

RÉPUBLIQUE DÉMOCRATIQUE DU CONGO ENSEIGNEMENT
SUPÉRIEUR ET UNIVERSITAIRE



FACULTÉ DES SCIENCES INFORMATIQUES

**Développement d'une application d'aide à la délibération des
étudiants de l'Université Nouveaux Horizons.**

TRAVAIL DE FIN DE CYCLE

Présenté par : Musanzi Mavula Wilfried
Filière : Informatique de Gestion

Directeur : Dr. Saint Jean Djungu
Co-directeur : Mr. Jonathan Kabemba

Année académique 2022 - 2023

RÉSUMÉ

ABSTRACT

RÉMERCIEMENTS

TABLE DES MATIÈRES

Résumé	1
Abstract	2
Rémerciements	3
Introduction générale	6
0.1 Contexte et motivation	7
0.2 Problématique	7
0.3 Etat de la question	8
0.4 Objectifs	8
0.5 Méthodes et techniques	8
0.5.1 Méthodes	8
0.5.2 Techniques	9
1 Généralités	10
1.1 La délibération	10
1.2 Une application	10
1.3 Un étudiant	10
1.4 Une API	11
1.5 L’université Nouveaux Horizons	11
1.6 Les méthodes de développement logiciel	12
1.6.1 La méthode en cascade	12
1.6.2 Les méthodes agiles	13
1.6.3 La méthode en V	13
1.6.4 La méthode RAD	14
1.6.5 La méthode DevOps	15
1.6.6 Processus unifié	15
1.7 Modélisation	16
1.8 UML (Unified Modeling Language)	16
1.8.1 Diagramme de classes	16
1.8.2 Diagramme de cas d’utilisation	16
1.8.3 Diagramme d’objets	16
1.8.4 Diagramme de séquence	16
1.8.5 Diagramme de collaboration	16
1.8.6 Diagramme d’états-transitions	17
1.8.7 Diagramme d’activités	17
1.8.8 Diagramme de déploiement	17
1.9 Language de programmation	17
1.10 IDE (Integrated Development Environment)	18

2	État de l'art	19
2.1	Présentation des principales applications existantes	19
2.1.1	Moodle	19
2.1.2	PowerSchool	20
2.1.3	Esis Salama	21
2.2	conclusion	21

Introduction générale

La délibération est un processus de prise de décision qui implique une discussion et une réflexion approfondies sur un sujet ou une question donnée. C'est une méthode qui permet de discuter, d'examiner et d'analyser les différentes options et perspectives avant de prendre une décision.

Dans le contexte d'une assemblée délibérative, telle qu'un conseil municipal, une assemblée générale ou un comité, la délibération se déroule généralement selon un processus formel qui permet à chaque membre de présenter son point de vue et d'argumenter en faveur de ses positions.

L'objectif de la délibération est d'aboutir à une décision éclairée et consensuelle, qui prend en compte les différents points de vue et les intérêts en présence. Ce processus permet également de renforcer la transparence et la légitimité de la décision prise, en impliquant tous les participants dans la prise de décision.

La délibération est souvent considérée comme une méthode de prise de décision plus démocratique et participative que d'autres approches, telles que la décision unilatérale ou le vote à la majorité simple. Elle est utilisée dans de nombreux domaines, tels que la politique, l'éducation, les affaires, et le droit, et bien d'autres pour n'en citer que ceux-ci.

0.1 Contexte et motivation

Ce projet est axé sur la délibération à l'université Nouveaux Horizons, où cette dernière est, chaque année confrontée à la prise des décisions importantes concernant les étudiants tout au long de leur parcours universitaire. La délibération a un impact significatif sur la vie des étudiants, et il est donc important qu'elle soit faite de manière éclairée et réfléchie car elle détermine le sort des étudiants de manière individuelle, qui passe et qui ne passe pas et avec quelle mention.

La motivation derrière ce projet est de fournir à l'université Nouveaux Horizons un outil pratique, efficace, sur mesure, flexible pour les aider à prendre des décisions éclairées et réduire le circuit de circulation des données.

En fin de compte, ce projet a pour objectif de développer une application web qui permettra à l'université Nouveaux Horizons de gérer la délibération des étudiants ; De la collecte des notes à la publication des résultats.

0.2 Problématique

Actuellement, à l'Université Nouveaux Horizons la délibération des étudiants se passe comme suit :

- A la fin des examens le professeur chargé du cours envoie soit les notes annuelles soit les notes de tous les travaux ainsi que l'examen au décanat de l'Université en copie au doyen de la faculté.
- Le décanat après réception transmet les notes reçues soit au format pdf soit Excel(xls) soit manuscrit au président du jury.
- Le jury à son calcul les moyennes et après délibération communique les résultats aux étudiants.

Vous conviendrez avec moi que le circuit de circulation des informations est assez long, de plus, il est difficile de gérer les données des étudiants de manière efficace et la publication des résultats n'est pas aussi évidente qu'elle le devrait. Les dites données peuvent être :

- les cours en compléments,
- le(s) relevé(s) de chaque année parce qu'un étudiant peut en avoir plusieurs en une année,
- etc.

0.3 Etat de la question

Etant dans une université, qui est loin d'être la seule au pays et dans le monde, nous avons pensé que d'autres universités ont forcément eu à faire face à la même problématique que nous. Nous nous sommes donc intéressés à la manière dont les autres universités gèrent la délibération des étudiants et les outils qu'elles utilisent. Nous avons trouvé des concepts qui nous ont aidé à mieux comprendre la problématique et à mieux cerner les besoins. Nous nous manquerons pas de les mentionner dans la suite de ce travail.

0.4 Objectifs

En vue d'apporter une solution efficace, flexible et solide à la problématique posée, nous avons défini l'objectif de fournir une application web qui :

- permettra de raccourcir le circuit de circulation des informations,
- permettra de faciliter la publication des résultats,
- permettra de faciliter le suivi des données des étudiants,
- peut s'intégrer facilement dans le système d'information de l'université, et aux autres applications existantes telles que moodle, Goole Classroom, etc.

0.5 Méthodes et techniques

Pour la réalisation de ce travail nous avons utilisé plusieurs méthodes et techniques qui nous ont permis de mener à bien notre projet en réunissant les informations nécessaires et en les analysant pour en tirer des conclusions.

0.5.1 Méthodes

1. La méthode analytico-déductive : Nous avons analysé la problématique évoquée dans une section ci-haut en partant des faits concrets pour aboutir à une conclusion générale.
2. La méthode descriptive : Nous avons décrit notre problématique de manière précise et objective.
3. La méthode comparative : Nous avons eu à comparer la manière dont la problématique est gérée ailleurs pour en dégager les similitudes et les différences.

0.5.2 Techniques

Nous avons utilisée est la technique documentaire et plus précisément la technique de la recherche bibliographique. Nous avons consulté des ouvrages, des articles, des documents et des sites web pour avoir des informations sur les applications existantes et les technologies utilisées pour leur développement. Nous avons aussi consulté des documents sur les méthodes de conception et de développement d'applications web.

Nous avons aussi utilisé la technique de l'entretien pour avoir des informations sur les besoins du corps académique de l'Université Nouveaux Horizons et sur les fonctionnalités qu'ils souhaiteraient avoir dans une application d'aide à la délibération également sur la manière dont les universités soeurs gèrent cette problématique.

Chapitre 1

Généralités

1.1 La délibération

La délibération est un processus de réflexion et de prise de décision collectif qui implique l'examen et l'évaluation de différentes options ou perspectives, souvent en vue d'arriver à un consensus ou à une décision commune. La délibération peut avoir lieu dans de nombreux contextes, tels que l'enseignement, les gouvernements, les organisations, les groupes de travail ou même les conversations informelles entre amis et collègues.

Dans une délibération, les participants discutent ouvertement et honnêtement des différents points de vue, arguments et preuves en faveur ou en défaveur de chaque option. L'objectif est souvent d'arriver à une décision qui soit juste, équitable et acceptable pour toutes les parties impliquées.

La délibération peut prendre différentes formes, allant des discussions informelles aux débats structurés et aux processus de consultation plus formels. Les participants peuvent également utiliser des outils tels que des sondages, des votes ou des simulations pour faciliter la prise de décision.

1.2 Une application

En informatique, une application (ou "app" en abrégé) est un logiciel conçu pour effectuer une tâche ou une série de tâches spécifiques sur un ordinateur, un smartphone, une tablette ou un autre appareil électronique. Les applications sont généralement créées pour répondre à un besoin spécifique de l'utilisateur ou pour fournir une fonctionnalité particulière, telle que la gestion de tâches, la navigation GPS, la communication en ligne, le traitement de texte, le montage de photos, etc.

Les applications peuvent être accessibles via un navigateur tel que Google Chrome, Firefox, Microsoft Edge, etc ou téléchargées et installées à partir de diverses sources, telles que les boutiques d'applications en ligne, les sites web des développeurs ou les disques d'installation. Certaines applications sont gratuites, tandis que d'autres sont payantes, et les prix peuvent varier considérablement en fonction de la complexité et de la popularité de l'application.

Les applications peuvent être développées pour différents systèmes d'exploitation, tels que iOS, Android, Windows, MacOS, Linux, etc. Les développeurs peuvent utiliser différents langages de programmation et environnements de développement pour créer des applications, tels que JavaScript, Typescript, Swift, Kotlin, Python, C++, etc.

1.3 Un étudiant

Un étudiant est une personne qui s'inscrit à un programme d'enseignement dans une école, un collège, une université ou un autre établissement d'enseignement pour poursuivre des études dans un domaine d'intérêt particulier. Les étudiants peuvent poursuivre des études à différents niveaux

d'enseignement, allant de l'éducation primaire à l'enseignement supérieur et à la formation professionnelle.

Les étudiants sont généralement motivés par un désir d'apprendre et d'acquérir des compétences et des connaissances dans leur domaine d'études choisi. Ils peuvent suivre des cours dans une variété de disciplines, telles que les sciences, les mathématiques, les arts, les sciences sociales, les sciences humaines, les affaires, etc.

En tant qu'étudiants, ils sont souvent tenus de participer à des cours, des séminaires, des projets de groupe et des travaux pratiques pour compléter leur programme d'études. Les étudiants peuvent également être impliqués dans des activités parascolaires, telles que des clubs, des équipes sportives, des organisations étudiantes, etc.

Les étudiants peuvent poursuivre différents objectifs en poursuivant des études, tels que l'acquisition de compétences professionnelles, l'obtention de diplômes, la poursuite d'une carrière, la poursuite de la recherche, l'exploration des intérêts personnels, etc.

1.4 Une API

Une API, ou "Application Programming Interface", est une interface de programmation qui permet à des applications ou des services informatiques de communiquer entre eux.

Plus précisément, une API expose un ensemble de fonctionnalités et de données spécifiques, généralement sous la forme de points d'entrée accessibles à distance par le biais d'Internet. Ces points d'entrée sont souvent basés sur des protocoles standardisés tels que HTTP (utilisé par les API REST) ou SOAP.

Les développeurs peuvent utiliser ces points d'entrée pour échanger des données et des instructions avec l'API et accéder aux fonctionnalités qu'elle expose, sans avoir à connaître les détails internes de la mise en œuvre de l'API.

Les API sont largement utilisées dans les applications web et mobiles, les services cloud, les réseaux sociaux, les systèmes d'intégration d'entreprise, et dans de nombreux autres contextes. Elles permettent aux développeurs de créer des applications plus rapidement en réutilisant des fonctionnalités existantes et en évitant de devoir écrire du code complexe pour chaque nouvelle fonctionnalité.

1.5 L'université Nouveaux Horizons

Le désir et la volonté d'offrir à la jeunesse une éducation de qualité sont les grandes motivations à l'origine de la création de l'Université Nouveaux Horizons. A cette immense volonté viennent en surcouche deux faits du vécu social qui sont : L'incitation et suggestion des tiers : Pendant plusieurs années des nombreuses personnes ont incités les promoteurs de l'Université Nouveaux Horizons à créer une institution académique, dans le prolongement de l'enseignement de fin d'études à l' GE D'OR, sans aucun doute on fait mention des parents des élèves de ladite école, des amis, et autres. Le regard de l'élève tourné vers l'étranger pour la suite des études après le cycle de l'enseignement secondaire : Le fait est que des nombreux élèves de la République Démocratique du Congo en fin de leurs études secondaires n'aspirent qu'à s'expatrier, pour aller s'inscrire, sous d'autres cieux car pensant que l'ailleurs et mieux que le chez soi ; dans des Universités supposées de meilleure qualité. Par ceci on aperçoit aisément la critique souvent formulée à l'endroit des titres académiques décernés par l'Université en République Démocratique du Congo. Par ailleurs, un autre fait ayant incité les initiateurs à créer l'Université Nouveaux Horizons est en quelque sorte ce qu'on pourrait appeler "l'héritage colonial" légué aux Universités Congolaises dans leur ensemble celui qui consiste à former des demandeurs d'emploi et non des créateurs d'emplois.

L'Université Nouveaux Horizons a pour but de dispenser des enseignements dans différentes filières, avec accent sur l'esprit d'entreprise (entrepreneuriat) ; de former des hommes et des femmes

capables de participer efficacement au développement de la communauté, davantage comme des créateurs que des demandeurs d'emplois. L'UNH assure un enseignement de qualité et de niveau international grâce à des compétences requises, judicieusement recrutées, pour cette fin.

1.6 Les méthodes de développement logiciel

Une méthode de développement logiciel est une approche structurée pour planifier, concevoir, construire, tester et maintenir un logiciel de manière efficace et efficiente. Les méthodes de développement logiciel fournissent un cadre pour l'ensemble du processus de développement, de la planification initiale à la livraison du produit final.

Les méthodes de développement logiciel peuvent inclure des techniques, des outils, des pratiques et des processus pour faciliter la collaboration entre les membres de l'équipe de développement, pour gérer les risques de projet, pour assurer la qualité du logiciel, pour planifier et suivre les progrès du projet, etc.

Les méthodes de développement logiciel peuvent varier en fonction des besoins et des objectifs du projet. Certaines méthodes sont axées sur la rapidité et la flexibilité, tandis que d'autres mettent l'accent sur la qualité, la rigueur et la planification minutieuse. Certaines méthodes sont plus adaptées aux projets de grande envergure, tandis que d'autres sont plus adaptées aux projets plus petits et plus agiles.

Il existe plusieurs méthodes de développement logiciel, chacune avec ses propres avantages et inconvénients. Voici un aperçu de quelques-unes des méthodes les plus courantes seront citées ci-dessous.

1.6.1 La méthode en cascade

La méthode en cascade est une méthode de développement logiciel qui suit une approche linéaire et séquentielle pour le développement de logiciels. Elle est également connue sous le nom de modèle de cycle de vie en cascade ou de modèle de développement en cascade. La méthode en cascade est composée de plusieurs étapes linéaires, chacune étant exécutée dans l'ordre séquentiel suivant :

1. Analyse des besoins : Dans cette étape, les exigences et les spécifications du logiciel sont identifiées et définies en détail. Cela implique une compréhension approfondie des besoins des utilisateurs, des fonctionnalités nécessaires du logiciel et des contraintes techniques.
2. Conception : Dans cette étape, une architecture de haut niveau du logiciel est conçue en utilisant les exigences et les spécifications établies dans l'étape précédente. Cette étape implique également la conception des interfaces utilisateur, des bases de données et des algorithmes nécessaires pour implémenter les fonctionnalités du logiciel.
3. Implémentation : Dans cette étape, le code est écrit et les fonctionnalités requises sont implémentées selon les spécifications établies dans les étapes précédentes.
4. Test : Dans cette étape, le logiciel est testé pour s'assurer qu'il fonctionne correctement et qu'il répond aux exigences et aux spécifications établies dans les étapes précédentes. Les tests peuvent inclure des tests unitaires, des tests de système, des tests de performance, des tests d'acceptation, etc.
5. Maintenance : Dans cette étape, le logiciel est maintenu et soutenu après sa livraison. Cela peut inclure des mises à jour, des corrections de bogues, des améliorations de performance, etc.

La méthode en cascade est simple et facile à comprendre, mais elle peut être rigide et ne permet pas de réagir facilement aux changements. Cela est dû au fait que chaque étape doit être complétée avant de passer à la suivante, et tout changement apporté à une étape ultérieure peut avoir des répercussions sur les étapes précédentes. Par conséquent, la méthode en cascade est souvent utilisée pour les projets bien définis avec des exigences claires et stables.

1.6.2 Les méthodes agiles

Les méthodes agiles sont des méthodes de développement logiciel qui mettent l'accent sur l'adaptabilité, la collaboration, la flexibilité et la réactivité aux changements. Les méthodes agiles cherchent à livrer des logiciels fonctionnels rapidement et régulièrement tout en s'adaptant aux changements des exigences du client et de l'environnement de développement. Voici quelques détails sur les méthodes agiles les plus courantes :

1. Scrum est une méthode agile populaire qui se concentre sur la collaboration étroite entre les membres de l'équipe de développement et le client. Les projets sont organisés en sprints, qui sont des périodes courtes de développement allant de deux à quatre semaines. Pendant chaque sprint, l'équipe de développement travaille pour livrer une fonctionnalité fonctionnelle. À la fin de chaque sprint, l'équipe se réunit pour une réunion de rétrospective pour discuter de ce qui s'est bien passé et de ce qui peut être amélioré.
2. XP (Extreme Programming) : XP est une méthode agile qui se concentre sur l'écoute du client et la rapidité de livraison. XP se concentre sur l'automatisation des tests, la programmation en binôme, la planification continue et l'intégration continue. Les pratiques d'XP aident les équipes à travailler ensemble efficacement et à livrer des logiciels fonctionnels rapidement.
3. Kanban : Kanban est une méthode agile qui se concentre sur la gestion visuelle de projet. Les projets sont organisés en tâches individuelles qui sont représentées sur un tableau Kanban. Les tâches passent par plusieurs étapes, de l'analyse à la livraison. Les équipes de développement travaillent à équilibrer le nombre de tâches en cours de façon à éviter les goulots d'étranglement.
4. Lean : Le développement Lean se concentre sur l'élimination des déchets dans le processus de développement, permettant d'optimiser le flux de travail. La méthode Lean met l'accent sur la collaboration étroite entre les membres de l'équipe de développement et le client, ainsi que sur la livraison rapide et continue de logiciels fonctionnels.

Les méthodes agiles sont généralement utilisées pour les projets de développement logiciel complexes et/ou dynamiques où les exigences du client peuvent évoluer au fil du temps. Les méthodes agiles sont souvent préférées pour leur capacité à s'adapter aux changements, à leur flexibilité et à leur efficacité.

1.6.3 La méthode en V

La méthode en V est une méthode de développement logiciel qui est similaire à la méthode en cascade, mais qui met davantage l'accent sur les tests. La méthode en V est également connue sous le nom de modèle de cycle de vie en V ou de modèle de développement en V.

La méthode en V suit un processus linéaire, mais elle met en correspondance chaque étape de développement avec une étape de test correspondante, formant ainsi une forme de V. Les étapes supérieures de la V représentent les étapes de conception et de spécification, tandis que les étapes inférieures représentent les étapes de test et de validation.

1. Analyse des besoins : Dans cette étape, les exigences et les spécifications du logiciel sont identifiées et définies en détail. Cela implique une compréhension approfondie des besoins des utilisateurs, des fonctionnalités nécessaires du logiciel et des contraintes techniques.
2. Spécification : Dans cette étape, les spécifications du logiciel sont détaillées et documentées en utilisant des techniques telles que les diagrammes UML, les descriptions textuelles, etc.
3. Conception : Dans cette étape, une architecture de haut niveau du logiciel est conçue en utilisant les spécifications et les exigences établies dans les étapes précédentes. Cette étape implique également la conception des interfaces utilisateur, des bases de données et des algorithmes nécessaires pour implémenter les fonctionnalités du logiciel.

4. Programmation : Dans cette étape, le code est écrit et les fonctionnalités requises sont implémentées selon les spécifications établies dans les étapes précédentes.
5. Tests unitaires : Dans cette étape, chaque unité de code est testée de manière isolée pour s'assurer qu'elle fonctionne correctement.
6. Tests d'intégration : Dans cette étape, les unités de code sont intégrées pour former des modules fonctionnels qui sont testés pour s'assurer qu'ils fonctionnent ensemble correctement.
7. Tests de système : Dans cette étape, le logiciel est testé dans son ensemble pour s'assurer qu'il fonctionne correctement et qu'il répond aux exigences et aux spécifications établies dans les étapes précédentes.
8. Tests d'acceptation : Dans cette étape, le logiciel est testé par les utilisateurs finaux pour s'assurer qu'il répond à leurs besoins et à leurs attentes.
9. Maintenance : Dans cette étape, le logiciel est maintenu et soutenu après sa livraison. Cela peut inclure des mises à jour, des corrections de bogues, des améliorations de performance, etc.

La méthode en V est rigoureuse et garantit un niveau élevé de qualité, mais elle peut être coûteuse et nécessite une planification minutieuse. La méthode en V est souvent utilisée pour les projets de développement logiciel critiques, tels que les logiciels de mission, les systèmes de contrôle de la sécurité, etc., où la qualité est primordiale.

1.6.4 La méthode RAD

La méthode RAD (Rapid Application Development) est une méthode de développement logiciel qui se concentre sur la rapidité de livraison des logiciels en utilisant des cycles de développement courts et itératifs combinés à des techniques de prototypage. Le RAD est souvent utilisé pour les projets de développement logiciel qui ont des exigences changeantes et des délais de livraison serrés.

Le RAD suit un processus de développement rapide qui se compose de quatre phases principales :

1. Planification : Dans cette phase, les exigences du client sont identifiées et une planification détaillée du projet est réalisée. Cette phase implique également la définition des objectifs du projet, la définition du budget et du calendrier, la définition des rôles et responsabilités de l'équipe de développement, etc.
2. Analyse : Dans cette phase, les exigences sont analysées en détail et les spécifications du logiciel sont définies. Cette phase implique également la conception des prototypes pour valider les exigences du client et pour recueillir leurs commentaires.
3. Conception : Dans cette phase, la conception détaillée du système est réalisée en utilisant les spécifications et les commentaires du client. Les prototypes sont également améliorés et affinés en fonction des commentaires du client.
4. Construction : Dans cette phase, le logiciel est développé de manière itérative et incrémentale en utilisant des cycles de développement courts. Les tests sont effectués tout au long de cette phase pour s'assurer que le logiciel est de haute qualité et fonctionne correctement.

Le RAD met l'accent sur la collaboration étroite entre les membres de l'équipe de développement et le client, ainsi que sur la livraison rapide et continue de logiciels fonctionnels. Les avantages du RAD incluent une meilleure réponse aux besoins changeants du client, une livraison plus rapide des logiciels et une meilleure qualité de l'application grâce à la collaboration étroite entre les membres de l'équipe de développement et le client.

Cependant, le RAD peut être inadapté pour les projets qui ont des exigences stables et bien définies ou pour les projets qui nécessitent une planification détaillée et rigoureuse. Le RAD est souvent utilisé pour les projets de développement logiciel qui ont des exigences changeantes et des délais de livraison serrés, tels que les projets de développement de logiciels Web.

1.6.5 La méthode DevOps

La méthode DevOps (Development and Operations) est une approche de développement logiciel qui vise à améliorer la collaboration et la communication entre les équipes de développement et d'exploitation, en vue d'accélérer le processus de développement et de déploiement de logiciels.

La méthode DevOps est basée sur trois principes clés :

1. Collaboration : La méthode DevOps encourage la collaboration étroite entre les équipes de développement et d'exploitation, afin de favoriser la communication, la compréhension mutuelle et la prise de décisions en commun.
2. Automatisation : La méthode DevOps utilise l'automatisation pour accélérer le processus de développement et de déploiement de logiciels. L'automatisation peut inclure des tests automatisés, des déploiements automatisés et des intégrations continues.
3. Amélioration continue : La méthode DevOps encourage une culture d'amélioration continue, où les équipes cherchent constamment à améliorer les processus de développement et de déploiement de logiciels en utilisant des feedbacks des utilisateurs finaux et des données de performance.

La méthode DevOps implique également l'utilisation d'outils et de technologies spécifiques, tels que les outils de gestion de source, les plates-formes de conteneurs, les outils de test automatisés et les outils de déploiement automatisé.

Les avantages de la méthode DevOps incluent une plus grande agilité et flexibilité, une amélioration de la qualité du logiciel, une réduction des coûts et une amélioration de la satisfaction des utilisateurs finaux grâce à une livraison plus rapide de logiciels fonctionnels.

Cependant, la mise en œuvre de la méthode DevOps peut être complexe et nécessiter une transformation culturelle significative au sein de l'organisation. La méthode DevOps nécessite également des compétences techniques et une compréhension approfondie des outils et des technologies utilisés.

1.6.6 Processus unifié

Le Processus Unifié (PU), également connu sous le nom de Rational Unified Process (RUP), est une méthode de développement logiciel qui suit une approche itérative et incrémentale. Le PU est un cadre de développement logiciel qui fournit des directives détaillées pour toutes les étapes du développement logiciel, de la planification initiale à la maintenance et la mise à jour continue du logiciel. Le processus Unifié est une méthode de développement logiciel agile de la lignée dite unifiée qui nécessite une adaptation à chaque projet qui y a recours, elle est générique, itérative et incrémentale.

Le PU est basé sur quatre principes fondamentaux :

- La modélisation visuelle : Le PU utilise des diagrammes UML (Unified Modeling Language) pour représenter le système logiciel.
- La gestion de projet : Le PU utilise une approche de gestion de projet itérative et agile qui permet de s'adapter aux changements des exigences et aux problèmes qui surgissent pendant le développement logiciel.
- La vérification et la validation : Le PU insiste sur la vérification et la validation du logiciel à chaque étape du processus pour s'assurer que le logiciel est conforme aux exigences du client et qu'il est de haute qualité.
- L'architecture centrée sur les cas d'utilisation : Le PU se concentre sur la conception de l'architecture du logiciel en se basant sur les cas d'utilisation qui décrivent les interactions entre les utilisateurs et le système.

Le PU est structuré en quatre phases principales :

1. La phase d'inception : Cette phase consiste à définir le périmètre du projet, à identifier les objectifs et les exigences, et à élaborer une vision claire du système logiciel.
2. La phase d'élaboration : Cette phase consiste à élaborer une architecture détaillée du système logiciel et à développer un plan de développement détaillé.
3. La phase de construction : Cette phase consiste à développer et à tester le logiciel en utilisant une approche itérative et incrémentale.
4. La phase de transition : Cette phase consiste à déployer le logiciel et à assurer la formation des utilisateurs, la maintenance et la mise à jour continue.

1.7 Modélisation

La modélisation est le processus de représentation d'un système, d'un processus, d'un objet ou d'un concept sous forme de modèle.

Un modèle est une représentation abstraite et simplifiée d'un système réel, qui peut être utilisée pour comprendre, analyser, prédire et concevoir le système.

1.8 UML (Unified Modeling Language)

UML (Unified Modeling Language) est un langage de modélisation visuel utilisé pour représenter des systèmes logiciels. UML fournit un ensemble de notations graphiques standardisées pour décrire les différents aspects d'un système, tels que les exigences, la conception, la mise en œuvre et le déploiement.

Voici les principales notations graphiques utilisées dans UML :

1.8.1 Diagramme de classes

Utilisé pour représenter la structure statique du système, en définissant les classes, les attributs, les méthodes et les relations entre les classes.

1.8.2 Diagramme de cas d'utilisation

Utilisé pour représenter les interactions entre les acteurs et le système, et pour définir les exigences fonctionnelles du système.

1.8.3 Diagramme d'objets

Utilisé pour représenter les instances des classes et les relations entre les objets.

1.8.4 Diagramme de séquence

Utilisé pour représenter les interactions entre les objets dans une séquence temporelle, en montrant les messages échangés entre les objets.

1.8.5 Diagramme de collaboration

Diagramme de collaboration : Utilisé pour représenter les interactions entre les objets en termes de messages échangés et de relations entre les objets.

1.8.6 Diagramme d'états-transitions

Utilisé pour représenter le comportement dynamique d'un objet ou d'un système, en montrant les états possibles et les transitions entre ces états.

1.8.7 Diagramme d'activités

Utilisé pour représenter le flux de contrôle dans un processus ou une méthode, en montrant les activités, les décisions et les boucles.

1.8.8 Diagramme de déploiement

Utilisé pour représenter la configuration physique du système, en montrant les composants matériels et logiciels et leurs relations.

1.9 Langage de programmation

Un langage de programmation est un langage formel utilisé pour écrire des instructions qui seront exécutées par un ordinateur. Les langages de programmation sont utilisés pour créer des logiciels, des applications, des sites web, des systèmes d'exploitation, des jeux vidéo et de nombreux autres types de programmes informatiques.

Un langage de programmation se compose d'un ensemble de règles syntaxiques et sémantiques qui définissent la structure et le sens des instructions. Les instructions sont écrites sous forme de code source, qui est ensuite traduit en code machine par un compilateur ou un interprète.

Les langages de programmation peuvent être classés en plusieurs catégories, telles que :

1. Les langages de programmation impératifs : qui décrivent les étapes à suivre pour accomplir une tâche. Les langages de programmation impératifs peuvent être classés en deux catégories : les langages de programmation procéduraux et les langages de programmation orientés objet.
2. Les langages de programmation fonctionnels : qui se concentrent sur les fonctions mathématiques et l'évaluation d'expressions, etc.
3. Les langages de programmation orientés objet : qui sont basés sur des objets et leurs interactions, etc. Les langages de programmation orientés objet peuvent être classés en deux catégories : les langages de programmation orientés objet basés sur les classes et les langages de programmation orientés objet basés sur les prototypes.
4. Les langages de script : qui sont utilisés pour automatiser des tâches et des processus, etc. Les langages de script sont généralement interprétés plutôt que compilés.
5. Les langages de programmation déclaratifs : qui décrivent le résultat souhaité plutôt que les étapes à suivre pour l'obtenir,

Il existe de nombreux langages de programmation différents, chacun ayant ses propres avantages et inconvénients. Voici quelques-uns des langages de programmation les plus courants :

- C++,
- JavaScript,
- Java,
- CSharp,
- Ruby,
- Python,
- Etc.

1.10 IDE (Integrated Development Environment)

Un IDE (Integrated Development Environment) est un environnement de développement intégré, qui fournit un ensemble d'outils et de fonctionnalités pour faciliter le développement de logiciels. Un IDE est un logiciel qui regroupe différents outils, tels qu'un éditeur de code, un compilateur, un débogueur, un gestionnaire de version, un outil de test et d'autres fonctionnalités.

Un IDE permet aux développeurs de travailler efficacement en offrant des fonctionnalités telles que :

- Coloration syntaxique : qui met en évidence les différents éléments du code pour améliorer la lisibilité.
- Complétion automatique : qui suggère des mots clés, des noms de variables, des fonctions et des méthodes pour accélérer la saisie de code.
- Refactoring : qui permet de réorganiser le code sans modifier son comportement, pour améliorer sa lisibilité et sa maintenabilité.
- Débogage : qui permet aux développeurs de détecter et de corriger les erreurs dans le code.
- Gestion de version : qui permet aux développeurs de suivre les modifications apportées au code et de collaborer avec d'autres développeurs.
- Gestion de projet : qui permet aux développeurs de gérer les fichiers et les ressources associées à un projet, et de suivre l'avancement du développement.
- Intégration d'outils tiers : qui permet aux développeurs d'utiliser des outils tiers, tels que des outils de test, des outils de déploiement et des outils de gestion de bases de données.

Les IDE sont disponibles pour de nombreux langages de programmation, tels que Java, C++, Python, Ruby, etc. Les IDE peuvent être installés localement sur un ordinateur ou utilisés en ligne via un navigateur Web. Les IDE sont largement utilisés par les développeurs pour améliorer leur productivité et leur efficacité dans le développement de logiciels.

Chapitre 2

État de l’art

Avant de commencer à développer notre application, nous avons d’abord comparer les différentes applications existantes pour en dégager les fonctionnalités clés, nous avons ensuite identifié les fonctionnalités susceptibles de répondre aux besoins de l’Université Nouveaux Horizons et nous avons ensuite justifié nos choix de technologies et fonctionnalités.

2.1 Présentation des principales applications existantes

Nous sommes conscients qu’il en existe certainement plusieurs autres, mais nous avons choisi de nous limiter à celles mentionner ci dessous car elles sont les plus utilisées, les plus populaires et couvrent la plupart des besoins des leurs utilisateurs.

2.1.1 Moodle

a Présentation de l’application

Moodle est un système de gestion de l’apprentissage (LMS) open source, c’est-à-dire un logiciel permettant la création, la gestion, la distribution et la surveillance de cours en ligne. Il permet aux enseignants et aux formateurs de créer des cours en ligne interactifs, d’organiser des activités d’apprentissage, de communiquer avec les étudiants et de suivre leur progression.

b Fonctionnalités clés

- Gestion des cours : Moodle permet de créer des cours en ligne et d’organiser des activités d’apprentissage en utilisant des outils tels que des forums de discussion, des leçons, des wikis, des glossaires, des quiz, des devoirs, des sondages, etc.
- Gestion des utilisateurs : Moodle permet de gérer les utilisateurs, de les inscrire aux cours, de les suivre et de leur attribuer des rôles spécifiques (enseignant, étudiant, administrateur, etc.).
- Communication : Moodle offre plusieurs outils de communication, y compris les forums de discussion, les messages privés, les chats en temps réel et les webinaires.
- Suivi des progrès : Moodle permet aux enseignants de suivre les progrès des étudiants en visualisant leurs notes, leurs activités et leurs contributions sur les forums de discussion et autres outils.
- Personnalisation : Moodle offre de nombreuses options de personnalisation pour les enseignants et les administrateurs, notamment la personnalisation de l’apparence du site, la gestion des catégories de cours, la création de rapports personnalisés, etc.
- Accessibilité : Moodle est conforme aux normes d’accessibilité pour les personnes handicapées, ce qui permet à tous les utilisateurs de profiter pleinement des activités d’apprentissage en ligne.

c Inconvénients et limites

- Expertise technique requise : Moodle est un logiciel complexe dont l'installation et la configuration requièrent une certaine expertise technique. Si vous ne disposez pas des compétences nécessaires, vous devrez peut-être faire appel à un consultant ou à un développeur pour vous aider.
- Sécurité : Moodle est une cible populaire pour les pirates informatiques, il est donc important de prendre des mesures pour sécuriser votre site Moodle. Cela inclut l'utilisation de mots de passe forts, l'installation de mises à jour de sécurité, et la surveillance du site pour toute activité suspecte.
- Evolutivité : Moodle peut être adapté à un grand nombre d'utilisateurs, mais il peut être difficile de gérer un grand site Moodle. Si vous envisagez d'utiliser Moodle pour une grande organisation, vous devrez peut-être engager un consultant ou un développeur pour vous aider à gérer votre site.
- Moodle est conçu pour répondre aux besoins généraux de l'enseignement en ligne. Il n'est pas conçu pour répondre aux besoins spécifiques d'une institution.

2.1.2 PowerSchool

a Présentation de l'application

PowerSchool est un système d'information pour les établissements scolaires qui permet de gérer les informations relatives aux étudiants, aux enseignants et aux programmes académiques. Il est utilisé par plus de 45 millions d'utilisateurs dans le monde entier, notamment dans les écoles primaires, secondaires et les établissements d'enseignement supérieur.

b Fonctionnalités clés

- Gestion des inscriptions et des admissions : PowerSchool permet de gérer les inscriptions et les admissions des étudiants, y compris la collecte des informations de base, la gestion des documents requis, la communication avec les parents et les étudiants, et la planification des cours.
- Gestion des notes et des absences : PowerSchool permet aux enseignants d'enregistrer les notes et les absences des étudiants, et de partager les informations avec les parents et les étudiants.
- Gestion des emplois du temps et des calendriers académiques : PowerSchool permet de gérer les emplois du temps des enseignants et des étudiants, ainsi que les calendriers académiques et les événements scolaires.
- Gestion des frais de scolarité et des paiements : PowerSchool permet de gérer les frais de scolarité et les paiements, y compris la collecte et le suivi des paiements, la génération de factures, et la communication avec les parents et les étudiants.
- Communication avec les parents et les étudiants : PowerSchool permet de communiquer avec les parents et les étudiants via des messages personnalisés, des bulletins électroniques, des alertes automatiques, et d'autres outils de communication.
- Gestion des bibliothèques : PowerSchool permet de gérer les bibliothèques de l'établissement, y compris la gestion des prêts de livres, la gestion des retours, et la gestion des réservations de livres.
- Rapports et analyses : PowerSchool permet de générer des rapports et des analyses sur les performances académiques des étudiants, les tendances de fréquentation, les statistiques de paiement, et bien plus encore.

c Inconvénients et limites

- Coût : Le coût initial de mise en place de PowerSchool peut être élevé, en particulier pour les établissements scolaires plus petits. Les coûts peuvent également augmenter avec l'ajout de fonctionnalités supplémentaires.
- Complexité : PowerSchool est un système complexe qui peut nécessiter une formation pour les administrateurs, les enseignants et les utilisateurs finaux. Les établissements scolaires doivent également consacrer du temps et des ressources à la configuration et à la gestion du système.
- Personnalisation : Bien que PowerSchool offre de nombreuses fonctionnalités, il peut être difficile de personnaliser le système pour répondre aux besoins spécifiques de chaque établissement. Les établissements peuvent être limités dans leur capacité à personnaliser le système en fonction de leurs besoins spécifiques.
- Dépendance : Les établissements qui utilisent PowerSchool dépendent du système pour gérer les données des étudiants et de l'établissement. En cas de panne du système ou de perte de données, cela peut avoir des conséquences importantes pour l'établissement.

2.1.3 Esis Salama

a Présentation

Esis salama est une institution d'enseignement supérieure de référence ici en République Démocratique du Congo, ce qui lui vaut une mention dans ce présent travail c'est la manière dont elle gère la délibération.

b Outils utilisés

- Excel : Ce dernier est un tableur édité par Microsoft pour les systèmes d'exploitation Windows et Mac OS X. Il est principalement utilisé pour le calcul, l'analyse de données et les graphiques. Excel est l'un des programmes les plus populaires pour la gestion des données et des calculs.
- Plateforme développée en interne : Cette dernière facilite la publication des résultats des étudiants et permet aux étudiants de consulter leurs résultats et de faire des recours en ligne.

c Inconvénients et limites

- Cette façon de faire est très chronophage,
- Nécessite beaucoup de ressources humaines

2.2 conclusion

En conclusion nous retiendront que l'idéal serait de produire une application qui comblerait les lacunes des applications existantes et les étendrait mais l'idéal n'est pas toujours réalisable. Nous proposons donc une solution qui résout les gros inconvénients et limitations des applications existantes et qui est facilement extensible et modifiable.

Les apports sont :

- Le fait que cette application soit développée en interne permettra de l'adapter aux besoins de l'Université Nouveaux Horizons et de rester indépendant des éditeurs de logiciels.
- L'application sera facilement extensible et modifiable.
- L'application peut facilement s'intégrer au système existant.
- L'application permettra de réduire les coûts en temps et ressources humaines car elle fera la grande partie du travail certes sous la supervision humaine.