

## Rapports développement d'applications réticulaires



Responsable de l'UE Dr.Romain Demangeon  
DAR

réalisé par  
K.FEKIR 3711938  
S.GHERSA 3525755  
N.CHOULLIT ;

19 novembre 2018

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Présentation de l'application</b>	<b>4</b>
<b>3</b>	<b>Manuel utilisateur</b>	<b>5</b>
3.1	Gestion de compte utilisateur . . . . .	6
3.2	Gestion des annonces . . . . .	7
3.3	Gestion des achats . . . . .	8
3.4	Consultation d'historique . . . . .	9
<b>4</b>	<b>Cas d'utilisation</b>	<b>10</b>
4.1	Reserver un produit . . . . .	10
4.2	Valider la réservation d'un produit . . . . .	10
4.3	Cycle de vie d'une annonce . . . . .	11
4.4	Scénario d'utilisation pour gestion de concurrence : . . . . .	11
<b>5</b>	<b>Technologies et architecture du projet</b>	<b>12</b>
5.1	Serveur . . . . .	12
5.2	Client . . . . .	13
5.3	Base de données utilisée . . . . .	14
5.4	API utilisées . . . . .	14
5.5	Sécurité . . . . .	15
<b>6</b>	<b>Points forts et faibles</b>	<b>16</b>
6.1	Points forts . . . . .	16
6.2	Perspectives et points à améliorer . . . . .	16
	<b>Introduction</b>	

# 1 Introduction

Actuellement, une application web permet de tout faire et se veut être facile d'utilisation pour l'internaute, moteur de recherche, shopping, réseaux sociaux, il y a quelques années de cela le web ne pouvait offrir qu'un contenu informatif statique et identique pour tout les utilisateurs.

Pour pouvoir suivre toute l'évolution dans le domaine informatique en général et dans le web plus précisément, les applications web ont connu une croissance vertigineuse pour adapter leur performances, celles-ci ont pu bénéficier et bénéficient toujours de différentes technologies permettant d'enrichir le contenu proposé à l'internaute d'une façon dynamique.

Effectivement, HTML ne permet pas de développer des fonctionnalités interactives. Ainsi, pour améliorer les fonctionnalités et l'interactivité des interfaces des applications, les développeurs ont utilisé des outils d'extension de leur capacités tel que les applets Java, les plugins Flash ou Silverlight, les widget, les machines virtuelles, . . . Aujourd'hui on parle beaucoup plus des librairies Javascripts. D'autre part, les applications web doivent non seulement s'adapter à différents types de supports ( smartphones, tablettes...), mais aussi fonctionner sur différents OS on parle alors d'applications hybrides.

Le projet présenté dans ce rapport est réalisé dans le cadre de l'unité *développement d'applications réticulaires*, est une excellente occasion pour mettre en oeuvre nos connaissances dans le développement web mais aussi de découvrir des technologies toutes nouvelles pour l'équipe, le rapport se présentera en trois grandes parties : pour commencer dans un premier temps nous présentons le manuel d'utilisation de l'application, ensuite l'architecture de cette dernière ainsi que les différentes technologies utilisées pour l'implémentation, pour finir avec des perspectives d'amélioration de l'application.

## 2 Présentation de l'application

Dans le cadre de ce projet nous avons réalisé une application pour annonces commerciales, permettant aux utilisateurs de déposer et consulter les annonces gratuitement en toute simplicité.

L'idée principale est venue pour réduire un phénomène de plus en plus répandu en îles de france qui est l'abandon d'objets encombrants sur les voies publiques, ainsi nous aidons les citoyens à faire un geste écologique et participer dans la réduction des dépenses étatiques inutiles. Nous avons fait le choix que ces produits soient payant, afin d'inciter le maximum de personnes à y participer.

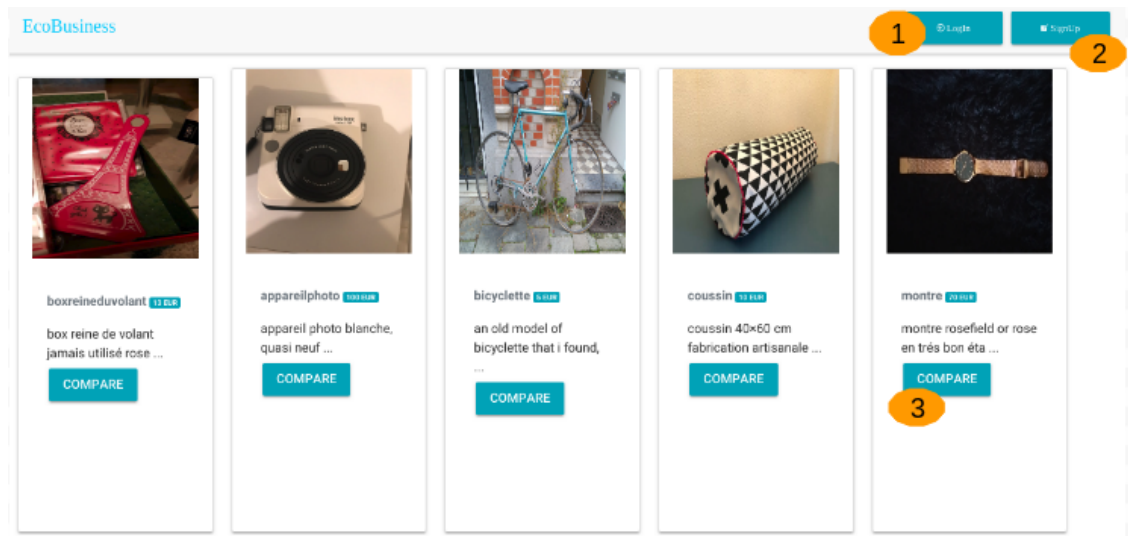
Nous avons donc créer [EcoBusiness](#) pour permettre aux utilisateurs de mettre en vente leur produits dont ils n'ont plus besoin.

[EcoBusiness](#) pour rassembler les deux buts principales de notre application à savoir un business écologique et économique au même temps. La figure suivante représente le Logo de notre application :



### 3 Manuel utilisateur

Tout d'abord nous présentons la page d'accueil qui donne un aperçu général sur le contenu de notre application, nous avons choisi de présenter les dix derniers produits mis en vente sur le site, comme le montre la figure suivante :



1. Redirige vers la page de connexion
2. Redirige vers la page d'inscription
3. Comparaison de prix : permet de comparer le prix du produit par rapport à d'autres magasins.

La totalité des fonctionnalités de l'application mise à part la comparaison exigent que l'utilisateur soit connecté.

Dans la section qui suit nous allons présenter les différentes fonctionnalités que nous proposons dans l'application et l'interface pour chacune .

### 3.1 Gestion de compte utilisateur

**Inscription** : Afin de créer un compte, l'utilisateur doit renseigner tout les champs de formulaire qui suit :

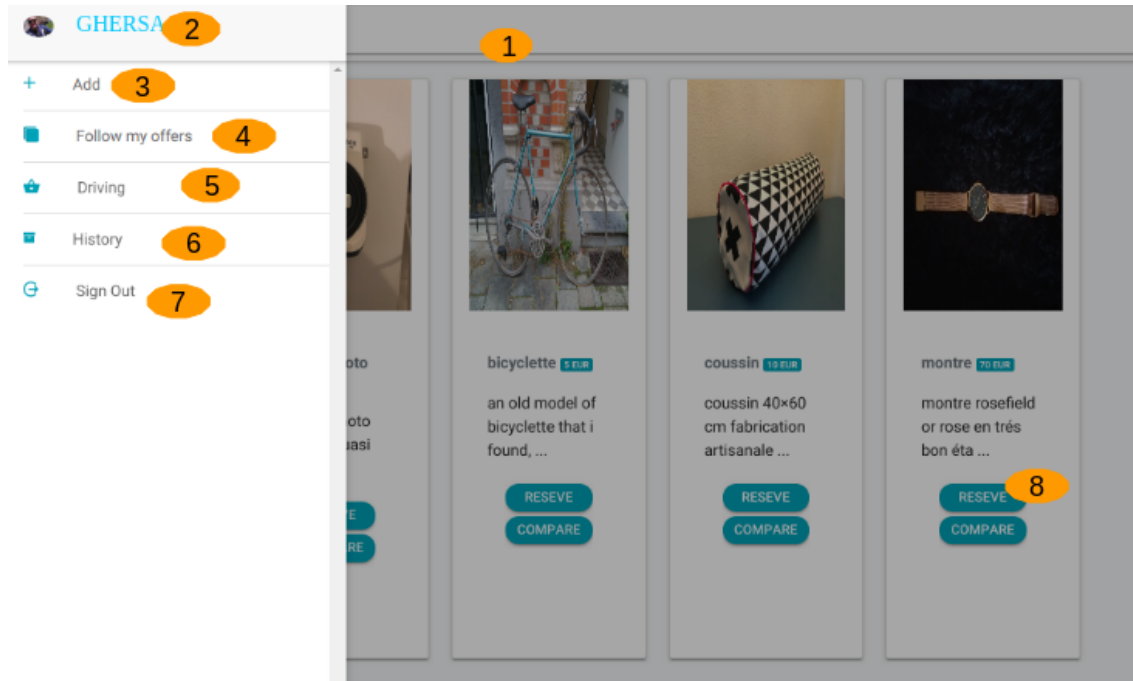
The screenshot shows a mobile application interface for user registration. At the top, there is a header bar with a back arrow and the text "Sign Up". Below this, a user icon is followed by the label "userName :" and the text "SofianeGHERSA". The "Email" field contains "m.ghersa.s@gmail.com". The "Password" field is empty, and a red error message "vous devez saisir un mot de passe" is displayed next to it, with an orange circle containing the number "2" indicating this is the second error. Below the password field is the "Confirm Password :" label. The "user Address :" field contains "34 RUE VICTOR BASCH". The "user Phone :" field contains "0755115459". The "user Picture :" field has a button labeled "Choisir un fichier" and the text "sofiane.jpg". At the bottom, there is a large blue button labeled "SIGN UP" with an orange circle containing the number "1" indicating it is the first element.

1. Inscription

2. Erreur correspondant à un champ mal renseigné, ici absence de mot de passe

Il existe d'autres fonctionnalités basiques que nous allons pas détailler dans cette section vu qu'elles sont standards et identiques à la majorité des applications web : [Connexion](#), [Déconnexion](#) et enfin [modification de profil](#).

Lors de la connexion l'utilisateur est redirigé vers la page principale, qui contient :

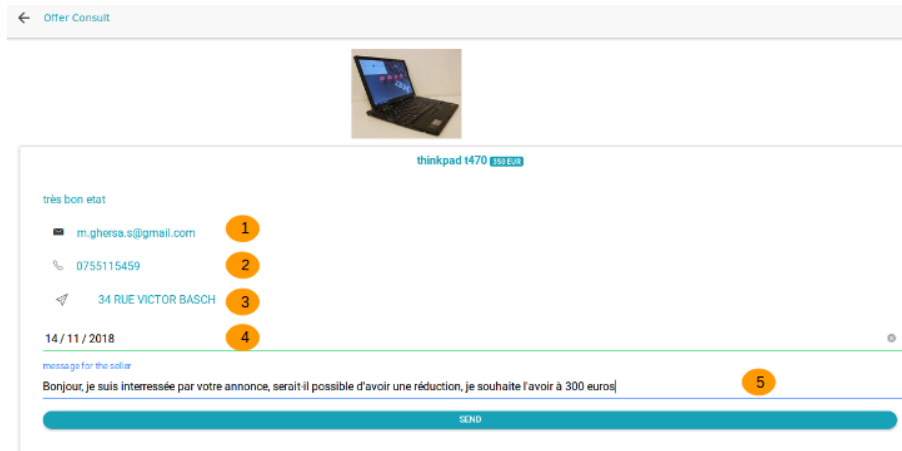


1. Barre de recherche par mot clé.
2. Modification de son profil en cliquant sur la photo de profil
3. Ajouter un produit.
4. Voir toutes ses offres postulées sur l'application en attente d'achat.
5. Voir l'ensemble de ses réservations déjà validées par le vendeur.
6. Voir ses achats et vente sur l'application.
7. Déconnexion.
8. Consulter les détails d'une annonce, on aura accès pour envoyer une demande de réservation aussi.

### 3.2 Gestion des annonces

D'autres opérations sont possibles pour la gestion des différentes offres dans notre application (Ajouter, Modifier, Supprimer une annonce).

**Consulter les détails d'un produit :** Permet de voir la description détaillée du produit ainsi que les informations complémentaires sur le vendeur, avec un simple clique sur l'image, à partir de cette page l'utilisateur peut demander la reservation de ce dernier.

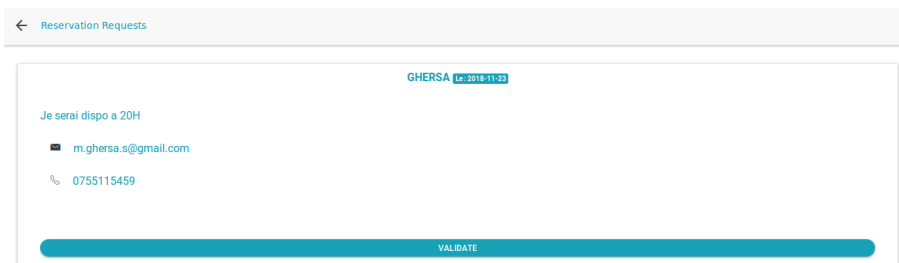


1. Redirige vers un nouveau mail pour le vendeur
2. Appelle le vendeur
3. Redirige vers google maps (adresse du vendeur)
4. Choisir une date de récupération du produit
5. Envoyer un message au vendeur

### 3.3 Gestion des achats

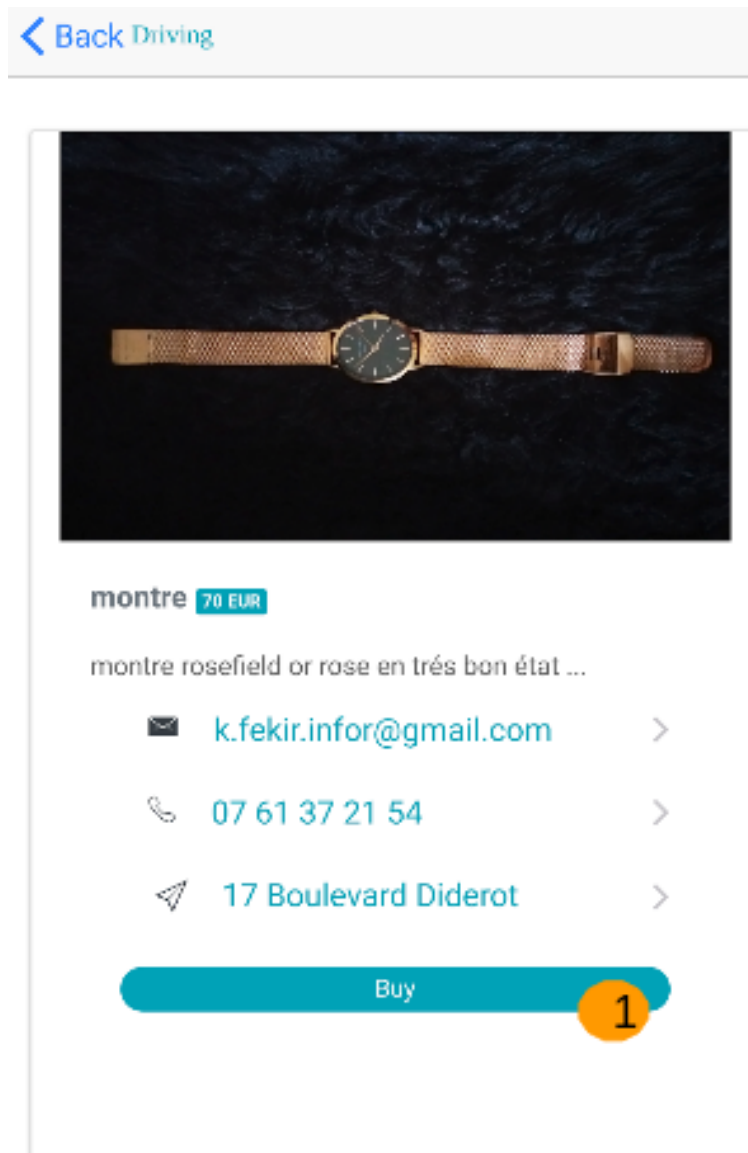
Dans cette section nous présentons les différentes opérations participant au processus de réservation.

**Valider la reservation d'un produit :** Le vendeur peut valider une des réservations du produit qui lui ont été proposées, un sms est envoyé aux utilisateurs qui ont demandé une réservation indiquant si leur demande est confirmée ou non dans le cas où ces dernier utilisent l'application mobile, la validation de l'achat d'un produit le retire de la liste des produits disponibles en vente.





**Finaliser l'achat :** L'utilisateur clique sur buy pour finaliser l'achat du produit en question.



Après cette étape le client peut accéder aux options suivantes :

- **Navigation :** En cliquant sur cette option l'utilisateur est redirigé vers google Maps.
- **Contact du vendeur :** L'application mobile nous permet de passer un appel téléphonique ou envoyer un mail au vendeur.
- **Valider l'achat :** Cette option passe le produit dans l'historique.

### 3.4 Consultation d'historique

L'historique peut référer les réservations (Driving), l'historique des produits achetés ou vendus (History) et enfin l'historique des produits en ligne (pas encore vendues, Follo my offers), qui sont accessibles à partir du menu déroulant.

## 4 Cas d'utilisation

Nous présenterons dans cette section quelques cas d'utilisation pour nos fonctionnalités.

### 4.1 Reserver un produit

<b>cas d'utilisation</b>	<b>Reserver produit " UC-1 "</b>
<b>Acteur</b>	Acheteur Toto
<b>Pré-condition</b>	Toto doit être connecté
<b>Description</b>	Toto consulte l'ensemble des dernières annonces, il clique sur l'image d'une table basse qui l'intéresse pour voir les détails, dans la page de description il choisit une date pour la récupération de cette dernière et envoie un message au vendeur.
<b>Post-condition</b>	Produit doit être ajouté dans l'historique des produits mis en vente du vendeur (dans la rubrique <b>follow my offers</b> ).

### 4.2 Valider la réservation d'un produit

<b>cas d'utilisation</b>	<b>valider la reservation d'un produit " UC-2 "</b>
<b>Acteur</b>	Vendeur Bob
<b>Pré-condition</b>	Bob doit être connecté
<b>Description</b>	Bob reçoit un message de la part de Toto, il consulte alors la rubrique <b>follow my offers</b> , il clique par la suite sur le bouton <b>see requests</b> dont il va voir l'ensemble des réservations faites sur la table basse, il valide finalement la réservation de Toto.
<b>Post-condition</b>	Produit doit être ajouté dans l'historique de ventes de Bob <b>History</b> , dans <b>Driving</b> de Toto et enfin retiré de liste des produits disponibles.

### 4.3 Cycle de vie d'une annonce

cas d'utilisation	Cycle de vie d'une annonce
Acteur	Acheteur Alice, Vendeur Bob
Pré-condition	Alice et Bob doivent être connectés
Description	Bob ajoute une annonce "iPhone 5" dans la page correspondante, il remplit tout les champs et clique sur <b>add</b> . Alice cherche un " iPhone 5 ", elle tape le mot clé " iPhone " sur la barre de recherche. Une liste d'annonces " iPhone " lui est affichée. Alice clique sur l'image de " iPhone 5 " ajouté par Bob. Voir " UC-1 " et " UC-2 " pour voir le processus de reservation et validation d'une annonce. Alice reçoit un sms validant sa réservation puis consulte la page <b>Driving</b> ou elle voit désormais que sa demande pour " iPhone 5 " est confirmée . A partir de cette page Alice peut naviguer pour voir sur Maps l'emplacement de récupération de l'iPhone ou contacter Bob. Pour enfin finaliser l'achat en validant l'opération.
Post-condition	Produit doit être ajouté dans l'historique de Alice et Bob "History" .

### 4.4 Scénario d'utilisation pour gestion de concurrence :

Alice demande de réserver un produit de Bob, au même moment Bob confirme la demande de réservation sur le même produit pour Sara. Lorsque la demande d'Alice arrive au serveur la validation de réservation par Sara est déjà enregistrée. Alice reçoit donc un message indiquant que le produit n'est plus disponible.

## 5 Technologies et architecture du projet

Nous présenterons dans cette section l'architecture utilisée pour la réalisation de notre projet.

Afin de permettre l'utilisation de l'application sur plusieurs supports nous avons fait le choix de développer la partie client en **Ionic 3** et un serveur en **RESTful**, on a modifié le modèle MVC d'une manière qu'il réponde à ce besoin, nous avons donc remplacé la couche *View* par un format spécifique *Json* qui seront traités par le client en fonction du besoin, ainsi les choix design seront réalisés au niveau de ce dernier, en d'autres termes au lieu de renvoyer des réponses au client sous forme de pages HTML, nous renvoyons des JSON contenant les informations nécessaires.

### 5.1 Serveur

La partie serveur a été implémentée en utilisant JEE, Java et PostgreSQL. le traitement d'une requête se fait comme suit :

(servlet appropriée), qui va aiguiller la requête vers les traitements correspondants (doPost, doGet ....) en fonction de la nature de traitement demandée par la requête, pour ensuite passer au package services qui va vérifier les données envoyées par le client, et enfin faire appel au Package DAO, qui représente la seule passerelle vers la base de données, ce choix a été fait afin de pouvoir séparer les différentes couches du serveur.

Le schéma ci-dessous illustre le fonctionnement du serveur :

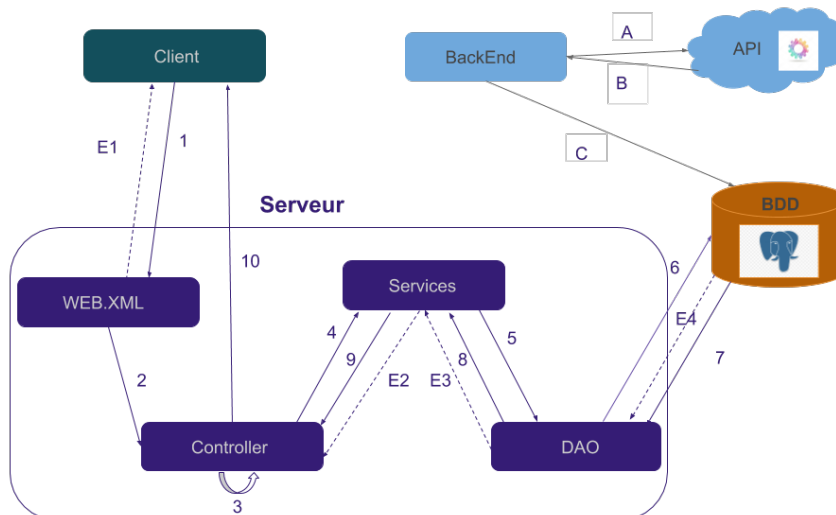


FIGURE 1 – Architecture partie serveur

1. Requête HTTP (peut contenir des données).
  2. Aiguiller la requête vers la bonne servlet appropriée.
  3. Distribue la requête HTTP à la méthode doXXX() correspondante.
  4. Passer les données vers la couche prenant en charge la vérification.
  5. Envoyer ces données à la seule passerelle vers la BDD (DAO).
  6. Faire les différents traitements sur la BDD (SELECT, UPDATE, ...).
- Le reste concerne des détails de réponses précisant si une erreur s'est passée ou dans le cas contraire succès.

A - On lance des appels aux API externes chaque jour à 00H.

B - Réponse de l'API.

C - Sauvegarder les données reçues dans notre base de donnée.

## 5.2 Client

Dans la partie client nous avons utilisé **Ionic 3** un framework basé initialement sur Angular, TypeScript et Apache Cordova, afin de permettre à notre application d'être multisupport iOS, Android, Chrome, Windows Phone... en utilisant HTML 5, CSS, TypeScript nous avons en outre intégré Bootstrap 4 au sein de Ionic pour permettre une plus grande facilité de design.

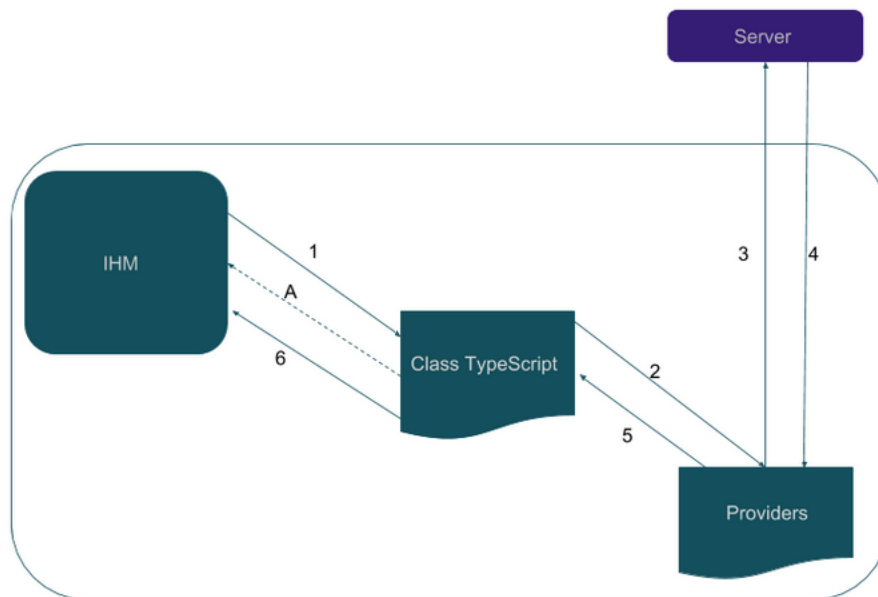


FIGURE 2 – Architecture partie client

1. Interaction avec la couche Angular (\*ngModel, \*ngIf, click, ...).
2. Appelle une méthode dans le provider pour communiquer avec le serveur.
3. Le provider est la seule couche qui communique avec le serveur.
4. Réponse de serveur en fonction de demande du client " Provider ".
5. Avec un simple " return " vers la class Angular "TS", qui traite la reponse reçu avec " subscribe ".
6. Mettre les donnée reçu sur l'écran de l'utilisateur.

A - On peut avoir une réponse directement dans notre class TS pas besoin de communiquer ou demander au serveur ( Données dans le cache, Traitement des erreurs, ...).

### 5.3 Base de données utilisée

Nous utilisons une base de données PostgreSQL dû au fait que le service de cloud computing heroku hébergeant notre application utilise de base cet SGBD, de plus c'est un SGBD relationnel parfaitement approprié à la DB de notre application.

### 5.4 API utilisées

Pour avoir plus de fonctionnalité on a amélioré notre application et on a ajouté une fonctionnalité qui permet à l'utilisateur de faire une comparaison de prix d'un produit qu'il lui intéresse.

[Price API https://app.priceapi.com/documentation](https://app.priceapi.com/documentation)

Receuille des informations de grandes entreprises comme (Amazon, Fnac, Darty, ...), nous l'utilisons dans notre application pour offrir au client un comparateur de prix.

Nous avons choisit d'implémenter l'API au niveau de serveur afin de gagner en consommation de temps car cette dernière offre un nombre de requête limité, d'autre part nous faisons une planification de tâche de telle sorte que chaque minuit on récupère tout les noms des produits existants dans notre Base de données, on interroge l'API et récupère les Json pour enfin stocker les informations générées dans la BDD.

← productComparing		
Shop Name	Price	URL
Pixmania.fr	201.23	<a href="#">go to store</a>
Darty	224.0	<a href="#">go to store</a>
Rakuten - Top_Occasion	145.0	<a href="#">go to store</a>
eGlobal Central France	172.99	<a href="#">go to store</a>
Tecobuy France	172.99	<a href="#">go to store</a>
eglobalcentral.fr	172.99	<a href="#">go to store</a>
Cdiscount Marketplace	227.48	<a href="#">go to store</a>
Rakuten - bestdigit	167.69	<a href="#">go to store</a>
Rakuten - Central-FR	177.99	<a href="#">go to store</a>
Bon Acheeter	172.99	<a href="#">go to store</a>

## 5.5 Sécurité

Afin d'améliorer la sécurité de notre application, nous nous sommes basé sur les points suivants :

- le mot de passe n'est jamais affiché en clair
- Demande de confirmation de mot de passe pour inscription et modification de profil.
- le mot de passe est codé avant l'envoi, décodé à la réception.
- le mot de passe est enregistré dans BDD sous forme d'un Hash.
- Le Token est codé avant l'envoi, décodé à la réception.
- Ajouter une durée de validation de Token.
- Après connexion à chaque nouvelle requête les tokens sont envoyés dans l'entête http, aucune requête n'est acceptée sans un Token valide.

## 6 Points forts et faibles

L'auto critique de l'application s'avère être une tâche difficile, dans cette section, nous présenterons les différents aspects que nous pensons faibles et forts de notre application :

### 6.1 Points forts

EcoBusiness possède les points forts suivants :

- Interface simple d'utilisation.
- Un seul modèle qui s'exécute sur différente plateforme .

### 6.2 Perspectives et points à améliorer

Nous avons pour le moment implémenter les grandes lignes de fonctionnalités, cependant quelques points peuvent être amélioré :

- Ajout de service de chat entre utilisateurs
- Ajout de service de notifications.
- Ajout d'option d'évaluation du vendeur.
- Ajout de filtrage de résultats de recherche.
- Ajout de filtrage d'historique suivant une période.
- Ajout de stockage interne.
- Ajout de possibilité de plusieurs images pour une annonce.