

Liste des symboles et abréviations

UVT : Université Virtuelle de Tunis
CNTE : Centre National des Technologies en
Eduserv : Education Services
XML : eXtensible Markup Langage ou Langage Extensible de Balisage
HTML : HyperText Markup Language
PHP : Hyper Text Preprocessor
ADT : Android Development Tools
IDE : Integrated Development Environment
SDK : Software Development Toolkit
OS : Operating System
EPL : Eclipse Public License
UML : Unified Modeling Language
OMT : Object Modeling Technique
OOSE : Object Oriented Software Engineering
MySQL : My Structured Query Language
JSON : JavaScript Object Notation
UP : Unified Process
2TUP : Two Tracks Unified Process
HTTP : Hypertext Transfer Protocol
URL : Uniform Resource Locator

Table des matières

Introduction Générale	1
Chapitre 1 : Présentation Générale	3
I. Introduction	3
II. Présentation de la société d'accueil	3
1. Historique de CNTE.....	3
2. Services et missions du CNTE	4
III. Contexte.....	5
IV. Problématique	6
V. Travail à Réaliser	7
VI. Méthodologies du travail	8
1. Méthodologie 2TUP.....	8
2. Langage de conception.....	10
VII. Conclusion.....	11
Chapitre 2 : Etude préliminaire.....	12
I. Introduction	12
II. Etude de l'existant	12
1. Existant au CNTE.....	12
2. Applications Web pour la gestion de la vie scolaire	13
III. Critiques de l'existant	16
1. Limites d'Eduserv.....	16
2. Limites des applications Web.....	16
IV. Solutions Proposées	16
1. Première solution	16
2. Deuxième solution.....	17
V. Solution retenue	17
VI. Conclusion.....	18
Chapitre 3 : Branche fonctionnelle	19
I. Introduction	19

II. Capture des besoins	19
1. Besoins fonctionnels détaillés	19
2. Besoins non fonctionnels détaillés.....	20
III. Besoins techniques	20
IV. Analyse des fonctionnalités.....	20
1. Identification des acteurs	20
2. Diagramme de cas d'utilisation	23
3. Description des cas d'utilisation	24
V. Conclusion	51
Chapitre 4 : Branche technique	52
I. Introduction	52
II. Architecture.....	52
III. Diagramme de déploiement :	54
IV. Diagrammes des séquences	54
1. Diagramme de séquence détaillé du cas « s'authentifier »	55
2. Diagramme de séquence détaillé du cas « Gestion des Rendez-vous»	56
3. Diagramme de séquence détaillé du cas « Consulter les notes»	56
V. Diagramme de classes.....	57
VI. Conclusion.....	58
Chapitre 5 : Réalisation	59
I. Introduction	59
II. Environnement du travail	59
1. Environnement matériel	59
2. Environnement logiciel.....	60
III. Choix techniques	61
1. Choix de développement	61
2. Protocole et formats de données	62
3. Choix de la technologie de sécurité	63
IV. Présentation de quelques interfaces	63
1. Logo de l'application :	63

2. Interface d'authentification	64
3. Interface "Parent de"	64
4. Interface « Divertissement »	65
5. Interface "Notes"	65
6. Interface "Emploi du temps"	66
7. Interface "Liste enseignants"	66
V. Conclusion	67
Conclusion générale	68
Webographie	70
Annexe	71

Table des figures

Figure 1: Le processus 2TUP.....	9
Figure 2: Espace numérique pour les établissements préparatoires et secondaires	13
Figure 3: Classement des OS en Tunisie	18
Figure 4 : Diagramme de cas d'utilisation	24
Figure 5: cas d'utilisation gestion des comptes.....	25
Figure 6: Cas d'utilisation Gestion des notes.....	29
Figure 7: Cas d'utilisation gestion de l'agenda.....	35
Figure 8: Cas d'utilisation gestion des notifications	39
Figure 9: Cas d'utilisation gestion des appréciations.....	43
Figure 10: Cas d'utilisation gestion des rendez-vous	47
Figure 11: Architecture de notre application	52
Figure 12: Diagramme de déploiement	54
Figure 13: Diagramme de séquence détaillé du cas « s'authentifier ».....	55
Figure 14: Diagramme de séquence détaillé du cas « Gestion des Rendez-vous»	56
Figure 15: Diagramme de séquence détaillé du cas « Consulter les notes»	56
Figure 16: Diagramme de classes	58
Figure 17: Logo de l'application	63
Figure 18: Interface d'authentification	64
Figure 19: Interface d'authentification avec menu "hors connexion"	64
Figure 20: Interface "Parent de".....	64
Figure 21: Interface "Parent de" avec menu	64
Figure 22: Interface "Divertissement"	65
Figure 23: Interface "Notes"	65
Figure 24 : Interface "Emploi du temps"	66
Figure 25: Interface "Liste enseignants"	66
Figure 26: Cycle de vie d'une application Android	73

Table des tableaux

Tableau 1: Identification de l'acteur "professeur"	21
Tableau 2: Identification de l'acteur "personnel administratif"	22
Tableau 3: Identification de l'acteur "Administrateur"	22
Tableau 4: Identification de l'acteur "Parent"	23
Tableau 5: Identification de l'acteur "Elève"	23
Tableau 6: Scénario « Créer un compte »	26
Tableau 7: Scénario « Consulter un compte »	26
Tableau 8: Scénario « Modifier un compte »	27
Tableau 9: Scénario « Consulter Emploi du temps»	28
Tableau 10: Scénario « Consulter Emploi du temps, Professeur»	29
Tableau 11: Scénario « saisir une note»	30
Tableau 12: Scénario « Consulter les notes »	31
Tableau 13: Scénario « Modifier une note »	32
Tableau 14: Scénario « Consulter Absences »	33
Tableau 15: Scénario « Consulter Absences »	34
Tableau 16: Scénario « Ajouter une tâche »	36
Tableau 17: Scénario « Consulter Agenda»	36
Tableau 18: Scénario « Modifier une tâche »	37
Tableau 19: Scénario « Supprimer une tâche »	38
Tableau 20: Scénario « Ajouter une notification »	40
Tableau 21: Scénario « Consulter les notifications »	40
Tableau 22: Scénario « Modifier une notification »	41
Tableau 23: Scénario « Supprimer une notification »	42
Tableau 24: Scénario « Ajouter une appréciation »	44
Tableau 25: Scénario « Consulter une appréciation »	44
Tableau 26: Scénario « Modifier une appréciation »	45
Tableau 27: Scénario « Supprimer une appréciation »	46
Tableau 28: Scénario « Demander un rendez-vous»	48
Tableau 29: Scénario « Consulter les rendez-vous»	49
Tableau 30: Scénario « Consulter Emploi du temps»	49
Tableau 31: Scénario « Purger système »	50
Tableau 32 : Présentation de l'environnement de travail utilisé	59

Introduction Générale

Pour évoluer et accroître leurs efficacités, toutes les organisations, quelle que soit leurs natures, ont besoin aujourd'hui d'assimiler la culture de l'innovation portée par les nouvelles technologies de l'information.

Actuellement les Technologies de l'Information et de la Communication (TIC) connaissent un développement vertigineux, ce qui concerne presque tous les domaines de notre société, dont l'éducation.

Ces technologies se présentent de plus en plus comme une nécessité dans le contexte de la société où les changements rapides, l'augmentation des connaissances et les demandes d'une éducation de haut niveau constamment mise à jour se transforment en une exigence permanente.

En effet, la place des applications smartphones et tablettes n'est plus à faire aujourd'hui. Avec un pourcentage croissant des utilisateurs des smartphones et des tablettes, les applications prennent de plus en plus d'espace et de poids dans l'utilisation de nos terminaux mobiles.

Et c'est dans ce cadre que se déroule notre projet au sein du Centre National des Technologies en Education. Le projet consiste à concevoir et développer une application mobile « Android » de services scolaires pédagogiques et administratifs.

Le présent rapport est composé de cinq chapitres :

- Le premier chapitre « Présentation générale » consiste à mettre notre projet dans son cadre général. Il présente l'organisme d'accueil et la méthodologie utilisée dans notre application.
- Le deuxième chapitre « Etude préliminaire » présente l'étude de l'existant, les besoins fonctionnels et non fonctionnels du système, les différents acteurs, et le diagramme de contexte statique et dynamique.
- Le troisième chapitre « Branche fonctionnelle » identifie et décrit tous les cas d'utilisation et les organise en package.

- Le quatrième chapitre « Branche technique » présente l'architecture de notre système ainsi que le diagramme de déploiement et les diagrammes de séquences détaillés.
- Le cinquième chapitre « Réalisation » présente les différents outils de développement utilisés, la phase de test et de validation et quelques interfaces de l'application.

Chapitre 1 : Présentation Générale

I. Introduction

J'entame ce chapitre par présenter le cadre général de notre travail. Ensuite, dans la deuxième section, je vais présenter l'entreprise accueillante, le Centre National des Technologies en Education (CNTE) ainsi que notre projet. Et finir par exposer la méthodologie de conception.

II. Présentation de la société d'accueil

Le Centre National des Technologies en Education (CNTE), sous tutelle du ministère de l'éducation, a pour mission de développer et d'intégrer les technologies de l'information et de la communication dans le système éducatif tunisien¹.

1. Historique de CNTE

Le Centre National des Technologies en Education a été créé en 1984, l'année 1988 a été le point de départ du projet de "l'Informatique pour Tous", l'informatique dans les établissements scolaires.

En 1989 un processus de décentralisation de l'institut a été engagé. Des centres régionaux ont ainsi été créés dans 14 gouverneras.

En 1990, le CNTE a fait l'objet d'une restructuration, elle a équipé 100 établissements secondaires de laboratoires d'informatiques, en parallèle avec la mise en œuvre d'un programme de formation des professeurs en bureautique et en informatique. C'est au cours de cette année que l'informatique a été inscrite comme discipline d'enseignement en classe terminale de l'enseignement secondaire.

En 1998, l'institut est devenu un fournisseur de services Internet dans le secteur de l'éducation : il a été ainsi procédé à la connexion de 30% des lycées à Internet et 1000 professeurs ont pu bénéficier, à titre gratuit, d'un abonnement Internet et par conséquent acquérir une adresse électronique. Le Portail Educatif Tunisien (Edunet) a été créé la même année. En 1999, le projet "l'Internet pour Tous" a offert la possibilité à tous les

¹ www.cnte.tn

établissements secondaires de se connecter au réseau et ce, par l'intermédiaire de la ligne téléphonique.

En 2002, le centre a commencé les opérations d'équipement des écoles primaires d'espaces d'informatique et de connexion à Internet.

En 2005, les sites web des écoles préparatoires et des lycées qui ont été hébergés sur Edunet ont atteint le nombre des agents administratifs et des professeurs qui possèdent une adresse électronique sur Edunet est de 30.000. En ce qui concerne le pourcentage des collèges connectés à Internet, il dépasse 89% tandis que celui des écoles primaires est déjà de 50%.

En 2008, L'ADSL généralisé dans les établissements scolaires et éducatifs. Le ministère de l'Education et de la Formation a signé un contrat de prestation de service avec Tunisie Télécom au terme duquel les établissements scolaires (écoles primaires, collèges et lycées, centres de formation professionnelle, etc.) seront raccordés au réseau Internet haut débit (ADSL) et ce dans le cadre du développement du Réseau Educatif National (EDUNET 2).

2. Services et missions du CNTE

Le Centre National des Technologies en Education a pour mission de participer à la mise en place de la stratégie nationale du développement de l'intégration des technologies de l'information et de la communication dans le système éducatif et d'assurer le suivi de son exécution notamment dans le domaine de l'enseignement, l'apprentissage et la formation².

- Développer un système d'information éducatif global et intégré fournissant des services pédagogiques et de formation.
- Développer des contenus pédagogiques numériques pour tous les cycles d'enseignement selon les programmes officiels en vigueur.
- Fournir au secteur de l'éducation des services relevant de ses attributions qui s'adaptent aux progrès des technologies de l'information et de la communication, y compris les services Internet.

² www.cnte.tn

- Contribuer au développement des compétences de ressources humaines du secteur de l'éducation dans l'intégration des technologies de l'information et de la communication.
- Réaliser les études et les analyses nécessaires convergeant vers l'exploitation des technologies de l'information et de la communication dans le système éducatif.
- Assurer la veille technologique, identifier et valoriser les expériences et les innovations dans le domaine de l'intégration des technologies de l'information et de la communication dans le système éducatif.
- Veiller à la rationalisation de l'exploitation des réseaux, des équipements et des applications informatiques pédagogiques au sein des établissements scolaires.
- Organiser des ateliers et des manifestations scientifiques et pédagogiques dans le domaine de l'intégration technologique.
- Promouvoir la coopération et conclure des conventions avec :
 - Les organisations et les établissements nationaux : publiques et privés.
 - Les organisations et les établissements internationaux, et cela après l'accord de la tutelle.
- Fournir des services et offrir des consultations au tiers dans le domaine des technologies de l'information et de la communication dans l'enseignement, l'apprentissage et la formation, et cela moyennant rémunération.

III. Contexte

Nous sommes aujourd'hui dans une situation où aucun parent n'est assuré que son enfant vivra dans le futur mieux que lui. C'est une rupture profonde avec ce qui a existé depuis deux siècles où la notion mythifiée, fantasmée, de l'ascension sociale a été un moteur puissant de l'espoir dans le futur.

Pendant des dizaines d'années restées dans les mémoires, l'école a été un symbole de continuité et de permanence bien évoqué dans de nombreux romans ou des films. Depuis plus d'une génération, l'école apparaît en crise permanente de plus en plus incompréhensible³.

³ <http://www.cafepedagogique.net/>

Trop souvent, encore, c'est aux familles de tenter de participer directement à la vie scolaire. Si les familles et notamment celles les plus éloignées de l'école ne sont pas accompagnées dans leur découverte du système scolaire actuel, il y a véritablement un risque que ces populations rejettent une école qu'elles ne comprennent pas et soient sensibles à toutes les campagnes visant à disqualifier l'école publique.

La question de la construction de relations de confiance entre les parents et les professeurs est aujourd'hui une question centrale pour tous les établissements scolaires pour donner plus de sens à l'école. Toutes les recherches menées montrent qu'un dialogue constant entre parents et professeurs, ce qui implique de ne pas « convoquer les parents que lorsqu'il y a une difficulté », mais de les « inviter à venir parler de leur enfant », qu'une véritable coopération, les uns s'appuyant sur les autres, entre les familles et l'école, permet un meilleur apprentissage des jeunes et amplifie leur réussite, et ceci par :

- Installer un espace « parents » au cœur de l'école ou de l'établissement
- Mettre en place différents moyens de communication avec les familles : tableaux d'affichages, cahier d'accueil, cahier de liaison, cahier de vie de la classe mais également communication orale
- Travailler avec les associations locales ou les associations de quartiers, notamment lorsque l'école accueille des enfants d'origine étrangère ou des enfants de la communauté du voyage
- Mettre en place un site informatique.
- Etc.

IV. Problématique

La plupart des lycées secondaires possèdent des systèmes informatiques qui stockent les informations (nom, prénom, notes, etc.) des élèves inscrits. Une application personnalisée pour la conception des emplois du temps est aussi présente, cette solution permet l'impression de ceux-ci afin de les distribuer aux élèves. Les professeurs relèvent les absences dans leurs classes et les transmettent à l'administration qui les saisit dans

le système. Les notes sont saisies dans le logiciel par les professeurs. Les informations sont diffusées aux parents sous forme de papier et souvent par l'intermédiaire des élèves. Ce moyen de communication est très problématique, il a des pertes et les élèves peuvent interférer dans son bon fonctionnement.

Pour les travaux fournis par les professeurs, il n'y a pas de support informatique. Les rendez-vous avec les professeurs ne sont pas gérés par l'école.

Actuellement tous les établissements secondaires possèdent les logiciels nécessaires au fonctionnement interne mais ne possèdent pas de solutions pour diffuser ces informations. La mise en place de ce système doit permettre d'améliorer la communication entre les parents et l'école et ne plus dépendre de l'élève pour transmettre l'information.

Cette mise en place va demander plus de moyen pour assurer la maintenance du système d'information de l'établissement mais aussi une forte implication de tous les employés de l'école pour faire vivre le faire vivre.

V. Travail à Réaliser

Pour remédier aux difficultés présentées dans la section précédente, je propose de concevoir et mettre en place une application Android pour la gestion de la vie scolaire. Cette solution doit fournir un accès à distance aux notes, à l'emploi du temps, aux absences, aux retards et d'autres, à travers une application Android connectée à Internet. De plus, les rendez-vous pourront être directement pris via l'application ainsi que les outils d'aide pour le travail à la maison seront mis à disposition des parents.

Le nouveau produit concernera aussi bien les parents pour la consultation et l'interaction, que l'école pour la mise à disposition des informations et l'utilisation du système. Ce dernier doit être considéré comme une extension du système actuel et comme premier pas vers la modernisation de l'existant.

Cette extension proposera aux parents, l'accès à un espace personnel dans lequel ils auront accès en consultation à toutes les informations concernant leurs enfants inscrits dans l'établissement. Pour cela l'école devra permettre à l'ensemble du

personnel (secrétaires, professeurs, responsable, etc.) de renseigner les données nécessaires concernant les élèves et les cours suivis par ces derniers. Cette application a pour but de faciliter la communication parent-école et de fournir tous les outils nécessaires pour s'impliquer dans la scolarité de leurs enfants.

VI. Méthodologies du travail

Dans cette partie, je définisse le langage et la méthode de cycle de vie utilisée tout au long du projet afin de modéliser d'une manière claire et précise la structure et le comportement de notre système indépendamment de tout langage de programmation.

1. Méthodologie 2TUP

Un processus définit une séquence d'étapes ordonnées qui permet de produire un système logiciel ou de faire évoluer un système existant. Un processus est décomposé suivant l'axe de développement technique et l'axe de gestion du développement. L'axe de développement technique se préoccupe de la production tandis que la gestion du développement mesure, puis prévoit les coûts et les délais.

La diversité des systèmes et des techniques de modélisation ne permet pas de définir un seul processus universel. Les créateurs d'UML ont travaillé à unifier les meilleures pratiques de développement orienté objet pour donner le processus unifié.

Le processus unifié (*UP : Unified Process*) est un processus de développement logiciel mené par UML. Le processus unifié est :

- itératif et incrémental : le projet est découpé en des itérations de courte durée. Ces itérations aident à mieux suivre l'avancement du système global. A chaque itération, il est produit un exécutable de façon incrémentale.
- piloté par les risques : il est identifié et écarté au plus tôt tout risque pouvant conduire à un échec du projet.
- centré sur l'architecture : le système est décomposé en modules pour des besoins de maintenabilité et d'évolutivité.

- conduit par les cas d'utilisations : le processus met en avant les besoins et exigences des futurs utilisateurs du système.

2TUP est un processus unifié qui a pour but d'apporter une réponse aux contraintes de changement fonctionnelles et techniques qui s'imposent aux systèmes d'information. 2TUP propose un cycle de développement qui dissocie les aspects techniques des aspects fonctionnels. Il part du constat que toute évolution imposée au système d'information peut se décomposer et se traiter parallèlement, suivant un axe fonctionnel et un axe technique. Il distingue ainsi deux branches (fonctionnelle et technique) dont les résultats sont fusionnés pour réaliser le système. On obtient un processus de développement en Y comme l'illustre la figure suivante (Figure 1). Il faut noter que le processus 2TUP commence d'abord par une étude préliminaire. Dans cette étude préliminaire, il s'agit d'identifier les acteurs qui vont interagir avec le système, les messages qu'échangent les acteurs et le système, puis à produire un cahier de charges et enfin à modéliser le contexte.

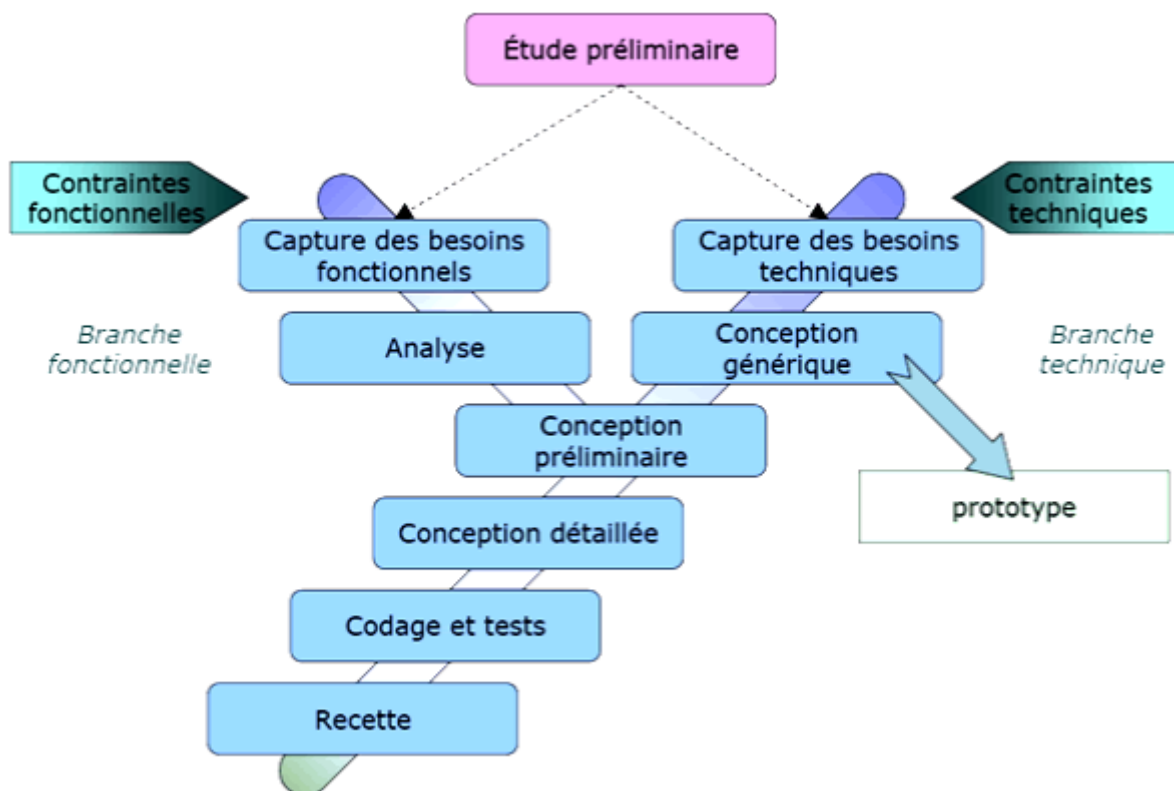


Figure 1: Le processus 2TUP

2. Langage de conception

Le langage de modélisation unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement et en conception orientée objet⁴.

L'UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

- Activité d'un objet/logiciel
- Acteurs
- Processus
- Schéma de base de données
- Composants logiciels
- Réutilisation de composants

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement une partie de code, par exemple en langage Java, à partir des divers documents réalisés.

UML 2.3 propose 14 types de diagrammes (9 en UML 1.3). UML n'étant pas une méthode, leur utilisation est laissée à l'appréciation de chacun, même si le diagramme

⁴ [http://fr.wikipedia.org/wiki/UML_\(informatique\)](http://fr.wikipedia.org/wiki/UML_(informatique))

de classes est généralement considéré comme l'élément central d'UML, des méthodologies, telles que l'Unified Process, axent l'analyse en tout premier lieu sur les diagrammes de cas d'utilisation (Use Case). De même, on peut se contenter de modéliser seulement partiellement un système, par exemple certaines parties critiques.

UML se décompose en plusieurs sous-ensembles :

- **Les vues** : Les vues sont les observables du système. Elles décrivent le système d'un point de vue donné, qui peut être organisationnel, dynamique, temporel, architectural, géographique, logique, etc. En combinant toutes ces vues, il est possible de définir (ou retrouver) le système complet.
- **Les diagrammes** : Les diagrammes sont des éléments graphiques. Ceux-ci décrivent le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues.
- **Les modèles d'élément** : Les modèles d'élément sont les briques des diagrammes UML, ces modèles sont utilisés dans plusieurs types de diagrammes. Exemple d'élément : cas d'utilisation (CU ou cadut'), classe, association, etc.

VII. Conclusion

Dans ce chapitre, j'ai présenté l'entreprise d'accueil le Centre National des Technologies en Education, le cadre général du travail et la méthodologie de travail utilisée. Dans le chapitre suivant, je passe à la phase de l'étude préliminaire.

Chapitre 2 : Etude préliminaire

I.Introduction

Dans ce chapitre, je commence par l'étude de l'existant pour une meilleure analyse des besoins. Ensuite, je présente les spécifications des besoins afin de déterminer et préciser les différentes fonctionnalités attendues de notre application. Puis, j'identifie les acteurs et les messages émis et reçus par le système. Enfin, je mets l'accent sur le diagramme de contexte statique et le diagramme de contexte dynamique.

II.Etude de l'existant

1. Existant au CNTE

Le Centre National des Technologies en Education possède un espace numérique⁵, pour les établissements préparatoires et les lycées secondaires, qui permet aux élèves, parents, professeurs et aux personnels administratifs d'accéder à ces espaces numériques depuis n'importe quel ordinateur connecté à l'Internet afin de consulter l'ensemble de services numériques proposés à l'établissement scolaire et à sa communauté éducative.

⁵ [http:// http://www.eduserv.tn](http://www.eduserv.tn)

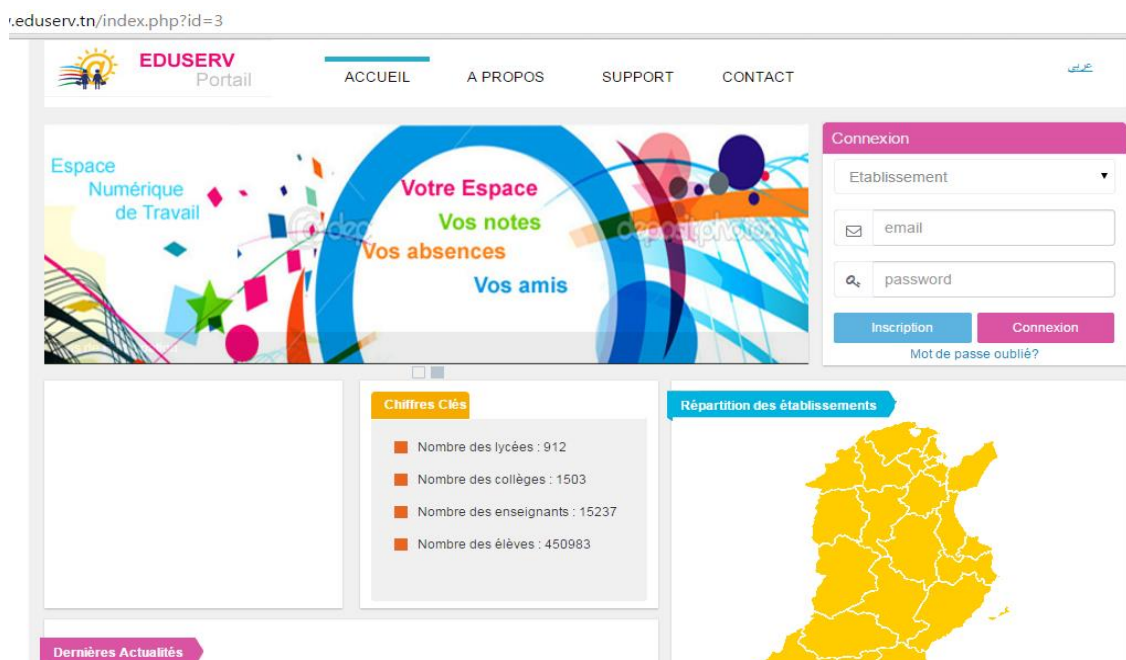


Figure 2: Espace numérique pour les établissements préparatoires et secondaires

Cet espace comporte un ensemble de services numériques proposés à l'établissement scolaire et à sa communauté éducative sous forme de portail web unique et sécurisé tels que :

- de vie scolaire : notes, absences, emplois du temps, punitions, etc.
- pédagogiques : cahier de texte numérique, banque de ressource, etc.
- de communication : messagerie, etc.

2. Applications Web pour la gestion de la vie scolaire

On désigne sous l'appellation de "Vie scolaire" toutes les tâches incombant majoritairement au Conseiller Principal d'Education (CPE) et aux surveillants, en relation avec les enseignants et l'équipe administrative. La dénomination et l'attribution des tâches changent selon les pays mais partout, elles assurent le bon fonctionnement de l'établissement d'enseignement.

Concrètement, le relevé des absences, la gestion des notes, l'information aux familles, la répartition des groupes dans les salles, la gestion des emplois du temps... font partie de ces tâches.

De nombreuses applications informatiques, accessibles sur poste ou en ligne, facilitent la vie des responsables de la vie scolaire. Ces applications sont de plus en plus

fréquemment intégrées sous forme de modules aux ENT (environnements numériques de travail) qui fournissent des services aux enseignants et aux élèves. Mais on trouve encore des applications de Vie scolaire autonomes, qui néanmoins sont généralement compatibles / intégrables aux ENT, étendant les fonctionnalités de ces derniers au-delà du pédagogique.

Une offre pléthorique d'ENT et d'applications spécifiques pour la Vie scolaire pour les lycées secondaires dont on cite :

Gest'Ecole : Est un logiciel GRATUIT de gestion d'un établissement. Il est la suite logique de Gest'Classe, logiciel de gestion d'une classe qui est développé de 2007 à 2011. Devant l'intérêt des professeurs et des demandes multiples d'avoir un même outil pour plusieurs classes, pour historier le parcours scolaire des élèves et pour offrir un outil commun au directeur, aux professeurs et aux parents, **DOX Conception** a décidé d'abandonner Gest'Classe et de lancer Gest'Ecole.

Ainsi, Gest'Ecole a pour but de permettre de gérer une école maternelle, élémentaire ou primaire. C'est un outil à destination des directeurs d'école, des professeurs et des parents.

En effet, Gest'Ecole permet :

- Au directeur d'établissement, de gérer les personnels de son établissement, les classes, les élèves, les livrets, les absences, etc. et ceci sur plusieurs années scolaires.
- Aux professeurs de gérer leurs élèves de l'année, les livrets scolaires, les absences, leur cahier-journal etc.
- Aux parents de suivre les activités de leur enfant, leurs devoirs, les livrets scolaires année par année, etc.

Avec le lancement de la version 2, **Gest'Ecole** s'est ouverte à l'international, En effet, l'application peut être entièrement traduite dans plusieurs langues. De plus, les

paramètres de découpage de l'année scolaire peuvent directement être définis par le directeur de l'établissement.

a. Zeus :

« Logiciel libre en ligne pour la gestion de la vie scolaire »

Un ensemble de modules destinés à l'usage des éducateurs, des professeurs, de la direction, voire de l'administration dans les écoles d'enseignement secondaires. En tous les cas, c'est dans le cadre d'un établissement d'enseignement secondaire que l'application fonctionne actuellement.

- une somme d'informations techniques et légales sur les fonctions d'enseignants. Autorisation de livraison éventuellement possibles moyennant accord de l'auteur.
- un module d'accès aux données exportées depuis EDT (logiciel de conception des horaires) pour rendre les informations consultables dans un navigateur web
- un module de gestion des passages à l'infirmerie
- un bulletin électronique, y compris le module d'assistance à la délibération et le carnet de notes en ligne
- un bloc-notes de fiches d'élèves
- le module d'administration de l'ensemble des applications

Chacune de ces applications peut être activée ou désactivée. L'accès à chacune d'elles peut être donné ou retiré par utilisateur. Exemple : les professeurs n'ont pas accès aux données de l'infirmerie.

b. Pronote :

« Toute la vie scolaire en un logiciel » telle est la formule choisie par l'éditeur pour présenter son logiciel, et pour une fois ce petit slogan se vérifie justement. Déjà utilisé par plusieurs collèges et lycées, ce programme est devenu en quelques années seulement une référence dans son domaine. Il permet au personnel de direction et aux enseignants de piloter leurs classes, établissements, et de communiquer simplement avec les familles des élèves. Les parents, quant à eux, peuvent suivre les notes de leurs enfants jour après jour, leur emploi du temps et demander un rendez-vous si besoin.

III. Critiques de l'existant

1. Limites d'Eduserv

Chaque établissement possède un système informatique qui stocke les informations (nom, prénom, notes, ...) des élèves inscrits. La solution personnalisée (Eduserv) permet l'impression de ceux-ci afin de les distribuer aux élèves, aux parents et aux professeurs. Ces derniers relèvent les absences dans leurs classes et les transmettent à l'administration qui les saisit dans le système. Les notes sont saisies dans le logiciel par les professeurs. Les informations sont diffusées aux parents sous forme de papier et souvent par l'intermédiaire des élèves.

En ajoutant l'absence de la communication entre les professeurs et les parents surtout que Les rendez-vous sont gérés par l'administration ce qui peut les retarder.

2. Limites des applications Web

Les spécificités techniques des applications constitueront également des facteurs de choix importants : applications libres (souvent gratuites), propriétaires, avec ou sans services d'assistance intégrés, tout en ligne sur serveurs distants, sur poste, sur serveur interne...

Presque toutes les applications existantes ont un coût qui peut être assez élevé, à l'exception de certaines applications libres qui néanmoins demandent généralement une formation (payante) avant d'être utilisée.

Les spécificités vous y trouverez certainement celle qui convient le mieux à la configuration de nos établissements.

IV. Solutions Proposées

1. Première solution

Continuer à améliorer la plateforme Eduserv en ligne en mettant l'accent sur le développement de la partie qui concerne la communication entre les professeurs et les parents en ajoutant les fonctionnalités de :

- ✓ gestion des rendez-vous
- ✓ gestion des punitions
- ✓ gestion des appréciations

✓ et en améliorant les autres fonctionnalités.

2. Deuxième solution

Le conseil de classe et la réunion parents-profs étaient auparavant les deux seuls moments de l'année où les parents avaient une visibilité sur le travail scolaire et les notes de leurs enfants, mais le monde de l'éducation, étant un monde de plus en plus high-tech, utilise également des applications mobiles.

Une application mobile remplace de ce fait le cahier de correspondance, trop souvent oublié ou perdu (parfois volontairement) par les élèves.

Une deuxième solution doit fournir un accès à distance aux notes, à l'emploi du temps et aux absences. De plus, les rendez-vous pourront être directement pris via l'Internet ainsi que les outils d'aide pour le travail à la maison seront mis à disposition des parents.

V. Solution retenue

Les technologies de l'Internet mobile connaissent actuellement un bouleversement radical : l'accès classique au web via des ordinateurs de bureau et portables n'est plus seulement complété, mais de plus en plus remplacé par l'usage de dispositifs mobiles tel que les smartphones.

Toutes les études récentes et les rapports sur les tendances à propos du développement d'Internet s'accordent à dire que l'Internet mobile est celui du futur. Le rapport Gartner intitulé « Utilisateur final prévisions pour 2010 » affirme que « d'ici 2013, dans le monde entier, le mobile aura supplanté le PC comme moyen d'accès le plus couramment utilisé au Web ». Le rapport de Morgan Stanley sur l'Internet Mobile si souvent cité prévoit également : « si l'on se base sur la rapidité du changement, je croyons qu'il y a de fortes chances pour que d'ici 5 ans plus d'utilisateurs se connectent à Internet via des outils mobiles plutôt qu'à des ordinateurs de bureau ».

En effet, l'utilisation des outils numériques tel que les smartphones dans la vie quotidienne est incontournable et apporte une aide précieuse dans des divers domaines comme l'apprentissage scolaire, d'où l'idée de développer une application mobile de services scolaires pédagogiques et administratifs.

Depuis quelques années, des systèmes d'exploitation mobiles sont développés. Voilà ci-dessous un datagramme qui dévoile le classement des systèmes d'exploitation en Tunisie de Mars 2014 à Mars 2015, d'après **Statcounter**⁶.

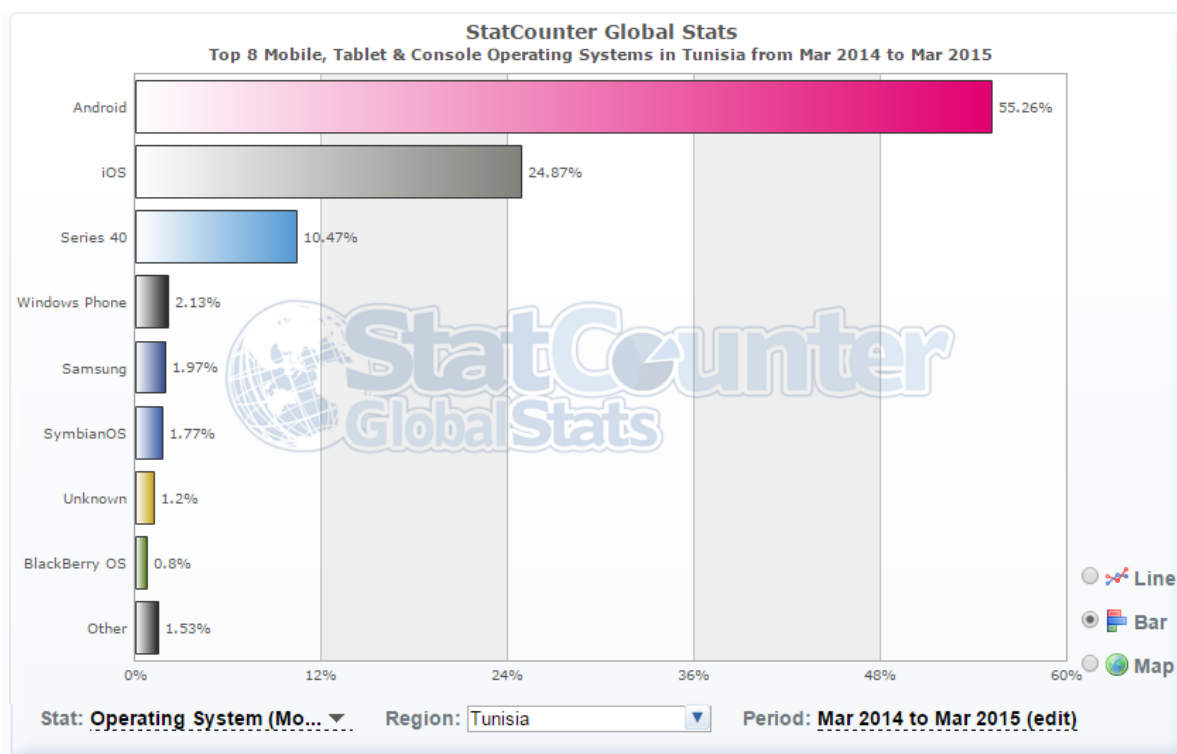


Figure 3: Classement des OS en Tunisie

Ces statistiques montrent que « Android » a décroché la plupart de marché tunisien avec un taux de 55,26%, et c'est pour cela que j'ai choisi de développer une application Android.

De ce fait une application Android va mieux répondre aux objectifs. Les parents, quant à eux, sont informés de l'absence de leurs enfants et peuvent ainsi mieux réagir et les suivre, surtout que toujours ils sont proches de leurs Smartphone et par la suite de l'application.

VI. Conclusion

Dans ce chapitre, j'ai commencé par étudier l'existant et ses limites, le critiquer avant de proposer deux solutions possibles pour faire mieux et présenter la solution retenue.

⁶ <http://gs.statcounter.com/#mobile+tablet+console-os-TN-monthly-201403-201503-map>

Chapitre 3 : Branche fonctionnelle

I.Introduction

Les chapitres précédents conduisent à construire une bonne vision du produit, capturer les besoins, identifier les acteurs et dégager les fonctionnalités principales ce qui permet d'exposer le diagramme de cas d'utilisation et la description textuelle de chaque cas.

II.Capture des besoins

Cette phase consiste à détailler l'ensemble des fonctionnalités que l'application doit fournir. Ces besoins seront présentés sous forme de besoins fonctionnels et non fonctionnels.

1. Besoins fonctionnels détaillés

Les besoins fonctionnels présentent les actions que le système doit assurer en répondant à une requête.

Ce système se place en tant qu'interface entre les parents et l'établissement. Ce logiciel a pour cible les parents souhaitant s'impliquer activement dans la scolarité de leur(s) élève(s). Pour les aider dans leurs démarches, l'application offrira un certain nombre de services via un espace personnel accessible depuis Internet.

Et maintenant, nous revenons à notre sujet de base, l'application offre un ensemble de services scolaires administratifs et pédagogiques à ses utilisateurs tels que :

- Gestion des notes : ce service permet la saisie et la consultation des notes.
- Gestion des absences : ce service permet l'édition de la feuille d'appel et la consultation des absences.
- Consultation de l'emploi de temps : Ce service permet la consultation de l'emploi de temps d'un professeur ou d'une classe.
- Gestion des actualités : ce service permet l'ajout et la consultation des actualités.
- Gestion des punitions : ce service permet l'ajout et la consultation des punitions.
- Gestion des appréciations : ce service permet à l'professeur d'envoyer une appréciation au parent d'un élève.

- Gestion du profil : ce service permet la consultation et la modification des informations personnelles.
- Gestion des rendez-vous : qui concerne la consultation et la demande des rendez-vous soit Parent-professeur ou parent-administration.

2. Besoins non fonctionnels détaillés

La simplicité et la lisibilité représentent les principaux besoins non fonctionnels que doivent fournir notre application mobile ainsi que d'autres contraintes :

- Contraintes ergonomiques : simplicité et convivialité des interfaces graphiques.
- Contraintes de sécurité : authentification, ressaisie du mot de passe lors d'un traitement dans la base de données.
- Contraintes de performance : accès facile, chargement rapide.
- Contrainte de fiabilité : sans ambiguïté.

III. Besoins techniques

- avoir un smartphone tournant sous le système d'exploitation Android de Google.
- Une connexion de haut débit (3G, 3G++, etc.).
- Une connexion WIFI/LIFI.

IV. Analyse des fonctionnalités

1. Identification des acteurs

Notre système identifie les acteurs principaux suivants :

Professeur	
Responsabilités :	Saisir les absences, les notes et les appréciations des élèves. Répondre aux demandes de rendez-vous avec les parents.
Niveau de Compétence :	Intermédiaire
Fréquence d'utilisation :	Courte utilisation quotidienne

Autorité :	Dans l'organigramme des écoles secondaires et préparatoires, les professeurs sont sous la responsabilité du directeur (responsable de l'établissement). Dans notre application cette autorité sera représentée par l'Administrateur qui sera lui-même sous la responsabilité du directeur.
Permissions :	Les privilèges des professeurs incluent l'accès aux données de leurs élèves. Ils peuvent uniquement modifier les notes des élèves qui suivent leur(s) cours. Ils peuvent aussi consulter leur emploi du temps mais sans la possibilité de le modifier. Ils n'ont pas d'autorité sur les autres utilisateurs du système.

Tableau 1: Identification de l'acteur "professeur"

Le personnel administratif	
Responsabilités :	Saisir et gérer les absences des élèves. Crée les groupes de cours et les descriptions. Consulter et répondre aux demandes des rendez-vous Modifier l'emploi de temps de manière temporaire
Niveau de Compétence :	Intermédiaire
Fréquence d'utilisation :	Utilisation quotidienne et constante.
Autorité :	Le responsable de l'établissement occupe le plus haut niveau d'autorité dans l'organisation mais dans notre application son compte ne possède pas les droits « Administrateur ». Il devra s'adresser à l'administrateur ou endosser les deux rôles pour effectuer les actions critiques. Les secrétaires ne possèdent pas d'autorité sur les professeurs.

Permissions :	Les secrétaires ont accès aux informations personnelles de tous les élèves et ils gèrent leur(s) absence(s) et leur emploi du temps. Cependant, ils ne peuvent modifier aucune des notes des élèves.
---------------	--

Tableau 2: Identification de l'acteur "personnel administratif"

L'administrateur (responsable de l'établissement)	
Responsabilités :	Saisir et gérer les comptes des utilisateurs. Modérer les messages entre parents et professeurs. Nettoyer le système en fin d'année scolaire pour supprimer les données qui ne sont plus utiles (absences, messages, travaux mis en ligne).
Niveau de Compétence :	Expert
Fréquence d'utilisation :	Fréquente
Autorité :	Le personnel technique ne dépend que du responsable de l'établissement. Cependant, il peut recevoir aussi des demandes des parents (message).
Permissions :	L'administrateur a les pleins pouvoirs sur le système et a donc le plus haut niveau de privilège.

Tableau 3: Identification de l'acteur "Administrateur"

Parent	
Responsabilités :	Consulter les notes, les absences et les emplois du temps de leur(s) enfant(s). Aider leur(s) enfant(s) dans le travail à la maison grâce à l'aide postée par les professeurs. Garder les informations personnelles à jour.

	Établir des rendez-vous au besoin.
Niveau de Compétence :	Novice
Fréquence d'utilisation :	Selon le besoin
Autorité :	Cependant, ils peuvent communiquer leurs problèmes au personnel de l'établissement.
Permissions :	Les privilèges qu'ils ont se limitent à consulter les données relatives à leurs enfants. Ils ne peuvent pas modifier les informations personnelles par le biais de notre système. Ils peuvent contacter les professeurs des leurs enfants.

Tableau 4: Identification de l'acteur "Parent"

Élève	
Responsabilités :	Consulter les notes, les absences et son emploi du temps.
Niveau de Compétence :	Novice
Fréquence d'utilisation :	Fréquente
Autorité :	Ils peuvent modifier leurs agendas.
Permissions :	Les privilèges qu'ils ont se limitent à consulter leurs données.

Tableau 5: Identification de l'acteur "Elève"

2. Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation permet de représenter un ensemble d'actions réalisées par un système, représenté par une boîte rectangulaire, produisant un résultat sur un acteur, appelé acteur principal, indépendamment de son fonctionnement interne. La figure suivante représente le diagramme de cas d'utilisation général de notre application :

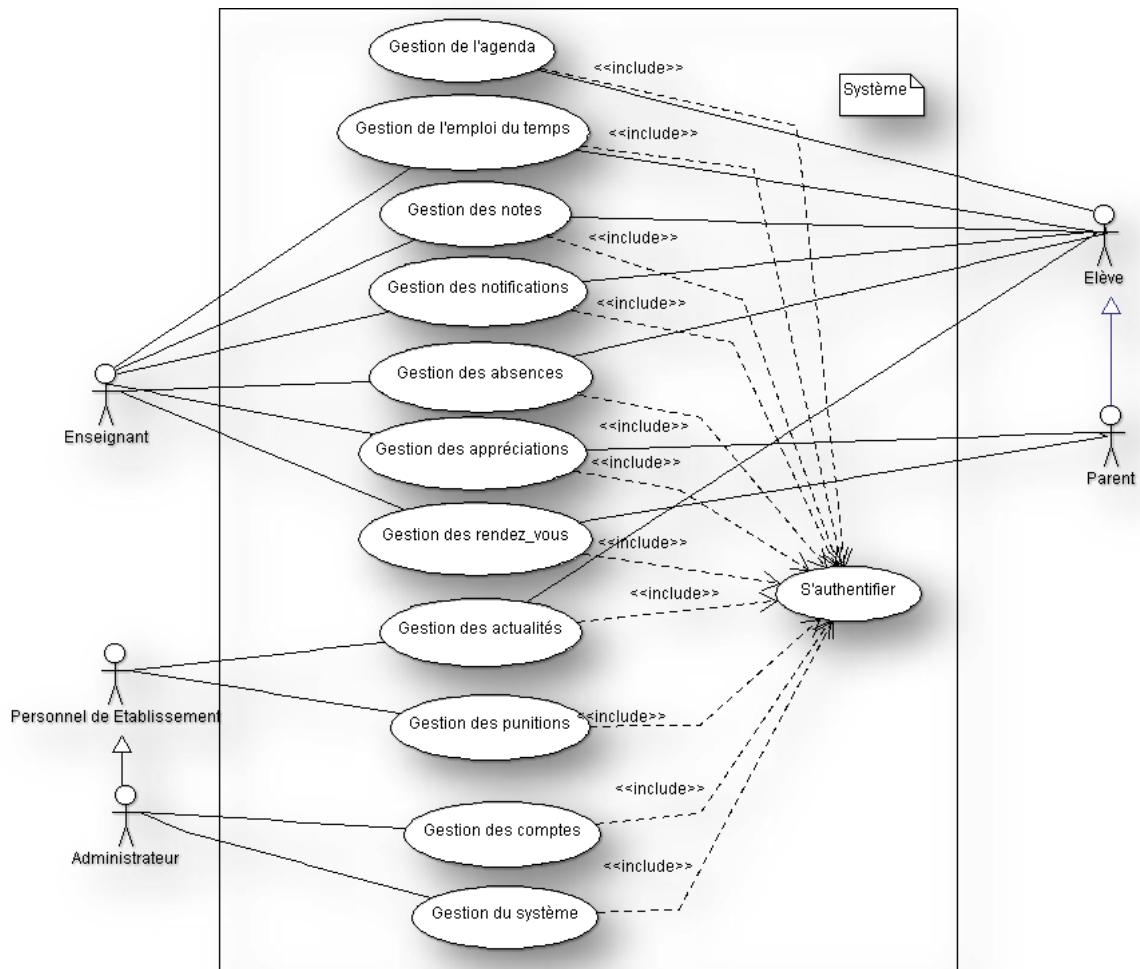


Figure 4 : Diagramme de cas d'utilisation

3. Description des cas d'utilisation

a. Gestion des comptes

Voici le diagramme d'utilisation du package « gestion des comptes » :

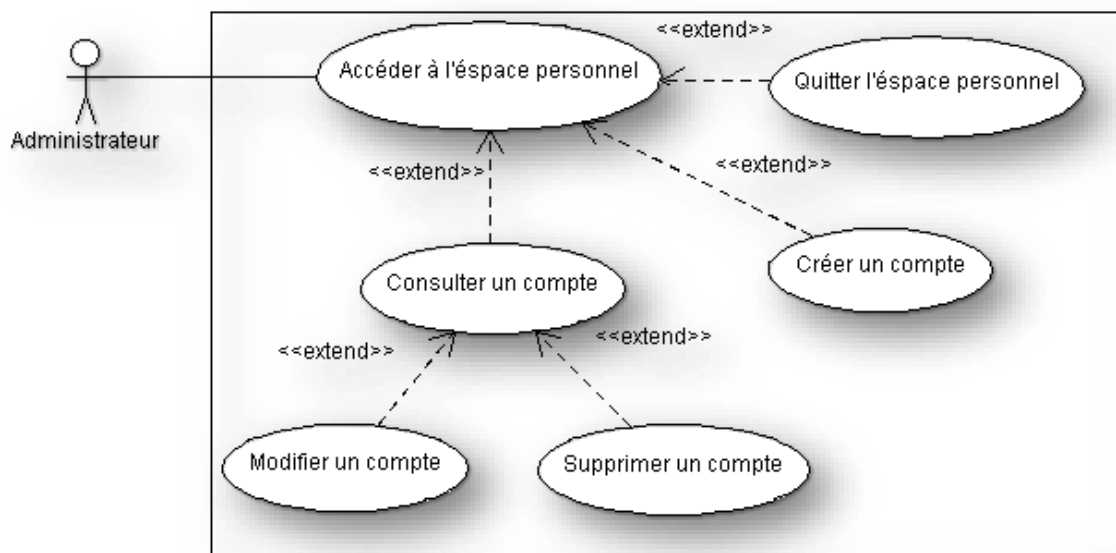


Figure 5: cas d'utilisation gestion des comptes

▪ Créer un compte

Nom : Créer un compte

Acteurs : Administrateur

Type : Primaire

Évènement déclencheur : L'administrateur veut créer un compte pour un nouvel utilisateur.

Description : L'administrateur veut créer un compte. Il choisit type d'utilisateur. Un login et un mot de passe sont générés automatiquement et sont affichés pour ce compte.

Dépendances : Un utilisateur doit être enregistré dans le système

Préconditions : L'administrateur doit être connecté dans le système

Scénario :

Acteurs	Actions	Système
Administrateur	1 Il choisit de créer un compte	2 Affichage de la création d'un compte Internet avec la liste des familles ne possédant pas de compte

	3 Il choisit un type dans la liste proposée	4 Génération d'un login et d'un mot de passe 5 Enregistrement du compte dans le système 6 Affichage du login et du mot de passe
--	---	---

Tableau 6: Scénario « Créer un compte »

Exception :

Aucune

Règles de terminaison :

- Le compte est créé
- [Consulter des comptes](#)

Nom : Consultation des comptes

Acteurs : Administrateur

Type : Primaire

Évènement déclencheur : L'administrateur veut consulter les comptes

Description : L'administrateur veut consulter la liste des comptes.

Dépendances : Aucune

Préconditions : L'administrateur doit être connecté dans le système

Scénario :

Acteurs	Actions	Système
Administrateur	1 Il choisit de consulter les comptes	2 Affichage d'un tableau avec les comptes (Les données affichées sont : le nom, prénom, type de compte,...)

Tableau 7: Scénario « Consulter un compte »

Exception :

Aucune

Règles de terminaison :

- La liste des comptes est affichée

- [Modifier un compte](#)

Nom : Modifier un compte

Acteurs : Administrateur

Type : Primaire

Évènement déclencheur : L'administrateur veut modifier un compte

Description : L'administrateur veut modifier un compte existant

Dépendances : Un compte doit être présent dans le système

Préconditions : L'administrateur doit être authentifié dans le système

Scénario :

Acteurs	Actions	Système
Administrateur	1 Il choisit de modifier un compte	
		2 Affichage de la liste des comptes (les données affichées sont le login, le nom du propriétaire et le type de compte)
	3 Il choisit un compte dans la liste	
	5 Il modifie les champs souhaités et valide	4 Affichage des informations reliées au compte
		6 Vérification des données
		7 Enregistrement des modifications

Tableau 8: Scénario « Modifier un compte »

Exception :

6. Si les données ne sont pas valides (ex : champ(s) vide(s)), retour au 4

Règles de terminaison :

- Le compte est modifié

b. Gestion de l'emploi du temps

Voici le diagramme d'utilisation du package « gestion de l'emploi du temps » :

- [Consulter l'emploi du temps](#)

Nom : Consulter l'emploi du temps de la session (internet)

Acteurs : Parent, Élève

Type : Primaire

Évènement déclencheur : Un parent désire consulter l'emploi du temps pour l'un de ses élèves.

Description : Le parent veut consulter l'emploi du temps de son élève. Il choisit la section emploi du temps. L'emploi du temps général pour l'élève concerné est affiché.

Dépendances : Un enfant doit être sélectionné

Scénario :

Acteurs	Actions	Système
Parent	1 Il sélectionne la section emploi du temps	2 Affichage de l'emploi du temps de l'élève concerné

Tableau 9: Scénario « Consulter Emploi du temps »

Exceptions :

Aucune

Règles de terminaison :

- L'emploi du temps de l'élève est affiché

- [Consulter l'emploi du temps \(professeur\)](#)

Nom : Consulter l'emploi du temps (professeur)

Acteurs : Professeur

Type : Primaire

Évènement déclencheur : Un professeur souhaite regarder son emploi du temps

Description : Le professeur souhaite regarder son emploi du temps. Il va dans la section emploi du temps et celui-ci lui est affiché

Dépendances : Aucune

Préconditions : Être authentifié dans le système

Scénario :

Acteurs	Actions	Système
Professeur	1 Il sélectionne la section de consultation d'un emploi du temps	2 Affichage de l'emploi du temps

Tableau 10: Scénario « Consulter Emploi du temps, Professeur »

Exception :

Aucune

Règles de terminaison :

- L'emploi du temps du professeur est affiché

c. Gestion des notes

Voici le diagramme d'utilisation du package « gestion des notes » :

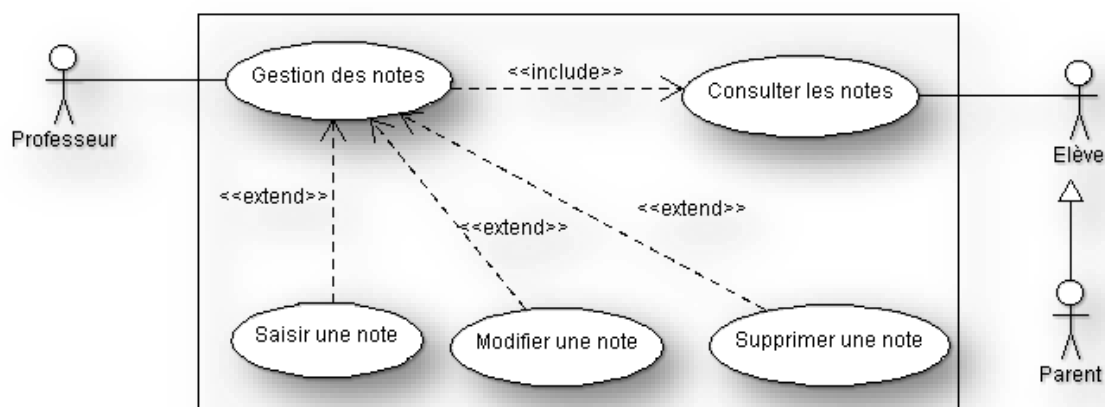


Figure 6: Cas d'utilisation Gestion des notes

▪ Saisir une note

Nom : Saisir une note

Acteurs : Professeur

Type : Primaire

Évènement déclencheur : Un professeur souhaite saisir une note pour un élève concernant un cours qu'il donne.

Description : Un professeur veut saisir une note concernant un élève pour un devoir noté. Il choisit tout d'abord le groupe concerné ainsi que le cours. Ensuite il choisit l'élève désiré et saisit les informations concernant la note.

Dépendances : Un groupe cours et un élève doivent exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Professeur	1 Il choisit de saisir une note 3 Il sélectionne le groupe souhaité 5 Il choisit le sigle du cours 7 Il saisit la note.	2 Affichage de la liste des groupes suivants ce cours 4 Affichage de la page pour saisir des notes. La liste des cours que le professeur enseigne est affichée 6 Affichage de la liste des élèves et la liste des devoirs notés mis en ligne 8 Vérification des données 9 Enregistrement des données

Tableau 11: Scénario « saisir une note »

Exception :

8 Si le champ note est vide, retour au 7

Règles de terminaison :

- La note est ajoutée

- [Consulter les notes](#)

Nom : Consulter les notes

Acteurs : Parent, élève, professeur.

Type : Primaire

Évènement déclencheur : Un élève ou parent souhaite consulter les notes d'un de ces enfants.

Description : Un parent veut voir les notes d'un de ses enfants pour la session en cours. La liste des cours qui suit son enfant est affichée, il choisit un cours parmi cette liste et les notes sont affichées.

Dépendances : Un enfant doit être sélectionné

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Parent, élève ou professeur	1 Il choisit de consulter les notes de son enfant 3 Il choisit le sigle du cours	2 Affichage de la page pour consulter les notes. La liste des cours suivis par l'enfant est affichée 4 Affichage de la liste des notes avec leurs informations relatives

Tableau 12: Scénario « Consulter les notes »

Exception :

Aucune

Règles de terminaison :

- Les notes sont affichées

- [Modifier une note](#)

Nom : Modifier une note

Acteurs : Professeur

Type : Primaire

Évènement déclencheur : Un professeur souhaite modifier une note pour un élève concernant un cours qu'il donne.

Description : Un professeur veut modifier une note concernant un élève pour un devoir noté. Il choisit tout d'abord le groupe concerné ainsi que le cours. Ensuite il choisit l'élève désiré et modifie les informations concernant la note.

Dépendances : Un groupe cours et un élève doivent exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Professeur	1 Il choisit de modifier une note	2 Affichage de la liste des groupes suivants ce cours
	3 Il sélectionne le groupe souhaité	4 Affichage de la page pour saisir des notes. La liste des cours que le professeur enseigne est affichée
	5 Il choisit le sigle du cours	6 Affichage de la liste des élèves et la liste des devoirs notés mis en ligne
	7 Il saisit la note.	8 Vérification des données 9 Enregistrement des données

Tableau 13: Scénario « Modifier une note »

Exception :

8 Si le champ note est vide, retour au 7

Règles de terminaison :

- La note est modifiée

d. Gestion des absences

▪ [Consulter Absences](#)

Nom : Consulter les Absences

Acteurs : Parent, élève, professeur.

Type : Primaire

Évènement déclencheur : Un élève ou parent souhaite consulter les absences d'un de ses enfants.

Description : Un parent veut voir les absences d'un de ses enfants pour la session en cours. La liste des cours qui suit son enfant est affichée, il choisit un cours parmi cette liste et les absences sont affichées.

Dépendances : Un enfant doit être sélectionné

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Parent, élève ou professeur	1 Il choisit de consulter les absences de l'élève	2 Affichage de la liste des absences avec leurs informations relatives

Tableau 14: Scénario « Consulter Absences »

Exception :

Aucune

Règles de terminaison :

- Les absences sont affichées

▪ [Ajouter une absence](#)

Nom : Saisir une absence

Acteurs : Professeur

Type : Primaire

Évènement déclencheur : Un professeur souhaite saisir une absence pour un élève concernant un cours qu'il donne.

Description : Un professeur veut ajouter une absence concernant un élève pour un devoir noté. Il choisit tout d'abord le groupe concerné ainsi que le cours. Ensuite il choisit l'élève désiré et saisit les informations concernant l'absence. **Dépendances :** Un groupe cours et un élève doivent exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Professeur	1 Il choisit d'ajouter une absence	2 Affichage de la liste des groupes suivants ce cours
	3 Il sélectionne le groupe souhaité	4 Affichage de la page pour saisir des absences. La liste des cours que le professeur enseigne est affichée
	5 Il choisit le sigle du cours	6 Affichage de la liste des élèves et la liste des devoirs notés mis en ligne
	7 Il ajoute l'absence.	8 Vérification des données 9 Enregistrement des données

Tableau 15: Scénario « Consulter Absences »

Exception :

8 Si le champ absence est vide, retour au 7

Règles de terminaison :

- L'absence est ajoutée

e. Gestion de l'agenda

Chaque élève peut créer ses mémos personnels dans son agenda. Il peut créer, modifier, supprimer ou consulter son agenda.

Voici le diagramme d'utilisation du package « gestion de l'agenda » :

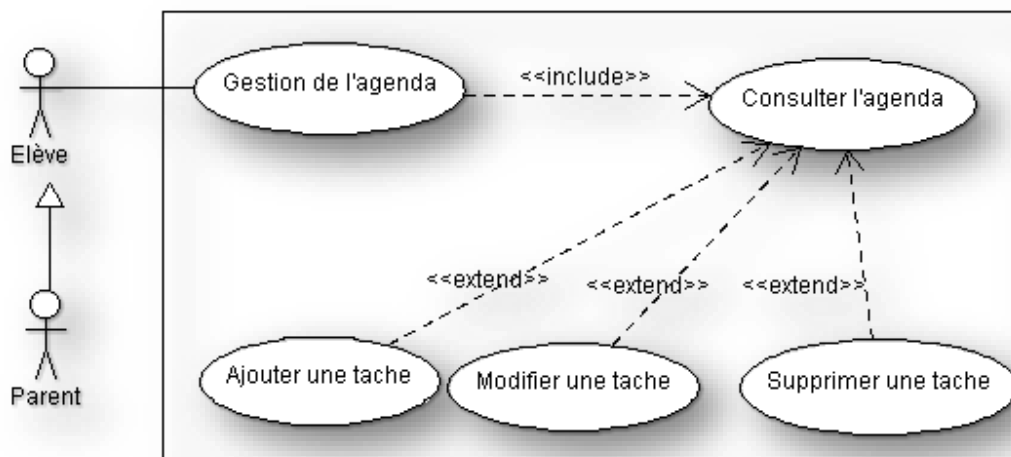


Figure 7: Cas d'utilisation gestion de l'agenda

▪ Ajouter une tâche

Nom : Ajouter une tâche

Acteurs : Élève, parent

Type : Primaire

Évènement déclencheur : Un élève souhaite ajouter une tâche à faire dans son agenda (exemple travail à faire, devoir de maison, etc.).

Description : Un élève veut ajouter une tâche concernant un devoir, un travail à faire ou autre. Il choisit tout d'abord le groupe concerné ainsi que le cours. Ensuite il choisit l'élève désiré et saisit les informations concernant la tâche à enregistrer.

Dépendances : Un groupe cours et un élève doivent exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Professeur	1 Il choisit de saisir une tâche	

	3 Il sélectionne le groupe souhaité 5 Il saisit le nom et les autres informations de la tâche à faire.	2 Affichage de la liste des groupes suivants ce cours. 4 Affichage de la page pour saisir le travail à faire.
--	--	---

Tableau 16: Scénario « Ajouter une tâche »

Règles de terminaison :

- La tâche est ajoutée

- [Consulter l'agenda](#)

Nom : Consulter l'agenda

Acteurs : Parent, élève.

Type : Primaire

Évènement déclencheur : Un parent ou un élève souhaite consulter son agenda.

Description : Un élève veut voir les taches à faire qui sont déjà ajoutées précédemment. La liste des tâches existantes est affichée, il choisit une parmi cette liste et ses informations sont affichées.

Dépendances : Une tache doit être existante

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Parent ou élève	1 Il choisit de consulter son agenda 3 Il choisit la tache désirée	2 La liste des taches ajoutées par l'élève est affichée 4 Affichage des informations relatives

Tableau 17: Scénario « Consulter Agenda»

Exception :

Aucune

Règles de terminaison :

- Les informations sont affichées
- [Modifier une tâche](#)

Nom : Modifier une tâche

Acteurs : Élève

Type : Primaire

Évènement déclencheur : Un élève souhaite modifier une tâche existante.

Description : Un élève veut modifier une tâche ajoutée. Il choisit tout d'abord la tâche concernée parmi la liste existante. Ensuite il modifie les informations concernant la tâche.

Dépendances : Une tâche doit exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Élève	1 Il choisit de consulter l'agenda	
		2 Affichage de la liste des tâches existantes.
	3 Il choisit la tâche désirée	
		4 Affichage des informations de la tâche concernée.
	5 Il modifie les informations.	
		6 Enregistrement des données

Tableau 18: Scénario « Modifier une tâche »

Exception :

Aucune

Règles de terminaison :

- La tâche est modifiée

▪ Supprimer une tâche

Nom : Supprimer une tâche

Acteurs : Élève

Type : Primaire

Évènement déclencheur : Un élève souhaite supprimer une tâche existante.

Description : Un élève veut supprimer une tâche existante. Il choisit tout d'abord la tâche concernée parmi la liste existante. Ensuite il la supprime.

Dépendances : Une tâche doit exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Élève	1 Il choisit de consulter l'agenda	
		2 Affichage de la liste des tâches existantes.
	3 Il choisit une tâche et la supprime	
		4 Enregistrement

Tableau 19: Scénario « Supprimer une tâche »

Exception :

Aucune

Règles de terminaison :

- La tâche est supprimée

f. Gestion des notifications

Voici le diagramme d'utilisation du package « gestion des notifications» :

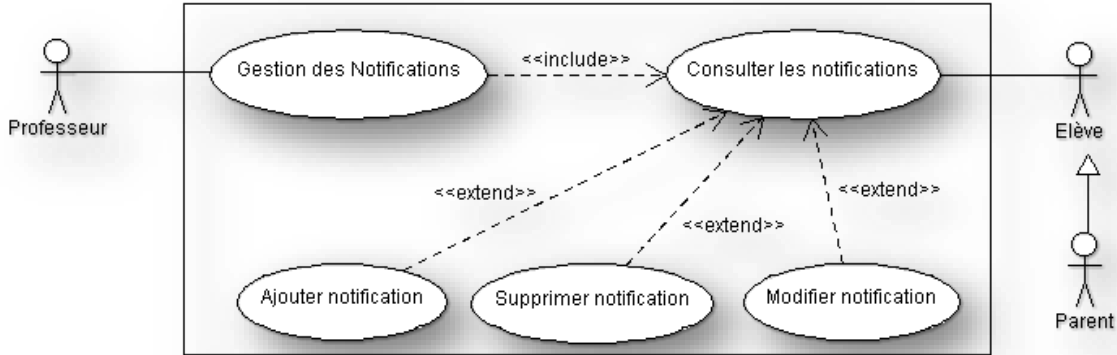


Figure 8: Cas d'utilisation gestion des notifications

- Ajouter notification

Nom : Ajouter notification

Acteurs : Professeur

Type : Primaire

Évènement déclencheur : Un professeur souhaite ajouter une notification pour un groupe de cours (Un classe).

Description : Un professeur veut saisir une notification pour une de ses classes. Il choisit tout d'abord le groupe concerné ainsi que le cours. Ensuite il saisit les informations concernant la notification.

Dépendances : Un groupe cours doit exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Professeur	1 Il choisit de saisir une notification 3 Il sélectionne le groupe souhaité	2 Affichage de la liste des groupes qui suivent le cours du professeur.

	5 Il saisit sa notification.	4 Affichage de la page pour saisir la notification.
		6 Enregistrement des données

Tableau 20: Scénario « Ajouter une notification »

Règles de terminaison :

- La notification est ajoutée
- [Consulter les notifications](#)

Nom : Consulter les notifications

Acteurs : Parent, élève, professeur.

Type : Primaire

Évènement déclencheur : Un élève, parent ou professeur souhaite consulter les notifications.

Description : Un parent ou un élève veut voir les notifications qui concernent son groupe de cours. La liste des notifications est affichée.

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Parent, élève ou professeur	1 Il choisit de consulter les notifications qui concernent la classe à qui appartient l'élève	2 Affichage de la page pour consulter les notifications avec leurs informations relatives

Tableau 21: Scénario « Consulter les notifications »

Exception :

Aucune

Règles de terminaison :

- Les notifications sont affichées

▪ Modifier une notification

Nom : Modifier une notification

Acteurs : Professeur

Type : Primaire

Évènement déclencheur : Un professeur souhaite modifier une notification pour une classe.

Description : Un professeur veut modifier une notification concernant une classe. Il choisit tout d'abord la notification concernée. Ensuite il modifie les informations concernant la notification.

Dépendances : Une notification doit exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Professeur	1 Il choisit de modifier une notification	2 Affichage de la liste des notifications ajoutées par le professeur.
	3 Il choisit laquelle veut modifier.	4 Affichage de la notification concernée
	5 Il modifie la notification.	6 Vérification des données 7 Enregistrement des données

Tableau 22: Scénario « Modifier une notification »

Exception :

Aucune

Règles de terminaison :

- La notification est modifiée

▪ Supprimer une notification

Nom : Supprimer une notification

Acteurs : Professeur

Type : Primaire

Évènement déclencheur : Un professeur souhaite supprimer une notification pour une classe.

Description : Un professeur veut supprimer une notification concernant une classe. Il choisit tout d'abord la notification concernée. Ensuite il supprime la notification.

Dépendances : Une notification doit exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Professeur	1 Il choisit de supprimer une notification	2 Affichage de la liste des notifications ajoutées par le professeur.
	3 Il choisit laquelle veut supprimer.	4 Affichage de la notification concernée
	5 Il supprime la notification.	7 Enregistrement des données

Tableau 23: Scénario « Supprimer une notification »

Exception :

Aucune

Règles de terminaison :

- La notification est supprimée

g. Gestion des appréciations

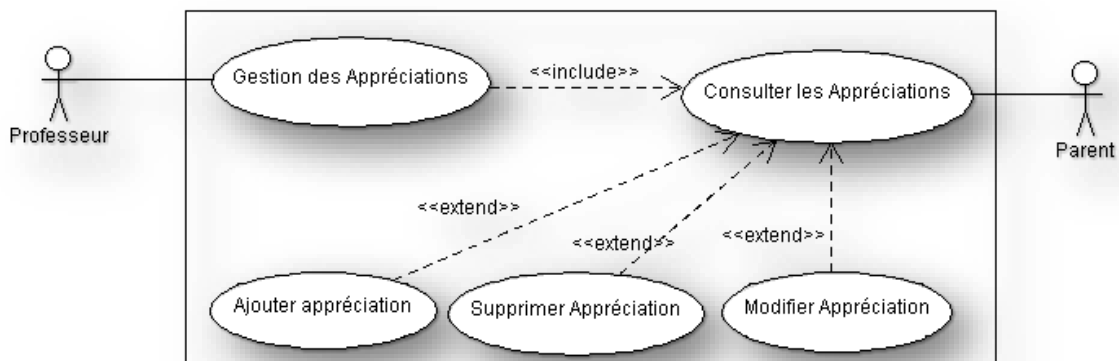


Figure 9: Cas d'utilisation gestion des appréciations

▪ Ajouter appréciation

Nom : Ajouter appréciation

Acteurs : Professeur

Type : Primaire

Évènement déclencheur : Un professeur souhaite ajouter une appréciation pour un groupe de cours (Un classe).

Description : Un professeur veut saisir une appréciation pour un de ses élèves. Il choisit tout d'abord le groupe concerné ainsi que le cours. Ensuite choisit l'élève concerné et il saisit les informations concernant l'appréciation.

Dépendances : Un groupe cours doit exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Professeur	1 Il choisit de saisir une appréciation	2 Affichage de la liste des groupes qui suivent le cours du professeur.
	3 Il sélectionne le groupe souhaité	

	5 Il choisit l'élève concerné et il saisit l'appréciation.	4 Affichage de liste des élèves du groupe choisi. 6 Enregistrement des données
--	--	---

Tableau 24: Scénario « Ajouter une appréciation »

Règles de terminaison :

- L'appréciation est ajoutée
- [Consulter les appréciations](#)

Nom : Consulter les appréciations**Acteurs :** Parent, élève, professeur.**Type :** Primaire**Évènement déclencheur :** Un élève, parent ou professeur souhaite consulter les appréciations.**Description :** Un parent ou un élève veut voir les appréciations qui les concernent. La liste des appréciations est affichée.**Préconditions :** Être authentifié dans le système.**Scénario :**

Acteurs	Actions	Système
Parent, élève ou professeur	1 Il choisit de consulter les appréciations qui concernent l'élève	2 Affichage de la page pour consulter les appréciations avec leurs informations relatives

Tableau 25: Scénario « Consulter une appréciation »

Exception :

Aucune

Règles de terminaison :

- Les appréciations sont affichées

▪ Modifier une appréciation

Nom : Modifier une appréciation

Acteurs : Professeur

Type : Primaire

Évènement déclencheur : Un professeur souhaite modifier une appréciation.

Description : Un professeur veut modifier une appréciation concernant un élève. Il choisit tout d'abord l'appréciation concernée. Ensuite il modifie les informations concernant l'appréciation.

Dépendances : Une appréciation doit exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Professeur	1 Il choisit de modifier une appréciation	2 Affichage de la liste des appréciations ajoutées par le professeur.
	3 Il choisit laquelle veut modifier.	4 Affichage de l'appréciation concernée
	5 Il modifie l'appréciation.	6 Vérification des données 7 Enregistrement des données

Tableau 26: Scénario « Modifier une appréciation »

Exception :

Aucune

Règles de terminaison :

- L'appréciation est modifiée

▪ Supprimer une appréciation

Nom : Supprimer une appréciation

Acteurs : Professeur

Type : Primaire

Évènement déclencheur : Un professeur souhaite supprimer une appréciation pour un élève.

Description : Un professeur veut supprimer une appréciation concernant une classe. Il choisit tout d'abord la notification concernée. Ensuite il supprime l'appréciation.

Dépendances : Une appréciation doit exister

Préconditions : Être authentifié dans le système.

Scénario :

Acteurs	Actions	Système
Professeur	1 Il choisit de supprimer une appréciation	2 Affichage de la liste des appréciations ajoutées par le professeur.
	3 Il choisit laquelle veut supprimer.	4 Affichage de l'appréciation concernée
	5 Il supprime l'appréciation.	7 Enregistrement des données

Tableau 27: Scénario « Supprimer une appréciation »

Exception :

Aucune

Règles de terminaison :

- La notification est supprimée

h. Gestion des rendez-vous

Voici le diagramme d'utilisation du package « gestion des rendez-vous » :

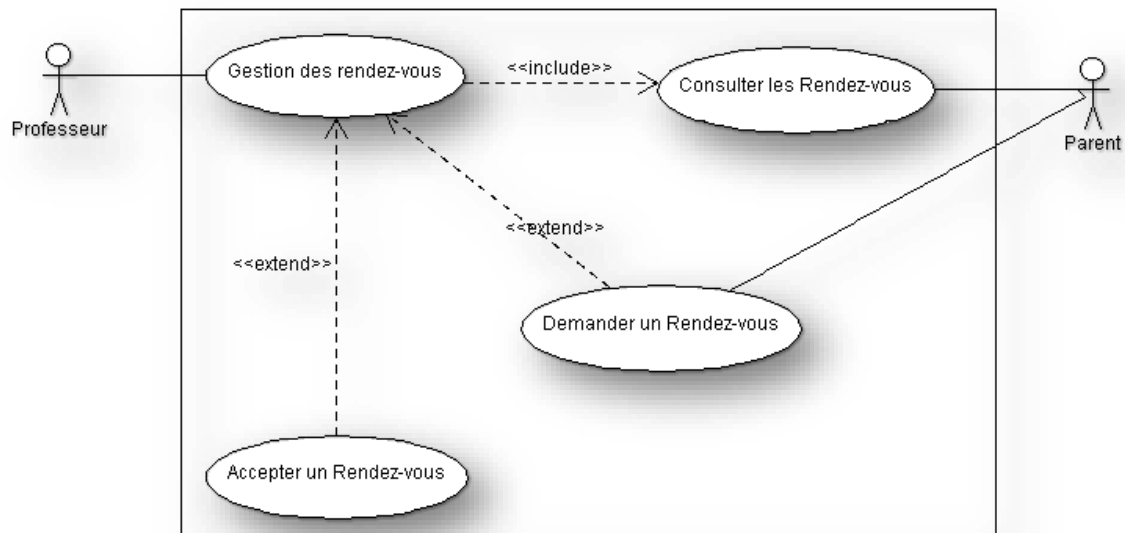


Figure 10: Cas d'utilisation gestion des rendez-vous

▪ Demander un rendez-vous

Nom : Faire une demande de rendez-vous

Acteurs : Parent

Type : Primaire

Évènement déclencheur : Un parent désire rencontrer un des professeurs de son enfant.

Description : Un parent fait une demande de rendez-vous auprès d'un des professeurs d'un de ses enfants. Il doit remplir le formulaire demandé.

Dépendances : Un élève doit être sélectionné

Préconditions : Être connecté au système.

Scénario :

Acteurs	Actions	Système
Parent	1 Il choisit la section des rendez-vous. 2 Il opte pour faire une demande	3 Affichage de la liste des professeurs qui enseigne à l'enfant du parent

	4 Il sélectionne un des professeurs de la liste.	5 Affichage du formulaire de demande de rendez-vous.
	6 Il remplit la demande et la soumet.	7 Une demande de rendez-vous est créée et ajoutée à la collection des rendez-vous du professeur.

Tableau 28: Scénario « Demander un rendez-vous »

Règles de terminaison :

Une demande de rendez-vous est créé et ajouter à la collection des rendez-vous du professeur.

- [Consulter les rendez-vous](#)

Nom : Consulter les rendez-vous

Acteurs : Parent, professeur

Type : Primaire

Évènement déclencheur : Un des acteurs décide de visionner la liste de ses rendez-vous.

Description : Le professeur ou le parent choisit la section des rendez-vous, une liste de ses rendez-vous est alors affichée.

Dépendances : Aucune

Préconditions : Être connecté au système

Scénario :

Acteurs	Actions	Système
Usager	1 Il sélectionne la section des rendez-vous.	
		2 Affichage de la liste des demandes de rendez-vous avec leur état et la date (passés et à venir)
Usager	3 Il consulte ses rendez-vous	



--	--	--

Tableau 29: Scénario « Consulter les rendez-vous »

Exception :

Aucune

Règles de terminaison :

La liste des rendez-vous est affichée.

- [Consulter un emploi du temps](#)

Nom : Consulter un emploi du temps

Acteurs : Tous

Type : Primaire

Évènement déclencheur : Un utilisateur désire consulter l'emploi du temps pour la session d'un élève.

Description : L'utilisateur veut consulter l'emploi du temps d'un élève. Il choisit la section emploi du temps. L'emploi du temps général pour la session en cours est affiché.

Dépendances : Un élève doit être sélectionné

Préconditions : Être connecté au système.

Scénario :

Acteurs	Actions	Système
Parent	1 Il sélectionne la section emploi du temps 3 Il choisit d'afficher l'emploi du temps de la session	2 Affichage de la page 4 Affichage de l'emploi du temps de la session pour l'élève sélectionné

Tableau 30: Scénario « Consulter Emploi du temps »

Exceptions :

Aucune

Règles de terminaison :

- L'emploi du temps de l'élève est affiché

i. Gestion du système

▪ Purger le système

Nom : Purger le système

Acteurs : Administrateur

Type : Primaire

Évènement déclencheur : L'administrateur souhaite vider le système

Description : L'administrateur veut purger le système. Il saisit une date et toutes les données antérieures à cette date sont supprimées.

Dépendances : Aucune

Préconditions : Être authentifié dans le système

Scénario :

Acteurs	Actions	Système
Administrateur	1 Il choisit de supprimer des données	2 Affichage de la page pour faire une purge du système
Administrateur	3 Il saisit une date	4 Vérification de la date 5 Demande de confirmation
Administrateur	6 Il confirme la suppression	7 Suppression des données (sauf Bulletin, descriptif de cours, les comptes et les personnes de l'application)

Tableau 31: Scénario « Purger système »

Exceptions :

- 4 La date est incorrecte, retour au 3

Règles de terminaison :

Toutes les données antérieures à cette date sont supprimées (sauf Bulletin, descriptif de cours, les comptes et les personnes de l'application).

V.Conclusion

Durant ce chapitre, j'ai commencé présenter les différents besoins attendus de notre application. Ensuite, j'ai identifié les acteurs de l'application, les messages émis et reçus par le système par identifier les cas d'utilisations. Puis la description du diagramme de cas d'utilisation global et celle textuelle du chaque cas. La partie de conception des données sera l'objectif du chapitre suivant.

Chapitre 4 : Branche technique

I.Introduction

J'entame ce chapitre par présenter l'architecture de notre système. Ensuite, je passe à la deuxième partie la présentation du diagramme de déploiement et des diagrammes de séquences détaillés.

II.Architecture

Opérer une connexion directe à la base de données depuis le téléphone Android n'est pas conseillé d'un point de vue architecture logicielle. Il est en effet préférable de passer par un middleware. Cette couche serveur intermédiaire sera la seule habilitée à se connecter à la base de données, ce qui est plus sécurisé. Donc La méthode la plus répandue pour se connecter à une base de données MySQL à distance à partir d'un appareil Android, est de mettre une sorte de service dans le milieu. MySQL est habituellement utilisé avec PHP, donc le plus évident est d'écrire des scripts PHP pour gérer la base de données et exécuter ces scripts en utilisant le protocole HTTP. J'ai codé les données dans le format JSON, afin de communiquer les données entre PHP et Android, en exploitant les options facile à utiliser construit dans les fonctions JSON dans les deux langages.

En fait, si je parle de l'architecture 3-tiers de point de vue technologie, le client est la plateforme Android, le serveur web est le PHP et le serveur de bases de données est le MySQL.

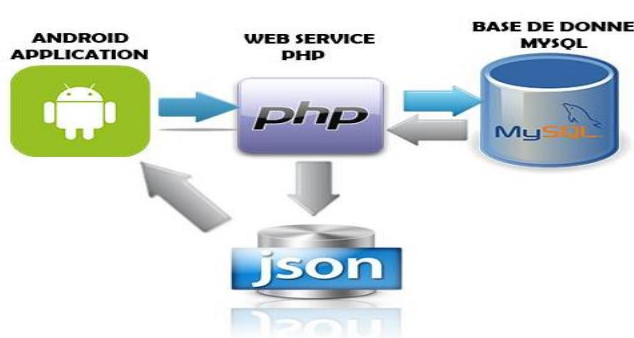


Figure 11: Architecture de notre application

Lorsque notre application Android s'exécute, elle va être connectée au serveur Web qui va récupérer les données depuis la base de données MySQL en utilisant les services web. Ensuite les données seront encodées au format JSON et envoyées au

système Android. L'application va obtenir ces données codées. Elle les analysera et les affichera sur le mobile.

En fait notre projet est divisé en deux parties à savoir l'écriture des scripts PHP, et le développement de l'application Android.

La partie serveur est composée de deux serveurs distants : le serveur web et le serveur de base données. Le serveur Web utilisé est le serveur Apache, il est le serveur le plus répandu sur internet pour déployer les scripts PHP. Le serveur de bases de données utilisé est le serveur MySQL.

III. Diagramme de déploiement :

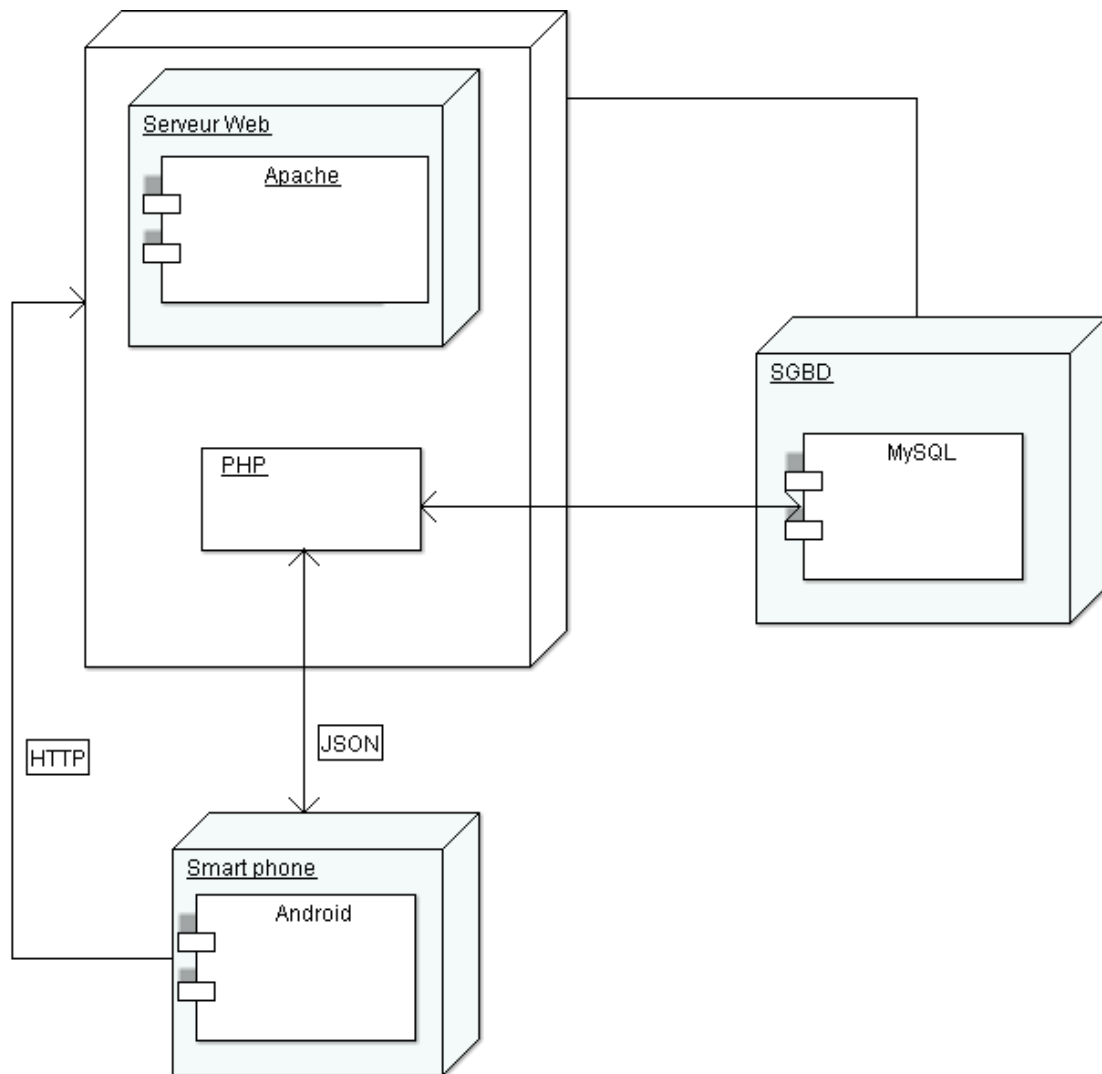


Figure 12: Diagramme de déploiement

IV. Diagrammes des séquences

Le diagramme de séquence permet de représenter la succession chronologique des opérations réalisées par un acteur. Il indique les objets qui seront manipulés par l'acteur et les opérations. Il représente les messages échangés entre les objets dans un ordre chronologique. Il donne une notion temporelle aux messages. Je l'utilise au niveau de la branche fonctionnelle de notre processus pour représenter l'opération de publication d'annonce par un utilisateur du système.

1. Diagramme de séquence détaillé du cas « s'authentifier »

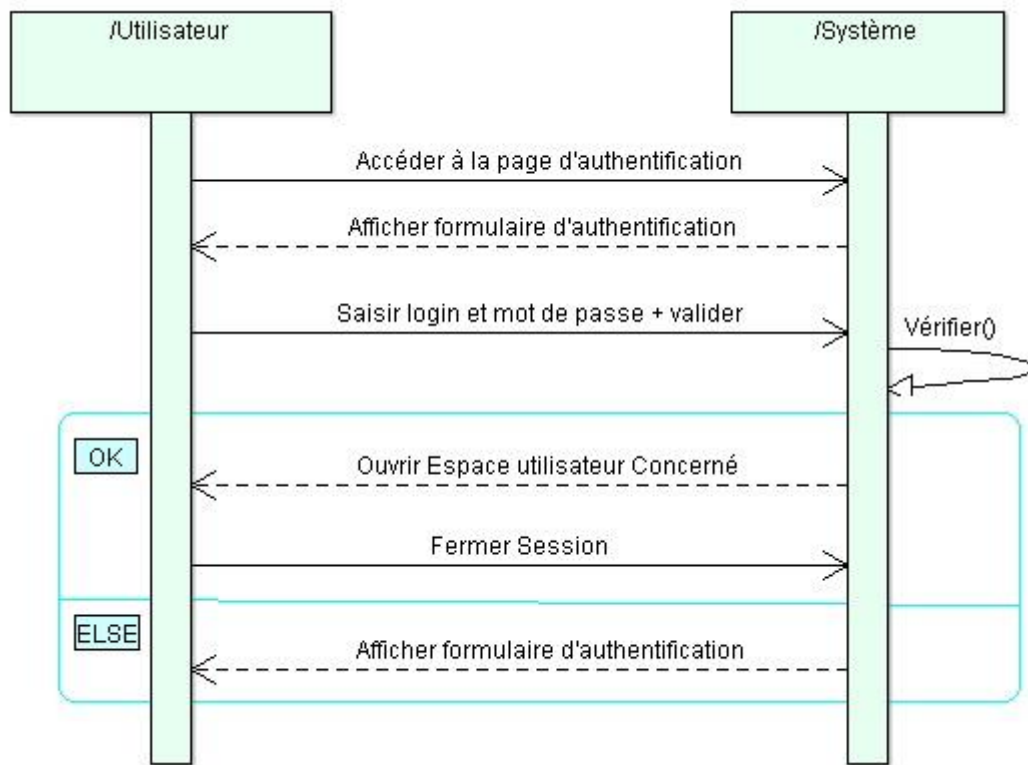


Figure 13: Diagramme de séquence détaillé du cas « s'authentifier »

L'utilisateur (élève, parent, enseignant et personnel de l'école) choisit l'espace concerné puis saisit les données d'authentification. Le système va vérifier ses droits d'accès dans la base de données et lui répondre par la suite soit par passage à son espace soit par un message d'erreur.

2. Diagramme de séquence détaillé du cas « Gestion des Rendez-vous»

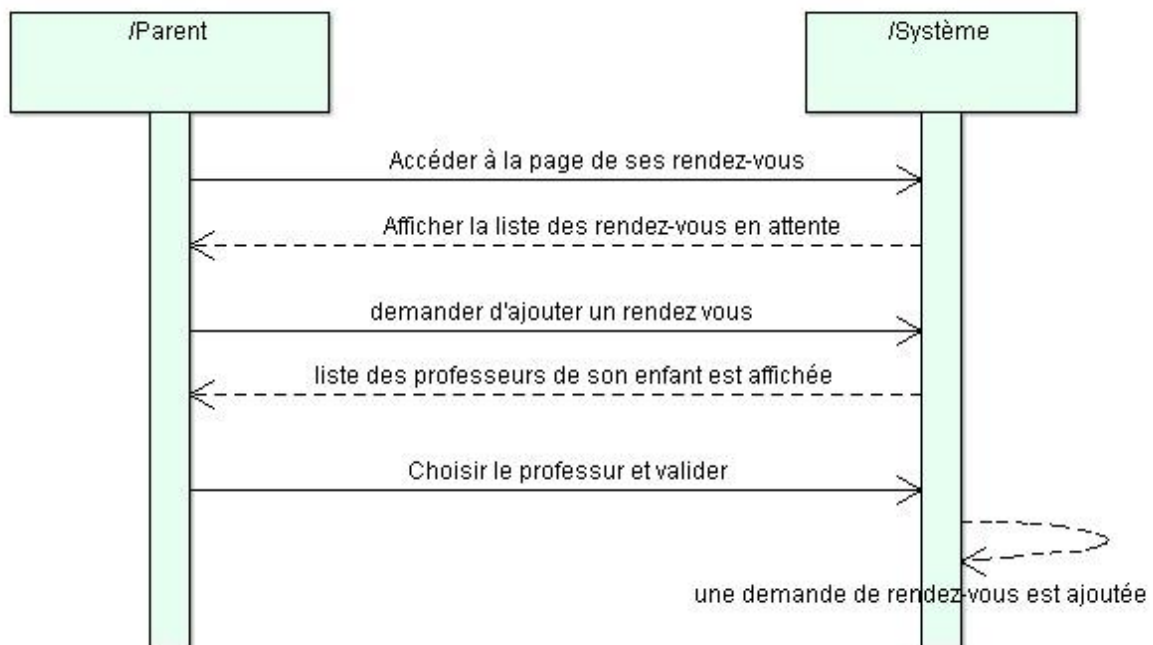


Figure 14: Diagramme de séquence détaillé du cas « Gestion des Rendez-vous»

Chaque parent peut demander sa liste des rendez-vous, et ensuite peut demander au système d'ajouter une demande de rendez-vous. Le système lui répondre par l'affichage de la liste des enseignants de son enfant. Le parent sélectionne un parmi eux et valide pour permettre au système d'ajouter cette demande à la base des données.

3. Diagramme de séquence détaillé du cas « Consulter les notes»

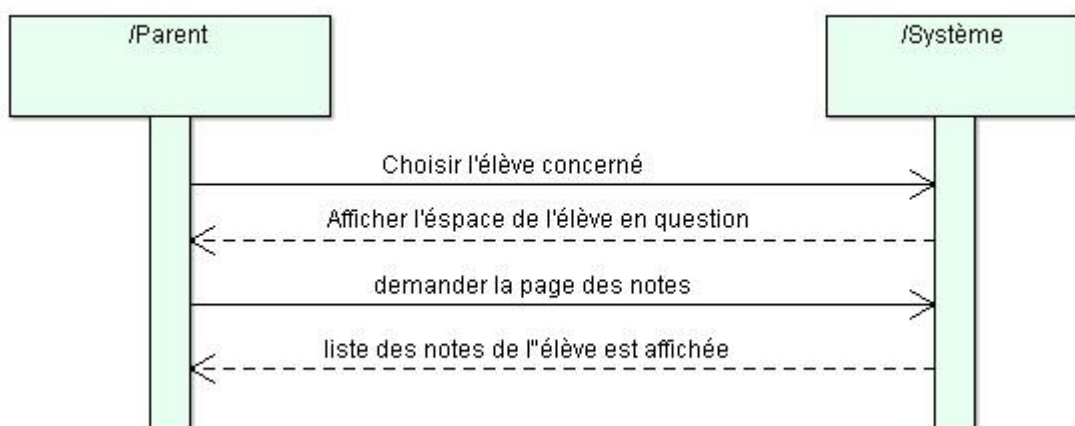


Figure 15: Diagramme de séquence détaillé du cas « Consulter les notes»

Un parent peut avoir plus qu'un enfant dans le même établissement et pour cela le système affiche la liste de ses enfants inscrits, le parent choisit l'élève voulu et puis demande la page des notes. Le système répond la liste des notes saisis par ses professeurs

V.Diagramme de classes

Le diagramme de classes est le point central dans le développement orienté objet. Coté analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. Coté conception, le diagramme de classes représente la structure d'un code orienté objet ou, à un niveau de détail plus important, les modules du langage de développement.

Une classe est composée des attributs et opérations, et admet des associations et multiplicité.

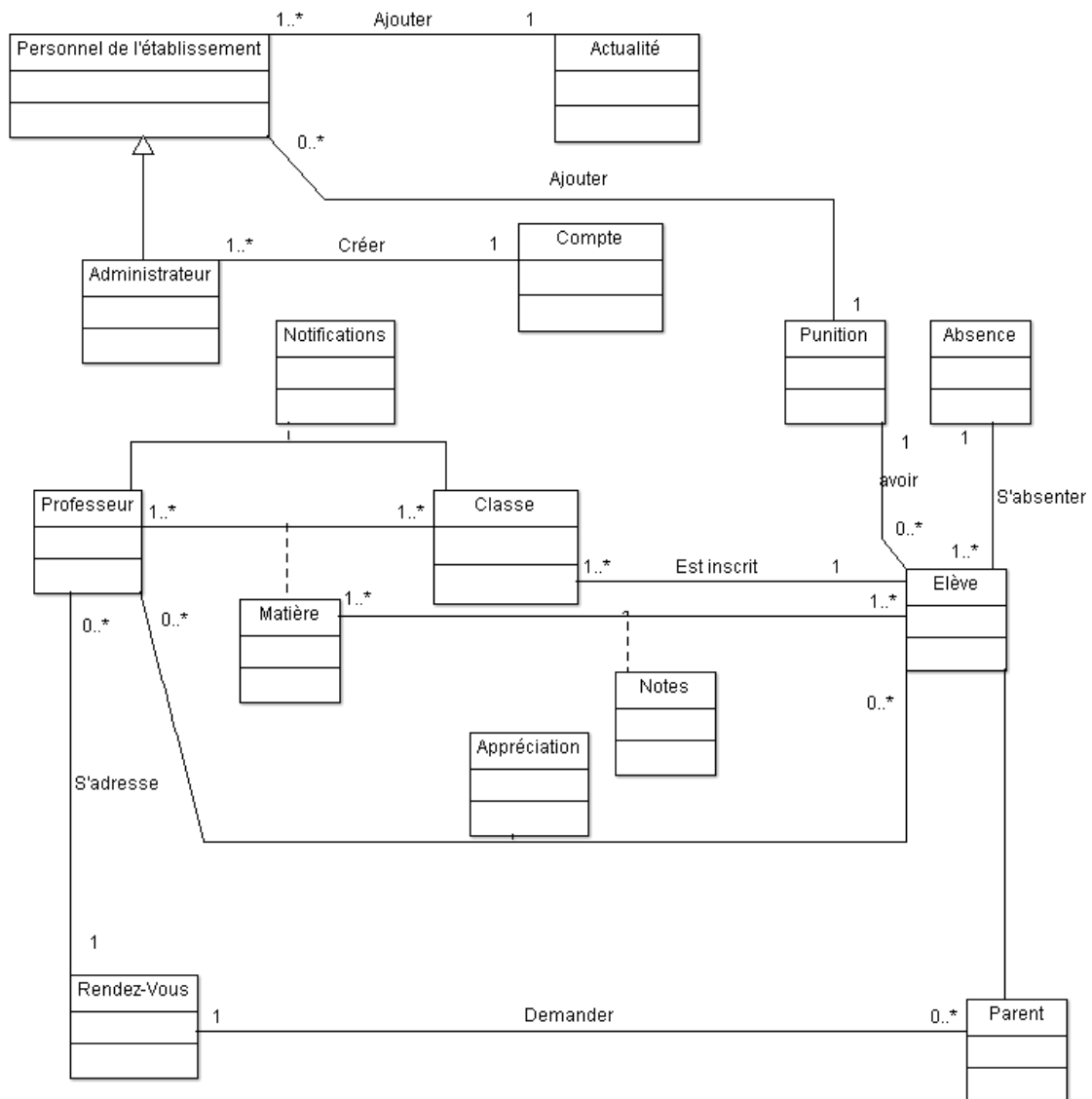


Figure 16: Diagramme de classes

VI.Conclusion

Tout au long de ce chapitre, j'ai détaillé la conception de notre application à travers le diagramme de classes ainsi que les diagrammes de séquences associées afin que la phase réalisation et la mise en place de l'application soit plus souple et plus aisée. Le chapitre suivant mettra en évidence, le fruit de ce passage et les différents résultats du développement de l'application demandée.

Chapitre 5 : Réalisation

I.Introduction

Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Pour cela, après avoir complété l'étude conceptuelle dans le chapitre précédent, nous allons aborder la partie implémentation dans ce qui suit. J'entame ce chapitre la description des environnements matériels et logiciels qui nous ont permis de réaliser notre projet ainsi que l'architecture physique de notre système à travers le diagramme de déploiement. Je passe ensuite à la phase d'implémentation dans laquelle Je vais présenter les différentes techniques que j'ai utilisées pour réaliser notre application.

II.Environnement du travail

Le choix de système Android OS et les programmes de développement est requis pour réaliser notre application, Ce chapitre couvre les prés requis matériels et logiciels qui sont utilisés pour la réalisation de notre application.

1. Environnement matériel

Notre application est réalisée sur un pc portable avec un smartphone pour l'exécution dont les caractéristiques sont résumées dans le tableau suivant :

	Micro-ordinateur (Développement)	Smartphone (Exécution)
Marque	HP Compaq	Huawei Y520-U22
Processeur	Intel(R) Core(TM) i3	1300 MHz, Dual-core Cortex-A7
Disque dur	500 Go	4 Go (Mémoire interne)
RAM	4 Go	512 MB
Système	Windows 7	Android 4.4.2

Tableau 32 : Présentation de l'environnement de travail utilisé

2. Environnement logiciel

Les logiciels utilisés pour la réalisation du projet sont les suivants :



WampServer : est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données. Les fonctionnalités de WampServer sont très complètes et simples à utiliser.

Contrairement aux autres solutions, WampServer permet de reproduire fidèlement son serveur de production.

WampServer est la solution idéale pour les programmeurs débutants, intermédiaires ou avancés qui ont besoin de tester leur site web. En plus des fichiers backend nécessaires, il inclut une interface utilisateur graphique et des panneaux de contrôle.

WampServer 2.5

Version Apache : **2.4.9**

Version de PHP : **5.5.12**

Server Software: **Apache/2.4.9**

Version de MySQL : **5.6.17**



Notepad++ C'est un éditeur de texte générique codé en C++, qui intègre la coloration syntaxique de code source pour les langages et fichiers C, C++, Java, C#, XML, HTML, PHP, JavaScript, etc.

Ce logiciel a pour but de fournir un éditeur léger et efficace. Il est également une alternative au bloc-notes de Windows.



ArgoUML est un logiciel libre de créations de diagrammes UML, fonctionnant sous Java. Il est édité sous licence EPL 1.0. Il est multilingue, supporte la génération de code et l'ingénierie inverse.

ArgoUML supporte sept types de diagrammes : cas d'utilisation, classes, séquences, état, collaboration, activité et déploiement.

En outre, il me permet d'exporter les diagrammes sous les formats PDF, PNG, EPS, SVG ou encore PGML. Enfin, ArgoUML bénéficie d'une interface à la fois conviviale et simple d'utilisation.

Android Studio :



Android Studio est un nouvel environnement pour développement et programmation entièrement intégré qui a été récemment lancé par Google pour les systèmes **Android**. Il a été conçu pour fournir un environnement de développement et une alternative à Eclipse qui est l'IDE le plus utilisé.

Android Studio permet de voir chacun des changements visuels que j'effectue sur notre application et en temps réel, je pourrai voir aussi son effet sur différents appareils Android, chacune avec différentes configurations et configurations simultanément.

Android Studio offre aussi d'autres choses :

- un environnement de développement robuste.
- une manière simple pour tester les performances sur d'autres types d'appareils.
- des assistants et des modèles pour les éléments communs trouvés sur tous les programmeurs Android.
- un éditeur complet avec une panoplie d'outils pour accélérer le développement de notre application.

Les plus d'Android Studio comparativement à Eclipse + ADT :

- solution packagée
- moins de risques liés aux dépendances logicielles (notamment avec Eclipse !)

Android Studio 1.1.0

Licence : Gratuit

Auteur : Google

III.Choix techniques

1. Choix de développement

a. PHP

Hyper Text Preprocessor est un langage de script extrêmement puissant et destiné pour le développement des applications web. PHP est l'un des langages de

programmation les plus populaire. Le point fort de ce langage c'est qu'il est portable et simple à utiliser.

b. JAVA

Est un langage de programmation moderne qui permet une programmation orientée-objet. Ce langage a l'avantage d'être modulaire, rigoureux (la plupart des erreurs se produisent à la compilation et non à l'exécution) et portable (un même programme compilé peut s'exécuter sur différents environnements).

Android est un système d'exploitation conçu pour téléphone mobile développé par Google, qui a mis à disposition un kit de développement logiciel (SDK) basé sur le langage Java.

c. XML (Extensible Markup Language ou Langage Extensible de Balisage)

Pour la réalisation des interfaces graphiques de l'application j'ai utilisé le langage XML. Le XML est un langage informatique qui sert essentiellement à stocker/transférer des données de type texte structurées en champs arborescents. Ce langage est qualifié d'extensible car il permet à l'utilisateur de définir des marqueurs (balises) qui facilitent le parcours au sein du fichier et donc la lecture de l'information. Ce langage est couramment utilisé et son apprentissage est aisé.

2. Protocole et formats de données

a. Protocole de communication

Dans notre projet, j'ai utilisé le protocole HTTP, afin de communiquer les données entre l'application Android et le serveur web. En effet, Le HTTP est un protocole qui définit la communication entre un serveur et un client. En général, j'ai utilisé la méthode Post pour envoyer des données au programme situé à une URL spécifiée.

b. Format de données communiquées

- **JSON** (JavaScript Object Notation) est un format léger d'échange de données. Il peut être aisément analysé et généré par des machines.

Lorsque l'application Android s'exécute, elle se connectera au script PHP. Le script PHP va récupérer les données depuis la base de données MySQL. Ensuite les

données seront encodées au format JSON et envoyées au système Android. Ensuite, l'application Android va obtenir ces données codées. Elle les analysera et les affichera sur l'appareil Android.

Format compréhensible par tous (humain et machine). Aucun apprentissage n'est requis puisque la syntaxe n'utilise que quelques marques de ponctuations, plus sa structure en arborescence et sa syntaxe simple qui lui permet de rester très "léger" et efficace.⁷

3. **Choix de la technologie de sécurité**

Afin d'assurer la sécurité des données et des informations personnelles des utilisateurs, j'ai choisi que chaque utilisateur doit saisir ses droits d'accès (login et mot de passe) pour consulter l'ensemble des services qui lui est associés.

IV. **Présentation de quelques interfaces**

Dans ce qui suit nous présentons quelques interfaces de l'application en citant les détails de chaque imprime écran.

1. **Logo de l'application :**



Figure 17: Logo de l'application

Cette figure présente le logo de notre application intitulée Smart Eduserv.

⁷ <http://www.alsacreations.com>

2. Interface d'authentification



Figure 18: Interface d'authentification



Figure 19: Interface d'authentification avec menu "hors connexion"

Cette interface permet à chaque utilisateur de s'authentifier afin d'accéder à son espace.

3. Interface "Parent de"

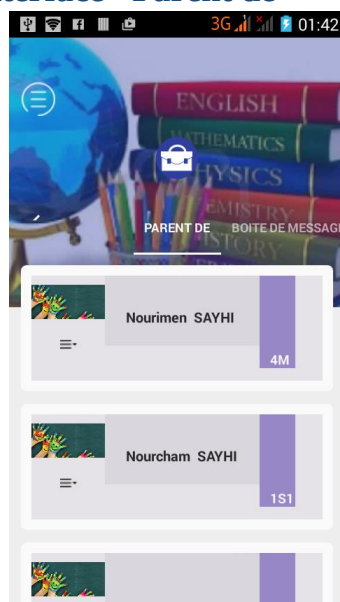


Figure 20: Interface "Parent de"

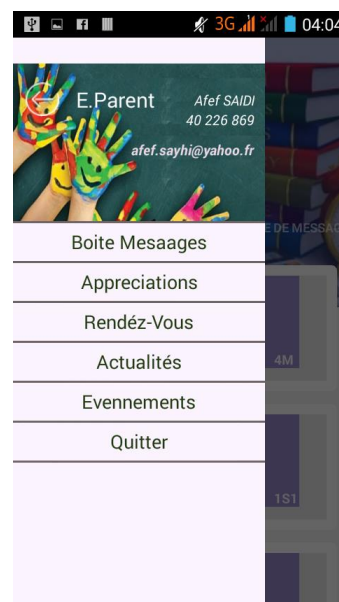


Figure 21: Interface "Parent de" avec menu

Un parent peut avoir plus qu'un enfant à suivre. Cette interface lui affiche la liste de ses enfants « élèves » qui peuvent les suivre.

4. Interface « Divertissement »

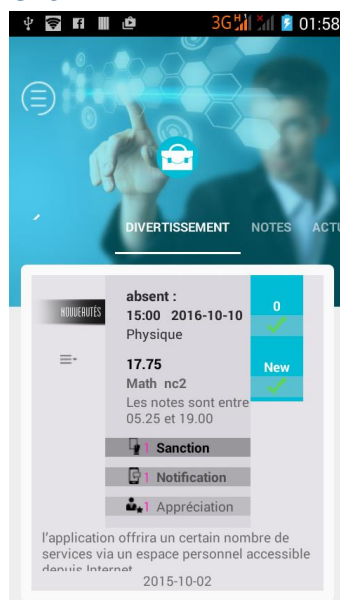


Figure 22: Interface "Divertissement"

Cette interface permet au parent de prendre une vue globale sur tout ce qui est nouveau et qui concerne son enfant.

5. Interface "Notes"

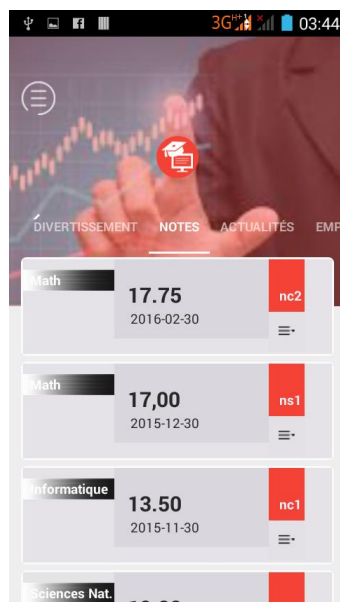


Figure 23: Interface "Notes"

Après avoir choisir un parmi ses enfants à suivre chaque parent peut consulter la liste de ses notes, avec les dates d'insertion et en plus il peut les faire comparer avec celles de ses camarades.

6. Interface "Emploi du temps"



Figure 24: Interface "Emploi du temps"

Un élève ou son parent peut consulter cette interface qui représente toutes les séances durant toute la semaine avec leurs horaires et les salles.

7. Interface "Liste enseignants"



Figure 25: Interface "Liste enseignants"

Cette interface montre la liste des enseignants de l'élève en question.

V.Conclusion

Tout au long de ce chapitre j'ai essayé de présenter les différents travaux qui se déroulent à la fin du cycle de développement 2TUP. J'ai précisé en premier lieu les différents outils utilisés pour réussir le développement de l'application. En deuxième lieu, j'ai présenté la planification de notre projet ainsi que les tests de validation. En troisième lieu, j'ai exposé quelques interfaces de l'application.

Conclusion générale

L'objectif escompté de notre projet de fin d'étude est la conception et le développement d'une application mobile de service scolaires administratifs et pédagogiques en tenant compte des différentes étapes de cycle de vie du développement de notre application. En effet, j'ai exposé la phase d'analyse et spécification des besoins, la phase d'étude préliminaire, la phase de la branche fonctionnelle, la phase de la branche technique, la phase de conception et la phase de réalisation.

Durant ce stage, j'ai appliqué mes connaissances et j'ai acquis des compétences relationnelles et techniques enrichissantes à ma formation. D'ailleurs, ce projet a été bénéfique à plusieurs niveaux :

- Au niveau technique, j'ai eu l'occasion d'enrichir mes connaissances concernant les outils de développement web tels que PHP, JAVA et bien connaître le système de gestion de base de données MySQL. Ainsi j'ai eu l'opportunité de maîtriser le langage de modélisation UML.
- Au niveau personnel, cette expérience pratique et professionnelle m'a permis de découvrir le milieu professionnel avec tout ce qu'il exige de discipline et de responsabilité ce qui va m'aider à enrichir ma vie professionnelle dans des meilleurs niveaux d'appartenance et savoir.

A la fin de ce travail, j'espère qu'il sera utilisé et enrichi dans le futur par d'autres fonctionnalités telles que la résolution et l'envoi des exercices à partir de l'espace de l'élève, la communication entre les enseignants et les parents à partir de leurs espaces, etc.

J'ai, à cet effet, essayé d'adopter les meilleures solutions techniques et méthodes de développement. Dans une première étape, j'ai commencé par introduire le cadre de l'élaboration du projet ainsi qu'une étude théorique sur les notions de base et les technologies employées, suivie d'une étude de l'existant.

Comme d'autres applications Android, notre application peut être aisément améliorée. En effet, grâce à son aspect ouvert, Android offre l'opportunité de créer des

logiciels mobiles innovants et révolutionnaires en encourageant les développeurs à avancer dans leur imagination et à mobiliser toutes leurs compétences pour le meilleur de cette plateforme. Comme travail de futur, je suggère de faire compléter cette application pour qu'elle soit multiplateforme. C'est-à-dire, compatible avec divers systèmes d'exploitation mobiles comme iPhone ou BlackBerry.

Webographie

- www.cnte.tn
- [http:// http://www.eduserv.tn](http://http://www.eduserv.tn)
- <http://www.cafepedagogique.net/>
- [http://fr.wikipedia.org/wiki/UML_\(informatique\)](http://fr.wikipedia.org/wiki/UML_(informatique))
- <http://stackoverflow.com/questions/8515936/android-activity-life-cycle-what-are-all-these-methods-for>
- <http://www.alsacreations.com>
- <http://gs.statcounter.com/#mobile+tablet+console-os-TN-monthly-201403-201503-map>
- <https://openclassrooms.com/courses/creez-des-applications-pour-android/installation-et-configuration-des-outils> <http://android.developpez.com/>
- <http://www.tutos-android.com/>
- <http://www.flaticon.com/>
- <http://www.androidhive.info/>
- <http://www.wampserver.com/>

Annexe

Cycle de vie d'une application Android

L'activité Android surveille et réagit à ces événements par l'instanciation des méthodes qui remplacent les méthodes de la classe d'activité pour chaque événement :

- **onCreate** : C'est la méthode appelée lorsque l'activité est d'abord créée.
- **onStart** : Cette méthode est appelée juste avant que l'activité devienne visible à l'écran.
 - Une fois onStart est terminé, l'activité peut devenir l'activité du premier plan, contrôle le transfert en méthode onResume. Si l'activité ne peut pas devenir l'activité du premier plan pour une raison quelconque, elle sera transférée en méthode onStop.
- **onResume** : Appelée juste après onStart si l'activité est l'activité du premier plan sur l'écran. A ce stade, l'activité est en cours d'exécution et d'interaction avec l'utilisateur. Ministère de l'Enseignement Supérieur et de la Recherche scientifique et de technologies de l'information et de la communication onResume est également appelé, si l'activité cède l'avant-plan à une autre activité et que l'activité existe encore.
- **onPause** : Appelée lorsqu'Android est sur le point de reprendre une activité différente. A ce stade, l'activité n'aura plus accès à l'écran.
- **onStop** : Appelée lorsque l'activité n'est plus visible, soit parce qu'une autre activité a pris le premier plan, soit parce que l'activité est détruite.
- **onDestroy** : C'est la dernière chance pour l'activité à faire tout traitement avant qu'elle ne soit détruite. Mais comme mentionné précédemment, la méthode peut être appelée dans le cas où le système décide d'allouer les ressources système à une autre activité.

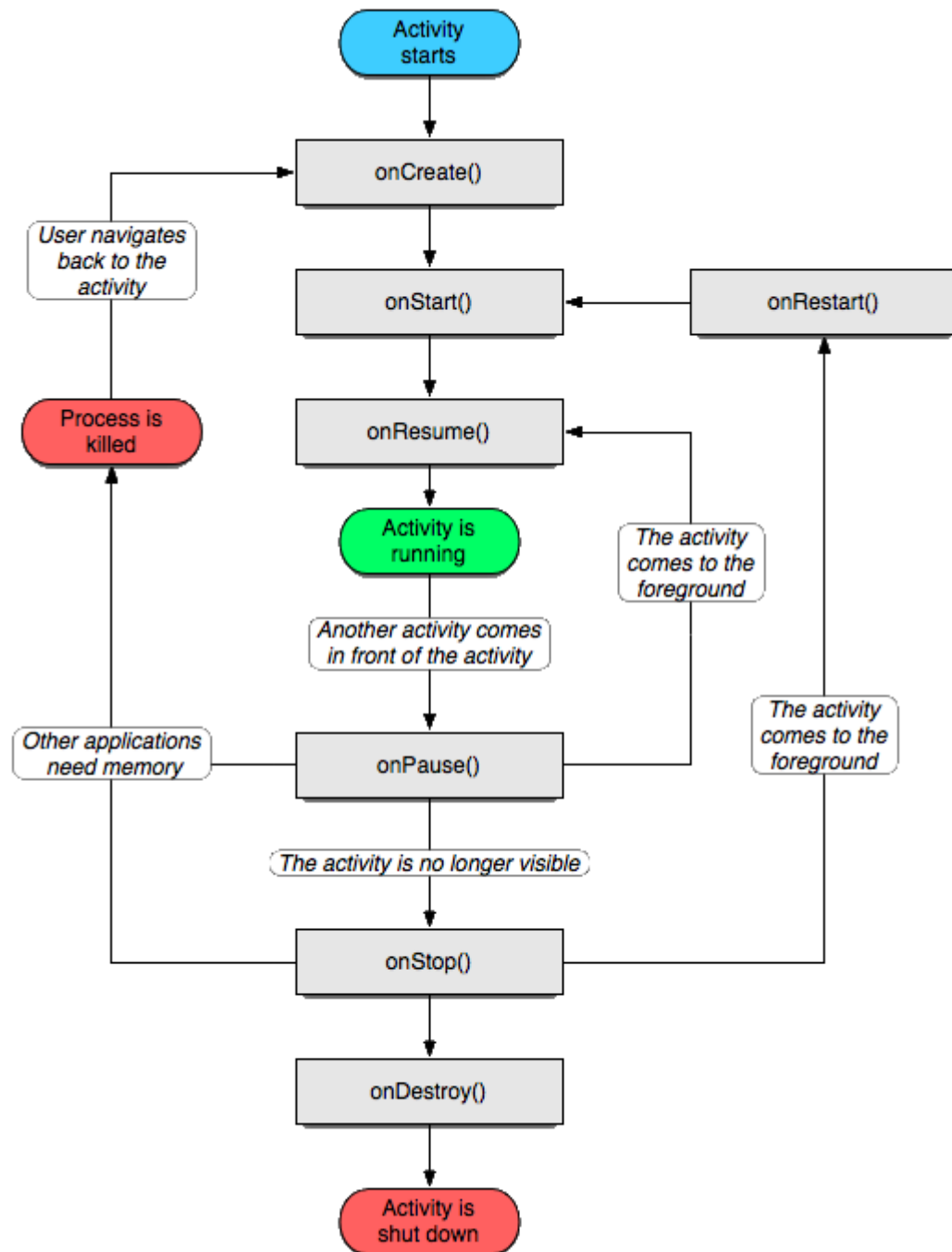


Figure 26: Cycle de vie d'une application Android

Prise en main de l'environnement Android

La première étape de notre travail avec l'environnement Android a été d'appréhender le SDK

- SDK Android

L'outil le plus important est le SDK Android. Facile à installer, il permet de télécharger tous les outils indispensables au développement d'applications. Un petit logiciel permet d'abord de télécharger les différentes versions du SDK (une version du SDK par version d'Android : 1.4, 1.5, 1.6, 2.0 ,4.0 etc.). Il permet également de télécharger les différentes versions des Google APIs (APIs pour intégrer des fonctionnalités liées aux services Google).

- Emulateur

Le SDK propose un émulateur Android. Il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone Android. C'est bien évidemment un outil indispensable pour le développement mobile. A chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel. Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone. Il ne permet par exemple pas d'émuler la gestion du Bluetooth.

- Création d'interfaces utilisateur sous Android

Sous Android, nous pouvons décrire nos interfaces utilisateur de deux façons différentes :

- avec une description déclarative XML ou directement dans le code d'une activité en utilisant les classes adéquates. La façon la plus simple de réaliser une interface est d'utiliser la méthode déclarative XML via la création d'un fichier XML que nous placerons dans le dossier /res/layout de notre projet.

En adoptant cette dernière solution, les interfaces de notre application sont décrites dans des fichiers XML, afin de faciliter leur modification en cas de besoin.

Exemple de code Java (FirstPageActivity.java)

```
package com.example.sayhi.materielviwpagersample;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.StrictMode;
import android.support.v4.app.ActionBarDrawerToggle;
import android.support.v4.widget.DrawerLayout;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.Toast;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONObject;
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;

public class FirstPageActivity extends ActionBarActivity implements View.OnClickListener
{
    private String[] drawerItemsList;
    private ListView myDrawer;
    private ActionBarDrawerToggle mDrawerToggle;
    private DrawerLayout drawerLayout;
    private Button btlogin;
    private EditText etIdent, etpwd;
    public ProgressDialog pDialog;
    public static String idparent="",idparent2="",idparent3="";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_first_page);
StrictMode.enableDefaults();
etIdent = (EditText) findViewById(R.id.eTident);
etpwd = (EditText) findViewById(R.id.eTpwd);
btlogin= (Button) findViewById(R.id.btlogin);
btlogin.setOnClickListener(this);
pDialog = new ProgressDialog(this);
drawerLayout = (DrawerLayout) findViewById(R.id.drawer_layoutFirst);
drawerItemsList = getResources().getStringArray(R.array.itemsfirst);
myDrawer = (ListView) findViewById(R.id.my_drawerlist);
myDrawer.setAdapter(new ArrayAdapter<String>(this,
    R.layout.drawer_item, drawerItemsList));
myDrawer.setOnItemClickListener(new MyDrawerItemClickListener());
getSupportActionBar().setTitle(" ");
getSupportActionBar().setHomeAsUpIndicator(R.drawable.menu_button);
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setHomeButtonEnabled(true);
mDrawerToggle = new ActionBarDrawerToggle(this, drawerLayout,
    R.drawable.sigleu, R.string.ouverture, R.string.fermeture) {
    public void onDrawerClosed(View view) {
        getSupportActionBar().setTitle(" ");
        getSupportActionBar().setHomeAsUpIndicator(R.drawable.menu_button);
        getSupportActionBar().setHomeButtonEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        invalidateOptionsMenu();
    }

    public void onDrawerOpened(View drawerView) {
        getSupportActionBar().setTitle("");
        getSupportActionBar().setHomeAsUpIndicator(R.drawable.retourpage);
        getSupportActionBar().setHomeButtonEnabled(true);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        invalidateOptionsMenu();
    }
};
drawerLayout.setDrawerListener(mDrawerToggle);
View header = getLayoutInflater().inflate(R.layout.headerfirst, myDrawer, false);
myDrawer.addView(header, null, false);
}

public void onClick(View VI) {
    switch (VI.getId()) {
        case R.id.btlogin:
            final String log = "" + etIdent.getText().toString();
            final String pd = "" + etpwd.getText().toString();

            if (log.trim().length() > 0 && pd.trim().length() > 0) {
                checkLogin(log, pd);
            }
        }
    }

```

```

    } else {
        Toast.makeText(getApplicationContext(),
            "Entrez vos paramètres d'authentification!", Toast.LENGTH_LONG)
            .show();
    }
    break;
}
}

private class MyDrawerItemClickListener implements
    ListView.OnItemClickListener {
    @Override
    public void onItemClick(AdapterView<?> adapter, View v, int pos, long id) {
        String clickedItem = (String) adapter.getAdapter().getItem(pos);
        switch (clickedItem) {
            case "A propos":
                Intent intent = new Intent(FirstPageActivity.this, Aprop.class);
                startActivity(intent);
                break;
            case "Demo":
                Intent intent1 = new Intent(FirstPageActivity.this, MainActivity.class);
                startActivity(intent1);
                break;
            case "Calendrier":
                Intent intent2 = new Intent(FirstPageActivity.this, Calendrier.class);
                startActivity(intent2);
                break;
        }
        drawerLayout.closeDrawer(myDrawer);
    }
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    if (mDrawerToggle.onOptionsItemSelected(item)) {
        return true;
    }
    switch (item.getItemId()) {
        case R.id.action_settings:
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

private void checkLogin(final String log, final String pd) {
    String result="";

    InputStream is = null;

```

```

ArrayList<NameValuePair> nameValuePairs = new ArrayList<NameValuePair>(2);

try {
    DefaultHttpClient httpClient = new DefaultHttpClient();
    HttpPost httpPost = new HttpPost("http://192.168.137.1/selectuser2.php");
    nameValuePairs.add(new BasicNameValuePair("username",log.trim())); //
    $Edittext_value = $_POST['Edittext_value'];
    nameValuePairs.add(new BasicNameValuePair("password",pd.trim()));

    httpPost.setEntity(new UrlEncodedFormEntity(nameValuePairs));
    HttpResponse response = httpClient.execute(httpPost);
    HttpEntity entity = response.getEntity();
    is = entity.getContent();
} catch (Exception e) {
    Log.e("log_tag", "Error in the http connection" + e.toString());
}

try { // "iso-8859-1" HTTP.UTF_8
    BufferedReader reader = new BufferedReader(new InputStreamReader(is, "iso-8859-1"),8);
    StringBuilder sb = new StringBuilder();
    String line = null;
    while ((line =reader.readLine())!= null){
        sb.append(line + "\n");
    }
    is.close();
    result = sb.toString();
    // resultView.setText(result);
} catch (Exception e) {
    Log.e("log_tag", "Error converting result" + e.toString());
}

// parse json data
try {
    JSONArray jArray = new JSONArray(result);

    for (int i=0; i<jArray.length();i++) {
        JSONObject json = jArray.getJSONObject(i);

        idparent = idparent + json.getString("idparent");
    }

    if(idparent.length()!=0)
    { pDialog.setMessage("Connexion...الاتصال بالانترنت");
      pDialog.show();
      idparent2=idparent;
      idparent3=idparent;
    }
}

```

```

Intent intent = new Intent(FirstPageActivity.this,Parentde.class);
startActivity(intent);

this.finish();

}
else {
    AlertDialog.Builder a_builder = new AlertDialog.Builder (FirstPageActivity.this);
    a_builder.setMessage("Utilisateur non enregistré !!! المعطيات المدرجة غير مسجلة لدينا");
    .setCancelable(false)
    .setNegativeButton("OK", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
            Intent intent = new Intent(FirstPageActivity.this,FirstPageActivity.class);
            startActivity(intent);
        }
    })
    ;
    AlertDialog alert = a_builder.create();
    alert.setTitle("SmartEduserv, Sécurité!!!");
    alert.setIcon(R.drawable.message_alerte);
    alert.show();

}
// idparent="";

} catch (Exception e) {
    AlertDialog.Builder a_builder = new AlertDialog.Builder (FirstPageActivity.this);
    a_builder.setMessage("Utilisateur non enregistré !!! المعطيات المدرجة غير مسجلة لدينا");
    .setCancelable(false)
    .setNegativeButton("Ok", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
            Intent intent = new Intent(FirstPageActivity.this,FirstPageActivity.class);
            startActivity(intent);
        }
    })
    ;
    AlertDialog alert = a_builder.create();
    alert.setTitle("SmartEduserv, Sécurité!!!");
    alert.setIcon(R.drawable.attention);
    alert.show();
}
}
}

```

Exemple de code XML (FirstPageActivity.xml)

```

<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.sayhi.materielviwpagersample.FirstPageActivity"
android:id="@+id/drawer_layoutFirst"
android:theme="@style/AppTheme"
android:clickable="false"
android:background="@drawable/arrplanf">
    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_alignParentBottom="true"
        android:weightSum="1"
    >
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:id="@+id/eTident"
            android:layout_marginTop="300dp"
            android:layout_marginLeft="32dp"
            android:layout_marginRight="32dp"
            android:autoText="false"
            android:hint="Identifiant"
            android:textColor="#ff000000" />
        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textPassword"
            android:id="@+id/eTpwd"
            android:layout_marginLeft="32dp"
            android:layout_marginRight="32dp"
            android:hint="Mot de passe"
            android:textColor="#ff000000" />
        <LinearLayout
            android:orientation="horizontal"
            android:layout_width="205dp"
            android:layout_height="wrap_content"
            android:layout_weight="0.13"
            android:layout_gravity="center_horizontal|right"
            android:layout_marginRight="32dp">

```

"المعرّف"

"كلمة السر"

```

<ImageView
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:id="@+id/imageView"
    android:src="@drawable/sigleu"
    android:layout_alignParentBottom="true"
    android:background="@drawable/sigleu" />
<LinearLayout
    android:orientation="vertical"
    android:layout_width="204dp"
    android:layout_height="fill_parent">

    <Button
        android:layout_width="match_parent"
        android:layout_height="25dp"
        android:text="Se connecter"
        android:id="@+id/btlogin"
        android:layout_marginRight="32dp"
        android:background="#FF38A2"
        android:layout_marginTop="10dp"
        android:textColor="#ffffff"
        android:layout_gravity="right" />
    <TextView android:text="Espace Parent"
android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#C71585"
        android:textSize="12dp"
        android:layout_marginRight="50dp"
        android:layout_marginLeft="0dp"
        android:layout_gravity="right" />
    </LinearLayout>
</LinearLayout>
</LinearLayout>
<ListView
    android:id="@+id/my_drawerlist"
    android:layout_width="240dp"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:background="@drawable/arrpl2"
    android:choiceMode="singleChoice"
    android:dividerHeight="2dp"
    android:drawSelectorOnTop="false"
    android:elevation="40dp"
    android:footerDividersEnabled="false"
    android:labelFor="@color/abc_primary_text_material_dark"

```

```
        android:divider="#ff706565"  
        android:clickable="false"  
        android:paddingTop="50dp" />  
</android.support.v4.widget.DrawerLayout>
```


Résumé :

Le présent mémoire a été rédigé dans le cadre du Projet de Fin d'Etudes dans le Centre National des Technologies en Education, pour l'obtention du diplôme Mastère en Nouvelle Technologies en Télécommunications et Réseaux (N2TR) de l'Université Virtuelle de Tunis,

Il a pour objectif la conception et la réalisation d'une application mobile de services scolaires administratifs et pédagogiques.

Mots Clés : *Android, Java, PHP, XML, MySQL*

ملخص:

تم تنفيذ هذا العمل ضمن المركز الوطني للتكنولوجيات في التربية، في إطار مشروع التخرج والحصول على شهادة الماجستير المهني "التقنيات الجديدة في الاتصالات والشبكات من الجامعة الافتراضية بتونس. الهدف من هذا العمل هو تطوير برنامج خدمات مدرسية يعمل على منظومة التشغيل أندرويد.

الكلمات المفتاحية: *Android, Java, PHP, XML, MySQL*

Summary :

This work was written as part of the End of Studies Project in the Technology Center in National Education, for obtaining the Master degree in New Technologies in Telecommunications and Networks (N2TR) of the Virtual University of Tunis

He aims for the design and implementation of a mobile application of administrative and pedagogical school services.

Keywords : *Android, Java, PHP, XML, MySQL*