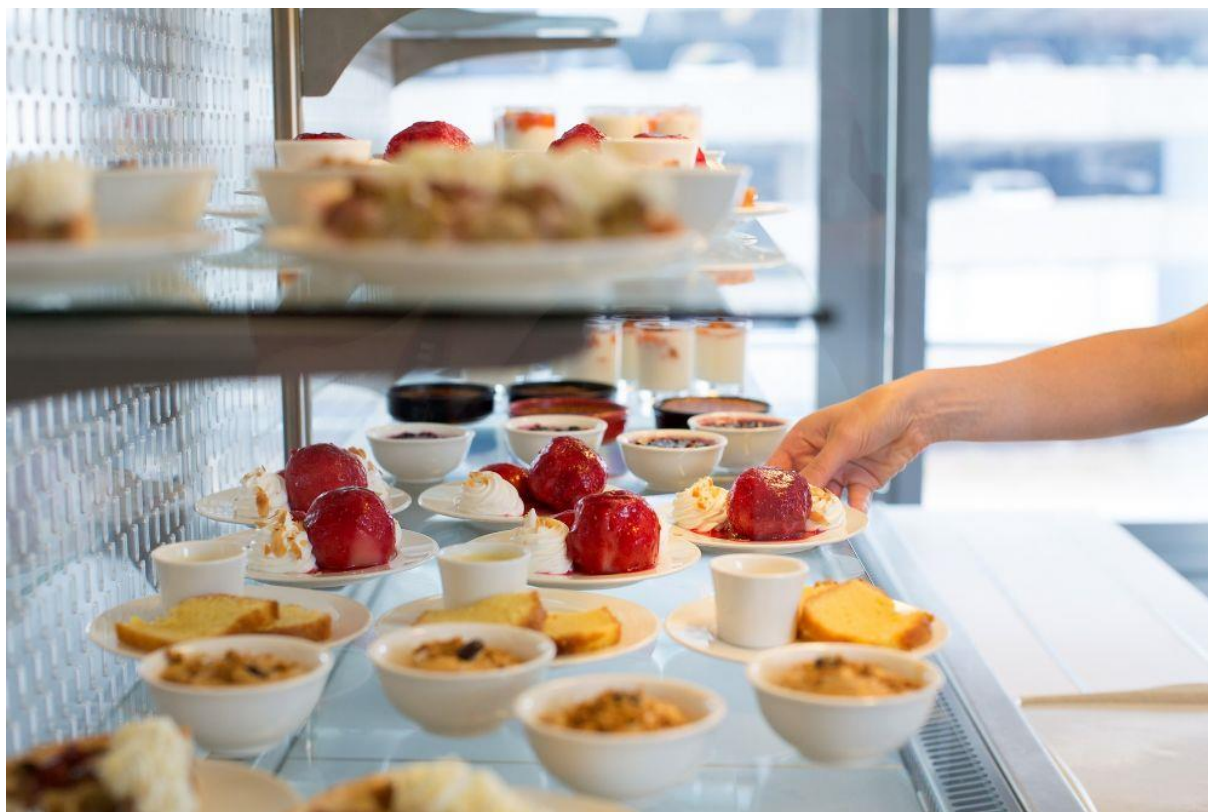


26 JUIN 2020



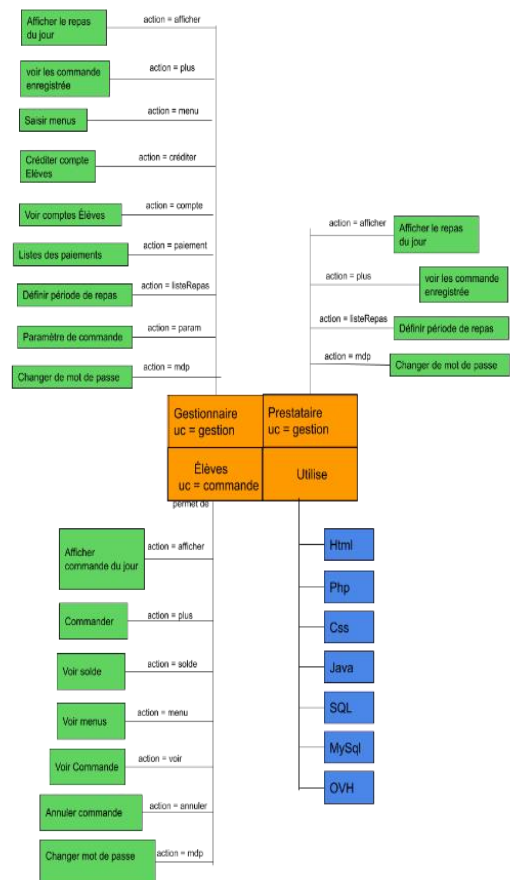
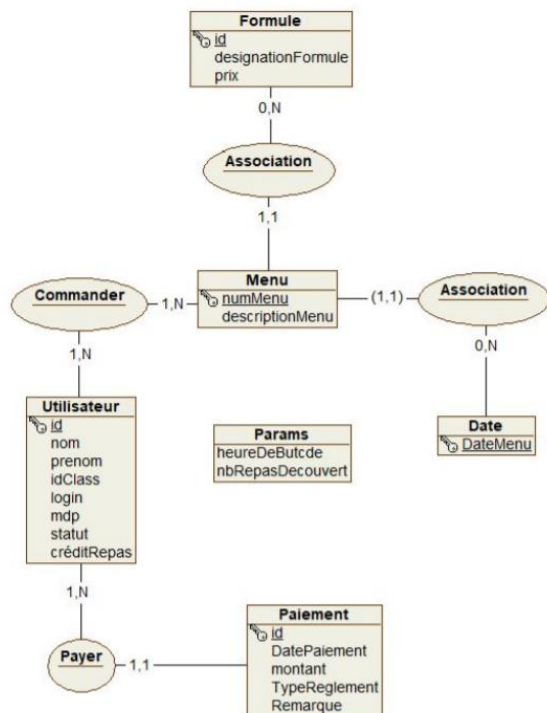
VALIDATION DE COMPETENCES

PPE_REPAS

THIBAUT MESLIN
SAINT ADJUTOR
Vernon

A4.1.3 Conception ou adaptation d'une base de données

C4.1.3.1 Modéliser le schéma de données nécessaire à la mise en place de la solution applicative



C4.1.3.2 Implémenter le schéma de données dans un SGBD



Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> elior_commande	☆ Parcourir Structure Rechercher Insérer Vider Supprimer	59	InnoDB	latin1_swedish_ci	16 kio	-
<input type="checkbox"/> elior_formule	☆ Parcourir Structure Rechercher Insérer Vider Supprimer	6	InnoDB	latin1_swedish_ci	16 kio	-
<input type="checkbox"/> elior_menu	☆ Parcourir Structure Rechercher Insérer Vider Supprimer	190	InnoDB	latin1_swedish_ci	64 kio	-
<input type="checkbox"/> elior_paiement	☆ Parcourir Structure Rechercher Insérer Vider Supprimer	16	InnoDB	latin1_swedish_ci	16 kio	-
<input type="checkbox"/> elior_params	☆ Parcourir Structure Rechercher Insérer Vider Supprimer	1	InnoDB	latin1_swedish_ci	16 kio	-
<input type="checkbox"/> elior_utilisateur	☆ Parcourir Structure Rechercher Insérer Vider Supprimer	62	MyISAM	latin1_swedish_ci	6,6 kio	88 o
6 tables	Somme	334	InnoDB	latin1_swedish_ci	134,6 kio	88 o



C4.1.3.3 Programmer des éléments de la solution applicative dans le langage d'un SGBD

Au cours de ce projet nous avons dû programmer des requêtes sql afin de récupérer des informations dans la base de données. On peut par exemple prendre l'exemple de la fonction *getFormulePrix()* qui permet de récupérer le prix d'une formule en fonction de sa désignation.

```
function getFormulePrix($designationFormule){  
    require "connexion.php";  
    $sql = "select * from elior_formule where designationFormule = '$designationFormule'";  
    $exec=$bdd->prepare($sql) ;  
    $exec->execute() ;  
    $curseur=$exec->fetch();  
    return $curseur;  
}
```

C4.1.3.4 Manipuler les données liées à la solution applicative à travers un langage de requête

On peut reprendre l'exemple au-dessus. En effet, on utilise le paramètre *designationFormule* qui correspond à la description de la formule inscrite dans la base de données. On compare donc celle entrée en paramètre et celle(s) de la base de données pour trouver ou non une correspondance.

```
function getFormulePrix($designationFormule){  
    require "connexion.php";  
    $sql = "select * from elior_formule where designationFormule = '$designationFormule'";  
    $exec=$bdd->prepare($sql) ;  
    $exec->execute() ;  
    $curseur=$exec->fetch();  
    return $curseur;  
}
```

A4.1.7 Développement, utilisation ou adaptation de composants logiciels

C4.1.7.1 Développer les éléments d'une solution

Lorsque cette application nous a été fournie, elle était incomplète. Afin de la rendre fonctionnelle nous avons bien sûr dû coder de nouvelles choses et implanter de nouvelles fonctionnalités. Nous avons également été amenés à modifier le code car celui-ci comprenait des erreurs ou nous devions simplement rajouter des choses dans ces endroits. Le fichier le plus modifié a été *gestion_bdd.php* car nous avons ajouté beaucoup de nouvelles fonctions afin de rendre fonctionnelles les codes que nous faisions.



C4.1.7.3 Analyser et modifier le code d'un composant logiciel

Comme dit au-dessus nous avons été amenés à modifier des codes existants. Par exemple, j'ai personnellement dû modifier le code de `v_commande_saisie.php` afin d'ajouter à l'affichage des menus : la formule dans laquelle il était répertorié et le prix de cette formule.

```
<?php
if ($nbMenus == 1) $checked = " checked " ; else $checked = "" ;
foreach ($lesMenus as $leMenu)
{
    echo '<input type="radio" name="numMenu" value="'. $leMenu['numMenu']. '"'.$checked.'>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;' .
        $leMenu['designationFormule']. ' : &nbsp;&nbsp;'. $leMenu['descriptionMenu']. '&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;'. $leMenu['Prix']. '&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<br />' ;
}

```

A4.1.8 Réalisation des tests nécessaires à la validation d'éléments adaptés ou développés

C4.1.8.1 Élaborer et réaliser des tests unitaires

Afin de tester les fonctionnalités modifiées ou ajoutés, nous avons dû faire des tests souvent concluants mais parfois montrant une erreur et là est l'utilité de ceux-ci. Un exemple de test de ma partie consiste à valider mon affichage ci-dessus en chargeant la page associée dans l'application.

MESLIN Thibaut
Solde = 100 €
Commande repas du 26/06/2020

- Plat + Dessert : gratin de macaronis au jambon - macaronis / mousse au cacao 4.50 €
- Plat + Dessert : filet de lieu sauce crème - macaronis / neige aux myrtilles (blanc d'oeuf, crème anglaise aux myrtilles) 4.50 €
 - Plat + Dessert : haut de cuisse de poulet - macaronis / neige aux myrtilles 4.50 €
 - Entrée + Plat : carottes rapées + andouillettes - lentilles 4.50 €
 - Entrée + Plat + Dessert : melon / moules - frites / tiramisu 5.25 €
 - Plat : coquillettes sauces harissa 3.50 €
 - Restauration rapide : Burito - salade 3.00 €
- Super Plat + Super dessert : Bœuf de Kobe - caviar - patates douces / Pièce montée 90.99 €

[Valider](#)

