

# Projet Alpha

Préparé pour : Jean-Philippe Demory, Client

Préparé par :

**Bigourd Paul**, Gestionnaire GIT, Web Developer

**Jean-Baptiste Calandreau**, Gestionnaire Base de données

**Valentin Lacour**, Chef de projet, Web Developer

**Rémi Lemaire**, Web Developer

**Antoine Richard**, Web Designer

**Nicolas Rocroy**, Gestionnaire Documentation et Tests

Septembre - Décembre 2016

Référence projet : E10

PROJET ALPHA	1
Sommaire de gestion	3
Premiers Pas	4
Charte Graphique	8
Base de données	10
Fichiers de paramètres	17
Fonctions Génériques	20
Fonctionnalités Back-End générales	22
Parties Communes	24
Publications d'informations	25

# SOMMAIRE DE GESTION

## Objectif

Permettre ou faciliter l'intercommunication entre les Tuteurs Entreprises (TE) et les Tuteurs IUT (TI) lors des périodes de stage.

## Buts

Donner la possibilité au TE de répondre au questionnaire de stage de leur stagiaire directement sur informatique. Avoir la possibilité d'accéder rapidement aux questionnaires répondus pour les TI et donner leur donner la possibilité de communiquer directement avec les TE pour compléter des informations sur le stage. Permettre à un administrateur de gérer tout les comptes (TE, TI et stagiaires) et d'installer l'application facilement. Rendre l'application multilingue pour les stages à l'étranger.

## Solution

Créer un site web de gestion en PHP répondant à toutes les demandes.

## Contour du projet

Le projet se séparera en différentes parties :

- **Gestion des utilisateurs et des élèves** (1): Gestion par l'administrateur, gestion de son compte personnel, affichage de coordonnées pour les utilisateurs ayant droit.
- **Questionnaires et Gestion de réponses** (1): Affichage du questionnaire pour les TE, affichage des réponses pour les TI, possibilité de donner des informations complémentaires et de définir (et rehausser) une date butoir.
- **Communication inter-utilisateur** (2): Envoi de message entre utilisateurs selon leurs droits.
- **Gestion multilingue** (2): Affichage de textes et des questionnaires en différentes langues (FR,EN,DE,ES,PO).
- **Notifications** (2): Notifications push d'informations, envois d'email d'information d'événement.
- **Installation guidé** (3): Installation de la base de données et création du compte administrateur plus simple lors du premier lancement du site.
- **Inscription, connexion et sécurité** (1): Permettre une inscription personnalisé pour les utilisateurs, un connexion plus simple grâce à une page de connexion intelligente, une sécurité des données grâce à des méthodes de cryptage récentes et testés.
- **Aide et documentation** (1): Fournis une documentation sous forme de PDF et une aide sur le site de gestion.
- **Publication de messages d'information** (4): Permet la publication de messages d'informations pour les utilisateurs autorisés par l'administrateur.

(1) => Niveau d'importance de traitement de 1 à 4

## PREMIERS PAS

### Matériel utilisé

IDE's : WebStorm, PHPStorm, Affinity Designer, DataGrip

Fonctionnement et Tests : MAMP, PHPMyAdmin

Documentations : Pack Office, Pack iLife

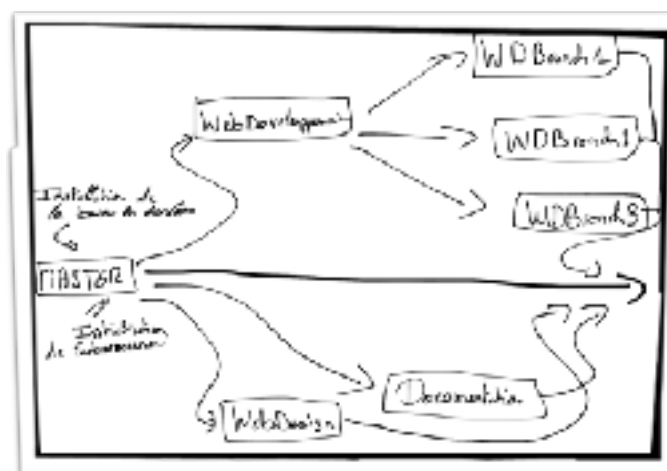
Versionning : Gitkraken, GitHub desktop, Bitbucket (dépôt distant) => Travail en workflow GIT

### Organisation de l'équipe de travail

Les tâches de développement du site on était réparti selon les qualifications de chacun :

- **Bigourd Paul**, Gestionnaire GIT, Web Developper
- **Jean-Baptiste Calandreau**, Gestionnaire Base de données
- **Valentin Lacour**, Chef de projet, Web Developper
- **Rémi Lemaire**, Web Developper
- **Antoine Richard**, Web Designer
- **Nicolas Rocroy**, Gestionnaire Documentation et Tests

Le travail workflow pour GIT oblige une organisation par branche optimisé pour le travail de groupe :



### Création d'un logo

Le logo du site sera donc celui-ci :



Il sera utilisé pour les documentations et le preloader du site.

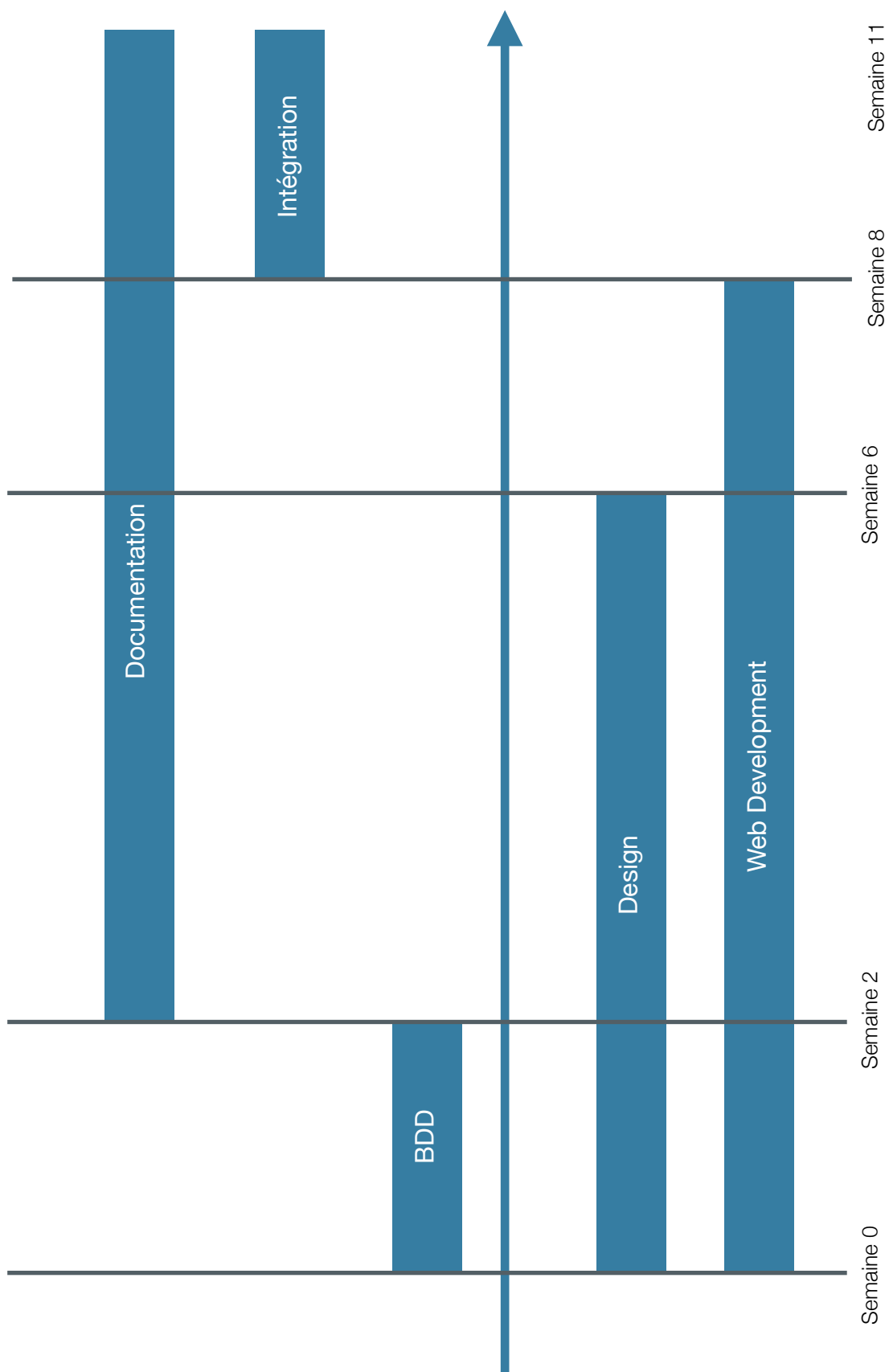
## Design du site

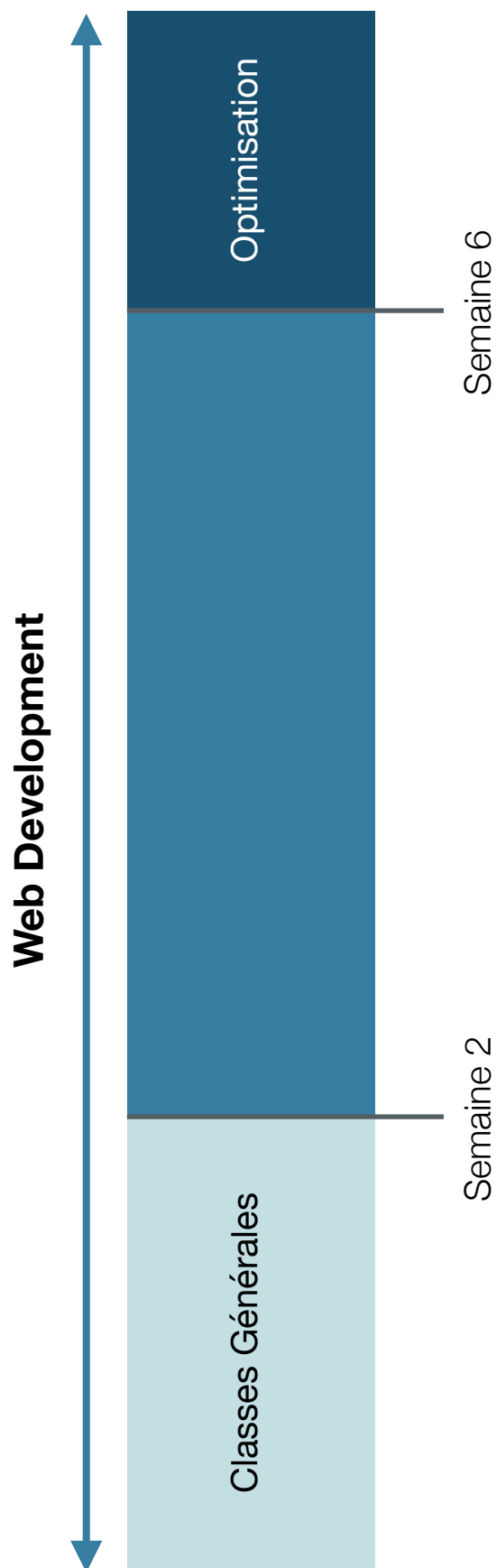
Pour le design du site on se tournera vers un template Bootstrap (Framework CSS) de Dashboard (Tableau de bord). Le Design devra être esthétique et devra répondre aux exigences du projet.

## Arborescence du projet

L'arborescence devra permettre le travail en collaboration et permettre une organisation simple des fichiers :

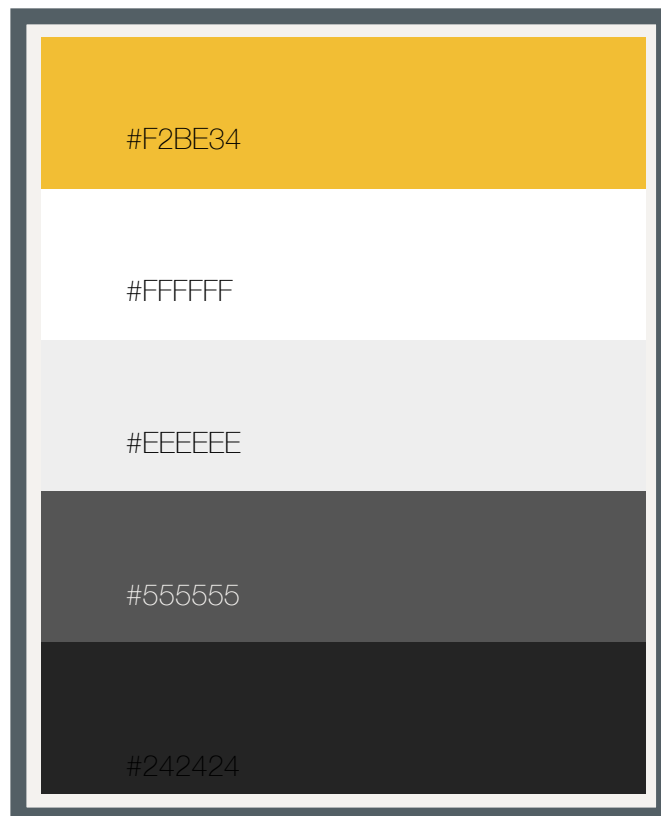
- **Public**
  - **CSS**
  - **Img**
  - **Scripts**
    - **Lib**
      - > Scripts JS de fonctionnement
    - **Controller**
      - > Scripts JS de mise à jour des données
  - **SCSS**
- **Private**
  - **Parameters**
    - > Fichier de paramétrage (BDD, Pages,...)
  - **Functions**
    - > Fonctions génériques
  - **Library**
    - > Package de classes PHP
    - **General**
    - **App**
    - **Other**
  - **Models**
    - > Fichiers de rendu de données
  - **Views**
    - **Components**
    - **Pages**
  - **Datas**
- **Tests**
  - **Back-End**
  - **Front-End**
  - **Other**
- **Docs**
  - **Deliverable**
  - **Functional**
  - **Other**
- **Dist**
  - Index.php
  - Readme.md





## CHARTRE GRAPHIQUE

### Couleurs :



### Police :

Opens Sans

(<https://fonts.google.com/specimen/Open+Sans>)

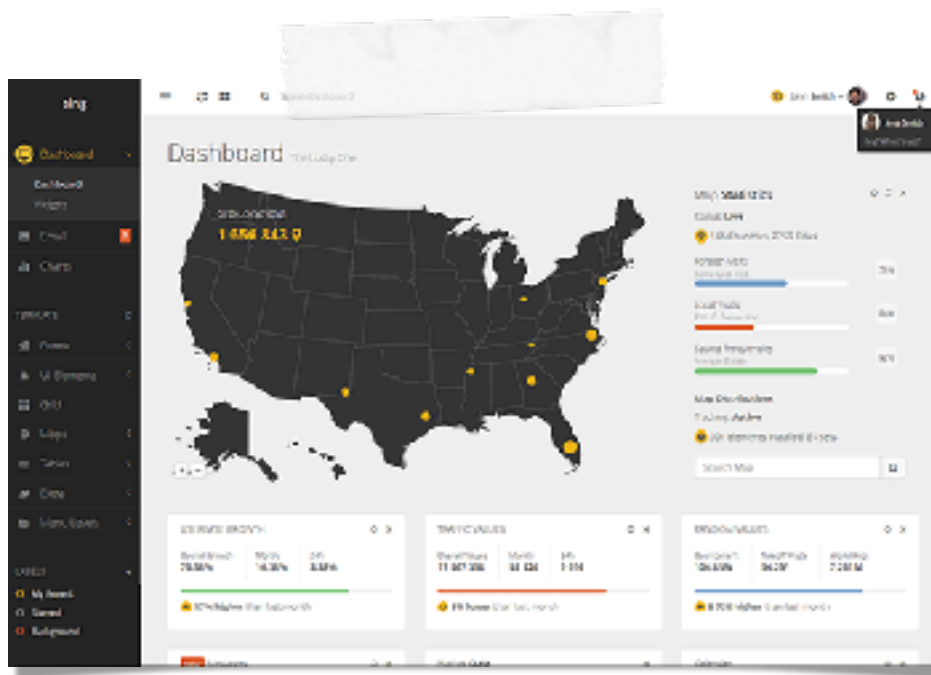
### Logo :





## Template :

Sing App (<http://demo.flatlogic.com/sing-wrapbootstrap-3.2.0/angular/>)



## Arborescence de vues :

Les vues seront placés dans le dossier « ./tests/front-end/ » et ne seront qu'en langage Front-End (HTML, CSS, JS)

Dossiers :	Admin	TE	TI	Mails
Fichiers	Pages Communes			Fichiers triés selon les besoins
	Pages Spécifiques			
	Donner les mêmes noms de pages dans la plupart des cas			

## BASE DE DONNÉES

### Tables :

users (ID\*, name, fname, type, email, password, phone, address, zip\_code, city, country, language, registration\_date, activation\_date, deletion\_date, last\_login\_date, publication\_entitled)

students (ID\*, ID\_TE, ID\_TI, name, fname, birth\_date, group, email, phone, address, zip\_code, city, country, informations, creation\_date, deadline\_date, answered)

answers (ID\*, ID\_STUDENT, id\_survey, id\_question, id\_answer, comments, request\_date)

messages (ID\*, ID\_SENDER, ID\_RECIPIENT, object, content, viewed, deleted, ID\_RESPOND, sending\_date)

messages\_attachements (ID\*, ID\_MESSAGE, link, description, type\_file)

notifications (ID\*, ID\_USER, content, link, viewed, requested\_date, deleted)

action\_report (ID\*, ID\_USER, content, requested\_date)

stats (ID\*, ID\_USER, page\_viewed, ip\_viewer, country\_viewer, platform\_viewer, os\_viewer, browser\_viewer, viewing\_date)

posts (ID\*, ID\_USER, content, deleted, publication\_date)

posts\_attachements (ID\*, ID\_POST, link, description, type\_file)

posts\_comments (ID\*, ID\_POST, ID\_USER, content, deleted, publication\_date)

posts\_like (ID\*, ID\_POST, ID\_USER, requested\_date)

### Vues :

Nom	Usage
show_all_users	Afficher tous les utilisateurs non supprimés (jointure avec les stats, affichage d'un booléen pour l'activation)
show_te_users	Afficher tous les tuteurs entreprise non supprimés (jointure avec les stats, affichage d'un booléen pour l'activation)
show_ti_users	Afficher tous les tuteurs IUT non supprimés (jointure avec les stats, affichage d'un booléen pour l'activation)

Nom	Usage
show_deleted_users	Afficher tous les utilisateurs supprimés (jointure avec les stats, affichage d'un booléen pour l'activation)
show_unactivated_users	Afficher tous les utilisateurs non activé
show_all_students	Afficher tous les étudiants non supprimés (jointure avec tuteurs)
show_deleted_students	Afficher tous les étudiants supprimés (jointure avec tuteurs)
show_answered_students	Afficher tous les étudiants qui ont répondu (jointure avec tuteurs)
show_posts	Affiche les publications des utilisateurs

**Procédures :**

Nom	Arguments	Usage
<b>add_user</b>	name, fname, type, email, password, phone, address, zip-code, country, language	Ajoute un nouvel utilisateur (registration_date = NOW() )
<b>edit_user</b>	id_user, name, fname, type, email, password, phone, address, zip-code, country, language	Modifier les données d'un utilisateur
<b>delete_user</b>	id_user	deletion_date = NOW()
<b>delete_all_user</b>	//	deletion_date = NOW() pour tous les utilisateurs
<b>undelete_user</b>	id_user	deletion_date = null ou "0000-00-00"
<b>activate_user</b>	id_user	activation_date = NOW()
<b>update_last_login_user</b>	id_user	last_login_date = NOW()
<b>add_student</b>	id_te, id_tu, name, fname, group, email, phone, address, zip-code, city, country, informations, deadline_date	Ajoute un étudiant (creation_date = NOW())

Nom	Arguments	Usage
<b>edit_student</b>	id_student, id_te, id_te, name, fname, group, email, phone, address, zip_code, city, country, informations	Modifie les données d'un étudiant
<b>delete_student</b>	id_student	deletion_date = NOW()
<b>delete_all_student</b>	//	deletion_date = NOW() pour tous les utilisateurs
<b>edit_deadline_date</b>	id_student, deadline_date	Modifie la date butoire d'un étudiant
<b>update_answer_status</b>	id_student, answered	Mets à jour le status de réponse au questionnaire
<b>undelete_student</b>	id_student	deletion_date = null ou "0000-00-00"
<b>add_answer</b>	id_user, id_survey, id_question, id_answer, comments	Ajoute une réponse (request_date = NOW())
<b>delete_answer</b>	id_answer	Supprime une réponse
<b>delete_all_answer</b>	id_student	Supprime toutes les réponse d'un étudiant
<b>publish_message</b>	id_sender, id_receipient, object,content, id_respond, tableau[ link, description, file_type]	Publie un message avec l'inclusion des pieces jointes
<b>delete_message</b>	id_message	Supprime un message
<b>delete_group_message</b>	id_sender, id_receipient	Supprime un groupe de message
<b>delete_all_message</b>	id_user	Supprime tous les messages d'un utilisateur
<b>add_notification</b>	id_user, content, link	Ajoute du notification (requested_date = NOW())
<b>viewed_notification</b>	id_notification	viewed = 1
<b>viewed_all_notifications</b>	id_user	viewed = 1
<b>delete_notification</b>	id_notification	deleted = 0

Nom	Arguments	Usage
<b>add_statistique</b>	id_user, page_viewed, ip_viewer, country_viewer, platform_viewer, os_viewer, browser_viewer	Ajoute une statistique (viewing_date = NOW())
<b>delete_statistique</b>	id_statistique	Supprime une statistique
<b>add_post</b>	id_user, content, tableau[ link, description, file_type]	Ajoute un publication avec l'inclusion des pièces jointes
<b>edit_post</b>	id_post, id_user, content, tableau[ link, description, file_type]	Modifie une publication
<b>delete_post</b>	id_post	Supprime une publication
<b>add_post_attachement</b>	id_post, link, description, file_type	Ajoute une piece jointe à une publication
<b>delete_post_attachement</b>	id_post	Supprime une piece jointe d'une publication
<b>add_comment</b>	id_post, id_user, content	Ajoute un commentaire (publish_date = NOW())
<b>edit_comment</b>	id_comment, id_post, id_user, content	Modifier un commentaire
<b>delete_comment</b>	id_comment	Supprime un commentaire
<b>like_post</b>	id_post, id_user	Ajoute un like à une publication
<b>unlike_post</b>	id_post, id_user	Retire un like à une publication

### Functions :

Nom	Argument	Usage	Valeur de retour
<b>show_user</b>	id_user	Affiche les données d'un utilisateur	Table
<b>show_student</b>	id_student	Affiche les données d'un étudiant	Table

Nom	Argument	Usage	Valeur de retour
show_answers	id_student	Affiche les réponses d'un étudiant	Table
show_students	id_user	Affiche les étudiants d'un tuteur (Entreprise ou IUT)	Table
show_sended_message	id_user	Affiche les message envoyés par un utilisateur	Table
show_received_message	id_user	Affiche les message reçus par un utilisateur	Table
show_deleted_message	id_user	Affiche les message supprimés par un utilisateur	Table
show_message	id_message	Affiche les données d'un message	Table
show_message_attachement	id_message	Affiche les pieces jointes d'un message	Table
show_notifications	id_user	Affiche les notification d'un utilisation (id_user = parametre ou 0)	Table
show_posts	id_user	Affiche les publications d'un utilisateur	Table
show_posts_attachement	id_post	Affiche les pieces jointes d'une publication	Table
show_comments	id_post	Affiche les commentaires d'une publication	Table
show_likes	id_post	Affiche les personnes qui ont « liké » une publication	Table
count_post	//	Compte le nombre de publications	Entier
count_like	id_post	Compte le nombre de like d'une publication	Entier
count_comments	id_post	Compte le nombre de commentaires d'une publication	Entier

Nom	Argument	Usage	Valeur de retour
<b>count_notifications</b>	id_user	Compte le nombre de notifications d'un utilisateur	Entier
<b>count_received_messages</b>	id_user	Compte le nombre de message reçu	Entier
<b>count_deleted_messages</b>	id_user	Compte le nombre de message supprimés	Entier
<b>count_sended_messages</b>	id_user	Compte le nombre de message envoyés	Entier
<b>count_new_messages</b>	id_user	Compte le nombre de nouveaux message	Entier
<b>count_users</b>	//	Compte le nombre d'utilisateurs non supprimés	Entier
<b>count_students</b>	//	Compte le nombre d'étudiants non supprimés	Entier
<b>count_deleted_users</b>	//	Compte le nombre d'utilisateurs supprimés	Entier
<b>count_deleted_students</b>	//	Compte le nombre d'étudiants supprimés	Entier
<b>verification_connection</b>	email,password	Vérifie s'il existe un compte avec le meme email et le meme mot de passe (conversion en MD5)	Booléen
<b>verification_email</b>	email	Vérifie s'il un email existe dans la table « users »	Booléen
<b>is_connected</b>	id_user	Vérifie si l'utilisateur à un « last_login_date » < 3s	Booléen
<b>is_activated</b>	id_user	Vérifie si un utilisateur est activé	Booléen
<b>is_deleted</b>	id_user	Vérifie si un utilisateur est supprimé	Booléen
<b>can_duplicate</b>	id_user	Vérifie si l'utilisateur a le droit de publier	Booléen

---

### Événements :

Nom	Type	Date ou période	Usage
auto_deletion_users	Récuratif	1 jours	Supprime tout les utilisateurs où « deleted_date » > 30j
auto_deletion_students	Récuratif	1 jours	Supprime tout les étudiant où « deleted_date » > 30j
initialize	Unique	(à choisir)	Supprime tous les étudiants, ainsi que les données liés

### Déclencheurs :

Nom	Événements	Usage
save-to-action-report	Insert, Update, Delete	Sauvegarde les actions réalisés en SQL dans la table action_report



---


## FICHIERS DE PARAMÈTRES

Tous les fichiers de paramètre seront de type « .json ».

### Configuration de l'application

Nom de fichier : app_config.json	
Nom	Usage
name	Nom du Site
description	Description du Site
keywords	Mots clés du Site
author	Auteur(s) du Site
copyright	Droits d'usage du Site
version	Version de l'application
last-update	Date de dernière mise à jour

### Référencement des pages de l'application

Nom de fichier : private/page_datas.json		
	Nom	Usage
	id	Identifiant de la page
	name	Nom de la page
	logo	Logo de la page (text)
	inNavigation	Booléen d'affichage dans la barre de navigation
	script_link	Tableau des scripts JS à inclure

### Identifiants de connexions à la base de données


Nom de fichier : private/db_datas.json	
Nom	Usage
host	Hôte de la base
login	Identifiants de la base

Nom de fichier : private/db_datas.json	
Nom	Usage
password	Mot de passe de la base
db_name	Nom de la base
prefix	Préfix des tables, vues, procédures et fonctions


### Textes de l'application

Nom de fichier : public/languages/fr_FR.json	
Nom	Usage
general	Tableau des parties generals
↳ date	Tableau des textes de date
...	..
pages	Tableau des pages
↳ user	Tableau du parties
↳ table1	Tableau des textes
other	Tableau des autres données

### Droits utilisateurs


Nom de fichier : public/rights.json		
	Nom	Contenu
	id	Identifiant du type
	name	Nom du type
	rights	Tableau des pages autorisés

### Référencement de pays


Nom de fichier : public/countries.json		
	Nom	Contenu
	Codet ISO 3166-1-alpha-2	Nom du pays

---


## Référencement des navigateurs

Nom de fichier : public/browser.json		
	Nom	Contenu
	id	Identifiant du navigateur
	name	Nom du navigateur
	header_mark	Marque dans l'entête-PHP

## Référencement des plateformes

Nom de fichier : public/platform.json		
	Nom	Contenu
	id	Identifiant de la plateforme
	name	Nom de la plateforme
	header_mark	Marque dans l'entête-PHP

## Référencement du système d'exploitation

Nom de fichier : public/os.json		
	Nom	Contenu
	id	Identifiant de l'OS
	name	Nom de l'OS
	header_mark	Marque dans l'entête-PHP

---

## FONCTIONS GÉNÉRIQUES

### Autoloader.php

Fonction d'auto-chargement des classes PHP.

```
//Fonction d'auto-chargement de classes PHP lors de l'appel à la construction
function __autoload($class){
    $class = str_replace('\\', '/', $class);
    $path = $_SERVER['DOCUMENT_ROOT']. "/private/libraries/{$class}.php";
    if (file_exists($path)) {
        require_once $path;
        return true;
    }
    return false;
}
```

### Linker.php

Fonction de récupération de paramètre sous forme d'un tableau.

```
//Fonction de récupération et de conversion des paramètres
function link_parameters($filename){
    $json_url = $_SERVER['DOCUMENT_ROOT']. "/private/parameters/{$filename}.json";
    if(!file_exists($json_url))
        return array();
    // Si le fichier n'existe pas on revoie un tableau vide
    $json = file_get_contents($json_url);
    return json_decode($json, TRUE);
    // Sinon on retourne le tableau des paramètres
}
```

---

## Db\_connect.php

Fonction de connexion à la base de données.

```
//Fonction de connexion à la base de données
function db_connect() {
    include_once $_SERVER['DOCUMENT_ROOT'].'/private/functions/linker.php';
    $datas = link_parameters("private/db_datas");

    $dsn = 'mysql:dbname='.$datas['db_name'].';host='.$datas['host'];
    $user = $datas['login'];
    $password = $datas['password'];

    try {
        return new PDO($dsn, $user, $password);
    } catch (PDOException $e) {
        die('Connexion échouée : ' . $e->getMessage());
    }
}
```

# FONCTIONNALITÉS BACK-END GÉNÉRALES

## General.php

Classe mère du namespace « general ».

## Date.php

Classe de traitement des dates

Functionnalités	Usage	
constructeur	Prend en compte une date et une langue	
update	Mise à jour d'une date	
add	Addition de date	
sub	Soustraction de date	
format	Affichage de date sous différentes formes possible	
	Id	Format
	d	jour (1 à 31)
	dd	jour (01 à 31)
	m	mois (1 à 12)
	mm	mois (01 à 12)
	M	mois (janvier à décembre)
	yy	année (00 à 99)
	yyyy	année (ex : 2016)
	w	jour semaine (1 à 7)
	W	jour semaine (lundi à dimanche)
	h	Heure (1 à 24)
	hh	Heure (01 à 24)
	m	Minutes (1 à 60)
	mm	Minutes (01 à 60)
	s	Secondes (1 à 60)
	ss	Secondes (01 à 60)
	dd/mm/yy	01/01/01
	d_m_yyyy	1 janvier 2001
	...	...



Gestion des dates en multilingues

## Upload.php

//Interface pour SQL

---

Parties communes



---

# PUBLICATIONS D'INFORMATIONS

## Front-end

Une seule page sera dédiée pour la partie publication, elle sera divisée en 2 parties :

- Publication (Pour les utilisateurs autorisés)
- Fil d'actualités

La partie publication devra présenter une zone de texte WYSIWYG qui permettra l'inclusion de fichiers multiples (photos, documents,...) ainsi qu'un bouton permettant de soumettre la publication. L'inclusion de fichiers se fera en AJAX et le framework TINYMCE (<https://www.tinymce.com/>) avec le thème « inlite » sera utilisé

Type d'envoi	Entrées HTML	Nom des entrées
P O S T	zone de texte	post_text_content
	bouton de soumission	post_publication_submit

La partie fil d'actualités devra présenter plusieurs publications ou aucune. Chaque publication sera composée de l'auteur, de la date, du contenu (texte + pièces jointes) et d'une partie « like & comment ». Deux types de pièces jointes pourront se présenter :

- Image(s) => Affiche d'une image ou d'une galerie d'image
- Documents => Icône précisant la présence d'un document à télécharger

## Fonctionnalités