Projet Increase - fonctionnalités à implémenter

Espaces de noms :

Pour éviter les conflits de nommage de classe, on utilisera les espaces de noms php sur les contrôleurs et les modèles :

Type de classe	Espace de nom
Contrôleur	increase\controllers
Modèle	increase\models

Voir Phalcon et espaces de noms

Requêtes Ajax

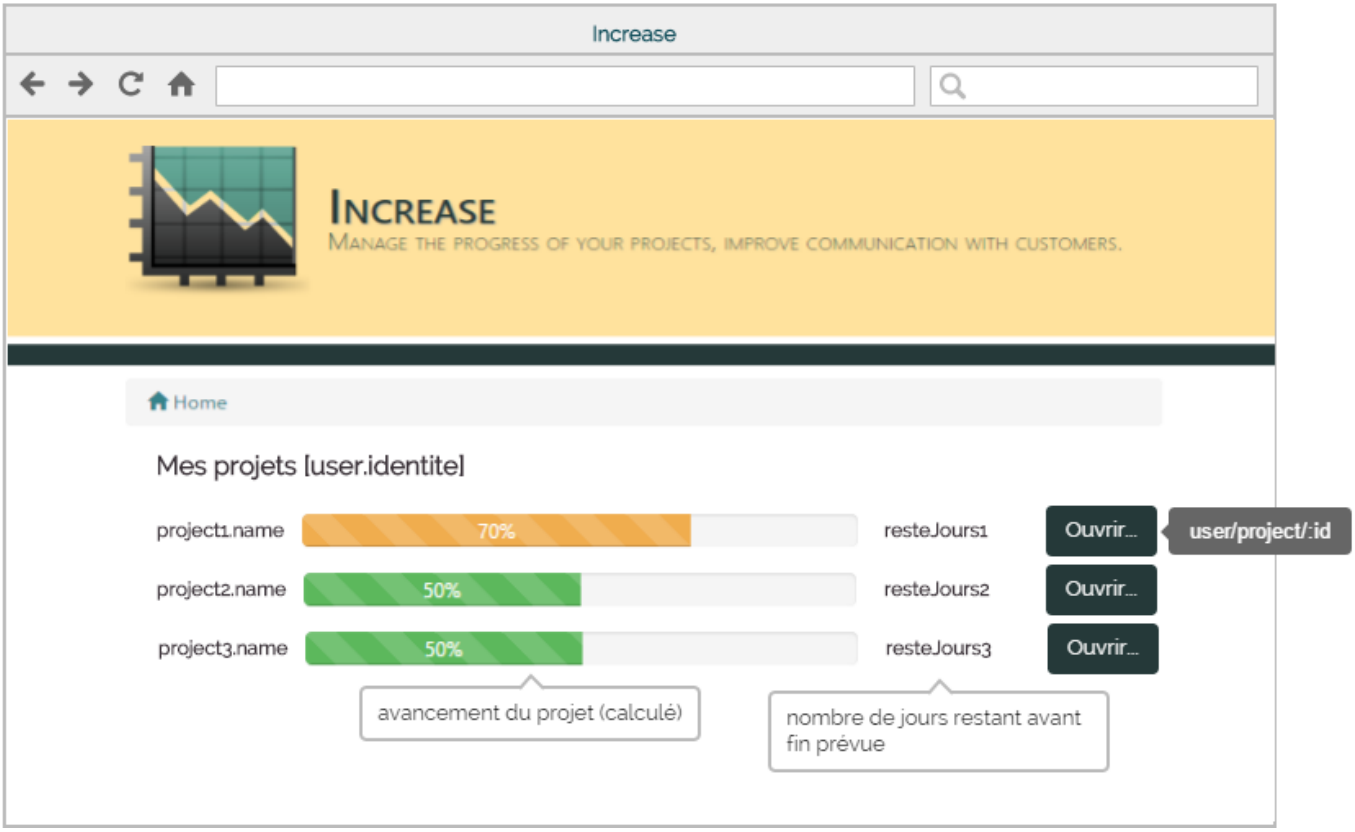
Il est indispensable d'améliorer la fluidité des pages en effectuant des requêtes partielles en Ajax.
Intégrer pour ce faire phalcon-jquery, voir également la documentation JQuery

-- Models

- Générer les modèles
- Vous ajouterez les relations qui seront nécessaires par la suite au fur et à mesure de vos besoins

-- Liste des projets d'un client

A l'adresse **user/projects/:idUser**, on souhaite afficher la liste des projets d'un client tel que celui-ci pourra la consulter (ce sera ensuite le client connecté).



-- Avancement

L'avancement est calculé de la façon suivante :

Soit le projet composé des uses cases suivants :

use case	poids	Avancement
UC-1	20	50%
UC-2	10	100%
UC-1	20	0%

On exprime le poids de chaque UC en % du total

use case	poids en %	Avancement
UC-1	40%	50%
UC-2	20%	100%
UC-1	40%	0%

- UC-1, qui représente 40% du projet ayant un avancement de 50% ⇒ il a contribué à l'avancement du projet de 40*50% soit 20%
- UC-2, qui représente 20% du projet ayant un avancement de 100% ⇒ il a contribué à l'avancement du projet de 20*100% soit 20%
- UC-1, qui représente 40% du projet ayant un avancement de 0% ⇒ il a contribué à l'avancement du projet de 40*0% soit 0%

L'avancement du projet est de 40%,
Avancement projet = somme sur les uc des (poids relatif*avancement)

-- Visualisation des retards

On considère qu'il y a retard dans l'avancement d'un projet si le pourcentage d'avancement est inférieur au pourcentage de temps écoulé.

Pour apprécier les éventuels retards ou avances dans l'avancement, on utilise les codes couleur suivants sur la progressBar d'avancement :

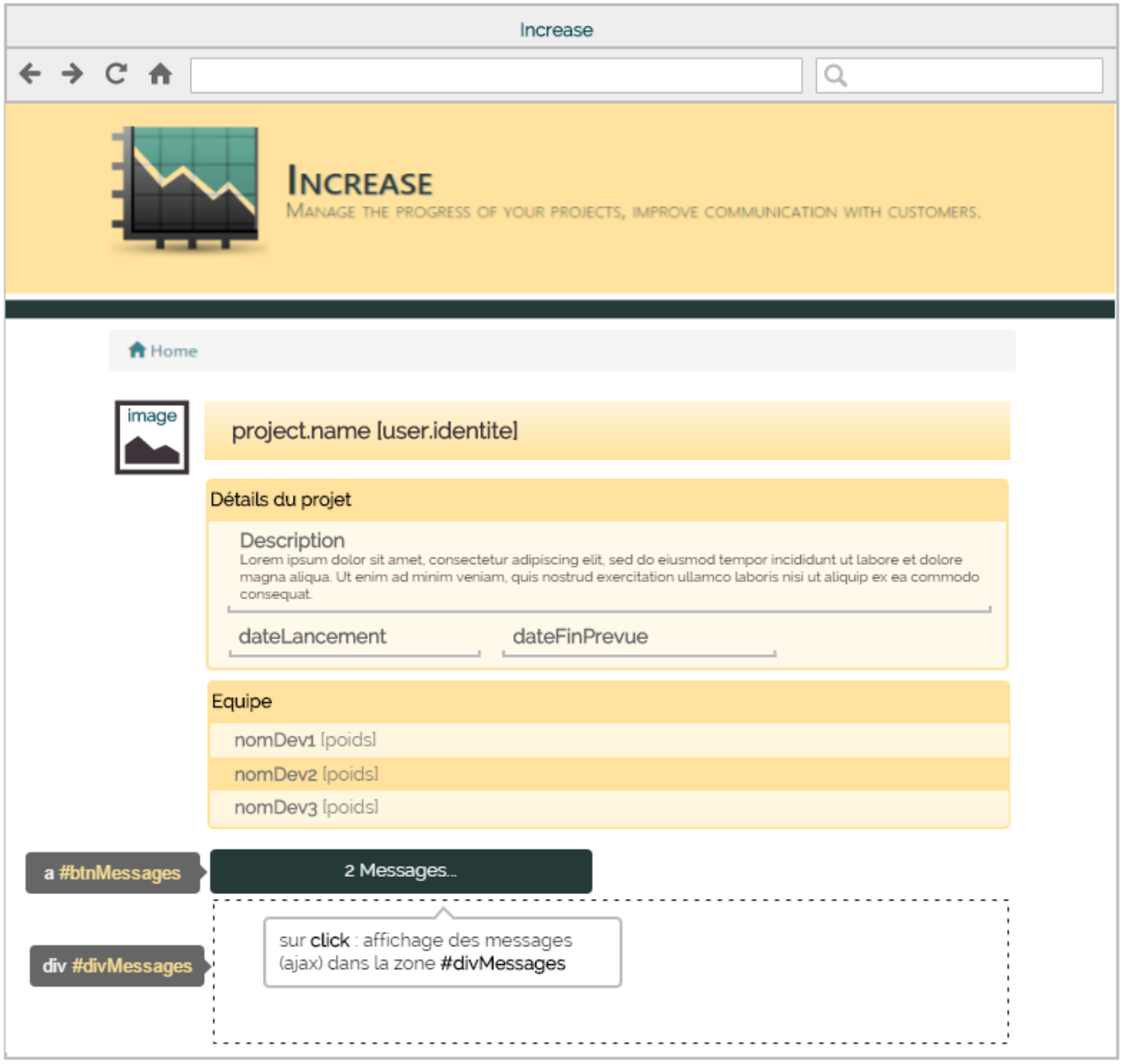
Couleur /classe	Signification
<fc #5CB85C>Vert/success</fc>	% d'avancement>= % Temps écoulé
<fc #F0AD4E>Orange/warning</fc>	% d'avancement< % Temps écoulé
<fc #D9534F>Rouge/danger</fc>	dateFinPrevue dépassée

-- Ouverture de projet

L'ouverture d'un projet par le click sur l'un des boutons **Ouvrir** doit se faire en Ajax.

-- Projet d'un client

A l'adresse **user/project/:id**, on souhaite afficher le projet dont l'id est passé en paramètre, conséquence de l'ouverture mentionnée précédemment. Toutes les informations affichées sont en lecture.



-- Equipe

La zone **Equipe** sera chargée en Ajax au chargement de la page dans une div **#detailProject**, à partir de l'adresse **project/equipe/:id** ou **id** est l'identifiant du projet.
Elle contient la liste des développeurs affectés au projet (par des cas d'utilisation). Le renseignement **poids** désigne le poids en % que représente l'intervention du dévelopeur dans le cadre du projet, compte tenu des uses cases dans lesquels il intervient, et du poids de ceux-ci.

Exemple sur un projet :

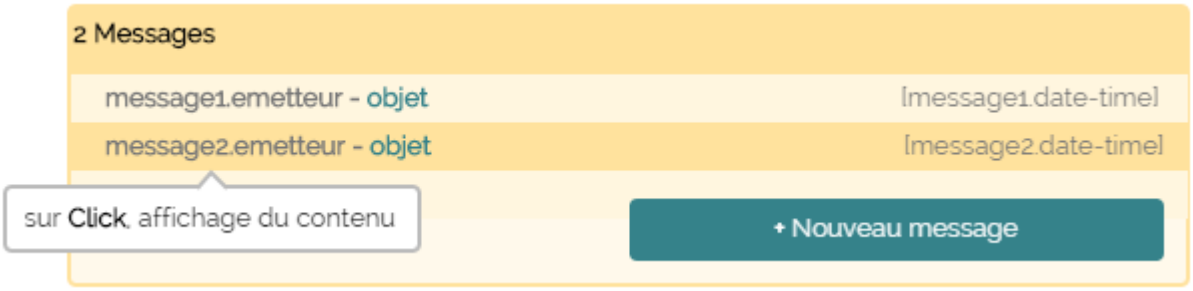
use case	poids	Développeur
UC-1	20	John DOE
UC-2	10	John DOE
UC-1	20	Martin SMITH

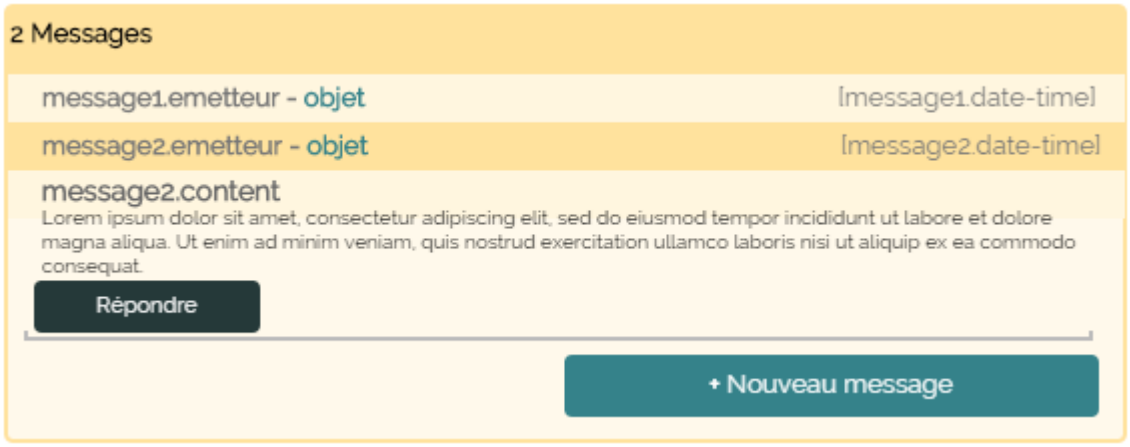
John DOE intervient sur UC-1 et UC-2 qui pésent (20+10)/(20+10+20) = 30/50 = 60%

Son poids dans le projet est de 60%, 40% pour SMITH.

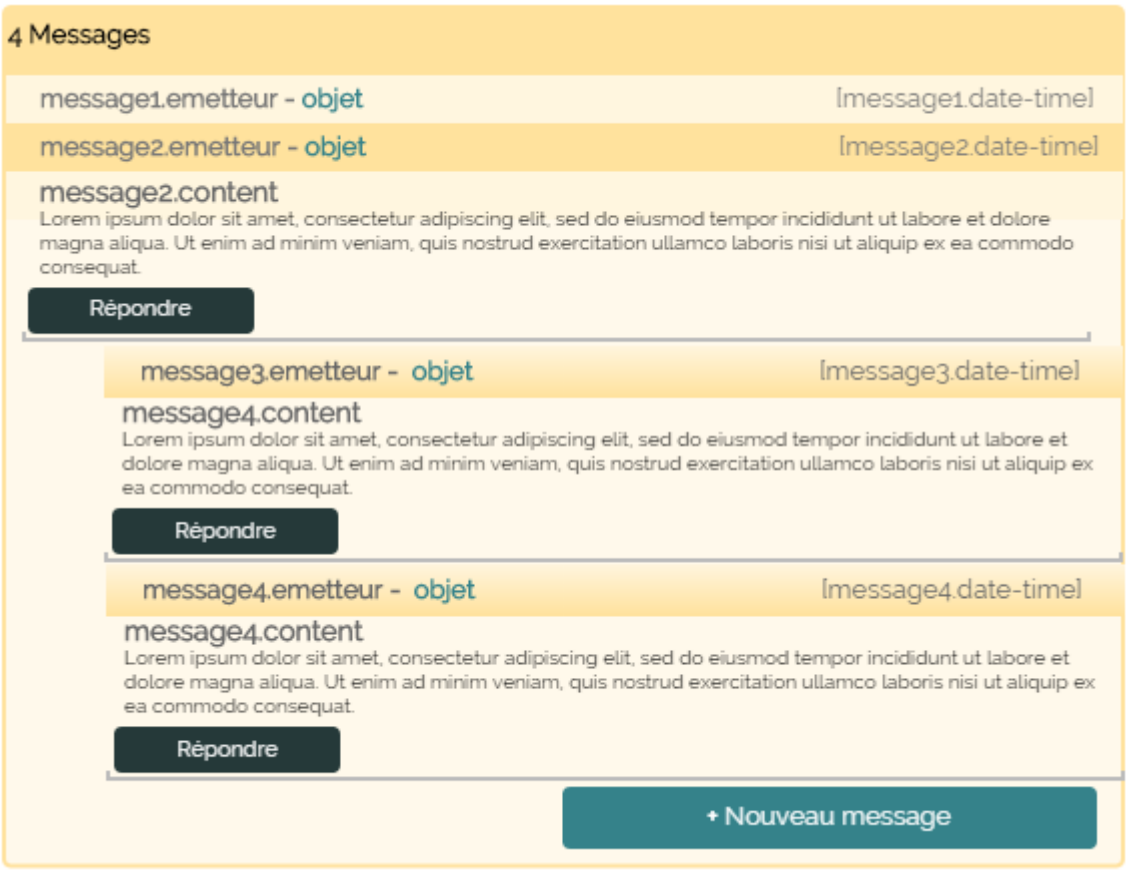
-- Message

La zone message devient visible sur click du bouton **#btnMessages**, et affiche les messages liés au projet : **project/messages/:id**





Certains messages appartiennent à un fil de discussion (ils ont été émis en réponse à un autre message ⇒ clé étrangère idFil dans la table message) :



-- Retour

Un bouton doit permettre de fermer le projet, et de retourner à la liste des projets du client (en ajax/Jquery)

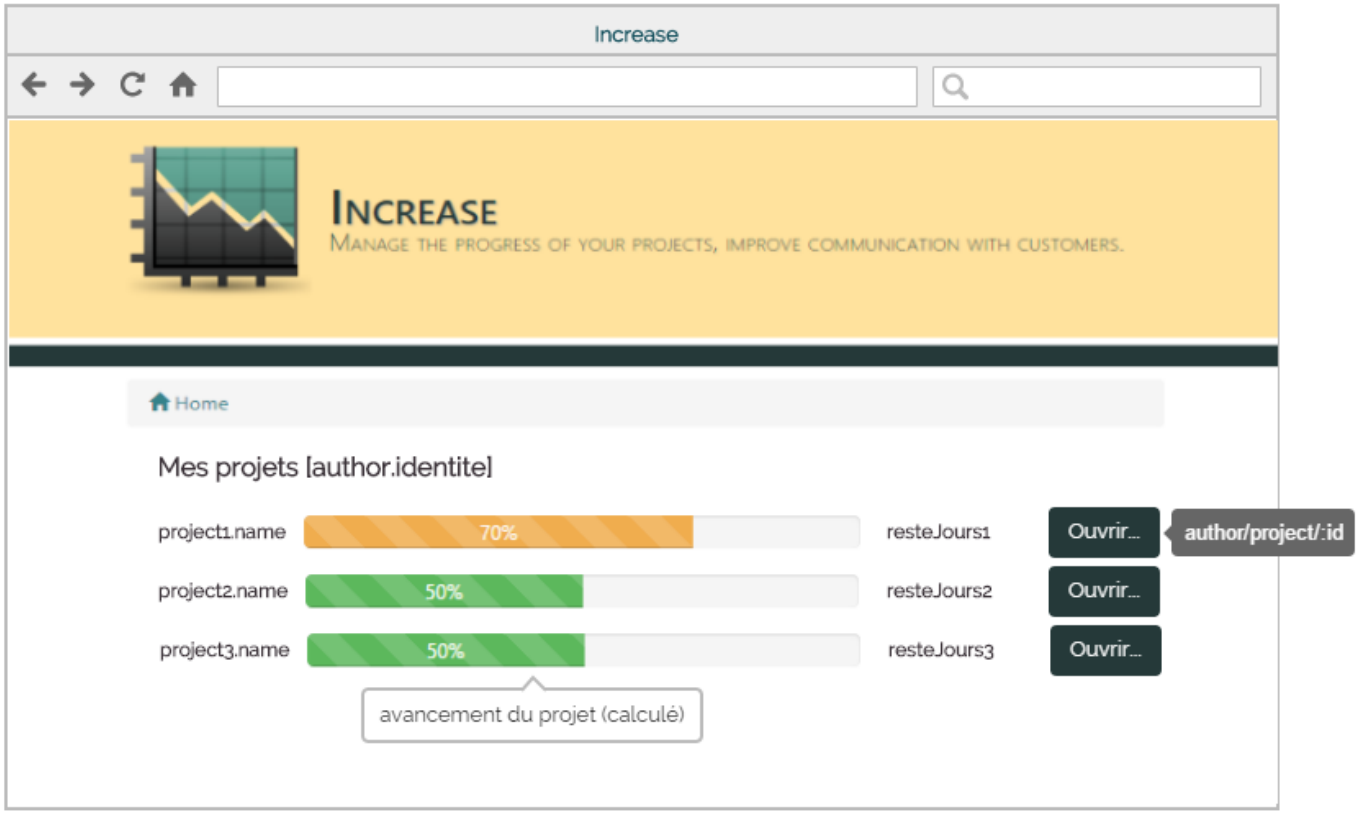
-- Liste des projets d'un "Author"

On désigne par “Author” les membres d'une équipe de développement.

A l'adresse **author/projects/:idUser**, on souhaite afficher la liste des projets auquels participe un “author”, et tel qu'il pourra la consulter (ce sera ensuite l'author connecté).

Cette liste est identique à celle visualisée par le client, à deux exceptions près :

1. Les données sont différentes, il s'agit des projets où l'author intervient, par le biais des cas d'utilisation du projet
2. Les liens Ouvrir (avec ajax) doivent pointer sur une adresse author/project/:id

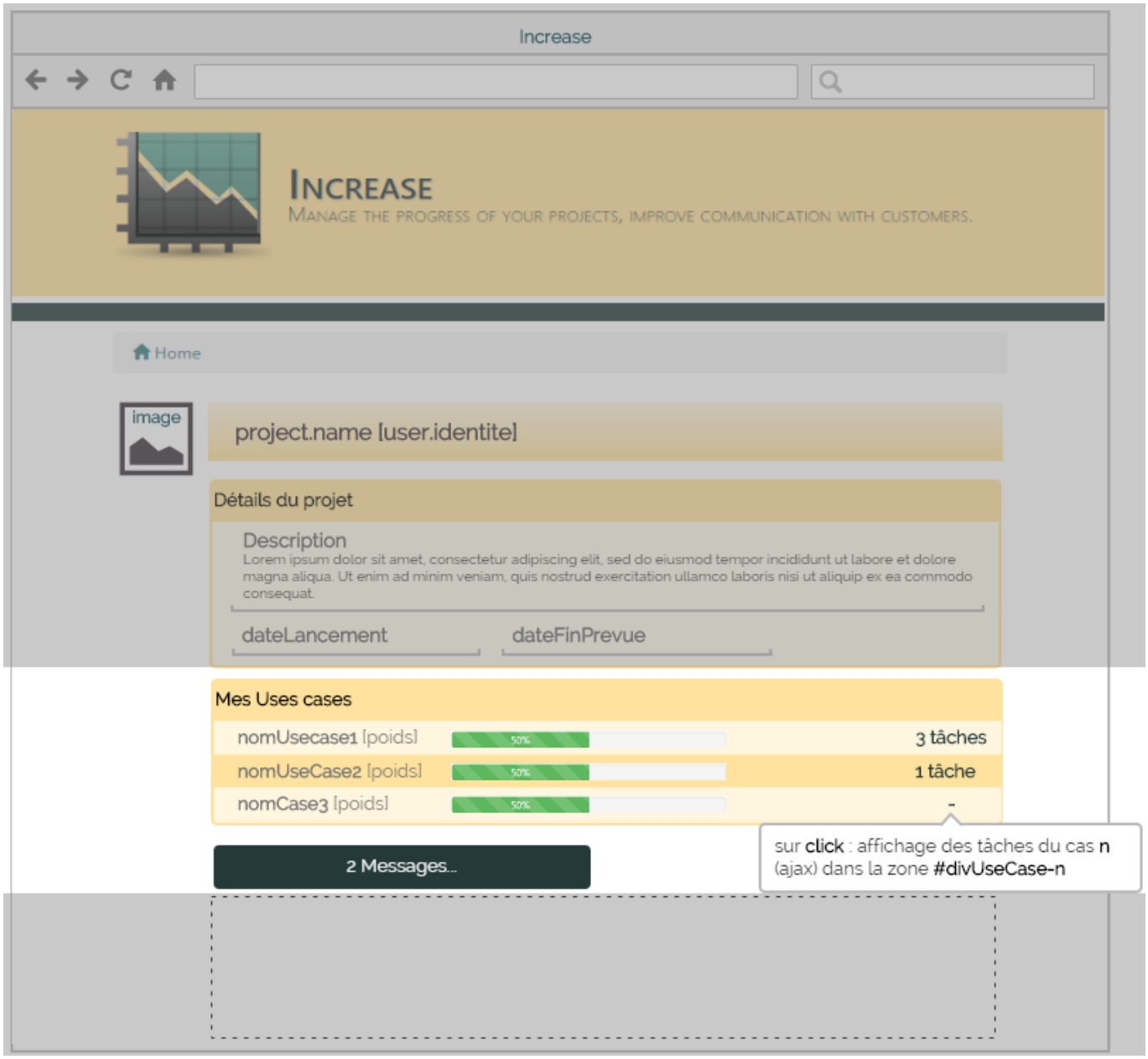


-- Projet client vu par un author

A l'adresse **author/project/:id**, on souhaite afficher le projet dont l'id est passé en paramètre. Toutes les informations affichées sont en lecture.

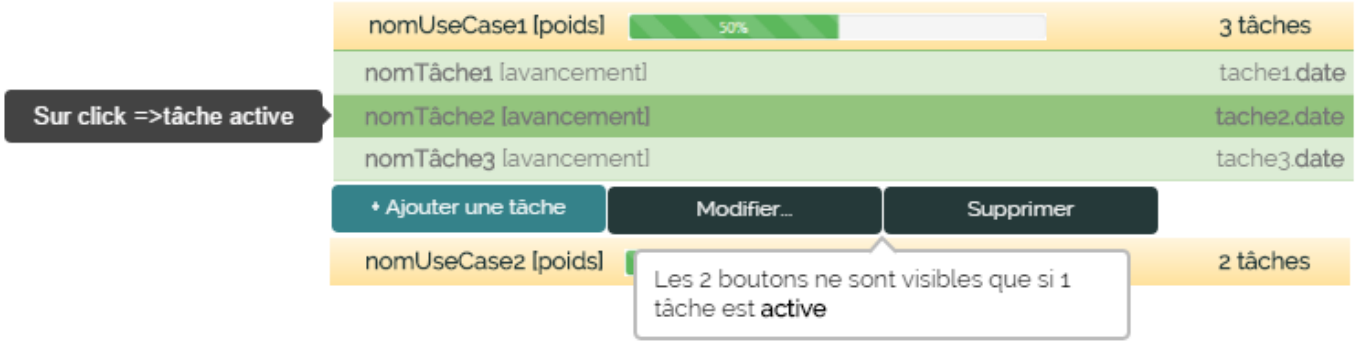
La seule partie modifiée par rapport à ce que voit le client est la zone **Equipe**, remplacée par la zone **Mes uses cases** :

La zone **Mes uses cases** sera chargée en Ajax au chargement de la page dans la div **#detailProject**, à partir de l'adresse **project/author/:idProject/:idAutor**.



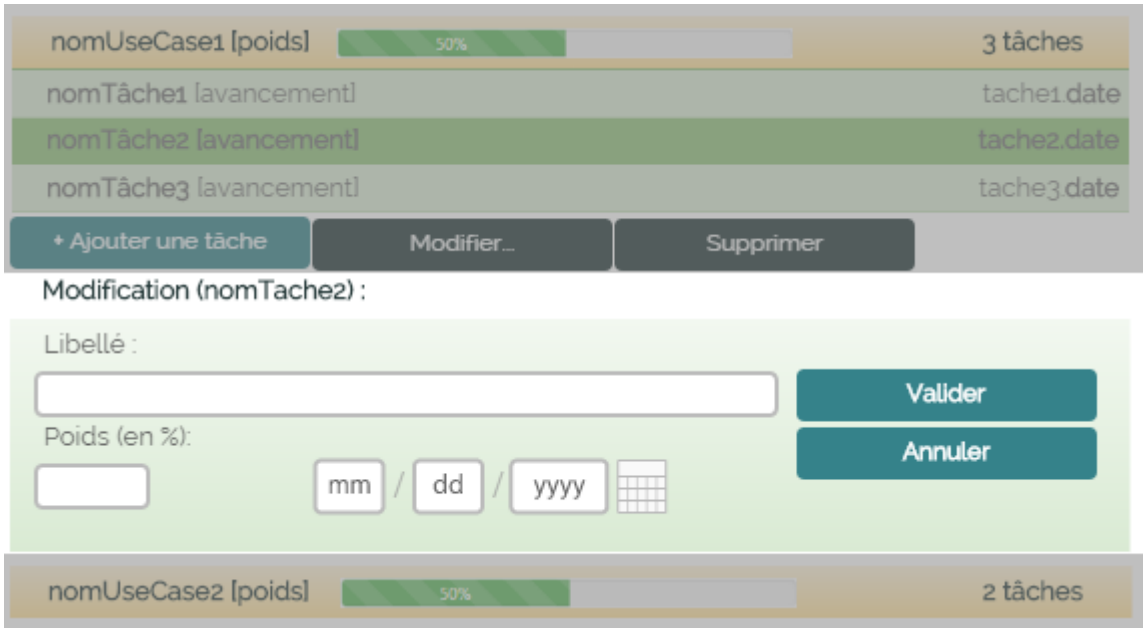
-- Liste des tâches

Les tâches apparaîssent dans la zone **#divUseCase-n** (n étant l'id du use case) à partir de l'adresse **usecase/taches**



-- Modification/ajout de tâche

La modification est obtenue sur click du bouton **Modifier...** lorsqu'une tâche est active, par appel de l'url **tache/update/:id**, id étant l'id de la tâche à modifier,
l'ajout par un click sur le bouton **Ajouter une tâche**, par appel de l'url **tache/insert**.



Sur validation, la zone correspondant au use case doit être mise à jour :

- Calcul du nouvel avancement du projet
- Affichage de la liste des tâches modifiée
- Affichage du nombre de tâches dans le cas (éventuellement mis à jour)

-- Connexion/droits

On pourra utiliser dans cette partie le projet **Invo** pour sa partie connexion (Contrôleur **SessionController**), et la gestion des ACL (**SecurityPlugin**).

-- Access list

Ajouter les ACL nécessaires dans le plugin **SecurityPlugin** :

Un user (client) doit pouvoir accéder aux Urls **user/project** ... et aux Urls associées (à vous de voir en fonction des contrôleurs et actions que vous avez créés)..\ Il ne peut accéder aux Urls **author/project** et aux Urls associées.

L'author peut pour l'instant accéder à toutes les urls.

-- Pseudo connexion

Ajouter sur la page d'accueil 3 boutons, pour simuler :

- La connexion en tant qu'user
- La connexion en tant qu'author

- La déconnexion

Les mots de passe de la base de données sont hashés en Sha256, mais ils l'ont été avec java. Il faudra donc utiliser la fonction ci-dessous pour les hasher correctement :

```
private function javaToPhpSha($str){
        $k=hash("sha256", $str,true);
        $hex_array = array();
        foreach (str_split($k) as $chr) {
            $o=ord($chr);
            if($o>127)
                $o=$o-256;
            elseif ($o<-127)
                $o=$o+256;
            $hex_array[] = sprintf("%02x", $o);
        }
        $key=implode('',$hex_array);
        return $key;
    }
```

Rôle	mail	Password
user	johndoe@kobject.net	0000
author	misko.hevery@gmail.com	123456789

La connexion doit permettre de stocker dans la session une instance de **User**

-- Post connexion

Après connexion :

- Un **user** doit accéder directement à **user/projects**, affichant les projets de l'utilisateur
- un **author** doit accéder directement à **author/projects**, affichant les projets de l'author
- La page d'accueil doit afficher dans une zone **#divActiveUser** l'identité de l'utilisateur et le bouton déconnexion :

La déconnexion affiche un message précisant que la déconnexion est effective et remet la zone **#activeUser** à :

-- Vérification des droits

Vérifiez :

1. Qu'un utilisateur **non authentifié** ne peut accéder à aucune des pages relatives aux projets
   Afficher un message “non autorisé” si c'est le cas, et proposer la connexion
2. Qu'un utilisateur authentifié n'ayant que le rôle “**user**” ne peut pas accéder aux pages relatives au rôle

"**author**"
Afficher un message "non autorisé" si c'est le cas
3. Qu'un utilisateur authentifié n'ayant que le rôle "user" ne peut accéder qu'aux informations relatives à ses propres projets
rediriger vers les pages de l'utilisateur dans ce cas
4. Qu'un utilisateur authentifié ayant le rôle "author" :
    1. peut accéder à toutes les pages relatives aux users, à condition qu'il appartienne au projet de cet utilisateur
    2. peut accéder en tant qu'author à ses projets, et puisse les modifier
    3. peut accéder en tant qu'author aux autres projets, mais sans pouvoir les modifier

-- Points bonus

Le développement étant partiel, toutes les fonctionnalités non citées peuvent être implémentée :

Pour prendre quelques exemples :

- Form connexion à l'application
- Gestion des projets/ajout/modification/Suppression
    - Gestion des uses cases d'un projet, ajout/modification/suppression
- Messages relatifs à un projet
    - Création
    - Modification
    - Suppression
    - Réponse à un message (clé étrangère idFil)

On veillera dans ce cas à bien les mentionner dans le fichier **readme.md** du projet sur gitHub

L'utilisation de **Phalcon-Jquery**, des composants Bootstrap ou de toute autre librairie utilisée avec intelligence fait partie des points bonus

-- Points malus

D'une manière générale :

- Le non respect des consignes
- Le non respect du cahier des charges

Dans la conception :

- Tout non respect du design pattern MVC :
    - Traitements ou affichage dans les models
    - Traitements ou logique applicative réalisés dans les vues (y compris avec JQuery ou javascript)

A ne pas faire :

- Utiliser des fichiers javascript externes pour y insérer une partie de la logique applicative (chargements de contrôleurs/actions)
- Utiliser JQuery ou Javascript dans les vues (avec la balise **script**)
- Utiliser **window.location…**

Dans la conduite du projet :

- Commits non réguliers ou insuffisement commentés
- Déséquilibre dans la répartition des tâches au sein du binôme

From:
http://slamwiki2.kobject.net/ - **SlamWiki 2.1**

Permanent link:
http://slamwiki2.kobject.net/slam4/php/phalcon/project/increase/todo

Last update: **2019/08/31 14:21**