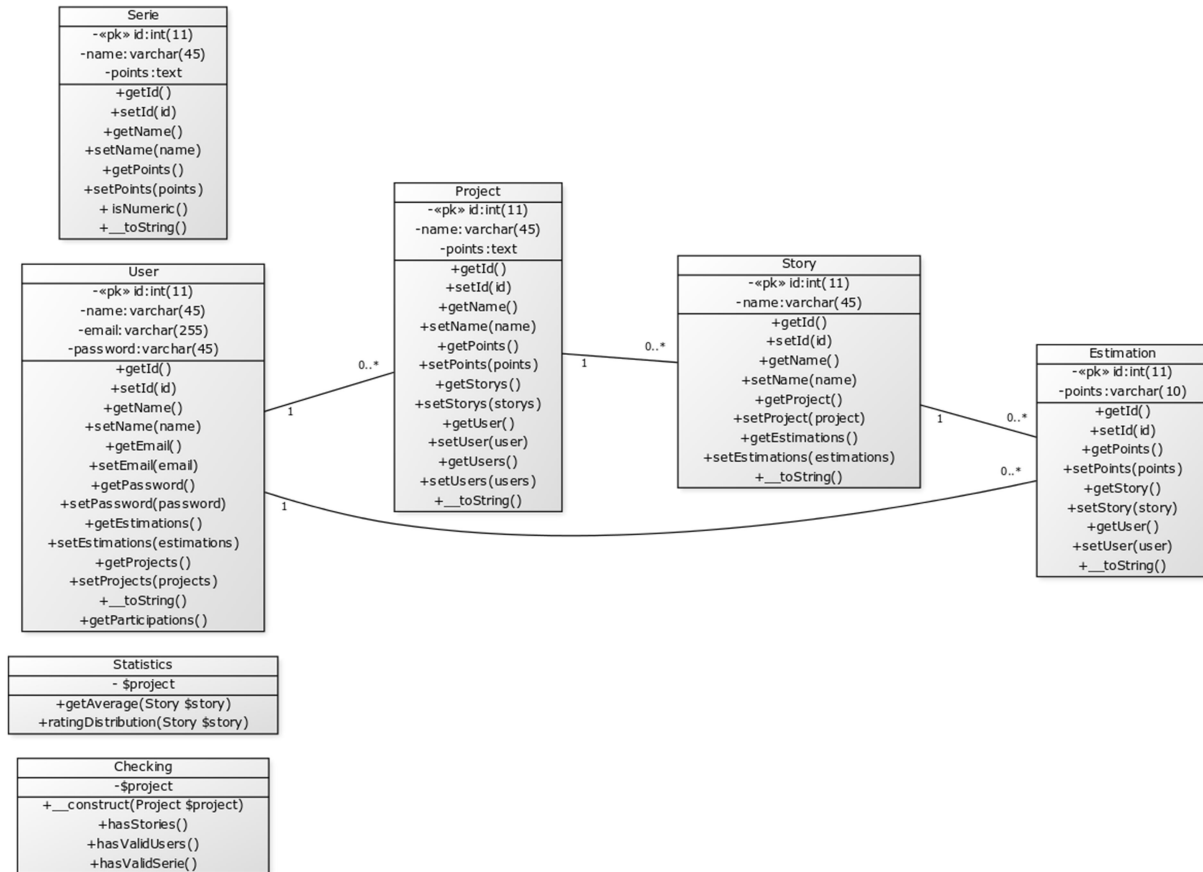


Préparation à E5	2 ^{ème} BTS SIO SLAM
SLAM4-5	J-C HERON
CONTENU : Solutions applicatives	Evaluation n°1
Durée max	1 heure 30
Domaines abordés	D4.1 - Conception et réalisation d'une solution applicative
Fichier	DS2.docx

ScrumPoker

A partir du diagramme de classes de l'application :



Implémenter les méthodes ci-dessous :

Dans la classe **Checking** :

1. Le Constructeur **__construct(Project \$project)**
//Affecte le projet passé en paramètre au projet de la classe
2. La méthode **boolean hasStorys()**
//Retourne vrai si le projet a au moins 1 story, faux dans le cas contraire
3. La méthode **boolean hasValidUsers()**
//Retourne vrai si le projet a bien été évalué par ses seuls participants, et non par un ou plusieurs membres externes à celui-ci.*
4. La méthode **boolean hasValidSerie()**
//Retourne vrai si les notes attribuées appartiennent bien à la série du projet (membre \$points)

* On pourra utiliser la méthode **findById(\$id,\$arrayOfModels)** retournant faux si aucun objet d'id \$id n'a été trouvé dans la liste des objets \$arrayOfModels, et l'instance trouvée dans le cas contraire.

Préparation à E5	2 ^{ème} BTS SIO SLAM
SLAM4-5	J-C HERON
CONTENU : Solutions applicatives	Evaluation n°2

Dans la classe Statistics :

5. La méthode **getAverage(Story \$story)**

//Retourne la moyenne des notes obtenues par une story dans les estimations. Attention, dans le cas d'une série non numérique (tailles de T-Shirt par exemple), on utilisera l'ordre des index pour calculer la moyenne :

Exemple de notes :

Notes attribuées	Index correspondant
XS	0
M	2
L	3
S	1
S	1

Moyenne des index = $(0+2+3+1+1)/5$ notes = $7/5 = 1,4$

1,4 arrondi à l'entier le plus proche est 2, ce qui correspond au M, valeur retenue de la moyenne pour ces notes

6. La méthode **ratingDistribution(Story \$story)**

//Retourne un tableau associatif composé du nombre de chacune des notes attribuées dans les estimations de la story, la note étant la clé, et le nombre de fois où cette note a été attribuée la valeur.

Dans l'exemple précédent, la fonction doit retourner le tableau ['xs'=>1,'S'=>2,'M'=>1,'L'=>1]

Remarques

1. Les classes **Checking** et **Statistics** contiennent une instance **\$project**, dont les membres sont supposés être tous correctement affectés (le projet a des stories, des users...)
2. Toutes les collections d'objets sont des tableaux php

Manipulation des tableaux

Pour créer un tableau : `$unTableau = [] ;`

Pour parcourir par itération tous les éléments d'un tableau :

```
foreach($tableau as $element){
    // instructions avec $element
}
```

Pour parcourir par itération tous les éléments d'un tableau associatif :

```
foreach($tableau as $key=>$element){
    // instructions avec $element ou $key
}
```

Ajout en fin de tableau d'un élément : `$unTableau []= $newElement ;`

Ajout d'un élément à une position (clé ou index) : `$unTableau [$k]= $newElm ;`

Taille d'un tableau : `$nb=sizeof($unTableau) ;`

Retourne un tableau de chaînes de caractères créées en découpant la chaîne du paramètre string en plusieurs morceaux suivant le paramètre delimiter.

`$tableau= explode($delimiter,$string) ;`

Arrondi un nombre à l'entier approchant : `$nbEntier=round($unFloat) ;`

Yuml :

[[[...]]]]