couverture logistique :**

* 10 millions de colis / an
* Distribution avant 8h
* 5000 points de livraison / jour
* 50 relais couvrant les 12 régions du Royaume
* 161 villes couvertes + de 300 collaborateurs



*Figure 6 – Représentants et distributeurs locaux de SAPRESS*

### L’entité Sapress

##### Fiche technique de l’entité SAPRESS :

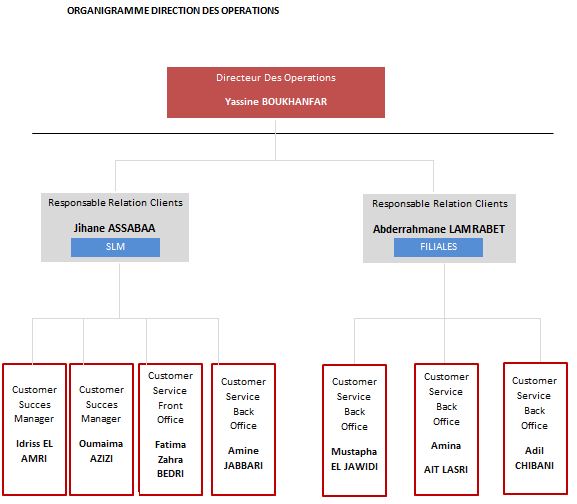
|  |  |
| --- | --- |
| Raison sociale | **SAPRESS Logistique & messagerie** |
| Forme juridique | Société anonyme |
| Nom du dirigent | M. Amine BENCHEKRI |
| Effectif | Entre 200 et 500 |
| Date de création | 1977 |
| Activité | Distribution de presse & messagerie |
| Siège social | Zone industrielle Sidi Maarouf, rue 80 n° 20 20280 Casablanca |
| Capital | 30,000,000 DH |
| Chiffre d’affaires | De 100,000,000 à 500,000,000 DH |
| Site web | [www.sapress.ma](http://www.sapress.ma/) |

*Tableau 2 – Fiche technique de l’entité SAPRESS*

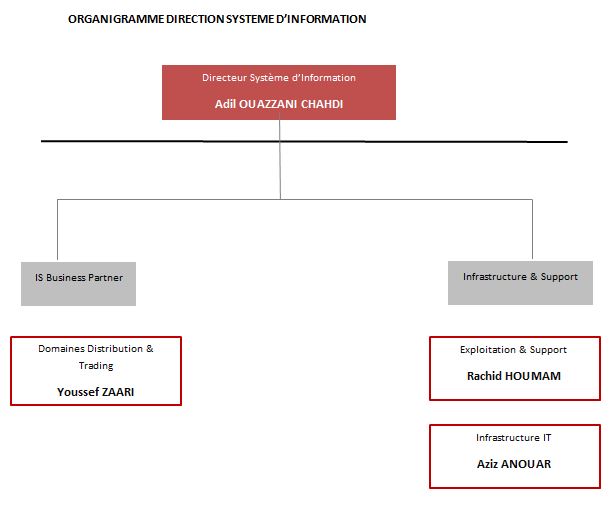
##### L’organigrammes de SAPRESS

Vous trouvez ci-dessous Deux organigrammes simplifiés Ayant pour but de montrer les liens de hiérarchies entre les personnes qui sont intervenues dans le projet.

Ceci dit, Ce projet c’est passer principalement dans la direction des opérations.



*Figure 12 – L’organigramme de la direction des opérations SAPRE*



*Figure 12 – L’organigramme de la direction système d’information SAPRESS*

### PRESENTATION GENERALE DU PROJET

### Définitions

##### RH :

Le processus Ressources Humaines est l’un des piliers de la performance des entreprises. Les rôles de recrutement, de l'orientation, et de la formation sur mesure aux personnels et chauffeurs permettent à l’entreprise de réaliser l’opération messagerie avec efficience en optimisant les couts et délais. Pour cette raison le capital Humain est l’un des points qui peut différencier une entreprise, la démarquer et la rendre compétitif envers ses concurrents.

### Etude de l’existant

Les agents du service RH entretient un journal de paie pour chaque employé. Toutes les informations sur les employés sont regroupées dans des fichiers Excel qui doivent être maintenu et mis à jour. On utilise ensuite des formules mathématiques spécifiques, pour retrouver un tableau de traitement qui doit être généré à la fin de chaque mois. Ce tableau de traitement est ensuite utilisé pour faire des prédictions en calculant l’évolution de la masse salariale dans les années futurs. Ce qui représente une information importante pour la trésorerie.

Tout ce travail est exclusivement fait dans des fichiers Excel ce qui rend la tâche longue et fastidieuse.

### Problématique

D’après ce qui précède, L’équipe Rh (en plus de leurs tache quotidiennes) s’occupe de faire deux tâches très importantes :

• Gérer et maintenir les données de tous les employés en temps réel

• Générer un tableau de Traitement a la fin de chaque mois

La gestion des employés dans des fichiers excès est non seulement pas pratique mais surtout risqué. Et Une ressource RH consacre en moyenne j’jusqu’à une semaine pour générer le tableau de traitement.

### Le but à atteindre

### Le but de notre projet est de créer et développer une application Desktop pour :

### Faciliter et organiser le travail des responsables RH

### Accélérer le processus de gestion de paie.

### Solution proposée

### METHODOLOGIE DE TRAVAIL

##### Pour garantir la satisfaction du client et l’accomplissement de ses exigences avec un cout et délais

##### minimum, nous avons opté pour une méthode agile tout au long du projet.

### Méthodologie Agile

Une méthode agile est une approche itérative et incrémentale, qui, au lieu de prévoir la planification totale du projet avant la phase de développement, On fixe des objectifs à court terme qu’on traite séparément.

On citera comme méthodes de développement Agile :

* L’extrême Programming (XP)
* Scrum
* Feature Driven Development (FDD)
* Lean Software Development
* Agile Unified Process (Agile UP ou AUP)
* Crystal
* et Dynamic Systems Development Method (DSDM)

### Pourquoi Scrum ?

Dans le cadre de notre projet, le processus Scrum s’adapte parfaitement à la décomposition du notre projet de fin

d’étude.

Voici quelques raisons qui justifie ce choix :

* Le besoin du client n’aitant pas très clair au début, une méthode whaterfall classique n’aurait pas servit
* Plus de souplesse et de réactivité,
* Un gain de temps et une meilleure réactivité grâce aux réunions fréquentes avec le client
* S’assurer de la satisfaction du client car il est intégré tous au long du processus de développement
* Facile à utiliser

### Scrum

### La méthode Scrum est une méthode agile de gestion de projets informatiques qui est conforme aux principes des [méthodes agiles](https://www.piloter.org/projet/methode/methode-agile.htm).

### Son but est d’arriver rapidement à une première itération de livrable (produit ou service) utilisable er fonctionnel En le validant au fur et à mesure par le client a la fin de chaque sprint.

### Les sprints constitut un mode itératif et incrémental pour pouvoir proposer, à chaque étape, un livrable et de s’adapter graduellement jusqu’à arriver au besoin réel et final du client

Donc Scrum est une méthodologie simple pour assurer une bonne collaboration de l’équipe sur des projets complexes.

Mais ce projet étant un projet individuel, Peut-on appliquer cette méthodologie ?

### Scrum individuel

Une autre raison de choisir Scrum, serait qu’il peut-être appliquer pour des projets individuels sans équipe.

Une Equipe Scrum est normalement constituer d’un **Poduct Owner**, **Scrum Master** et d’une **équipe de**

**développement**. Pour le bon Déroulement Du projet, Tout ces rôles doivent être assurer par une seule

personne,

Voici comment ceci est manifesté :

* Product owner : en tant que product owner, Le produit final doit être bien comprit et ses exigences et caractéristiques bien définit. Il est aussi primordial de garder une vue globale sur le produit sans se perdre dans le développement.
* Scrum master : Le rôle de Scrum master consiste en identifiant les problèmes puis à chercher à les résoudre. Il faut donc pouvoir identifier les obstacles qui se présentes et s’assurer que le travail finisse durant le time box prédéfinie.
* Equipe de développement : Faire le développement demandé tout en restant motivé et organisé

En ce qui concerne les évents de Scrum, voici comment ils ont été intégrés :

### Sprint Planning : Avant chaque sprint on choisit quel Product Backlog Items va être inclut dans le sprint

### Daily Scrum : Pas appliqué

### Sprint Review : A la fin de chaque Sprint, une réunion avec le client est organisée pour montrer le livrable et voir s’il y’a des changements à faire dans le prochain sprint

### Sprint Rétrospective : Pas appliqué

### PLANIFICATION DU PROJET

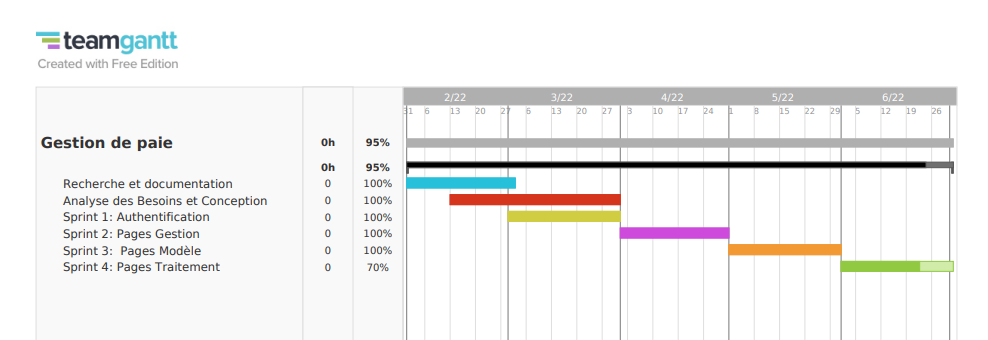
### Diagramme de Gantt

Pour s’assurer de la bonne réalisation du travail demandé dans les délais établis par la convention de stage, il est primordial de définir toutes les étapes essentielles et d’estimer le temps à consacrer pour chacune.

### Pour faire cela, nous allons utiliser Le diagramme de Gantt qui est l’un des outils les plus efficaces pour

### représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet.

En résumé, un diagramme de Gantt serre à regrouper toutes les tâches à accomplir pour que le projet finit dans le temps imparti et indique la date à laquelle ces tâches doivent être effectuées (le planning).



*Figure 12 – Diagramme de Gantt du planning du projet*

### Résumé des taches accomplis

### Cette section a pour objectif de détailler les opérations effectuées durant les tâches du planning réel

### Énoncées dans la section précédente.

* **Recherche et documentation**

##### Cette période a été en premier lieu consacré aux choix des technologies qui seront utilisés tout aux longs

##### du projet (Angular, Typescript, Django). Ensuite Pour les apprendre en s’appuyant sur la

##### documentation et des cours.

##### Le but n’étant pas de les maitriser mais plutôt d’avoir un niveau acceptable qui nous permettra d’entamer

##### le développement ou la plupart de l’apprentissage sera fait.

* **Analyse des besoins et Conception**

##### Durant cette période, il est question de comprendre le besoin en programmant des réunions avec l’équipe

##### RH tout en faisant des recherches sur le sujet (pour mieux comprendre leurs jargon). En plus de concevoir

##### une première version de la base de données.

* **Sprint 1**

##### Le premier Sprint consiste à s’occuper de l’authentification en créant une page login qui communique

##### avec un backend api.

##### 

* **Sprint 2**

##### Ce Sprint est ou les tables essentielles ont été créer dans la base de données. Puis la Page Gestion ou on

##### peut voir tout les enregistrements des tables a été créer avec des filtres.

* **Sprint 3**

##### Le troisième Sprint est celui ou les pages Modèle ou on peut Ajouter, Supprimer et Modifier des

##### enregistrements dans les tables ont été développés.

* **Sprint 4**

##### Durant ce sprint, on a développé la page Traitement ou on calcule les formules

##### La rédaction du rapport est repartie tout au long des sprints

# Conclusion :

Ce Chapitre a mis l’accent sut le cadre général du projet. Nous avons commencé par une description générale du groupe « EDITO » et plus particulièrement de la société « Sapress ». Puis une présentation générale du projet qui sera traité avec une brève présentation, le but à atteindre et le travail demandé en finissant avec l’environnement de développement et la planification temporelle.

## CHAPITRE 2 : « ANALYSE DE L’APPLICATION ET

## SPECIFICATION DES BESOINS »

# Introduction :

# Ce chapitre mettra le point sur toutes les fonctionnalités importantes de cette application et les

# besoins fonctionnels et non fonctionnels qui les as alimenter.

### SPECIFICATIONS DES BESOINS

### Les besoins fonctionnels

### L’application permet de :

### S’authentifier

### Gérer les Employés : Ajouter, modifier, supprimer, rechercher, afficher et imprimer la liste.

### Gérer les Fonctions : Ajouter, modifier, supprimer, rechercher, afficher et imprimer la liste.

### Gérer les centres Couts : Ajouter, modifier, supprimer, rechercher, afficher et imprimer la liste.

### Gérer les Directions : Ajouter, modifier, supprimer, rechercher, afficher et imprimer la liste.

### Gérer les Villes : Ajouter, modifier, supprimer, rechercher, afficher et imprimer la liste.

### Gérer les Contrats : Ajouter, modifier, supprimer, rechercher, afficher et imprimer la liste.

### Gérer les Affectations : Ajouter, modifier, supprimer, rechercher, afficher et imprimer la liste.

### Gérer les Formules : Ajouter, modifier, supprimer, rechercher, afficher et imprimer la liste.

### Gérer les Variables : Ajouter, modifier, supprimer, rechercher, afficher et imprimer la liste.

### Gérer les Rubriques : Ajouter, modifier, supprimer, rechercher, afficher et imprimer la liste.

### Faire des traitements

Le but étant de Créer un tableau de traitement ou pour chaque employé on calcule les formules basées sur les variables et les informations sur l’employé.

### Les besoins non fonctionnels

### L’application doit être :

### Convivial

### Simple, facile à utiliser et intuitif

### Rapide

### Sécurisé

### Performance : Un minimum de délai d’attente.

### Analyse des cas d’utilisations

### Présentation des acteurs

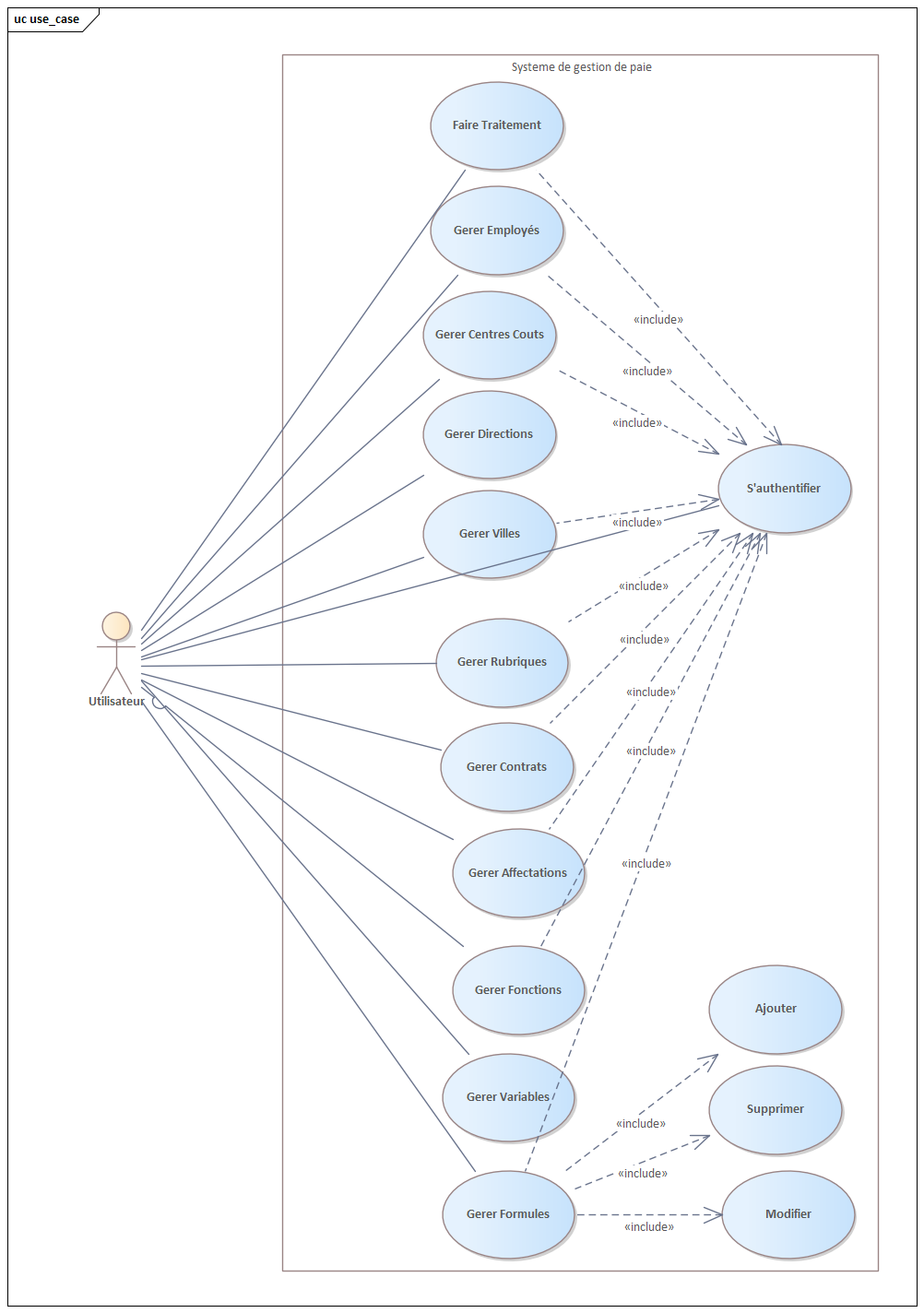
### Un acteur est la personne ou le matériel qui interagit avec notre système afin de réaliser une valeur ajoutée.

### Notre Plateforme ne fait intervenir qu’une seule personne de l’équipe Rh.

### Donc nous disposons d’un seul Acteur qu’on appellera utilisateur.

### Diagramme de cas d’utilisation

### La figure ci-dessous représente le diagramme de cas d’utilisation :



*Figure 12 – Diagramme de Cas d’utilisation*

Pour résumer ce diagramme de cas d’utilisation. Notre utilisateur doit pouvoir gérer toutes les

tables de la base de données et faire des traitements.

### Le Backlog du produit

### Le product backlog est défini par le guide scrum comme étant “une liste ordonnée et émergente de ce qui est nécessaire pour améliorer le produit”. Xavier Koma dit que c’est une liste ordonnée de toutes les fonctionnalités/items, peu importe la forme ou le format de l’item, qui peut être de la user story, des spécifications, un brief, une maquette avec un powerpoint ou encore un gherkin, tant que l’équipe se met d’accord sur le format, de façon à conserver la même communication et la même compréhension.

### En tant que Product Owner, cette responsabilité doit être assurer

|  |  |  |
| --- | --- | --- |
| User Story | Points du Story | Priorité |
| Je peux me connecter en insérant mon email et mot de passe | 2 | 1 |
| Je peux voir la liste des enregistrements de toutes les tables et les filtrer | 3 | 2 |
| Je peux Consulter un enregistrement en particulier. Le supprimer, Modifier et Ajouter. | 3 | 3 |
| Je peux faire des traitements | 2 | 4 |

*Figure 12 Tableau du Product Backlog*

## CHAPITRE 2 : « Conception »

# Introduction :

# La conception d’un système informatique est une étape très importante qui va

# Influencer la qualité et la fiabilité de toute application.

# Ce chapitre détaillera les différents éléments de la conception (générale et détaillée).

### Conception Générale

# Le but de cette partie est d’aborder l’architecture technique utilisé et de montrer les choix des technologies

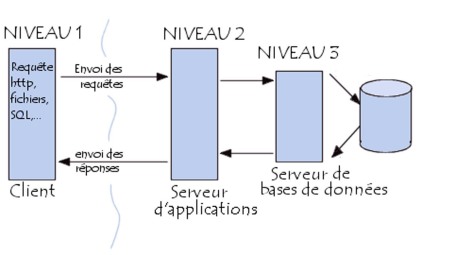
# les plus adaptés aux besoins et aux contraintes de l’organisation d’accueil.

### Architecture trois tiers

##### Définition

L'architecture à trois tier ou architecture à trois niveaux est une architecture d'application logicielle bien établie qui divise les applications en trois couches logiciel logiques et physiques

* **Niveau Présentation** : correspondant à l'[affichage](https://www.techno-science.net/definition/4195.html), Son objectif principal est d'afficher des informations et de collecter des informations auprès de l'utilisateur.
* **Niveau Application** : Ce niveau est le cœur de l'application ou toutes les informations collectées dans le niveau de présentation sont traitées
* **Niveau Données** : aussi appelé la couche base de données, elle correspond à l'endroit où les informations traitées par l'application sont stockées et gérées.

*Figure 12 – Architecture trois tiers*

Dans notre cas La couche présentation correspond à la partie frontend de notre application, la couche Application est notre api et la couche données est notre base de données

En d’autres termes, Votre application (dans le serveur Applications) ne communique pas directement avec la base de données mais communique plutôt via le Protocol http avec une Rest api hostée dans un serveur web.

##### Justification du choix

La communication directe entre l’application et la base de données est appelé architecture 2 tier ou client-serveur et elle présente des plusieurs inconvenants :

* Un coût élevé :  Le coût impliqué dans la configuration et la maintenance du serveur est généralement élevé dans le réseau client-serveur
* Un maillon faible (single point of Failure) : le serveur est le seul maillon faible du réseau client/serveur, Si par malheur il tombe en panne, le client ne peut plus accéder aux données car tout le réseau est architecturé autour de lui.
* Congestion du trafic : si beaucoup de clients lancent des requêtes à au même serveur, ceci peut entrainer un problème d'accès aux informations. Ainsi que des plantages ou un ralentissement de la connexion.

### Conception détaillé

### Dans cette partie nous présentons le diagramme de classes ainsi de séquences constituant le

### Système et les associations entre elles afin de mieux structurer les différentes classes prise en

### compte dans notre application.

### Langage UML

##### Présentation du langage UML

##### UML (en anglais Unified Modeling Language, « langage de modélisation unifié ») est un

##### langage graphique de modélisation objet unifié est une démarche orientée objet .il a été créé pour forger un langage de modélisation visuel normalisée, sémantiquement et syntaxiquement riche pour l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes.

##### Raisons de la modélisation informatique

La modélisation d’un système d’information permet d’obtenir des représentations consolidées sous différents domaines : processus métiers, capacités fonctionnelles, couches applicatives et échanges de données, infrastructures et technologies… Chaque domaine étant interconnecté avec les autres.

Un modèle est donc comparable aux cartes géographiques, ils peuvent être déclinés à différentes échelles qui permettront de zoomer sur une partie du domaine traité pour en connaître les détails les plus subtils. Ce qui nous permet avant tout d’avoir une vue macro sur notre système d’information.

##### Avantages de l’UML

##### Le langage Uml présente plusieurs avantages. Parmi eux, on cite :

* Anticipation de problèmes qui peuvent se manifester avant même qu’ils n’arrivent
* Langage formel et normalisé
* Support de communication performant
* L’analyse devient plus facile
* Facilite la compréhension de représentations abstraites complexes

Un diagramme UML est une représentation graphique, et à chaque vue correspondent des

diagrammes qui sont répartis selon leurs aspects statiques ou dynamiques :

Modéliser les aspects statiques

1. [Diagramme de cas d'utilisation (déjà présenté)](http://www.iro.umontreal.ca/~dift6825/diagrammeUtilisation.htm)
2. [Diagramme d'objets](http://www.iro.umontreal.ca/~dift6825/diagrammeObjets.htm)
3. [Diagramme de classes](http://www.iro.umontreal.ca/~dift6825/diagrammeClasses.htm)
4. [Diagramme de composants](http://www.iro.umontreal.ca/~dift6825/diagrammeComposants.htm)
5. [Diagramme de déploiement](http://www.iro.umontreal.ca/~dift6825/diagrammeDeploiement.htm)

Modéliser les aspects dynamiques

1. [Diagramme de collaboration](http://www.iro.umontreal.ca/~dift6825/diagrammeCollaboration.htm)
2. [Diagramme de séquences](http://www.iro.umontreal.ca/~dift6825/diagrammeSequences.htm)
3. [Diagramme d'états-transitions](http://www.iro.umontreal.ca/~dift6825/diagrammeEtatsTransitions.htm)
4. [Diagramme d'activités](http://www.iro.umontreal.ca/~dift6825/diagrammeActivites.htm)

### Diagrammes de classes

##### Définition

Le diagramme de classe représente les

entités utilisées dans l’application, et plus précisément leurs attributs ainsi que les méthodes

qu’ils utilisent pour traiter les données.

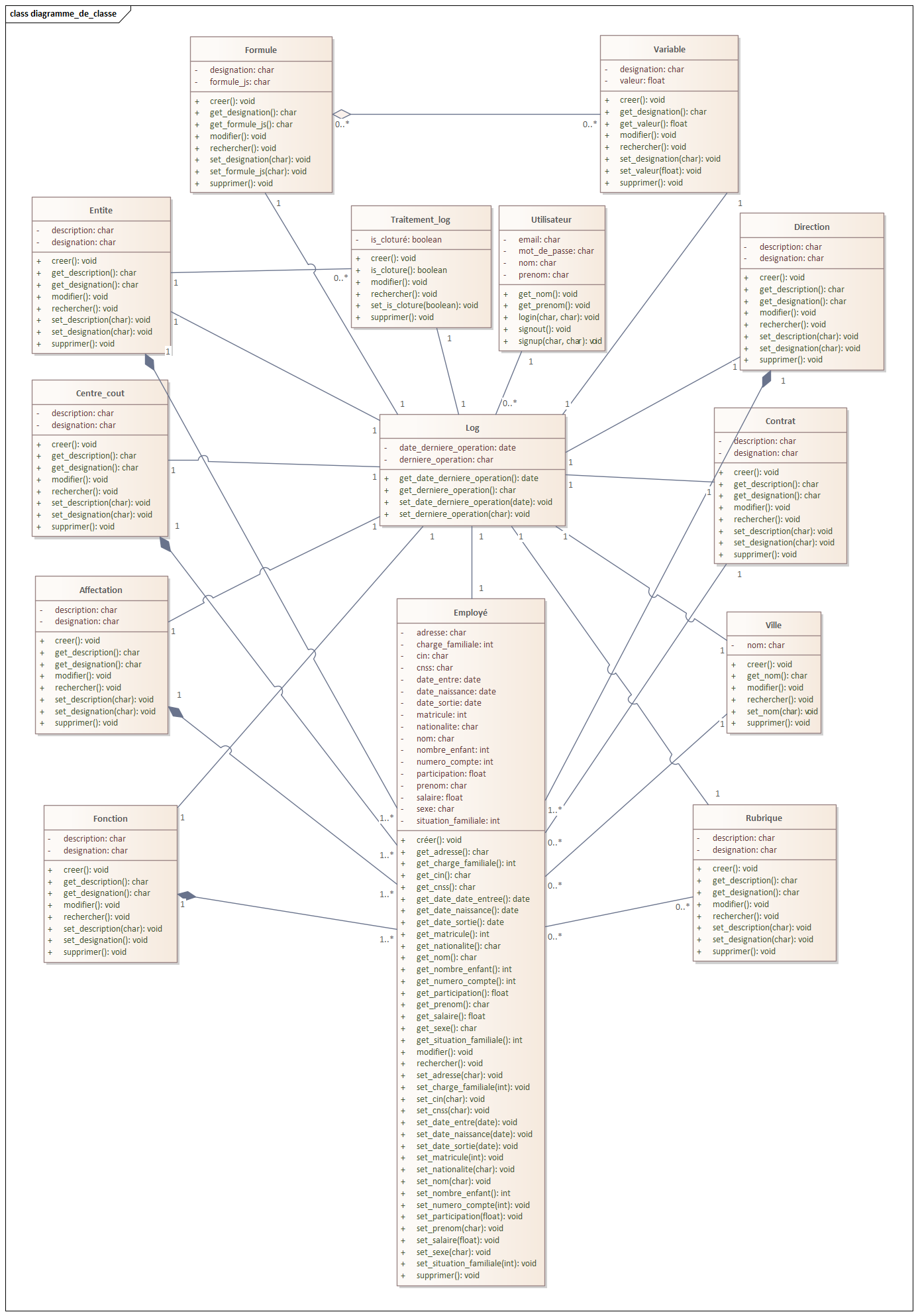
Ils expriment aussi de manière générale la structure statique d’un système, en termes de classes et de relations entre elles.

##### Diagramme

Nous disposons dans notre application de deux bases de données, une principales ou toutes les tables importantes de notre application Et une autre des utilisateurs.

Le but de ce séparage est de rester flexible et garder une certaine marge de scalabilité si le nombre d’utilisateur augmente drastiquement.

Dans la figure suivante nous présentons les classes associées à notre projet :



##### Description des classes

##### Ici on fera une description de chaque classe

##### Entité :

##### On veut dire par entité entreprise. Donc comme déjà expliqué, le groupe Edito est un regroupement de Troie Entités : Sapress, SochePress et Warak trading.

##### Employé

##### L’employé est bien sur tout salarié de l’entreprise.

##### Affectation :

On appelle affectation l’agence ou l’employé travaille actuellement. Par exemple :

##### Agence Marrakech, Agence Tétouan etc.

##### Centre Cout :

##### Au sein d’une même Affectation, Le fonctionnement de l’entreprise s’articule autour différents centres Couts, ou processus. Parmi eux, on peut citer :

##### Logistique, Support, commercial

##### Fonction :

##### C’est une classe qui contient les fonctions principales que les employés effectuent au sein de l’entreprise

##### Exemple : Responsable de la caisse

##### 

##### Direction :

##### Un employé, en plus d’appartenir à un centre cout, appartient réellement à une direction spécifique comme :

##### La Direction logistique et développement ou encore la direction réseau et proximité

##### Contrat :

##### Ceci correspond aux différents types de contrat de travail qui existent et varie selon la durée

##### Les plus fréquents sont le [contrat de travail à durée indéterminée](https://travail-emploi.gouv.fr/droit-du-travail/les-contrats-de-travail/article/le-contrat-de-travail-a-duree-indeterminee-cdi)(CDI) et le [contrat à durée déterminée](https://travail-emploi.gouv.fr/droit-du-travail/les-contrats-de-travail/article/le-contrat-a-duree-determinee-cdd) (CDD)

##### Ville :

##### Dans cette table, on stocke les villes ou réside les salariés

##### Rubrique :

##### Les rubriques incluent toutes les primes, indemnité et avantages que l’employé possède. Exemple : Prime de Transport, Avantage nature carburant

##### Log :

##### Une table ou on garde une trace de l’utilisateur qui a dernièrement modifier l’enregistrement, ainsi que la nature et la date de cette modification

##### Traitement log :

##### Cette classe contient tous les traitements qui ont été fait et s’ils ont été clôturé ou pas

##### Formule :

##### Les formules qu’on utilisera pour créer notre tableau. En guise d’exemple voici une des formules les plus simples :

##### Ancienneté = (date d’aujourd’hui – date d’entrée) / 365

##### Variable :

##### Les variables utilisées par toutes ces formules et qui peuvent être modifier par notre utilisateur.

##### Dans la formule de l’ancienneté, on peut considérer 365 comme une variable.

##### Utilisateur :

##### Cette classe contient les informations de tous les utilisateurs qui peuvent accéder à notre application

##### Règles Et Commentaires

* Plusieurs Employés peuvent appartenir à la même entité, centre cout, affectation, direction et ville
* Plusieurs Employés peuvent avoir la même fonction et le même contrat de travail
* Un Employé est doit avoir une combinaison matricule entité unique. Car la matricule est l’identifiant de l’employé dans l’entreprise et elle est unique dans l’entité.
* Une formule peut avoir plusieurs ou aucune variable
* Pour chaque Entité, on peut faire un traitement des employés qui y existent
* Chaque Employé peut avoir une ou plusieurs rubriques (primes ou indemnités). Plusieurs Employés peuvent avoir la même prime par exemple mais avec un montant qui peut changer.

Dans la base de données, ceci se traduit en ajoutant une table Employé\_rubrique qui enregistre les montants des rubriques.

|  |
| --- |
| Employé\_rubrique |
| FK : employé\_id FK : rubrique\_id  montant |

Un exemple d’enregistrement dans cette table serait qu’un employé X a une Indemnité de transport de 200dh

* Chaque objet de toutes les classes contient un objet de la classe Log pour garder son historique

Si on traduit ça en tables dans la base de données, chaque table doit avoir une clé étrangère vers un enregistrement dans la table Log.

Exemple pour la base Affectation :

|  |
| --- |
| Affectation |
| PK : id  designation  description  FK : log\_id |

### Diagrammes de séquences

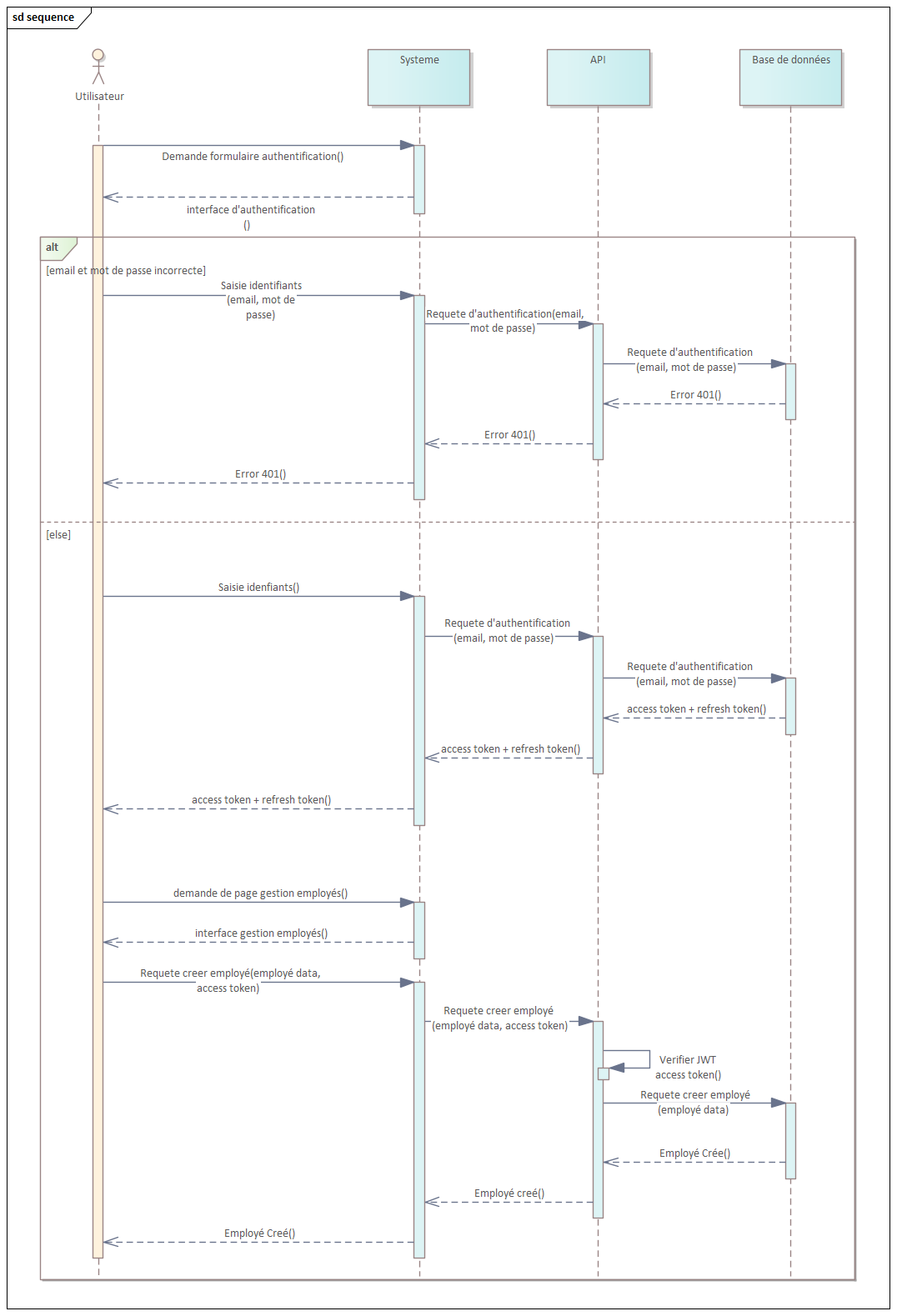
##### Définition

Un diagramme de séquence est un diagramme UML qui représente des [interactions](https://fr.wikipedia.org/wiki/Unified_Modeling_Language) entre les [acteurs](https://fr.wikipedia.org/wiki/Acteur_(UML)) et le système. Il décrit les scénarios d’un cas d'utilisation en mettant

L’accent sur la chronologie des opérations en interaction avec les objets.

##### Diagramme

Dans la figure suivante nous présentons le diagramme de classes de l’authentification :



Ce diagramme cela expliqué dans la partie de l’authentification.

# Conclusion :

Avant de commencer la partie de réalisation, il est primordial de conceptualiser notre système. Cette conception ensuite se perfectionne et change au fur et à mesure qu’on avance dans les sprints.

## CHAPITRE 2 : « Réalisation »

# Introduction :

Nous arrivons maintenant à la phase ultime. Après avoir détaillé la conception de notre application, on passe maintenant à l’tape de la réalisation. On commencera par une présentation de l’environnement matériel et logiciel de développement.

### Environnement de travail

### Environnement Matériel

##### Pour la réalisation de ce projet nous avons utilisé le matériel suivant :

|  |  |
| --- | --- |
| **PC portable** | Lenovo |
| **RAM** | 8 Go |
| **Microprocesseur** | Intel(R) Core (TM) i5-9300H CPU @ 2.40GHz 2.40 GHz |
| **Carte graphique** | NVIDEA GeForce GTX 1050 |
| **Disque dur** | 1,81To |
| **Système d’exploitation** | Windows 10 |

### Environnement Logiciel

### Visual Studio Code est désormais utilisable directement dans un navigateurVisual code studio : Visual Studio Code (connu sous le nom de VS Code) est un éditeur de code gratuit et open-source développé par Microsoft supportant un très grand nombre de langages grâce à des extensions. Il est disponible pour Windows, Linux et MacOs et supporte l’autocomplétions, la coloration syntaxique, le débogage, et les commandes git. Ainsi que beaucoup d’autre fonctionnalités puissantes qui ont fait de VS Code l'un des outils d'environnement de développement les plus populaires ces derniers temps.

### Postman : Postman est un excellent outil lorsque vous essayez de disséquer des API RESTful créées par d'autres ou de tester celles que vous avez créées vous-même. C’est une application utilisée pour les tests d'API. Il s'agit d'un client HTTP qui teste les requêtes HTTP, en utilisant une interface utilisateur graphique, Postman API Platform | Sign Up for Freeà travers laquelle nous obtenons différents types de réponses qui doivent ensuite être validées.

### Logo github - Icônes des médias sociaux gratuitesGitHub : GitHub est un service d'hébergement Web [open source](https://www.lemagit.fr/definition/Open-Source) pour les projets de développement de logiciels qui utilisent le système de contrôle de version Git. Qui a été créé par Linus Torvalds dans le but d'accélérer le développement logiciel. Livrée en tant que logiciel à la demande ([SaaS](https://www.lemagit.fr/definition/SaaS), Software as a Service), Le site assure également un contrôle d'accès et des fonctionnalités destinées à la collaboration comme le suivi des bugs, les demandes de fonctionnalités, la gestion de tâches et un wiki pour chaque projet.

### Langages de Programmation

### Typescript : C’est un langage de programmation libre et open source développée par Microsoft en 2012. Pour être plus précis, c’est un sur ensemble de Javascript qui a pour but d'améliorer et de sécuriser la production de code JavaScript. Son ambition principale est d’améliorer la productivité de développement d’applications complexes. Le langage introduit des fonctionnalités optionnelles comme le typage ou la programmation orientée objet. Et ceci, sans l’utilisation d’aucune librairie. Il suffit d’utiliser l’outil de compilation de TypeScript pour le transcompiler en Javascript.

### 

### Python : Le [langage Python](https://datascientest.com/python-tout-savoir) est un langage de programmation [open source](https://www.futura-sciences.com/tech/definitions/informatique-open-source-18154/) multi-plateformes et orienté objet puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Parce que sa syntaxe est élégante, que son typage est dynamique et qu'il est interprété, Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes.

### 

### Html 5 : HTML est le [langage de balisage](https://fr.wikipedia.org/wiki/Langage_de_balisage) conçu pour représenter les [pages web](https://fr.wikipedia.org/wiki/Page_web). Htlml 5 n’est rien d’autre qu’une révision de ce dernier développé pour résoudre les problèmes de compatibilité qui se posait

### SASS :Sass est un [langage de script](https://fr.wikipedia.org/wiki/Langage_de_script) [préprocesseur](https://fr.wikipedia.org/wiki/Pr%C3%A9processeur) qui est [compilé](https://fr.wikipedia.org/w/index.php?title=Langage_compil%C3%A9&action=edit&redlink=1) ou [interprété](https://fr.wikipedia.org/wiki/Interpr%C3%A8te_(informatique)) en [CSS](https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade) (Cascading Style Sheets). . Il vous permet d'utiliser des variables, des règles imbriquées, des mixins, des fonctions, etc., le tout avec une syntaxe entièrement compatible CSS. Sass propose deux syntaxes et donc deux extensions, .SASS et .SCSS.Dans Ce projet on utilisera SCSS.

### 

### Frameworks utilisés

### Angular 13 :Angular est une plate-forme et un Framework écrit en TypeScript permettant de créer  des applications Web et plus particulièrement de ce qu’on appelle des « Single Page Applications » à l'aide de HTML et TypeScript . Il implémente les fonctionnalités de base et facultatives sous la forme d'un ensemble de bibliothèques TypeScript que vous importez dans vos applications

### Django : Django est un Framework Python de haut niveau, gratuit et open source, permettant un développement rapide de sites internet, sécurisés, et maintenables. Il est donc clairement orienté pour les développeurs ayant comme besoin de produire un projet solide rapidement et sans surprise. C’est pourquoi, le projet a pour [slogan](https://fr.wikipedia.org/wiki/Slogan) « Le Framework pour les perfectionnistes avec des deadlines ». En Effet, Django Te permet de ne plus réinventer la roue mais plutôt de vous concentrer sur l'écriture de votre application alors qu’il prend en charge la plupart des tracas du développement web. Le tout couronné par une bonne documentation et une communauté active.

### Electron : Electron.js est un Framework qui permet à l'utilisateur de créer des applications desktop avec les technologies web comme HTML5, CSS et JavaScript. Electron s'occupe de la partie technique pour que vous puissiez vous concentrer sur le cœur de votre application. C'est un projet open source lancé par Cheng Zhao, ingénieur chez GitHub.

### MySQL : MySQL est un système de gestion de base de données relationnelle SQL open source développé et pris en charge par Oracle.Il améliore la rapidité et la souplesse de l'ensemble en séparant les données dans des table reliées par des relations définies. Le [SQL](https://www.futura-sciences.com/tech/definitions/informatique-sql-2524/) dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données.

### Material UI :Angular Material est une bibliothèque de composants d'interface utilisateur. Il est spécialement conçu pour les développeurs AngularJS. Cela aide à concevoir l'application de manière structurée. Les composants Angular Material aident à créer des pages Web et des applications Web attrayantes, cohérentes et fonctionnelles tout en respectant les principes de conception Web modernes tels que la portabilité du navigateur, l'indépendance des appareils et la dégradation gracieuse.

### Justification des choix

* Electron

Il existe plusieurs technologies pour créer des applications desktops.

On a le choix entre du développement native comme :

-[Cocoa software framework](https://developer.apple.com/)  pour MacOS

-[.NET framework](https://dotnet.microsoft.com/) pour windows

### Mais vue que notre utilisateur peut très bien utiliser différents systèmes d’exploitation,

### on a opté pour une approche cross plateformes. La aussi on trouve plusieurs choix comme :

### -Le Framework java Swing

### -Java fx

### -Electron.js

### Mais le choix d’Electron nous offre plusieurs avantages comme :

### -Facile à apprendre :

### Vu qu’il utilise seulement JavaScript, HTMLet CSS, tout développeurs avec des compétences en développement web peut commencer sans effort. Ceci résout aussi un problème de cout puisqu’en générale, un développeur d’applications desktop coute plus cher.

-Vitesse de développement accrue

Le langage javascript est un avantage qu’on ne peut pas nier car contrairement aux autres langages de programmations bas niveau qu’on utilisait, on n’a pas besoin de tout développer de zéro (from scratch). Typescript rend le développement encore plus sûr et efficace.

### Plusieurs applications desktop très connues utilise Electron comme : Web Torrent desktop app, WordPress desktop app, WhatsApp Desktop app ou encore Discord.

* Angular

Maintenant qu’on a choisie d’utiliser Electron, développer une application seulement avec Javascript, HTML et CSS n’est pas pratique pour un projet important comme le nôtre. Il faut à présent choisir un Framework convenable. Et bien qu’il y’ai plusieurs frameworks, le choix reste entre les deux plus utilisés React.js et Angular.

On s’est arrêter sur Angular car il nous offre une grande communauté active, l’efficacité incontestable de Typescript avec une expérience de développement plus propre et organisé, une séparation claire entre l’interface UI et la partie métier (fichiers html et fichiers ts) et une façon très simple de faire des changements, qui est l’Angular CLI. De plus, Angular est généralement utiliser pour des projets d’entreprise.

Material UI est une librairie de composants qui remplace Bootstrap.

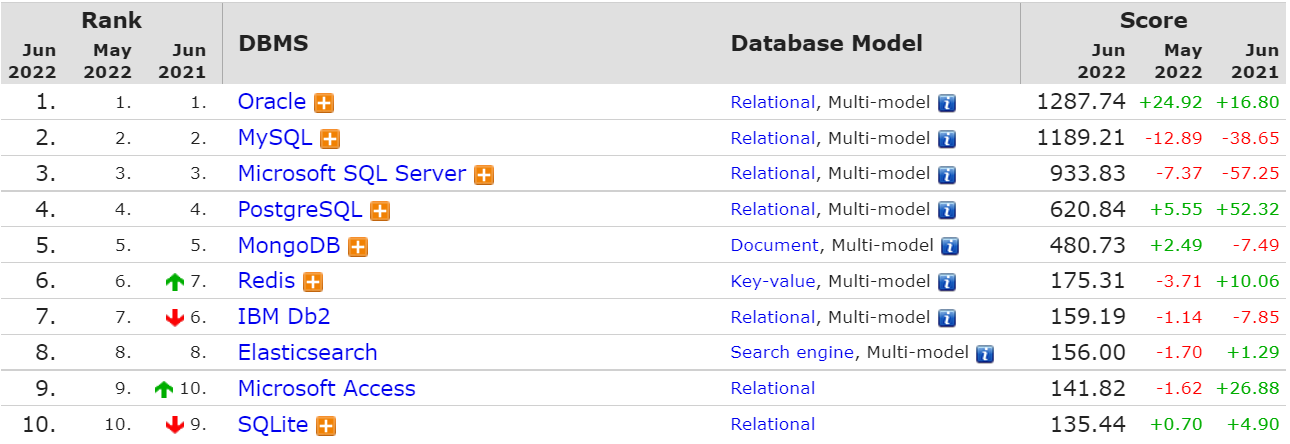
* Django

Devant la complexité de Angular, python se montre comme un bon choix pour la partie backend avec un code facile à lire et écrire. Django est un Framework crée par les développeurs pour les développeurs, ce qui veut dire que tous les problèmes communs sont déjà résolus avec plusieurs fonctionnalités. Le tout avec une sécurité indéniable.

* MySQL

Toujours dans l’optique d’une optimisation de l’outil, pour notre SGBD, il est important de savoir d’abord quel type de base de données est le plus adapté à notre projet entre SQL ou NoSQL. Une fois on choisit SQL car on a besoin d’un modèle relationnel clair, MySQL devient un choix raisonnable pour assurer une rapidité de traitement optimale. Chose qui peut être parfois compromise pour une plus grande scalabilité et intégrité des données comme dans le SGBD PostgreSQL.

De plus, on trouve ci-dessous un classement des 10 SGBD les plus utilisé actualisé chaque mois et on peut clairement voir que MySQL prend la deuxième place.



### Frontend

### Description des interfaces graphiques

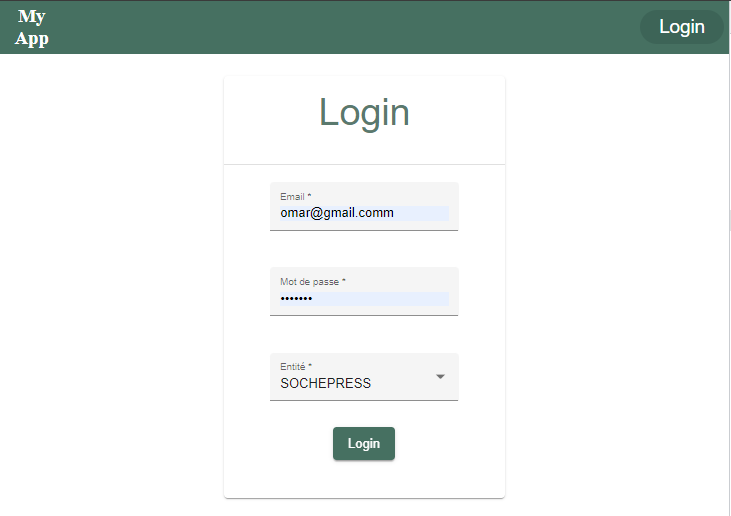
### L’interface graphique est une partie très importante pour la réalisation d’une application

### Convenable offrant un certain plaisir à l’utilisateur lors de sa navigation.

Nous allons consacrer cette partie pour la présentation des principales interfaces de l’application :

##### Interface authentification

L’interface authentification est une interface est primordiale, c’est la phase d’identification pour accéder à l’application.

Elle s’affiche dès la connexion et l’utilisateur est amener à saisir son email, son mot de passe, et l’entité à laquelle il veut accéder (l’entité peut être modifier une fois dans l’application) pour qu’il puisse accéder à l’application.

Une gestion des erreurs est bien sûr effectuée si les identifiants ne sont pas saisis. Si par exemple l’input n’est pas un email ou s’ils ne sont pas valides ou encore si l’email et le mot de passe n’existe pas.

Nous ne disposons pas de page signUp car personne de doit pouvoir créer son propre compte. Pour l’instant quelques comptes ont été crée dans la base de données mais dans des versions future un administrateur aura le droit de le faire dans l’application elle-même.

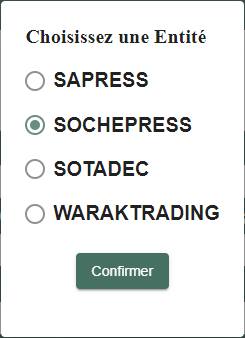
##### Voici donc les deux états du header de notre application

##### -Si l’utilisateur n’est pas connecté

##### 

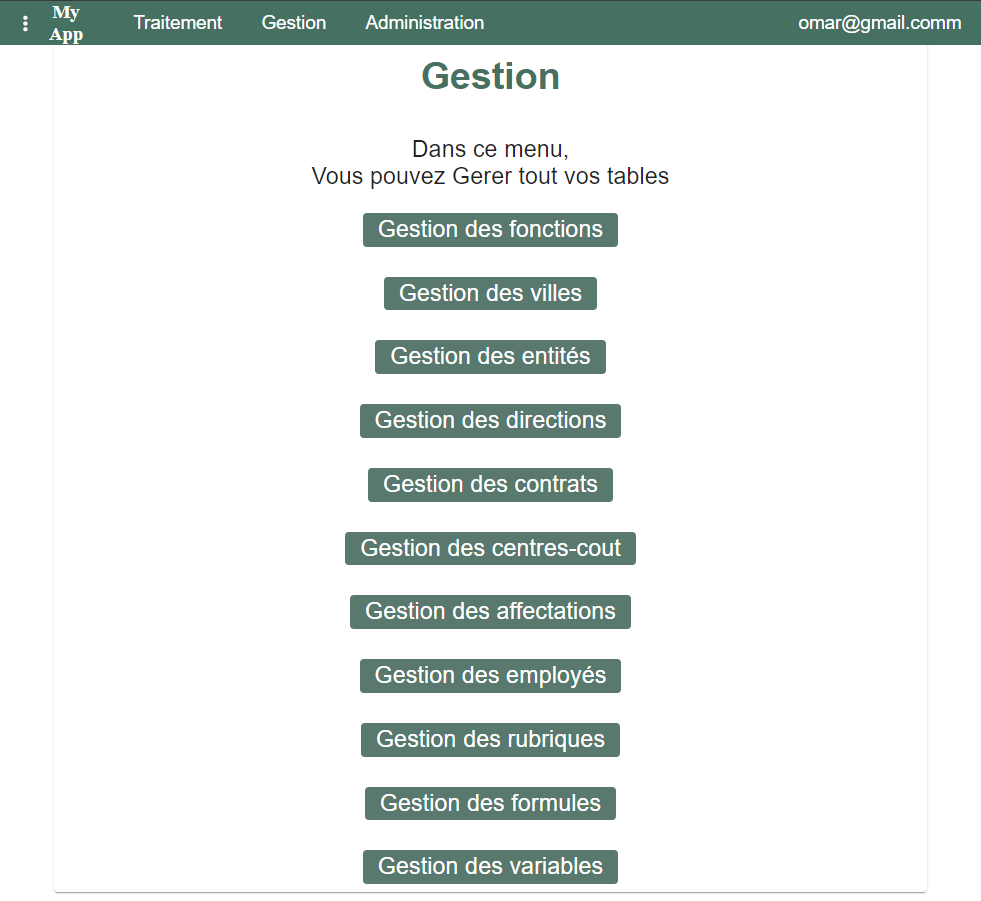
-Si l’utilisateur est connecté

Si vous appuyez sur votre email, vous pouvez soit vous déconnecter soit changer d’entité.

Si vous appuyer sur changer d’entité le dialogue suivant s’affiche :

##### Interface Menu Gestion

Cette page est un menu ou vous pouvez choisir quelle table vous voulez gérer.

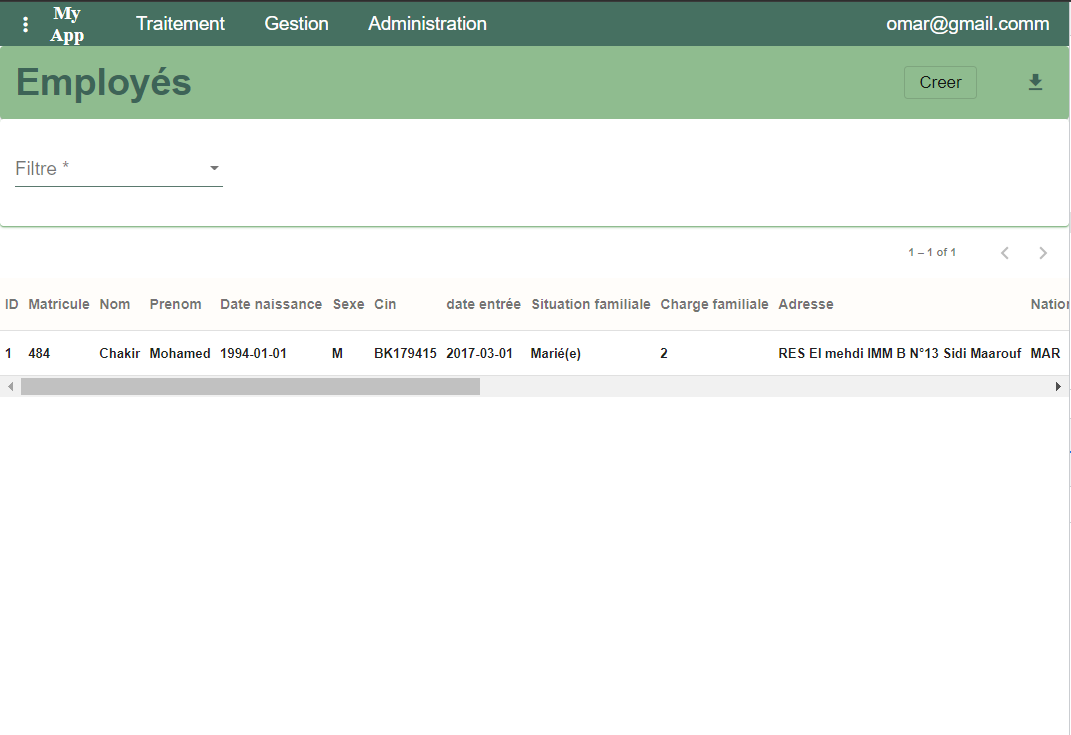


Si vous n’êtes pas dans cette page et vous voulez changer de page de gestion sans passer par

cette interface. Le menu apparait dans le header en appuyant sur

##### Interface Gestion Employés

Une fois que vous appuyez sur l’un des boutons de gestion dans le menu, vous êtes automatiquement redirigée vers la page gestion de la table correspondante. On considère par la suite que vous avez appuyé sur le bouton « Gestion Employés » car les mêmes interfaces apparaitront pour les autres tables. Voici l’interface de gestion employés :



On peut avoir plusieurs pages avec un maximum de 100 employés par page.

Avec un nombre d’employés importants, il est primordial de pouvoir filtrer ces derniers. Et ceux en se basant sur n’importe quel champ de l’employé.

Pour se faire, On choisit le champ visé dans la première liste déroulante, puis le type du filtrage :

Egal, Contient, Différent, Supérieure Strictement, Inferieur, Supérieure, Compris entre,

Puis on saisit la valeur soit manuellement soit dans une liste déroulante en fonction de si on le choix ou pas.

------------------------------------------------image filtre ---------------------------------------------------

##### Interface Liste Employé

##### Une fois on a trouvé l’employé qu’on veut consulter et on appuie dessus, on se retrouve dans la page liste employé ou on peut voir toutes les informations relatives à ce dernier.

------------------------------------------------image employé liste ----------------------------------------

****Les propriétés normales comme Adresse, Nombre enfant etc.… sont regrouper en groupe de deux tandis que ceux avec lesquels la table a une relation « One to Many » comme centre cout, fonction ou encore direction sont affiché individuellement avec leurs informations respectives. Comme suit :

##### En appuyant sur, On est directement redirigé vers le centre cout « Logistique »

##### Pour la table Rubrique avec laquelle employé à une relation « Many to Many », une liste des rubriques est affichée comme suit :

-----------------------------------image employé rubrique ----------------------------------------

##### Interface Modifier Employé

##### Dans l’interface liste employé a droite, on trouve Modifier. On appuie dessus et deux boutons Sauvegarder s’affiche d’abord. Puis tous les champs normaux se transforment en input (date ou texte).

-----------------------------------image Modifier Employé ----------------------------------------

##### Une liste déroulante apparait pour les tables « One to Many » comme suit :

-----------------------------------image centre cout liste déroulante -------------------------------------

##### Pour Rubrique avec lequel employé entretient une relation « Many to Many », on peut par exemple ajouter, supprimer ou modifier une prime de l’employé en précisant le montant en DH.

-----------------------------------image centre cout liste déroulante -------------------------------------

##### Une fois finit, vous pouvez appuyer sur sauvegarder pour garder la modification ou annuler et vous serez rediriger vers l’interface Liste Employé

##### Supprimer Employé

##### Si vous appuyez sur supprimer un dialogue apparait ou vous pouvez confirmer la suppression ou pas

-----------------------------------image popup supprimer -------------------------------------

##### L’interface traitement

L’interface traitement est en cours de développement

Comme vous pouvez voir, la couleur omniprésente dans toutes les interfaces est le vert avec des intensités et des valeurs différentes. Ce choix est fait car la couleur verte est une couleur froide associé a la douceur, la tranquillité, la confiance et la contemplation. Rien qu’en faisait ce choix, l’utilisateur est tout de suite plus amené à faire confiance à notre application. Le blanc nous apporte une couleur neutre et un certain contraste avec le vert.

### Structure des fichiers

Pour des projets importants et consistants, il est essentiel d'avoir une compréhension globale de la manière dont l'application doit être structurée dès le début. Le but est d’arriver à une structure de fichier évolutive et maintenable qui respecte les meilleurs pratique d’Angular et ses lignes directives.

### 

##### Description des Dossiers importants

### Angular présente une structure modulaire ou le code est organisé en différents modules donc tout les services et composantes sont divisés en différents groupes.

### Styles

### Dans ce dossier, on regroupe tous les fichiers de style qui identifie le style global de l’application.

### Et puisqu’on travaille avec Sass, on définit ici aussi les variables globales et les Mixins qui ne sont rien d’autre qu’un extrait de code(style) réutilisable n’importe où.

### Assets

### Généré par Angular CLI, il sert à garder tous les fichiers médias.

### Environment

### Utiliser pour préciser l’Endpoint de l’api et séparer en l’environnement prod et dev

### Notre application réside au sein du module principale app, voici les dossiers qui s’y trouve :

### Module Core

Ce module joue le rôle du module racine de l'application ou on regroupe tout ce qui ne doit être charger qu’une seul fois. En gros, Il contient tout ce qui est statique et qui ne dépend pas de l’utilisateur comme les gardes (seront expliqué plus tard) et les constantes.

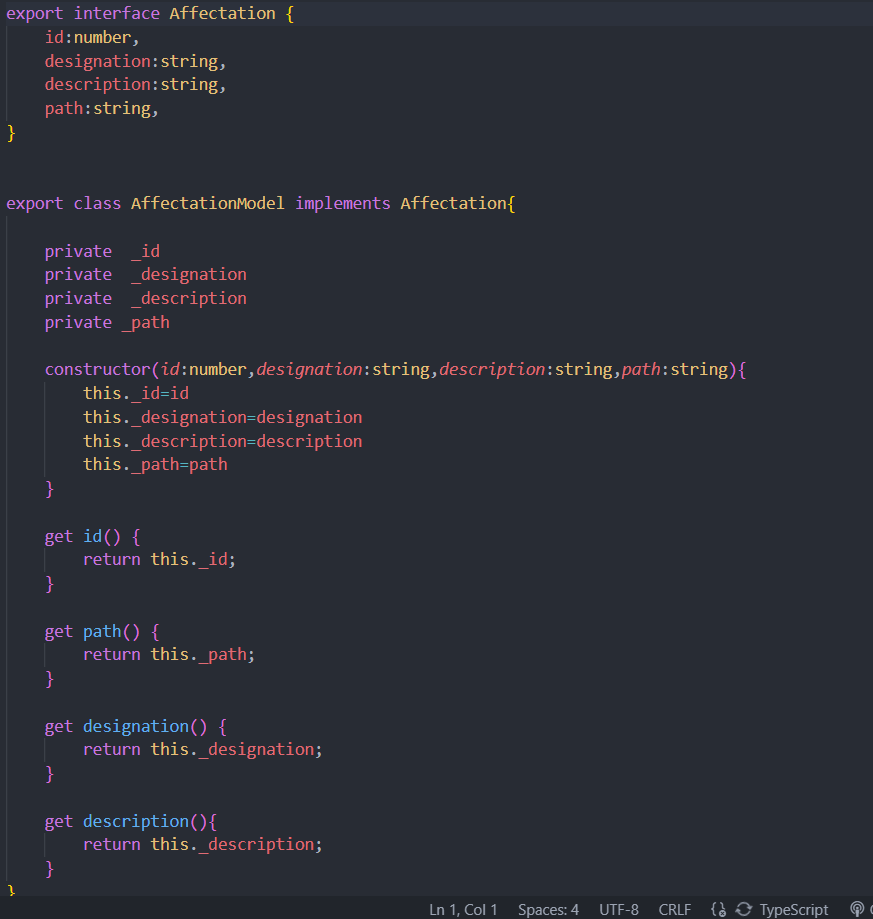
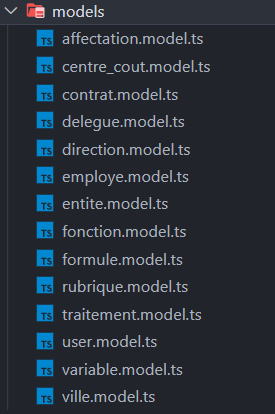
Le module Core contient aussi les fichiers Typescript qui envoie les requêtes a notre API

### Module Shared

Ici on regroupe les composant dynamiques qui seront réutilisé souvent dans toute l’application, comme le Header, le Footer, les pipes (fonctions utilisées dans les fichiers html pour transformer les valeur) et directives (Modifie l'apparence ou le comportement des éléments DOM et des composants angulaires) ect…

On trouve aussi l’un des dossiers les plus importants qui est « Models » ou tous les modèles des classes existent.

Vous trouverez ci-dessous le modèle de la table Affectation :



Vous remarquerez que les attributs sont tous privé. Cette encapsulation nous aide à éviter des erreurs humaines et protège l’objet de tout accès non voulu. Aucun setter n’est créé car aucune modification n’est requise.

### Module Modules

### Modules contient les modules principaux de notre application. Le module gestion ou les pages de gestions existent, le module auth pour la page login et finalement le module traitement.

### Gestion des états

##### Définition

##### Pour une application monopage (SPA) comme la nôtre avec beaucoup de composants, la gestion des états est un élément clé.

### Avant tous, l’état est l’ensemble des données et informations contenu dans le composant, son état actuel. Il est bien sur possible de passer l’état directement du composant parent au fils et vice versa, ceci est la méthode traditionnelle. Mais cette méthode arrive à sa limite si une donnée doit être accessible partout dans l’application par exemple. De plus, de multiples états sont utilisés à travers une application, ce qui explique la complexité inhérente de la gestion de ces états.

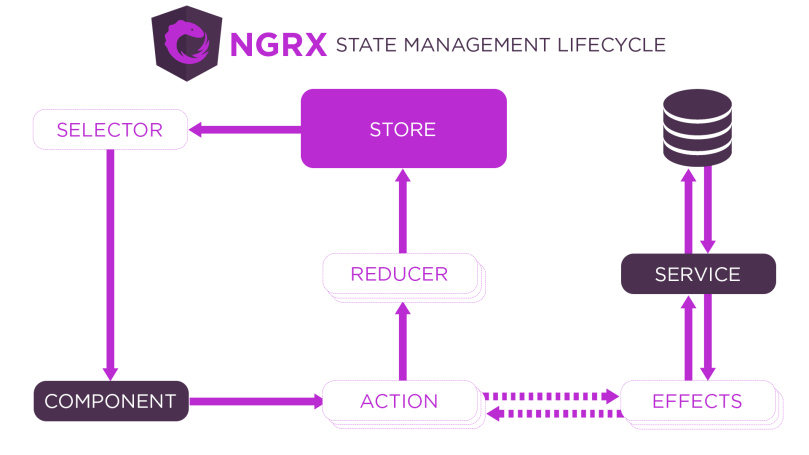
### C’est là ou intervient les Framework de gestion d’états.

##### NGRX

NgRx est un Framework de gestions d’etats pour créer des applications réactives dans Angular.

NgRx est un groupe de bibliothèques d’Angular pour les extensions réactives et la gestion des états. Il facilite le développement angulaire en simplifiant l'état de l'application dans les objets et en appliquant un flux de données unidirectionnel.

##### NGRX cycle de vie

Toute notre application respecte ce cycle de vie :

### Store

Le Store est le composant le plus important de NgRx. Il fournit un magasin unique ou toutes les données de l’application sont centralisés et reflète donc l’état globale de l’application. C’est une structure de données unique et immuable.

* Action

Les actions expriment des événements uniques qui se produisent dans notre application. Ces événements

Peuvent être entre autres des interactions des utilisateurs ou des requêtes réseau. Les actions sont la façon dont l'application communique avec NgRx pour lui dire quoi faire.

### Reducer

Les réducteurs sont responsables des transitions d'état de votre magasin. Ils réagissent aux actions envoyées et exécutent une fonction pure (fonction qui produit la même sortie pour une entrée donnée) pour mettre à jour le magasin.

### Selector

Les sélecteurs sont essentiellement le mécanisme de détection de changement NgRx. Ils permettent à notre application d'écouter les changements d'état.

### Effect

Les effets gèrent les effets secondaires de chaque action. Ces effets secondaires sont généralement des communications avec l’API externe via HTTP. Ils isolent les effets secondaires des composants. Cela nous donne des composants "plus purs" qui sélectionnent l'état et exécutent des actions.

### Réactivité

### Comme déjà mentionné, la librairie de composant utilisé est Angular Material UI. Elle propose une

### fonctionnalité intéressante qui remplace le système Flexbox CNSS qui est Flexbox Layout.

### Toute l’application en se basant sur ça est extrêmement réactive et s’adapte à une multitude de taille

### d’écran selon des breakpoints spécifiques : xs, gt-xs, sm, gt-sm, md, gt-md, lg, gt-lg, xl

### voici un exemple de la page gestion:

### -----------------------------------page gestion changer -------------------------------------

### Le header s’adapte lui aussi au screen size :

### -----------------------------------header changer -------------------------------------

### SCSS

### Guards

### Lazy loading

## Conclusion

## Annexes

1. *Liste des figures*

*Figure 1 – Logo du groupe Edito 12* [*Figure 2 – Les trois entités du groupe EDITO 13*](#_bookmark11) *Figure 3 – Logo de l’entreprise Sochepress 13 Figure 4 – Logo de l’entreprise Warak 13 Figure 5 – Logo de l’entreprise SAPRESS 14 Figure 6 – Représentants et distributeurs locaux de SAPRESS 14 Figure 7 – Organigramme Fonctions SUPPORT & BUSINESS UNITES 15 Figure 8 – Organigramme Directions Relations Client & Transformation Digitale 16 Figure 9 – Organigramme Directions Logistique et Développement 16 Figure 10 – Organigrammes Directions Systèmes D’information 17 Figure 11– Camion de SAPRESS 17* [*Figure 12 – L’organigramme de SAPRESS 18*](#_bookmark15) *Figure 13 – Cartographie des processus de SAPRESS 19 Figure 14 – les étapes de la démarche AIP 23* [*Figure 15 – Exemple du diagramme de Gantt 25*](#_bookmark19)[*Figure 16 – Définition d'un logiciel ERP (ou PGI) 26*](#_bookmark22) *Figure 17 – Différents ERP 28* [*Figure 18 – Processus Pick-up 30*](#_bookmark26)[*Figure 19 – le processus Livraison 37*](#_bookmark30) *Figure 20 – Les tournées de centre-ville 38 Figure 21 – Les tournées de périphérie 38 Figure 22 – Les tournées de campagne 39 Figure 23 – le processus Retour 42* [*Figure 24 – Construction de business Plan avant Covid 44*](#_bookmark35) *Figure 25– Construction de business Plan après Covid 44* [*Figure 26 –Lancement du nouveau business 45*](#_bookmark36)[*Figure 27 – Cartographie de processus messagerie 46*](#_bookmark38)[*Figure 28 – Planification d'une opération de messagerie 49*](#_bookmark39)[*Figure 29 – Comparaison des ERP sur google trends 53*](#_bookmark43)[*Figure 30 – Logo d' Odoo 53*](#_bookmark45) *Figure 31 – Différents modules d'Odoo 55 Figure 32 – la version 7 du logiciel OpenERP 55 Figure 33 – Exemples des clients de SAPRESS 57* [*Figure 34 – le parcours opérationnel du métier messagerie 58*](#_bookmark52) *Figure 35 – Description du processus du retour de fond 64 Figure 36 – Maquette d'une session de tournée 66 Figure 37 – Description du processus retour des clients 66 Figure 38 – Les instances PROD et pré-Prod 71 Figure 39 – Le menu Configuration 71* [*Figure 40 – Les droit d'accès d'utilisateurs et gestion des habilitations 72*](#_bookmark76)[*Figure 41 – L'interface d'accueil 72*](#_bookmark78)[*Figure 42 - message d'erreur lors de la configuration d'une liste de prix 73*](#_bookmark80)[*Figure 43 - Disfonctionnement lors de l'implantation de nouvelles parties développer 74*](#_bookmark81)[*Figure 44 - Exemples d'un message de bug sur le retour du colis 74*](#_bookmark83)[*Figure 45 - l'architecture technique préparée pour le déploiement 75*](#_bookmark86)[*Figure 46 - support de formation Odoo 76*](#_bookmark88)[*Figure 47 - Etat d'avancement de la formation Odoo pour les agences 76*](#_bookmark90) *Figure 48 - Pourcentage de la partie formée 77 Figure 49 - Plateforme de gestion de projet Trello 77* [*Figure 50 - interface pcAnywhere 79*](#_bookmark94)

[*Figure 51 - l'interface help desk Karizma 80*](#_bookmark96)[*Figure 52 - Exemples des tickets créés pour les bugs 80*](#_bookmark97) *Figure 53 - Interface « Sapmosaada » 81* [*Figure 54 - l'interface de l'instance Prod prête à utiliser 82*](#_bookmark100)[*Figure 55 - nombre de colis livrés par mois 83*](#_bookmark102)

1. *Liste des tableaux*

[*Tableau 1 – Type de la clientèle cible 15*](#_bookmark12)

[*Tableau 2 – Fiche technique de l’entité SAPRESS 18*](#_bookmark14)

[*Tableau 3 – le processus Pick-in 33*](#_bookmark27)

[*Tableau 4 – le processus Cross docking 34*](#_bookmark28)

[*Tableau 5 – le processus Cross docking 36*](#_bookmark29)

[*Tableau 6 – le processus Cross docking 40*](#_bookmark31)

[*Tableau 7 – les étapes du retour de fons 41*](#_bookmark32)

[*Tableau 8 – le processus de retour 43*](#_bookmark33)

[*Tableau 9 – Le rôle des acteurs dans la demande messagerie 61*](#_bookmark58)

*Tableau 10 – Description du processus Demandes 61*

[*Tableau 11 – Les règles de gestion des Demandes 61*](#_bookmark59)

[*Tableau 12 – Le rôle des acteurs dans le volet Clients et Vendeurs 62*](#_bookmark60)

*Tableau 13 – Description du processus Création client 62*

[*Tableau 14 – Les règles de gestion des comptes clients et vendeurs 62*](#_bookmark61)

[*Tableau 15 – Le rôle des acteurs dans la facturation et paiement 63*](#_bookmark62)

[*Tableau 16 – Description du processus facturation & paiement 63*](#_bookmark63)

[*Tableau 17 – Les règles de gestion de la facturation et paiement 63*](#_bookmark64)

[*Tableau 18 – Le rôle des acteurs dans les caisse et retours de fonds 63*](#_bookmark65)

[*Tableau 19 – Les règles de gestion des retours clients 65*](#_bookmark66)

[*Tableau 20 – Le rôle des acteurs dans les retours clients 66*](#_bookmark67)

[*Tableau 21 – Les règles de gestion des retours clients 67*](#_bookmark68)

[*Tableau 22 – Le rôle des acteurs dans l'amélioration du portail 67*](#_bookmark69)

[*Tableau 23 – Description de l'amélioration des portails clients 68*](#_bookmark70)

[*Tableau 24 – Les règles de gestion sur le portail 68*](#_bookmark71)

## Bibliographie

* <https://konekt.agency/gestion-dentreprise/top-10-des-meilleurs-erp/>
* <https://www.glossaire-international.com/pages/tous-les-termes/messagerie.html>
* <https://fr.wikipedia.org/wiki/Odoo>
* https://[www.allegrologistics.co.uk/courier-delivery-service-how-does-it-work-and-](http://www.allegrologistics.co.uk/courier-delivery-service-how-does-it-work-and-) what-are-its-benefits/
* https://[www.gantt.com/fr/](http://www.gantt.com/fr/)
* https://[www.silog.fr/sites/silog/files/Livre-blanc-SILOG-5-etapes-clefs-pour-reussir-](http://www.silog.fr/sites/silog/files/Livre-blanc-SILOG-5-etapes-clefs-pour-reussir-) son-projet-ERP.pdf