**DASHBOARD**

**Application de mesure de performance pour aide a la prise de décision d’un Restauration**

SHEY Louis CHIA

Master-1, IDEV

# REMERCIEMENTS

Je remercie OUSMANE WADE, mon enseignant de gestion de projet qui m’a formé sur la gestion des projets, ce qui m’a permi d’obtenir par la suite une certification SCRUM et à bien mener ce projet jusqu’a bout. Je remercie également l’équipe pédagogique de Webitech pour la formation ainsi que ma famille pour m’avoir soutenu pendant cette période où tout n’a pas été facile. Enfin, j’exprime ma gratitude à tous ceux qui de près ou de loin m’ont soutenue dans ma formation.

# RESUME

# INTRODUCTION

Le Rapport d'activité pour les étudiants de 4ème année de Webitech fait partie intégrante de la formation en Big Data et Intelligence Artificielle tout en constituant le virage essentiel de leur intégration dans le monde du travail.

C’est ainsi que j’ai travaillé sur mon projet personnel de mise en place d’une application pour resoudre un problem que j’ai remarqué dans un restaurant fictive, cequi m’a permis de mettre en pratique les notions appris pendant mon année académique, de faire valoir mes connaissances et monter en compétences de la bonne manière.

# PARTIE I: PRESENTATION DU PROJET

# I – Presentation DASHBOARD

1. Dashboard

Dashboard est une application qui complemente l’application principale du restaurant “La Pêche” en produisant des rapports ou statistique sur les commandes/achats, les clients, les produits/menus et category de produits sur les cinq (05) villes pilotes ou se trouve le restaurant.

1. Visions et mission

La vision de DASHBOARD est d’être un outil qui aide des entreprises à avoir plus de visibilité sur leur performance.

La mission de DASHBOARD concernant “La Pêche” restaurant est de tracker sa performance, voir en temps reels ces données et permet une meillieur appreciation sur le fonctionnement du restaurant et aide à la prise de decision stratègique telque :

* Le client qui à faire le plus gros achat (on peut prevoir un cadeau pour celui-ci)
* Les produits les plus achétés (une classification des 05 meillieurs produits)
* Le nombre total de commandes
* Le pourcentage de depenses entre les hommes et les femmes
* Le pourcentage de depenses entres les differrentes tranches d’âges (les adolescents, les jeunes et les adultes)
* Le pourcentages des commandes classes par villes
* La liste total des clients
* Statistiques sur les habitudes de consummation par villes

II - ECOSYSTEME DASHBOARD

1-Les partenaires

Le partenaire principale de DASHBOARD est le restaurant “La Pêche” don’t l’experimentation a été fait. Il faudrai quand meme relever que DASHBOARD compte s’étendre à toutes les entreprises partenaires de “La Pêche”. Il faut noter que si l’on peut traquer les entreprises partenaires de “La Pêche” (notamment ceux qui approvision en matières premiers pour la restauration) avec leur accords s’il n’ont pas déja un système mise en place, on pourai anticiper du coté de “La Pêche” comment combler aux lacunes d’un stock et anticiper sur sa commande chez une autre entreprise.

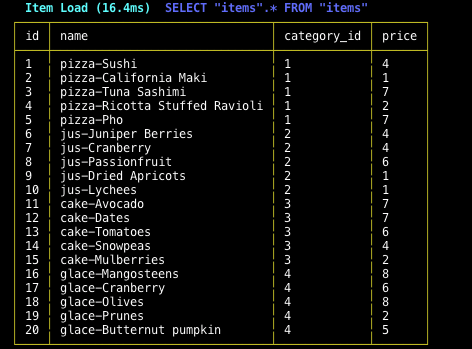
On pourrai également après observation prévoir s’il est nécéssaire de commencer la production d’une matière prémière nècessaire pour le bon fonctionnement du restaurant “La Pêche”.

2- Partenaires prospet

DASHBOARD souhaite étendre ses services aux maximum de restaurants. Mais il souhait proposer ses services après experimentation et observation avec son partenaire principale “La Pêche”. Apres cette première étape de perfectionnement de l’outil taillé pour un restaurant et ensuite pour d’autres restaurants, le but est de créer un outil de monitoring pour aider n’importe quel entreprise pour ameliorer sa gestion. Ainsi, au moment opportune, le pus de partenaires possibles sera la bienvue.

# PARTIE II : MISE EN APPLICATION DE DASHBOARD DANS UN RESTAURANT “LA PÊCHE”

Le restaurant “**La Pêche**” est un restaurant installé dans 05 grandes villes de France (Paris, Marseille, Lyon, Bordeaux, Nice). Elle vend 04 catégories de produits alimentaires (pizza, boissons, gateau et glaces). Pour l’instant, il exist 20 produits differents telque indiqué ci-dessous:



# I – PROBLEM DE “LA PÊCHE”

Je n’arrive plus a vendre certains produits,

# II - LES HYPOTHÈSES DE RECHERCHE

1. guide d’entretien du personnel
2. etude de marché de l’écosysteme de {name}
3. Résultats attendus

{name} souhaite

* une augmention de son chiffre d’affaire d’au moin 40% en un (01) an
* une meillieur visibilité de son dashboard

(plats commandés, number command par jours, nbre clients etc…)

- meillieur gestion des stocks

# III – LA PROBLEMATIQUE DEGAGÉ

Comment est-ce que DASHBOARD va resoudre les problemes de

gestion de stock, augmentation du chiffre d’affaires et avoir une meilleur visibilité sur la gestion du restaurant

# PARTIE III : SOLUTION PROPOSÉ

# I - LA MÉTHODOLOGIE UTILISÉE

1 – langages et outils utilizes

- HTML/CSS – HTML est un langage de balisage servant à écrire des pages web et le CSS est un langage de formatage de pages web

* Javascript/React – Javascript est utilisé pour implementer la logique coté front-end surtout la consummation des web services, le formattage et la presentation des données récupérées. React est une framework javascript .
* Ruby/Ruby on Rails – Ruby est un langage de scripting utilisé coté back-end. J’ai utilisé plus précisement une framework Ruby nommé Ruby on Rails
* Faker : Faker est un GEM Ruby permettant la production des données test. Elle a été utilisée pour la production des clients tests, les menu tests, les achats effectués dans notre restaurant fictif.
* database\_cleaner : C’est un GEM Ruby utilisé pour vider la base de données tests afin de le re-initialiser pour pouvoir recommencer les tests.
* Irbtools : est un GEM Ruby permettant une affichage des données sous forme de tableau (plus lisible) quand on faire des requête sur la console du serveur web
* SQLite3 : SQLite3 est une base de données gratuite et compacte que vous pouvez utiliser facilement pour créer et utiliser une base de données. Bien que SQLite3 ne soit pas une base de données complète, il prend en charge un ensemble étonnamment large de normes SQL et est idéal pour ceux qui commencent tout juste à apprendre SQL ainsi que pour les développeurs qui ont besoin d'un moteur de base de données simple pour se connecter à leurs applications.
* rack-cors : est un GEM Ruby permettant de resoudre les problèmes CORS. Avec CORS par défaut, toutes les demandes entre deux origines différentes sont bloquées. Ceci afin d'empêcher les gens d'accéder aux données / API sur des serveurs qu'ils ne contrôlent pas.
* SASS : Sass est un pré-processeur CSS Sass est totalement compatible avec toutes les versions de CSS Sass réduit la répétition de CSS et fait donc gagner du temps Sass a été conçu par Hampton Catlin et développé par Natalie Weizenbaum en 2006
* VSCode : mon éditeur de texte
* GIT/Github: GitHub, Inc. est un fournisseur d'hébergement Internet pour le développement de logiciels et le contrôle de version à l'aide de Git.Il offre la fonctionnalité de contrôle de version distribué et de gestion du code source de Git, ainsi que ses propres fonctionnalités.
* - JSON : format de structuration de données en javascript utilisé  notamment par AJAX pour retourner les réponses et transmettre des  informations au serveur.
* Hiechart : Une bibliothèque JavaScript robuste qui permet aux développeurs de créer facilement des graphiques interactifs pour allouer, coordonner et afficher des tâches, des événements et des ressources le long d'une chronologie.

2 – justification des outils utilisés

Le choix des outils a été fait en tenant compte des avancés des technologies web (choix des langages, choix de framework, choix des techniques etc.)

* **Ruby/Ruby on Rails :**

Afin de garantir :

• Un gain de productivité,

• Un travail en groupe entre développeurs du projet,

• Une clarté dans l’organisation des sources du projet grâce à

l’architecture « MVC »,

• Une maintenance évolutive et plus aisée du système,

• Une réutilisation des composants sans les réinventer complètement,

• Une robustesse du code, l’utilisation d’un Framework est apparue,

incontournable pour mener à bien ce projet.

Mais qu’est-ce qu’un Framework ? Un Framework est un ensemble de composants qui servent à créer les fondations, l'architecture et les grandes lignes d'un système (logiciel, site web...). Il en existe des centaines pour différentes langages de programmation. Mais en revanche, cet outil demande aussi une courbe d’apprentissage plus élevée par rapport au langage natif dont il se sert pour faciliter les tâches du développeur. Ruby est utilisé pour contruire le backend pour l’application du restaurant pilote (“La Pêche”). Elle a été utilisée pour mettre sur pied les web services consommés par DASHBOARD.

* Javascript/ReactJs : JavaScript est un langage de programmation utilisé pour rendre les pages Web interactives. C'est ce qui donne vie à une page: les éléments interactifs et l'animation qui engagent un utilisateur.

React (une librarie javascript) facilite la création d'interfaces utilisateur interactives. Concevez des vues simples pour chaque état de votre application, et React mettra à jour et restituera efficacement les bons composants lorsque vos données changent. Elle est utilisée pour mettre sur pied DASHBOARD pour la consommation des web services mis sur pied sur le backend de “La Pêche”

* Hiechart : est un tres bon outil avec des bon proposition de graphiques pour presenter des statistiques. Etant un package npm, cela m’a permit de bien l’integrer dans ReactJs.

# II – IMPLEMENTATION DE DASHBOARD

1. MODELISATION DU SYSTEME

Dans cette partie, nous nous évoquerons la modélisation du système «LA PECHE» qui va nous permettre de ressortir le système d’information associé. Pour se faire, le langage de modélisation UML sera employé, avec SQLite comme SGBD (gestion de la base de données).

1. **Les acteurs du système**
2. **Le client**

Sera considéré comme visiteur, tout internaute se connectant sur le site « LA PECHE » avec un compte utilisateur, il pourra en outre :

• modifier les parametres de son compte.

• Se créer un compte « candidat » ou « recruteur ».

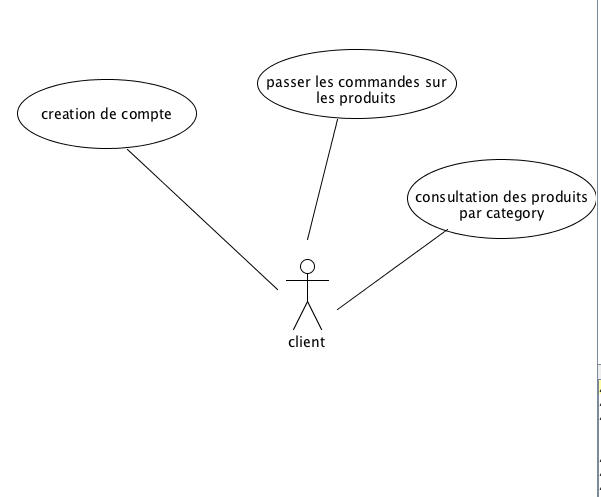
• Faire un achat dans le site a base des produits proposé.

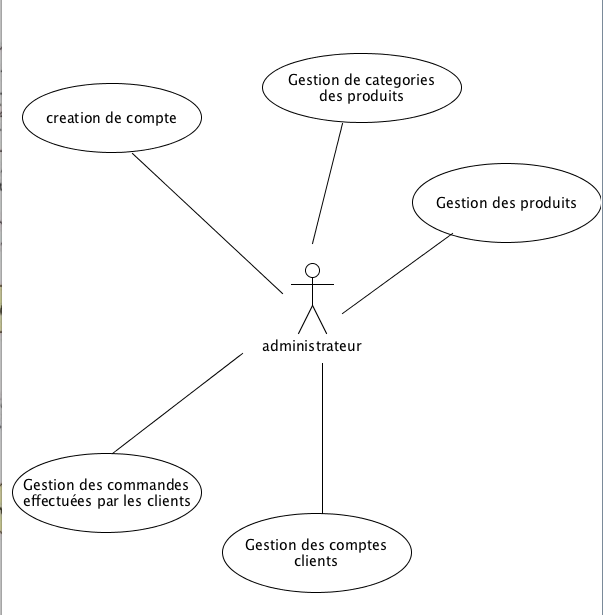
1. administrateur

* Il constitue ici la face cachée du système pour tout internaute. Son rôle est de gérer dans l’ensemble les différentes fonctionnalités du site dans son ensemble, de surveiller leurs activités, de mettre à jour le site. Il pourra donc :
  1. - ajouter/supprimer/modifie la categorie des produits dans le restaurant
  2. - ajouter/supprimer/modifier les produits proposés par le restaurant
  3. - Gérer les comptes des clients
  4. - consulter et voir l’ensemble des achats effectués dans le restaurant
  5. - Consulter les différents états d’activités sur le site...

** Les cas d’utilisation « Use Cases »**

Il s’agit ici tout simplement de schématiser les différentes actions que pourront effectuer les acteurs du système mentionnés précédemment.

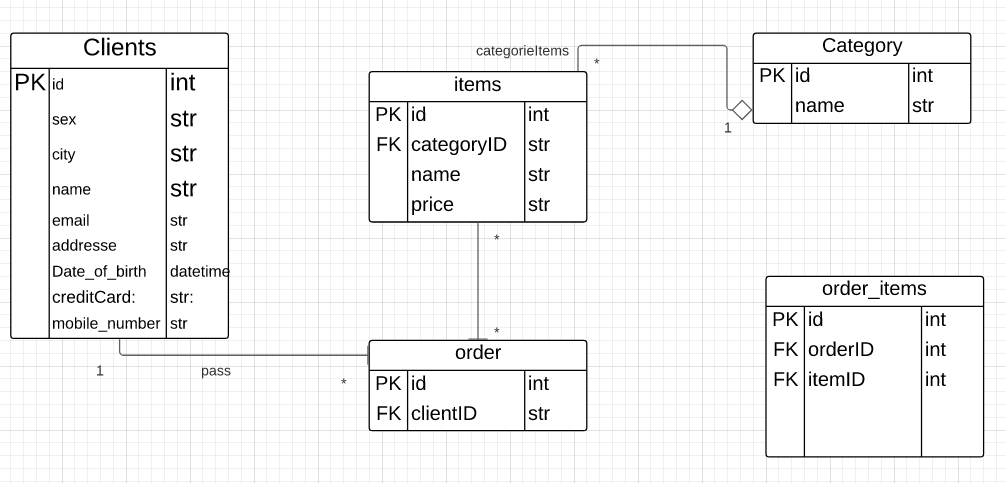
* 
* **Figure 1 : Diagramme cas d’utilisation : le client**

****

* **Figure 1 : Diagramme cas d’utilisation : l’administrateur**

iii ) Diagrammes de classes

En génie logiciel, un diagramme de classes dans le langage de modélisation unifié (UML) est un type de diagramme de structure statique qui décrit la structure d'un système en montrant les classes du système, leurs attributs, opérations (ou méthodes) et les relations entre les objets.



int – interger/entire

str – string/chaine de caractere

datetime – format de date

PK – Primary Key/Clé primaire

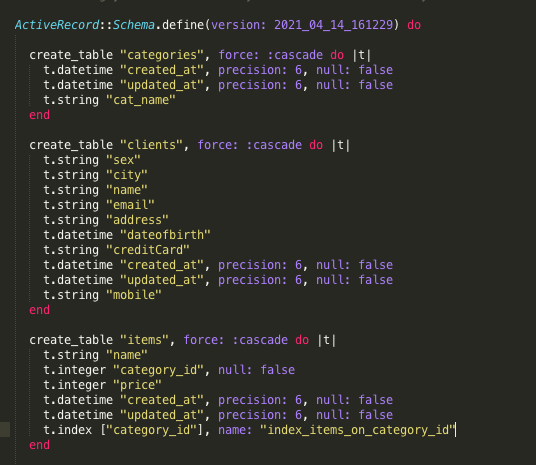
FK – Foreign key/ Clé étrangère

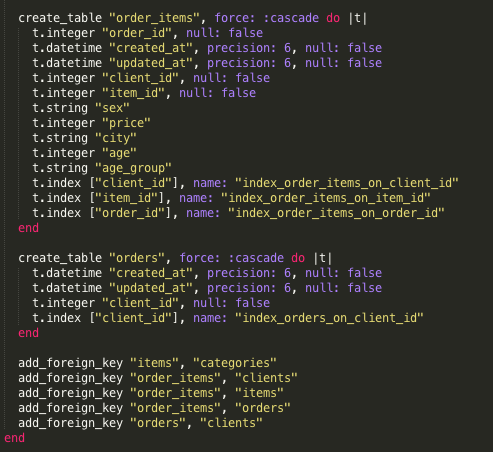
iV ) Vue générale du système d’information

Le système d’information d’ « LA PECHE » a été implémenté sur SQLite comme SGBD. Il s’agit ici, à ce niveau de réalisation du projet, de passer à la structuration des informations concernant l’application web dans une base de données. Grâce aux règles régissant le passage d’un modèle conceptuel de données matérialisé ici par nos diagrammes de classes, à un modèle logique de données donc à la base de données, nous avons généré le système d’information.

Sous Ruby on rails, nous avons le fichier « *{DOSSIER\_DU\_PROJET}/db/schema.rb »* qui est reproduit à chaque fois que vous exécutez une migration. Il reproduit l'intégralité du schéma de votre base de données. Il est également utilisé par certaines commandes de migration dans Rails pour éviter de refaire toutes les migrations une à une (dans le cas d'une réinitialisation de migration par exemple).

C'est aussi le seul endroit où vous pouvez voir toutes vos tables à la fois pour la référence des colonnes.





in order to create the appropriate database data from this `schema.rb` file in Ruby on rails, the following command needs to be executed on the terminal:

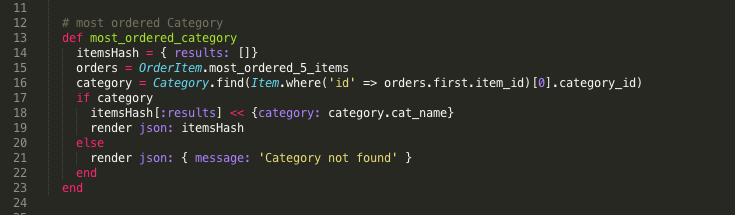
`rails db:migrate`

V ) implementations des Web services

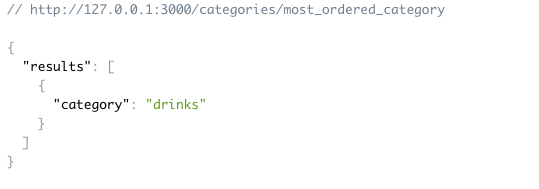
Après avoir construit les bases de données du restaurant «LA PECHE», il a fallu implémenter des web services (API) au niveau des contrôleurs et exposer ces données afin qu'ils puissent être consommés par une application front-end. Ces web services consist principalement à interroger la base de données afin d'obtenir diverses statistiques concernant les informations que le restaurateur souhaite tracker et les rendre disponibles (avec Ruby on rails) sous la forme de JSON, qui est le format de données le plus utilisé pour les API.

Quelques examples de web services

- web service pour la catégorie de produit la plus commandée:



accessible via l’url `*http://127.0.0.1:3000/categories/most\_ordered\_category*`, elle produit le resultat JSON ci-dessous:



* web service pour demontrer les proportions d’achat des produits par tranches d’ages:

# Macintosh HD:Users:macbookpro:Desktop:Screen Shot 2021-05-13 at 2.28.01 AM.png

0 – 26 ans : represente les adolescents

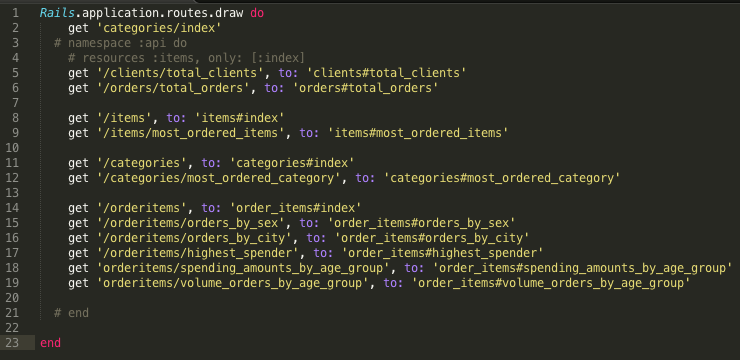
27 – 35 ans represente les jeunes

35 ans et plus, represente les adultes

accessible via l’url `*http://127.0.0.1:3000/categories/most\_ordered\_category*`, elle produit le resultat JSON ci-dessous:



Pour voir la liste complet des web services disponibles, il suffit de regarder l’ensemble des URL mise a` disposition depuis le fichier ‘*app/config/routes.rb*’ don’t le contenu est ci-dessous:



Il est aussi possible de verifier l’ensemble des chemins d’accès via le terminal avec la commande : ‘*rails routes*’

# III - LES RÉSULTATS SOUHAITÉS

-Vise augmentation 20% en 06 mois, et 35% en 1 un

-employés formés a la gestion des stocks via DASHBOARD

# IV - LES OUTILS DE CONTRÔLE ÉVENTUELLEMENT MIS EN PLACE

- alerts, newsletter, email, impression pdf

# CONCLUSION

*{talk about things that can ameliorated, or unfinished tasks}*

# WEBOGRAPHIE

* <https://www.w3schools.com/sass/sass_intro.asp>
* <https://blog.webdevsimplified.com/2021-05/cors/>
* <https://en.wikipedia.org/wiki/GitHub>
* <https://www.highcharts.com/>
* https://www.thoughtco.com/what-is-javascript-2037921