

Environnement Numérique de Travail pour enseignants

Implémentation de l'application mobile

Réalisé par : Jean-Emmanuel Baillat et Kylroil Gaubert

Encadré par : Mr. Kante, professeur et responsable de formation

Sommaire

<u>Introduction</u>	p.5
<u>I. Développement du projet en commun</u>	
1. Fonctionnalités à implémenter	p.6
2. Ensemble des contraintes	p.7
3. Modélisation de la base de données et répartition des tâches	p.8
<u>II. Les technologies choisies</u>	
1. Les tests	p.10
2. Prototype de l'interface	p.10
<u>III. Création de l'application</u>	
1. Technologie Android	p.12
2. Développement de l'application	p.14
3. Les améliorations envisageables	p.16
<u>Conclusion</u>	p.17
<u>Annexes</u>	p.18

Table des abréviations

UE	Unité d'Enseignement
ENT	Environnement Numérique de Travail
CM	Cours Magistral
TP	Travaux Pratiques
TD	Travaux Dirigés
PDF	Portable Document Format
SGBD	Système de Gestion de Base de Données
XML	Extensible Markup Language
JDK	Java Development Kit

Introduction :

Le Master en Informatique de l'Université Blaise Pascal propose lors du deuxième semestre une unité d'enseignement (UE) intitulée Génie logiciel par la pratique. Celle-ci propose aux étudiants de suivre un projet durant 3 mois, où ils pourront mettre en œuvre ce qu'ils auront appris. Ils devront créer des services spécifiques suivant un cahier des charges qu'ils auront rédigé. Deux projets furent proposés, le premier : créer un service de partage de photos et le second : un environnement numérique de travail (ENT) pour enseignants. Les étudiants qui ont choisi le second sujet et qui formeront l'équipe de conception sont les suivants : Jean-Emmanuel Baillat, Rémi Colas, Zakaria El Idrissi, Kylroil Gaubert, Rachid Jijane, Cyril Pannier, Reda Samlali et Karim Sayabou. Ils seront supervisés par Mr. Kante, professeur et responsable de notre formation.

Le travail demandé se déroulera en deux phases. La première sera de définir, avec l'ensemble des groupes, le cahier des charges, les services à implémenter et la base de données correspondante. La seconde phase consistera à se répartir les différentes tâches de l'implémentation parmi les membres de l'équipe de conception. Les étudiants suivant créeront les algorithmes : Rachid Jijane, Rémi Colas, Cyril Pannier. Le serveur, le site web et les services seront implémentés par : Zakaria El Idrissi, Karim Sayabou et Reda Samlali. La dernière équipe, constituée de Jean-Emmanuel Baillat et Kylroil Gaubert, concevra une application mobile en Android.

Ayant choisi le dernier groupe, nous allons devoir dans un premier temps apprendre cette nouvelle technologie pour l'utiliser dans un cas concret. Il sera ensuite nécessaire de tester les technologies utilisables pour une application Android et d'en faire ressortir les points positifs et négatifs. Nous choisirons alors les plus adaptées. Puis, nous soumettrons nos choix technologiques aux autres groupes et nous nous mettrons d'accord pour utiliser les mêmes. Avec l'aide d'un logiciel, nous créerons un prototype de l'application pour définir son architecture et les différents éléments que nous utiliserons. Après quoi, nous mettrons en place un système de travail pour qu'on puisse implémenter avec plus de rapidité et d'efficacité. A ce moment-là, nous serons confrontés à différentes possibilités d'implémentation du code dont nous commenterons les choix. De plus, nous listerons les services non implémentés dans l'application et nous en expliquerons les raisons. Enfin, nous aborderons la mise en commun des différents groupes et nous conclurons avec l'expérience que le projet nous a apportée ainsi que la remise en question de certaines décisions.

I. Développement du projet en commun

1. Fonctionnalités à implémenter

L'équipe de conception de l'environnement numérique de travail doit avant toute chose réaliser un cahier des charges pour définir les différentes fonctionnalités du projet et l'ensemble des contraintes dont il faudra tenir compte.

Nous allons maintenant aborder les fonctionnalités de l'Environnement Numérique de Travail.

Le système pourra gérer 3 types d'utilisateur :

- Un professeur : c'est l'utilisateur le plus courant, cet Environnement Numérique de Travail lui est directement destiné pour son travail. C'est à travers notre outil qu'il pourra modifier son emploi du temps et les notes de ses élèves.
- Un responsable de filière : en plus d'avoir les mêmes fonctionnalités d'un professeur, il pourra mettre en place les dates d'examens de la filière dont il a la charge. C'est aussi lui qui pourra générer automatiquement l'emploi du temps de sa filière grâce aux services de l'application.
- Un administrateur : celui-ci veillera au bon fonctionnement de l'application, il modifiera la base de donnée selon les besoins de l'Université (ouverture de filières, nouveaux professeurs, etc.)

Lorsqu'un professeur se rend sur le site ou sur l'application, il arrive sur la page d'accueil où il devra se connecter pour accéder au contenu personnalisé.

Les identifiants du professeur lui seront envoyés par l'administrateur. Si toutefois, il les avait perdus, il lui sera possible de demander une réinitialisation du mot de passe et qu'on le lui renvoie.

Quand un professeur est authentifié, il peut :

- Afficher son planning avec les différentes manipulations possibles suivantes :
 - consulter son emploi du temps ainsi que les informations de chaque séance (la salle, le groupe d'étudiant, etc.).
 - annuler le cours de son choix.
 - transférer son cours (avec un créneau ou une salle différente).
- Afficher l'emploi du temps d'une salle avec les interactions possibles suivantes :
 - réserver une salle pour un cours si le créneau correspondant est libre.
 - supprimer un ou plusieurs de ses cours pour la salle correspondante.
- Réserver une salle en remplissant des informations sur le type de la salle, le créneau horaire souhaité, l'effectif que la salle peut contenir, sa localisation et l'équipement qu'elle possède. Une requête sera lancée dans la base de données et si des possibilités existent, le professeur choisira la plus adaptée pour lui.
- Ajouter ou supprimer des contraintes horaires durant lesquelles le professeur sera indisponible.
- Une section profil lui sera dédiée pour qu'il puisse consulter ou modifier les données suivantes ses informations personnelles.
- Le professeur pourra également ajouter les unités d'enseignement qu'il souhaite enseigner durant le semestre, ou bien en supprimer. Pour chacune d'entre elles, il pourra préciser s'il souhaite enseigner le cours magistral (CM), les travaux pratiques (TP) ou les travaux dirigés (TD). Il pourra sélectionner plusieurs types d'enseignement.
- Concernant la gestion des notes, le professeur pourra afficher la liste des étudiants inscrits à l'UE qu'il aura au préalable sélectionnée, avec la note de chacun dont il pourra en modifier la valeur, ajouter une nouvelle note, ou les exporter toutes en format : PDF (Portable Document Format).

Après s'être authentifié, un responsable de filière aura les mêmes fonctionnalités qu'un professeur, mais avec quelques-unes supplémentaires qui lui seront spécifiques :

- Ajouter ou supprimer les examens sur l'emploi du temps.
- Vérifier que les professeurs aient attribué une note à tous leurs élèves.
- Exporter en format PDF les notes des élèves inscrits à sa filière.
- Décider de la date de convocation du jury pour valider les notes des étudiants de sa filière.
- Choisir le délai durant lequel les professeurs choisiront leurs unités d'enseignement pour le prochain semestre.
- Générer l'emploi du temps de deux manières différentes :
 - manuellement : avec un bouton à cet effet.
 - automatiquement : à une date qu'il aura sélectionnée.
- Indiquer la semaine où se dérouleront les examens terminaux.

L'administrateur pourra, quant à lui, modifier toute la base de données. Il faut qu'il puisse mettre à jour les différentes filières et unités d'enseignement proposées par l'Université, ainsi que l'arrivée ou départ de professeurs. De plus, l'administrateur pourra modifier l'emploi du temps si cela est nécessaire. L'Université pouvant développer ses locaux, il pourra mettre à jour les équipements et les salles disponibles. Enfin, il gèrera les comptes qui auront accès aux services de l'Environnement Numérique de Travail, en veillant à bien affecter les droits appropriés à chaque professeur.

[cf. Annexes]

2. Ensemble des contraintes

Pour qu'un cours soit réservé, il faut toujours prendre en compte l'effectif du groupe et la place disponible afin d'optimiser la distribution des salles. L'un des points importants est l'affectation des unités d'enseignement aux professeurs telle qu'ils puissent être plusieurs à enseigner la même matière. Des contraintes géographiques doivent être gérées, comme celles de pouvoir réserver deux salles côte à côte pour un grand groupe d'étudiants, et éviter que deux professeurs soient affectés dans une salle en même temps. Chaque unité d'enseignement a un besoin spécifique en équipement, et il est nécessaire de le prendre en compte. Un professeur ayant des indisponibilités régulières durant l'année, peut le signaler pour qu'aucune affectation réalisée par l'algorithme ne se fasse durant ces périodes. De plus, il peut sélectionner l'unité d'enseignement et le type de cours (CM, TD ou TP) qu'il souhaite enseigner. Enfin, il définit le coefficient des notes qu'il attribuera à chaque élève.

[cf. Annexes]

Les contraintes des services proposés :

L'interface que nous implémentons dans cette application doit être simple d'utilisation et intuitive, en effet, l'utilisateur doit pouvoir repérer rapidement les fonctionnalités disponibles pour chaque service du projet et l'utiliser intuitivement. Nous devons également respecter une certaine homogénéité entre l'interface des deux applications, web et mobile, telle que l'apparence (couleur, disposition des éléments, etc.) et les fonctionnalités proposées. Celles-ci devront être identiques et au même endroit afin d'éviter à l'utilisateur de les chercher. Enfin, l'application devra offrir une réponse à l'utilisateur dans un temps acceptable.

Les contraintes de réalisation :

Les différentes contraintes de réalisation auxquelles nous serons confrontés sont :

- Contraintes sur l'analyse du projet. Tout d'abord, les principales fonctionnalités du projet doivent être mise en place pour une utilisation minimale du service par ses utilisateurs. C'est par la suite qu'on ajoutera d'autres manipulations, sans pour autant modifier la structure déjà existante.
- Contraintes légales. Chaque logiciel possède une charte d'utilisation, c'est pourquoi il est nécessaire d'en prendre acte avant d'en faire usage. Sachant que l'utilisation de logiciels piratés est punie par la loi et que notre application se veut être gratuite, nous allons privilégier des logiciels Open Source pour notre projet. De plus, il existe des contraintes légales sur des algorithmes déjà implémentés. Avant d'en faire usage, il nous sera nécessaire de vérifier s'ils sont libres de droit.

3. Modélisation de la base de données

Présentation du Modèle Logique de Données [cf. Annexes] :

Une faculté possède plusieurs départements de recherche ou d'enseignement. Ces départements ont des promotions et ces dernières sont divisées en classes, puis en groupes, les élèves étant affectés à ces groupes.

Les professeurs sont reliés à des contraintes d' et des préférences d'enseignement qu'ils auront eux-mêmes définies. Ils ont tous des comptes d'utilisateur de l'ENT. Certains d'entre eux sont responsables d'une promotion. Ils possèdent aussi une ou plusieurs compétences et ils travaillent dans une ou plusieurs facultés.

Les étudiants sont inscrits dans des UE, ces dernières étant constituées de modules, eux-mêmes constitués de séances (CM, TD ou TP). Le module réunit des compétences et nécessite du matériel pour être enseigné.

Chaque séance possède des modalités d'examen. Une ou plusieurs notes sont affectées à chaque étudiant dans le module qu'ils auront suivi pendant le semestre, et selon le type de séance, les modalités changent.

Des équipements sont présents dans les salles. On affecte des professeurs, des salles et des séances à un groupe d'étudiants.

Une base de données interne, externe, ou les deux ?

La base de données utilisée est une base de données externe. Nous avons la possibilité d'utiliser une base de données interne, celle de SQLite qui nécessite peu de mémoire lors de son exécution. Malheureusement, ce ne serait qu'une copie de la base de données du site web, et avoir une copie modifiée sur chaque appareil mobile serait ingérable (il faudrait que la base stockée sur chaque appareil soit identique, or c'est impossible car tous les appareils ne seront pas connectés en permanence à internet).

Quel SGBD utiliser ?

Notre base de données externe reposera sur le Système de Gestion de Base de Données (SGBD) utilisé pour le site web, il s'agit de PostgreSQL. Ce SGBD a été choisi pour manier efficacement de gros volumes de données en JAVA.

Accès à la base de données :

Pour que notre application puisse accéder aux données de la base externe, nous allons utiliser un web service. Celui-ci sera utilisé dans les pages de l'application mobile pour récupérer les données. Le web service sera composé des fonctions CRUD (Create, Read, Update and Delete), ainsi, l'application n'aura qu'à passer des paramètres aux fonctions du web service, sans que l'on s'occupe de son implémentation.



Figure 1. Représentation des flux de données

Transfert des données :

Pour utiliser ce web service et accéder à la base de données dans le but d'en récupérer des informations, le format de transfert des données sera XML (Extensible Markup Language). En effet, le Middleware entre l'application et la base de données sera géré par le Framework Hibernate qui s'occupe de générer automatiquement un XML pour l'échange de données.

II. Les technologies choisies

1. Les tests

Le but des tests est de nous permettre de choisir les technologies les plus adaptées à notre besoin. Pour une application mobile, la taille de l'écran est à prendre en considération afin de choisir les boutons ou la police. Il faut que le résultat final soit ergonomique et harmonieux, tout en respectant des points communs avec le site web. Il est nécessaire grâce à un logiciel d'édition de prototype d'interface, de définir le futur design et l'architecture de l'application. Ensuite, nous simulerons des actions que notre application mobile devra gérer. Celles-ci concernent l'accès à la base de données à travers des services. Elle doit être accessible afin que le professeur puisse changer son emploi du temps, ses préférences et les notes de ses élèves.

2. Prototype de l'interface

Architecture de l'application :

Une fois la page de connexion passée, nous arrivons sur la page d'accueil. A partir de là, il sera possible via un menu de se déplacer entre les pages suivantes : accueil, réservation de salle, emploi du temps professeur, emploi du temps salle, les notes, le profil et la déconnexion. Il est possible d'accéder à la page des préférences des UE du professeur depuis son profil.

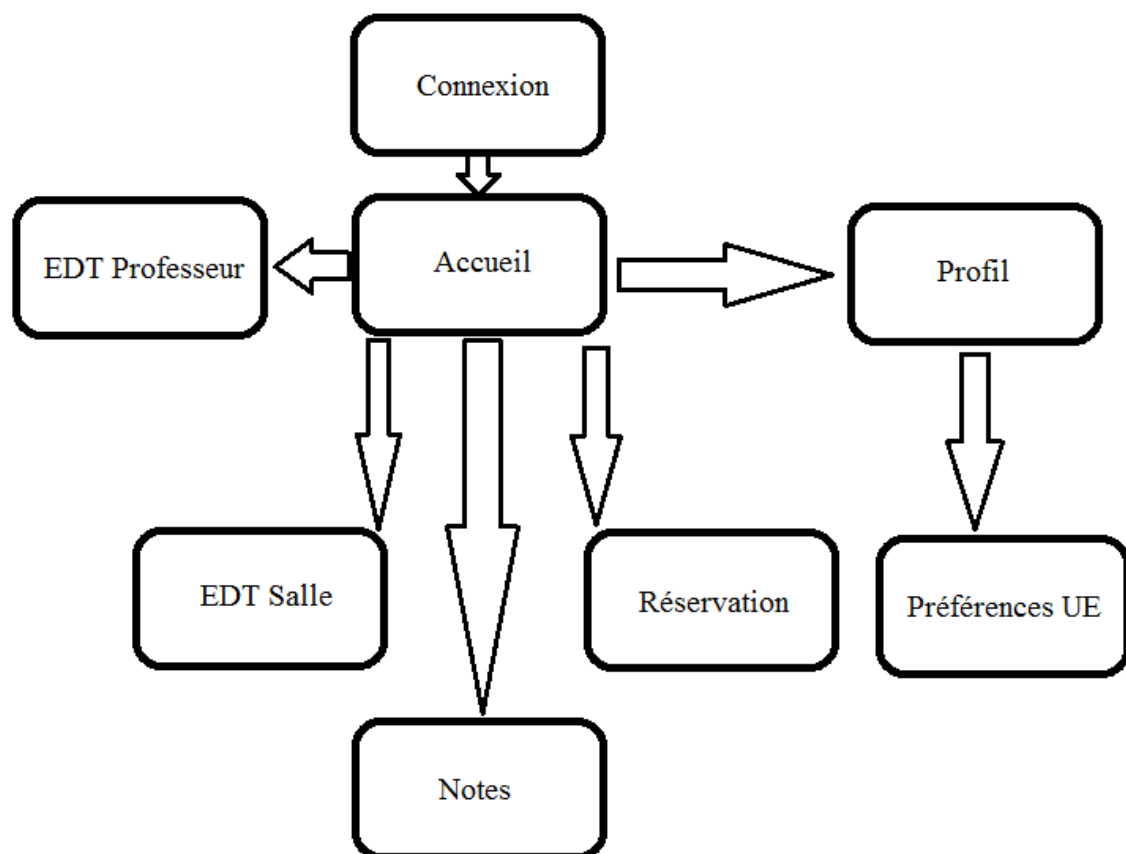


Figure 2. Architecture depuis la page de Connexion

Allure générale de l'application :

L'application permet à ses utilisateurs de consulter et de gérer leur emploi du temps. Pour que cela soit plus facile de naviguer d'une page à une autre, un bouton de retour à l'accueil est positionné en haut à gauche de chaque page, et nous avons créé un menu des liens principaux, accessible par un glissé du doigt de gauche à droite depuis le bord gauche de l'écran. Pour concilier la taille de l'écran avec l'avantage d'afficher un calendrier complet, l'application s'utilisera uniquement en position paysage (sauf la page de connexion). Ainsi, on peut parcourir l'emploi du temps en glissant très peu. La taille moyenne d'un écran de téléphone nous empêche d'afficher un emploi du temps supérieur à une semaine. Afficher jour par jour les cours d'un professeur ne lui permet pas une vue globale de sa semaine. Nous allons donc afficher l'emploi du temps d'un professeur sur une semaine. De plus cela va lui permettre de le consulter ou de le modifier sans nécessairement zoomer.

Page de connexion :

Lors du lancement de l'application mobile, l'utilisateur va directement accéder à la page de connexion en mode portrait. Celle-ci lui permettra de se connecter à son compte et d'ensuite accéder aux services de l'application. Si l'utilisateur ne se souvient pas de son mot de passe, il est possible qu'il soit réinitialisé puis renvoyé par mail à l'utilisateur

Page d'accueil :

On suppose que l'utilisation première de l'application soit de consulter son emploi du temps, c'est pourquoi la page d'accueil s'ouvre sur la semaine en cours du professeur sans possibilité de modification.

Profil :

Un affichage des données déjà enregistrées du professeur avec la possibilité de les modifier avec le bouton correspondant. Un bouton "Liste des préférences UE" redirige l'utilisateur sur une activité où il peut manipuler les UE qu'il souhaite enseigner.

Préférences des unités d'enseignement :

On peut supprimer une UE de manière intuitive en cliquant dessus ou en ajouter en faisant une recherche via un champ conçu pour ça.

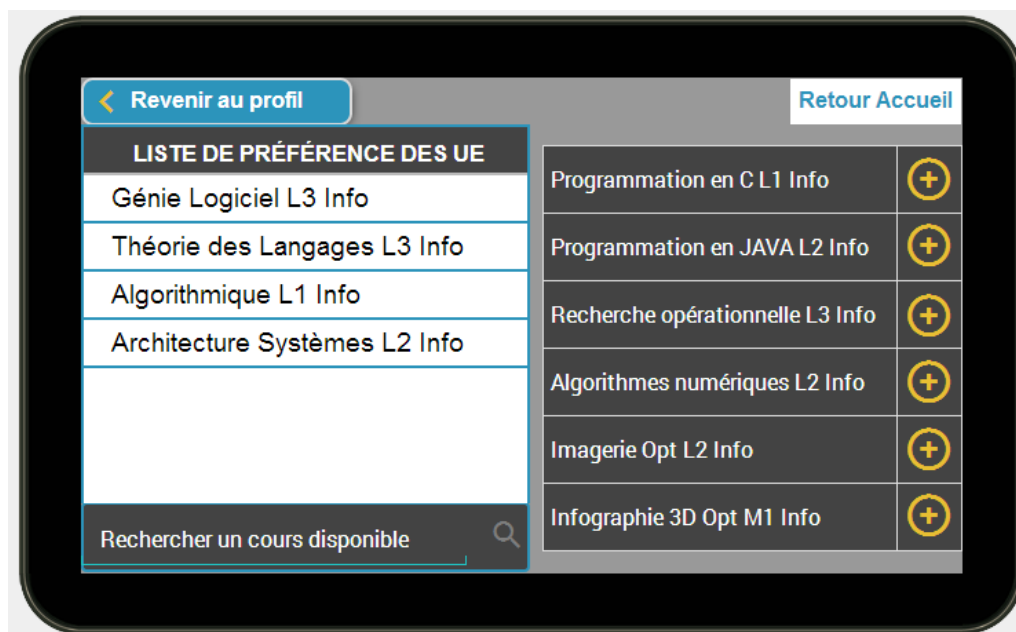


Figure 3. Préférences des UE

Emploi du temps pour un professeur :

Les manipulations possibles sur l'emploi du temps d'un professeur se font en cliquant sur le cours correspondant. Par deux boutons affichés sur un "touché", il peut soit transférer son cours, soit le supprimer. Cette méthode a été privilégiée pour que cela soit plus intuitif à l'utilisateur et que cela lui évite de remplir des formulaires.



MENU PRINCIPAL	U TEMPS - PROFESSEUR		SEMAINE 6 - FEVRIER	
ACCUEIL	MARDI	MERCREDI	JEUDI	VENDREDI
PLANNING SALLE		Génie Logiciel M1 Info 9011 Dpt Physique	 	Génie Logiciel M1 Info 9011 Dpt Physique
RÉSERVER SALLE	Info Grp 2 Dpt Physique			Algo L1 Info Grp 2 9110 Dpt Physique
NOTES		Théorie des Langages L3 Info SC 205		
PROFIL	les p C 205		Théorie des Langages L3 Info SC 205	
DÉCONNEXION				

Figure 4. EDT Professeur et Affichage Menu

Emploi du temps pour une salle :

On réserve la salle si le créneau est disponible en cliquant sur le bouton correspondant. Après confirmation, une liste déroulante est affichée pour choisir le cours que le professeur souhaite ajouter. Nous souhaiterions pouvoir cliquer sur chacun des cours de la salle pour afficher les informations supplémentaires de ces derniers.

Réservation de salle :

Elle sera possible en remplissant un formulaire pour savoir ce que désire le professeur. Une liste de propositions de cours lui est faite et il choisira celle qui lui convient le plus en cliquant dessus.

Gestion des notes :

Le professeur peut accéder à la liste des UE qu'il enseigne grâce à un menu déroulant. La liste des élèves inscrits à l'UE est affichée par défaut, l'utilisateur peut ajouter ou modifier la note de chaque élève en cliquant dans la case correspondante.

III. Création de l'application

1. Technologie Android

La majorité des applications Android a la même architecture, ce sont différentes fenêtres dans lesquelles l'utilisateur peut se déplacer. Ces fenêtres sont appelées des activités, et l'application ne peut en afficher qu'une à la fois. Par exemple, une application A qui diffuse de la musique et une autre B qui navigue sur internet, l'activité affichée est la B contrairement à la A, pourtant exécutées toutes les deux en même temps.

Une activité nous permet de manipuler des objets graphiques qui apparaîtront à l'écran, ces objets sont stockés dans des fichiers XML et ils formeront l'interface graphique. Selon les actions de l'utilisateur, ils pourront réagir différemment grâce au code Java qui les modifie.

Les activités d'une application ont une structure de données de type « LIFO ». L'activité affichée est celle au sommet de la pile. Cela permet à l'utilisateur de pouvoir dépiler les activités sur lesquelles il a déjà accédé grâce au bouton « Retour » du téléphone.

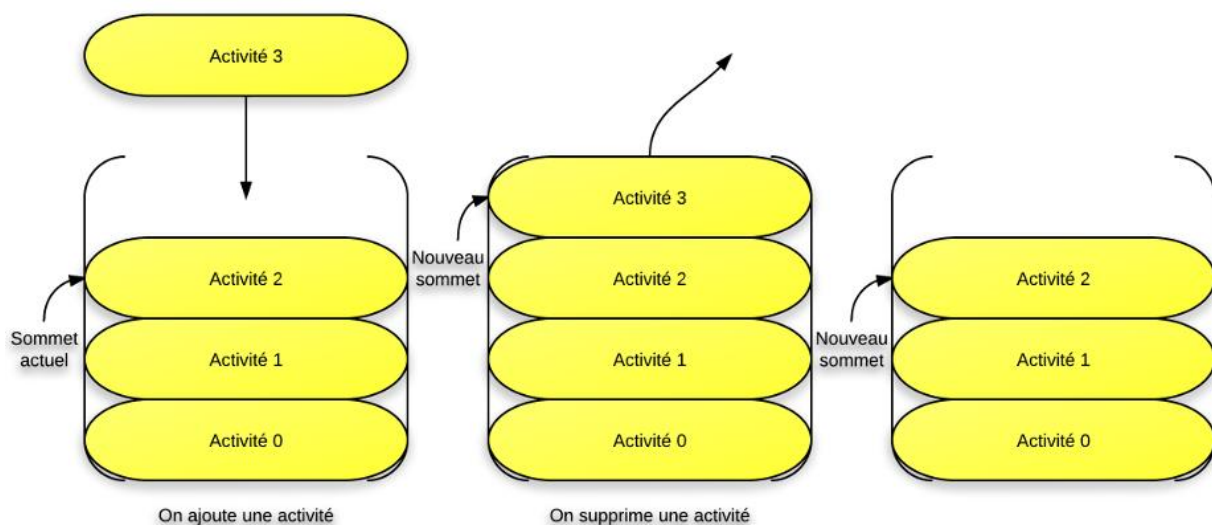


Figure 5. Organisation des activités

Nous utiliserons l'API 15 : Android 4.0.3 (IceCreamSandwich) car elle permet d'exécuter les applications sur près de 90 % des téléphones sous Android.

L'IDE que nous utiliserons est Android Studio. Il nous permet d'utiliser le gestionnaire de versions Git de manière simplifiée et très fonctionnelle par rapport à nos besoins (comparaison de fichiers modifiés d'une version à une autre, création de branches spécifiques à une fonctionnalité, etc.). Une interface graphique est directement implémentée pour l'utiliser dans le logiciel. Nous pouvons aussi tester l'implémentation de l'application en l'installant sur nos téléphones ou via un émulateur intégré. Enfin, il propose de manipuler directement les objets XML grâce à une interface graphique simulant un écran de téléphone, ce qui permet une grande facilité de prise en main et une mise en place des éléments graphiques plus rapide. La version du JDK (Java Development Kit) demandée par le logiciel est le jdk1.7, que nous utiliserons pour implémenter l'application.

Mr Kante nous a mis en place un espace de travail dans la forge de l'Université Blaise Pascal. Nous avons pu, au fur et à mesure que le projet avançait, poster des petits rapports dans le Wiki pour qu'il puisse non seulement suivre notre avancement, mais aussi avoir une sauvegarde de

notre évolution. Cela nous facilite la restitution de notre travail pour la rédaction du rapport. Un dépôt a été mis à notre disposition, dans lequel nous pouvons déposer le travail effectué par le groupe, afin de récupérer les mises à jour de chacun des membres. Nous avons utilisé le logiciel Git, qui permet de sauvegarder le code à travers des « branches » que les utilisateurs peuvent créer, sans altérer la version originale ou celles des autres étudiants.

2. Développement de l'application

Design général

Afin de respecter l'aspect défini dans le prototype, nous avons débuté l'implémentation de l'application en éditant les fichiers ressources du projet. Ces derniers sont des fichiers XML dont le contenu est commun à tout le projet. Ainsi, on a accès depuis n'importe quel fichier à ce contenu. Ces fichiers sont répartis dans divers dossiers appartenant à un dossier principal nommé « res » (pour ressources). Les dossiers principalement utilisés sont « values », « anim », « drawable » (respectivement pour valeurs, animations et les tracés).

Les ressources dites « values »

Dans un premier temps, nous avons édité le fichier colors.xml qui contiendra chacune des couleurs précisément utilisées dans la maquette. Ce fichier est divisé en deux parties, une pour les couleurs opaques, et une pour les couleurs transparentes.

Nous avons ensuite modifié le fichier styles.xml qui est composé du thème appliqué à l'application dans sa globalité : nous appliquons un thème d'Android légèrement changé. Ce fichier comprend également le style que nous définissons pour chacun des éléments utilisés par les activités : les boutons, les champs de texte, les boîtes de dialogues, et les tableaux.

Tous les textes bruts sont inscrits dans le fichier strings.xml, afin qu'un même texte soit disponible dans plusieurs fichiers sans le réécrire. Cette méthode est recommandée par Android pour éviter les problèmes tels qu'un mauvais formatage ou encore l'internationalisation de l'application.

Les ressources dites « anim »

Il contient plusieurs fichiers XML dont chacun définit une animation particulière. Ils seront utilisés pour appliquer un effet à un événement, tel un clic sur un bouton ou une apparition d'un élément. Nous avons utilisé des effets de disparition et d'apparition.

Les ressources dites « drawable »

Il y a deux types de ressources drawable, le premier regroupant les formes non-prédéfinies dans Android (par exemple, pour nos boutons, nous avons dessiné des formes basées sur un rectangle aux coins arrondis), et le second, les images existantes dans l'environnement Android. Celles-ci sont réparties dans quatre sous-dossiers pour quatre tailles différentes en fonction de l'écran de l'appareil utilisé. Nous utilisons ces images pour rendre plus intuitif l'utilisation de notre application sur certaines interactions telles que l'ajout, la suppression ou la modification d'un élément.

Le Manifest

Le fichier manifest.xml est un fichier à part puisqu'il contient l'ensemble de la structure de l'application. C'est dans ce fichier que l'on précise que l'application aura une icône, un thème

défini et qu'elle contiendra nos activités. Celles-ci sont d'ailleurs éditables dans le XML, on leur fixe un label à chacune, ainsi qu'un affichage en position paysage. C'est grâce au Manifest que l'on pourra appeler une activité depuis une autre.

Les activités

Pour chacune des activités créées, il a été nécessaire de définir un nouveau fichier XML que l'on appelle un layout et qui représente la structure de l'activité liée. C'est dans le layout que l'on définit l'affichage par défaut d'une activité quand elle est appelée. On mettra alors en place dans le layout les éléments nécessaires à l'activité, et ceux-ci ont de multiples propriétés qui sont éditables. On peut par exemple appliquer un style que l'on a prédéfini à un élément, attribuer un texte, une couleur, dire s'il est visible ou non, et surtout, définir sa position dans l'activité. Le layout de chaque activité est appelé dans la classe java de l'activité liée pour définir le design de celle-ci. Chaque activité est composée d'une ActionBarActivity qui correspond à la barre d'action située en haut de l'écran. Celle-ci nous permet d'afficher le label correspondant à l'activité affichée, d'ajouter des actions spécifiques à l'activité, dont une qui est commune à toutes : un bouton pour afficher le menu.

Le menu

L'ajout du menu a permis la suppression de tous les boutons visibles sur les activités, tel que le bouton de déconnexion ou de retour à l'accueil. En effet, il est désormais possible d'accéder à toutes les activités importantes grâce à ce menu. Étant caché par défaut, le menu nous a permis un gain de place assez important sur l'ensemble des activités, et notamment sur la page d'accueil, dont la suite de boutons positionnés en haut a été supprimée. De plus, nous avons rajouté un bouton dans la barre d'action pour que le menu soit activable plus intuitivement.

La page de connexion

Nous avons choisi d'implémenter une page de connexion sobre. Ce design a été pensé de manière à pouvoir rajouter facilement des fonctionnalités telles qu'un lien vers le site web de l'ENT, l'affichage de la date du jour, un affichage des notes de mises à jour ou d'actualités en lien avec l'Université, etc. Le seul élément visible sur cette page à l'ouverture de l'application est un bouton qui a pour action d'afficher le panel d'identification dans lequel l'utilisateur rentrera ses identifiants afin de se connecter à sa session et qu'il puisse accéder au contenu de l'application. Après la vérification de la concordance des identifiants saisis avec ceux inscrits dans la base de données, l'utilisateur est redirigé vers la page d'accueil qui affiche son planning personnel pour la semaine en cours, afin qu'il ait rapidement accès aux informations principales.

Les emplois du temps de la page d'accueil, planning prof et planning salle

D'une manière générale, les cellules du tableau correspondant aux créneaux horaire contenant un texte trop long, affichent alors le texte tronqué, indiqué par les points de suspension à la fin du texte. Nous permettons à l'utilisateur de visionner ce texte en entier avec un clic prolongé sur la cellule visée. Le texte non-tronqué apparaîtra dans un message temporaire.

La page de profil

Nous avons implémenté cette activité différemment par rapport à la maquette. En effet, sur le prototype, le tableau recensant les informations liées à l'utilisateur ne prenait que la moitié de l'écran. C'est pourquoi nous avons décidé d'étendre ce tableau à tout l'écran dans l'application. Nous avons gardé la possibilité de valider la modification par un bouton à droite du tableau et avons rajouté un bouton pour l'annuler si besoin. Désormais, l'utilisateur n'a qu'à cliquer directement sur les champs qu'il souhaite modifier, puisque ce sont des champs de texte éditables.

La page de réservation de salle

Afin d'optimiser l'affichage, le design de cette activité a été révisé pour que l'utilisateur ait à l'écran soit le formulaire de recherche, soit l'affichage des résultats de la requête. Cependant, l'affichage des résultats se fait dans la même activité. Pour réaliser cela, nous avons défini l'affichage des résultats comme invisible, pour que quand l'utilisateur clique sur le bouton de recherche, le formulaire devienne invisible et laisse alors apparaître l'affichage des résultats. Nous également avons repensé le moyen de sélection des critères de recherche pour que cela soit plus intuitif. En effet, il est désormais possible de glisser son doigt sur l'écran pour faire défiler les nombres jusqu'à trouver celui voulu. Enfin, nous avons pensé à rajouter un bouton de réinitialisation du formulaire pour qu'effectuer une nouvelle recherche soit plus facile.

La page de gestion des notes

Nous avons respecté le design du prototype pour cette activité, si ce n'est que la liste des notes des élèves n'est affichée qu'après avoir sélectionné une unité d'enseignement. Le bouton de sélection prend toute la largeur de l'écran puisqu'il sert également de titre informatif pour connaître l'unité d'enseignement sélectionnée.

La page de gestion des unités d'enseignement

Cette activité a subi d'importantes modifications puisqu'on a supprimé le champ de texte éditable pour le remplacer par un champ de recherche situé dans la barre d'action. De plus, les boutons d'ajout d'unité d'enseignement dans la liste de préférence ont été supprimés, il est actuellement possible d'en ajouter en cliquant directement sur une ligne parmi les résultats affichés. Ceci a été implanté de sorte à mieux correspondre à l'action de suppression d'unité d'enseignement qui fonctionnait déjà ainsi dans le tableau de la liste de préférence, par un simple clic sur une ligne.

3. Les améliorations envisageables

L'utilisation de l'application demande le transfert de données via Internet, celles-ci peuvent être confidentielles concernant un professeur. Il aurait été intéressant de mettre en place un système de sécurisation des données pour éviter qu'elles ne soient récupérables par quelqu'un de malintentionné.

Nous aurions voulu aussi pouvoir développer le design de l'application pour qu'elle soit plus ergonomique et surtout plus adaptable à tous les écrans, surtout en ce qui concerne les tableaux des emplois du temps, auxquels on fixe une taille non modifiable.

Un tutoriel aurait pu proposer une prise en main rapide de l'application pour les utilisateurs.

Nous aurions aimé gérer les utilisateurs de type « Responsable de promotion » ou « administrateur », pour que tous ceux possédant un compte puissent accéder à l'intégralité des services depuis l'application.

Nous aurions voulu également gérer le zoom sur l'écran en utilisant ses doigts et que la taille du texte s'adapte en fonction. On aurait alors pu remplacer la fonctionnalité du message tronqué affiché intégralement en appuyant longuement sur une case de l'emploi du temps.

Enfin, pour que l'application soit le plus fidèle possible au site web, nous pourrions intégrer un export des notes en format PDF téléchargeable sur le téléphone portable ou envoyé par e-mail.

Conclusion :

Le développement de l'application mobile nous a permis de réaliser que les tâches communes liées à l'analyse du projet sont primordiales pour une bonne cohésion du travail avec chacun des groupes. Utilisant les services implémentés par le groupe web, nous étions dépendants de leur travail. Par la suite, le travail en binôme devait être bien réparti pour éviter d'effectuer des tâches redondantes. De plus, il nous a fallu prendre en compte les domaines dans lesquels nous avions le plus de facilités afin d'optimiser le travail réalisé.

En remettant en question notre implémentation de l'application, on peut par exemple évoquer le fait que nous aurions dû utiliser des classes Java pour chacun des éléments graphiques qui sont redondants dans plusieurs activités, afin de rendre notre code plus modulaire et de ne pas manipuler la disposition des éléments uniquement dans les fichiers XML. La mise en commun des tâches de chaque groupe fut l'étape la plus difficile à cause du contexte de fin de semestre. Nous aurions dû écourter les tâches de chacun pour nous laisser le temps d'en faire la mise en commun.

Enfin, ce projet nous a permis d'imaginer l'organisation du travail en entreprise et de se familiariser avec les réunions hebdomadaires, afin de suivre régulièrement l'évolution de chacun. C'est une bonne simulation pour le stage en entreprise que nous devons effectuer pour la fin de notre semestre.

ANNEXES

La liste exhaustive des droits d'un administrateur :

- Ajouter et supprimer des professeurs.
- Attribuer la responsabilité d'une filière à un professeur.
- Ajouter ou supprimer un administrateur.
- Réinitialiser des mots de passe des utilisateurs.
- Modifier le profil d'un utilisateur.
- Ajouter et supprimer une unité d'enseignement.
- Ajouter, modifier ou supprimer des facultés.
- Ajouter, modifier ou supprimer des départements.
- Ajouter, modifier ou supprimer des promotions.
- Ajouter, modifier ou supprimer des classes.
- Ajouter, modifier ou supprimer des compétences.
- Ajouter, modifier ou supprimer des groupes.
- Ajouter, modifier ou supprimer des modules.
- Ajouter, modifier ou supprimer des salles.
- Ajouter, modifier ou supprimer des séances de CM, TP, TD ou examen.
- Ajouter, modifier ou supprimer un équipement.
- Ajouter, modifier ou supprimer des modalités d'examen.
- Ajouter, modifier ou supprimer des notes.
- Ajouter, modifier ou supprimer des étudiants.
- Affecter des unités d'enseignement à une promotion.
- Affecter des étudiants à une unité d'enseignement.
- Affecter des groupes à une classe.
- Affecter l'équipement nécessaire à une unité d'enseignement.
- Affecter de l'équipement à une salle.
- Affecter des compétences à un professeur.
- Affecter des compétences à une unité d'enseignement.
- Affecter des professeurs à une faculté.
- Réserver, annuler ou modifier une salle au nom d'un professeur.
- Générer l'emploi du temps de la filière de son choix.

Liste exhaustive des contraintes :

- Essayer d'avoir les salles proches, afin de gagner du temps sur les déplacements entre les cours
- Gérer les salles proches les unes des autres : pour certains groupes de grande taille, il peut être nécessaire de réserver deux salles côte à côte pour que l'effectif corresponde à la place disponible.
- Gérer le nombre de places : lors de l'attribution vérifier automatiquement que la taille de la salle corresponde au mieux à l'effectif du groupe.
- Il ne faut pas :
 - avoir deux cours dans la même salle en même temps.
 - réserver un cours dans deux salles différentes en même temps.
 - un professeur avec deux cours en même temps.
- Perte de mot de passe : lorsqu'un utilisateur perd son mot de passe, il pourra le signaler et demander l'envoi d'un nouveau mot de passe généré automatiquement.

- Indisponibilité des professeurs : si un professeur n'est pas disponible pour une plage horaire quelconque, il peut la préciser avant la génération de l'emploi du temps.
- Préférence des professeurs : au début du semestre le professeur peut énoncer des choix d'UE qu'il veut enseigner. Il pourra préciser s'il veut être le chargé de CM, de TD ou de TP. Il peut choisir plusieurs de ces types d'enseignement.
- Contrainte UE libre : lors de génération des emplois du temps, on doit veiller à placer les UE libres uniquement sur les plages horaires prévues à cet effet.
- Partition en groupe : lorsque l'effectif d'une classe est supérieur à un seuil, on peut penser à former des groupes pour les TD, TP ou même CM.
- Gestion des notes : Ce service permet à chaque professeur d'assigner une ou plusieurs notes pour l'UE qu'il enseigne. Ces dernières posséderont un coefficient qui permettra de calculer la note finale de l'unité d'enseignement à la fin du semestre. Il sera possible de les exporter en format PDF.
- Plusieurs professeurs peuvent enseigner la même unité d'enseignement.
- Pour transférer un cours ou réserver une salle, on doit s'assurer que le professeur, la classe et la salle impliqués soient libres pour le jour et l'horaire en question.
- Lors d'une réservation de salle : on doit vérifier qu'il n'existe pas un nouveau schéma d'affectation meilleur que celui déjà défini.
- L'équipement matériel, logiciel d'une salle et de leurs nombres doit être pris en compte.
- Pour une réservation automatique des salles de TP, les besoins matériels pour l'unité d'enseignement doivent être renseignés dans la base de données.
- Une unité d'enseignement peut être composée d'un ou plusieurs modules.
- Des étudiants peuvent suivre des unités d'enseignement libres ou en option, et qui peuvent être obligatoires pour d'autres promotions.
- Le chargé du CM de chaque unité d'enseignement devra définir le coefficient de son cours, du TD et du TP correspondants.

Les tables de la base de données :

- Entités :
 - Salle (idSalle, nom, effectif, type, localisation)
 - Compétence (idCompétence, nom)
 - Contrainte (idContrainte, idProf, jour, heureDebut, heureFin)
 - Faculte (idFaculté, nom)
 - Departement (idDepartement, idFaculté, nom)
 - Promotion (idPromotion, idDepartement, idProf, nom)
 - Classe (idClasse, idPromotion, effectif)
 - Groupe (idGroupe, effectif)
 - Etudiant (idEtudiant, idGroupe, nom, prenom, mail, adresse)
 - Modalite (idModalite, idSeance, coefficient, nom)
 - Seance (idSeance, idModule, type, nom)
 - Equipement (idEquipement, nom, logiciel)
 - Professeur (idProf, idUtilisateur)
 - Module (idModule, idUE, nom)
 - Utilisateur (idUtilisateur, mail, adresse, nom, prenom, motdepasse, siteWeb, numTel)
 - UE (idUE, nom)
- Associations :
 - Professeur_has_Competence (id, idProf, idCompétence)
 - Faculte_has_Professeur (id, idFaculte, idProf)

- Promotion_has_UE(id,*idPromotion*,*idUE*,option,libre)
- Preference(id,*idProf*,*idSeance*)
- Groupe_has_Classe(id,*idGroupe*,*idClasse*)
- Note(id,*idModalite*,*idEtudiant*,*idModule*,note)
- Salle_has_Equipement(id,*idSalle*,*idEquipement*,nombre)
- Affectation(id,*idProf*,*idSeance*,*idSalle*,*idGroupe*,annee,jour,heure,dureeMinute)
- Module_has_Equipement(id,*idModule*,*idEquipement*)
- Module_has_Compotence(id,*idModule*,*idCompotence*)
- Etudiant_has_UE(id,*idEtudiant*,*idUE*)

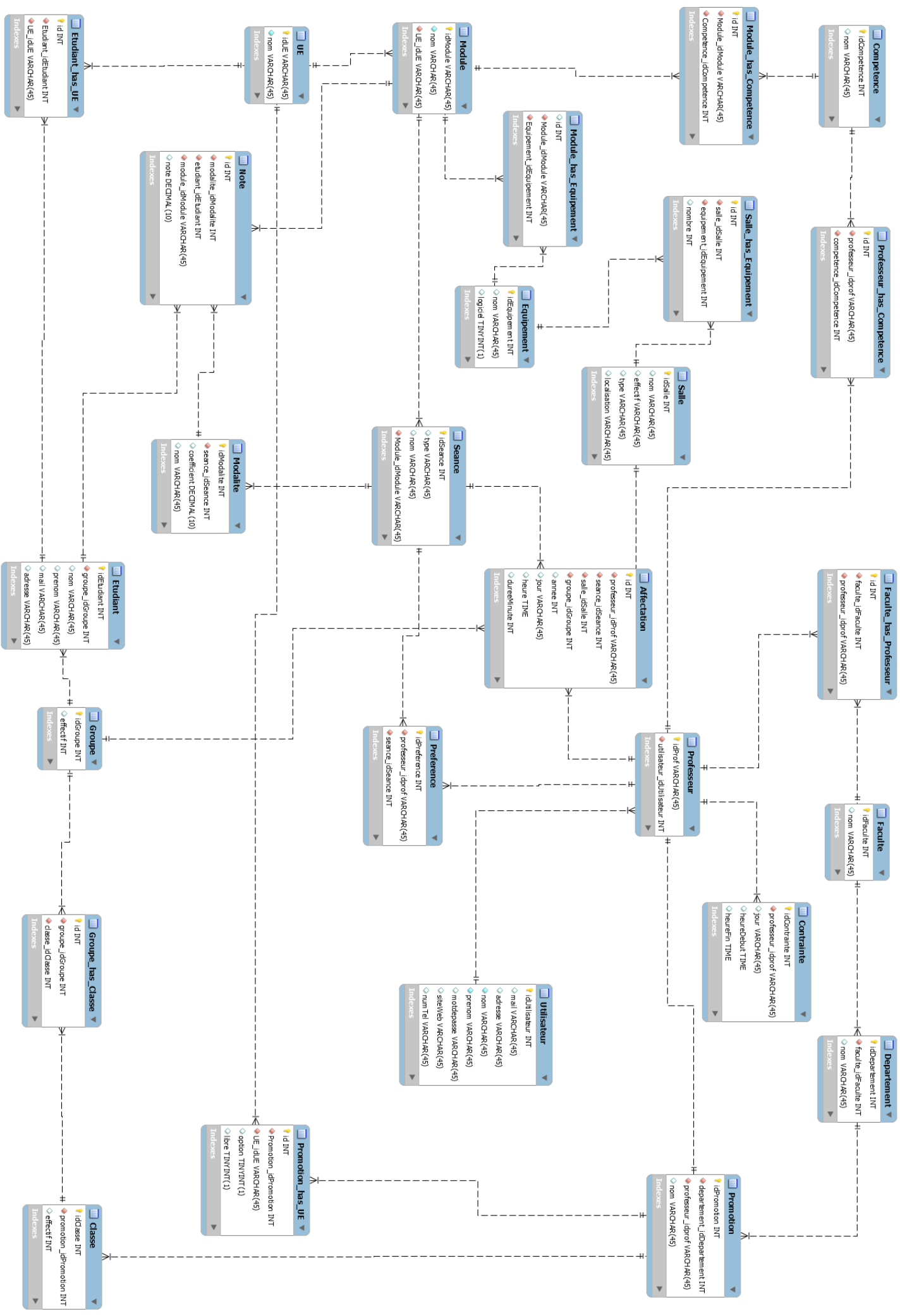


Diagramme de classes :

