

République du Bénin

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Africaine de Technologie et de Management

Filière: Génie Electrique

Mémoire

Présenté pour l'obtention de la Licence en informatique
Option : Système Informatique et Logiciel

Thème

Système de gestion de salles de cours :
cas de l'Université d'Abomey-Calavi

Réalisé et soutenu par :

Salaou-Deen Henri-J. A. PARAISO

Sous la supervision de :

Dr ATINDEHOU Mêton Mêton

Année universitaire 2020 - 2021

Table des matières

Dédicaces.....	3
Remerciements.....	4
Liste des sigles et abréviations.....	5
Figures et tableaux.....	6
Résumé.....	7
Abstract.....	7
Introduction.....	8
Problématique.....	9
Objectifs et valeurs de ClassMan.....	10
I – Étude bibliographique.....	11
A – Étude des systèmes de gestion automatisée d’emploi du temps.....	11
1 – FET.....	11
2 - UniTime.....	15
3 - TimeTabler.....	19
B – Etude des algorithmes d’élaboration des emplois du temps.....	21
1 – La méthode de descente.....	21
2 – La méthode de recuit simulé.....	22
3 – La méthode de tabou.....	23
4 – Les algorithmes génétiques.....	24
5 – Les algorithmes de colonies de fourmis.....	25
II – Etude du problème de gestion de salles de cours à l’UAC.....	26
A - L’organisation des formations à l’UAC.....	26
B - Gestion informatisée des salle de cours.....	29
V - Modélisation du système.....	33
B - Diagramme de classes.....	33
C – Diagramme de cas d’utilisation.....	33
IV – Conception de la solution.....	37
A - Présentation de l’achitecture du système.....	37
B – Les technologies utilisées lors de la conception.....	37
V – Présentation du noyau du système.....	40
A – Interface graphique.....	40
1 – Splash screen.....	40
2 – Inscription.....	41
3 – Connexion.....	42
Conclusion.....	43
References.....	44

Dédicaces

À ma mère et ma sœur, les deux plus grandes femmes de ma vie, elles m'ont toujours soutenu tout au long de mon parcours.

À mon père qui a su me mettre sur le chemin de la droiture et m'a appris à trouver ma voie.

À Denakpo Vincent pour tout le soutien qu'il m'a apporté durant toutes ces années.

À mon feu oncle Mansour Idelphonse Paraïso.

À mon frère Fawaz qui est toujours resté à mes côtés.

À Kemal Dara qui m'a fait découvrir la programmation.

À Salim et Sameerath pour les efforts qu'ils ont eu à faire pour me soutenir.

À tous mes frères et soeurs.

Aux familles Paraïso et Amoussa.

À mes amis Nelson, Crispus, Marina, Marie-Ange, Ursule pour tout leur soutien.

Aux apprenants et acteurs du centre des aveugles de Segbeya.

Merci !

Remerciements

À notre directeur de mémoire Docteur ATINDEHOU Mêton Mêton, malgré vos multiples occupations, vous avez accepté de nous guider jusqu'à l'aboutissement de ce travail. Vos précieux conseils nous ont beaucoup édifiés, nous vous remercions sincèrement.

Au corps professoral de l'Université Africaine de Technologie et de Management, pour les précieux enseignements reçus de vous, nous vous prions de bien vouloir accepter nos humbles compliments et reconnaissances.

À l'ensemble du personnel tant administratif que de service de l'Université Africaine de Technologie et de Management, nous vous remercions sincèrement pour votre disponibilité et compréhension.

À tous ceux qui de près et de loin m'ont apportés leur soutien lors réalisation de ce mémoire. Sincères remerciements.

Liste des sigles et abréviations

ClassMan : Class Manager

API : Application Programming Interface

REST : Representational State Transfer

UAC : Université D'Abomey Calavi

REESAO : Réseau pour l'Excellence de l'Enseignement Supérieure en Afrique de l'Ouest

UE : Unité d'enseignement

LMD : Licence Master Doctorat

ECUE : Unités Constitutifs d'une Unité d'Enseignement

JS : JavaScript

Figures et tableaux

Tableaux

Tableau 1: Tableau comparatif des trois différents systèmes.....	20
Table 2: Tableau de présentation de l'offre de formation suivant la norme du REESAO.....	27
Table 3: Tableau de présentation d'une Unité d'Enseignement.....	28
Table 4: Tableau de présentation d'une ECUE.....	29
Table 5: Tableau synthétique présentant une offre de formation.....	29

Figures

Figure 1: Principe général d'un algorithme génétique standard.....	25
Figure 2: Diagramme de classes.....	33
Figure 3: Diagramme de cas d'utilisation des administrateurs.....	35
Figure 4: Diagramme de cas d'utilistation des enseignants.....	36
Figure 5: Digramme de cas d'utilisation des étudiants.....	36
Figure 6: schéma de l'architechture du système.....	37
Figure 7: Logo de python.....	38
Figure 8: Logo de django.....	38
Figure 9: Logo de django rest framework.....	38
Figure 10: Logo de postgresql.....	38
Figure 11: Logo de kotlin.....	38
Figure 12: Logo de JS.....	39
Figure 13: Splashscreen.....	40
Figure 14: Interface d'inscription.....	41
Figure 15: Interface de connexion.....	42

Résumé

Nous étudions le problème de gestion de salles de cours dans le contexte particulier d'une université africaine du type de l'Université d'Abomey-Calavi. Nous proposons un outil, dont nous construisons un noyau viable, afin d'apporter des solutions. Nous dénommons cet outil, ClassMan. C'est un outil automatisé de gestion de salles de cours qui permet de calculer automatiquement les emplois du temps, de “matcher” aux cours les salles qui leur correspondent le mieux, d'établir une planification des emplois du temps en accord avec le calendrier académique fixé.

Abstract

We study the problem of classroom management in the particular context of an African university such as the University of Abomey-Calavi. We propose a tool, of which we build a viable core, in order to provide solutions. We call this tool ClassMan. It is an automated classroom management tool that automatically calculates schedules, matches courses with the rooms that best correspond to them, and establishes a schedule in accordance with the academic calendar.

Introduction

La planification horaire est un problème qu'on rencontre dans de nombreux secteurs d'activités. Dans les écoles, les cours doivent être programmés de manière à ce que le calendrier soit respecté. Dans les entreprises, les tâches doivent être réparties de manière à ce que les délais soient respectés. Cette planification se fait généralement à la main et de façon périodique ce qui est très complexe et pas efficace étant donné qu'il y a généralement beaucoup de contraintes à respecter. De plus à la moindre modification d'une donnée, le planning élaboré peut être corrompu. De façon basique, le problème de l'emploi du temps consiste à assigner des tâches à un groupe d'individus tout en allouant des ressources matérielles et en tenant compte d'un certain nombre de contraintes bien définies sur une période déterminée. Dans le but d'épargner cette tâche ardue qu'est l'établissement des emplois du temps, les chercheurs ont réfléchi à des algorithmes de gestion automatisée de ces derniers.

Dans l'objectif de régler ce problème de gestion d'emploi du temps qui entraîne à l'UAC un véritable problème d'allocation de salles de classes, nous proposerons dans notre étude un système de gestion automatisée (ClassMan) qui permettra de calculer les emplois du temps automatiquement et de manière à faire correspondre chaque cours à une salle appropriée au bon déroulement de ce dernier tout en respectant le calendrier établi.

Il s'agira dans un premier temps, de faire une étude approfondie du système à réaliser et à modéliser un noyau solide de notre système. Puis nous présenterons dans un second temps notre solution.

Problématique

L'allocation des salles de l'Université d'Abomey-Calavi (UAC) est un problème crucial qui se pose sous bien des aspects. Il s'agit principalement pour l'UAC de répartir au mieux ses salles entre ses entités afin que ses formations puissent se tenir dans les délais fixés par le calendrier académique et dans les meilleures conditions que cette université sait fournir (avec les équipements dont elle dispose). Le problème se pose également sous la forme de répartir les salles pour une bonne tenue des évaluations. Il y a aussi le fait que l'université peut utiliser des salles afin de générer des ressources financières, en les mettant par exemple en location pour des événements publics / privés du type réunions, conférences.

Dans le cas de l'allocation des salles pour la tenue des cours, la méthode utilisée à l'UAC reste manuelle et inefficace. Elle consiste chaque semaine à répartir les salles disponibles aux facultés qui en ont besoin. Il revient alors à chaque vice-doyen de calculer l'emploi du temps de sa faculté. Cette méthode laisse parfois des salles de cours potentiellement inutilisées dans certaines facultés alors qu'elles pourraient l'être dans d'autres. Il y a également le fait que certaines salles ne sont pas utilisées de la meilleure manière possible. C'est-à-dire, en mettant optimalement en correspondance les fonctionnalités des salles aux besoins des formations. Aussi, les facultés qui ont leurs propres salles de cours pourraient les mettre à disposition de la communauté dans les créneaux horaires dont elles n'ont pas utilisé.

Le problème d'allocation de salles est en fait un vieux problème qui a déjà beaucoup été étudié (dans des domaines comme la recherche opérationnelle et les sciences informatiques) et pour lequel il a été prouvé que quand la taille du problème est aussi grande que celle de l'UAC, il est très difficile de trouver une solution optimale en

essayant simplement à la main toutes les possibilités d'allocation. La méthode manuelle est intrinsèquement mauvaise. Il nous faut maintenant utiliser des méthodes automatisables et offrant des solutions satisfaisantes de répartition des salles disponibles de l'UAC pour la bonne tenue des formations de cette université.

L'automatisation de l'allocation des salles de cours de l'UAC donnera aussi une visibilité exacte sur les types de salles qu'il faut construire pour répondre aux besoins spécifiques des différentes formations dispensées dans cette université. Ceci permettra aux organes dirigeants de mettre en place une politique éclairée de construction de salles atténuant le plus efficacement possible le problème de pénurie de salles, en fonction des ressources financières qui pourront être allouées à cela dans les prochaines années.

Objectifs et valeurs de ClassMan

ClassMan est un système de gestion de salles de classe autonome et dynamique. Basé sur un certain nombre d'algorithmes de timetabling, ses principaux objectifs se présentent comme suit :

- Élaboration automatisée des emplois du temps
- Éviter les problèmes d'inadaptation des salles de classes aux cours
- Établir une planification horaire efficace, concordant avec le calendrier établi
- Faire un tracking efficace de l'effectivité des formations
- Prévenir les problèmes de coïncidences de cours

I – Étude bibliographique

A – Étude des systèmes de gestion automatisée d’emploi du temps

Le problème de la gestion des emplois du temps date de quelques décennies. Plusieurs études et travaux ont déjà été menés sur le sujet. Nous avons donc actuellement sur le marché, un certain nombre de systèmes de gestion automatisés des emplois du temps. Nous avons choisi de présenter les systèmes FET, Unitime et TimeTabler.

1 – *FET*

FET est un logiciel gratuit qui permet d’établir de façon automatisée, les emplois de temps pour les écoles primaires, secondaires, et même l’université. Il est basé sur un algorithme de timetabling rapide et fluide. Il est déposé sous la licence open source GNU Affero General Public License 3. Sa dernière version est la version FET-6.0.3 du 27 mai 2021.

En matière de performance, FET n’est pas à plaindre. Il est capable de générer un emploi de temps compliqué qui prends en compte beaucoup de contraintes en 5 à 20 minutes selon les performances de votre ordinateur. Cependant pour des opérations moins compliquées, le temps d’exécution peut varier entre 1 minute et 5 minutes. Pour des cas de complexité super élevée, le temps d’exécution peut varier entre 30 minutes et 1 h ou plus.

Fonctionnalités et avantages :

- FET est multilingues et est disponible dans une vingtaine de langues.
- Son algorithme est automatisé, flexible et supporte l'allocation semi-automatique et manuelle.
- Est multi-plateformes et disponible sur Windows, Mac, Linux et sur tous les systèmes supportant Qt.
- Fonctionne de façon flexible avec le XML comme format d'entrée de données, dispose aussi d'un éditeur XML et aussi d'un système d'entrée de données à la main.
- Importe et exporte les données au format CSV.
- Retourne les emplois de temps générés en HTML, XML et CSV.
- Dispose d'une organisation des données flexible et bien structurée.
- Dispose d'une documentation bien élaborée.
- Personnalisable à volonté.
- Dispose d'une très grande communauté.

Limites de l'algorithme de FET :

- Nombre d'heures de travail maximum par semaine : 1000
- Nombre d'heures maximum journalier : 1440
- Nombre de professeurs virtuels : illimité
- Nombre maximum de sous-groupes d'étudiants : 30000
- Nombre de sujets illimités.
- Nombre illimité de catégories d'activités.
- Nombre maximum d'activités : 500000
- Nombre maximum de salles : 30000
- Nombre maximum de bâtiments : 30000

- Possibilité d'ajouter plusieurs ensembles d'enseignants et élèves pour chaque activité. (il est possible aussi de n'avoir ni professeurs ni élèves pour une activité)
- Nombre de contraintes de temps et d'espaces virtuels illimités.
- Une large possibilité de contraintes de temps :
 - Période de pause
 - Contraintes des enseignants :
 - Période d'indisponibilité;
 - Nombre de jours maximum et minimums hebdomadaire;
 - Nombre maximum de lacunes par Jour/Semaine;
 - Nombre d'heures journalier/continu;
 - Durée maximale par jour;
 - Nombre d'heures minimal par jour.
 - Nombre maximum d'heures par jour/en continu avec une balise d'activité;
 - Nombre d'heures minimum par jour avec une étiquette d'activité;
 - Respecter travailler dans un intervalle horaire un nombre maximum de jours par semaine;
 - Heures de repos minimum.
- Pour les étudiants :
 - Périodes d'indisponibilité;
 - Nombre de jours maximum par semaine;
 - Commencer plus tôt que prévu (précisez les débuts maximum autorisés à la deuxième heure);
 - Écarts max par jour/semaine;
 - Heures maximales par jour/en continu;
 - Durée maximale par jour;

- Nombre d'heures minimum par jour;
- Nombre maximum d'heures par jour/en continu avec une étiquette d'activité;
- Nombre d'heures minimum par jour avec une étiquette d'activité;
- Écarts minimaux entre une paire ordonnée de balises d'activités;
- Respecter travailler dans un intervalle horaire un nombre maximum de jours par semaine;
- Heures de repos minimum;
- Pour une activité ou un ensemble d'activités/sous-activités :
 - Une seule heure de départ préférée;
 - Un ensemble d'heures de départ préférées;
 - Un ensemble de plages horaires préférées;
 - Jours min/max entre les activités;
 - Fin(s) de la journée des étudiants;
 - Même heure/jour/heure de début;
 - Occuper les plages horaires max/min de la sélection (deux contraintes complexes et flexibles, utiles dans de nombreuses situations);
 - Consécutives, ordonnées (et ordonnées si le même jour), groupées (pour 2 ou 3 sous-activités);
 - Pas de chevauchement (également pour les balises d'activités);
 - Max/min simultanées dans les plages horaires sélectionnées;
 - Écarts minimaux entre un ensemble de (sous)-activités.
- Une palette large et flexible de contraintes d'espace :
 - Périodes de non-disponibilité de salle;
 - Pour les enseignants :

- Salle d'accueil;
 - Changements maximums de pièce/bâtiment par jour/semaine;
 - Écarts minimums entre les changements de pièce/bâtiment.
-
- Pour les étudiants:
 - Salle d'accueil;
 - Nombre maximum de changements de pièce/bâtiment par jour/semaine;
 - Écarts minimaux entre les changements de pièce/bâtiment.
-
- Salle(s) préférée(s) :
 - Pour un sujet;
 - Pour une balise d'activité;
 - Pour un sujet et une balise d'activité;
 - Individuellement pour une (sous)activité.
-
- Pour un ensemble d'activités :
 - Avoir la même chambre si elles sont consécutives;
 - Occuper un maximum de pièces différentes;

2 - UniTime

UniTime est un système de planification pédagogique complet qui prend en charge l'élaboration d'horaires de cours et d'examens, la gestion des modifications apportées à ces horaires, le partage de salles avec d'autres événements et la planification des étudiants dans des cours individuels. Il s'agit d'un système distribué qui permet à plusieurs gestionnaires d'horaires universitaires et départementaux de coordonner leurs

efforts pour créer et modifier un horaire qui répond à leurs divers besoins organisationnels tout en permettant de minimiser les conflits de cours des étudiants. Il peut être utilisé seul pour créer et tenir à jour le calendrier des cours et/ou examens d'une école, ou interfacer avec un système d'information étudiant existant.

Le système a été développé à l'origine dans le cadre d'un effort de collaboration entre les professeurs, les étudiants et le personnel des universités d'Amérique du Nord et d'Europe. Le logiciel est distribué gratuitement sous une licence open source dans l'espoir que d'autres collèges et universités puissent en bénéficier. Le projet UniTime est devenu un projet parrainé de la Fondation Apereo en mars 2015.

L'objectif principal de la planification horaire est de placer chaque cours à une heure (ou à un ensemble d'heures) qui n'entre pas en conflit avec l'heure ou les heures assignées à tout autre cours requis par les étudiants qui y assistent. C'est relativement facile s'il n'y a que quelques cours. Cela devient considérablement plus difficile lorsque la quantité de ressources et de contraintes augmentent. La disponibilité des professeurs, des salles et une variété d'autres contraintes compliquent encore le problème.

UniTime intègre les dernières recherches sur les algorithmes de solutions conçus pour résoudre efficacement des problèmes d'emploi de temps complexes. Il fournit également un modèle de structure de cours très complet qui permet de définir facilement les relations entre les composants de cours avec plusieurs types d'enseignements (par exemple, cours magistral, discussion, laboratoire) et différentes exigences en matière de temps de réunion. De plus, UniTime propose des approches centralisées, distribuées et hybrides pour construire votre horaire de cours en fonction de vos besoins. Le module d'horaires de cours peut également être utilisé pour tester plusieurs scénarios, tels que l'effet d'un nombre réduit de salles de classe ou des changements dans les exigences des cours.

Gestion de cours

Même le meilleur calendrier peut nécessiter des modifications en raison de l'évolution des besoins ou de la disponibilité des ressources. Une augmentation de la demande pour un cours, le départ d'un membre du corps professoral ou la perte d'une salle de classe nécessitera un changement de salle ou d'affectation de temps. UniTime permet aux utilisateurs de rechercher facilement des alternatives qui ont un impact minimal sur l'emploi de temps global, d'apporter des modifications à l'emploi de temps des cours et de communiquer ces modifications aux étudiants concernés et aux autres systèmes.

Les changements de classe après la publication de l'emploi de temps utilisent un solveur interactif qui suggère un ensemble de taille limitée d'affectations de temps et/ou de salle alternatives pour les classes qui permettront à un changement proposé d'avoir lieu. Initialement, le solveur recherche les changements d'affectation possibles qui correspondent le mieux à toutes les exigences et préférences définies sur les classes sans affecter l'affectation de plus de deux classes supplémentaires. L'utilisateur peut alors sélectionner l'ensemble de modifications qui répond le mieux à ses besoins parmi les options présentées. Chaque option fournit des informations sur les conflits d'élèves supplémentaires créés et sur toute autre préférence susceptible d'être violée. Si une solution n'affectant que deux autres classes n'est pas possible, la recherche peut être ajustée pour permettre des modifications supplémentaires. Les modifications peuvent être limitées pour autoriser les échanges de salle uniquement, les échanges de temps uniquement ou les modifications à la fois de la salle et de l'heure.

1 - Points forts du système:

- Disponible publiquement
 - Open source;

- Application serveur-client avec interface Web;
- Indépendant de la plate-forme (implémenté à l'aide de Java J2EE et d'une base de données SQL);
- Couvre tous les besoins d'horaires universitaires
 - Horaires des cours, découpage des élèves, horaire des examens et gestion des événements;
 - Permet la décomposition en plusieurs problèmes d'horaires si désiré;
 - Fournit une gestion et une coordination réparties entre plusieurs unités organisationnelles;
 - S'adapte au comportement concurrentiel;
- Applications
 - Système d'emploi de temps appliqué avec succès dans la pratique à l'Université de Purdue;
 - Gros problème à l'échelle de l'université:
 - 9 000 cours, 570 salles
 - 2 400 examens, 39 000 étudiants
 - 190 000 demandes de cours
 - Permet des modifications interactives;
 - Peut-être utilisé dans des modes allant de la saisie manuelle des données à la programmation entièrement automatisée;
- Extensible et personnalisable
 - Applicable à une variété de problèmes d'horaires universitaires.

2 - Solveur

- Bibliothèque de résolution de contraintes

- Framework basé sur la recherche locale utilisant des primitives de programmation par contraintes;
- Disponible publiquement sous licence open source (GNU LGPL);
- Gagnant de deux prix de l'International Timetabling Competition 2007 (finaliste dans les trois pistes);
 - Applicable à une variété de problèmes de satisfaction de contraintes et d'optimisation;
- Identifie les incohérences et les problèmes potentiels dans les données d'entrée.

3 - Sectionnement des étudiants

- Sectionnement par lots
 - Une fois un emploi du temps créé, les étudiants pré-inscrits sont immédiatement inscrits dans les classes les plus adaptées.
- Sectionnement en ligne.
 - Les inscriptions d'étudiants supplémentaires et les demandes de changement sont faites en ligne avec des horaires disponibles immédiatement.
- Les réservations dynamiques protègent la disponibilité des cours, permettant un peu moins de choix, mais entraînant considérablement moins de demandes de cours non satisfaites.

3 - TimeTabler

TimeTabler est un programme informatique rapide et convivial, basé sur de nombreuses années d'expérience en matière d'horaires, soigneusement conçu pour planifier les emplois de temps rapidement et avec précision. De plus, son utilisation nécessite aucune connaissance particulière en informatique. Il est utilisé par plusieurs centaines d'écoles et de collèges, dans plus de 80 pays.

TimeTabler s'adaptera facilement à la plus grande école. Et pourtant, il est également convivial pour les petites écoles. Multiplateforme, il est disponible sur Windows, Mac os et Linux.

TimeTabler permet de gérer efficacement :

- l'horaire des enseignants à temps partiel;
- les modèles d'options, les ensembles mathématiques;
- les pauses déjeuner échelonnées, les jours variables;
- les écoles à sites partagés, les journées de consortium;
- l'enseignement partagé dans la « sixième forme »;
- les blocs « établis de manière cohérente ».

En utilisant la vitesse élevée de l'ordinateur, combinée à des algorithmes de planification efficaces et éprouvés, TimeTabler anticipe et montre à chaque étape une liste recommandée de priorités à prendre en compte. C'est-à-dire qu'il recommande quelle leçon vous devez placer sur l'emploi de temps, et où vous devez la placer !

À chaque étape, vous pouvez suivre les conseils du programme ou prendre votre propre décision.

De cette façon, vous obtenez le meilleur résultat de la vitesse de l'ordinateur combiné avec votre jugement, votre expérience et votre connaissance de vos collègues.

Mais en plus de cette vérification mécanique et de cette hiérarchisation, TimeTabler gère la programmation réelle de l'emploi du temps de manière interactive ou automatique.

Tableau 1: Tableau comparatif des trois différents systèmes

	Fet	Unitime	Timetabler
Open Source	✓	✓	x
Gratuit	✓	✓	✓ (Partiellement)

Licence	GNU Affero General Public License 3	Apache	-
Reutilisable	✓	✓	✗
Multilingue	✓	✓	✓
Multi-plateformes	✓	✓	✓
Format d'import de données	XML, CSV, Entrée brute à la main	XML	Entrée brute à la main
Format de sortie de données	HTML, CSV, XML	XML	Affichage graphique
Personnalisation	✓	✓	✗
Gestion des contraintes avancées	✓	✓	✗
Gestion automatisée des contraintes	✓	✓	✓
Temps moyen d'exécution	Rapide	Rapide	Peu rapide

B – Étude des algorithmes d'élaboration des emplois du temps

1 – La méthode de descente

Dans la famille de méthodes de recherche locale, la méthode la plus populaire et la plus élémentaire est la méthode de descente. Basée sur les itérations, elle choisit à chaque itération un point dans le voisinage de la solution courante. Si le point choisi améliore la solution courante, il devient la nouvelle solution courante. Dans le cas contraire, l'algorithme choisit un autre point. Cette opération est répétée tant que la valeur de la fonction objective diminue et s'arrête lorsqu'on atteint un minimum local f .

2 – La méthode de recuit simulé

Introduite dans les années 80 par trois chercheurs d'IBM cette méthode est inspirée du processus de recuit utilisé en métallurgie pour améliorer la qualité d'un solide. En effet le recuit simulé peut être vu comme une version étendue de la méthode de descente. En réalité, la méthode du recuit simulé est un algorithme d'optimisation consistant en la recherche du minimum d'une fonction donnée dans divers domaines. En tant que technique générale d'optimisation, la méthode du recuit simulé apporte une solution appropriée aux modèles hautement non-linéaires, aux données bruitées ou soumises à de fortes contraintes. L'un de ses principaux avantages est dû à son pouvoir de polyvalence provenant de la propriété intrinsèque dans laquelle se repose l'algorithme Metropolis-Hastings pour la recherche de la solution optimale. Cependant il existe clairement un compromis entre la qualité de la solution trouvée et le temps de calcul qui est court, l'algorithme peut se trouver alors bloqué dans un état méta-stable relativement éloigné de l'état de moindre énergie. D'autre part, la méthode du recuit simulé est aussi confrontée aux perturbations aléatoires et aux plusieurs conditions requises afin de garantir la convergence de la procédure.

3 – La méthode de tabou

La méthode de tabou a été introduite par Fred Glover. Une partie de la communauté scientifique, estime que cette méthode est plus acceptable sur le plan scientifique que le recuit simulé car la partie aléatoire de la méthode n'est plus. En réalité le principe de cette méthode analyse un ensemble de solutions $N(S)$ et retient la meilleure solution S' même dans le cas où S est meilleure que S' . Avec cette méthode, la recherche ne s'arrête

pas même lorsqu'on atteint un optimum, ce qui peut engendrer des cycles. Pour donc éviter ces mouvements cycliques, on enregistre les dernières configurations explorées dans une mémoire temporaire appelée liste de tabous tout en interdisant les mouvements conduisant à ces dernières.

Tout ceci permet à la méthode tabou de prendre moins de place mémoire et d'éliminer plus de solutions que celles visitées effectivement puisque ces listes sont générées en FIFO (first in first out). Les avantages de la recherche tabou sont dus à sa grande efficacité et à son fonctionnement plus simple à comprendre. Par contre, la recherche tabou présente des paramètres plus intuitifs et demande une quantité importante de ressources si la liste des tabous est imposante puisque cette dernière (méthode tabou) ne s'occupe d'aucune démonstration de la convergence. Mais, le statut tabou, peut créer un autre danger qui consiste à bloquer le processus de recherche en rendant tous les mouvements tabous. Cependant, le critère d'aspiration permet certainement de remédier à cet inconvénient en acceptant un mouvement même si il est tabou.

4 – Les algorithmes génétiques

De la famille des algorithmes dit évolutionnistes, les algorithmes génétiques ont pour but de trouver une solution approchée aux problèmes d'optimisation complexe. Ils sont souvent utilisés lorsqu'il n'existe pas de méthode exacte pour résoudre le problème en un temps record. Basé sur une notion de sélection naturelle. Le principe de ces algorithmes est de partir d'une population de p individus, puis de sélectionner ceux capables de se reproduire. On croise ensuite ces derniers, de manière à obtenir une population fille dont on peut faire muter de façon aléatoire certains gènes. La performance des enfants est évaluée grâce à la fonction fitness. On sélectionne ensuite dans la population résultante les individus autorisés à survivre de manière à ce qu'on puisse repartir d'une nouvelle population de p individus.

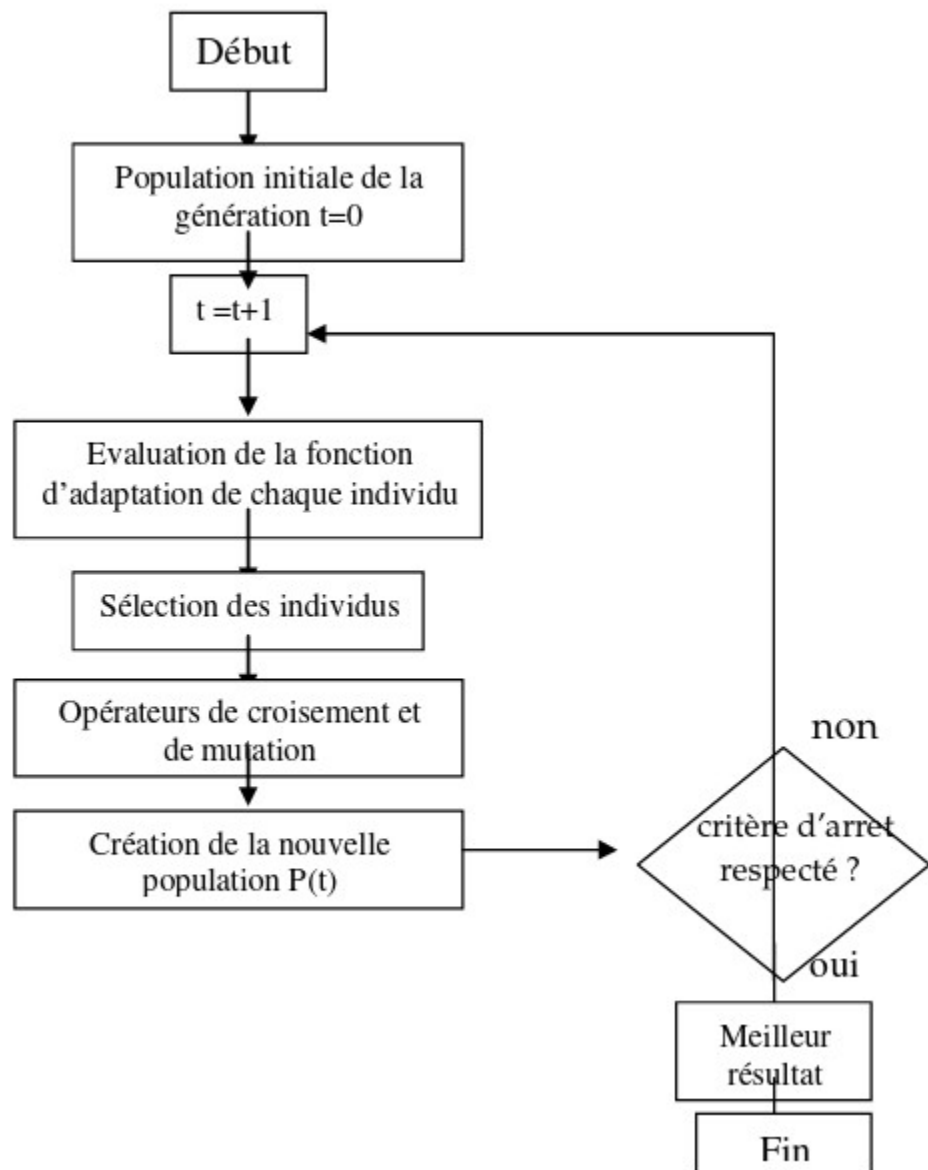


Figure 1: Principe général d'un algorithme génétique standard

5 – Les algorithmes de colonies de fourmis

Les algorithmes de colonie de fourmis proposés par Dorigo pour les problèmes d'optimisation complexe s'inspirent des comportements des fourmis. Leur principe est basé sur le comportement des fourmis qui font évoluer une population d'agents sur la base d'un modèle stochastique : lorsqu'elles vont en exploration hors de leur fourmilière

à la recherche de nourriture, ils finissent par élaborer des chemins qui s'avèrent fréquemment être les plus optimaux pour aller de la fourmilière à une source de nourriture intéressante. Chaque fourmi laisse en effet derrière elle une traînée de phéromone à l'attention de ses congénères. Les fourmis choisissant avec une plus grande probabilité les chemins contenant les plus fortes concentrations de phéromones.

Le premier algorithme conçu selon ce modèle était destiné à résoudre le problème du voyageur de commerce.

II - Étude du problème de gestion de salles de cours à l'UAC

A - L'organisation des formations à l'UAC

L'UAC dispose de plusieurs formations dispensées dans ses entités que sont les facultés, écoles et instituts. Pour faciliter l'organisation des formations, une entité peut être subdivisée en départements, qui peuvent être subdivisés en filières. Il revient alors aux filières d'organiser des formations spécifiques. Une filière a des années d'étude. Un groupe pédagogique correspond alors au groupe d'étudiants d'une année d'étude. Étant donné les effectifs d'étudiants de certains groupes pédagogiques et l'impossibilité actuelle de trouver des salles de cours suffisamment grandes pour contenir ces effectifs, ces groupes sont subdivisés en sous-groupes pédagogiques.

Les formations dispensées à l'UAC respectent la norme standard Licence Master Doctorat (LMD) définie par le Réseau l'Excellence de l'Enseignement Supérieur en Afrique de l'Ouest (REESAO). Une formation est donc constituée d'un ensemble d'Unités d'Enseignement. Une UE peut être une unité de connaissance fondamentale, une unité de spécialité ou de découverte, une unité de méthodologie, ou enfin une unité de culture générale. Une UE peut se décomposer en sous-unités, dénommées Eléments Constitutifs d'une Unité d'Enseignement (ECUE).

Pour mettre en place le système informatique de gestion des salles de l'UAC, nous aurons besoin de construire une base de données contenant les informations des différentes offres de formations de l'UAC.

Présentation de l'Offre de Formation suivant la norme du REESAO

Identification de la formation	
Etablissement	UAC
Domaine de formation	
Mention	
Spécialité	
Grade	
Durée de la formation	
Entité de formation	
Entité de recherche pilote	
Entités de formation associées	
Entités de recherche associées	
Responsables de la formation	
Principaux	
Associés	
Objectifs de la formation	
Objectif général	
Objectifs spécifiques	
Profils de sortie	
Modalités d'admission	
Profils d'entrée conseillés	
Formations requises	
Modalités particulières d'admission	
Débouchés	
Principales	
Secondaires	
Domaines d'auto-emploi	

Table 2: Tableau de présentation de l'offre de formation suivant la norme du REESAO

Présentation d'une Unité d'Enseignement	
Numéro du semestre	
Intitulé et code de l'UE	
Objectif général	
Objectifs spécifiques	
Contenu (principaux thèmes)	
Modalités	
Evaluation	
Enseignement	
Apprentissage	
Bibliographie de base	
Responsables de l'UE	
Principal	
Associés	
Composantes (ECU)	
Ressources	
<i>Ressources humaines</i>	
Personnel enseignant et qualification	
Personnel administratif	
Personnel Technique et de soutien	
<i>Ressources logistique et équipements pédagogiques nécessaires</i>	
Salles de cours	
Salles de TP	
Salles de TD	
Salles d'études	
Bibliothèque(s)	
Laboratoires	

Table 3: Tableau de présentation d'une Unité d'Enseignement

Présentation d'une ECUE	
Cours Théoriques (CT)	
Travaux Dirigés (TD)	
Travaux Personnel de l'Etudiant (TPE)	
Charge Total de Travail (CTT)	
CECT	
Modalités d'évaluation	
Contrôle Continu	
Examen Terminal	
Travaux pratiques	

Table 4: Tableau de présentation d'une ECUE

Code des UE	Contenu des enseignements		Enseignements			TPE	CTT	CECT	Modalités d'évaluation			Fonctionnalité des UE			Normes à respecter pour meubler le tableau
	UE	ECU	CT	TD	TP				CC	ET	CC+ET	Maj. 60%	Min/Opt. 30%	Lib. 10%	
Unités de Connaissances fondamentales															Total CECT = 12 à 16 CECT /UE= 4 à 6 Nombre d'UE = 3 ou 4 Semestres concernés L = 1 à 4 M = 1 & 2. Importance : décroissante
Unité de découverte ou de spécialité															Total CECT = 11 au plus CECT /UE= 2 à 4 Nombre d'UE = 3 à 5 Semestre concernés L = 1 à 6 M = 1 à 4. Importance : croissante
Unités de méthodologie															Total CECT = au plus 12 CECT /UE= 2 ou 3 Nombre d'UE = 1 à 4 Semestre concernés L = 1 à 6 M = 1 à 4. Importance : stable
Unités de culture générale															Total CECT = 1 à 4 CECT /UE = 1 ou 2 Nombre d'UE = 1 ou 2 Semestres concernés L = 1 à 4 ; M = 1 & 2. Importance : décroissante
TOTAL						750	30								
NB : l'ensemble des activités des stages peuvent être logé dans les UE de spécialité et/ou de Méthodologie !															

Table 5: Tableau synthétique présentant une offre de formation

B - Gestion informatisée des salle de cours

L'objectif de ce projet est d'automatiser la gestion des salles de l'UAC, à travers la mise en œuvre d'un système informatique sûr, efficace et équitable. Le système devra être

suffisamment flexible pour permettre de calculer des allocations des salles à plusieurs niveaux. Par exemple, il pourra calculer la répartition des salles d'une entité fonctionnant avec les salles dont elle dispose. Pour les autres entités, le système devra utiliser les salles affectées à la mutualisation des ressources, afin de calculer des allocations équitables. Le système devra aussi permettre d'avoir un suivi régulier de la tenue des formations. Le système devra finalement fournir des suggestions éclairées aux organes dirigeants de l'UAC pour qu'ils puissent décider le plus efficacement possible des politiques de construction de salles dans les prochaines années.

Les outils fournis par le système

Nous vous présentons maintenant la liste des outils que le système mettra à la disposition de la communauté universitaire. Cette liste restreinte pourra être étendue au grès des besoins spécifiques formulés dans les retours postérieurs à la lecture de ce rapport préliminaire.

Le calcul des emplois du temps

Nous prévoyons de calculer les emplois du temps tout d'abord pour toutes les entités. Ensuite le calcul sera fait pour les entités disposant de leur propre salle de formation. Enfin nous calculerons les emplois du temps pour les entités devant mutualiser leurs ressources tout en prenant en compte les modifications de dernières minutes.

Les emplois du temps seront donc présentés d'abord pour toute l'université, ensuite par entité de formation, puis par département, par groupe pédagogique et pour finir les modifications de dernières minutes.

Nous prévoyons aussi de présenter les statistiques d'occupation des salles pour toute l'université en premier lieu, ensuite par entité de formation, puis par département, par groupe pédagogique et enfin par enseignant.

Les niveaux d'accréditation du système

Le système informatique de gestion des salles doit permettre de garantir la confidentialité des données générées par son utilisation. Cela va se faire en contrôlant rigoureusement les accès des utilisateurs aux données. Nous proposons de créer des catégories de données accessibles que par les groupes d'utilisateurs réellement concernés par celles-ci. Nous associons pour cela à chaque groupe d'utilisateurs un niveau d'accréditation dans le système informatique.

En commençant à partir du niveau le plus élevé, ça donne les groupes suivants :

Les administrateurs

Ce groupe est formé des sous-groupes suivants :

1. Le Recteur de l'UAC;
2. Le Vice-Recteur chargé des affaires académiques;
3. Le Directeur du service de gestion des salles de cours;
4. Les membres du service de gestion des salles;
5. Les invités ayant une accréditation temporaire spécialement délivrée par le service de gestion.

Les enseignants

Ce groupe est formé des sous-groupes suivants :

1. Les Doyens / Directeurs des facultés / écoles / instituts;
2. Les vices-doyens / Directeurs Adjointes des facultés;
3. Les Chefs de Département;
4. Les enseignants;
5. Le personnel technique chargé des travaux pratiques.

Les étudiants

Ce groupe est formé des sous-groupes suivants :

1. Les responsables des Bureaux Universitaires des Étudiants;
2. Les responsables d'amphithéâtre (premier, deuxième et troisième responsables);
3. Les étudiants du groupe pédagogique.

Ils doivent s'inscrire dans chaque cours.

Les personnes externes au service de gestion des salles

Les contraintes liées au calcul des emplois du temps

Nous avons :

1. Affecter à chaque séance de cours une salle qui est la plus adaptée à cette séance.
2. S'assurer qu'il y a suffisamment de place pour que les étudiants qui doivent suivre ce cours puissent y assister.

Dans le contexte LMD, le nombre d'étudiants devant suivre un cours peut être différent du nombre d'étudiants de l'année d'étude pour laquelle ce cours est dispensé.

Dans le cas où le nombre d'étudiants devant suivre un cours est supérieur à la capacité maximale des salles disponibles, le système doit suggérer une subdivision de l'entité correspondante en groupe pédagogique. L'effectif maximal d'un groupe pédagogique est actuellement de six cent (600) étudiants. Après validation du vice-doyen de la faculté concernée par cette subdivision. Un nouvel emploi du temps pourra alors être calculé pour tenir compte de cette subdivision

validée et des nouvelles disponibilités des enseignants renseignées dans le système.

3. S'assurer que les formations se tiennent au calendrier fixé. Établir des emplois du temps corrects et ajustables en fonction des besoins.
4. Retracer simplement et à n'importe quel moment les cours qui ont été dispensés. Pouvoir identifier le nombre d'heures effectuées et le nombre d'heures restantes, par module de cours.
5. Ne pas attribuer à deux cours différents une même salle de classe.
6. Notifier aux utilisateurs du système, les changements de dernières minutes qui les concernent.
7. Faciliter la réservation des salles de conférence et de soutenance de mémoire.
8. Signaler un dysfonctionnement des salles au service de maintenance.
9. Une entité ne doit pas faire plus de 6 h pour un cours dans une journée.
10. Tous les cours programmés doivent se dérouler avant la fin du semestre.
11. Une salle ne doit pas être attribuée à deux entités différentes au même moment.
12. Une salle doit être attribuée en fonction des besoins du cours.
13. Une salle doit être adaptée au cours auquel il est attribué.
14. Le nombre de places dans une salle doit être supérieur ou égal au nombre d'étudiants dans l'entité à laquelle il est attribué.
15. Toute modification de l'emploi du temps doit être signalée aux membres de l'entité concernée.
16. Les étudiants doivent s'inscrire aux cours qu'ils doivent suivre.

V - Modélisation du système

Dans cette partie de notre travail, nous présenterons les différents diagrammes UML issuent de l'analyse du système.

B - Diagramme de classes

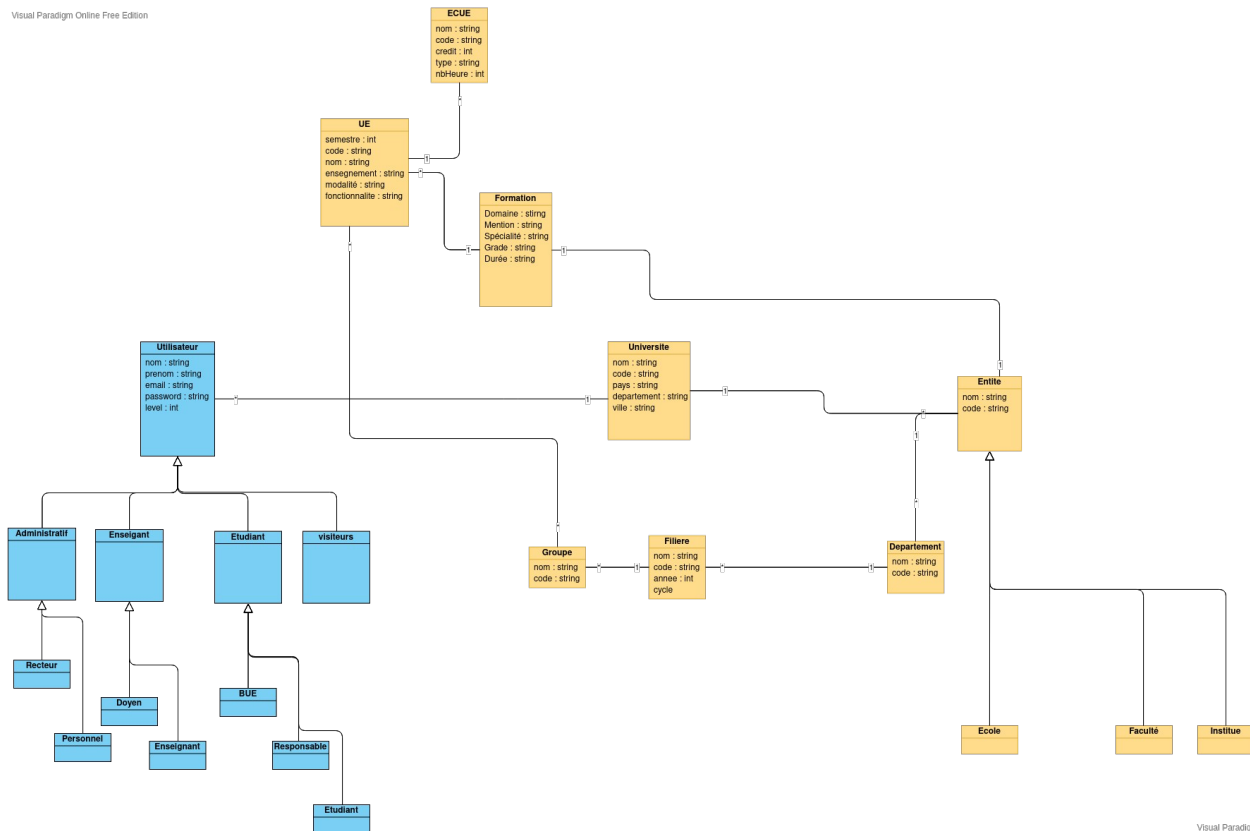


Figure 2: Diagramme de classes

C - Diagramme de cas d'utilisation

Notre système est composé de trois (03) grands acteurs décomposés en sous-acteurs avec des niveaux d'accréditation bien définis. Ces trois grands acteurs sont entre autres :

- Les administrateurs

- Les enseignants
- Les étudiants

Liste des cas d'utilisation par famille d'acteurs.

- Les administrateurs seront en mesure de :

- S'authentifier
- Gérer les formations
- Gérer les entités
- Gérer les filières
- Gérer les groupes
- Gérer les ECUE et UE
- Gérer les enseignants
- Gérer les étudiants
- Gérer les salles de cours
- Gérer les emplois du temps
- Envoyer des alertes
- Faire du tracking
- Consulter les emplois du temps
- Gérer le calendrier universitaire

Les enseignants seront en mesure de :

- S'authentifier
- Confirmer un cours
- Consulter l'emploi du temps
- Faire des demandes d'ajustement de l'emploi du temps
- Gérer les ECUE et UE les concernant

Les étudiants seront en mesure de :

- S'authentifier
- S'inscrire à un cours
- Consulter l'emploi du temps
- Consulter les ECUE et UE les concernant

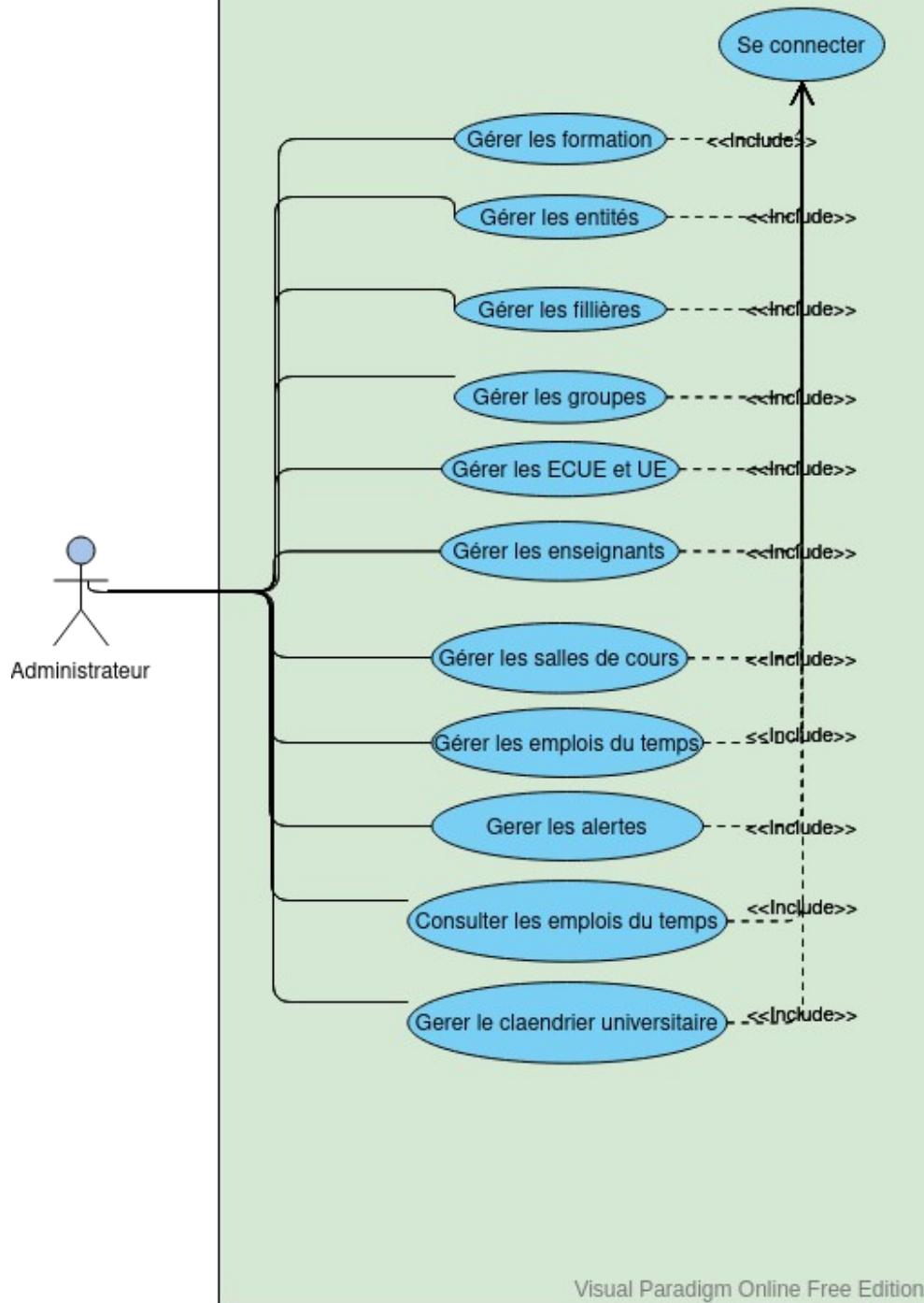


Figure 3: Diagramme de cas d'utilisation des administrateurs

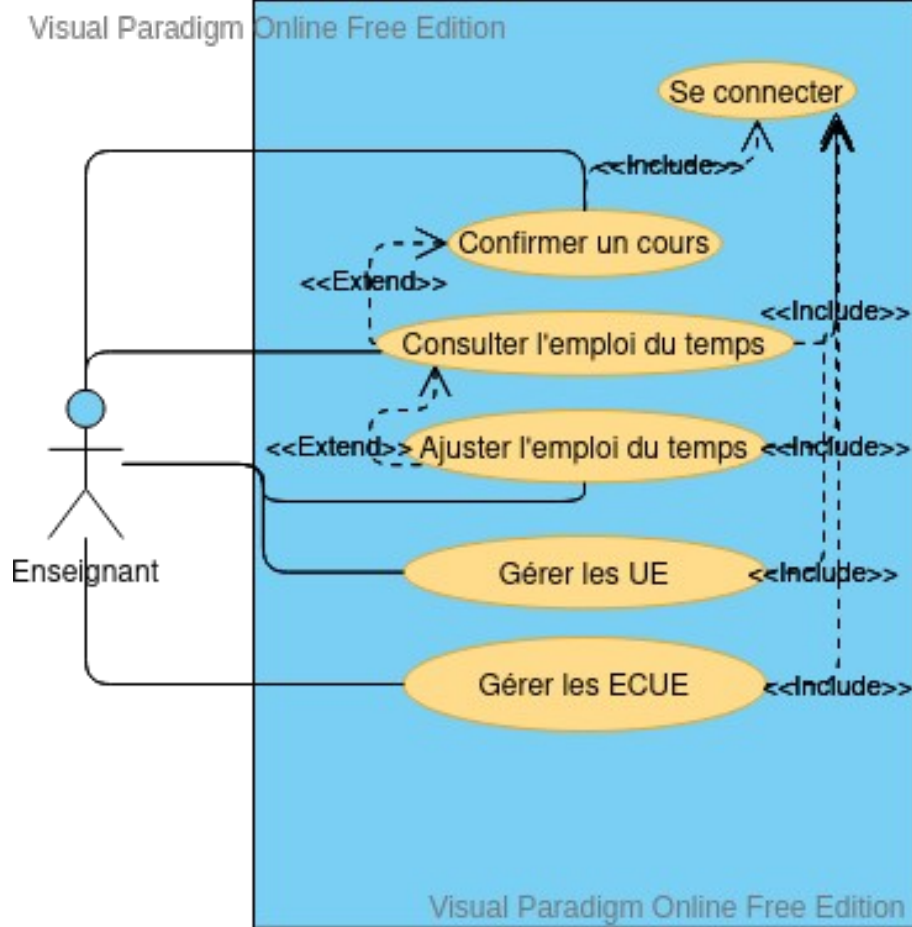


Figure 4: Diagramme de cas d'utilistation des enseignants

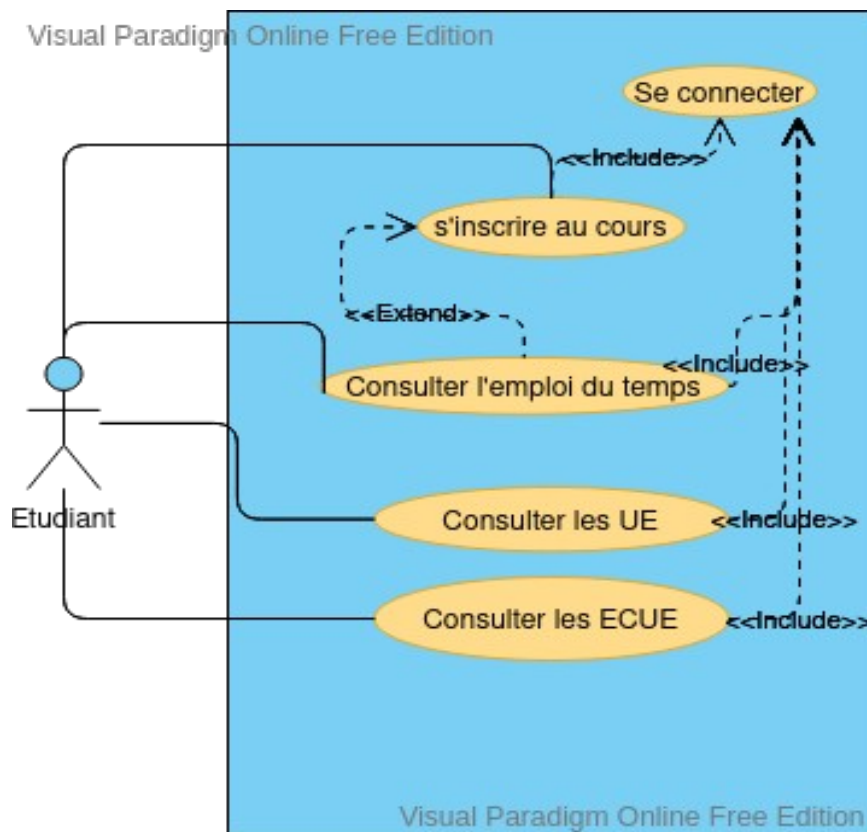


Figure 5: Digramme de cas d'utilisation des étudiants

IV – Réalisation de la solution

A - Presentation de l'achitecture du système

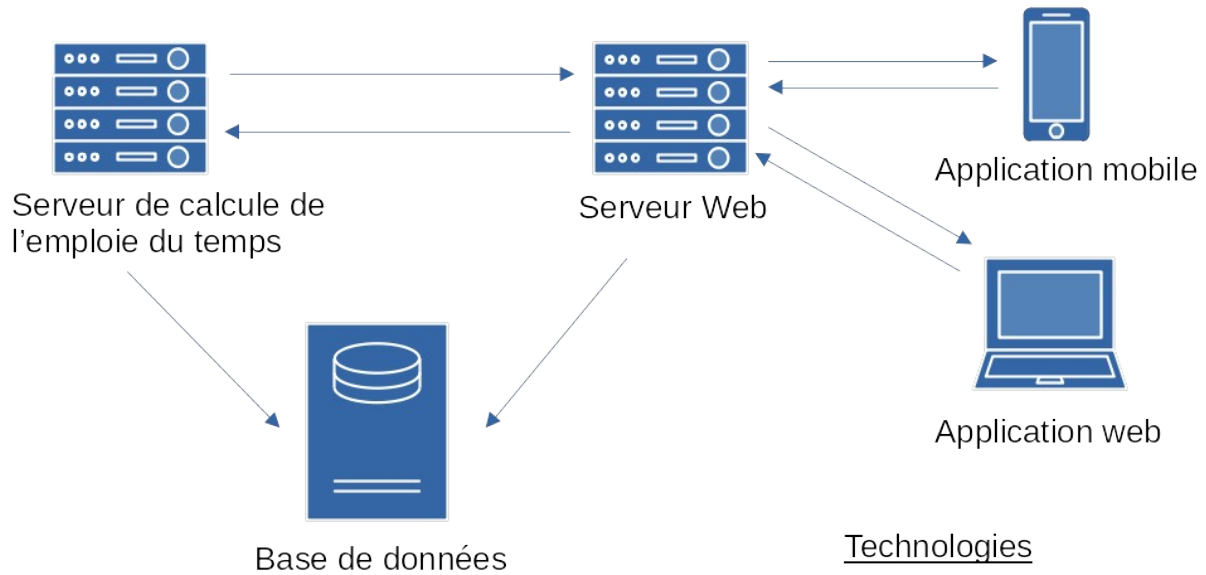


Figure 6: schéma de l'architecture du système

B – Les technologies utilisées lors du développement



Figure 7: Logo de python

Python est un langage de programmation interprété de haut niveau. Il est utilisé au niveau du serveur web et du serveur de calcul de l'emploi du temps.



Figure 8: Logo de django

Django est un framework web basé avancé sur python. Nous avons opté pour la version 3.0 de celle-ci pour concevoir l'API Rest du serveur web.



Figure 9: Logo de django rest framework

Django Rest framework comme son nom l'indique est une extension de django. Il est utilisé pour la conception de services web Rest.



Figure 10: Logo de postgresql

Postgresql est un système de gestion de bases de données gratuit et open source.



Figure 11: Logo de kotin

Kotlin est un langage de programmation open source créé par jetbrain à la base pour succéder java en matière de programmation mobile. Nous l'avons utilisé pour consommer notre API avec l'application mobile

Figure 12: Logo de JS



JavaScript est un langage de programmation web qui permet de créer des sites web dynamiques. Nous l'avons utilisé pour créer consommer notre API avec l'application web.

V – Présentation du noyau du système

A – Interface graphique

1 – Splash screen

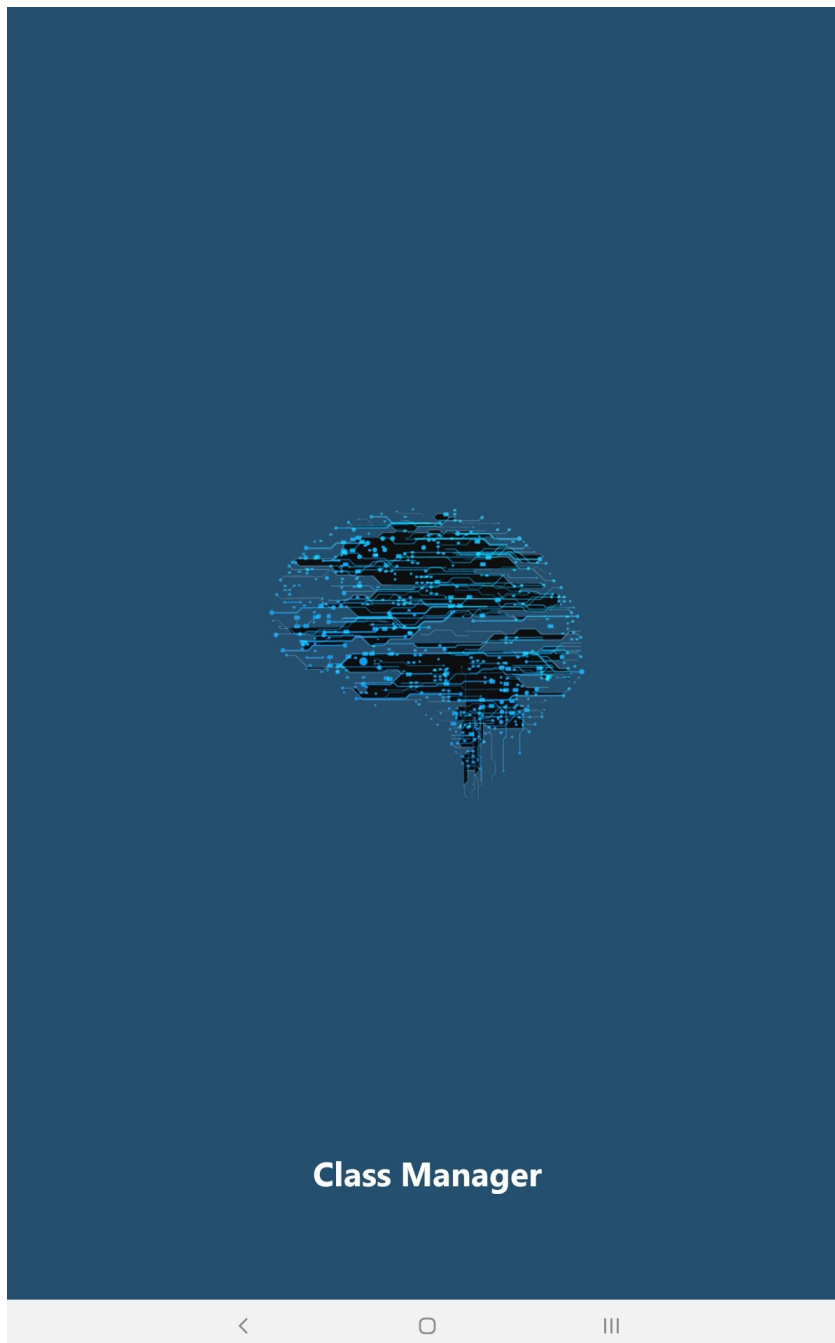
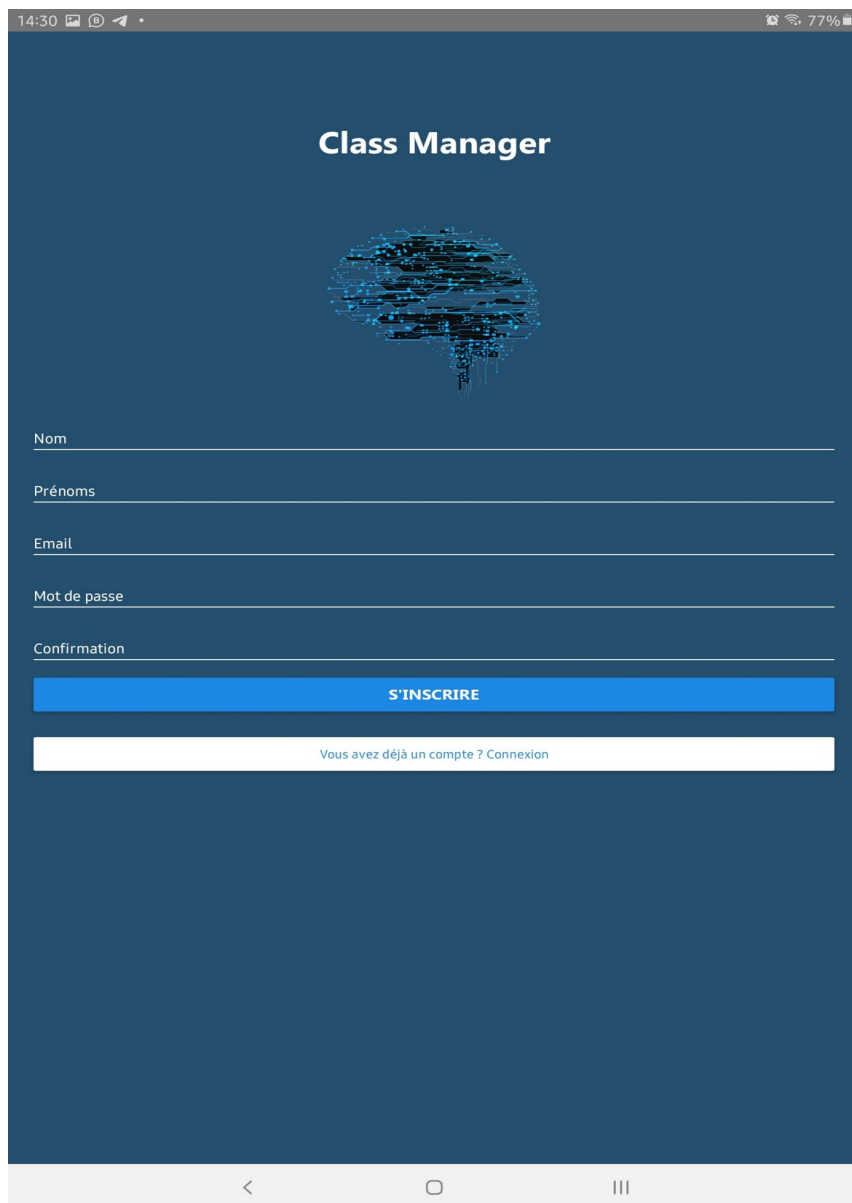


Figure 13: Splashscreen


2 – Inscription



The screenshot shows a mobile application interface for 'Class Manager'. The background is a dark blue gradient. At the top, the title 'Class Manager' is displayed in white. Below the title is a stylized graphic of a brain composed of binary code (0s and 1s). The registration form consists of five text input fields with labels: 'Nom', 'Prénoms', 'Email', 'Mot de passe', and 'Confirmation'. Below these fields is a prominent blue button with the text 'S'INSCRIRE' in white. Underneath the button is a white rectangular box containing the text 'Vous avez déjà un compte ? Connexion'. The top status bar of the phone shows the time as 14:30 and a battery level of 77%. The bottom navigation bar features three icons: a back arrow, a home circle, and a menu hamburger icon.

14:30 77%

Class Manager



Nom

Prénoms

Email

Mot de passe

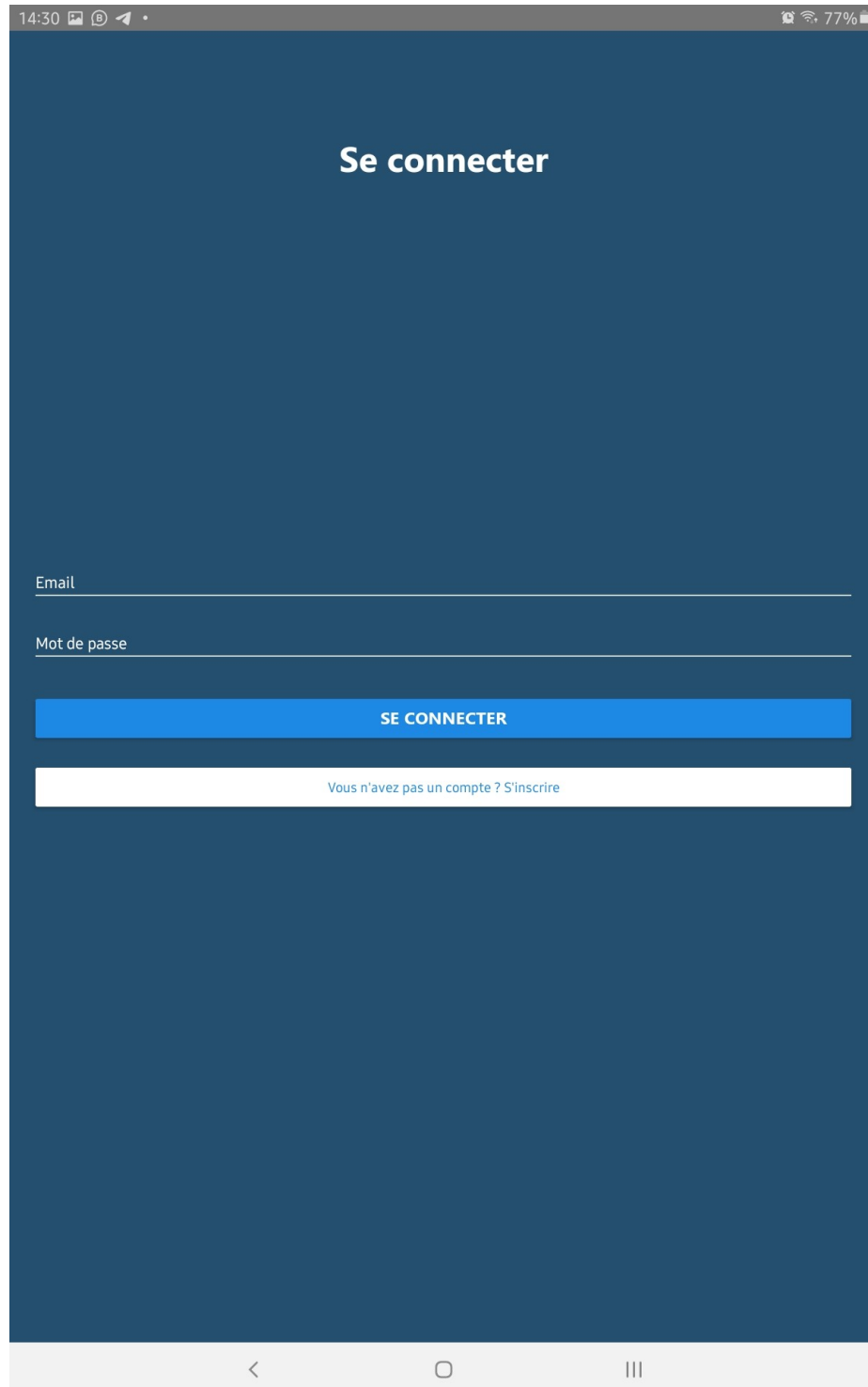
Confirmation

S'INSCRIRE

Vous avez déjà un compte ? Connexion

Figure 14: Interface d'inscription

3 – Connexion



The image shows a mobile application interface for logging in. At the top, there is a status bar with the time 14:30, signal strength, Wi-Fi, and battery level at 77%. The main background is a solid dark blue color. Centered at the top is the text "Se connecter" in white. Below this, there are two input fields: "Email" and "Mot de passe", both with white text and thin white borders. Under the password field is a blue button with the text "SE CONNECTER" in white. Below the button is a white rectangular box containing the text "Vous n'avez pas un compte ? S'inscrire" in blue. At the bottom of the screen is a white navigation bar with three icons: a back arrow, a circle, and three vertical bars.

14:30 77%

Se connecter

Email

Mot de passe

SE CONNECTER

Vous n'avez pas un compte ? S'inscrire

Figure 15: Interface de connexion

Conclusion

Nous nous sommes intéressés aux problèmes difficiles de la gestion des salles de cours dans une université de type Université d'Abomey-Calavi. Après avoir présenté ce problème, nous avons tout d'abord étudié les différents systèmes et algorithmes existants permettant de résoudre le problème. Nous avons ensuite étudié le fond du problème en faisant le point sur la manière dont l'université est organisée puis nous avons trouvé une approche pour informatiser ce qui est fait actuellement à l'UAC. Enfin nous avons modélisé et démarré la conception de notre solution.

Au sujet de notre développement, nous avons déjà établi un système de gestion des utilisateurs et des ressources. Il nous reste à présent à implémenter la solution pour l'élaboration des emplois de temps et le suivi des ressources.

Étant donné l'importance de notre contribution pour permettre à nos universités d'utiliser au mieux les ressources dont elles disposent, nous allons poursuivre notre travail et l'achever.

Références

- Model and Algorithms for school Timetabling – Michael Marte
- Résolution du problème du temps : Proposition d'un algorithme évolutionnaire multi objectif – Mme Troudi Fatiha
- Méthode et outils d'optimisation – Matias Kleiner
- Contribution à la réalisation du problème d'emploi de temps par une approche évolutionnaire – Boukerroucha
- <http://www.tangentex.com/RecuitSimule.htm>
- <https://www.timetabler.com/timetabler/>
- <https://www.unitime.org/>
- <https://github.com/UniTime/unitime>
- <https://www.lalescu.ro/liviu/fet/>
- [https://fr.wikipedia.org/wiki/Recuit simul%C3%A9](https://fr.wikipedia.org/wiki/Recuit_simul%C3%A9)
- [https://fr.wikipedia.org/wiki/Algorithme %C3%A0 directions de descente](https://fr.wikipedia.org/wiki/Algorithme_%C3%A0_directions_de_descente)
- [https://fr.wikipedia.org/wiki/Algorithme g%C3%A9n%C3%A9tique](https://fr.wikipedia.org/wiki/Algorithme_g%C3%A9n%C3%A9tique)
- https://perso.liris.cnrs.fr/pierre-edouard.portier/teaching_2015_2016/ia/sima/sima.html
- <https://online.visual-paradigm.com>
- <https://www.django-rest-framework.org/>
- <https://docs.djangoproject.com/fr/3.0>