

Rapport de codage de DIET par taskForce

Objectifs de codage

Coder le cas d'utilisation « confectionner un menu sans contrainte ».
Plus précisément, cela consiste à gérer des aliments, et à mettre en place un indicateur d'équilibre du menu grâce aux caractéristiques des aliments (calories, glucides, protides, lipides).

Résultats

Solution technique choisie

Il n'y avait pas langage imposé dans le dossier de conception. Nous avons donc choisi d'utiliser le PHP, langage orienté Web et très pratique pour la manipulation d'une petite base de données. En effet, nous avons choisi d'utiliser une base de données MySQL comme solution de stockage des données, plus sûr et plus pratique qu'en mode raw dans un fichier.

Objectifs atteints

Il est possible de dire que la partie du logiciel à coder est terminée car le cas d'utilisation est parfaitement fonctionnel. Affichage des menus et aliments, suppression. Un besoin décrit dans le dossier d'analyse a même été réalisé (sortie formatée sur feuille A4 des menus pour l'impression). En effet il s'agissait d'un point intéressant au niveau technique pour les codeurs ; il a donc été mis en œuvre.

Méthode

Application des consignes de codage

Le PHP permettant le codage par classes, nous avons appliqué les classes citées dans le DSL. Un diagramme de classes participantes du cas d'utilisation à coder nous a été fourni. Dans celui-ci une classe semblant inutile, et la table qui lui était associée dans la BDD également. J'ai fait part de cette remarque au groupe chargé de la conception ; ils m'ont précisé que cette classe était nécessaire pour le fonctionnement du logiciel global, mais certes inutile dans le cas codé. Nous avons donc codé ce qui était prévu, si jamais quelqu'un désirait continuer le codage, sans avoir à retoucher le code.

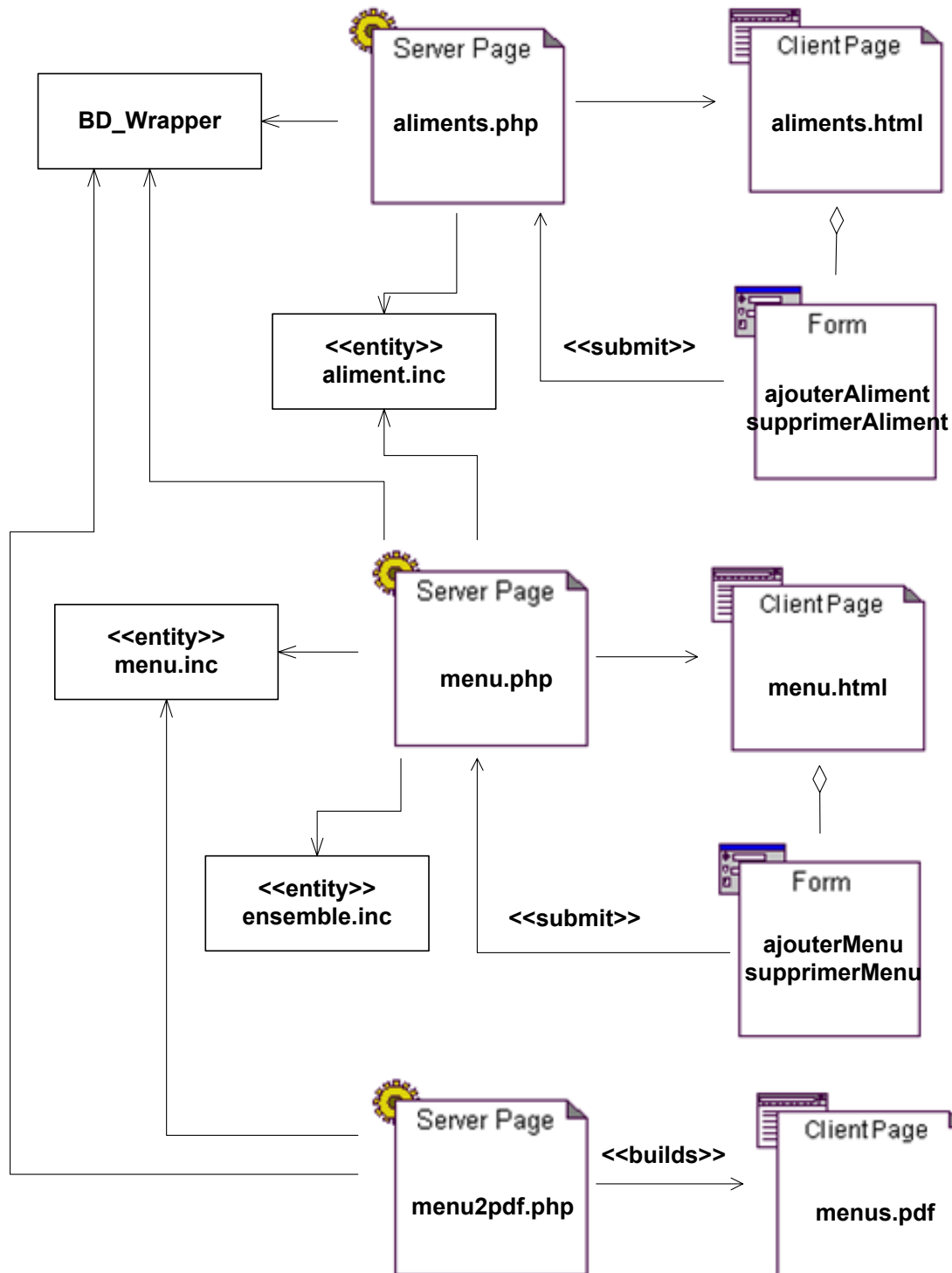
Utilisation de CVS

CVS a été utilisé pour le développement de DIET. Cependant le nombre de commit a été assez limité (les versions les plus élevées sont à 1.8 environ). Le nombre de codeurs a également été limité, cela par commodité de répartition de la charge de travail.

Stéréotype de CONALLEN

Le codage en PHP étant orienté objet, et non procédural, il a été facile d'appliquer le stéréotype de CONALLEN. En page suivante vous trouverez une figure décrivant la transformation.

Stéréotype de CONALLEN



Conclusion

L'objectif est atteint, et les acquis suite à ce projet sont nombreux, sans qu'un codage d'un projet entier ait été nécessaire.

Rapport de codage de SUIVI par le TT7 (Blizzard)

Objectifs de codage

Coder le cas d'utilisation « Création et visualisation d'un projet de 2 parties composées de 2 tâches par le chef de projet. ». Il n'était pas demandé de pouvoir exporter une sauvegarde du projet pour constituer un historique.

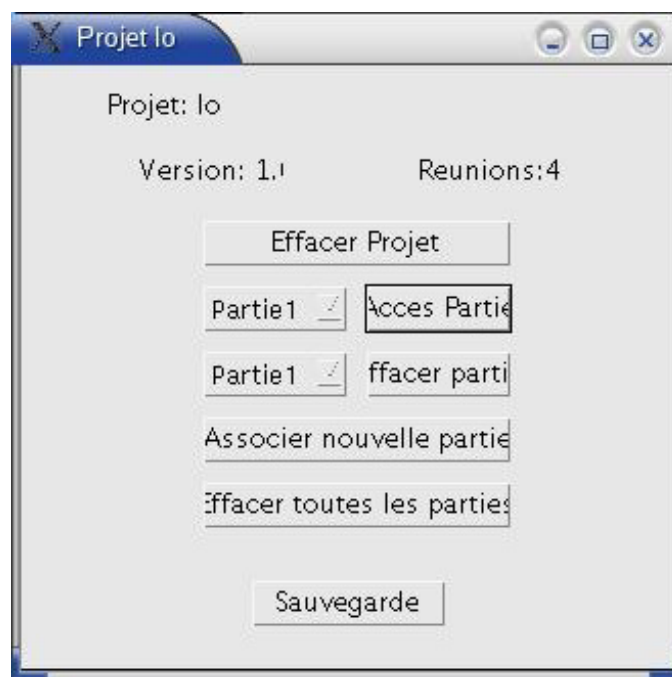
Résultats

Solution technique choisie

Les Blizzard (TT7) ont décidé de coder leur projet en JAVA. Certes, ce langage est très pratique au niveau de la programmation des classes mais son inconvénient majeur est que l'interface graphique est complexe à réaliser. Nous ne pourrions donc pas juger le logiciel sur l'interface actuelle.

Objectifs atteints

Au niveau des fonctionnalités offertes, nous avons une boîte de dialogue proposant de se logger au démarrage du logiciel. Ensuite nous arrivons sur un écran de dialogue permettant de faire notre choix entre la création d'un projet ou l'accès à notre projet, s'il est déjà créé. Choisissons notre projet, et voici l'écran suivant :



Il existe quelques bugs graphiques mineurs selon le compilateur employé. Cependant l'interface graphique n'était pas l'objet principal du logiciel. Le bouton « sauvegarde » n'est pas fonctionnel. En effet, la sauvegarde de données en JAVA peut être faite en mode raw dans des fichiers ou par interface ODBC pour utiliser une base de données. Cependant les codeurs ayant dépassé leurs quota de codage, nous avons préféré alléger.

Nous pouvons donc créer nos parties, et ensuite accéder à l'interface permettant de créer des tâches. La voici :



Le seul regret que nous puissions avoir est l'impossibilité de quitter cette boîte de dialogue. Si nous la fermons l'exécution du programme est arrêtée. Nous avons créé une tâche dont le nombre d'heures estimées est de 300. Ensuite un développeur viendra modifier son avancement comme suit :



Il a effectué 50 heures, indiquées ici, avec un calcul automatique du pourcentage d'avancement. Les champs à remplir auraient pu comporter un petit label, indiquant si la saisie doit être textuelle ou numérique. Dans le cas d'un logiciel réel, des tests devraient être effectués sur les données entrées.

Conclusion

Le codage réalisé est fidèle aux instructions que nous avons données. Cependant le choix du langage est limitant à mon goût, du fait que la création graphique est demandeuse en temps. Bravo et merci au Blizzard pour ce logiciel, somme tout semblable à l'idée que nous en avions.