

# Spécifications techniques

**2tech**

## Panel de gestion de stock

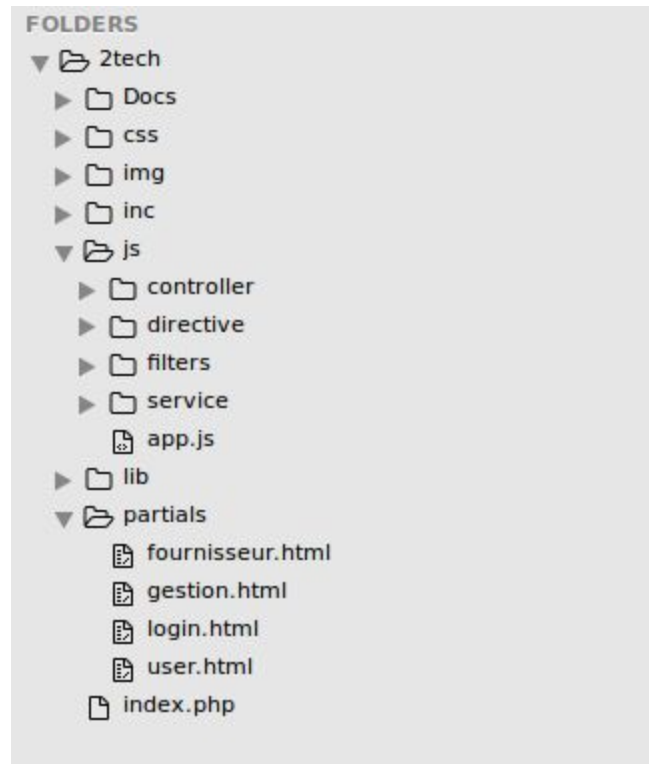
### 1. Contexte

L'application est développée avec Angular js qui est un framework MVC / MVVM côté client, développée en Javascript pour sa partie Front-End, PHP 7 qui est un langage coté serveur pour sa partie Back-End et enfin avec une base de donnée type MySQL.

---

## 2. Description générale

### 2.1 Arborescence



## 2.2. Rubriquage et workflow Tec 1.0 voir Req 1.0

Table des correspondances fonctionnels et techniques

Req	Fonctions	Tec
1.1	Ajouter un article	1.1
1.2	Sélectionner un article	1.2
1.3	Modifier un article	1.3
1.4	Supprimer un article	1.4
2.1	Ajouter un fournisseur	2.1
2.2	Sélectionner un fournisseur	2.2
2.3	Modifier un fournisseur	2.3
2.4	Supprimer un fournisseur	2.4
3.1	Ajouter un utilisateur	3.1
3.2	Modifier un utilisateur	3.2
3.3	Supprimer un utilisateur	3.3
3.4	Se connecter	3.4
3.4	Se reconnecter	3.5
3.5	Se déconnecter	3.6

Les sections au format HTML sont appelées dans une directive définie par le paramètre "templateUrl" . L'intérêt de cette technique est que le fichier HTML va être mis en cache par le navigateur.

### 3. Fonctions de service

#### 3.1 Ajouter un article Tec 1.1 voir Req 1.1

La fonctionnalité d'ajout s'effectue via un CRUD implémenté pour le projet, par la fonction **add\_article()**

qui est déclenché par l'appel ajax suivant

**"\$http.post()"**

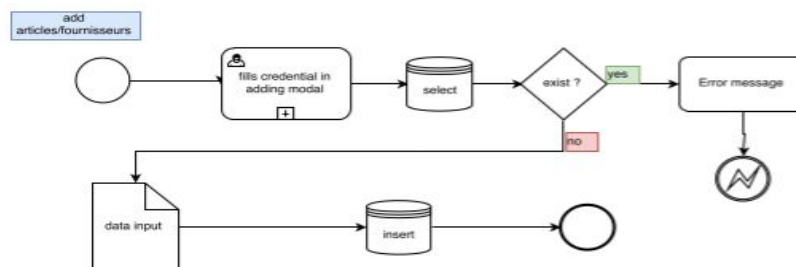
Une fois la fonction déclenchée, la commande SQL suivante est alors exécutée :

**"INSERT INTO article (articles, description, reference, fournisseurs, date, stock) VALUES ('.\$articles.', ".\$description.", ".\$reference.", ".\$fournisseurs.", ".\$date.", ".\$stock." )"**

Le contenu est alors ajouté dans la base de données en recevant un ID généré automatiquement, qui est de type *"integer"*.

Les champs contenus dans la table **article** peuvent être remplis dans la base de données tel que :

Champs	Type	Usage
"Articles"	varchar(255)	Le champ définit le nom de l'article et est obligatoire
"Description"	varchar(255)	Permet d'entrer une description de l'article
"Référence"	varchar(255)	La référence du produit doit être saisie pour être ajoutée à la base de données
"Fournisseurs"	varchar(255)	Permet d'entrer le nom du fournisseur pour indiquer la provenance de l'article
"Date"	datetime	Permet de spécifier la date d'entrée via un datepicker (calendrier jquery)
"Stock"	varchar(255)	Permet de spécifier une quantité à l'article en entrée, ce champs doit obligatoirement être saisi



### 3.2 Sélectionner un article Tec 1.2 voir Req 1.2

Lorsque l'utilisateur effectue une recherche via le champs prévu à cet effet,

`<input type="radio">`

l'appel ajax suivant:

`"$http.get()"`

est alors effectué pour appeler la fonction

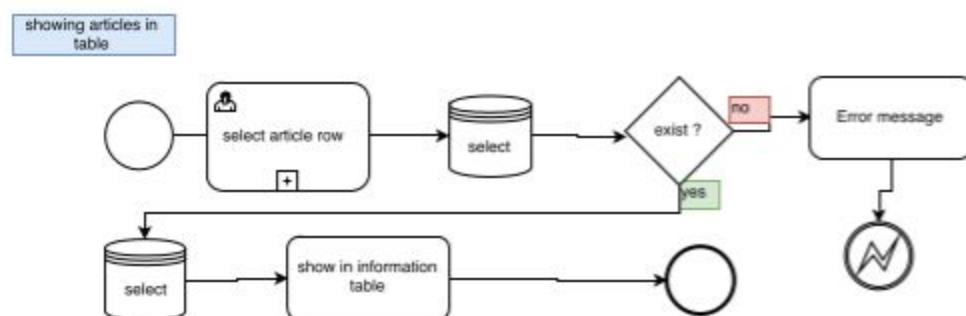
`"get_articles()"`

qui exécutera la commandes sql suivante :

**'SELECT \* FROM article'**

qui sélectionne les champs correspondants dans la base de données :

Champs	Type	Usage
"Articles"	varchar(255)	Le champ définit le nom de l'article et est obligatoire
"Description"	varchar(255)	Permet d'entrer une description de l'article
"Référence"	varchar(255)	La référence du produit doit être saisie pour être ajoutée à la base de données
"Fournisseurs"	varchar(255)	Permet d'entrer le nom du fournisseur pour indiquer la provenance de l'article
"Date"	datetime	Permet de spécifier la date d'entrée via un datepicker (calendrier jquery)
"Stock"	varchar(255)	Permet de spécifier une quantité à l'article en entrée, ce champs doit obligatoirement être saisi



### 3.3 Modifier un article *tec 1.3 voir req 1.3*

L'utilisateur aura la possibilité de modifier un article dans la base de données en cliquant sur l'icône via le tableau,  
l'appel ajax

`"$http.post()"`

est alors effectué pour appeler la fonction

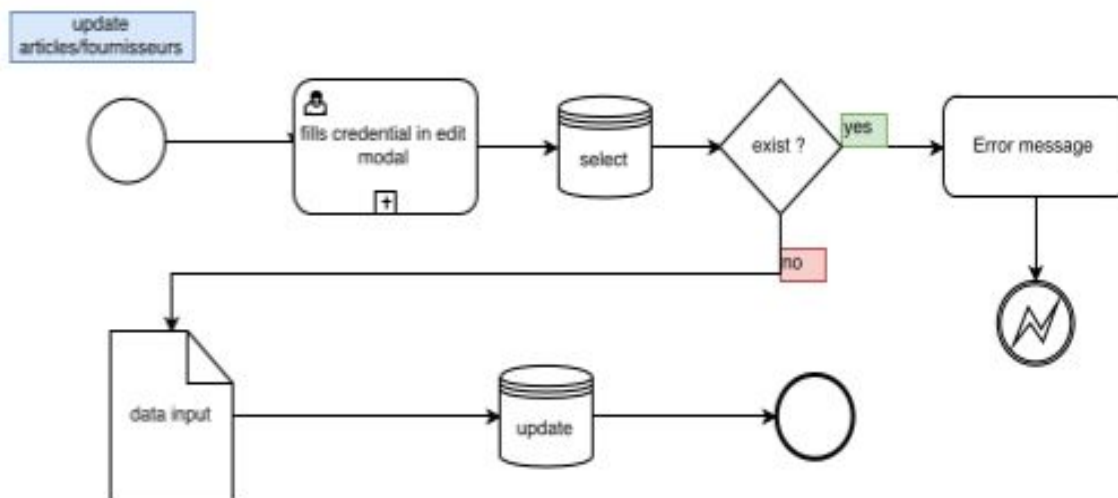
`"update_article()"`

`'SELECT * FROM article WHERE id = :id'`

qui sélectionne les champs correspondants dans la base de données.  
puis la fonction

`'UPDATE article SET articles =:articles , description= :description, reference = :reference, fournisseurs =:fournisseurs, date = :date, stock = :stock WHERE id = :id'`

qui modifie les champs correspondants dans la base de données.



### 3.4 Supprimer un article Tec 1.4 voir Req 1.4

L'utilisateur aura la possibilité de supprimer un article dans la base de données en cliquant sur l'icône via le tableau.

L'appel ajax suivant

`"$http.get()"`

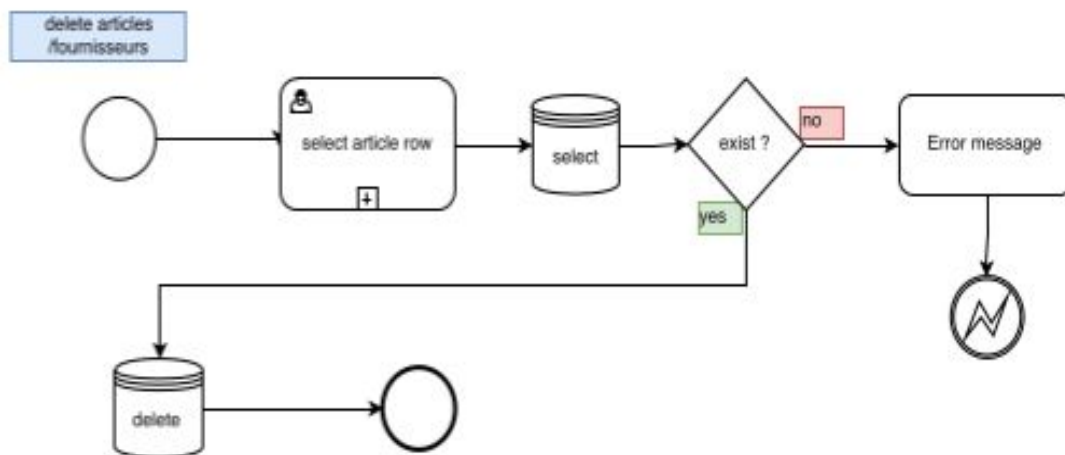
déclenchera la fonction suivante:

`"delete_article()"`

La commande SQL suivante sera alors exécutée:

`"DELETE FROM article WHERE id =$id"`

Une modal de confirmation s'ouvrira.



### 3.5 Ajouter un fournisseur Tec 2.1 voir Req 2.1

La fonctionnalité d'ajout s'effectue via un CRUD, par la fonction

**add\_fournisseur()**

qui est déclenchée par l'appel ajax suivant :

**"\$http.post()"**

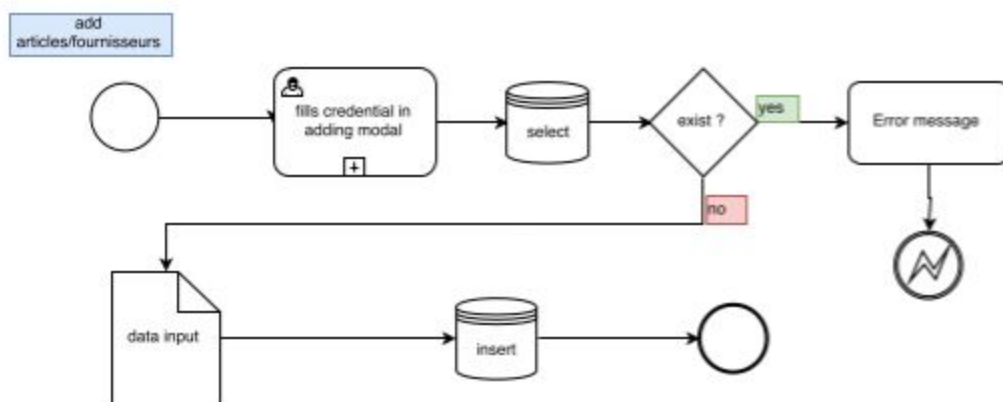
Une fois la fonction déclenchée, la commande SQL suivante est alors exécutée :

**"INSERT INTO fournisseurs (name, localisation, reference, date) VALUES ('".\$name."',  
"'.\$localisation."', "'.\$reference."', "'.\$date.'" )"**

Le contenu est alors ajouté dans la base de données en recevant un ID généré automatiquement, qui est de type *"integer"*.

Les champs contenus dans la table **fournisseurs** peuvent être remplis dans la base de données tel que :

Champs	Type	Usage
"Name"	varchar(255)	Le champ définit le nom du fournisseur et est obligatoire
"Localisation"	varchar(255)	Permet d'entrer une localisation du fournisseur
"Référence"	varchar(255)	La référence du fournisseur doit être saisie pour être ajoutée à la base de données
"Date"	Datetime	Permet de spécifier la date d'entrée via un datepicker (calendrier jquery)





### 3.6 Sélectionner un fournisseur Tec 2.2 voir Req 2.2

Lorsque l'utilisateur effectue une recherche via le champs prévu à cet effet.

`<input type="radio">`

L'appel ajax

`"$http.get()"`

est alors effectué pour appeler la fonction

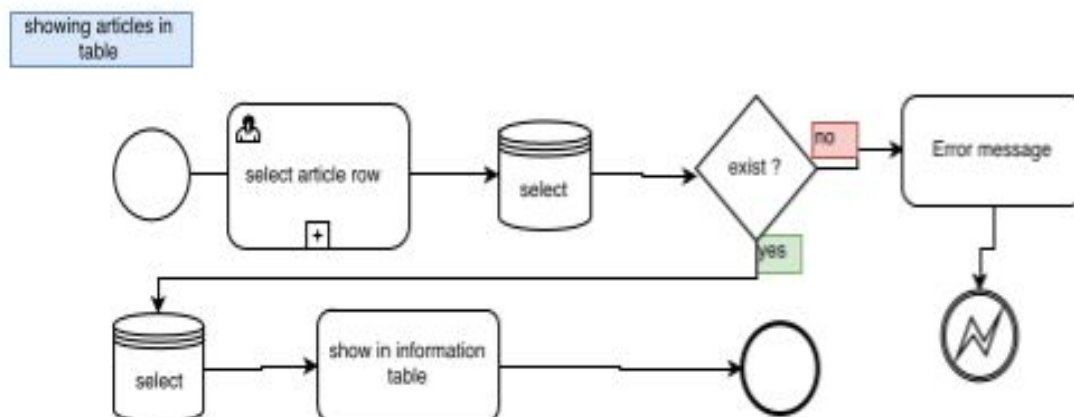
`"get_fournisseurs()"`

qui exécutera la commande sql suivante :

`'SELECT * FROM fournisseurs'`

qui sélectionne les champs correspondants dans la base de données :

Champs	Type	Usage
"Name"	varchar(255)	Le champ définit le nom du fournisseur et est obligatoire
"Localisation"	varchar(255)	Permet d'entrer une localisation du fournisseur
"Référence"	varchar(255)	La référence du fournisseur doit être saisie pour être ajoutée à la base de données
"Date"	Datetime	Permet de spécifier la date d'entrée via un datepicker (calendrier jquery)



### 3.7 Modifier un fournisseur Tec 2.3 voir req 2.3

L'utilisateur aura la possibilité de modifier un article dans la base de données en cliquant sur l'icône via le tableau,  
L'appel ajax suivant :

**"\$http.post()"**

est alors effectué pour appeler la fonction:

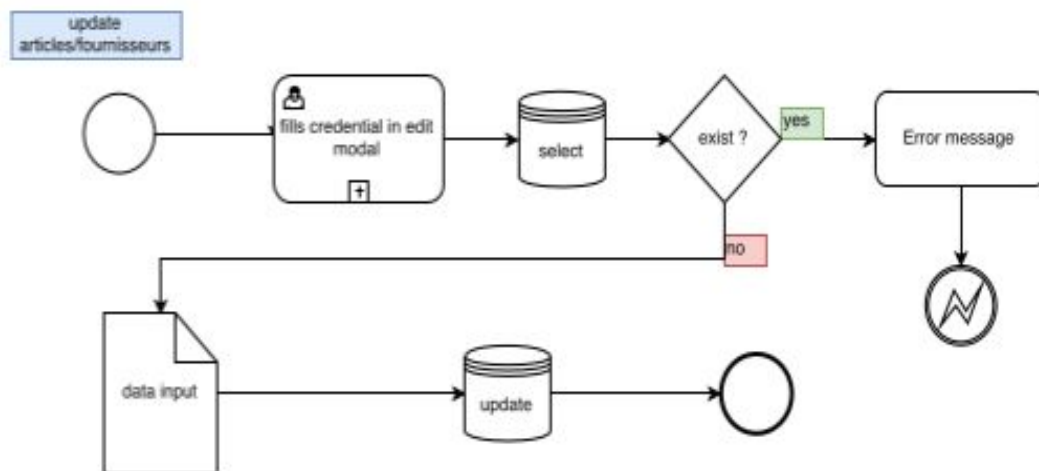
**"update\_fournisseur()"**

**'SELECT \* FROM fournisseurs WHERE id = :id'**

qui sélectionne les champs correspondants dans la base de données.  
puis la fonction:

**'UPDATE article SET articles = " ".\$articles." , description = " ".\$description." , reference = " ".\$reference." , fournisseurs = " ".\$fournisseurs." , date = " ".\$date." , stock = " ".\$stock." WHERE id = " ".\$id'**

qui modifie les champs correspondants dans la base de données.



### 3.8 Supprimer un fournisseur Tec 2.4 voir Req 2.4

L'utilisateur aura la possibilité de supprimer un article dans la base de données en cliquant sur l'icône via le tableau, l'appel ajax suivant

"\$http.get()"

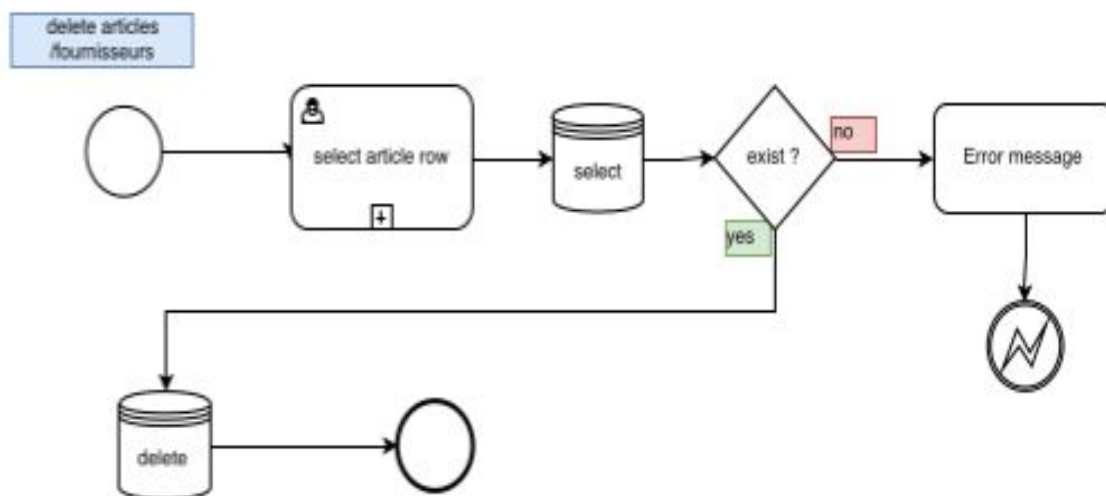
déclenche la fonction suivante:

"delete\_fournisseur()"

La commande SQL suivante sera alors exécutée:

"DELETE FROM fournisseurs WHERE id = ".\$id"

Une modal de confirmation s'ouvrira



### 3.9 Ajouter un utilisateur Tec 3.1 voir Req 3.1

La fonctionnalité d'ajout s'effectue via un CRUD, par la fonction

**"add\_user()"**

qui est déclenchée par l'appel ajax suivant

**"\$http.post()"**

Une fois la fonction déclenchée, la commande SQL suivante est alors exécutée :

**"INSERT INTO users (username, password) VALUES ('".\$username."', '".\$password."')"**

Le contenu est alors ajouté dans la base de données en recevant un ID généré automatiquement, qui est de type *"integer"*.

Les champs contenus dans la table **users** peuvent être remplis dans la base de données tel que :

Champs	Type	Usage
"Username"	varchar(255)	Le champ définit le nom de l'utilisateur et est obligatoire
"Password"	varchar(255)	Permet d'entrer un mot de passe
"Admin"	tinyint(1)	La référence du fournisseur doit être saisie pour être ajoutée à la base de données
"Date"	datetime	Permet de spécifier la date d'entrée via un datepicker (calendrier jquery)
"Token"	datetime	Suite de caractères générée automatiquement et aléatoirement pour les sessions de connexions

## **4.2 Modifier un utilisateur Tec 3.2 voir Req 3.2**

**L'utilisateur** aura la possibilité de modifier un article dans la base de données en cliquant sur l'icône via le tableau,  
L'appel ajax

**"\$http.post()"**

est alors effectué pour appeler la fonction

**"update\_users()"**

**'SELECT \* FROM users WHERE id = :id'**

qui sélectionne les champs correspondants dans la base de données,  
puis la fonction

**'UPDATE users SET username = ".\$username." , password = ".\$password." WHERE  
id=".\$id'**

qui modifie les champs correspondants dans la base de données.

## **4.3 Supprimer un utilisateur Tec 3.3 voir Req 3.3**

**L'utilisateur** aura la possibilité de supprimer un article dans la base de données en cliquant sur l'icône via le tableau.  
L'appel ajax suivant

**"\$http.get()"**

déclenche la fonction suivante:

**"delete\_user()"**

La commande SQL suivante sera alors exécutée:

**"DELETE FROM users WHERE id = ".\$id"**

Une modal de confirmation s'ouvrira

#### 4.4 Se connecter Tec 3.4 voir Req 3.4

L'utilisateur saisit les champs pour la connexion  
l'appel ajax suivant

`"$http.get()"`

déclenche la fonction suivante:

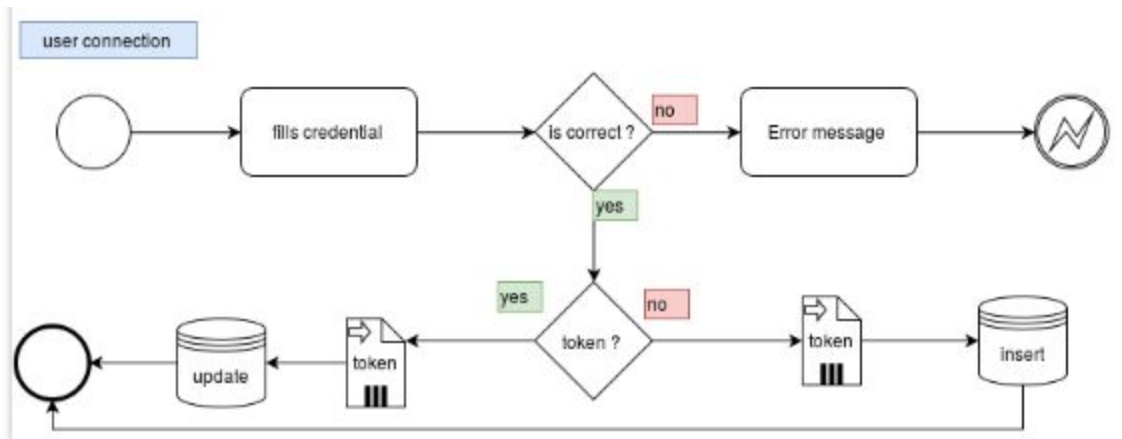
`"login_user()"`

La commande SQL suivante sera alors exécutée:

**"SELECT username FROM users WHERE username='\$username' AND  
password='\$password'"**

puis pour modifier l'identifiant du token de connexion :

**"UPDATE users SET token=:token WHERE username=:username AND  
password=:password"**



#### 4.5 Reconnexion Tec 3.5 voir Req 3.4

L'utilisateur pourra fermer la fenêtre sans se déconnecter puis la réouvrir et retrouver sa connection, si l'utilisateur ne s'est pas déconnecté alors :

l'appel ajax suivant

`"$http.get()"`

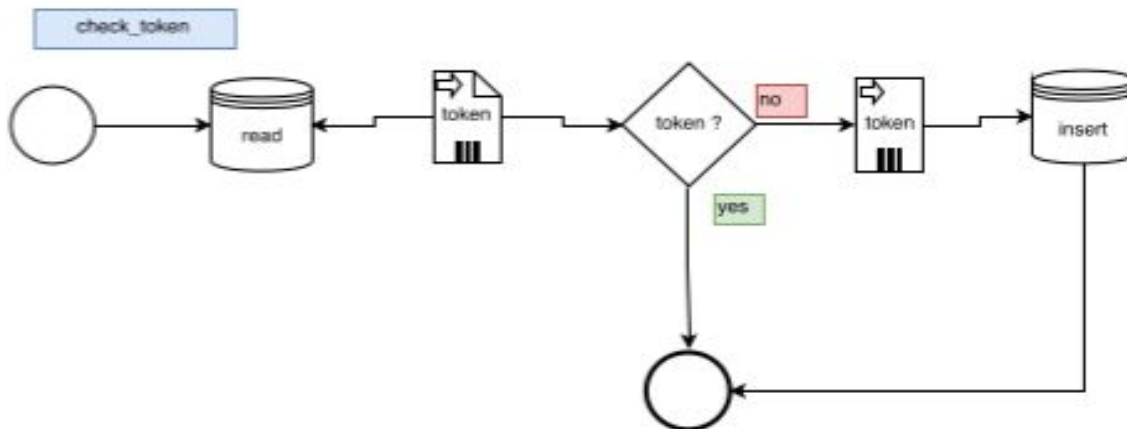
déclenche la fonction suivante:

`"check_token()"`

La commande SQL suivante sera alors exécutée:

`"SELECT * FROM users WHERE token='$token'"`

et vérifiera l'identifiant de connexion dans la base de données



## **4.6 Déconnexion Tec 3.6 voir Req 3.5**

L'utilisateur pourra se déconnecter en cliquant sur le bouton prévu à cet effet:  
l'appel ajax suivant

**"\$http.get()"**

déclenche la fonction suivante:

**"logout\_user()"**

La commande SQL suivante sera alors exécutée:

**"UPDATE users SET token='LOGGED OUT' WHERE token=\$token"**

Pour réinitialiser la valeur du token à "logged out"

---

# **4. Contraintes**

## **4.1 Techniques**

Les technologies utilisées pour le développement sont :

- Linux
- PHP 7
- MySQL
- Angular JS
- jQuery

## **4.2 Temporelles**

Le temps imposé est de 1 mois. L'estimation de la charge de travail par l'équipe de développement est de 7 semaines.

## **4.3 Graphiques**

Pas de charte requise. L'ensemble de l'application sera simple, fonctionnel, facile d'accès avec une interface utilisateur rapide et efficace.

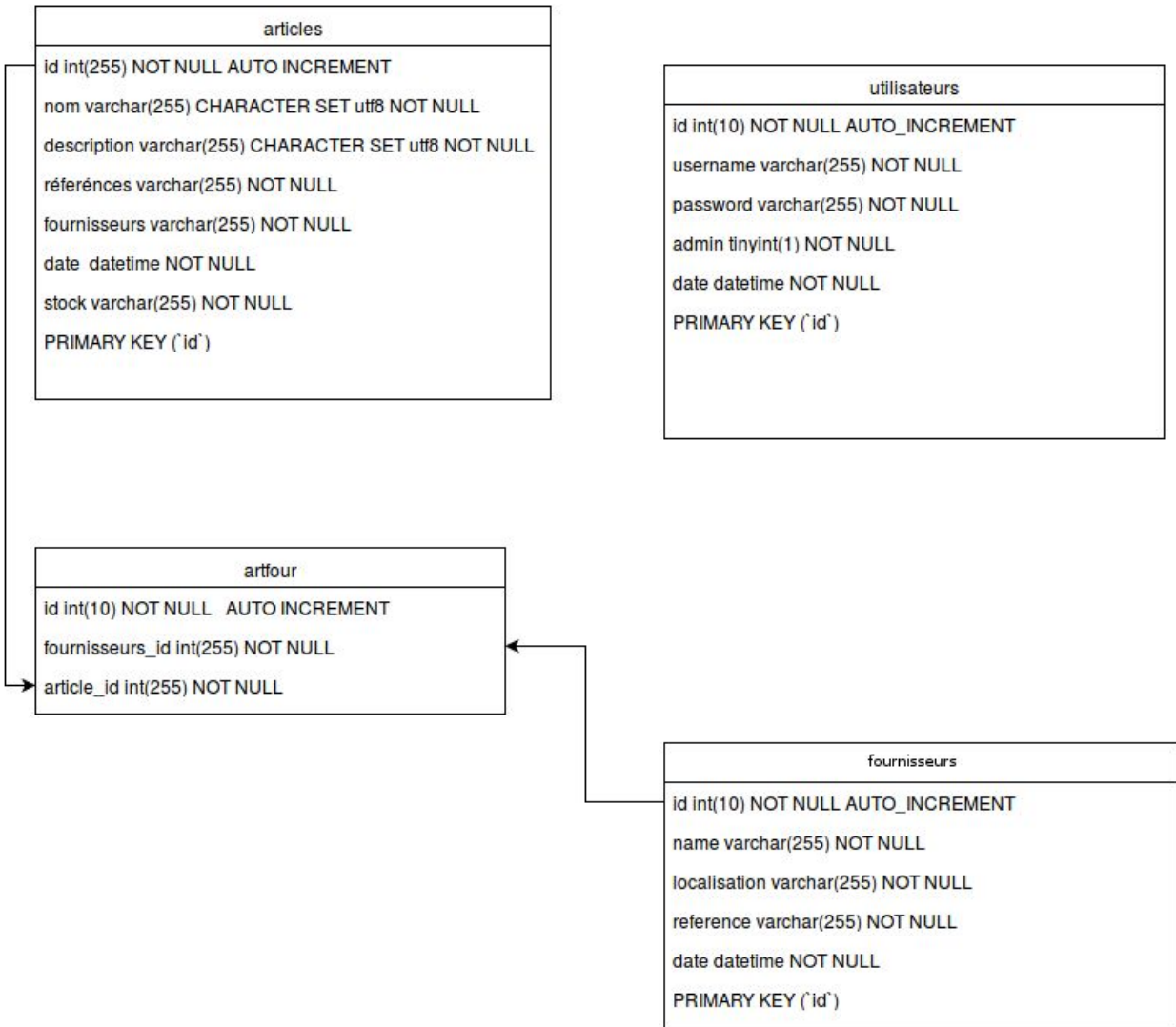
---



## 5. Description des données

Ceci est la représentation de la base de données.

### 5.1 Unified Modeling Language (UML)



## 5.2 Business Process Modeling Language (BPML)

