

# Documentation Technique : Refile tes Gogosses

par: Alves L. & Rodrigues dos Santos F.



# Table des matières

---

## Table des matières

### 1. Introduction

### 2. Plan de Travail

### 3. Étude d'opportunité

#### 3.1 Le projet

#### 3.2 Analyse des autres solutions disponibles

##### 3.2.1 WordPress

##### 3.2.2 Shopify

#### 3.3 Pourquoi notre solution ?

### 4. Analyse Fonctionnelle

#### 4.1 Pages : Fonctionnement

##### 4.1.1 Index

##### 4.1.2 Inscription

##### 4.1.3 Connexion

##### 4.1.4 Créer nouvel article

##### 4.1.5 Article

###### 4.1.5.1 Afficher article

###### 4.1.5.2 Modifier article

##### 4.1.6 Annonces

###### 4.1.6.1 Annonces

###### 4.1.6.2 Mes annonces

#### 4.2 Pages : Maquettes

### 5. Analyse Organique

#### 5.1 Choix de développement

#### 5.2 Organisation Générale

#### 5.3 Principales données

#### 5.4 Structure du projet

#### 5.5 Description des fichiers

##### 5.5.1 Index.php

##### 5.5.1 Signup.php

##### 5.5.2 Login.php

##### 5.5.3 NewArticle.php

##### 5.5.4 Article

###### 5.5.4.1 Article

###### 5.5.4.2 Modifier Article

##### 5.5.5 Annonces.php

##### 5.5.6 Nav.inc.php

##### 5.5.7 Page\_func.inc.php

##### 5.5.8 Sql.inc.php

##### 5.5.9 Login\_func.inc.php

##### 5.5.10 Crud\_article\_func.inc.php

#### 5.6 Organigrammes des principales fonctions

### 6. Tests

### 7. Conclusion

# 1. Introduction

---

Cette documentation qui suit détaille le site web en PHP que l'on a réalisé dans le cadre du module M306.

Ce projet est un site de vente d'articles où l'on peut poster des annonces ou ajouter à son panier plusieurs articles et les acheter.

Dans cette documentation se trouve le plan que l'on a suivi afin de réaliser ce projet, une étude d'opportunité expliquant pourquoi notre solution de site web de ventes d'articles est un choix pertinent pour toute personnes intéressée, une analyse fonctionnelle et organique ainsi qu'un diagramme de fonctionnement général du site.

## 2. Plan de Travail

---

Afin de mener ce projet à bien nous avons jugé comme plus pertinent la méthode "Waterfall" pour réaliser ce projet qui consiste à suivre une multitudes d'étapes jusqu'à arriver à la mise en production du projet.

- Analyse
- Conception
- Réalisation
- Bilan

## 3. Étude d'opportunité

---

### 3.1 Le projet

---

Ces derniers temps, des centaines de millier de sites de vente d'articles en ligne font surface tous les jours. Cependant pas tout le monde est capable de mettre en place un site ergonomique, simple d'implémentation et sécurisé. Voici donc pourquoi nous avons décidé de réaliser ce site web composé actuellement:

1. Une page d'accueil permettant de rechercher rapidement et simplement un petit nombre d'articles.
2. Une page affichant une plus grande quantité d'articles avec des critères plus précis.
3. Une page d'inscription et de connexion
4. Une page de création d'annonce

### 3.2 Analyse des autres solutions disponibles

---

Comme évoqué dans le point qui précède, la demande de sites de ventes en ligne est de plus en plus grande, or, il existe de nombreuses solutions disponibles sur internet cependant la majorité sont soit trop chères, difficiles d'implémentation ou simplement étant de mauvaise qualité.

#### 3.2.1 WordPress

Par exemple, leader du marché, WordPress, propose sa solution de création de sites populaire auprès de nombreux commerçants et aimée par beaucoup d'entre-eux. Hélas, cette solution possède de nombreux désavantages.

1. Un coût élevé: le simple fait d'acheter leur solution revient à plus de 200.- et ensuite chaque mois de 10 à 50.- et c'est sans les frais venant avec un designer professionnel qui ferait grimper la note d'avantage.

2. Implémentation difficile: Afin d'utiliser la solution WordPress il sera nécessaire de créer soit-même la base de donnée et de mettre en place tout le système interagissant avec cette dernière ce qui pourrait amener à des problèmes de sécurité si elle est mal installée.
3. Solution beaucoup trop lourde et complexe: Bien que la solution de WordPress soit en théorie très complète et capable de répondre à la plupart des demandes, nous estimons que c'est justement un de ses points faibles car très souvent les commerçants ne cherchent qu'un moyen simple de mettre en place leur commerce en ligne et non de se tracasser avec un système bien trop compliqué et nécessitant un développeur ou designer ou que la personne en question possède les compétences nécessaires.

### 3.2.2 Shopify

Ensuite, une autre solution en seconde position juste derrière WordPress, Shopify. Tout comme WordPress, Shopify offre une solution intéressante mais qui après analyse comporte quelques défauts majeurs et pouvant poser problème à beaucoup de commerçants.

1. Serveur non self-hosted: La solution de Shopify contrairement à celle de WordPress ou de la notre ne fonctionne que si l'on loue un de leur serveur et que l'on paie pour n'importe quelle de leur fonctionnalité ce qui va rendre la note très salée au fur et à mesure.
2. Aucun contrôle sur la base de donnée: En effet, la base de donnée de votre site sera entièrement gérée par Shopify qui ne vous laissera que l'option de gérer les données fournies et non de comment vous souhaitez structurer votre base de donnée directement vous même car il sera nécessaire de passer par Shopify ce qui pourrait nuire à la sécurité de votre commerce.

## 3.3 Pourquoi notre solution ?

---

Notre solution elle contrairement à celles citées précédemment est bien meilleure et répond à une plus grande demande.

1. Simplicité d'implémentation: Notre solution ne requiert qu'une chose, un serveur et ensuite il ne suffira que de déposer la solution dessus, de remplir la base de donnée avec vos articles et autre données et il sera prêt à être utilisé.
2. Compact: En effet, notre solution ne requiert pas tout un système de gestion de site comme WordPress ou Shopify car il n'est composé que de pages et composants fonctionnant tout simplement rien qu'en les incluant dans les pages les nécessitant
3. Droit au but: Notre solution n'est pas remboursée de centaines de fonctionnalités que beaucoup de commerçant n'utiliseront jamais mais est simplement là pour répondre à la demande de vente et d'achat d'articles, vos utilisateurs pourront simplement créer leurs annonces et les modifier ou supprimer comme bon leur semble.

## 4. Analyse Fonctionnelle

---

### 4.1 Pages : Fonctionnement

---

#### 4.1.1 Index

Cette page est la toute première vue par n'importe quel utilisateur. Sur cette page l'utilisateur peut faire des recherches simples en rentrant un mot afin de trouver l'annonce portant un nom égal ou semblable à la recherche.

Après la recherche effectuée 6 articles maximum s'affichent et l'utilisateur peut cliquer sur l'article désiré

### 4.1.2 Inscription

Cette page ne peut être accédée que lorsque l'utilisateur n'est pas connecté. L'utilisateur rentre **un nom d'utilisateur, un mail** conforme aux normes de mails ainsi qu'un mot de passe de plus de 6 caractères et qu'il soit égal à la **2ème entrée de ce mot de passe**.

Après ces étapes l'utilisateur aura fini la création de son compte et se verra connecté dans le cas où toutes les données entrées sont correctes et que l'email n'est pas déjà utilisé. sinon l'utilisateur verra une erreur apparaître lui demandant de corriger cette dernière.

### 4.1.3 Connexion

Cette page ne peut être accédée que lorsque l'utilisateur n'est pas connecté. L'utilisateur rentre son **email** et **son mot de passe**. Si premièrement l'email correspond à celui d'un compte existant va retourner le mot de passe hashé et le comparer au mot de passe rentré.

Dans le cas idéal l'utilisateur se verra connecté, dans le cas échéant l'utilisateur verra une erreur apparaître lui demandant de corriger cette dernière.

### 4.1.4 Créer nouvel article

Cette page ne peut être accédée que lorsque l'utilisateur est connecté. L'utilisateur doit entrer un nom d'article, une quantité, un prix, une description et une à plusieurs images montrant l'article en question.

Après tous ces champs remplis, l'article va être créé.

### 4.1.5 Article

Cette page peut être accédée en tout temps. Elle affiche les informations concernant un article en particulier.

#### 4.1.5.1 Afficher article

L'affichage de base de la page article. elle affiche toutes les informations de l'article en question ainsi qu'un bouton ajouter au panier.

#### 4.1.5.2 Modifier article

Si la personne la consultant est la personne ayant créée l'annonce, elle verra un bouton modifier apparaître qui après pression de ce dernier changera l'affichage de la page pour remplacer le texte par des champs dans lesquels l'utilisateur pourra modifier à son bon vouloir ces derniers ainsi que les images de l'article.

### 4.1.6 Annonces

#### 4.1.6.1 Annonces

Affiche à l'aide d'une recherche avancée basée sur plusieurs critères les annonces correspondant à cette recherche.

#### **4.1.6.2 Mes annonces**

Fonctionnement similaire à la page Annonces, cependant n'affiche uniquement les annonces de l'utilisateur actuellement connecté et n'est accessible que par un utilisateur connecté et uniquement lui-même.

## 4.2 Pages : Maquettes

Page d'Accueil		* visible après connexion	
Nom du site ou Logo	Accueil	Annonces	*Mes annonces
		*Panier   Se connecter	
Logo + phrase catchy			
Barre de recherche (simple) + bouton			
Photo article + desc. + prix			
Photo article + desc. + prix			
Photo article + desc. + prix			

Maquette de la page index

Nom du site ou Logo

Formulaire de création de compte

Having an account? Login

Maquette de la page d'inscription



Nom du site ou Logo
<div><div>Formulaire de connexion</div><div>Crée un compte</div></div>

\* visible après connexion

Nom du site ou Logo	Accueil	Annonces	*Mes annonces	*Panier   Se connecter
<div></div>				
<div><div>Formulaire pour nouvelles annonces</div></div>				

*Maquette de la page création d'article*

Nom du site ou Logo	Accueil	Annonces	*Mes annonces	*Panier   Se connecter
---------------------	---------	----------	---------------	------------------------

Titre de l'article

Description

Photo article

Prix + date de poste

Ajouter au panier

Footer

Maquette de la page article

Nom du site ou Logo	Accueil	Annonces	*Mes annonces	*Panier   Se connecter
---------------------	---------	----------	---------------	------------------------

Formulaire pour modifier une annonce

Maquette de la page article

Liste des articles dans le panier

Maquette de la page panier

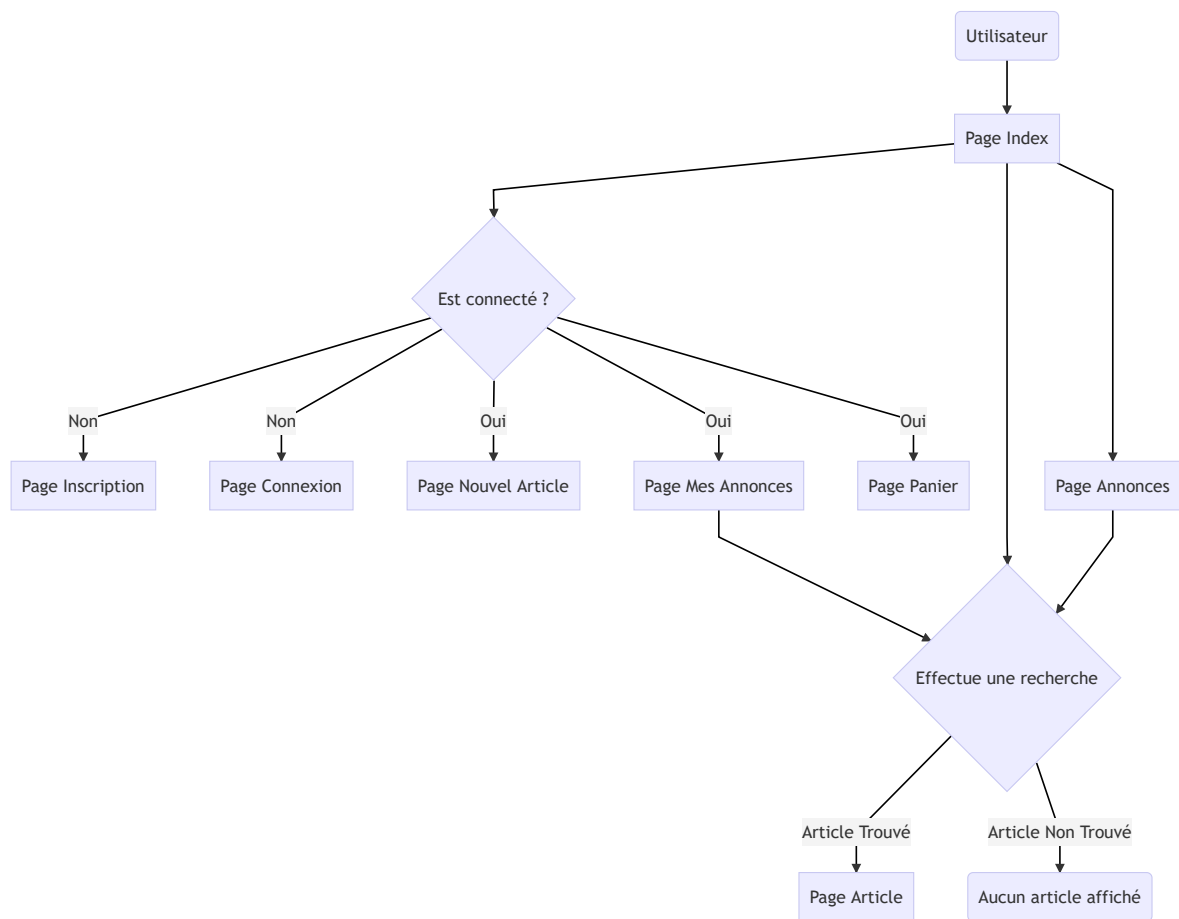
## 5. Analyse Organique

### 5.1 Choix de développement

Afin de réaliser ce projet de site de vente d'articles, le langage qui nous semblait le plus pertinent était le PHP ainsi que d'une base de donnée MYSQL. Nous aurions pu nous servir également de NodeJS ou de VueJS mais nous avons décidé de garder un certain niveau de sécurité au dépend du dynamisme. Notre site est principalement en PHP et HTML/CSS avec très peu voire pas du tout de Javascript afin de garder une sécurité globale élevée et d'éviter l'accès non-autorisé de

certaines pages. La gestion des pages et des différentes opérations sont également faites en PHP pour réduire la charge de travail pour l'utilisateur. Pour finir, tous ces choix ont été faits avec pour idée principale la facilité de modification du site par tout particuliers souhaitant ajouter ou retirer des fonctionnalités ou modifier l'affichage.

## 5.2 Organisation Générale



## 5.3 Principales données

- Utilisateur
  - ID : int
  - Email : string
  - Prénom : string
  - Connecté : bool
- Annonce
  - ID Article : int
  - Nom Article : string
  - Quantité Article : int
  - Description Article : string
  - Prix Article : int
  - Images : string|array
- Panier : Array<ID Article>

## 5.4 Structure du projet

```
RefileTesGogosses/  
├─ annonces.php || Affiche les annonces à l'aide d'une recherche avancée |  
Affiche les annonces de                                     l'utilisateur  
├─ article.php || Page d'affichage d'un article  
├─ img || Images utilisées sur le site  
|   ├─ background.png  
|   ├─ cart.svg  
|   ├─ download.svg  
|   ├─ icon.png  
|   └─ search.svg  
├─ index.php  
├─ js  
|   ├─ bootstrap.bundle.js  
|   ├─ bootstrap.bundle.js.map  
|   ├─ bootstrap.bundle.min.js  
|   ├─ bootstrap.bundle.min.js.map  
|   ├─ bootstrap.js  
|   ├─ bootstrap.js.map  
|   ├─ bootstrap.min.js  
|   ├─ bootstrap.min.js.map  
|   ├─ newArticle.js  
|   └─ register.js  
├─ jsconfig.json  
├─ LICENSE || License  
├─ login.php || Page de connexion  
├─ newArticle.php || Page de création d'article  
├─ panier.php || Page du Panier  
├─ php  
|   ├─ constantes.inc.php || Constantes de connexion à la BDD  
|   ├─ crud_article_func.inc.php || Fonctions de Création, Suppression, MàJ et  
récupération d'articles  
|   ├─ footer.inc.php || Footer du site à inclure aux pages  
|   ├─ login_func.inc.php || Fonctions de connexion  
|   ├─ nav.inc.php || Barre de navigation à inclure aux pages  
|   └─ page_func.inc.php || Fonctions de changement de page
```

```
| └─ sql.inc.php || Variable de connexion à la BDD
|─ signup.php || Page d'inscription
|─ sql
| └─ EmployMeDB1_3_0.sql || Base de Donnée
└─ tmp || Dossier contenant les logos des articles
```

## 5.5 Description des fichiers

### 5.5.1 Index.php

Lorsque l'utilisateur est sur cette page, seul un champ texte est présent ainsi que le layout de base du site.

Lorsque l'utilisateur rentre un article qu'il souhaite trouver. Le code suivant va s'exécuter :

```
$search = filter_input(INPUT_POST, 'search', FILTER_SANITIZE_STRING);
//Si une recherche a été effectuée, récupère les articles ressortant de la
recherche
if (isset($search)) {
    $articles = recherche($search);
}
```

Avec la variable **\$articles**, contenant toutes les annonces résultant de la recherche, elles vont être affichées grâce à la fonction `afficherArticlesRecherche()`, juste en dessous du champ texte :

```
function afficherArticlesRecherche($articles)
{
    echo "<div class=\"row justify-content-center pt-2\">";

    //Affiche les articles en rapport avec la recherche effectuée
    for ($i = 0; $i < count($articles); $i++)
    {
        if ($i == 3 && count($articles) > 3)
        {
            echo "</div>";
            echo "<div class=\"card-group col-11\">";
        }
        if ($i == 0)
            echo "<div class=\"card-group col-11\">";
        echo "
            <div class=\"card m-2\">
            <a href=\"./article.php?idA=" . $articles[$i]['id'] . "\">
            <img class=\"card-img-top\" src=\"tmp/" . $articles[$i]
['nomImage'] . "\" . $articles[$i]['typeImage'] . "\" alt=\"#\">
            </a>
            <div class=\"card-body\">
            <h5 class=\"card-title\">" . $articles[$i]['nom'] . "</h5>
            <p class=\"card-text\">" . $articles[$i]['description'] . "</p>
            <p class=\"card-text\"><small class=\"text-muted\">Annonce créée le
            ". $articles[$i]['dateCreation'] . "</small></p>
            </div>
            </div>";
        }
    }
    if (count($articles) > 3)
```



```

        echo " </div>";
    echo"</div>";
}

```

L'utilisateur pourra ensuite cliquer sur l'annonce voulue pour consulter l'article en question.

## 5.5.1 Signup.php

Lorsque l'utilisateur a rentré ses informations et appuyé sur s'inscrire, procède à l'inscription à l'aide du code suivant :

```

//Teste si le mail entré est conforme aux normes email, si oui, le mail sera
stocké dans $matches
preg_match("/^([\w\d._\-\#])+@([\w\d._\-\#]+[.][\w\d._\-\#]+)+$/",$uEmail,$matches);
//S'assure qu'un mail est entré, qu'il est conforme aux normes email, que le
mot de passe est plus long que 6 caractères et qu'il est égal au MDP de
vérification
if(strlen($uEmail) > 0 && $matches!= null && strlen($uPswd) > 6 && $uPswd ==
$uPswdVer){
    //Teste si l'email existe, si oui, Procède à la création du compte
    if(checkIfEmailExists($uEmail) == null)
    {
        //Hash le mot de passe
        $hashedPswd = password_hash($uPswd,PASSWORD_DEFAULT);
        //Si l'utilisateur est bien créé, affecte des informations à la
session et se débarrasse des informations pour questions de sécurité
        if(CreateNewUser($uName,$uEmail,$hashedPswd))
        {
            unset($hashedPswd);
            $_SESSION['user']['name'] = $uName;
            $_SESSION['user']['email']= $uEmail;
            $_SESSION['user']['id'] = getUserInfo($uEmail)['id'];
            $_SESSION['loggedIn'] = true;
            echo "Succès!";
            unset($userInfo);
            unset($uPswd);
        }
        //Sinon, affiche une erreur
        else
            echo "erreur lors de la création du compte";
    }
    //Sinon, affiche une erreur
    else
        echo "Ce mail existe déjà!";
}
else
    echo "Erreur?";

```

Renvoie ensuite l'utilisateur à l'index si l'inscription est réussie, sinon lui affiche des erreurs

## 5.5.2 Login.php

Lorsque l'utilisateur appuie sur se connecter après avoir fournis un email et un mot de passe, le code suivant va s'exécuter:

```
//S'assure que l'email entré existe
if(checkIfEmailExists($uEmail))
{
    //Récupère les infos de l'email en question dans la BDD
    $userInfo = getUserInfo($uEmail);
    //Vérifie le mot de passe fourni avec celui dans la BDD
    if(password_verify($uPswd,$userInfo['mdp']))
    {
        $_SESSION['loggedIn'] = true;
        $_SESSION['user']['name'] = $userInfo['nomUtilisateur'];
        $_SESSION['user']['id'] = $userInfo['id'];
        $_SESSION['user']['email'] = $userInfo['email'];
        unset($userInfo);
        unset($uPswd);
        header('location: ./index.php');
    }
    else
        echo "Mot de Passe invalide!";
}
else
    echo "Cet email n'existe pas!";
```

Si la connexion est réussie, renvoie l'utilisateur à l'index, sinon lui affiche des erreurs.

## 5.5.3 NewArticle.php

Après que l'utilisateur ait fournit les informations nécessaires à la création d'un article, le code suivant va s'exécuter premièrement pour récupérer le logo et le préparer à la création de l'annonce :

```
//S'assure qu'une image est bien fournie
if ($_FILES['imgSelect']['error'][0] == 0) {
    //Va parcourir l'ensemble des images fournies afin de les traiter
    for ($i = 0; $i < count($_FILES['imgSelect']['name']); $i++) {
        $Orgfilename = $_FILES["imgSelect"]["name"][$i];
        $filename = uniqid();
        $ext = explode("/", $_FILES["imgSelect"]["type"][$i])[1];
        $dir = "./tmp/";
        $file = $filename . '.' . $ext;
        //S'assure que le total d'images n'excède pas 4, sinon affiche une
        erreur
        if (count($_FILES['imgSelect']['name']) > 4) {
            $msg = "<div id=\"errorDiv\" class=\"alert alert-danger\"
            role=\"alert\">Attention vous avez sélectionné trop de fichiers!</div>";
            return;
        } else {
            //S'assure que le format de l'image est valide
            if (in_array($ext, ["png", "bmp", "jpg", "jpeg", "gif"])) {
                if ($titreArticle == "" || $quantiteArticle == "" ||
                $descriptionArticle == "" || $prixArticle == "") {
```

```

        $msg = "<div id=\"errorDiv\" class=\"alert alert-
danger\" role=\"alert\">Veuillez remplir tous les champs !</div>";
        return;
    } else {
        //Affiche un message d'erreur si rien n'a été push dans
l'array
        if (array_push($imgArray, [$filename, $ext]) <= 0)
            $msg = "Erreur ";
        }
    } else {
        $msg = "<div id=\"errorDiv\" class=\"alert alert-danger\"
role=\"alert\">Veuillez sélectionner uniquement des images !</div>";
        return;
    }
}
}
}
else
    $msg = "Veuillez sélectionner une image pour votre article";

```

Ensuite, lorsque le logo a passé l'étape de traitement, procède à l'étape de création de l'annonce :

```

//Ne s'exécute que si des images ont été fournies
if (!empty($imgArray)) {
    //Teste si un nouvel article a bel et bien été créé et va procéder à
l'upload de l'image sur le serveur
    if (CreateNewArticle($titreArticle, $quantiteArticle,
    $descriptionArticle, $prixArticle, $imgArray, $_SESSION['user']['id'])) {
        for ($i = 0; $i < count($_FILES['imgSelect']['name']); $i++) {
            //Effectue l'upload des images sur le serveur en les déplaçant
hors du cache du serveur
            if (move_uploaded_file($_FILES["imgSelect"]["tmp_name"][$i],
$dir . $file)) {
                $msg = "Upload was successful";
                unset($resultArticleCreation);
                unset($titreArticle);
                unset($quantiteArticle);
                unset($descriptionArticle);
                unset($prixArticle);
                unset($imgArray);
                unset($_POST);
                header('location: ./index.php');
            } else {
                $msg = "<div id=\"errorDiv\" class=\"alert alert-danger\"
role=\"alert\">Erreur lors de l'upload des fichiers !</div>";
            }
        }
    } else {
        echo $resultArticleCreation;
        return;
    }
    unset($imgArray);
} else
    $msg = "<div id=\"errorDiv\" class=\"alert alert-danger\"
role=\"alert\">Aucune image n'a été fournie !</div>";

```

Après cela, une annonce sera créée et l'utilisateur sera renvoyé à l'index

## 5.5.4 Article

### 5.5.4.1 Article

Cette page n'est accessible uniquement lorsqu'un id d'article est fourni en GET, ensuite le code suivant va permettre récupérer les informations de l'article en question avant d'afficher ses différentes informations

```
//Si on essaie d'accéder à la page sans avoir cliqué au préalable sur un article
ou entré un id d'article, affiche une page vide
if (!isset($_GET['idA'])) {
    die("veuillez sélectionner un article");
}
//Sinon, récupère les informations en rapport avec l'article
else {
    $infoArticle = ReadArticleById($_GET['idA']);
}
```

### 5.5.4.2 Modifier Article

Lorsque l'utilisateur passe en mode modification d'article, le code suivant s'exécutera s'il modifie le logo :

```
//Si des images ont été sélectionnées autre que les existantes
if ($_FILES['imgSelect']['error'][0] == 0) {
    //Parcourt les images fournies
    for ($i = 0; $i < count($_FILES['imgSelect']['name']); $i++) {
        $Orgfilename = $_FILES["imgSelect"]["name"][$i];
        $filename = uniqid();
        $ext = explode("/", $_FILES["imgSelect"]["type"][$i])[1];
        $dir = "./tmp/";
        $file = $filename . '.' . $ext;
        //Affiche une erreur si trop d'images ont été sélectionnées
        if (count($_FILES['imgSelect']['name']) > 4) {
            echo "<div id=\"errorDiv\" class=\"alert alert-danger\"
role=\"alert\">Attention vous avez sélectionné trop de fichiers!</div>";
            return;
        } else {
            //S'assure que l'image fournie est du bon format
            if (in_array($ext, ["png", "bmp", "jpg", "jpeg",
"gif"])) {
                //S'assure que tous les champs sont remplis
                if ($titreArticle == "" || $quantiteArticle == "" ||
$descriptionArticle == "" || $prixArticle == "") {
                    echo "<div id=\"errorDiv\" class=\"alert alert-
danger\" role=\"alert\">Veuillez remplir tous les champs !</div>";
                    return;
                } else {
                    array_push($imgArray, [$filename, $ext]);
                }
            } else {
                echo "<div id=\"errorDiv\" class=\"alert alert-
danger\" role=\"alert\">Veuillez sélectionner uniquement des images !</div>";
                return;
            }
        }
    }
}
```

```

    }
} else {
    echo "Pas d'images sélectionnées";
    foreach ($infoArticle as $img) {
        array_push($imgArray, [$img['nomImageArticle'],
$img['typeImageArticle']]);
    }
}
}

```

Ensuite, le code suivant va modifier l'article ainsi qu'upload le/les image/s :

```

//S'assure que la MàJ de l'article fonctionne
if (UpdateArticle($titreArticle, $quantiteArticle,
$descriptionArticle, $prixArticle, $imgArray, $_GET['idA'])) {
    if ($_FILES['imgSelect']['error'][0] == 0) {
        foreach ($imgArray as $img) {
            //Effectue l'upload des images sur le serveur en les
déplaçant hors du cache du serveur
            if (move_uploaded_file($_FILES["imgSelect"]["tmp_name"]
[$i], $dir . $img[0] . "." . $img[1])) {
                echo "Upload was successful";
                unset($resultArticleCreation);
                unset($titreArticle);
                unset($quantiteArticle);
                unset($descriptionArticle);
                unset($prixArticle);
                unset($imgArray);
                unset($_POST);
            } else {
                echo "Erreur lors de l'upload des fichiers";
            }
            $i++;
        }
    }
}
}

```

## 5.5.5 Annonces.php

Si un id utilisateur est fourni en GET et qu'il est égal à celui de l'utilisateur actuel, toutes les recherches seront faites par rapport aux annonces de l'utilisateur, sinon effectue des recherches globales.

## 5.5.6 Nav.inc.php

Fichier inclut au haut de chaque pages et contient la barre de navigation qui peut changer selon certains facteurs à l'aide du code suivant :

```

<?php
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <title>Barre de navigation</title>

```

```

<link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.min.css"
integrity="sha384-
TX8t27ECRE3e/ihu7zmQxVncDAY5uIKz4rEkgIXeMed4M0j1fIDPvg6uqKI2xXr2"
crossorigin="anonymous">
</head>

<body style="font-family: Arial, Helvetica, sans-serif;">
    <nav class="navbar navbar-expand-lg navbar navbar-light" style="background-
color: #EEEEEF;">
        
        <a class="navbar-brand" href="index.php">Refile tes Gogosses</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarText" aria-controls="navbarText" aria-expanded="false" aria-
label="Toggle navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarText">
            <div class="navbar-collapse collapse justify-content-center">
                <ul class="navbar-nav">
                    <li class="nav-item">
                        <a class="nav-link" href="index.php">Accueil<span
class="sr-only">(current)</span></a>
                    </li>
                    <?php
                        if ($_SESSION["loggedIn"] == true) {
                            echo "<li class=\"nav-item\"> <a class=\"nav-link\"
href=\"annonces.php\">Annonces</a> </li>";
                        }else{
                            echo "<li class=\"nav-item\"> <a class=\"nav-link\"
href=\"login.php\">Annonces</a> </li>";
                        }
                    <?>
                    <?php
                        if ($_SESSION["loggedIn"] == true) {
                            echo "<li class=\"nav-item\"> <a class=\"nav-link\"
href=\"newArticle.php\">Nouvelle annonce</a> </li>";
                        }
                    <?>
                </ul>
            </div>
            <ul class="navbar-nav">
                <?php
                    if ($_SESSION["loggedIn"] != true) {
                        echo "<li class=\"nav-item\"> <a class=\"nav-link\"
href=\"login.php\">Se connecter</a> </li>";
                    }else{
                        echo "<li class=\"nav-item\"> <a class=\"nav-link\">Se
deconnecter</a> </li>";
                    }
                <?>
                <?php
                    if ($_SESSION["loggedIn"] != true) {
                        echo "<li class=\"nav-item\"> <a class=\"nav-link\"
href=\"login.php\"><img src=\"./img/cart.svg\"></img></a> </li>";
                    }else{
                        echo "<li class=\"nav-item\"> <a class=\"nav-link\"
href=\"panier.php\"><img src=\"./img/cart.svg\"></img></a> </li>";

```

```

        }
    ?>
</ul>
</div>
</nav>
</body>

</html>

```

## 5.5.7 Page\_func.inc.php

Permet de facilement retourner les bonnes pages avec uniquement leur nom

```

function setLinks($path)
{
    if (strpos($path, 'index.php') !== false) {
        $arrayPath = ["index.php", "annonces.php", "annonce.php", "login.php",
            "./img/icon.png", "./img/cart.svg", "signup.php"];
    } else {
        $arrayPath = ["../index.php", "annonces.php", "annonce.php", "login.php",
            "./img/icon.png", "./img/cart.svg", "signup.php"];
    }
    return $arrayPath;
}

```

## 5.5.8 Sql.inc.php

Cette page permet d'effectuer la connexion à la BDD et d'utiliser des fonctions permettant de faire des requêtes et autre:

```

function db()
{
    static $dbc = null;

    // Première visite de la fonction
    if ($dbc == null) {
        // Essaie le code ci-dessous
        try {
            $dbc = new PDO('mysql:host=' . HOST . ';dbname=' . DBNAME, DBUSER, DBPWD,
                array(
                    PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8",
                    PDO::ATTR_PERSISTENT => true
                ));
        }
        // Si une exception est arrivée
        catch (Exception $e) {
            echo 'Erreur : ' . $e->getMessage() . '<br />';
            echo 'N° : ' . $e->getCode();
            // Quitte le script et meurt
            die('Could not connect to MySQL');
        }
    }
    // Pas d'erreur, retourne un connecteur
    return $dbc;
}

```

## 5.5.9 Login\_func.inc.php

Cette page contient les fonctions nécessaires au bon fonctionnement à la connexion des utilisateurs au site

La fonction suivante permet de tester si un email existe déjà dans la base et de le retourner

```
function checkIfEmailExists($email)
{
    static $ps = null;
    $sql = 'SELECT email FROM t_user WHERE email = :EMAIL';

    if ($ps == null) {
        $ps = db()->prepare($sql);
    }
    $answer = false;
    try {
        $ps->bindParam(':EMAIL', $email, PDO::PARAM_STR);

        if ($ps->execute())
            $answer = $ps->fetch(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        echo $e->getMessage();
    }

    return $answer;
}
```

La fonction suivante permet de récupérer les informations d'un utilisateur selon leur email

```
function getUserInfo($uEmail)
{
    static $ps = null;
    $sql = 'SELECT * FROM `t_user` WHERE email = :EMAIL';

    if ($ps == null) {
        $ps = db()->prepare($sql);
    }
    $answer = false;
    try {
        $ps->bindParam(':EMAIL', $uEmail, PDO::PARAM_INT);

        if ($ps->execute())
            $answer = $ps->fetch(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        echo $e->getMessage();
    }
    return $answer;
}
```

Pour finir, la fonction suivante sert à créer des utilisateurs

```
function CreateNewUser($uName, $uEmail, $uPswd)
{
    static $ps = null;
```



```

$sql = "INSERT INTO `t_user` (`nomUtilisateur`, `email`, `mdp`) ";
$sql .= "VALUES (:NAME, :EMAIL, :MDP)";
if ($ps == null) {
    $ps = db()->prepare($sql);
}
$answer = false;
try {
    $ps->bindParam(':NAME', $uName, PDO::PARAM_STR);
    $ps->bindParam(':EMAIL', $uEmail, PDO::PARAM_STR);
    $ps->bindParam(':MDP', $uPswd, PDO::PARAM_STR);

    $answer = $ps->execute();
} catch (PDOException $e) {
    echo $e->getMessage();
}
return $answer;
}

```

## 5.5.10 Crud\_article\_func.inc.php

La fonction suivante sert à récupérer les informations d'un article par rapport à son ID

```

function ReadArticleById($idArticle)
{
    static $ps = null;
    $sql = @"SELECT t_annonce.id as 'idArticle', nom as 'nomArticle'
    , prix as 'prixArticle',
    quantite as 'quantiteArticle',
    description as 'descriptionArticle',
    nomImage as 'nomImageArticle', typeImage as
    'typeImageArticle', idUser as 'idUser'
    FROM t_annonce JOIN t_image on (`idAnnonce` = t_annonce.id) JOIN t_user on
    (t_user.id = t_annonce.idUser) where t_annonce.id = :IDARTICLE";

    if ($ps == null) {
        $ps = db()->prepare($sql);
    }
    $answer = false;
    try {
        $ps->bindParam(':IDARTICLE', $idArticle, PDO::PARAM_INT);

        if ($ps->execute())
            $answer = $ps->fetchAll(PDO::FETCH_ASSOC);
    } catch (PDOException $e) {
        echo $e->getMessage();
    }

    return $answer;
}

```

La fonction suivante, elle, va effectuer une recherche dans la BDD par rapport aux noms d'articles et renvoyer un array contenant tous les articles ayant une correspondance à la recherche

```

function recherche($search)
{
    static $ps = null;

```

```

$search = "%".$search."%";
$sql = "SELECT * FROM t_annonce JOIN t_image on (`idAnnonce` = t_annonce.id)
WHERE nom LIKE :search LIMIT 6";

if ($ps == null) {
    $ps = db()->prepare($sql);
}
$answer=false;
try {
    $ps->bindParam(':search',$search , PDO::PARAM_STR);

    if($answer = $ps->execute())
    $answer = $ps->fetchAll(PDO::FETCH_ASSOC);

} catch (PDOException $e) {
    echo $e->getMessage();
}
return $answer;
}

```

La fonction suivante permet de créer des articles tout en insérant dans la base les images liées à ce dernier à l'aide d'une transaction PDO

```

function CreateNewArticle($titreArticle, $quantiteArticle, $descriptionArticle,
    $prixArticle, $arrayImages, $idUser)
{

    $sqlCreateArticle = "INSERT INTO
    `t_annonce`(`nom`,`description`,`prix`,`quantite`,`idUser`) VALUES (:NOM,
    :DESCRIPTION, :PRIX, :QUANTITE, :IDUSER)";
    $sqlAddImage = "INSERT INTO `t_image` (`nomImage`,`typeImage`,`idAnnonce`)
    VALUES (:NOMIMAGE, :TYPEIMAGE, :IDARTICLE)";

    try {
        static $req = null;
        if($req == null)
            $req = db()->prepare($sqlCreateArticle);

        db()->beginTransaction();
        $req->bindParam(':NOM',$titreArticle,PDO::PARAM_STR);
        $req->bindParam(':DESCRIPTION',$descriptionArticle,PDO::PARAM_STR);
        $req->bindParam(':PRIX',$prixArticle,PDO::PARAM_INT);
        $req->bindParam(':QUANTITE',$quantiteArticle,PDO::PARAM_INT);
        $req->bindParam(':IDUSER',$idUser,PDO::PARAM_INT);

        $req->execute();

        $idArticle = db()->lastInsertId();

        static $req2 = null;
        if($req2 == null)
            $req2 = db()->prepare($sqlAddImage);

        foreach($arrayImages as $img)
        {
            $req2->bindParam(':NOMIMAGE',$img[0],PDO::PARAM_STR);

```

```

$req2->bindParam(':TYPEIMAGE', $img[1], PDO::PARAM_STR);
$req2->bindParam(':IDARTICLE', $idArticle, PDO::PARAM_INT);
$req2->execute();
}

db()->commit();

return true;
} catch (PDOException $e) {
    db()->rollback();
    return $e;
}
}

```

La transaction suivante, comme celle pour la création d'articles, est une transaction PDO permettant de mettre à jour l'article ou l'article et ses images

```

function UpdateArticle($titreArticle, $quantiteArticle, $descriptionArticle,
    $prixArticle, $arrayImages, $idArticle)
{
    static $psUpdateAnnonce = null;
    static $psAddImageAnnonce = null;
    static $psDeleteImage = null;
    $sql = "UPDATE `t_annonce` SET ";
    $sql .= "`nom` = :NOM, ";
    $sql .= "`quantite` = :QUANTITE, ";
    $sql .= "`description` = :DESCRIPTION, ";
    $sql .= "`prix` = :PRIX ";
    $sql .= "WHERE (`id` = :ID)";
    $sqlDeleteImage = "DELETE * FROM t_image WHERE idAnnonce = :IDANNONCE";
    $sqlAddImage = "INSERT INTO `table` (`nomImage`, `typeImage`, `idAnnonce`)
VALUES (:NOMIMAGE, :TYPEIMAGE, :IDANNONCE)";

    if ($psUpdateAnnonce == null) {
        $psUpdateAnnonce = db()->prepare($sql);
    }
    if ($psAddImageAnnonce == null) {
        $psAddImageAnnonce = db()->prepare($sqlAddImage);
    }
    if ($psDeleteImage == null)
    {
        $psDeleteImage = db()->prepare($sqlDeleteImage);
    }

    try{
        db()->beginTransaction();
        $psUpdateAnnonce->bindParam(':NOM', $titreArticle, PDO::PARAM_STR);
        $psUpdateAnnonce->bindParam(':QUANTITE', $quantiteArticle,
PDO::PARAM_INT);
        $psUpdateAnnonce->bindParam(':DESCRIPTION', $descriptionArticle,
PDO::PARAM_STR);
        $psUpdateAnnonce->bindParam(':PRIX', $prixArticle, PDO::PARAM_INT);
        $psUpdateAnnonce->bindParam(':ID', $idArticle, PDO::PARAM_INT);
        $psUpdateAnnonce->execute();

        $psDeleteImage->bindParam(':IDANNONCE', $idArticle, PDO::PARAM_INT);
        $psDeleteImage->execute();
    }
}

```

```

        foreach($arrayImages as $img)
        {
            $psAddImageAnnonce->bindParam(':NOMIMAGE',$img[0],PDO::PARAM_STR);
            $psAddImageAnnonce->bindParam(':TYPEIMAGE',$img[1],PDO::PARAM_STR);
            $psAddImageAnnonce->bindParam(':IDANNONCE',$idArticle,PDO::PARAM_INT);
            $psAddImageAnnonce->execute();
        }

        db()->commit();

        return true;
    }
    catch (PDOException $e) {
        db()->rollBack();
        return $e;
    }
}

```

Pour finir, la fonction suivante est une fonction PHP permettant d'afficher tous les résultats de recherche dans un tableau de

```

function afficherArticlesRecherche($articles)
{
    echo "<div class=\"row justify-content-center pt-2\">";

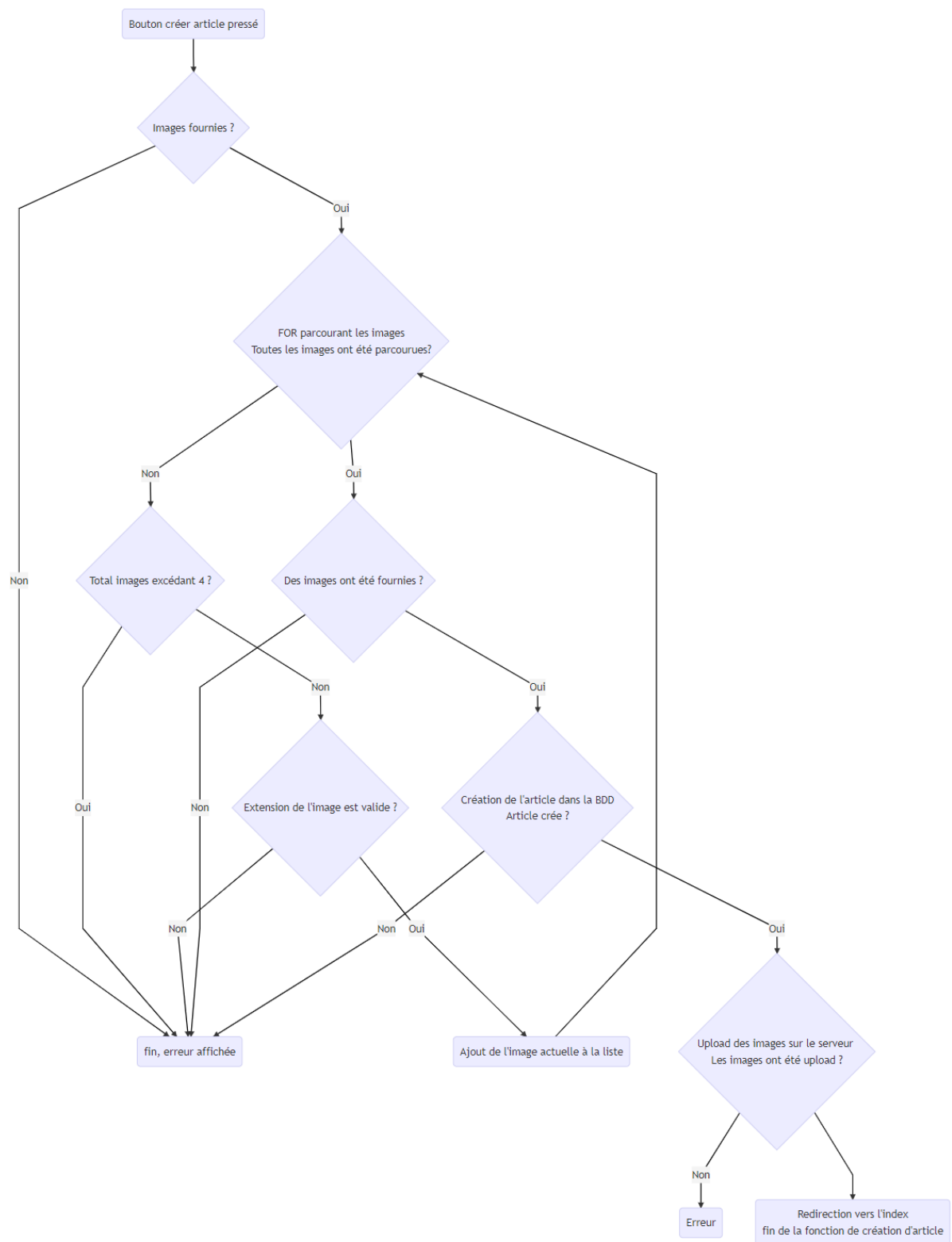
    //Affiche les articles en rapport avec la recherche effectuée
    for ($i = 0; $i < count($articles); $i++)
    {
        if ($i == 3 && count($articles) > 3)
        {
            echo "</div>";
            echo "<div class=\"card-group col-11\">";
        }
        if ($i == 0)
            echo "<div class=\"card-group col-11\">";
        echo "
            <div class=\"card m-2\">
            <a href=\"./article.php?idA=" . $articles[$i]['id'] . "\">
            <img class=\"card-img-top\" src=\"tmp/" . $articles[$i]
['nomImage'] . ". " . $articles[$i]['typeImage'] . "\" alt=\"#\">
            </a>
            <div class=\"card-body\">
            <h5 class=\"card-title\">" . $articles[$i]['nom'] . "</h5>
            <p class=\"card-text\">" . $articles[$i]['description'] . "</p>
            <p class=\"card-text\"><small class=\"text-muted\">Annonce créée le
            ". $articles[$i]['dateCreation'] . "</small></p>
            </div>
            </div>";
        }
        if (count($articles) > 3)
            echo " </div>";
    }
    echo "</div>";
}

```

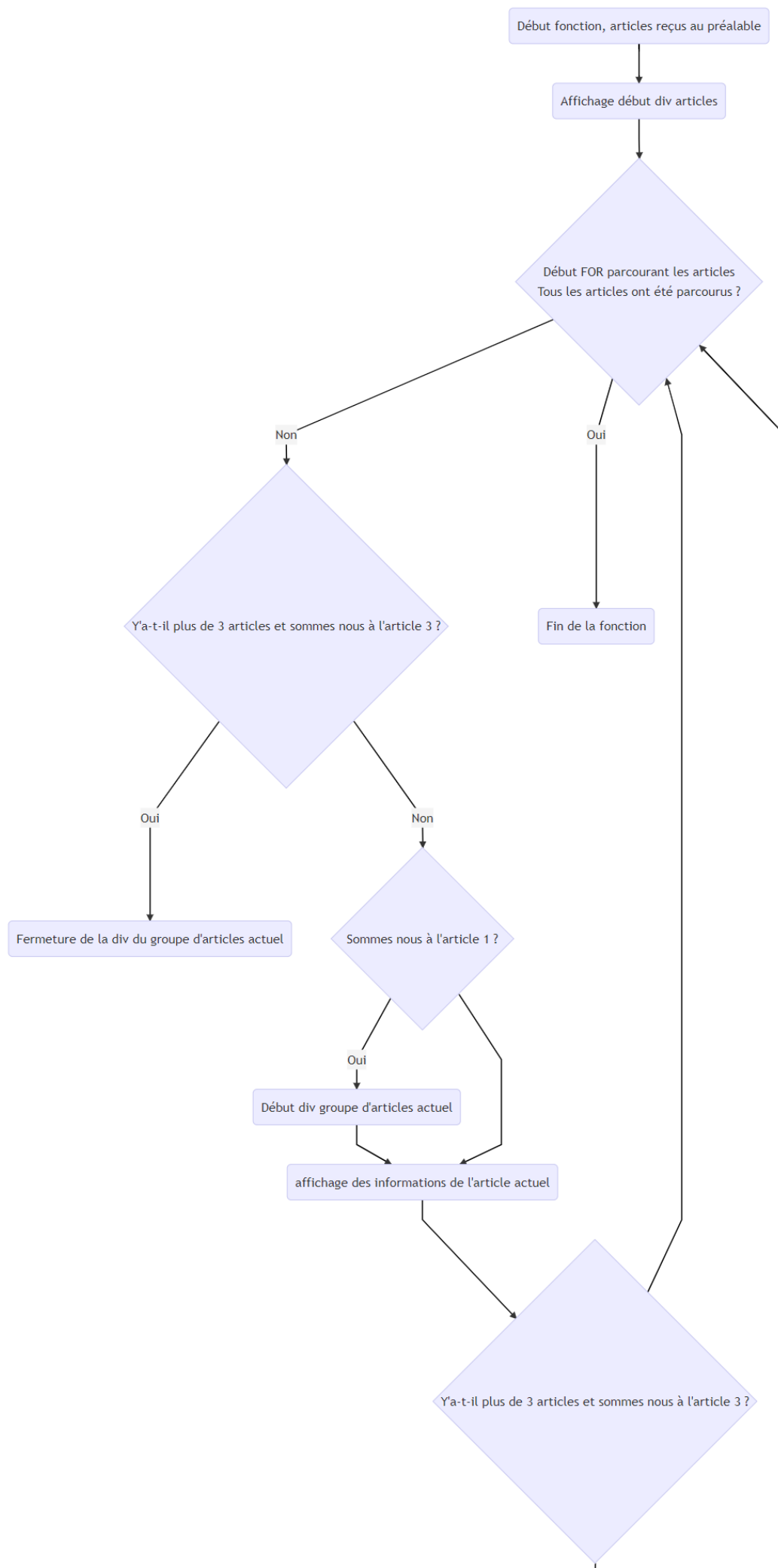


## 5.6 Organigrammes des principales fonctions

La fonction suivante est celle permettant de créer un nouvel article dans la BDD avec ses images.



La deuxième fonction que l'on juge comme importante est celle permettant d'afficher les articles d'une recherche



Oui

Fermeture de la div du groupe d'articles actuel

## 6. Tests

Pages	Description / Explication	Résultats
nav.inc.php	L'utilisateur non connecté a accès à la page Accueil depuis la barre de navigation.	✓ (23.04.21)
	L'utilisateur non connecté a accès à la page Se connecter depuis la barre de navigation.	✓ (23.04.21)
	L'utilisateur non connecté a la possibilité de créer un compte depuis la page Se connecter.	✓ (23.04.21)
	La page Panier depuis la barre de navigation redirige l'utilisateur non connecté à la page Login.	✓ (24.04.21)
	L'utilisateur non connecté a accès à la page Annonces depuis la barre de Annonces.	✓ (24.04.21)
	L'utilisateur connecté a accès à la page Accueil depuis la barre de navigation.	✓ (23.04.21)
	L'utilisateur connecté a accès à la page Annonces depuis la barre de navigation.	✓ (23.04.21)
	L'utilisateur connecté a accès à la Page Nouvelle annonce depuis la barre de navigation.	✓ (23.04.21)
	L'utilisateur connecté a accès à la Page Panier depuis la barre de navigation.	✓ (23.04.21)
	Lorsque l'utilisateur non connecté et l'utilisateur connecté clique sur le tire/logo renvoie sur la page Accueil.	✓ (23.04.21)
	Lorsque l'utilisateur est connecté, se connecter devient se déconnecter sur la barre de navigation.	✓ (24.04.21)
	Le lien Se déconnecter sur la barre permet de se déconnecter du site. Ne fait absolument rien.	✗ (24.04.21)
footer.inc.php	Lorsque l'utilisateur non connecté clique sur le bouton S'inscrire, le lien redirige sur la page Login.	✓ (24.04.21)
	Lorsque l'utilisateur connecté clique sur le bouton S'inscrire, une alerte affiche que l'utilisateur possède un compte. Redirige sur la page Accueil.	✗ (24.04.21)
	Lorsque l'utilisateur non connecté et l'utilisateur connecté clique sur RefileTesGogosses.ch, le lien redirige sur la page Accueil.	✓ (24.04.21)



login.php	Lorsque l'utilisateur clique sur "Pas de compte? Inscrivez-vous", le lien le redirige sur la page "S'inscrire".	✓ (24.04.21)
	Lorsque l'utilisateur clique sur le bouton Se connecter, les champs sont vérifiés.	✓ (24.04.21)
	Si les données dans les champs n'existent pas ou ne correspondent pas, un message est affiché.	✓ (24.04.21)
signup.php	Si les données dans les champs ne correspondent pas à ce qui est demandé ou sont vides, un message est affiché.	✓ (24.04.21)
	Lorsque l'utilisateur clique sur le bouton Envoyer, envoie le texte des champs dans la base de données	✓ (24.04.21)
	Le bouton Envoyer redirige à la page Accueil et connecte l'utilisateur.	✓ (24.04.21)
index.php	L'utilisateur non connecté et l'utilisateur connecté peuvent effectuer une recherche sur la page Accueil.	✓ (24.04.21)
	La recherche affiche le(s) article(s) recherchés sur la page Accueil.	✓ (24.04.21)
	Lorsque l'utilisateur non connecté clique sur un article, le lien le renvoie sur la page Article.  Lorsque l'utilisateur non connecté clique sur un article, la page Accueil est rafraîchi, le texte dans la barre de recherche et le(s) article(s) disparaissent.	✗ (24.04.21)
	L'utilisateur non connecté peut consulter la page Article.  L'utilisateur non connecté n'a pas accès à cette page.	✗ (24.04.21)
	Lorsque l'utilisateur connecté clique sur un article, le lien le renvoie sur la page Article.	✓ (24.04.21)
	L'utilisateur connecté peut consulter la page Article.	✓ (24.04.21)
article.php	Lorsque l'utilisateur non connecté est sur la page Article, le bouton Modifier l'article est désactivé.  L'utilisateur non connecté n'a pas accès à cette page.	✗ (24.04.21)
	Lorsque l'utilisateur non connecté est sur la page Article, le bouton Ajouter au panier redirige sur la page Login.  L'utilisateur non connecté n'a pas accès à cette page.	✗ (24.04.21)
	Lorsque l'utilisateur connecté est sur la page Article, le bouton Modifier l'article est désactivé tant que ce n'est pas son annonce.  Le bouton ne sera jamais activé car la vérification d'identification n'est jamais effectuée.	✗ (24.04.21)
	Lorsque l'utilisateur connecté est sur la page Article, le bouton Ajouter au panier ajoute l'article au panier.	✗ (24.04.21)

	Le bouton ne fait rien.	
	Lorsque l'utilisateur connecté clique sur le bouton Modifier, les champs deviennent accessibles et le bouton Modifier devient Enregistrer.  La modification n'a pas été implémentée.	✗ (24.04.21)
	Lorsque l'utilisateur connecté clique sur le bouton Enregistrer, le boutons Enregistrer devient Modifier et les données sont mises à jour dans la base de données.  La modification n'a pas été implémentée.	✗ (24.04.21)
annonces.php	L'utilisateur non connecté et connecté peut effectuer une recherche via le champ texte.	✓ (24.04.21)
	L'utilisateur non connecté et connecté peut sélectionner la catégorie pour sa recherche.  Les catégories sont entrées en dur et la table catégorie est vide et non utilisée.	✓ (24.04.21)
	L'utilisateur non connecté et connecté peut sélectionner le prix pour sa recherche.  Les valeurs minimum et maximum ne sont pas correctement définies.	✓ (24.04.21)
	Lorsque l'utilisateur clique sur le bouton Filtrer, la recherche affiche les articles.  Oui mais en prenant seulement en compte le champ texte.	✓/✗ (24.04.21)
	Si la recherche n'a aucun résultat, affiche un message.  N'affiche absolument rien.	✗ (24.04.21)
	Lorsque l'utilisateur non connecté clique sur un article, le lien le renvoie sur la page Article.  Lorsque l'utilisateur non connecté clique sur un article, la page Accueil est rafraîchi, le texte dans la barre de recherche et le(s) article(s) disparaissent.	✗ (24.04.21)
	Lorsque l'utilisateur connecté clique sur un article, le lien le renvoie sur la page Article.	✓ (24.04.21)
panier.php	Affiche le panier de l'utilisateur avec tous les articles avec leurs informations, la quantité, le prix total.  La page est vide. Seul un texte "Afficher le panier ici" est affiché.	✗ (24.04.21)
	L'utilisateur a la possibilité de modifier la quantité d'un article.  Fonction non implémentée.	✗ (24.04.21)

	L'utilisateur a la possibilité de supprimer un (des) article(s). <i>Fonction non implémentée.</i>	✗ (24.04.21)
	Le panier ne se vide pas si l'utilisateur se déconnecte ou change de page. <i>Fonction non implémentée.</i>	✗ (24.04.21)
	L'utilisateur a la possibilité de payer. <i>Fonction non implémentée.</i>	✗ (24.04.21)
<u>newArticle.php</u>	Tous les champs sont vérifiés.	✓ (24.04.21)
	Une image doit être fournie pour pouvoir créer un article sinon une erreur s'affiche.	✓ (24.04.21)
	La quantité ne peut qu'être un nombre sans virgule.	✓ (24.04.21)
	Le prix peut avoir n'importe quel nombre avec un 1 ou 2 chiffres après la virgule. <i>Seul 0.5 peut être mis après la virgule, 5.9 ou 14.2 n'est pas possible.</i>	✗ (24.04.21)
	On ne peut que fournir des images de type png, jpeg ou jpg.	✓ (24.04.21)
	On ne peut que fournir des images.	✓ (24.04.21)
	On ne peut fournir plus de 4 images. <i>La page devient vide et une seule image est insérée</i>	✗ (24.04.21)
	L'image fournie est bien uploadée sur le serveur.	✓ (24.04.21)
	L'image fournie et les données des champs sont bien insérées dans la base de données.	✓ (24.04.21)
	Le bouton Submit crée l'annonce grâce au donnée de la base de données.	✓ (24.04.21)

## 7. Conclusion

Maintenant que le temps nous étant alloué pour ce projet est écoulé nous avons pu réévaluer notre projet en comparaison à ce que nous avons prévu et la première chose que l'on a pu constater est que nous sommes assez proche de notre idée principale autant visuellement qu'au niveau fonctionnel. Cependant, le projet n'est pas abouti et nécessite encore l'implémentation de quelques fonctionnalités comme la recherche avancée, l'affichage des "Mes Articles" et plus important encore, la mise en place d'un panier dans lequel l'utilisateur peut ajouter des articles et ensuite procéder au paiement de ces derniers. Nous sommes tout de même plutôt satisfaits de notre travail effectué malgré le fait que nous n'avions pas pu aboutir au projet comme prévu.

En ce qui concerne les raisons pour lesquelles nous n'avons pas pu terminer à 100% le projet, la première est le fait que nous avons rencontré un problème au niveau d'une fonctionnalité et que nous ne l'avons remarqué que bien plus tard. Or, quand nous l'avons repérée au lieu de faire les modifications sur la branche actuelle afin de ne pas gêner le développement, nous avons fait l'erreur de modifier la branche précédemment fusionnée avec le main ce qui a entraîné des problèmes de compatibilité et de code manquant car la version n'était pas la plus récente. Ce qui en a résulté de tout cela est la perte d'une journée entière, cependant à l'avenir nous empêcheront une telle situation de se reproduire en créant une nouvelle branche pour corriger cette anomalie et ensuite baser la branche sur laquelle on travaillait sur cette dernière et pour les suivantes que de revenir sur une branche déjà fusionnée avec le main et considérée comme "terminée". Ensuite, nous avons rencontrés des problèmes dans le développement de la fonctionnalité update qui après une journée entière n'était toujours pas fonctionnelle, ce qui a résulté avec l'abandon de cette dernière au profit de programmer le plus possible l'ensemble du site. Une des principales causes de ce blocage a été le manque crucial de commentaires à ce

moment et la séparation plus claire de chaque tâches qui a donc mené à un problème qui aurait pu être plus facilement réglé si l'on avait appliqué ces bonnes pratiques et surtout si l'on avait programmé par "morceaux" cette fonctionnalité afin de mieux repérer les erreurs et les corriger. Pour finir, un des facteurs qui a fait que l'on n'a pas pu faire ce projet de la meilleure manière possible était notre léger manque d'organisation claire. Il y avait certes une bonne base d'organisation et le projet était parti dans la bonne direction initialement, cependant l'organisation n'était pas assez solide pour faire face à certains problèmes et à ralenti le projet dans son ensemble. Bien que cela se soit produit nous avons appris de nos erreurs et feront en sorte dans nos futurs projets que de telles situations ne se reproduisent plus.

Pour conclure, ce projet était une expérience enrichissante tant au niveau du développement que de la gestion globale du projet. Nous avons fait beaucoup de bonnes choses comme un certain nombre d'erreurs, or c'est grâce à ces dernières que nos futurs projets se dérouleront de manière encore plus optimale que précédemment, toujours en s'améliorant. De plus, la gestion de projet n'étant pas notre point fort nous avons beaucoup appris et tout cela nous sera bien utile à l'avenir et sommes donc satisfait de notre travail effectué lors de ce projet.