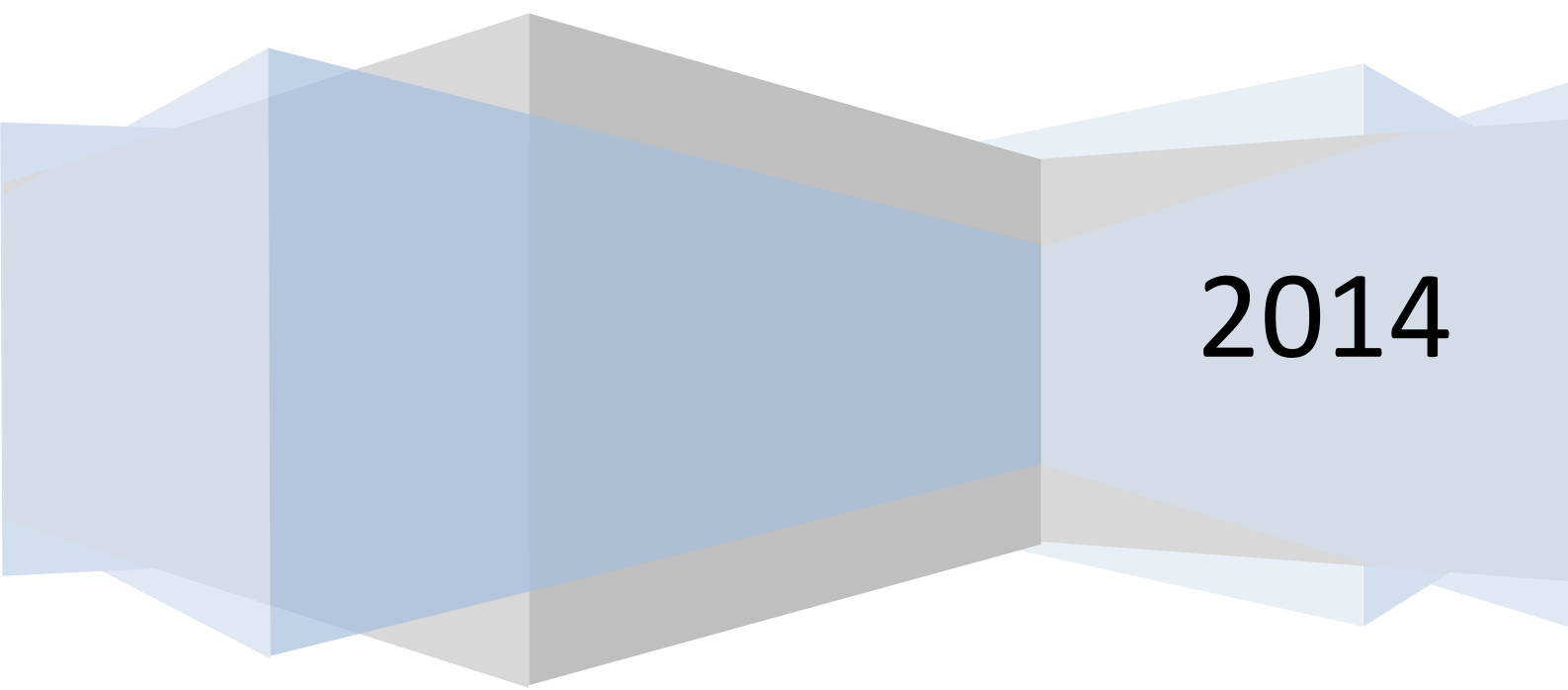


Art Plume

Documentation technique

Falhun Sidney



2014

I. Contexte

Art Plume est une association culturelle et créative créée par un collectif artistique en 1995 qui a pour but de soutenir à la création. Elle abrite un collectif réunissant tous les horizons artistiques, de l'art plastique à l'art de rue et héberge ainsi des artistes en résidence sur des durées plus ou moins longue.

L'association organise également "Les Hétéroclites" qui est un festival populaire d'une durée de 3 jours qui se veut être le reflet des actions menées tout au long de l'année dans ses locaux.

Pour améliorer son image de marque Art Plume demande de réaliser une application pour la distribution Android qui devrait permettre de présenter au mieux l'association sur les nouvelles plates-formes mais aussi d'en améliorer l'image de marque.

Pour cela une application tournant sur Android va être créé en front office ainsi qu'une application web développé en php qui devrait permettre à l'administrateur de mieux gérer les informations disponibles dans l'application.

II. Fonctionnalités attendues

Les fonctionnalités attendues pour les visiteurs sont:

- La consultation des informations de l'association (description, histoire, informations de contact, ...)
- L'enregistrement à une Newsletter pour recevoir par mail les nouveautés de l'association.
- La consultation des différents événements organisés par Art plume.
- La consultation d'un itinéraire pour se rendre aux locaux d'Art plume.
- La consultation des fiches sur les différents artistes résidents dans l'association.
- Le contact de l'association par un formulaire de contact.
- La consultation d'un événement en direct.

Les fonctionnalités attendues pour l'administrateur sont:

- La modification des informations concernant l'association.
- La gestion des messages envoyées par le formulaire de contact.
- La modification et l'ajout des fiches concernant les artistes résidents dans l'association.
- La gestion d'un lien pour le stream des différents événements.
- La modification et l'ajout des fiches concernant les événements dans l'association.

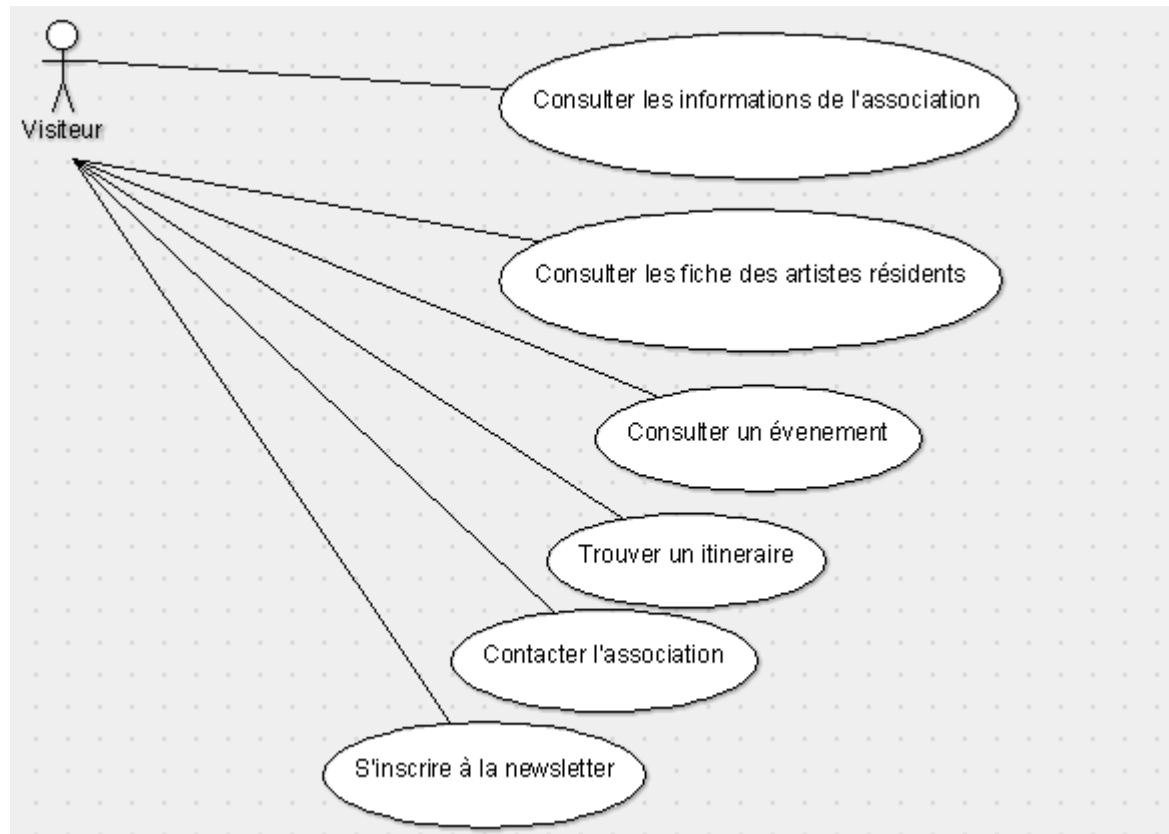
III. Les outils utilisés

Les outils utilisés pour le projet sont:

- Java developpement kit + Le SDK Android + Eclipse (pour développer l'application Android)
- Prototyper Free (pour créer les maquettes de l'application Android)
- Microsoft Word (pour rédiger les différentes documentations)
- Windows 8 & OS X Maverick
- Xampp (pour pouvoir développer en local l'application pour l'administrateur)
- Notepad++ ou sublimeText (pour développer en php)
- Filezilla (pour transférer les différents fichiers sur FTP)
- Photoshop (pour créer les différentes icônes de l'application)
- Github accessible à l'adresse: <https://github.com/Drkuhybride/Art-Plume-Android> (pour le versionning et la sauvegarde des fichiers)

IV. Cas d'utilisations

Cas d'utilisation pour l'application sous Android:



V. Explications techniques sur l'administration

A) Connexion à la base de données

La connexion à la base de données se fait dans le fichier connexion.inc.php dans le dossier inc.

Pour se connecter à la base de données on instancie une nouvelle classe Bdd avec en paramètre les informations de connexion à la base de données (cette classe est une classe d'abstraction PDO présente dans "classe/clstBdd.class.php").

B) Connexion à l'application

Le formulaire pour se connecter à l'administration est disponible dans "FO/vues/FO_connexion.php". Dans ce formulaire le login et le mot de passe seront envoyés par des inputs texte.

Ces informations seront transmises par la méthode POST à la page "inc/verifLogin.inc.php". Dans la page verifLogin.inc.php on récupère le login et le mot de passe dans une variable puis on recherche le nombre d'utilisateurs ayant ce login et ce mot de passe. Si le nombre d'utilisateurs n'est pas vide on remplit la session login (\$_SESSION['login']) avec le login entré. Si le nombre d'utilisateurs est null alors on redirige vers l'accueil.

En remplissant la session login cela devrait permettre de se connecter à l'application.

C) Consulter les messages

La consultation des messages se fait à l'aide de la page "FO/vues/messages/FO_consulterMessages.php" on récupère alors la liste des messages contenu dans la base de données avec la fonction getLesMessages() contenu dans la page "FO/modeles/messages/lireMessages.inc.php". Si des messages existent dans la base de données on va afficher le sujet du message et l'adresse mail de celui qui a créé le message. On va également mettre les messages non lus en gras. Si il n'y a pas de messages alors on affichera le message "Vous n'avez reçu aucun messages".

D) Consulter les Événements

La consultation des événements se fait à l'aide de la page "FO/vues/evenements/FO_consulterEvenements.php" on récupère alors la liste des événements contenu dans la base de données avec la fonction getLesEvenements() contenu dans la page "FO/modeles/evenements/lireEvenements.inc.php". Si des événements existent dans la base de données on va afficher le titre de l'événement, le lieu, la date, la photo, la description, le lien et le type d'événements. Si il n'y a pas de messages alors on affichera le message "Il n'y a aucun événement, pensez à en ajouter."

E) Consulter les artistes

La consultation des artistes se fait à l'aide de la page "FO/vues/artistes/FO_consulterArtistes.php" on récupère alors la liste des artistes contenu dans la base de données avec la fonction getLesArtistes() contenu dans la page "FO/modeles/artistes/lireArtistes.inc.php". Si des artistes existent dans la base de données on va afficher le nom de l'artiste, son prénom, sa photo, son âge, sa description et un lien vers sa page perso si il en a une. Si il n'y a pas de messages alors on affichera le message "Il n'y a aucun artiste, pensez à en ajouter."

F) Consulter un message

En cliquant sur un message dans liste des messages de la page

"FO/vues/messages/FO_consulterMessages.php" on est redirigé vers la page

"FO/vues/messages/FO_consulterUnMessage.php". On récupère alors les informations concernant un message de la base de données avec la fonction `getLeMessage($code)` contenu dans la page

"FO/modeles/messages/lireMessages.inc.php", on récupère également l'état du message avec la fonction `getEtatMess($code)` contenu dans la même page.

L'état du message va être ensuite réutilisé pour changer l'état du message en lu avec la fonction `changeLu($code)` contenu dans la page "FO/modeles/messages/modifierEtatMess.inc.php". Si le message avec l'id récupéré avec la méthode POST existe bien alors on affiche le sujet du message, le nom de la personne qui a envoyé le message, le prénom de la personne qui a envoyé le message, l'email de cette personne et le contenu du message. Si il n'y a pas de messages correspondant avec cet id alors on sera redirigé directement avec la liste des messages.

G) Ajouter un événement

L'ajout des événements se fait à l'aide de la page

"FO/vues/evenements/FO_ajouterEvenements.php". Les informations liées à l'événement sont envoyés avec la méthode POST et en redirection vers la page

"FO/modeles/evenements/ajouterEvenement.inc.php". Si les informations relatives à l'événement sont null alors on sera redirigé vers la page de consultation des événements avec le message "Veuillez remplir tous les champs." sinon ces informations seront insérées dans la base de données puis on sera redirigé vers la page de consultation des événements avec le message "Ajout de l'événement effectué avec succès."

H) Ajouter un type d'événement

L'ajout d'un type d'événement se fait à l'aide de la page

"FO/vues/typeevenements/FO_ajouterTypeevenements.php". Le type d'événement sera envoyé avec la méthode POST et en redirection vers la page

"FO/modeles/typeevenements/ajouterTypeevenement.inc.php". Si le type d'événement est null alors on sera redirigé vers la page de consultation des événements avec le message "Veuillez remplir tous les champs." sinon le type d'événement sera inséré dans la base de données puis on sera redirigé vers la page de consultation des événements avec le message "Ajout du type d'événement effectué avec succès."

I) Ajouter un artiste

L'ajout des artistes se fait à l'aide de la page "FO/vues/artistes/FO_ajouterArtiste.php". Les informations liées à l'artiste sont envoyées avec la méthode POST et en redirection vers la page "FO/modeles/artistes/ajouterArtiste.inc.php". Si les informations relatives à l'artiste sont null alors on sera redirigé vers la page de consultation des artistes avec le message "Veuillez remplir tous les champs." sinon ces informations seront insérées dans la base de données puis on sera redirigé vers la page de consultation des artistes avec le message "Ajout de l'artiste effectué avec succès."

J) Modifier un événement

La modification d'un événement se fait à l'aide de la page "FO/vues/evenements/FO_modifierEvenement.php". On récupère les informations de l'événement sélectionné précédemment avec la fonction `getUnEvenement($code)` de la page "FO/modeles/evenements/lireEvenements.inc.php". Ces informations seront alors insérées dans les input texte. Après modification on sera redirigé vers la page "FO/modeles/evenements/modifierEvenement.inc.php", toutes les informations liées à l'événement seront envoyées avec la méthodes POST. Si les informations sont null alors le message "Veuillez remplir tous les champs." s'affichera et on sera redirigé vers la page de consultation des événements sinon les informations seront insérées dans la base de données et on sera redirigé vers la page de consultation des événements avec le message "Modification de l'événement effectuée avec succès."

K) Modifier un artiste

La modification d'un artiste se fait à l'aide de la page "FO/vues/artistes/FO_modifierArtiste.php". On récupère les informations de l'artiste sélectionné précédemment avec la fonction `getUnArtiste($code)` de la page "FO/modeles/artistes/lireArtistes.inc.php". Ces informations seront alors insérées dans les input texte. Après modification on sera redirigé vers la page "FO/modeles/artistes/modifierArtiste.inc.php", toutes les informations liées à l'artiste seront envoyées avec la méthodes POST. Si les informations sont null alors le message "Veuillez remplir tous les champs." s'affichera et on sera redirigé vers la page de consultation des artistes sinon les informations seront insérées dans la base de données et on sera redirigé vers la page de consultation des artistes avec le message "Modification de l'artiste effectuée avec succès."

L) Modifier les informations de l'association

La modification des informations de l'association se fait à l'aide de la page "FO/vues/artistes/FO_modifierInfos.php". On récupère les informations de l'association avec la fonction `getInfos()` de la page "FO/modeles/infos/lireInfos.inc.php". Ces informations seront alors insérées dans les input texte. Après modification on sera redirigé vers la page "FO/modeles/infos/modifierInfos.inc.php", toutes les informations de l'association seront envoyées

avec la méthodes POST. Si les informations sont null alors le message "Veuillez remplir tous les champs." s'affichera et on sera redirigé vers la page de modification des informations de l'association sinon les informations seront insérées dans la base de données et on sera redirigé vers la page de modification des informations de l'association avec le message "Modification des informations effectuée avec succès."

M) Modifier le lien de stream

La modification du lien de stream se fait à l'aide de la page "FO/vues/stream/FO_modifierLien.php". On récupère le lien de stream avec la fonction `getUnLien()` de la page "FO/modeles/stream/lireStream.inc.php". Ce lien sera alors inséré dans les input texte. Après modification on sera redirigé vers la page "FO/modeles/stream/modifierLien.inc.php", le lien de stream sera envoyé avec la méthodes POST. Si le lien est null alors le message "Veuillez remplir tous les champs." s'affichera et on sera redirigé vers la page de modification de modification de lien de stream sinon le lien sera inséré dans la base de données et on sera redirigé vers la page de modification du lien de stream avec le message "Modification du lien effectuée avec succès."

N) Gestion des liens avec l'index

Les pages sont lues par l'utilisateur directement depuis l'index et non vers les pages en dures. Le lien sera alors du type `index.php?page=` pour toutes les pages dans le dossier "vues" mais aussi pour les pages dans le dossier "modeles" qui concernent une modification ou un ajout dans la base de données.

À noter que `verifLogin.inc.php` est aussi inclus dans l'index et appelé par la méthode GET avec `index.php?page=`

VI. Explications techniques sur l'application android

L'application Android fonctionne en tant que liseuse. On récupère alors les pages php disponibles dans le sous-dossier "android" et on les affiche avec l'objet webview directement intégré avec eclipse.

Les pages dans le dossier android fonctionnent de la même manière que l'administration au niveau des relations entre les tables