

# Miage Blog

## Langages du Web - Serveur

[vincent.bouvier@kware.fr](mailto:vincent.bouvier@kware.fr)

### Modalités

---

#### Les Jalons

Ce TD est un TD noté qui s'étale sur 3 semaines. **Chaque dimanche**, vous devrez m'envoyer une version de votre TD dans un fichier **ZIP**. Chaque version doit être accompagné d'un mini-compte rendu qui fait état:

- répartition du travail dans la semaine
- Les différents schémas de travail si il y en a
- des fonctionnalités implémentées
- des fonctionnalités qui seront implémentées la semaine suivante
- des problèmes/difficultés rencontrés pendant la semaine
- des éventuelles solutions trouvées
- des captures d'écrans si vous le souhaitez pour illustrer vos propos

---

#### La note finale

La note finale sera déterminée d'après les critères suivants:

- Assiduité aux cours et TD
- Qualité des rapports hebdomadaires
- Fonctionnalités implémentées à la fin du projet
- Qualité du code
  - Utilisation de fonction
  - Distinction des **MODELS** (les utils) / **VUES** (le html) / **CONTROLLERS** (les scripts)
- Utilisabilité du site (inclus le CSS même simple qui contribue à l'expérience utilisateur)

---

#### Aide

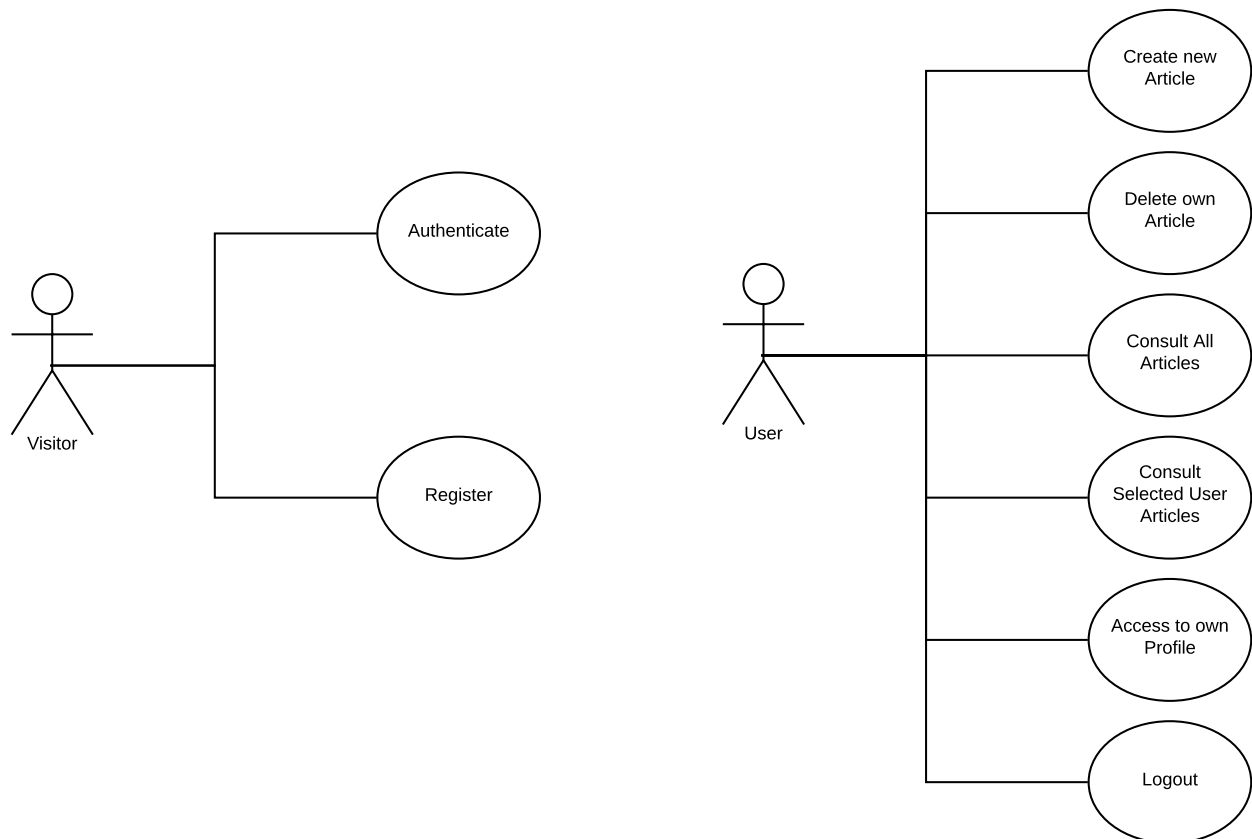
À la fin de ce document, une petite page d'aide sur l'utilisation des Exceptions.

# Énoncée

## Introduction

Dans ce TD, vous allez devoir implémenter un Blog. Ce Blog devra utiliser une base de données, les sessions et les cookies afin de rendre la navigation pour l'utilisateur la plus agréable possible.

Voici tout d'abord la liste des fonctionnalités que vous devrez implémenter selon les cas d'utilisations:



**FIGURE 1. DIAGRAMME DES CAS D'UTILISATION**

Les cas d'utilisation sont séparées en deux parties selon les différents acteurs:

- **un visiteur:** correspond à l'utilisateur qui navigue sur votre site sans être authentifié. Sur la figure 1, nous voyons que les fonctionnalités attribuées à celui-ci sont très limitées. Un utilisateur non authentifié ne doit avoir aucun accès aux articles du blog.
- **un utilisateur:** correspond à l'utilisateur du blog qui est connecté. Celui-ci a accès toutes les fonctionnalités du blog qui incluent la création d'article, la suppression de ses propres articles, etc....

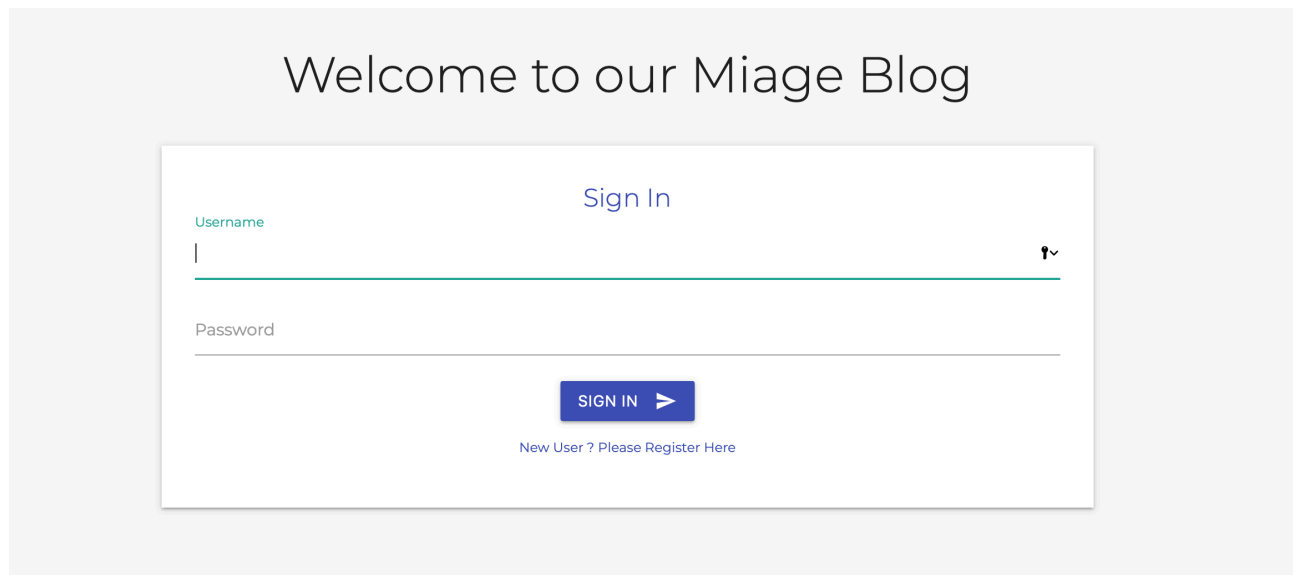
## Les cas d'utilisation

Les pages suivantes détaillent tous les cas d'utilisation présent sur la figure 1. Elles vous donnent des indices et des check-list à vérifier. Un TD qui répond correctement à l'ensemble des points de check-list peut prétendre à la note de 20/20 sachant les autres critères spécifiés plus haut.

# AUTHENTICATE

---

Un utilisateur non authentifié doit pouvoir s'identifier sur le site afin d'avoir accès aux fonctionnalités complètes du site. L'utilisateur doit s'être enregistré au préalable.



## **Check List:**

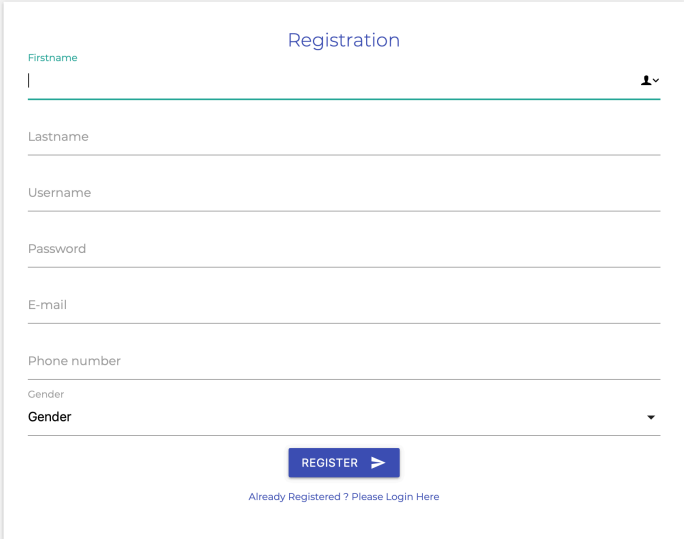
- ☐ Ce formulaire s'affiche en page d'accueil si l'utilisateur n'est pas connecté
- ☐ Si l'utilisateur saisie un nom d'utilisateur ou un mot de passe incorrecte, un message d'erreur s'affiche.
- ☐ Si l'utilisateur clique sur le lien "New User ? Please Register Here", un formulaire d'enregistrement doit remplacer le formulaire d'authentification
- ☐ Un utilisateur doit être automatiquement re-connecté lors de sa première visite de la journée à l'aide des COOKIES.

## **Hints**

- Utilisez les COOKIES pour restaurez une SESSION lorsque celle-ci n'est pas encore initialisée

# REGISTRATION

Le formulaire d'enregistrement doit s'afficher seulement sur demande de l'utilisateur. Il permet à l'utilisateur de s'enregistrer sur le site. Une fois enregistrer, l'utilisateur doit être connecté au site. On ne demandera pas à l'utilisateur de s'authentifier après son enregistrement.



The image shows a registration form titled "Welcome to our Miage Blog". The form is titled "Registration" and contains the following fields: Firstname, Lastname, Username, Password, E-mail, Phone number, and Gender (a dropdown menu). There is a "REGISTER" button with a right arrow. Below the button, it says "Already Registered ? Please Login Here".

## Check List:

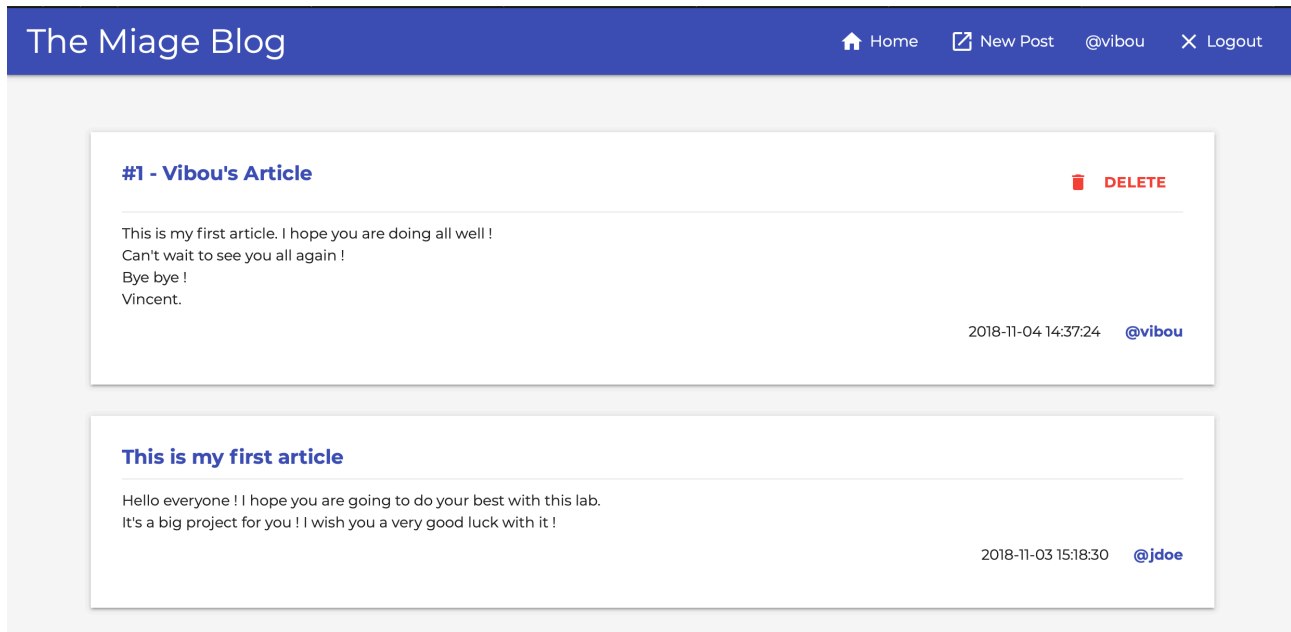
- ☐ Ce formulaire s'affiche en page d'accueil si l'utilisateur a cliqué sur le lien "New User ? Please Register Here" sur le formulaire d'authentification
- ☐ Si l'utilisateur clique sur le lien "Already Registered ? Please Login Here", il doit être redirigé vers le formulaire d'authentification.
- ☐ Si le nom d'utilisateur existe déjà dans la base de données, une erreur doit être affichée à l'utilisateur.
- ☐ Si une erreur se produit lors de la requête d'insertion, une erreur doit être affichée à l'utilisateur.
- ☐ Si une erreur se produit, assurez vous de revenir sur le formulaire d'enregistrement et pas d'authentification

## Hints

- Vous pouvez utiliser le passage de variable par URL pour savoir si vous devez afficher tel ou tel formulaire
- Pour les erreurs, utilisez le passage de variable par l'URL (ATTENTION à bien utiliser **urlencode**)

## CONSULT ALL ARTICLES (HOME)

Ce cas d'utilisation correspond à la page d'accueil d'un utilisateur connecté. Ici nous allons détailler différents détails qui devront être présent sur l'ensemble de vos pages où l'utilisateur est connecté.



### Check List:

- ☐ Chacune de vos pages doit avoir le même menu et doit utiliser le même "fichier" php pour afficher le menu partout.
- ☐ Le menu doit contenir 4 items:
  - ☐ **HOME:** retour à l'accueil avec l'affichage de tous les articles
  - ☐ **NEW POST:** affiche un formulaire de création d'article
  - ☐ **@<username>:** affiche le profil utilisateur ainsi que ses articles.
  - ☐ **LOGOUT:** permet à l'utilisateur de se déconnecter
- ☐ Chaque article doit utiliser le même fichier PHP qui contient la vue d'un article. Vous allez devoir utiliser **include** et non pas **include\_once** car il se répètera pour chaque article.
- ☐ Un bouton **DELETE** devra apparaître sur chaque article qui appartient à l'utilisateur connecté
- ☐ Chaque article doit afficher:
  - ☐ Le titre de l'article
  - ☐ Le contenu
  - ☐ La date de publication
  - ☐ L'auteur
- ☐ Les articles doivent être classés dans par ordre descendant de DATE (le plus récent en premier). Utilisez **ORDER BY article.date DESC** dans votre requête.

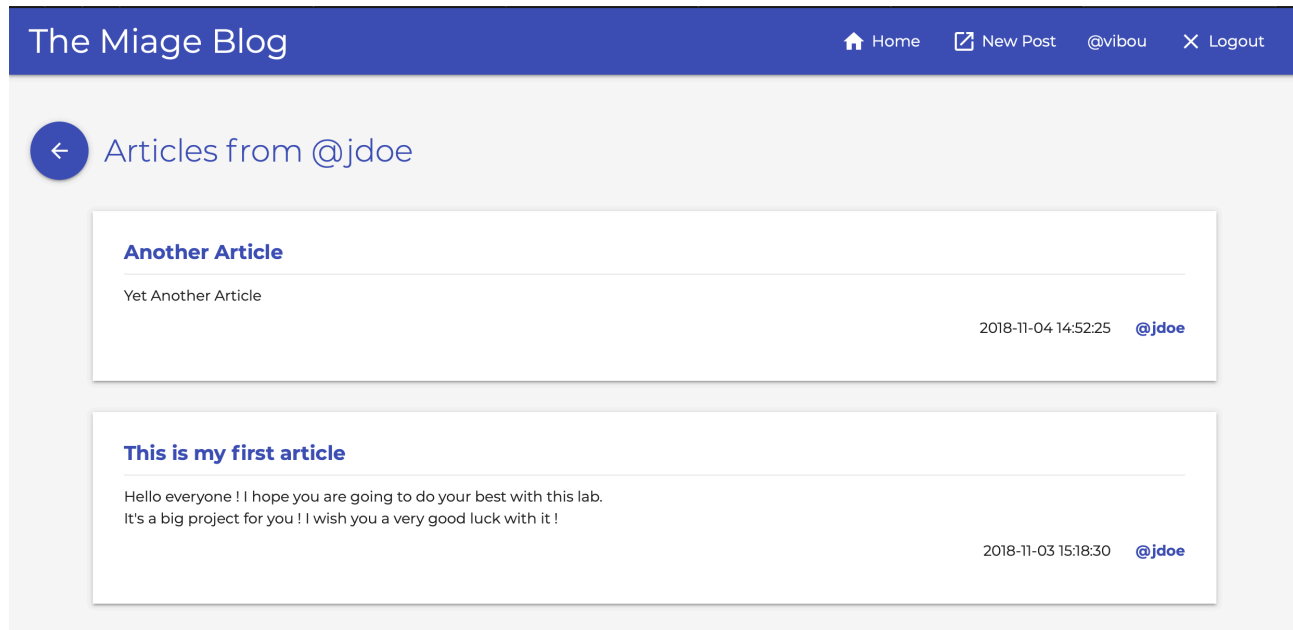
- ☐ Si l'utilisateur clique sur le nom de l'auteur, seulement les articles de l'auteur en question devront être affichés

**BONUS:**

- ☐ Afficher seulement 5 articles et utiliser de la pagination pour afficher les autres articles:
  - ☐ Utiliser **COUNT** en MySQL pour connaître le nombre d'articles
  - ☐ Utiliser **LIMIT** et **OFFSET** pour effectuer de la pagination
    - **LIMIT** donne le nombre max de d'article à obtenir
    - **OFFSET** définit le numéro du premier article à partir duquel la limite sera appliquée.

## CONSULT SELECTED USER ARTICLES

Sur la page d'accueil si un utilisateur clique sur le nom de l'auteur d'un des articles, l'utilisateur doit alors voir apparaître que les articles de l'auteur en question comme sur l'illustration suivante.



### Check List:

- ☐ Assurez vous d'utiliser la même vue MENU et ARTICLE que sur toutes les autres pages
- ☐ Vous devrez faire apparaître un lien ou un bouton pour revenir sur la page d'accueil avec tous les articles

### Hints


- Vous pouvez utiliser le passage de variable par URL pour savoir si vous devez afficher tous les articles ou seulement ceux d'un utilisateur.

## DELETE OWN ARTICLE

---

Dans votre vue ARTICLE assurez vous de prévoir un bouton **DELETE** qui doit apparaitre seulement si l'article a été écrit par l'utilisateur actuellement connecté.

#1 - Vibou's Article

 DELETE

---

This is my first article. I hope you are doing all well !  
Can't wait to see you all again !  
Bye bye !  
Vincent.

2018-11-04 14:37:24

@vibou

### Check List:

- ☐ Dans le script de suppression d'article assurez vous que l'article appartient bien à l'utilisateur connecté
- ☐ Si une erreur se produit lors de la suppression de l'article affichez cette erreur sur la page d'accueil lorsque vous revenez.




## CREATE NEW ARTICLE


Le formulaire de création d'article doit s'afficher lorsque l'utilisateur clique sur le bouton "New Post" du menu.

Create a new Post

Title

Content

 CANCEL

 CREATE

### Check List:

- ☐ Si le titre OU le contenu est vide alors un message d'erreur doit s'afficher
- ☐ Lorsque l'article est créé, un message d'information doit alors s'afficher pour dire que l'article a bien été créé
- ☐ Assurez vous que seul un utilisateur connecté puisse créer un article



### Hints

- Vous pouvez utiliser le passage de variable par URL pour savoir si vous devez afficher une information ou une erreur

## ACCESS YOUR OWN PROFILE

Cette page doit afficher les informations de l'utilisateur actuellement connecté. Attention à afficher seulement les informations présentes sur la capture d'écran suivante.

The Miage Blog

[Home](#) [New Post](#) [@vibou](#) [Logout](#)

### Your Info

username:	vibou
firstname:	Vincent
lastname:	Bouvier
email:	vincent.bouvier@kware.fr
phone:	0634567890
gender:	MALE

### Your Articles

#1 - Vibou's Article

DELETE

This is my first article. I hope you are doing all well !  
Can't wait to see you all again !  
Bye bye !  
Vincent.

2018-11-04 14:37:24 @vibou

### Check List:

- ☐ Affichez les informations de l'utilisateur de manière claire. Vous pouvez utiliser un liste ou un tableau
- ☐ Utilisez toujours la même vue article
- ☐ Assurez vous que l'utilisateur puisse supprimer un de ses articles

## LOGOUT

---

La fonction logout doit déconnecter l'utilisateur courant et retourner sur la page d'accueil avec le formulaire d'authentification.

### **Check List:**

- ☐ Supprimez bien la SESSION
- ☐ Supprimez aussi les COOKIES
- ☐ Retour au formulaire d'authentification

---

## Capturer les erreurs

Une erreur peut être envoyée à n'importe quel moment: lorsque la connexion à la base ne fonctionne pas, ou lors de l'utilisation de la méthode **execute(request, data)** que nous avons fait dans **dbUtils**.

Par ailleurs, vous pouvez également programmer que vos fonctions présentes dans vos différents fichiers **utils** peuvent envoyer des erreurs lorsqu'il y a un comportement habituelle.

La mécanique d'envoi et de capture d'erreur est la suivante:

La fonction login ci-dessous est une fonction qui est en charge d'authentifier un utilisateur d'après son nom d'utilisateur et son mot de passe.

Si l'utilisateur n'existe pas, cette fonction envoie une exception à l'aide du mot clé **throw new Exception(<message>);**. Le message donné lors de la création d'exception sera alors récupérable par le développeur qui **capture** l'exception.

```
function login($username, $password) {
    if (!userExists($username, $password)) {
        throw new Exception("Wrong Username or Password ($username,
$password)");
    }

    //Set Session
    //Set Cookie
}
```

Par exemple le bout de code suivant capture l'exception envoyée par la fonction login et redistribue le message à index.php à l'aide du passage de variable par URL. Si une exception est envoyée par la fonction login alors le code qui suit l'appelle de la fonction login n'est pas exécuté, et le code inclus dans le bloque **CATCH** est exécuté. Dans ce bloque on peut récupérer le message de l'exception à l'aide de **\$ex->getMessage();**. Notez également l'utilisation d'**urlencode** pour le passage de valeur par URL.

```
try {
    login($username, $password);
    header("location: ../index.php");
    exit();
} catch (Exception $ex) {
    $msg = $ex->getMessage();
    $error = urlencode($msg);
    header("location: ../index.php?error=$error");
    exit();
}
```