

# Pratique des Techniques Informatiques

## BTI IG Développeur d'Application

Session 2010-2012

Romain BELLINA

### FICHE DE SYNTHESE n°2

## « Création de cocktail »

### OBJECTIF DE L'ACTIVITE

- ♦ Création de classes à partir d'un diagramme de classe,
- ♦ Application des concepts d'héritage,
- ♦ Développer une application utilisant des tableaux d'objets,
- ♦ Chargement d'une interface graphique en PHP.

### SUPPORT DE L'ACTIVITE

Lieux de réalisation	Outils utilisés
♦ Mac OS X	♦ PHP ♦ JavaScript

### COMPETENCES CONCERNEES

- C32 ☒ Développer à l'aide d'un langage de programmation procédural.  
 C33 ☒ Maquetter une application, la développer à l'aide d'un langage de programmation événementielle.  
 C34 ☒ Développer à l'aide d'un langage de programmation objet.  
 C35 ☒ Développer autour d'une base de données relationnelles.  
 C36 ☒ Développer dans le cadre d'une architecture client-serveur.

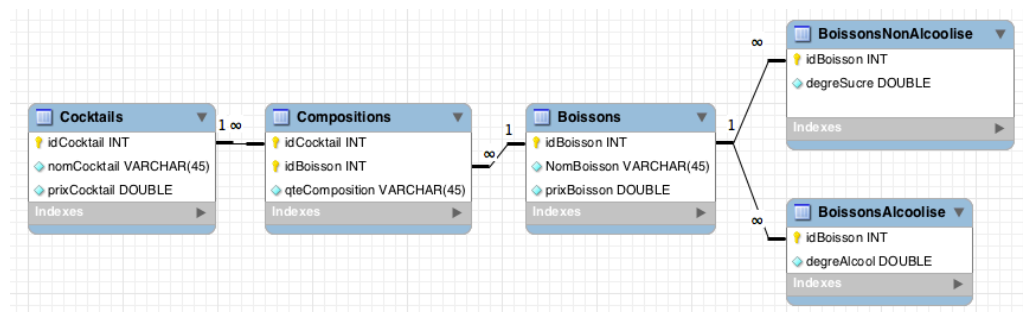
## Présentation de l'activité :

« Cocktail 3000 » est une application codée en PHP avec une interface graphique. Elle permet la création de boisson, alcoolisé ou non, pour ensuite créer des cocktails. L'application permet aussi la modification et la suppression des boissons ou cocktails. Les boissons et les cocktails sont ensuite stockés sur un serveur SQL.

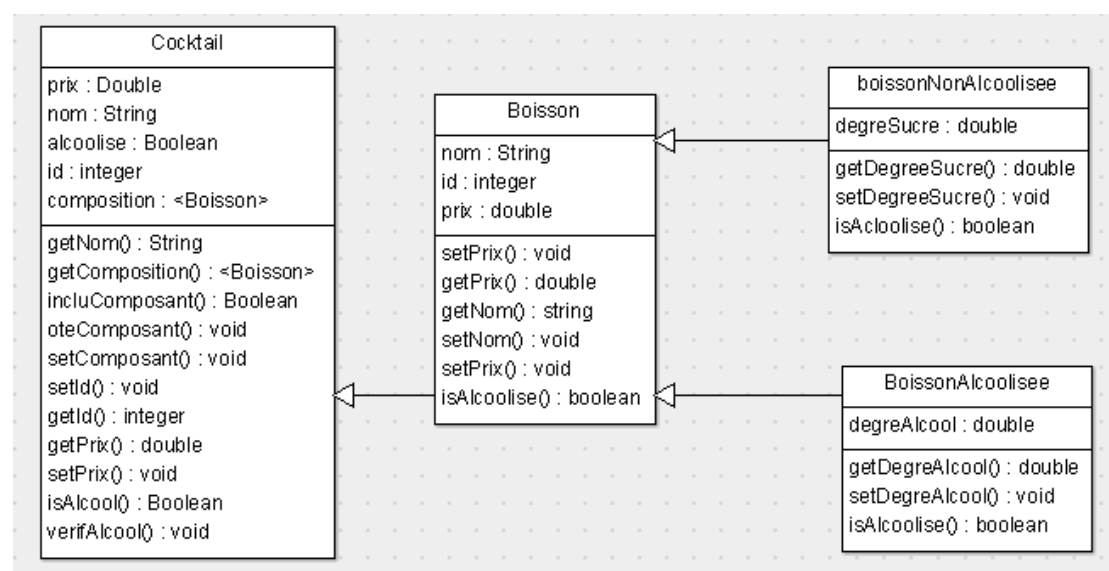
Au lancement, l'application se connecte au serveur et charge la totalité des boissons et cocktails dans des tableaux de données.

Lors de la création d'une boisson ou d'un cocktail, il a été choisi de l'insérer dans la base de données puis par la suite de recharger l'intégralité des données pour mettre à jour les tableaux

## Modèle conceptuel des Données



## Diagramme de classe :



# Héritage et Polymorphisme :

---

L'un des aspects important de ce logiciel réside dans l'héritage d'une classe par rapport à une autre.

Ici nous avons 3 classes :

- 'Boissons'
- 'BoissonAlcoolise'
- 'BoissonNonAlcoolise'

C'est trois classes sont lié entre elle. En effet il existe un héritage entre ces classes. C'est à dire que les attributs et méthodes de la classe mère 'Boissons' sont transférés aux classes filles 'BoissonNonAlcoolise' et 'BoissonElcoolise'.

Ainsi, la méthode 'getNom()' sur un objet 'BoissonAlcoolise', retournera le nom de la boisson.

Avec l'héritage, une autre notion est importante, il s'agit du polymorphisme. Il consiste à faire appel au constructeur de la classe mère dans la classe fille.

Comme dans l'exemple précédent, j'ai appelé le constructeur de la classe 'BoissonsNonAlcoolise' et passé des arguments qui sont le nom, le prix, l'id et le pourcentage de sucre.

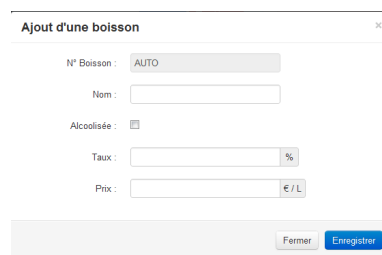
```
public function __construct($id, $nom = "", $prix = 0, $qte=0, $degreSucre=0)
{
    parent::__construct($id,$nom,$prix,$qte);
    $this->setDegreeSucre($degreSucre);
}
```

Le polymorphisme va appeler le constructeur de la classe 'Boissons' grâce a la fonction `__construct()`; et passera les argument nécessaire pour créer l'objet 'Boissons'.

```
public function __construct($id, $nom = "", $prix = "") {
    $this->prix = $prix;
    $this->nom = $nom;
    $this->id = $id;
}
```

## Aspect fonctionnel et technique :

---



L'ajout d'une boisson ce fait par l'intermédiaire d'une interface graphique. L'utilisateur saisi les informations nécessaires à la création d'une boisson et click sur le bouton 'Enregistrer'. Lors de l'appui sur le bouton, nous appelons une fonction JavaScript qui va envoyer en Ajax les informations récupérées dans les champs. Ainsi nous créons deux requete SQL

```
INSERT INTO Boissons(NomBoisson,prixBoisson) VALUE('nom',prix) ;
```

```
INSERT INTO BoissonsAlcoolise(idBoisson,degreAlcool) VALUE(id,degre);
```

Une fois les boissons ajoutées, nous pouvons créer un cocktail.

Dans cette interface, l'utilisateur choisie les composants de son cocktail. A l'aide du menu déroulant, il ajoute le composant sélectionné dans la liste à son cocktail. Il doit aussi indiquer la quantité de boisson qu'il désire. Si le composant est déjà présent dans la liste, la quantité sera ajoutée à la quantité déjà présente.

L'utilisateur doit dans un premier temps créer le cocktail. Après avoir saisie le nom et le prix, il doit appuyer sur le bouton 'Enregistrer' pour déclencher une fonction JavaScript et créer la requête d'insertion dans la base de données

```
INSERT INTO Cocktails(nomCocktail,prixCocktail) VALUES('nom',prix) ;
```

L'appui sur le bouton '+' crée les requêtes d'insertion dans la base de données des boissons à partir des informations présentes, par le biais d'une fonction JavaScript et d'une requête Ajax.

Chaque appuis sur le bouton '+' déclenche une fonction JavaScript qui ajoute ou met à jour le composant sélectionné dans le menu déroulant en fonction des composant déjà présent dans le cocktail.

Si le composant existe déjà, alors on effectu un update, sinon un insert dans la table Compositions

```
UPDATE Compositions SET qteComposition = (qteComposition + $volume)
WHERE idBoisson = $idBoisson AND idCocktail = $idCocktail;
```

```
INSERT INTO Compositions (idBoisson,idCocktail,qteComposition)
VALUES('$idBoisson','$idCocktail', $volume);
```

La page d'accueil récupère tous les cocktails présents dans la base et affiche leurs informations.

Liste des cocktails				
N°	Nom Cocktail	Alcoolisé	Prix	Actions
1	Mojito	✓	4.65 €	<button>Modifier</button> <button>Supprimer</button>
2	Black Jack	✓	4 €	<button>Modifier</button> <button>Supprimer</button>
3	Martini Gin	✓	4 €	<button>Modifier</button> <button>Supprimer</button>

Rafraichir + Créer

## Conclusion :

---

L'objet a toujours été mon talon d'Achille, mais grâce au développement de cette application, j'ai compris les fondements de la programmation orienté objet comme l'héritage, et le polymorphisme. L'utilisation des méthodes de classe permet de faire des actions relativement facilement tout en gardant un code clair et simple.

Cette application m'a aussi permis de faire le lien entre le langage web et le Java qui peut paraître très éloigné mais qui se confond finalement.