

PROJET DEVELOPPEMENT APPLICATIONS RELATIONNELLES

WHY NOTES BRASS BAND

<http://php-wnbb.rhcloud.com>

LABAT Coline

Informatique et Gestion 3^{ème} année
Polytech Montpellier

Année Scolaire 2014/2015

REMERCIEMENTS

Je souhaite remercier mes camarades d'Informatique et Gestion 3^{ème} année pour leurs encouragements, leurs soutiens et leurs aides, ainsi que les étudiants d'Informatique et Gestion 4^{ème} année qui m'ont rassurée.

Table des matières

| | |
|--|-----------|
| Partie 1 – Définition de l'application | 5 |
| 1. Cahier des charges..... | 5 |
| a. Présentation rapide de mon application | 5 |
| b. La méthode de modélisation | 5 |
| 2. Le système de gestion de base de données (SGBD) | 6 |
| 3. Le langage | 6 |
| 4. La mise en page..... | 7 |
| 5. L'hébergeur | 7 |
| 6. Architecture Client/Serveur..... | 8 |
| Partie 2 – Les fonctionnalités de mon application | 9 |
| 1. Au niveau des musiciens | 9 |
| a. Rechercher..... | 9 |
| b. Ajouter | 9 |
| c. Modifier | 9 |
| d. Supprimer | 9 |
| e. Ajouter un musicien à un évènement | 9 |
| 2. Au niveau de l'inscription de personnes « extérieures » au groupe | 10 |
| a. Ajouter | 10 |
| Partie 3 – Difficultés, apports et autocritique | 11 |
| ANNEXES | 12 |

Le sujet du projet était libre, nous n'avions pas beaucoup de contraintes. J'ai donc choisi de créer une application en rapport avec une de mes activités extra-scolaire : la musique.

Pour présenter mon application, je vais décomposer mon rapport en trois grandes parties : dans un premier temps je vais définir l'application et énoncer mes choix de système de gestion de base de données, de langage, de mise en page et d'hébergeur, puis dans une deuxième partie, je présenterai les fonctionnalités de mon application et enfin, dans une troisième partie j'analyserai les difficultés que j'ai rencontrées, j'exposerai les points qui m'ont apportés des connaissances et je ferai une critique de mon travail et sur ma façon d'aborder ce projet.

Partie 1 – Définition de l'application

1. Cahier des charges

Le choix du sujet était libre, mais nous avons quelques contraintes à respecter. Notre base de données devait être composée d'au moins trois tables, avec des contraintes statiques, c'est-à-dire des contraintes de clés primaires et de clés étrangères sur les tables, et des contraintes dynamiques : deux triggers.

a. Présentation rapide de mon application

Il y a une activité extra-scolaire qui me tient très à cœur : la musique. Je joue depuis très longtemps dans une harmonie, et c'est naturellement que j'ai voulu orienter mon application vers la gestion d'un groupe de musique.

Un orchestre peut être composé d'un grand nombre de musiciens, il est parfois difficile de connaître les disponibilités de chacun afin d'accepter ou non un contrat d'animation qui est proposé à l'association.

Mon objectif était de réaliser une application qui puisse permettre de gérer la participation des musiciens à des contrats, les informations des musiciens, et de gérer l'inscription d'amis à des événements, tout en gérant la connexion de toutes les personnes susceptibles de participer à un événement.

b. La méthode de modélisation

Pour modéliser ma base de données, deux choix s'offraient à moi : l'utilisation de la méthode Merise2 ou l'utilisation de la méthode UML, que nous avons toutes les deux étudiées en cours durant le semestre.

J'ai choisi d'utiliser la méthode Merise2, car j'avais plus de recul, et donc je me sentais plus à l'aise et sûre avec cette méthode.

Les règles de gestion de ma base de données sont les suivantes :

- Un musicien ne peut jouer que d'un seul instrument
- Un musicien peut participer à plusieurs événements
- Un événement peut avoir plusieurs musiciens participant
- Un événement peut avoir plusieurs inscriptions
- Une inscription n'est relative qu'à un seul événement

J'ai donc obtenu le Schéma Relationnel suivant :

instrument (NumInst, NomInst)

personne (Num, Nom, Prenom, Ville, CodePost, #NumInst)

participer (Num, NumEv)

evenement (NumEv, NomEv, Description, VilleEv, DateEv, NbPlaces)

inscription (NumInscrip, NomP, PrenomP, VilleP, CodePostP, #NumEv)

(Le Modèle Conceptuel des Données est disponible en Annexe 1.)

2. Le système de gestion de base de données (SGBD)

Une base de données peut être gérée grâce à un système de gestion de base de données, il en existe beaucoup, c'est pour cela que j'ai décidé de m'intéresser qu'à quelques-uns de ces systèmes : Oracle, PostgreSQL et MySQL.

J'ai effectué des recherches sur chacun de ces systèmes pour choisir le plus adapté à mes besoins.

Oracle est un système très performant qui peut gérer un très grand nombre de données, mais il demande beaucoup de ressources.

De plus, durant mon cursus scolaire de DUT Statistique et Informatique Décisionnelle, il est le système de gestion de base de données que j'ai le plus souvent utilisé, donc j'ai préféré choisir un nouveau SGBD pour acquérir des connaissances sur un autre système.

PostgreSQL est un système qui permet de gérer un très grand nombre de données, et permet de bons contrôles. Mais j'avais choisi dans un premier temps d'utiliser une base de données appartenant à un hébergeur, il s'agissait de MySQL. Cependant, il y a eu un problème de connexion à la base de données, c'est à ce moment-là, que j'ai commencé à me renseigner sur PostgreSQL. Comme j'avais débuté avec MySQL je n'ai pas voulu changer de SGBD, j'ai préféré continuer sur MySQL car j'avais déjà en main son interface graphique PhpMyAdmin et il s'agissait également de mon premier choix.

3. Le langage

Le langage HTML est un langage statique. Il peut être complété par un langage dynamique qui permettra de faire évoluer le contenu des pages Web en fonction des données contenues dans la base de données ou encore des données variables en fonction du jour, de l'heure, etc.

Pour réaliser un site Web, il existe beaucoup de langages dynamiques comme notamment Python, Ruby, Javascript et PHP.

J'avais déjà utilisé le langage Python pour le projet d'algorithmique du premier semestre. Là encore, je n'ai pas souhaité utiliser Python pour m'inciter à développer de nouvelles connaissances.

Ruby est un langage simple et efficace, qui rend la lecture facile. Mais il s'agit aussi d'un langage de programmation assez nouveau qui possède son langage de programmation assez unique, et qui demande des recherches approfondies.

Enfin, Javascript, est un langage de programmation coté client. Il s'exécute différemment en fonction de chaque client, et en fonction de chaque navigateur. Ce qui est selon moi un inconvénient, et qui rend le langage Javascript très peu sécurisé.

J'ai finalement choisi le langage PHP, qui signifie Hypertext Preprocessor. C'est un langage de programmation coté serveur, donc il permet de développer avec beaucoup de puissance. Il s'exécute coté serveur, ce qui fait de lui un langage sécurisé puisque son code n'est donc pas visible par l'utilisateur. N'ayant jamais réalisé d'application web, ce langage m'a paru plutôt simple. De plus, c'est un langage très utilisé que je pourrai rencontrer en entreprise.

4. La mise en page

Bien qu'un langage dynamique permette de créer une application interactive, la notion de mise en page est importante.

Pour effectuer cette mise en page, j'ai choisi d'utiliser les feuilles de style CSS, qui signifie Cascading Style Sheets. Ces feuilles de style permettent de gérer les polices de caractères, les couleurs, les marges, les alignements, les bordures, etc. Elles structurent les pages HTML.

5. L'hébergeur

Le choix de l'hébergeur a été pour moi l'étape la plus difficile car je ne connaissais aucun hébergeur.

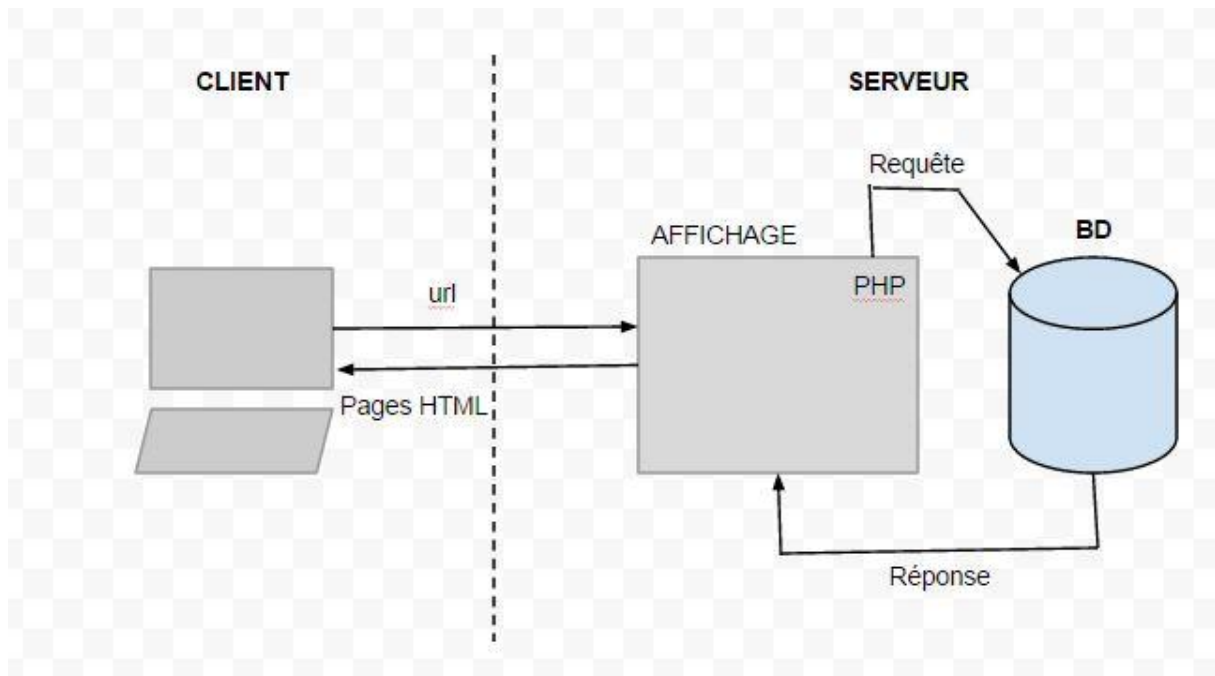
J'ai donc fait confiance à certains de mes camarades qui me conseillaient d'utiliser Olympe. Mais suite à un problème de connexion au serveur, j'ai décidé de changer d'hébergeur.

Après quelques rapides recherches, j'avais le choix entre Heroku et OpenShift. Malheureusement, ma base de données étant MySQL, l'utilisation de MySQL sur Heroku était payante, c'est pour cette raison que je me suis finalement dirigée sur OpenShift qui permet d'héberger des applications générées avec de nombreux langages, et notamment PHP, et qui supporte des bases de données MySQL gratuitement.

6. Architecture Client/Serveur

L'architecture Client/Serveur désigne un mode de communication entre deux entités d'un même réseau. Le client envoie des requêtes ou des demandes au serveur qui retourne la réponse au client.

Voici le schéma d'architecture Client/Serveur de mon application :



Lorsque le client inscrit une URL dans la barre de recherche d'un navigateur, il obtient l'affichage d'une page PHP. Tout se passe du côté du serveur, et il n'est restitué au client qu'un affichage de page HTML.

Le PHP envoie des requêtes à la base de données qui exécute les requêtes et qui restitue des réponses, qui sont ensuite affichées en PHP et retournées au client sous forme d'une page HTML.

Partie 2 – Les fonctionnalités de mon application

1. Au niveau des musiciens

a. Rechercher

Pour recherche les musiciens d'un pupitre, l'utilisateur sélectionne un instrument dans une liste déroulante, et clique sur le bouton 'Rechercher' pour afficher la liste des musiciens.

b. Ajouter

On peut aussi ajouter un nouveau musicien. Il est demandé de remplir le nom, prénom, ville et code postal du musicien et de choisir dans la liste déroulante l'instrument joué. Si l'un des champs n'est pas rempli, l'inscription ne peut pas être effectuée, il faut recommencer.

c. Modifier

Il est possible de modifier des informations de musiciens. Il faut rechercher un musicien dans la liste déroulante, et modifier les informations en remplissant tous les champs (même s'il n'y a pas de changements).

Si tous les champs ne sont pas remplis, la modification n'est pas effectuée et il faut recommencer.

d. Supprimer

La suppression d'un musicien fait également intervenir une liste déroulante. Il faut le sélectionner le musicien dans la liste et cliquer sur le bouton 'Supprimer'.

e. Ajouter un musicien à un évènement

Il existe une dernière fonctionnalité associée aux musiciens : l'ajout d'un musicien à un évènement. Pour cela, il faut choisir un musicien ainsi qu'un évènement dans une liste déroulante et valider le choix.

Un trigger s'exécute pour permettre d'incrémenter le nombre de participation d'un musicien à chaque fois que l'on inscrit un musicien à un nouvel évènement.

2. Au niveau de l'inscription de personnes « extérieures » au groupe

a. Ajouter

Pour s'inscrire à un évènement, une personne doit indiquer son nom, son prénom, sa ville et son code postal, et elle doit sélectionner l'évènement dans une liste déroulante.

Il y a ici l'exécution d'un trigger qui permet de ne pas ajouter une personne à un évènement s'il n'y a plus de place. Dans ce cas, l'inscription n'est pas finalisée et un message est inscrit sur la page.

Partie 3 – Difficultés, apports et autocritique

Une des principales difficultés du projet a été la recherche du sujet, et la gestion de mon temps.

En effet, nous disposions de très peu de temps. J'ai donc établi un planning pour me permettre d'avancer pas à pas sans perdre de temps.

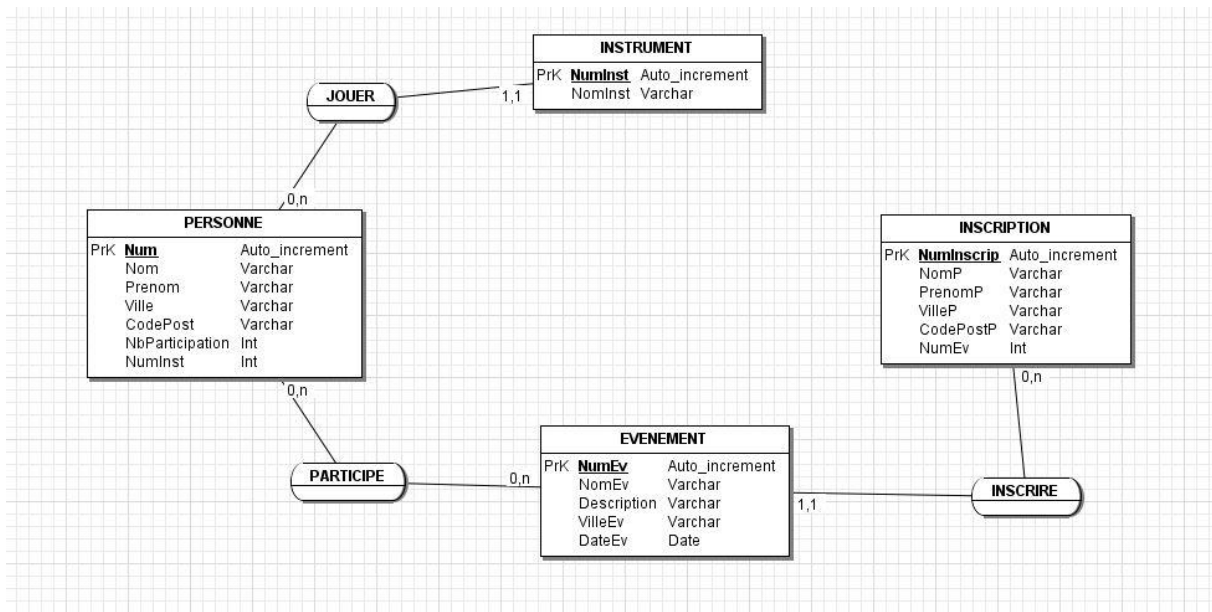
Dans un premier temps, j'ai choisi de me consacrer à la création de ma base de données, c'est-à-dire mes tables, les relations entre ces tables et également les triggers. Ensuite, j'ai commencé mes recherches sur les systèmes de gestion de base de données, les langages et les hébergeurs. Puis j'ai commencé le codage grâce à des tutoriels sur internet.

Je me sens plus motivée, réactive et impliquée lors de projets plutôt que lors de cours magistraux, travaux dirigés et travaux pratiques, même s'ils sont indispensables à notre enseignement. Les projets nous permettent d'apprendre à notre rythme personnel, ce qui est un très bon point dans le cas où le sujet du projet nous intéresse et que nous sommes acteurs du projet.

Je n'ai pas atteint tous les objectifs que je m'étais fixés mais je suis fière de mon implication et du rendu de mon application. De nombreuses fonctionnalités supplémentaires pourront être ajoutées plus tard durant mon temps libre comme la recherche par mot clé ou encore l'authentification des visiteurs.

Ce projet m'a permis d'acquérir des connaissances et de développer mon autonomie de travail mais aussi mon esprit d'équipe du fait de l'aide apportée à certains de mes camarades.

ANNEXES



Annexe 1 - Modèle Conceptuel des Données

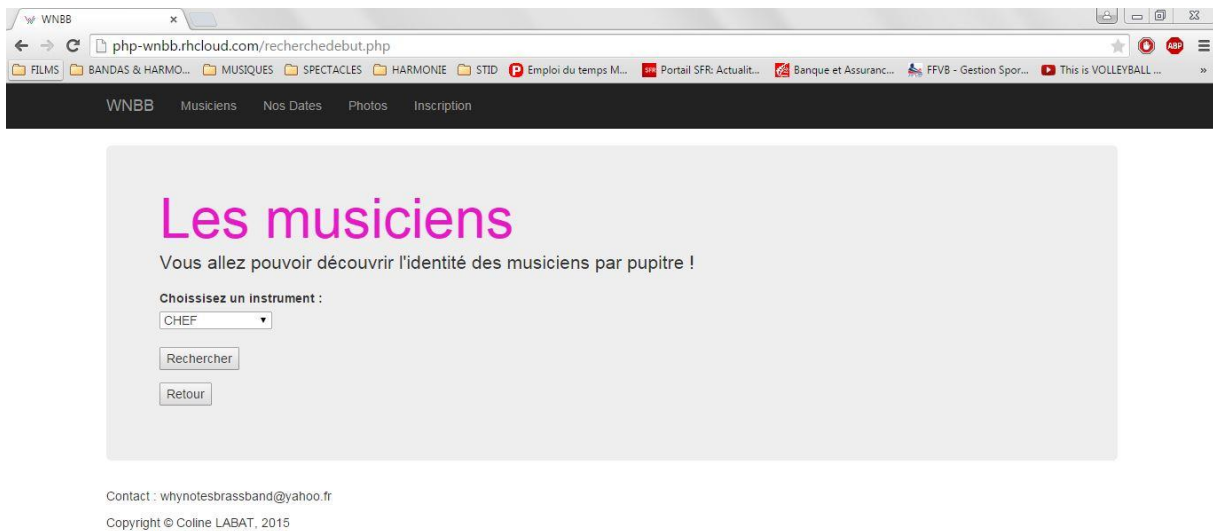


Image 1 - Recherche de musiciens

The screenshot shows a web browser window with the address bar displaying 'php-wnbb.rhcloud.com/insciremusicien.php'. The page has a dark navigation bar with links: WNBB, Musiciens, Nos Dates, Photos, and Inscription. The main content area has a light gray background and features the title 'Les musiciens' in large pink font. Below the title, it says 'Vous pouvez inscrire un nouveau musicien'. The form includes input fields for 'Nom', 'Prenom', 'Ville', and 'Code Postal'. There is a dropdown menu for 'Selectionner l'instrument' with 'CHEF' selected. At the bottom of the form are 'Valider' and 'Retour' buttons.

Image 2 - Inscription d'un musicien

The screenshot shows a web browser window with the address bar displaying 'php-wnbb.rhcloud.com/modifier.php'. The page has the same dark navigation bar as Image 2. The main content area has a light gray background and features the title 'Les musiciens' in large pink font. Below the title, it says 'Choisissez un musicien :'. There is a dropdown menu showing '37 - LABAT COLINE'. Below the dropdown are 'Continuer' and 'Retour' buttons. At the bottom of the page, there is contact information: 'Contact : whynotesbrassband@yahoo.fr' and 'Copyright © Coline LABAT, 2015'.

Image 3 - Choix d'un musicien pour la modification

The screenshot shows a web browser window with the URL `php-wnbb.rhcloud.com/modif2.php`. The page has a dark navigation bar with links: WNBB, Musiciens, Nos Dates, Photos, and Inscription. The main content area is titled 'Les musiciens' in large pink font, with the subtitle 'Vous pouvez modifier les informations'. Below this, there is a form with the following fields: 'Numero :' with the value '37', 'Nom :' with the value 'LABAT', 'Prenom :' with the value 'COLINE', 'Ville :' with the value 'AIRE SUR ADOUR', 'Code Postal :' with the value '40800', and 'Selectionner l'instrument :' with a dropdown menu showing 'CHEF'. A 'Valider' button is at the bottom of the form.

Image 4 - Formulaire de modification d'un musicien

The screenshot shows a web browser window with the URL `php-wnbb.rhcloud.com/supprimer.php`. The page has the same dark navigation bar as Image 4. The main content area is titled 'Les musiciens' in large pink font, with the subtitle 'Vous pouvez supprimer un musicien'. Below this, there is a form with the label 'Choisissez un musicien :'. A dropdown menu shows the selected option '37 - LABAT COLINE'. Below the dropdown are two buttons: 'Supprimer' and 'Retour'. At the bottom of the page, there is contact information: 'Contact : whynotesbrassband@yahoo.fr' and 'Copyright © Coline LABAT, 2015'.

Image 5 - Suppression d'un musicien

WNBB Musiciens Nos Dates Photos Inscription

Les musiciens

Choisissez un musicien :
37 - LABAT COLINE

Selectionner l'événement :
AIRE SUR ADOUR, Concert de Printemps

Continuer

Retour

Contact : whynotesbrassband@yahoo.fr
Copyright © Coline LABAT, 2015

Image 6 - Inscription d'un musicien à un évènement

WNBB Musiciens Nos Dates Photos Inscription

Inscription

Sur cette page, vous pouvez vous inscrire à certains de nos événements (en fonction du nombre de places) !

Nom :

Prenom :

Ville :

Code Postal :

Selectionner l'événement :
AIRE SUR ADOUR, Baptême Candice

Valider

Image 7 - Inscription d'une personne à un évènement