*Projet @vosClick*

***Introduction***

Le projet informatique étudié utilise une architecture client-serveur, avec un Front-end Angular 13 ,

Un back-en Laravel 7, une base de données MySql, et l’utilisation de JSON pour les échange de données. Cette architecture offre une séparation claire des responsabilités entre les différents composant .

L’objectif de ce dossier est d’analyser en détail cette architecture et d’identifier les force et les faiblesse de cette application .Nous examinerons également les amélioration possibles pour optimiser les performances , la securité et l’expérience utilisateur.

1/***Contexte***

1.1/**Origine du besoin :**

L’idée est de mettre en relation les utilisateurs et les professionnels du bien-être via une application ; aider les utilisateurs à trouver des services adaptés à leur besoins.

Le but de l’applications est d’aider les utilisateur à améliorer leur santé physique et mentale en leur fournissant des outils(vente article), des conseils et des ressources qui leur permettent de prendre soin d’eux même(vente de services), d’améliorer leur vie de manière durable.

1.2/ **Fonctionnalité de l application**

Voici quelque fonctionnalités clés que l’on peut inclure dans notre application :

* Rechercher un professionnel(vendeur) en fonction de ses besoins de sa localisation géographique ,éventuellement cette recherche pourra inclure des filtre tel que la catégories de service, le prix, les commentaire et évaluations des utilisateurs pour aider les utilisateurs a prendre leur décision a deux prestation identique .
* Dans un premier temps la prise de rdv une fois la prestation achter se fera par téléphone

Mais il sera intéressant par la suite de mettre en place dans l application un calendrier de disponibilités, des notifications de rappel de rendez vous

* Dans un premier temps tous les paiement se feront en ligne avec un compte paypal , par la suite il sera intéressant de développer les autre forme de paiement(carte bleu) ;

Pour les article il y aura un choix de proposition de livraison en fonction de la zone géographique (livraison express, point relais, domicile) avec un surcoût éventuel.

* L’application devra permette aux utilisateur de communiquer avec les professionnels via une messagerie intégrer ,cela permettra aux utilisateur de poser des question et de clarifier les détails du service ou description de l’article
* Les utilisateur pourront laisser des commentaires et des évaluations des professionnels avec lesquels ils on travaillé

1.3**/ Situation Actuel**

1.3.1/**IHM(partie front)**

Actuellement l’application à une IHM avec des fonctionnalités qui ne sont pas complètement fini telque:

* Navbar (Esotérisme,Bien-être,Yoga,Voyance)sur le coter est relier à aucun élément
* Le lien Enrgeistrer un produit renvoie à un formulaire de création de compte



* Lien Accueil et home permette à l’utilisateur de voir ou il se trouve dans l application (a ce jour non exploiter pas de composant lier) ses lien son dans un bandeau fixe et disparait avec le scroll de la page (voir amélioration ou utilité)
* IHM est constituer de 3 bouton pour les recherche d’article, service, vendeur (fonctionnalité principale : voir annexe 1)
* La liste des articles ou services ou des vendeurs est disponible via les 3 boutons centraux
* Les boutons recherche peuvent être associé à des filtre prix, catégories, offre spéciale
* La fonctionnalité « triée par »  est non fonctionnel (menu déroulant en place sur l’IHM)
* Formulaire inscription ne permet pas de créer un compte à ce jour (info retranscrit sur la console du navigateur)
* Pas de champ password dans le formulaire d’inscription
* Lorsque 2 utilisateur différents acheter un article ou service celui affiche les 2 noms panier cumuler des utilisateur avec paiement différencier (a voir lord d’un compte connecter)

Les prochaines réunions vont permettre de voir le différent fonctionnalisé que le client désire améliorer ou développer (IHM et useCase en annexe 1)

1.3.2/ **Mockage des donne(** partie front)

JSON Server est un outil qui permet de créer rapidement une API RESTful à partir d'un fichier JSON.

Il a permis aux développeurs front de créez de faux service web pour tester rapidement l’IHM sans base de donnée relier le fichier json comme source de données est constituer de :

"id": l'identifiant unique de l'article ou du service

"seller": un objet contenant des informations sur le vendeur de l'article ou du service, telles que son nom, son adresse email, sa date de création et de mise à jour, ses informations de paiement, sa catégorie de vente, etc.

"creationDate": la date à laquelle l'article ou le service a été créé.

"updateDate": la date à laquelle l'article ou le service a été mis à jour pour la dernière fois.

"category": un objet contenant des informations sur la catégorie de l'article ou du service, telle que son identifiant unique et son nom.

"name": le nom du service ou de l’article

"description": une description de l'article ou du service

"tag": description de l’article ou service.

"acceptedPayments": les modes de paiement acceptés pour l'article, tels que PAYPAL, VISA, etc.

"pricingOptions": un tableau contenant les différentes options de prix pour l'article, telles que le nom de l'option, le taux de TVA appliqué, le prix de l'article, etc.

"renewal": un indicateur de renouvellement automatique pour l'article.ou service

"renewalDate": la date à laquelle l'article sera renouvelé automatiquement.

"images": une liste d'URLs pour les images de l'article.

"video": une URL pour une vidéo de l'article.

"bvideo": un indicateur de disponibilité d’une vidéo.

"stock": le nombre d'articles disponibles en stock.

"deliveryTypes": une liste d'options de livraison disponibles pour l'article, tels que DHL, UPS, etc

(Arborescent de l’objet complet en annexe 2)

1.3.3/Partie Angular(front-end)

Actuellement l’application est sous la version Angular 13.3.0 , qui est une Framework structuré et basé sur Type-Script (langage fortement Type), qui facilite les erreur et la maintenance du code,

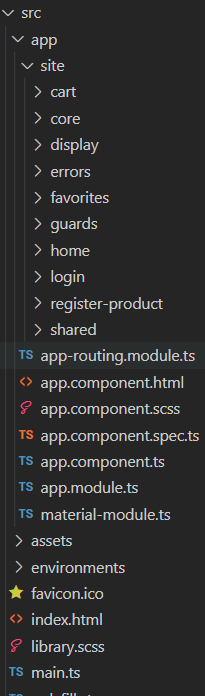
il est complexe et sa courbe d’apprentissage est importante pour les développeurs débutants il est compatible avec la plupart des navigateurs modernes et offre de bonnes performances en termes de temps de chargement et d’exécution de l’application (lazy-loading),

Il est base sur des composants réutilisable, et services commun à l’application

On utilise le Framework bootstrap pour le css et la librairie Material cela simplifie le développement, la mise en page et permet une application responsive.

Pour notre analyse compodoc a étais rajouté afin de bien comprendre les différents éléments en place leur dépendance, les interconnexions avec les services (voir annexe)

L’arborescence fichier et décrie ci-dessous :



Celle-ci devra être revue et analyser et éventuel a rediscuter avec le client (dossier en double ex article service, etc..)

Il y a des composants qui ne son pas utiliser voir non développer, lors de la réunion client il faudra discuter d’une stratégie de restructuration en priorisant les fonctionnalités principale.

Le projet est constituer de 13 module, 24 composants, 16 service ,2 classes, 37 interfaces(voir annexe pour schématisation 3)

1.3.4/partie Laravel

Le back end est supporter par le Framework Laravel 7 qui est basé sur le modèle d’architecture MVC (Modèle-Vue-Contrôleur) qui permet de développer des applications web de manière structurée, modulable et maintenable cette version a étais choisit pour des problèmes de compatibilité avec l’hébergeur web IONOS.

Il utilise le langage PHP 7.2.5 qui est verbeux et facile à apprendre, il possède une grande communauté de développeurs qui partage leur connaissance et leur package open source, il est gourmand en ressource par rapport a d’autre Framework PHP ,laravel 7 n’est pas compatible avec les anciennes version de PHP , la documentation est complète.

* La partie Back end étant a son état initial les fichiers des contrôleurs sont mit en place mais leur fonctionnalité ne sont pas ou très peu développer (ex : adminController partiellement développer)

A partir du projet récupère il ni a pas de fichier de migration mise a part ceux implanter par laravel à sa création

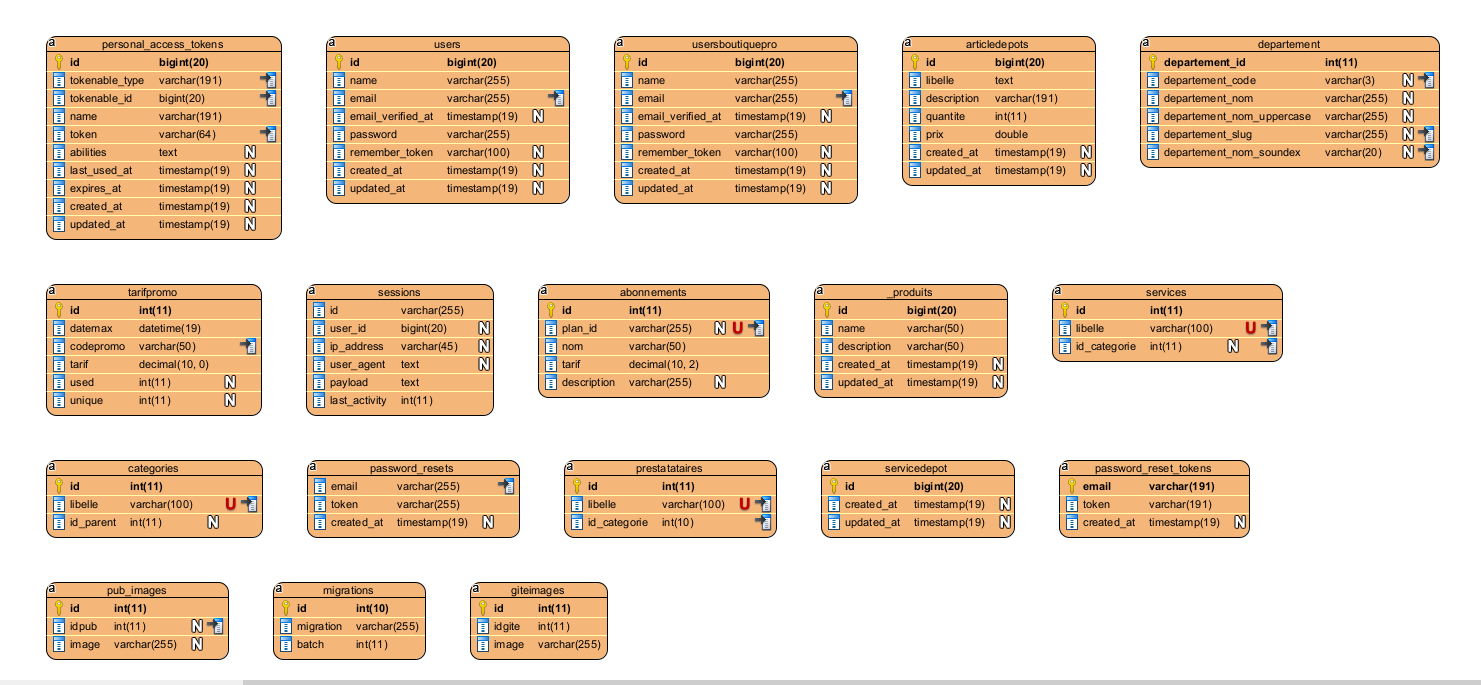
Sachant que Aujourd’hui Laravel peut être hébergé sur des web tels que IOnos mais les exigences minimales sont :

* PHP version 7.3 ou supérieure
* Extensions PHP : OpenSSL, PDO, Mbstring, Tokenizer, JSON, BCMath, Ctype, Fileinfo
* Serveur Web : Apache ou Nginx
* Base de données : MySQL 5.6 ou supérieure, PostgreSQL 9.4 ou supérieure ou SQL Server 2017 ou supérieur

La question de la migration ver la version supérieur devra être discutée lors des échanges client.

1.3.5/partie Base de données

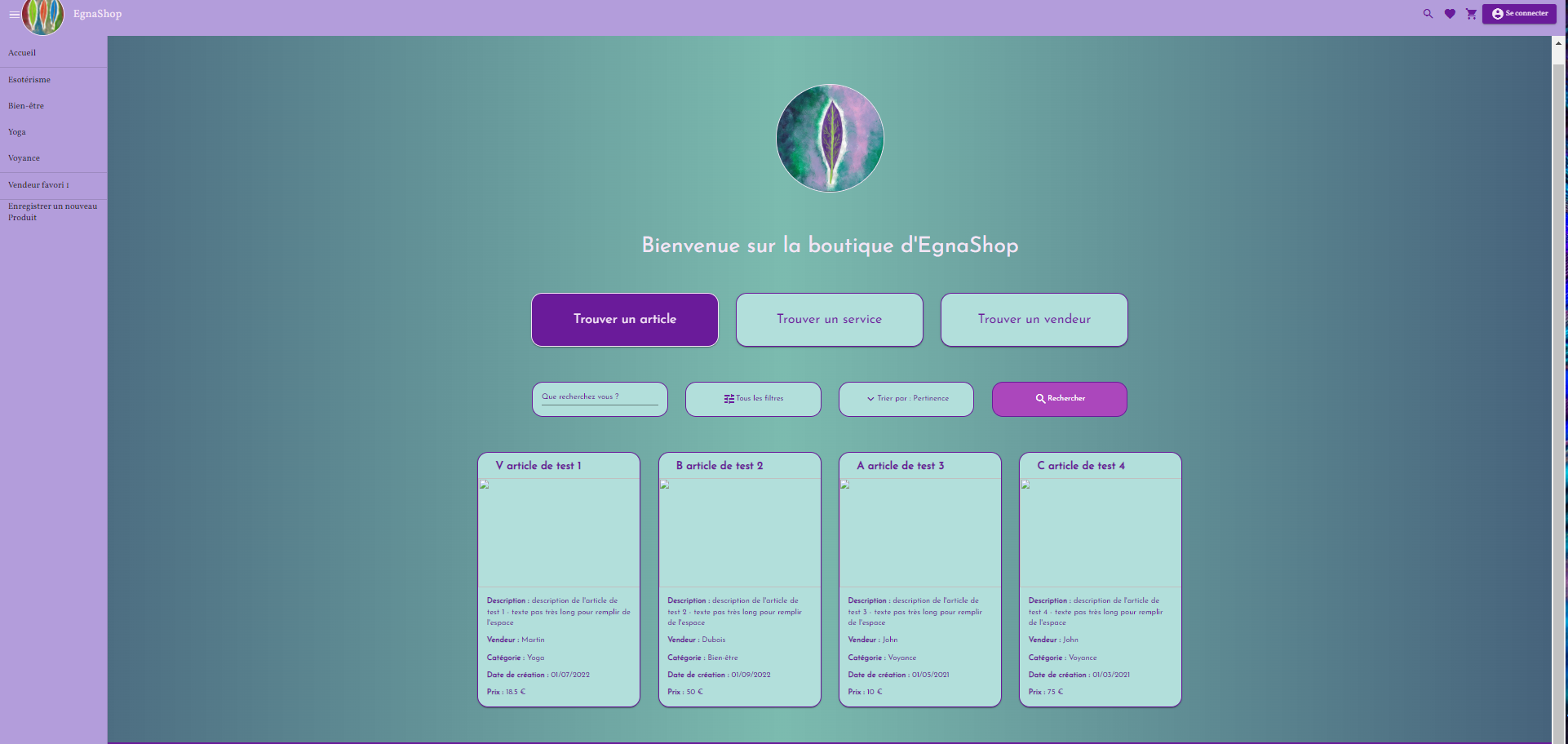
La mise en place et le déploiement a partir de la requête sql donne par le client nous on permit de schématise tables celle-ci sont ci-dessous :



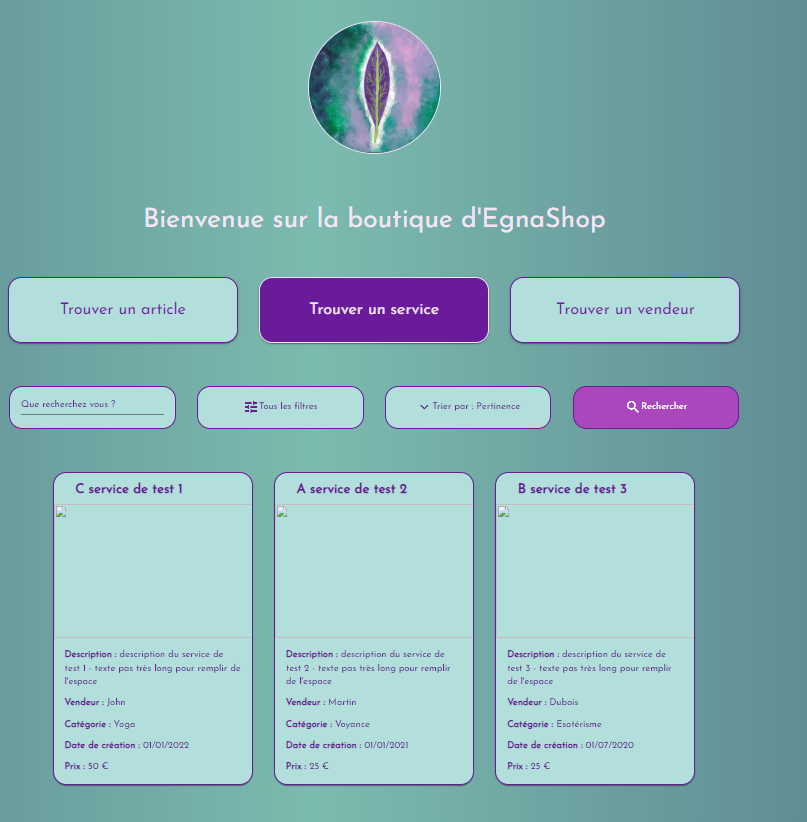
On constate qu’il ni pas de relation entre elle il faudra donc épurer les tables et en établir les associations entre elle, la conception d’un diagramme entités association se fera après le rdv client

**Anexe1**

* IHM page Accueil liste article



* IHM page accueil liste des services



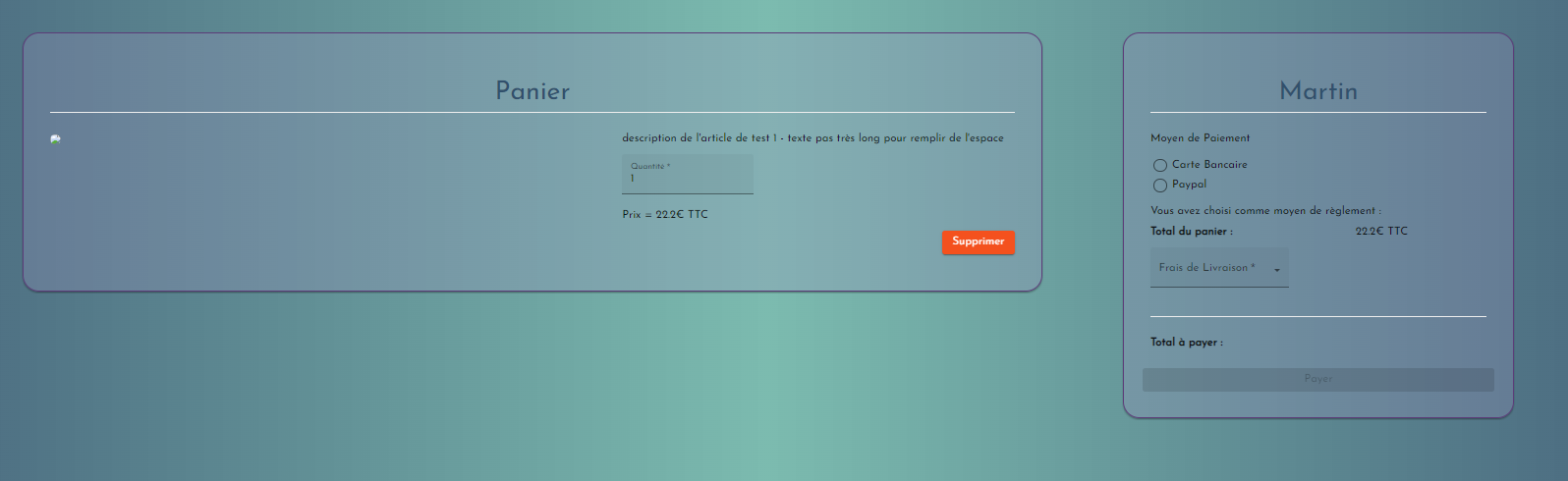
* IHM partie filtre



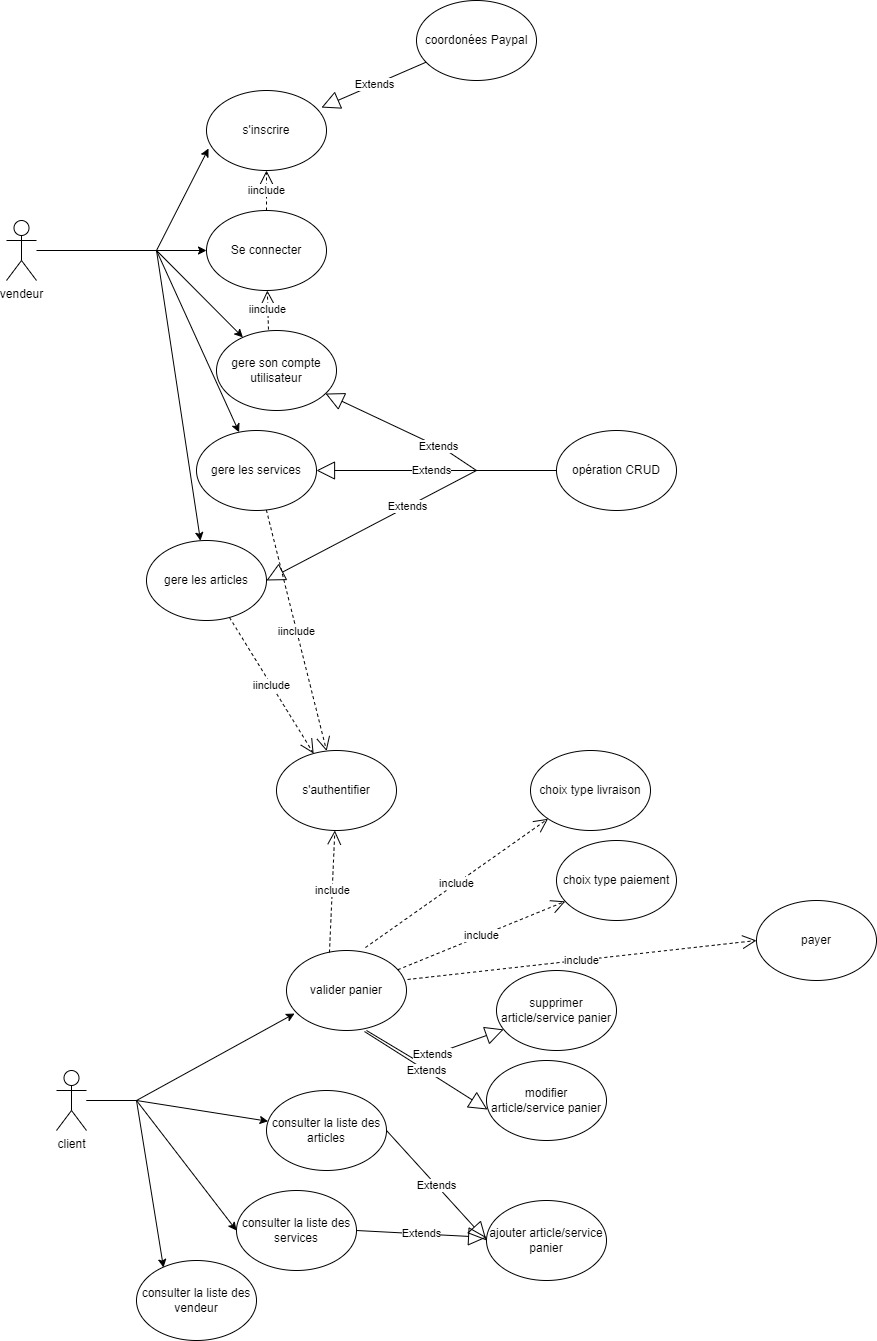
-IHM récapitulatif panier



* IHM panier



* Use Case



* **Annexe 2**

**-**Structure du fichier json

**"itemRecords": [**

**{**

**"id": 4,**

**"seller": {**

**"id": 1,**

**"name": "Martin",**

**"email": "vendeur1@seller.org",**

**"aboutUser": "infos du vendeur Martin, plus du texte pour remplir un peu d'espace pour la présentation",**

**"photo": "https://thiscatdoesnotexist.com/",**

**"birthday": "1990-05-01",**

**"address": [**

**{**

**"id": 1,**

**"address": "10, rue Martin",**

**"city": "Martinville",**

**"zipcode": "01001",**

**"country": "MartinLand"**

**}**

**],**

**"created": "2022-03-01",**

**"updated": "2022-08-01",**

**"orders": [],**

**"favorites": [],**

**"paypalPaymentEnabled": true,**

**"paypalReference": "ref paypal du vendeur test 1",**

**"isValidCGV": true,**

**"isValidML": true,**

**"isAdult": true,**

**"subscription": {**

**"id": 1,**

**"name": "subscription test service 1",**

**"description": "description de la subscription test service 1",**

**"price": 80,**

**"planid": 4**

**},**

**"categories": [**

**"Voyance"**

**],**

**"tag" : "tagtest2,tagtest3,tagtest5"**

**},**

**"creationDate": "2022-07-01",**

**"updateDate": "2022-09-01",**

**"category": {**

**"id": 1,**

**"name": "Yoga"**

**},**

**"name": "V article de test 1",**

**"description": "description de l'article de test 1 - texte pas très long pour remplir de l'espace",**

**"tag": "tagtest1,tagtest2,tagtest3",**

**"acceptedPayments": ["PAYPAL"],**

**"pricingOptions": [**

**{**

**"id": 1,**

**"name": "prix 1",**

**"appliedTVA": {**

**"rate": 20,**

**"countryCode": "Fr",**

**"tvaType": "tva type 1"**

**},**

**"price": 18.5**

**}**

**],**

**"renewal": "AUTO",**

**"renewalDate": "2022-11-01",**

**"images": [**

**"https://thiscatdoesnotexist.com/"**

**],**

**"video": "",**

**"bvideo": true,**

**"stock": 12,**

**"deliveryTypes": [**

**{**

**"id": 1,**

**"name": "DHL",**

**"price": 25**

**}**

**]**

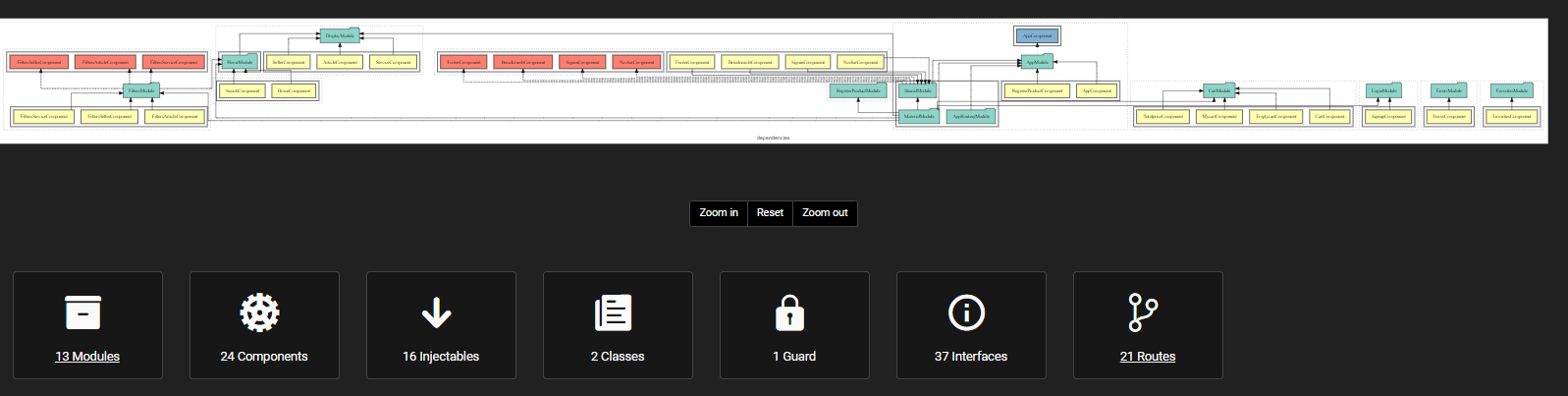
**},**

**{**

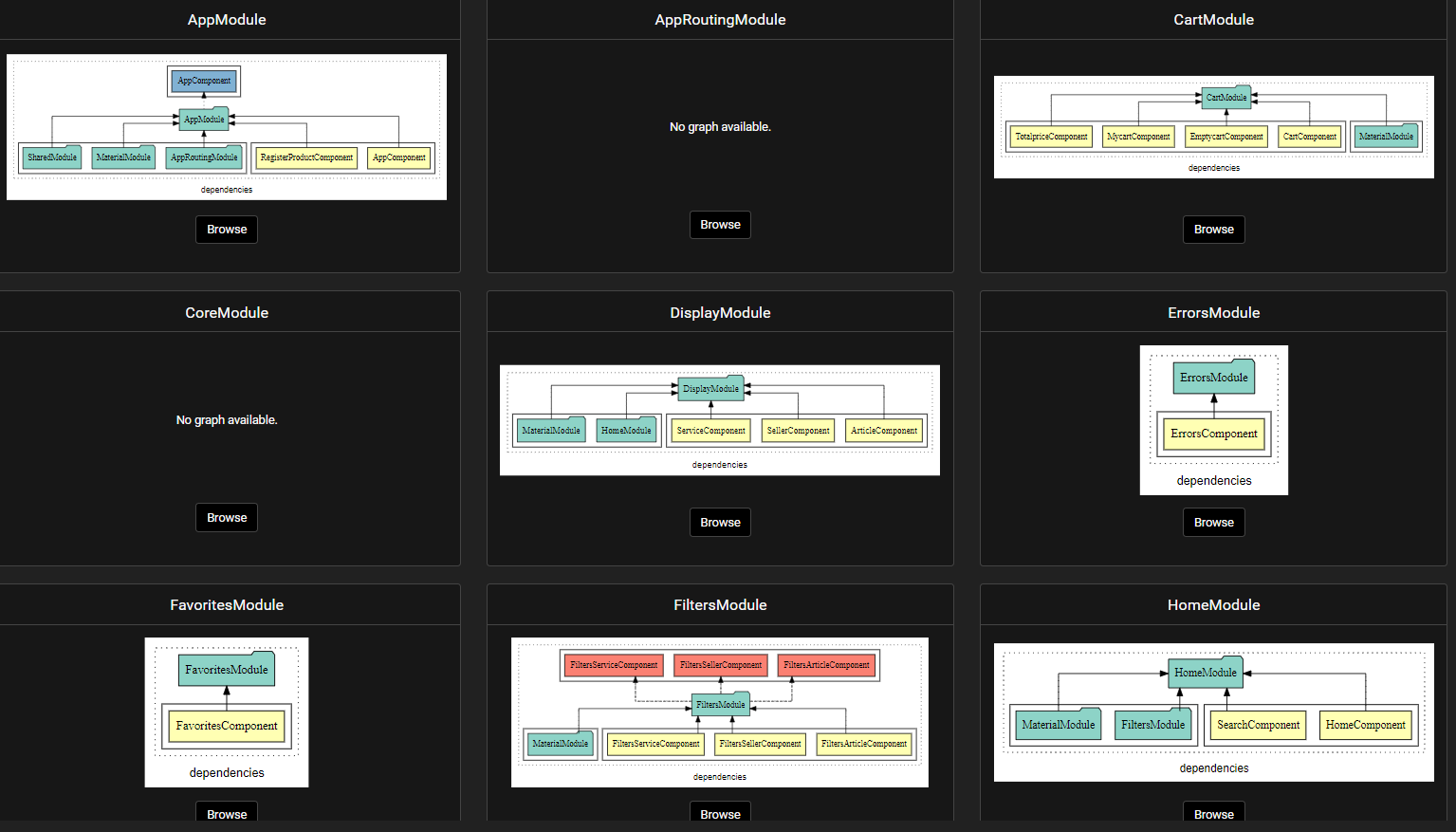
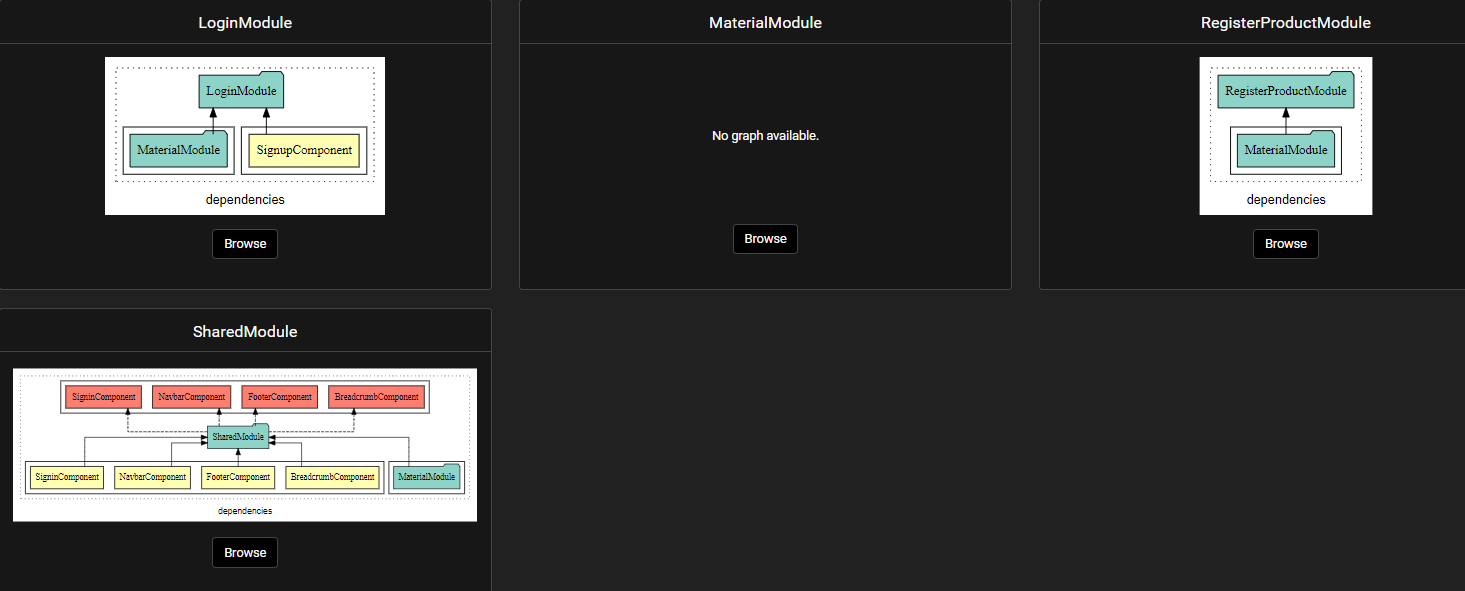
**Annexe 3 :**

Module et services et structure de l’application

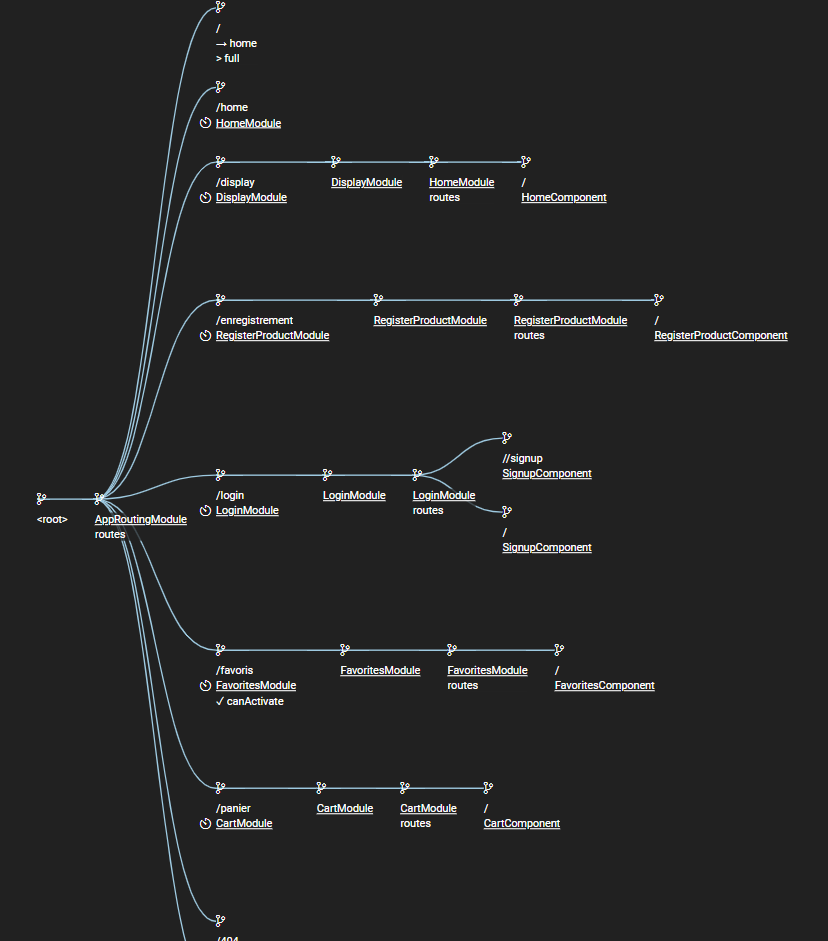
-structure de l’application



* Module front de l’apllication



* Routage



-Liste des interface et service

