**Plan d’Assurance Qualité EPSI (PAQE)**

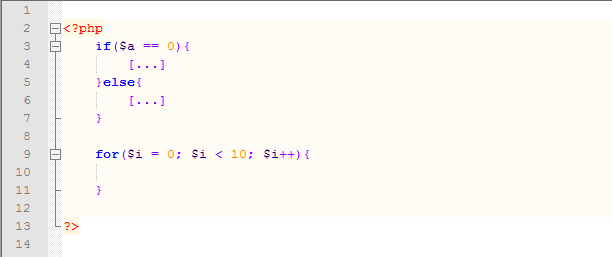
**I - Normes de codage**

* 1. ***Accolades***

Les accolades doivent être situées à la fin d’une ligne commençant par « if », « while », « for »…

En revanche, les accolades de fin doivent être sur une ligne spécifique. Dans le cas d’un « if », s’il y a un « else », l’accolade de fin de « if » se trouve juste avant le « else ».

Exemple :



* 1. ***Indentation***

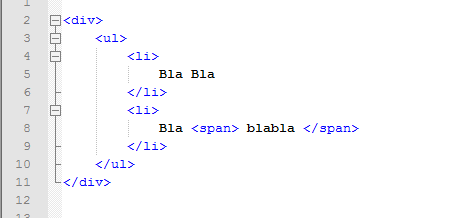
L’indentation doit être parfaitement respectée pour que le code soit agréable à lire.

Une indentation correspond à un Tab, et non pas à des espaces, qui prennent plus de place en mémoire.

Après l’accolade de chaque « if », « while », « for », « do…while », et après toutes les balises HTML ayant un contenu, « <div> », « <a> », « <ul> », il faut obligatoirement passer une ligne, et décaler l’indentation de 1 vers la droite.

Seuls quelques exceptions autorisent la non-indentation : « <span> », qui est une balise in-line, et les titres « <h1> », « <h2> », … qui ne prennent pas de place… Dans ce cas, le texte est placé directement entre les deux balises, sur la même ligne.

Exemple :



* 1. ***Commentaires***

Chaque fonction doit être commentée. De même, chaque grande partie de code doit être commentée.

On privilégiera la forme : « /\* …. \*/ », qui peut s’étaler sur plusieurs lignes.

En revanche, il est tout de même possible de faire un commentaire de type « // » en fin de ligne, notamment pour expliquer ce à quoi correspond une variable.

**II - Normes d’écriture**

Tous les noms de classes, variables, méthodes, … doivent être écrits en ANGLAIS.

Aucun nom ne doit contenir de caractères spéciaux.

* 1. ***Classes***

Le nom de la classe doit toujours commencer par une majuscule.

S’il y a plusieurs mots dans le nom de la classe, chaque première lettre des mots doit être écrite en majuscule.

Exemples :

maClasse MaClasse maclasse Maclasse

En règle générale, le nom de la classe doit aussi correspondre au nom du fichier. Elle doit aussi être le plus explicite possible.

* 1. **Variables**

Le nom d’une variable doit toujours commencer par une minuscule.

S’il y a plusieurs mots dans le nom de cette variable, chaque première lettre des mots doit être écrite en majuscule.

Exemples :

maVariable MaVariable

Le nom de la variable doit être explicite, on doit savoir tout de suite à quoi elle sert.

L’utilisation d’une variable à un caractère (exemple : a, i, …) est exclusivement réservée aux boucles.

* 1. **Constantes**

Les noms de constantes doivent toujours être écrits intégralement en majuscule.

Les mots d’une constante doivent être séparés par un UNDERSCORE (\_).

Exemple : MA\_CONSTANTE

* 1. **Base de données**

Tous les noms de table doivent commencer par « m5f\_ »

Exemple :

La table utilisateur doit se nommer :

« mf5\_utilisateur »

**III – Normes de conception**

Pour que les besoins d'un client soient bien compris, celui-ci doit être constamment en communication avec le développeur. Il doit y avoir en permanence un « feedback », c'est à dire un retour utilisateur/client.

De plus, le client doit être le plus explicite possible dans ses souhaits. Les User Stories sont là pour ça : le client exprime un besoin par le biais d'une petite phrase qui définit ce qu'il désire vraiment.

Chaque phrase se voit alors assigner un poids. Il représente la masse de boulot correspondant à cette phrase. On affecte généralement à la plus petite User Stories le poids de 1. Grâce à cela, on peut notamment calculer le temps que durera le projet, suivant le nombre de stories points au total et, par exemple, le nombre de stories point exécuté en moyenne par semaine.

En complément, on affecte aussi pour chaque phrase une priorité. On peut pour cela utiliser la nomenclature du Modèle de MoSCoW.

* M : « Must-have », signifie qu'une fonctionnalité est Indispensable.
* S : « Should-have », signifie qu'une fonctionnalité est Souhaitable.
* C : « Could-have », signifie qu'une fonctionnalité est Possible.
* W : « Want to have but won't have », signifie qu'une fonctionnalité est Peu Importante, voire Eliminée.

On retrouve le même principe dans le modèle KANO, où le client a trois types d'exigences :

* Le « Must Have », c'est à dire les fonctionnalités qui se doivent d'exister, sans quoi le client est mécontent. Ces fonctionnalités ne sont pas toujours exprimées, mais bien souvent exigées tout de même.
* Les « Linear », que le client demande, qui le rendra satisfait.
* Les « Delighters », les fonctionnalités qui surprennent le client, qui n'y avait pas pensé, et le rendent parfois très satisfait.

**IV – User stories**

* 1. **Administrateur**
* Il peut naviguer sur l’intégralité de l’application.
* Il peut ajouter, modifier et supprimer un nouveau document.
* Il peut ajouter, modifier et supprimer un nouveau menu.
* Il peut ajouter, modifier et supprimer un nouveau sous-menu.
* Il doit confirmer l’ajout, la modification et la suppression d’un document par tous utilisateurs.
* Il peut consulter sa messagerie en cliquant sur l’onglet : « ma messagerie » depuis le menu déroulant disponible en survolant son nom en haut à droite.
* Il doit classifier un nouveau document.  
    
  1. **Contributeur**
* Il peut ajouter et modifier un nouveau document.
* Il peut consulter l’historique de ses ajouts et modifications depuis la page d’accueil.

* 1. **Tous**
* Tout utilisateur doit rentrer un identifiant et un mot de passe, puis cliquer sur le bouton « valider » pour entrer dans l’application.
* Tout utilisateur peut se déconnecter en cliquant sur « Déconnexion », disponible dans le menu déroulant qui apparait en survolant son nom en haut à droite.
* Tout utilisateur peut regarder le contenu des menus et sous-menus.
* Tout utilisateur peut effectuer des recherches depuis la barre de recherche.

**V –** **Fiches de compétences**

**Marianne DEMOULIN**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Savoir | Savoir faire | Faire savoir |
| **Langages** |  |  |  |
| C |  |  |  |
| C++ |  |  |  |
| C# |  |  |  |
| Java |  |  |  |
| Delphi |  |  |  |
| xHTML | X | X |  |
| PHP | X | X |  |
| MySQL | X | X |  |
| JavaScript |  | X |  |
| VB.net | X |  |  |
| PowerShell | X |  |  |
|  |  |  |  |
| **Conception & Outils** |  |  |  |
| Merise | X |  |  |
| UML | X | X |  |
| Symfony 2 | X |  |  |
|  |  |  |  |
| **Systèmes & Réseaux** |  |  |  |
| Adressage IP |  |  |  |
| VLAN |  |  |  |
| Modèle OSI | X |  |  |
| Linux (Debian, Ubuntu...) |  |  |  |
| Windows | X |  |  |
|  |  |  |  |
| **Divers** |  |  |  |
| Photoshop |  |  |  |
|  |  |  |  |

**Savoir être**

**Marianne STALTER**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Savoir | Savoir faire | Faire savoir |
| **Langages** |  |  |  |
| C | X |  |  |
| C++ | X |  |  |
| C# | X | X |  |
| Java | X |  |  |
| Delphi |  |  |  |
| xHTML | X | X |  |
| PHP | X | X |  |
| MySQL | X | X |  |
| JavaScript | X | X |  |
| VB.net |  |  |  |
|  |  |  |  |
| **Conception & Outils** |  |  |  |
| Merise | X | X |  |
| UML | X | X |  |
| Symfony 2 | X |  |  |
| PowerShell |  |  |  |
|  |  |  |  |
| **Systèmes & Réseaux** |  |  |  |
| Adressage IP | X |  |  |
| VLAN | X |  |  |
| Modèle OSI | X |  |  |
| Linux (Debian, Ubuntu...) | X |  |  |
| Windows | X |  |  |
|  |  |  |  |
| **Divers** |  |  |  |
| Photoshop |  |  |  |
|  |  |  |  |

**Savoir être**

**Grégoire MEURILLON**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Savoir | Savoir faire | Faire savoir |
| **Langages** |  |  |  |
| C | X |  |  |
| C++ | X |  |  |
| C# | X |  |  |
| Java | X |  |  |
| Delphi |  | X |  |
| xHTML | X |  |  |
| PHP | X |  |  |
| MySQL | X |  |  |
| JavaScript | X |  |  |
| VB.net |  |  |  |
| PowerShell |  |  |  |
|  |  |  |  |
| **Conception & Outils** |  |  |  |
| Merise | X | X |  |
| UML | X |  |  |
|  |  |  |  |
| **Systèmes & Réseaux** |  |  |  |
| Adressage IP | X |  |  |
| VLAN | X |  |  |
| Modèle OSI |  |  |  |
| Linux (Debian, Ubuntu...) | X |  |  |
| Windows | X |  |  |
|  |  |  |  |
| **Divers** |  |  |  |
| Photoshop |  |  |  |
|  |  |  |  |

**Savoir être**

**Matthieu JAGOURY**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Savoir | Savoir faire | Faire savoir |
| **Langages** |  |  |  |
| C |  |  |  |
| C++ |  |  |  |
| C# |  |  |  |
| Java | X |  |  |
| Delphi |  |  |  |
| xHTML | X | X |  |
| PHP | X | X |  |
| MySQL | X | X |  |
| XML |  | X |  |
| JavaScript |  |  |  |
| VB.net | X |  |  |
| PowerShell |  |  |  |
|  |  |  |  |
| **Conception & Outils** |  |  |  |
| Merise | X | X |  |
| UML | X | X |  |
| Symfony 2 |  |  |  |
|  |  |  |  |
| **Systèmes & Réseaux** |  |  |  |
| Adressage IP | X |  |  |
| VLAN | X |  |  |
| Modèle OSI | X |  |  |
| Linux (Debian, Ubuntu...) |  |  |  |
| Windows | X |  |  |
|  |  |  |  |
| **Divers** |  |  |  |
| Photoshop |  |  |  |
|  |  |  |  |

**Savoir être**

**Sébastien DESZCZ**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Savoir | Savoir faire | Faire savoir |
| **Langages** |  |  |  |
| C | X |  |  |
| C++ | X |  |  |
| C# | X |  |  |
| Java | X |  |  |
| Delphi |  |  |  |
| xHTML | X | X |  |
| PHP | X | X |  |
| MySQL | X | X |  |
| JavaScript | X |  |  |
| VB.net |  |  |  |
| PowerShell |  |  |  |
|  |  |  |  |
| **Conception & Outils** |  |  |  |
| Merise |  |  |  |
| UML |  |  |  |
| Symfony 2 |  |  |  |
|  |  |  |  |
| **Systèmes & Réseaux** |  |  |  |
| Adressage IP | X | X | X |
| VLAN | X | X |  |
| Modèle OSI | X |  |  |
| Linux (Debian, Ubuntu...) | X | X |  |
| Windows | X | X | X |
|  |  |  |  |
| **Divers** |  |  |  |
| Photoshop |  |  |  |
|  |  |  |  |

**Savoir être**

**Rémy MAURICE**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Savoir | Savoir faire | Faire savoir |
| **Langages** |  |  |  |
| C | X |  |  |
| C++ | X |  |  |
| C# | X |  |  |
| Java | X |  |  |
| Delphi |  |  |  |
| xHTML | X | X |  |
| PHP | X | X |  |
| MySQL | X | X |  |
| JavaScript | X | X |  |
| VB.net |  |  |  |
| PowerShell |  |  |  |
|  |  |  |  |
| **Conception & Outils** |  |  |  |
| Merise | X | X |  |
| UML | X | X |  |
| Symfony 2 | X |  |  |
| CMS |  | X |  |
| **Systèmes & Réseaux** |  |  |  |
| Adressage IP | X |  |  |
| VLAN | X |  |  |
| Modèle OSI |  |  |  |
| Linux (Debian, Ubuntu...) | X |  |  |
| Windows | X | X |  |
|  |  |  |  |
| **Divers** |  |  |  |
| Photoshop |  |  |  |
|  |  |  |  |

**Savoir être**

**Thomas DEBAS**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Savoir | Savoir faire | Faire savoir |
| **Langages** |  |  |  |
| C# |  | X |  |
| Java | X |  |  |
| xHTML | X | X |  |
| XML | X | X |  |
| PHP | X | X |  |
| Ajax | X |  |  |
| JQuery |  | X |  |
| MySQL | X | X |  |
| JavaScript | X | X |  |
| Python | X |  |  |
| VB.net | X |  |  |
| PowerShell | X |  |  |
|  |  |  |  |
| **Conception et outils** |  |  |  |
| Merise | X | X |  |
| UML | X | X |  |
| Symfony 2 | X |  |  |
| CMS |  | X |  |
| GLPI | X | X |  |
| OCS Inventory NG | X | X |  |
| Unity3D |  | X |  |
| GitHub | X | X |  |
| SVN | X | X |  |
|  |  |  |  |
| **Systèmes et réseaux** |  |  |  |
| Adressage IP | X |  |  |
| VLAN | X |  |  |
| Modèle OSI | X |  |  |
| Linux (Debian, Ubuntu...) |  | X |  |
| Windows | X | X |  |
| Serveur Web |  | X |  |
| Serveur DNS |  | X |  |
| Serveur NAS |  | X |  |
|  |  |  |  |
| **Divers** |  |  |  |
| Photoshop |  |  |  |
|  |  |  |  |

**Savoir être**

**Groupe**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Savoir | Savoir faire | Faire savoir |
| **Langages** |  |  |  |
| C | X |  |  |
| C++ | X |  |  |
| C# | X |  |  |
| Java | X | X |  |
| xHTML | X | X |  |
| XML | X | X |  |
| PHP | X | X |  |
| Ajax | X | X |  |
| JQuery | X | X |  |
| MySQL | X | X |  |
| JavaScript | X | X |  |
| Python | X |  |  |
| VB.net | X | X |  |
| PowerShell | X |  |  |
|  |  |  |  |
| **Conception et outils** |  |  |  |
| Merise | X | X |  |
| UML | X | X |  |
| Symfony 2 | X |  |  |
| CMS |  | X |  |
| GLPI | X | X |  |
| OCS Inventory NG | X | X |  |
| Unity3D |  | X |  |
| GitHub | X | X |  |
| SVN | X | X |  |
|  |  |  |  |
| **Systèmes et réseaux** |  |  |  |
| Adressage IP | X | X | X |
| VLAN | X | X |  |
| Modèle OSI | X |  |  |
| Linux (Debian, Ubuntu...) | X | X |  |
| Windows | X | X | X |
| Serveur Web |  | X |  |
| Serveur DNS |  | X |  |
| Serveur NAS |  | X |  |
|  |  |  |  |
| **Divers** |  |  |  |
| Photoshop |  |  |  |
|  |  |  |  |

**Savoir être**

**VI – Fiches types/squelettes**

Tous les utilisateurs disposent d’un identifiant et d’un mot de passe permettant d’accéder au contenu de l’entreprise.

L’accesseur pourra faire une recherche de compétences grâce au moteur de recherche.

Le contributeur pourra soumettre ou mettre à jour une nouvelle compétence qui sera ensuite validée par l’administrateur.

Les administrateurs sont composés des 7 membres du groupe.

Un administrateur pourra accéder au moteur de recherche, soumettre ou mettre à jour une nouvelle compétence, il devra vérifier et valider les compétences qui seront soumises.

**VII – Moyens de communications**

L’équipe de développement utilise une discussion SKYPE pour élaborer des stratégies, mettre en place le projet. Pour l’échange de fichiers entre développeurs, nous avons choisi SVN.

**VIII - Plan de communications multicanal**

La communication entre le client (M. Patrick LEFEVRE) & M. Sébastien DESZCZ, se fera par les moyens suivants :

* M. Patrick LEFEVRE
  + Mail 🡺 [info.lefevre@laposte.net](mailto:info.lefevre@laposte.net)
  + Réunion d’avancement
  + Rapport d’avancement
* M. Sébastien DESZCZ
  + Mail professionnel 🡺 [sebastien.deszcz@epsi.fr](mailto:sebastien.deszcz@epsi.fr)

Mail personnel 🡺 [sebastien.deszcz@gmail.com](mailto:sebastien.deszcz@gmail.com)

* + Skype 🡺 sebasten.deszcz
  + Téléphone 🡺 06.77.13.47.46
  + Réunion d’avancement
  + Rapport d’avancement