ECOLE PRATIQUE DES HAUTES ETUDES COMMERCIALES

Avenue du Ciseau, 15
1348 Louvain-la-Neuve

Développement d'applications pour le Brussels Philharmonic Orchestra

Travail de fin d'études présenté en vue de l'obtention du diplôme de bachelier en Informatique et Systèmes : finalité Technologie de l'informatique

Thomas Pické 3TL2

Rapporteur : Arnaud Dewulf

Année Académique 2015 – 2016

REMERCIEMENTS

Tout d’abord, je remercie Mr Dewulf, professeur rapporteur de mon travail de fin d’études, qui m’a guidé tout au long de ce projet.

Ensuite, je tiens à remercier le Brussels Philharmonic Orchestra, sans qui je n’aurais pas pu réaliser ce travail de fin d’études.

Enfin, je remercie également toutes les personnes qui ont participé à la relecture de ce rapport et qui m’ont conseillé lors de la réalisation de ce travail.

Table des matières

1. Introduction 1

2. Rencontre avec l’orchestre et cahier des charges 1

3. Plan de travail 3

4. Méthode de travail 3

5. Conception de la base de données 4

6. Spécifications applicatives 6

    a. Application des spectateurs 6

    b. Interface d’administration 8

    c. Application des musiciens 13

7. Spécifications techniques 15

8. Spécifications de réalisation 16

9. Spécifications administratives 16

10. Technologies apprises 17

11. Difficultés rencontrées 17

12. Améliorations possibles 22

13. Conclusion 22

14. Bibliographie 24

Liste des annexes

I. Page de contact **1**

II. Informations générales **2**

III. Page de connexion du site **3**

IV. Liste des utilisateurs **3**

V. Ajout d’un utilisateur **4**

VI. Modification d’un utilisateur **4**

VII. Ajout d’un événement **5**

1. Introduction

Dans le cadre de mon bachelier en Informatique et Systèmes, j'ai pu réaliser ce travail de fin d'études. Lors de mes recherches d'un sujet pour ce projet, j'ai reçu, grâce à mon professeur rapporteur, Mr Dewulf, une proposition de la part du Brussels Philharmonic Orchestra. Après concertation avec les représentants de l'orchestre, il a été décidé qu'il fallait développer deux applications Android et une interface web d'administration.

En effet, dès le début de mes recherches quant au sujet de mon travail de fin d'études, je désirais me diriger vers du développement. De plus, le domaine du "mobile" m'intéresse particulièrement, car en pleine expansion et parce que j'ai pu en avoir un aperçu lors du projet d'intégration du premier quadrimestre de cette année.

De son côté, le Brussels Philharmonic Orchestra désirait se moderniser et disposer d'un système de communication centralisé interne, car jusqu'ici, les contacts entre musiciens se faisaient via divers moyen de communication tels que les réseaux sociaux, téléphones, email, et autres applications mobiles, et externe, afin que les spectateurs puissent disposer des informations concernant l'orchestre de manière plus aisée et n'importe où. C'est pourquoi il a été décidé dès notre première rencontre que deux applications mobiles étaient nécessaires, une pour les spectateurs, disposant principalement d'un agenda et d'une page de contact, et une pour les musiciens et membres de l'orchestre, afin de pouvoir communiquer entre eux et être informés des éventuelles informations concernant les répétitions et autres événements internes à l'orchestre. Afin de gérer et d'administrer toutes ces données, il est également nécessaire de disposer d'un site web, pour bénéficier d'une interface claire et précise.

Il est important de préciser que les deux applications sont nécessaires, car l'orchestre désire que tout soit tout à fait transparent du point de vue des spectateurs. L'idée d'une application unique fût bien entendu évoquée au début, mais nous nous sommes rapidement rendu compte que cela impliquait d'office que les spectateurs verraient la page de connexion, ce qu'il fallait justement éviter. De ce fait, il a été décidé de réaliser deux applications séparées, ce qui garantirait un plus grand confort aux spectateurs.

Dans un premier temps, je décrirai la première rencontre avec l'orchestre et l'élaboration du cahier des charges. Ensuite, j'expliquerai le plan et la méthode de travail que j'avais prévus. Cela sera suivi d'une description des fonctionnalités implémentées, ainsi que d'une explication du développement et des difficultés rencontrées. Enfin, je terminerai par une conclusion.

2. Rencontre avec l'orchestre et cahier des charges

Le vendredi 27 novembre 2015, j'ai pu rencontrer les représentants du Brussels Philharmonic Orchestra. Ce fût l'occasion de discuter des différentes fonctionnalités nécessaires, des différents besoins de l'orchestre, et de mettre en place un plan de travail.

L'objectif principal de l'orchestre était de s'introduire sur le marché des Smartphones avec une application mobile. Très vite, il a été décidé de s'orienter vers le système Android, car il représente près de 70% des parts de marché des Smartphones. Les besoins du Brussels Philharmonic Orchestra étaient d'un côté de pouvoir disposer d'un système de communication de ses informations au public plus moderne et pratique, et d'un autre côté de pouvoir centraliser les informations internes des événements, comme un changement de lieu ou d'horaire, ainsi que de les cibler sur l'événement correspondant. De plus, il fallait non seulement pouvoir disposer d'une gestion simple des données des événements et des utilisateurs, mais aussi que tous les membres de l'orchestre, ainsi que la plus grande partie possible du public, puissent en profiter, et donc offrir un support pour le français, le néerlandais, et l'anglais.

A l'issue de cette rencontre, j'ai appris que l'orchestre possédait déjà un site web en cms (Django). Je pouvais donc en récupérer les différentes informations concernant les événements afin de remplir la base de données. Cependant, le site web et les deux applications mobiles que je devais réaliser sont totalement indépendants du site web d'origine de l'orchestre.

Pour ce qui est de l'application des spectateurs, il a été convenu qu'elle serait disponible sur le Play Store, le "marketplace" des applications Android, et qu'un QR code contenant le lien de l'application sur le Play Store serait imprimé sur les affiches des concerts du Brussels Philharmonic Orchestra. Au niveau du contenu de cette application, les informations générales de l'orchestre devaient être présentes, ainsi qu'une page avec un formulaire de contact pour les demandes d'informations, remarques ou suggestions des spectateurs, et pour finir, un agenda sous forme de liste, reprenant les événements publics de l'orchestre. De plus, les utilisateurs doivent pouvoir réserver leurs billets à l'aide d'un lien amenant vers le site de réservation.

En ce qui concerne l'application des musiciens et membres de l'orchestre, il devait y avoir plus de fonctionnalités. En effet, en plus d'un agenda général de l'orchestre, il fallait un système de messagerie instantanée pour la communication entre tous les membres, ainsi que l'affichage d'une liste des membres avec possibilité de voir leur profil, dans le but de prendre contact avec la personne. De plus, chaque membre doit pouvoir disposer d'une liste d'événements auxquels il est associé. Sur chacun de ces événements, les administrateurs doivent pouvoir envoyer des "alertes" pour signaler certaines informations liées à l'événement, telles qu'un changement d'heure ou d'endroit. Cette application sera disponible en téléchargement sur le Play Store, mais seuls les membres de l'orchestre pourront s'y connecter.

Les deux applications disponibles sur le Google Play Store seront évidemment proposées gratuitement aux spectateurs et aux musiciens.

Enfin, quant à l'interface web d'administration, celle-ci doit permettre la gestion de la base de données de manière simple. Pour cela, il y a deux parties. D'une part, les utilisateurs, et d'autre part, les événements. Il fallait donc avoir une vue d'ensemble des utilisateurs, pouvoir en ajouter, en supprimer, et les modifier. Pour les événements, c'est la même chose, mais avec d'autres informations. De plus, ce site web ne doit être accessible qu'aux administrateurs de l'orchestre, les autres membres inscrits ne peuvent pas s'y connecter.

3. Plan de travail

La première chose à faire dans le déroulement de mon travail de fin d'études fût bien évidemment l'élaboration de la base de données. Après sa réalisation je devais la présenter à Mr Lambeau afin de la faire valider, pour être sûr qu'il n'y ait pas d'incohérence dans les tables ou dans les liens entre celles-ci. Pour finaliser la conception de cette base de données, il ne restait plus qu'à apporter les éventuelles modifications nécessaires indiquées par Mr Lambeau.

Ensuite, j'avais planifié de développer en premier lieu l'application des spectateurs, pour après m'en servir de base pour le développement de l'application des musiciens, afin de garder le même design, et de finir par l'interface d'administration. Cependant, après avoir réalisé l'application des spectateurs, je me suis tout de suite rendu compte que je devais faire le site web avant d'enchaîner sur la deuxième application. En effet, afin de pouvoir tester les applications et vérifier que les données étaient bien affichées, il fallait d'abord être capable d'encoder ces données.

4. Méthode de travail

J'ai eu l'occasion d'utiliser plusieurs outils pendant la réalisation de mon travail de fin d'études. Tout d'abord, j'ai commencé le développement de l'application des spectateurs grâce à l'IDE Android Studio, permettant de créer des applications pour Smartphone tournant sous Android, avec deux langages, le Java et le XML. La première étape fût de créer le "squelette" de l'application, c'est à dire faire le design et la navigation entre les différents écrans. Ensuite, je me suis occupé de rédiger les scripts PHP destinés à récupérer les informations nécessaires en base de données. Enfin, la dernière partie du développement de cette première application consistait à traiter les données reçues par les scripts PHP, et l'affichage de celles-ci.

Ensuite, je suis passé au développement du site web. Afin de réaliser cette interface d'administration, j'ai utilisé l'IDE PHP Storm, qui est très pratique et tout à fait indiqué pour le développement web. Pour cette partie, j'ai d'abord réalisé la connexion, puis la section concernant la gestion des utilisateurs, pour après passer aux événements. Le fait de pouvoir bénéficier d'une connexion à la base de données directement dans PHP Storm fût particulièrement utile.

La dernière étape est la réalisation de l'application des musiciens. Pour ce faire, je me suis basé sur l'application des spectateurs réalisée auparavant, et j'ai également suivi la même méthode. J'ai donc dans un premier temps réalisé tous les écrans nécessaires et leur design, ainsi que la navigation entre eux et le menu latéral. Ensuite, il fallait faire les scripts PHP nécessaires à l'application des musiciens, afin de pouvoir communiquer avec la base de données (récupérer des données à afficher, ou mettre des données à jour), et pour finir, gérer l'affichage de ces données ainsi que les traitements nécessaires.

Afin de garder une sauvegarde de tout le travail effectué, et aussi pour pouvoir travailler à plusieurs endroits, j'ai utilisé Github, qui me permettait d'enregistrer les modifications apportées et de récupérer celles-ci sur une autre machine. En effet,

comme je disposais de quelques heures de libres pendant mon stage pour travailler sur mon travail de fin d'études, cet outil me fût particulièrement utile.

Pour ce qui est de la documentation, je me suis aidé de plusieurs tutoriaux trouvés sur internet, ainsi que de certains livres qui m'ont permis de résoudre plusieurs problèmes et bugs présents dans mon code.

Enfin, en ce qui concerne le planning de travail, c'était peut-être la partie la plus difficile de ce projet. Rentrer chez soi le soir après une journée complète de travail au stage en devant encore trouver le temps et la motivation de continuer le développement du travail de fin d'études n'était vraiment pas une chose facile. C'est pourquoi j'ai principalement travaillé le weekend, même si je prenais quand même le temps que je pouvais en semaine, lorsque j'étais assez motivé et pas trop fatigué.

5. Conception de la base de données

Avant de réaliser la base de données, il fallait tout d'abord analyser les besoins de l'orchestre, discutés lors de notre première rencontre. Cette première étape allait permettre d'y voir plus clair lors de la partie de développement et de pouvoir directement ajouter des informations dans la base de données afin de vérifier le fonctionnement des applications.

Les deux principales entités sont la table Users et la table Events. Dans la table Users, les informations de chaque utilisateur sont présentes (le nom, le prénom, le mot de passe, le sel servant à crypter le mot de passe, la date d'inscription, la date de dernière connexion, l'adresse mail, et le numéro de téléphone), ainsi qu'un ID en auto-incrémentation servant de clé primaire.

Dans la table Events, les informations concernant chaque événement sont reprises (le titre, la description, la date, l'adresse, le prix et le lien de réservation des billets), et la clé primaire est aussi un ID en auto-incrémentation.

Il y a également une table de "droits" définissant le grade de chaque utilisateur, une table d'instruments, une table "activation" servant à (ré)activer les utilisateurs, une table "Alerts" pour enregistrer les alertes ciblées sur les événements, une table de catégories pour savoir si un événement est public ou privé, une table message pour sauvegarder la messagerie instantanée, et une table de Labels, afin de rendre les applications mobiles multi langages.

Cette table de Labels fût ajoutée un peu après le début du développement de l'application des spectateurs. En effet, je ne savais pas encore comment rendre l'application multi langages, et c'est au début de mon stage que j'ai découvert comment m'y prendre. La société dans laquelle j'étais utilisait des dictionnaires XML, ce sont des fichiers dans lesquels il y a plusieurs entrées. Chacune de ces entrées contient quatre éléments : un numéro, appelé labelID, et le label concerné traduit en trois langues, français, néerlandais et anglais. J'ai donc décidé d'employer la même méthode pour mon travail de fin d'études, mais en stockant tous ces labels dans une table de la base de données et non dans un dictionnaire XML.

En plus de ces tables, il y a plusieurs tables de liaison servant à lier les informations qui doivent l'être, comme la table User_right, qui regroupe les associations entre les utilisateurs et leur droit respectif, ou la table Event_users, qui permet de retenir les utilisateurs associés à chaque événement.

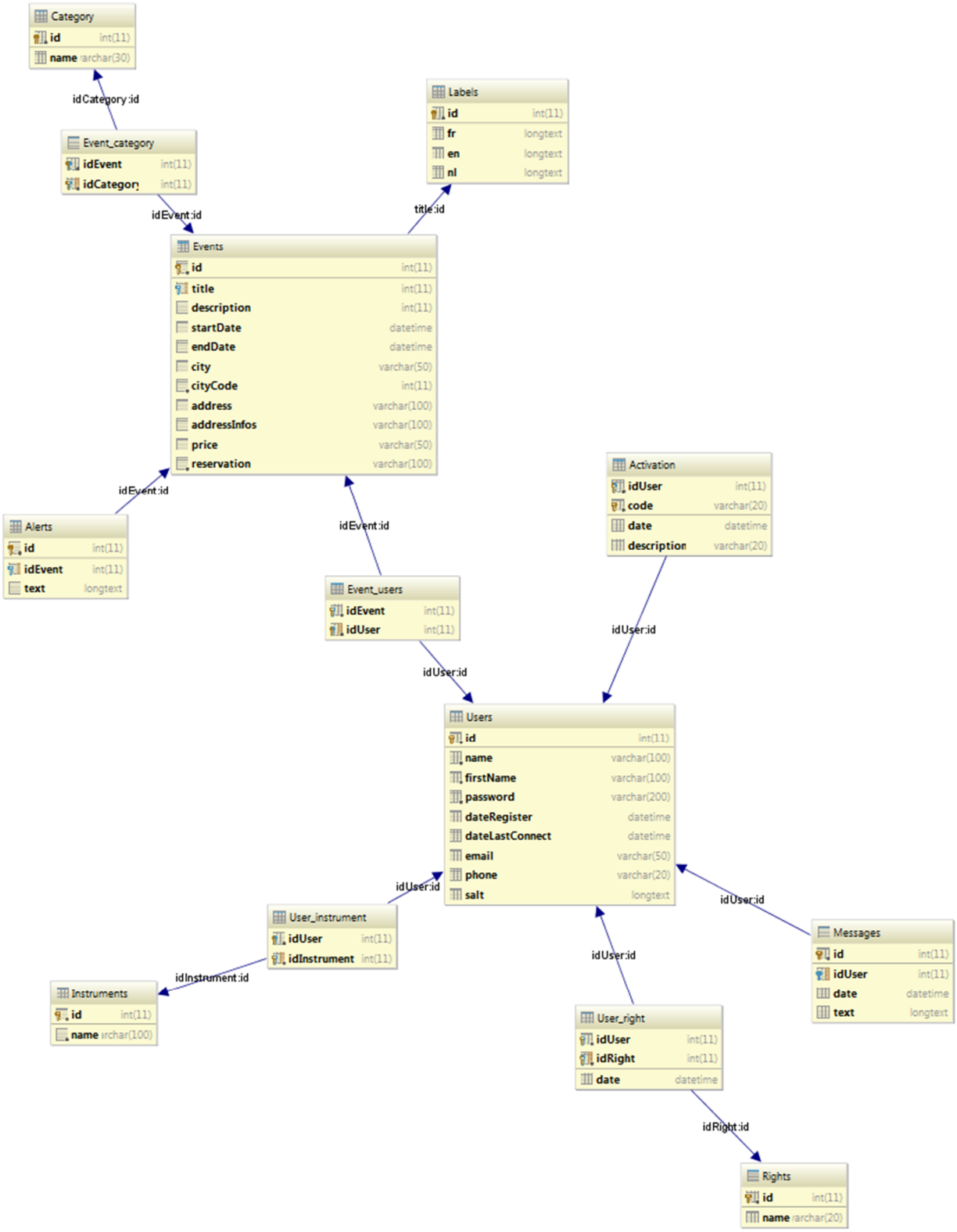

Image 1 : Schéma relationnel de la base de données

Quand j'ai eu fini de concevoir la base de données, j'ai pu la faire valider par Mr Lambeau, qui m'a cependant indiqué l'oubli d'un clé étrangère, qui a été rajoutée par la suite, et qui m'a conseillé l'ajout de la table Instrument, à laquelle je n'avais pas pensé, servant à définir l'instrument que joue chaque musicien.

6. Spécifications applicatives

Dans cette partie, les différents écrans des applications et du site web seront expliqués, afin d'avoir une meilleure compréhension de leur fonctionnement ainsi qu'une vue d'ensemble plus précise. Certains seront montrés ici, les autres seront disponibles dans la section des annexes.

a. Application des spectateurs

- **Écran de l'agenda**

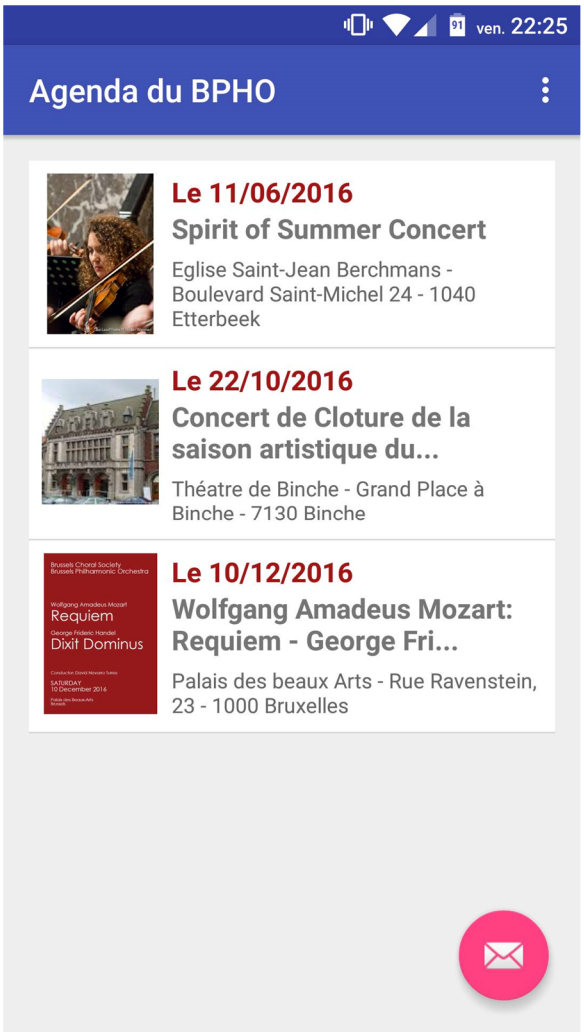

Image 2 : Agenda de l'application des spectateurs

La principale fonctionnalité de cette application est l'agenda public de l'orchestre, comme présenté ci-dessus. Il s'agit des événements qui sont enregistrés dans la base

de données. Ils sont triés sur la date, du plus proche au plus lointain, mais les événements déjà passés ne sont pas affichés. Cependant, il est possible d'y accéder en remontant la liste. La page est alors actualisée pour afficher des événements plus anciens de six mois. Il est possible de répéter ce processus jusqu'à ce qu'il n'y ait plus d'événements supplémentaires, la liste n'est alors plus actualisée et l'utilisateur en est informé par un message dans le bas de l'écran.

- **Écran d'un événement**

Lors de la sélection d'un événement de la liste, une autre page est affichée, reprenant toutes les informations nécessaires concernant cet événement, c'est à dire la date, le titre, l'adresse où a lieu l'événement, son prix, un lien de réservation des billets emmenant sur le site de l'orchestre si ils ne sont pas encore disponibles à l'achat, ainsi qu'une description. Cette dernière étant formatée en HTML, le texte peut être affiché en gras, italique, souligné, et l'on peut mettre deux sortes de titres.

Afin de rendre la navigation plus facile au sein de l'application, un bouton situé à côté du titre de la page est disponible pour revenir en arrière, sauf sur la page de l'agenda, car il s'agit de l'activité principale de l'application[1].

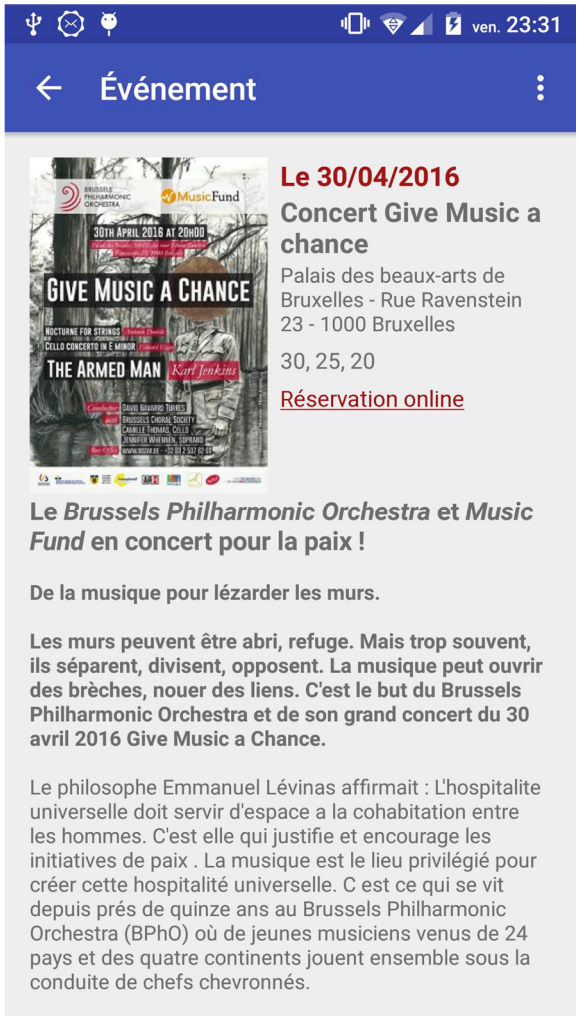

Image 3 : Ecran d'un événement

[1] Une "activité", dans le langage d'Android, représente simplement une page de l'application.

- **Page de contact**

Sur la page de l'agenda, il y a une petite icône en bas à droite permettant d'arriver sur la page de contact. Il s'agit d'un formulaire basique contenant trois champs pour le nom, l'adresse mail de la personne, et le message adressé à l'orchestre. Lors de la validation du formulaire, le message est envoyé à l'orchestre et une copie est envoyée à l'auteur du message. De plus, afin de garantir une certaine sécurité, certains caractères sont automatiquement supprimés du message, pour éviter l'éventuelle présence d'un script malveillant.

Grâce à cet écran, les spectateurs peuvent communiquer leurs remarques, suggestions à l'orchestre, ainsi que leur demander certaines informations et leur poser des questions.

- **Page d'informations générales**

La dernière fonctionnalité de cette application est la page des "informations générales", contenant une description de l'orchestre. Celle-ci raconte l'histoire du Brussels Philharmonic Orchestra, explique leurs objectifs, et détaille la raison et le contexte de leurs concerts. Il est possible d'y accéder sur toutes les pages de l'application, via le menu symbolisé par les trois points alignés verticalement en haut à droite.

Enfin, il est important de préciser que l'application est disponible dans trois langues : le français, le néerlandais, et l'anglais. La langue utilisée est définie en fonction de celle sélectionnée dans le téléphone. Si le Smartphone n'est dans aucune de ces trois langues, ou si l'information recherchée n'est pas disponible dans la langue voulue, l'anglais sera utilisé par défaut. Cependant, lorsque la langue du téléphone est modifiée après le lancement de l'application, il est nécessaire de redémarrer celle-ci pour observer les changements effectués.

b. <u>Interface d'administration</u>

Afin de réaliser le site web destiné à la gestion des événements et des utilisateurs, j'ai choisi d'utiliser un template fonctionnant avec le framework CodeIgniter et Bootstrap, dont voici un aperçu :

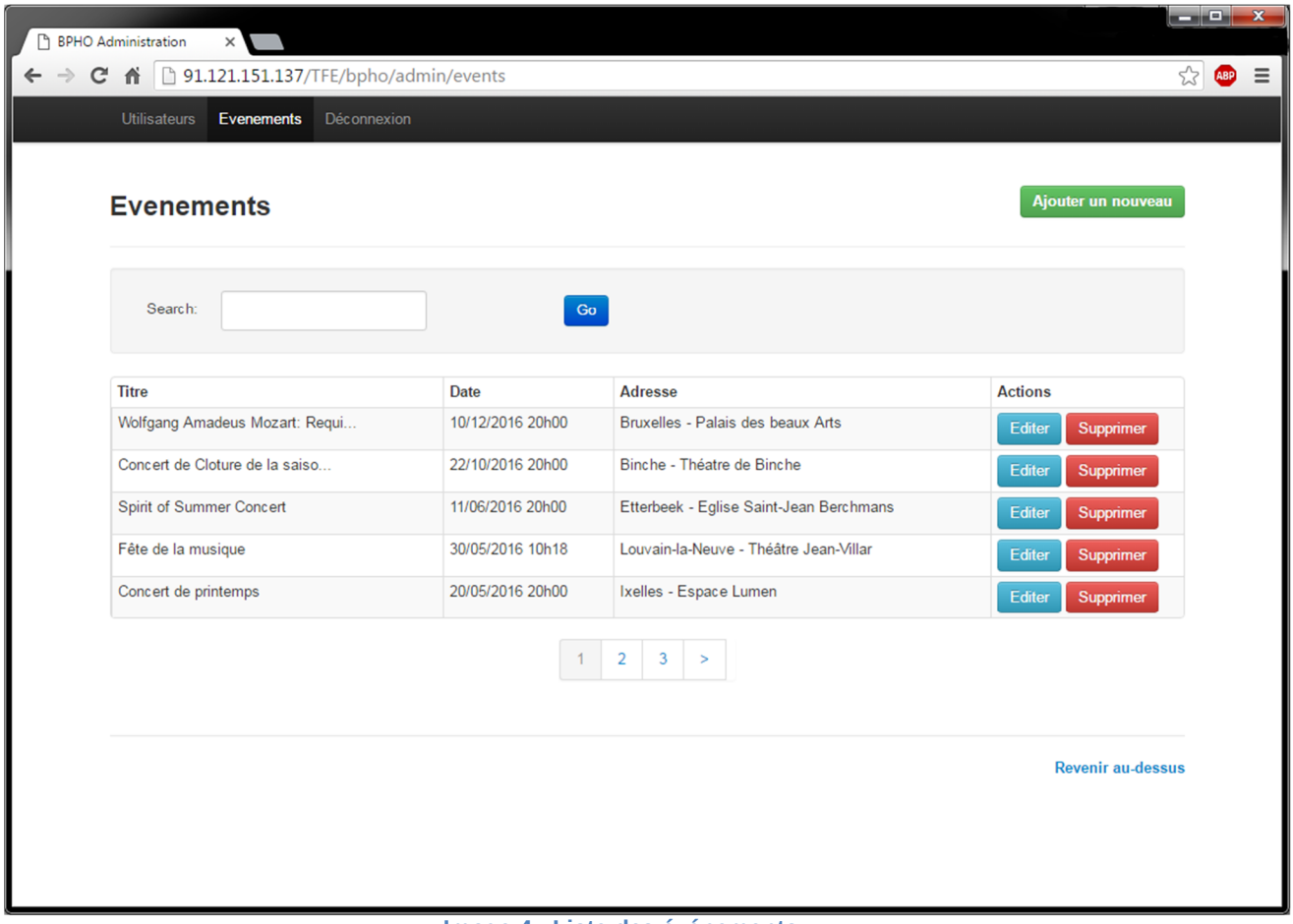

Image 4 : Liste des événements

- **Écran de connexion**

Avant d'arriver sur le site en lui-même, il faut passer par une page de connexion, car pour des raisons évidentes de sécurité, seuls les administrateurs enregistrés dans la base de données peuvent se connecter sur l'interface d'administration. Lorsqu'un utilisateur se connecte, il y a une vérification qui regarde s'il existe bien dans la base de données, que le mot de passe entré est correct, et qu'il possède bien le statut d'administrateur. Il y a aussi la possibilité de réinitialiser sont mot de passe, à l'aide du lien "mot de passe oublié ?" de cette page de connexion.

- **Écran de mot de passe oublié**

Cet écran permet à l'utilisateur de réinitialiser son mot de passe si jamais celui-ci l'a oublié. Il peut alors entrer son adresse mail dans le champ prévu à cet effet et un mail de réactivation lui est alors envoyé. Ce mail contiendra un lien qui amènera l'utilisateur sur une page lui permettant de choisir un nouveau mot de passe. Il pourra alors à nouveau se connecter comme avant.

Avant d'envoyer le mail de réactivation, il y a une vérification sur l'existence de l'adresse mail dans la base de données. Si cette adresse n'est utilisée par aucun utilisateur, un message d'erreur est affiché et aucun mail n'est envoyé. Si l'adresse mail

existe bien, un code est généré de façon aléatoire et est ajouté dans la table Activation de la base de données ainsi que dans le lien, et le mail de réactivation est bien envoyé. Lorsque l'utilisateur a finalisé la réinitialisation de son mot de passe, l'entrée créée dans la table Activation est supprimée.

- **Menu**

Le menu, présent sur toutes les pages, hormis à la connexion, contient trois boutons : utilisateurs, événements, et déconnexion. Les deux premiers mènent vers leur section correspondante de l'interface d'administration, et le bouton déconnexion permet de se déconnecter du site web.

- **Liste des utilisateurs**

Après l'authentification, nous arrivons directement sur la page de la liste des utilisateurs. Dans cette liste, les informations de chaque musicien sont affichées. De plus, il est impossible de supprimer un utilisateur disposant du grade d'administrateur, pour éviter de se retrouver sans aucun administrateur, et donc de ne plus pouvoir administrer la base de données.

Il est possible, dans cette liste, d'effectuer une recherche, qui se fait sur base du nom et du prénom, tout en étant insensible à la casse, et qui ne prend pas en compte les accents. A la fin de chaque ligne, il y a deux boutons : "éditer" et "supprimer". Le bouton "éditer" emmène sur une page permettant de modifier les données de l'utilisateur concerné, tandis que le bouton "supprimer" permet tout simplement la suppression de l'utilisateur dans la base de données, à moins que, comme dit précédemment, il s'agisse d'un utilisateur ayant le grade d'administrateur.

Le nombre d'utilisateurs affichés par page est de dix, et ces derniers sont triés par ordre alphabétique sur le prénom et ensuite sur le nom. La pagination se présente de la même façon que sur la page des événements ci-dessus.

Enfin, il y a un bouton pour ajouter un nouvel utilisateur, emmenant vers la page correspondante, situé au-dessus de la liste.

- **Ajout d'un utilisateur**

Sur cette page, nous pouvons ajouter un utilisateur dans la base de données. Les champs précédés d'un astérisque sont obligatoires, c'est à dire l'adresse mail, le mot de passe et sa confirmation, le nom, le prénom, et enfin le grade de l'utilisateur. L'adresse mail servira notamment à la connexion au site et à l'application des musiciens. Lors de l'envoi du formulaire, il y a une vérification sur l'existence de l'adresse mail dans la base de données, et un message d'erreur est affiché si tel est le cas. L'utilisateur n'est alors pas ajouté dans la base de données.

Pour ce qui est de l'instrument, il est possible de laisser le champ vide, de sélectionner un instrument parmi la liste proposée, ou même d'en mettre un nouveau, qui sera automatiquement ajouté dans la base de données lors de l'envoi du formulaire. L'existence de l'instrument indiqué est alors vérifiée, et s'il n'existe pas, il est ajouté

dans la table correspondante. Dans tous les cas, un lien est créé entre l'utilisateur et son instrument dans la table de liaison User_instrument.

- **Modification d'un utilisateur**

Cette page permet de modifier les données d'un utilisateur. Comme pour l'ajout d'un utilisateur, les champs précédés d'un astérisque sont obligatoires. Il n'est pas possible ici de changer le mot de passe ni l'adresse mail d'un utilisateur, car ce sont des fonctions réservées à l'utilisateur lui-même, disponibles dans l'application des musiciens.

Le bouton "Annuler" permet de remettre les valeurs telles qu'elles étaient avant, annulant toute modification sans pour autant devoir quitter la page.

- **Liste des événements**

La liste des événements est similaire à la liste des utilisateurs. Cette liste est triée sur la date des événements, les plus anciens étant plus bas dans la liste. Cinq éléments sont affichés par page sur l'image que l'on peut voir plus haut, afin de montrer comment la pagination fonctionne. Cependant, en temps normal dix éléments sont affichés sur chaque page, comme sur la liste des utilisateurs.

Ici aussi il est possible de créer un nouvel événement, d'en supprimer un existant, ou encore de les modifier avec les boutons adéquats. Ces fonctionnalités sont similaires à celle présentes sur la section des utilisateurs.

Pour ce qui est de la recherche, elle s'effectue sur le titre des événements, n'est pas sensible à la casse, et ne tient pas compte des éventuels accents, tout comme pour la liste des utilisateurs.

- **Ajout d'un événement**

Lors de l'ajout d'un événement dans la base de données, il est possible de spécifier un titre et une description en français, en anglais, et en néerlandais. La langue est choisie en fonction du bouton cliqué sur le dessus du formulaire. De plus, la description peut contenir deux sortes de titres ainsi que du texte en gras, italique et souligné à l'aide d'une interface WYSIWYG[2] de traitement de texte, permettant d'observer directement le résultat final de la mise en forme.

La date et l'heure de l'événement sont choisies grâce à un plugin JQuery qui affiche un calendrier et une horloge digitale. Le format de la date est changé entre son affichage et l'enregistrement de celle-ci dans la base de données, afin de correspondre au format nécessaire tout en étant agréable à lire.

Pour ce qui est du champ du lien de réservation, l'utilisateur doit obligatoirement entré l'adresse d'un site internet, sous peine de ne pas pouvoir valider le formulaire.

[2] WYSIWYG : What you see is what you get : aperçu direct des applications de styles.

Cependant, si le champ est laissé vide, par exemple parce que les billets ne sont pas encore disponibles à la vente, l'adresse du site internet public de l'orchestre sera ajoutée par défaut dans la base de données à l'endroit correspondant.

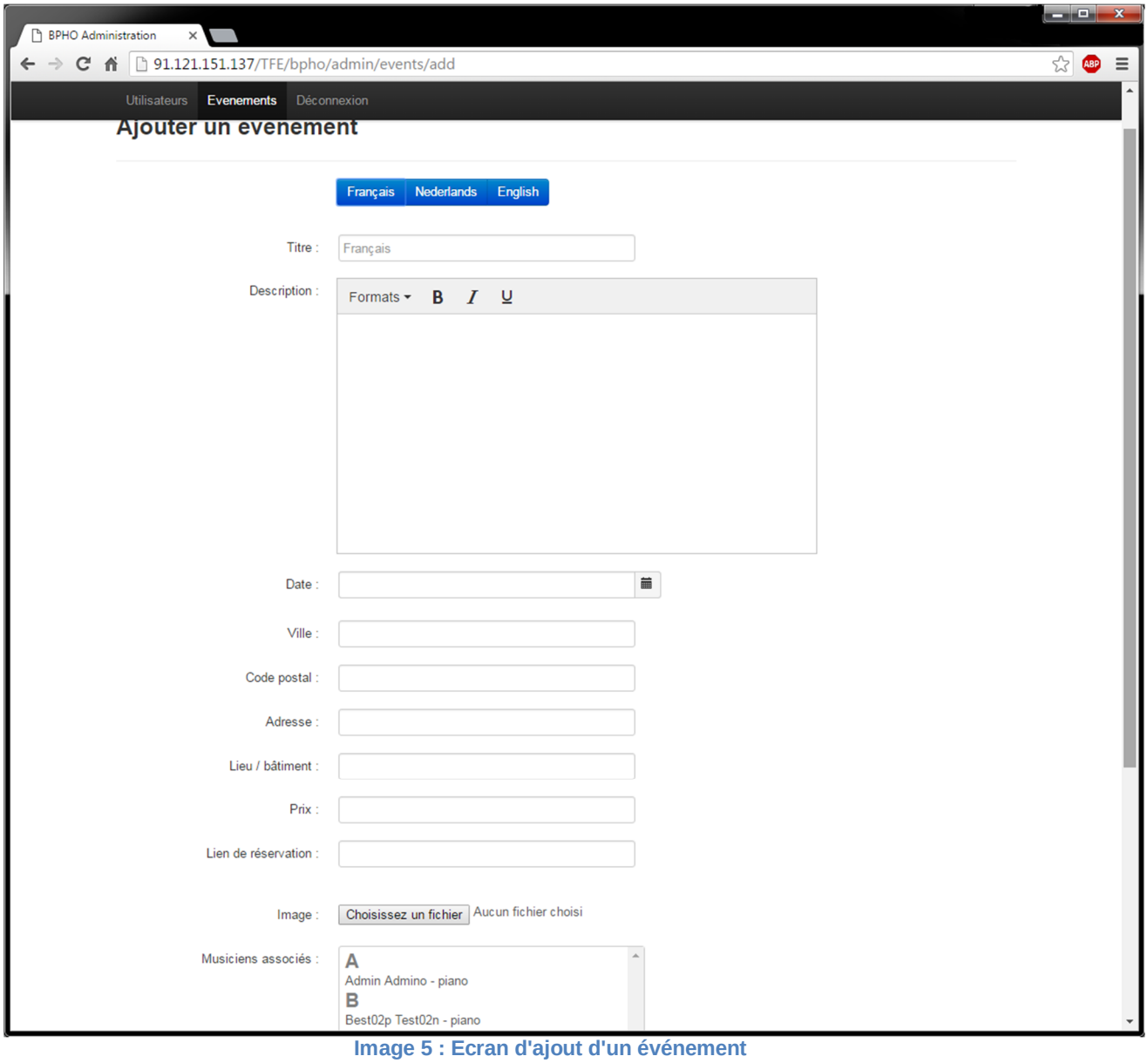

Image 5 : Ecran d'ajout d'un événement

Il est également possible d'ajouter une image à l'événement. Les formats acceptés sont GIF, PNG, et JPG. Lorsqu'une image est sélectionnée, un aperçu est affiché juste au-dessus du champ de sélection. Quand le formulaire est envoyé, l'image est redimensionnée afin d'avoir une taille correcte pour l'affichage sur téléphone mobile, d'économiser de la place sur le serveur, et de réduire la quantité de données envoyées sur le réseau. Si aucune image n'a été sélectionnée, une image par défaut est associée à l'événement, copiée à partir d'une autre image prévue à cet effet présente sur le serveur.

Enfin, on peut ajouter des utilisateurs lors de la création d'un événement. Cela a pour effet d'associer certains utilisateurs à cet événement, dans le but qu'ils reçoivent

les alertes lancées par les administrateurs sur leurs événements dans l'application des musiciens. Cette fonctionnalité se présente sous la forme d'une liste, dans laquelle sont présents tous les utilisateurs, triés par ordre alphabétique. Ils sont ensuite classés selon la première lettre de leur prénom. Ils peuvent être sélectionnés ou désélectionnés par simple clic. Lors de l'enregistrement de l'événement avec la validation du formulaire (champs requis remplis, lien de réservation correspondant à un lien internet, etc…), les associations entre celui-ci et les utilisateurs sélectionnés seront enregistrés dans la table de liaison Event_users.

- **Modification d'un événement**

Sur cette page, il est possible de modifier toutes les données concernant un événement. Lors de l'arrivée sur la page, les informations de l'événement sont reprises dans les champs correspondant du formulaire. De plus, l'aperçu de l'image de l'événement est directement affiché.

Le titre et la description sont différents selon la langue, et les trois versions peuvent donc être modifiées séparément, de nouveau grâce aux trois boutons situés sur le dessus du formulaire. La description dispose toujours de la même interface de traitement de texte permettant de lui appliquer différents styles.

Lors de l'envoi du formulaire, toutes les informations sont évidemment mises à jour dans la base de données. Pour ce qui est des labels, au lieu d'être mis à jour, un nouveau est créé, car il est possible qu'un même label soit utilisé plusieurs fois, mais qu'on ne souhaite le modifié qu'à un seul endroit.

En ce qui concerne les liaisons dans la table Event_users entre un événement et les utilisateurs qui y sont associés, elles sont toutes supprimées, pour ensuite être remplacées par les nouvelles, afin d'éviter les doublons.

c. Application des musiciens

- **Page de connexion**

A l'ouverture de l'application des musiciens, nous arrivons tout d'abord sur la page de connexion. En effet, seuls les musiciens et les membres inscrits du Brussels Philharmonic Orchestra peuvent accéder à cette application. Afin de se connecter, les utilisateurs doivent indiquer leur adresse mail ainsi que leur mot de passe, choisis lors de leur ajout par un administrateur.

Comme pour l'interface d'administration, il est possible de réinitialiser son mot de passe avec la fonctionnalité de "mot de passe oublié". Le déroulement de cette demande de réinitialisation de mot de passe est identique à celle utilisée pour l'interface d'administration. L'adresse mail est vérifiée, un code est généré aléatoirement dans la table Activation, et le mail contenant le lien de réactivation est envoyé à l'utilisateur.

- **Menu**

Lorsqu'un utilisateur est connecté à l'application des musiciens, il a accès à un menu latéral, arrivant par la gauche de l'écran en appuyant sur l'icône à côté du titre de la page.

Depuis ce menu, il est possible d'accéder aux différents écrans de l'application, comme l'agenda général, l'agenda spécifique, la liste des utilisateurs, son profil, ainsi que la messagerie.

- **Agenda général**

Après être passé par l'écran de connexion, nous arrivons sur l'agenda général, qui est similaire à l'agenda présent dans l'application des spectateurs, à la différence près que tous les événements sont affichés, y compris les événements privés de l'orchestre, comme les répétitions, auxquels les spectateurs n'ont évidemment pas accès. De plus, le bouton présent dans l'application des spectateurs amenant vers la page de contact n'est plus nécessaire.

On y retrouve donc les mêmes informations, telles que le titre, l'adresse de l'événement, la date à laquelle il a lieu, ainsi que l'image associée.

- **Agenda spécifique**

Dans cet agenda, seuls les événements auxquels l'utilisateur est associé sont affichés. En effet, lors de la création d'un événement, l'administrateur peut sélectionner les utilisateurs qui y participent ou qui sont concernés. Ici aussi la liste des événements est triée sur la date de ceux-ci, et ceux ayant déjà eu lieu peuvent être affichés en remontant la liste.

- **Liste des utilisateurs**

Sur cet écran, la liste de tous les utilisateurs est disponible. Ceci permet de retrouver le profil de quelqu'un afin de, par exemple, pouvoir le contacter. Cette liste est triée par ordre alphabétique sur le prénom et puis sur le nom.

Dans cette liste sont donc affichés le prénom suivi du nom de chaque utilisateur. Lorsqu'on clique sur un utilisateur dans la liste, son profil est affiché avec en plus de son prénom et de son nom, l'instrument qu'il joue, son adresse mail, et son numéro de téléphone s'il est présent dans la base de données.

- **Profil**

C'est sur cette page que l'utilisateur pourra voir son propre profil. Toutes ses informations seront donc affichées, et il est également possible de les modifier. De plus, si l'utilisateur modifie son adresse mail, une confirmation lui est envoyée sur la nouvelle adresse, après avoir vérifié que la nouvelle adresse mail choisie n'est pas déjà utilisée par quelqu'un d'autre. Si la nouvelle adresse mail est effectivement déjà attribuée, un

message d'erreur est affiché et la modification n'a pas lieu. Comme lors de la création des utilisateurs, certains champs (adresse mail, mot de passe, nom et prénom) sont obligatoires, et il est impossible de sauvegarder les changements tant qu'un de ces champs est vide. Dans ce cas, un message d'erreur est affiché. Bien entendu, le grade de l'utilisateur n'est pas modifiable dans cet écran.

L'utilisateur a également la possibilité d'ajouter une image de profil personnelle, qui sera redimensionnée automatiquement lors de la confirmation des modifications afin d'obtenir une taille normale pour l'affichage sur téléphone. Tant que l'utilisateur n'a pas choisi d'image de profil, une image par défaut lui est attribuée, copiée à partir d'une image de base présente sur le serveur.

- **Messagerie instantanée**

C'est ici que les musiciens et membres du Brussels Philharmonic Orchestra pourront se parler. En effet, les utilisateurs pourront envoyer un message sur un fil de discussion général.

Chaque message est constitué du nom de l'utilisateur, de son image de profil, et du texte qu'il a écrit. De plus, les messages sont triés du plus récent en bas au plus ancien en haut. Seuls les vingt derniers messages sont affichés à l'arrivée sur la page, mais il est tout à fait possible de remonter la discussion.

7. Spécifications techniques

La base de données MySQL et le serveur utilisés sont hébergés sur un serveur OVH. Le choix de cet hébergeur est dû à sa renommée et la qualité de ses services. En effet, OVH est reconnu pour fournir un accès stable à leurs serveurs et dispose d'un très bon service clientèle. De plus, il offre une protection anti DDOS fiable, et le coût de cet hébergement rentre dans le budget du Brussels Philharmonic Orchestra.

Au niveau des technologies utilisées, je me suis servi du JAVA et du XML pour développer les applications mobiles. Pour l'interface web d'administration, j'ai pu utiliser plusieurs langages web appris lors de mon cursus à l'Ephec, tels que le HTML et le CSS pour la partie statique du site, ou le PHP et le JavaScript pour la partie dynamique, sans oublier le SQL pour effectuer les requêtes vers la base de données. J'ai également utilisé des scripts PHP lors du développement des applications, afin de faire la transition entre elles et la base de données.

En ce qui concerne la sécurité, tout d'abord afin d'éviter les injections SQL, toutes les requêtes sont faites avec PDO (PHP Data Object). Ensuite, les mots de passes choisis pour les utilisateurs sont accompagnés d'un sel généré aléatoirement, différent pour chaque membre, et hashés en SHA-256.

Pour ce qui est de l'obtention des applications mobiles, ce qui était prévu à la base était qu'elles soient toutes les deux disponibles sur le Play Store. De plus, comme expliqué précédemment, un QR code devrait dorénavant être imprimé sur les affiches des concerts de l'orchestre, qui mènera directement sur la page de téléchargement de

l'application des spectateurs. Cependant, si un spectateur ou quelqu'un qui n'est pas inscrit comme utilisateur dans la base de données télécharge l'application des musiciens, cette personne ne pourra pas s'y connecter et ne pourra donc rien en faire.

Malheureusement, par soucis de temps (la mise en ligne d'applications sur le Play Store peut prendre plusieurs semaines) il est pour le moment impossible de les télécharger via le marketplace d'Android. C'est également lorsque cette dernière étape sera accomplie qu'un nom de domaine sera réservé, et que le site web et la base de données seront migrés vers un serveur OVH loué au nom du Brussels Philharmonic Orchestra.

8. Spécifications de réalisation

Afin de pouvoir développer les deux applications mobiles, j'ai n'ai pas eu le choix des langages utilisés. En effet, le développement sous Android Studio requiert l'emploi du Java et du XML. Ayant appris le Java lors de la deuxième année de mon cursus à l'Ephec, cela ne m'a pas dérangé, et j'ai pu approfondir mes compétences dans ce langage. Pour le XML, fonctionnant avec un système de balises un peu comme le HTML, il ne s'agit pas d'un langage très complexe et n'est pas difficile à prendre en main. De plus, ce langage permet de customiser les différents widgets de l'application comme on veut.

Pour la réalisation de l'interface d'administration, la seule contrainte de mon client étant d'obtenir un site web simple d'utilisation et disposant d'une présentation claire et précise, j'ai décidé de m'orienter vers les différents langages web appris à l'Ephec, comme expliqué au point précédent.

Après la première rencontre avec les représentants du Brussels Philharmonic Orchestra qui a permis de poser les bases, de détailler les besoins du projet, et suite à laquelle j'ai pu élaborer le cahier des charges, je suis resté en contact avec eux par l'intermédiaire du professeur rapporteur de mon travail de fin d'études, Mr Dewulf. Suite à ses conseils j'ai pu modifier certains aspect du site et des applications, afin de convenir aux besoins de l'orchestre et de respecter leurs désidératas.

9. Spécifications administratives

Au moment de la rédaction de ce rapport, aucun coût n'a été nécessaire à la réalisation de mon travail de fin d'études. Cependant, afin de finaliser entièrement ce dernier et de rendre le site web et les applications totalement fonctionnelles, certains coût seront à prendre en compte :

- La création d'un compte développeur sur le Google Play Store, afin de pouvoir mettre en ligne les applications, coûtant 25€ (coût unique).

- La location d'un hébergement Pro chez OVH, coûtant 7,19€ TTC/mois.

- La réservation d'un nom de domaine, gratuit la première année avec la location de l'hébergement Pro, et coûtant 8,39€ TTC/an.

10. Technologies apprises

Grâce à ce travail de fin d'études, j'ai eu l'occasion d'approfondir les connaissances acquises lors de mon cursus à l'Ephec. Afin de mener à bien ce projet, j'ai employé certains langages et certaines librairies nécessitant un temps d'apprentissage avant de les maîtriser correctement.

Tout d'abord, il y a l'utilisation d'un framework de développement web, CodeIgniter, que j'ai utilisé pour réaliser l'interface d'administration. C'était la première fois et, étant habitué à d'autres méthodes, j'ai mis un certain temps à pouvoir m'adapter à cette nouvelle technologie.

Ensuite, j'ai pu apprendre un nouveau langage lors de l'élaboration du site web : le JavaScript et sa librairie JQuery. Bien que n'étant pas beaucoup présent dans mon travail de fin d'études, j'ai tout de même pu prendre le temps de me renseigner sur son fonctionnement et sur les possibilités qu'il apporte au dynamisme des sites internet. J'ai donc eu l'occasion de lire pas mal de documentation et de découvrir plusieurs tutoriaux sur internet.

En ce qui concerne le développement sur Android, j'en avais déjà eu un aperçu au premier quadrimestre, lors du projet d'intégration. Cependant, la réalisation des deux applications pour le Brussels Philharmonic Orchestra m'a permis de pousser mon apprentissage dans ce domaine. J'ai donc appris un certain nombre de nouveaux aspects propres à la programmation sous Android, comme l'utilisation de liste personnalisée, de layout communs à plusieurs écrans de l'application avec une interaction entre eux, ou même l'interaction entre plusieurs applications différentes. De plus, j'ai également appris à utiliser des librairies externes, qui peuvent parfois s'avérer très utiles. Par exemple, j'utilise dans ce travail de fin d'études la librairie apache, permettant de se connecter à la base de données afin d'accéder aux scripts PHP, de les exécuter, et d'en récupérer les résultats.

11. Difficultés rencontrées

Tout au long de la réalisation de mon travail de fin d'études, j'ai rencontré de nombreuses difficultés, bien que beaucoup n'étaient que de simples erreurs d'inattention, souvent dues à la fatigue et à un manque de concentration de ma part. Certains de ces problèmes m'ont pris un temps considérable à corriger, c'est pourquoi je vais décrire seulement les plus importants dans cette partie du rapport.

- **Rythme de travail**

La première difficulté, et la plus dure à surpasser, fût l'adaptation au rythme de travail. En effet, le fait de devoir réaliser un stage et un travail de fin d'études en même temps est vraiment difficile. Après la journée de travail dans l'entreprise de mon stage, je devais rentrer chez moi, et trouver, tant bien que mal, le temps et surtout la motivation de travailler. C'est un rythme fort différent de celui d'un étudiant qui doit uniquement se rendre en cours, ayant un horaire avec des pause, et pouvant se permettre de n'écouter que d'une oreille ce que le professeur explique à cause du manque de sommeil.

Lors de cette période de stage, il fallait tout le temps être capable de fournir son maximum afin de travailler correctement, ce qui entraînait une grande fatigue le soir lorsque je rentrais chez moi. C'est pourquoi je ne travaillais pas beaucoup en semaine, à peine deux heures par jour lorsque j'étais assez motivé. J'ai donc effectué la majeure partie du travail pendant les weekends, car j'avais eu le temps de me reposer et était donc prêt à avancer sur mon travail de fin d'études du samedi matin au dimanche soir. De plus, j'ai eu de temps en temps l'occasion, durant mon stage, de travailler sur ce projet. Cela n'arrivait pas souvent, mais ça m'a permis d'avancer correctement grâce à une bonne ambiance de travail.

Au final, j'ai réussi à acquérir une bonne méthode et un bon rythme de travail, même si le début de la réalisation de ce travail de fin d'études fût vraiment très long à démarrer.

- **Applications multi langues**

Comme le Brussels Philharmonic Orchestra regroupe de nombreux musiciens de nationalités différentes, il était important de pouvoir proposer un contenu disponible en plusieurs langues, à savoir le français, le néerlandais, et l'anglais. Pour ce qui est du contenu statique des applications, il y a une fonctionnalité très pratique dans la création d'applications dans Android Studio, qui est les "fichiers de ressources". En effet, ces fichiers peuvent avoir plusieurs utilités, comme l'adaptation aux différentes versions du système d'exploitation présent sur le téléphone, au sens dans lequel est tenu l'appareil (portrait ou paysage), à la disponibilité des différents réseaux (data 3G/4G, Wi-Fi, Bluetooth, etc…), au pays dans lequel se trouve l'utilisateur, ou encore selon la langue dans laquelle est réglée le Smartphone. Dans ce dernier cas, un fichier est défini pour chaque langue choisie et contient les différents textes présents dans l'application. Il suffit alors d'appeler la ressource correspondante dans le code Java.

Cependant, au début de mes recherches je ne savais pas comment rendre le contenu dynamique des deux applications disponible en plusieurs langues. J'ai pensé à plusieurs idées, mais je n'en trouvais aucune qui était performante et non redondante.

Les premières idées auxquelles j'ai pensé étaient, notamment, enregistrer les différentes traductions dans la table Events, ou créer une table différente pour chaque langue, mais ces solutions, qui n'en étaient pas vraiment, allaient évidemment à l'encontre des bonnes pratiques de conception de bases de données.

C'est au cours de mon stage que j'ai trouvé la solution : l'utilisation de "dictionnaires de labels". La société où se déroulait mon stage utilisait donc ces fichiers XML pour que leurs programmes soient disponibles en plusieurs langues. Il s'agit de fichiers contenant des balises similaires à du HTML, ressemblant à ceci :

```
<Label>
    <ID>501283</ID>
    <FR>texte en Français</FR>
    <NL>tekst in het Nederlands</NL>
    <EN>text in English</EN>
</Label>

<Label>
    <ID>501284</ID>
```

Image 6 : Exemple de label dans un fichier XML

J'ai donc adapté cette solution à mon travail de fin d'études. J'ai créé une table Labels dans la base de données, contenant les quatre champs nécessaires, et dans la table Events, j'ai modifié le type des champs "title" et "description" en Integer. Ces deux champs font donc maintenant référence à l'ID de leur label respectif pour chaque événement.

Malheureusement, bien que j'aie trouvé la solution, cela m'a posé un autre problème : j'ai dû refaire une partie du code, car le fonctionnement était dorénavant différent. Au lieu de simplement récupérer les informations d'un événement, il fallait maintenant aussi récupérer les labels correspondant, et donc faire une requête SQL supplémentaire.

Cependant, cette étape fût assez rapide. Au final, en plus du contenu statique des applications, le titre et la description des événements sont également disponibles en français, en néerlandais, et en anglais. Il est aussi important de préciser que, si la langue voulue par l'utilisateur, par exemple de l'espagnol, ou bien si la traduction n'est pas disponible, l'anglais sera la langue par défaut.

- **Plugin Jquery**

Lors du développement du site web, j'ai voulu, pour la création des événements, utiliser un plugin JQuery, un « dateTimePicker ». Celui-ci consistait à faire apparaître un petit calendrier, dans lequel il est possible de sélectionner une date, et sous lequel un bouton permet d'afficher une horloge digitale afin de choisir une heure. Ce plugin est présent sur l'interface d'administration, mais ce ne fût pas sans peines.

En effet, au début de son ajout, je ne parvenais pas à le faire fonctionner car je n'avais encore aucune connaissance en JavaScript. Je me suis donc renseigné et ai appris les bases de ce langage extrêmement pratique, pour être confronté à un deuxième problème : la langue du calendrier. Comme c'était la première fois que j'utilisais un plugin externe lors de la création d'un site web, j'avais du mal à comprendre comment interagir avec. Il ne s'agissait que d'un détail d'affichage, mais je ne parvenais pas à mettre les noms des jours et des mois en français.

Après avoir compris que je devais créer un fichier de localisation comportant la traduction française des noms de jours et de mois voulus, j'ai réussi à obtenir un résultat satisfaisant. Cependant, le premier jour de la semaine se trouve être le dimanche, et je n'ai malheureusement pas trouvé comment changer ce petit détail.

J'ai également choisi de désactiver l'affichage des secondes dans l'horloge digitale, car une telle précision n'est pas nécessaire pour l'horaire d'un événement. De plus, cela ne ferait que surcharger l'interface graphique.

- **Adaptation au framework**

Ensuite, le problème suivant consistait à m'adapter au template que j'ai choisi pour l'interface d'administration de la base de données. Cette étape était particulièrement complexe, car elle nécessitait de comprendre certaines fonctions, comment il fallait employer les variables "super globales" en PHP, ou encore comment s'agençaient les différentes parties du site et comment les données passaient d'une page à l'autre. En effet, les méthodes employées ne m'étaient pas familières.

Par exemple, pour afficher la liste des utilisateurs, il faut passer par un fichier appelé "admin_users.php". Dans ce fichier, qui contient la classe Admin_users, il y a quatre fonctions en plus du constructeur : index, add, update et delete. En fonction de l'URL, la fonction appelée est différente. Dans le cas de la liste, c'est la fonction index qui est appelée. Elle se charge d'aller cherche la liste des utilisateurs en base de données, éventuellement sur base du champ de recherche, et de mettre cette liste dans une cellule de la variable $data qui est un tableau associatif. Ensuite, la page list.php, située dans views/users est chargée et la variable $data lui est envoyée. Dans ce fichier, nous avons accès aux variables qui sont les cellules de la variable $data, mais la variable $data en elle-même n'est pas accessible.

Enfin, les utilisateurs peuvent être affichés dans un tableau HTML. Voici un aperçu du code présent dans controllers/admin_users.php afin de mieux visualiser la problématique :

```
}else{

    $filter_session_data['search_string_selected'] = null;
    $this->session->set_userdata($filter_session_data);

    $data['search_string_selected'] = '';

    $data['count_users']= $this->users_model->countUsers();
    $users = $this->users_model->getUsersOrderByFirstName('', $config['per_page'],$limit_end);
    $config['total_rows'] = $data['count_users'];

}

foreach ($users as &$user) {
    $user['droit'] = $this->users_model->getUserDroit($user['id']);
}
$data['users'] = $users;

$this->pagination->initialize($config);

//load the view
$data['main_content'] = 'admin/users/list';
$this->load->view('includes/template', $data);
```

Image 7 : exemple de code 1

Et ici un aperçu du code présent dans views/users/list.php, où l'on peut voir que la variable $users est récupérée. Celle-ci correspond à la variable $data['users'] de l'image précédente :

```
            <th class="red header">E-mail</th>
            <th class="red header">Date d'inscription</th>
            <th class="red header">Dernière connexion</th>
            <th class="red header">Téléphone</th>
            <th class="red header">Actions</th>
        </tr>
        </thead>
        <tbody>
        <?php
        foreach($users as $row) {
            echo '<tr>';
            echo '<td style="display:none;">'.$row['id'].'</td>';
            echo '<td>'.$row['firstName'].'</td>';
            echo '<td>'.$row['name'].'</td>';
            echo '<td>'.$row['email'].'</td>';
            echo '<td>'.$row['dateRegister'].'</td>';
            echo '<td>'.$row['dateLastConnect'].'</td>';
            echo '<td>'.$row['phone'].'</td>';
            echo '<td class="crud-actions">';
            echo '<a href="'.site_url("admin").'/users/update/'.$row['id'].'" class="btn btn-info">Editer</a>';
            if ($row['droit'] != 1) {
                echo '<a href="' . site_url("admin") . '/users/delete/' . $row['id'] . '" class="btn btn-danger" >Supprimer</a>';
            }
            echo '</td>';
            echo '</tr>';
        }
        ?>
        </tbody>
    </table>


    <?php echo '<div class="pagination">'.$this->pagination->create_links().'</div>'; ?>

    </div>
</div>
```

Image 8 : Exemple de code 2

Ceci n'est qu'un exemple de l'utilisation de ce framework. N'ayant encore jamais développé de site web à l'aide du PHP orienté objet, cela m'a demandé beaucoup de temps avant de comprendre l'entièreté du framework.

- **Messagerie instantanée**

La messagerie de l'application des musiciens est une fonctionnalité qui m'a pris vraiment beaucoup de temps, car c'était quelque chose de tout à fait nouveau. En effet, afin de réaliser cette partie, il fallait que le contenu de la messagerie soit dynamique, d'où l'appellation "instantanée". Malheureusement, les nombreuses tentatives que j'ai essayées pour la faire fonctionner n'ont pas porté leurs fruits.

Comme je n'avais plus le temps de régler ce problème, j'ai décidé de laisser une messagerie non instantanée, qu'il faut donc rafraîchir afin de recevoir les éventuels nouveaux messages.

12. Améliorations possibles

Comme dans tout travail, il est toujours possible d'y apporter certaines améliorations. Celles-ci n'ont pas pu être mises en place essentiellement par manque de temps, mais peuvent cependant être conseillées.

La principale amélioration possible de ce travail de fin d'études serait, comme expliqué ci-dessus, de rendre la messagerie de l'application des musiciens réellement instantanée. Cela rendrait cette fonctionnalité plus facile à utiliser, plus agréable, et plus dynamique.

Une autre amélioration pourrait être la possibilité aux utilisateurs de l'application des musiciens de pouvoir discuter entre eux grâce à une messagerie privée. Bien que les représentants du Brussels Philharmonic Orchestra souhaitent qu'il n'y ait qu'un seul fil de discussion au sein de l'application, il est possible que cette fonctionnalité voie son utilité se déclarer dans le futur.

Au niveau des améliorations graphiques, les couleurs pourraient être modifiées, ou le nombre de nouveaux messages dans la discussion pourrait être affiché de façon dynamique. Ce sont des aspects sur lesquels je ne me suis pas attardé, mais qui pourraient éventuellement apporter un plus aux applications.

Enfin, il est sûrement possible de rendre le code plus performant et plus léger à certains endroits. En effet, par exemple, limiter le nombre et le volume des transitions de données entre la base de données et l'interface d'administration ou les applications mobiles permettrait de gagner en performance et en rapidité d'exécution.

13. Conclusion

Lors de la réalisation de ce travail de fin d'études, j'ai pu me rendre compte de toutes les étapes nécessaires au développement d'un tel projet. Une demande parfois simple d'un client, comme l'envie de s'introduire sur le marché des applications mobiles dans ce cas-ci, nécessite la réflexion et l'analyse des différents besoins du client, afin de réaliser un produit qui lui convient et répond à ses attentes.

L'idée de départ du Brussels Philharmonic Orchestra de vouloir être présent dans le domaine mobile sur Smartphone est une situation de plus en plus commune. En effet, les avantages qu'offre ce marché, bien qu'ils peuvent différer grandement d'une société ou d'une entreprise à une autre, sont nombreux et non négligeables. De plus, les possibilités offertes par les technologies mobiles sont très vastes, il y a de ce fait de nombreuses pistes à explorer.

Il a donc fallu, avant de passer à l'étape de développement de l'interface d'administration et des applications Android, analyser les demandes formulées par le client lors de notre première rencontre au premier quadrimestre.

Suite à cela, j'ai pu élaborer le cahier des charges reprenant l'entièreté des fonctionnalités requises ainsi que les spécifications techniques et administratives. Après validation de ce cahier des charges, je me suis occupé de l'élaboration de la base de

données, validée par Mr Lambeau, puis je suis passé à l'étape du développement, en commençant par l'application des spectateurs, pour ensuite réaliser l'interface d'administration de la base de données, et enfin développer l'application des musiciens.

En ce qui concerne les moments un peu lents dans le développement, majoritairement dus au rythme de travail ou à quelques bugs dans le code, il m'a parfois fallu beaucoup de temps afin de trouver les solutions appropriées. En effet, la fatigue accumulée avec le stage a causé une certaine perte de concentration et donc rallongé le temps de développement. Cependant, mise à part la messagerie qui n'est pas instantanée, toutes les fonctionnalités requises ont été implémentées.

De plus, j'ai eu l'occasion de bénéficier des conseils de Mr Dewulf, mon professeur rapporteur de travail de fin d'études, que je tiens à remercier et qui m'a grandement aidé dans la réalisation de ce projet, notamment pour ce qui est de la méthode de travail et pour le feedback sur le travail effectué.

Au final, ce travail de fin d'études m'a beaucoup apporté, autant au niveau du rythme de travail, qui s'est amélioré jusqu'au bout, des compétences acquises et renforcées, telles que le développement sur Android ou l'apprentissage du JavaScript/Jquery, et de mes capacités personnelles. La combinaison du stage et du travail de fin d'études était réellement difficile, mais j'en sors avec plus de connaissances et de confiance en moi, car je me sais capable de réaliser ce genre de projet.

Pour finir, je suis très content d'avoir eu l'occasion de créer ces deux applications mobiles et l'interface web d'administration qui va avec pour le Brussels Philharmonic Orchestra, car cela m'a permis de confirmer mon intérêt pour le développement, et plus particulièrement dans le domaine mobile.

14. Bibliographie

AWT, *Fiche méthodologique : Rédiger un cahier des charges*, http://www.lomag-man.org/e_business/e_business_docu_telech/cahier_charge_ebusinessP6.pdf, consulté le 21/05/2016

Espiau F., *Créez des applications pour Android*, https://openclassrooms.com/courses/creez-des-applications-pour-android, consulté le 25/02/2016

Grant Allen, *L'Art du Développement Android*, 4e Edition, Pearson, France, 2012

McFarland D. S., *JavaScript & Jquery : The Missing Manual*, 3e Edition, O'Reilly Media, Londres, 2014