ANNEE ACADEMIQUE 2021-2022 2021-2022 ACADEMIC YEAR

**REPUBLIC OF CAMEROON**

*Peace – Work – Fatherland*

- . - . - . –

**UNIVERSITY OF YAOUNDÉ I**

**Faculté des Sciences**

*Department of Computer Science*

*P.O.Box 812 Yaoundé*

**REPUBLIQUE DU CAMEROUN**

*Paix – Travail – Patrie*

- . - . - . –

**UNIVERSITÉ DE YAOUNDÉ I**

**Faculté des Sciences**

*Départementd'Informatique*

*B.P. 812 Yaoundé*

**ICT201: INTRODUCTION TO SOFTWARE ENGINEERING**

**&**

**ICT205: INTRODUCTION TO PROGRAMMING IN .NET**

**THEME:**

**CAHIER D’ANALYSE ET DE CONCEPTION DU FORMULAIRE**

**DE GESTION DES ETUDIANTS**

MEMBRES DU PROJET :

* **MBENGONO AKOA Elisée Salomon……………..20T2234**
* **MBEH Gabriel Jerry Stelane……………………......20R2332**
* **MBIADA FOSSO Cédric Geordan…………………20R2313**
* **MBIDA WANG Jean Boris……………………........19k2793**
* **MBONO TEKAM Ulrich…………………………....18I2275**
* **MINKA Caleb…………………………………….....21S2771**
* **MOKOM HOPE Morel Asongwed………………...20V2391**
* **MVONDO BILOUNGA Jean Francis………………20R2070**
* **MOLUH YICHE Ousmane………………………….19M2234**
* **NJICHOU Elie ………………………………………20R2333**

*Encadreurs***: Mr. MOYOU & Mr. CATCHA**

***PLAN DE TRAVAIL***

***INTRODUCTION***

1. ***PRESENTATION DU PROJET……***
2. ***Contexte et justification***
3. ***Objectifs***
4. ***NORMES STANDARTS, OUTILS ET DETAILS***
5. ***Méthode de conception***
6. ***Outils***
7. ***Détails***
8. ***METHODE Z***
9. ***MODELES***
10. ***Modèle Conceptuel de Données (MCD)***
11. ***Modèle Logique de Données (MLD)***
12. ***Modèle Physique de Données (MPD)***
13. ***DIAGRAMME DE CAS D’UTILISATION***
14. ***DESCRIPTION TEXTUELLE DES CAS D’UTILISATION***

***CONCLUSION***

**INTRODUCTION**

Le génie logiciel est l’ensemble des méthodes, des techniques et des outils dédiés à la conception, au développement et à la maintenance des systèmes informatiques. Le C# ou C sharp est un langage de programmation orienté objet, commercialisé par microsoft depuis 2002 et destiné à développer sur la plateforme Microsoft.NET***.*** Nous avons pour tâche la création d’un formulaire de gestion des notes qui est un mécanisme destiné à enregistrer des étudiants et leurs notes puis modifier ces informations.

1. **PRESENTATION DU PROJET**
2. **Contexte et justification**

Ce projet nous est soumis dans le cadre du processus d’apprentissage académique, pendant le premier semestre de l’année académique 2021/2022, des unités d’enseignement ICT201 et ICT205 respectivement intitulés Introduction to software engineering et Introduction to Programming in .NET. Pour être désormais capable de créer des logiciels de qualité, le projet porte sur le développent d’un logiciel de gestion des notes.

1. **Objectifs**

Le but de cette manœuvre sera entre autre, de réaliser le projet définit dans les délais et selon l’attente du client, tout en réalisant des maquettes et développer la couche interface utilisateur (UI) du formulaire gestion des notes.

1. **NORMES STANDARTS, OUTILS ET DELAIS**
2. **Méthode de conception**

La méthode de conception utilisée est Merise.

* La démarche de développement utilisée dans ce projet de gestion des notes est incrémentale et itérative. Le cycle de vie en « V » a été appliqué. Cette démarche permet de spécifier en parallèle les contraintes et l’architecture :
* Fonctionnelles
* Techniques.

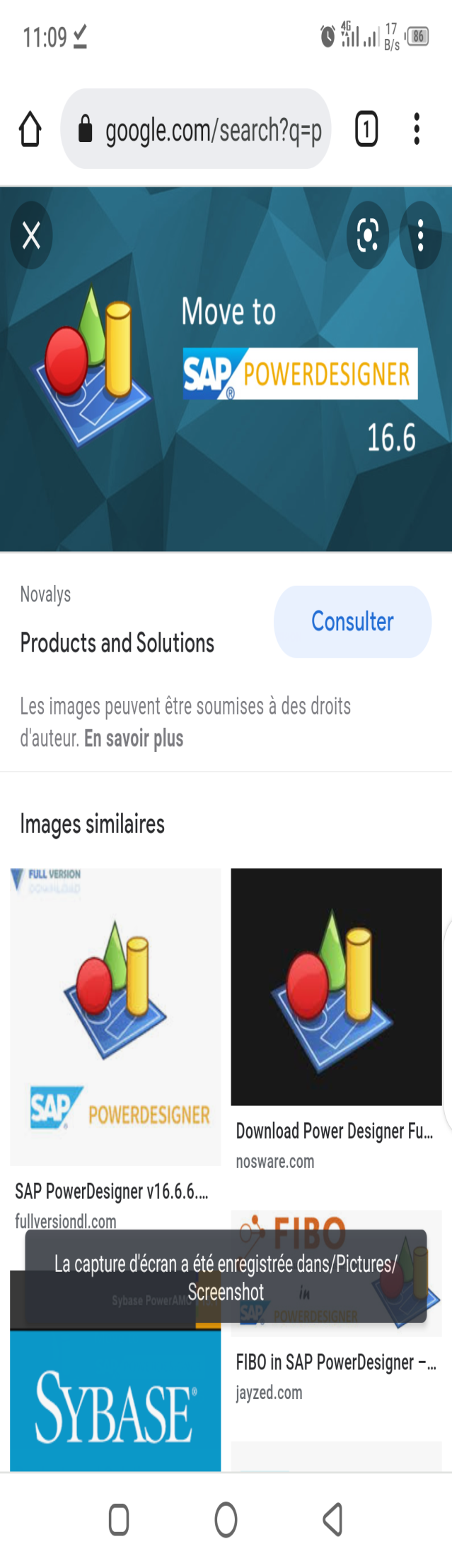
1. **Environnent et outils de développement**

* **Microsoft Visual Studio 2010 ou Visual Studio 2012**



Microsoft Visual studio est une suite de logiciel de développement pour Windows et Mac OS conçu par Microsoft. C’est un ensemble complet d’outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles.

* **Power designer AMC**

Power designer est un logiciel de conception crée par la société SAP, qui permet de modéliser les traitements informatiques et leurs bases de données associées.

1. **Délais**

Il nous faudra du temps et de l’effort pour ce travail et sa réussite. À partie du tableau ci-dessous nous évaluerons en combien de temps nous pouvons développer ce logiciel :

|  |  |
| --- | --- |
| Tâches | Période requise |
| Recueil des besoins | Cela a été fait en 3 jours |
| Analyse des besoins | Fait en 21 jours |
| Conception globale | Sera fait en 21 jours |
| Conception détaillée | Sera fait en 21 jours |
| L’implémentation | Sera fait en 7 jours |
| Les tests d’intégration | Se feront en 7 jours |

Les délais seront respectés pour chaque étape du projet.

1. **METHODE Z**

* **Etudiant**

|  |
| --- |
| Etudiants002D : P[étudiants]  Filière : [étudiants] —>string  Niveau : [étudiants] —>string  Prénom : [étudiants]🡪string  Nom : [étudiants]🡪string  Sexe :[étudiants] :string  DateNaissance : [étudiants] —>Date  LieuNaissance : [étudiants] —>string  Matricue : [étudiants]🡪string |
| Dom prénom = dom nom= dom numero = étudiants |

* **Δétudiant**

|  |
| --- |
| Etudiants  Prénom’: [etudiants]🡪string  Nom’ : [etudiants]🡪string  Filière’ : [étudiants] —>string  Niveau’ : [étudiants] —>string  Sexe’ :[étudiants] :string  DateNaissance’ : [étudiants] —>Date  LieuNaissance’ : [étudiants] —>string  Matricule’ : [etudiants]🡪string |
| Dom prénom’= dom nom’=dom numero’=etudiants |

* **AjouterE**

|  |
| --- |
| Δ étudiants  n?: string  p?: string  mat?: String  e?: [etudiants] |
| nom’ = nom U {e? 🡪n ?}  prenom’ = prenom U {e? 🡪p ?}  Matricule’ = Matricule U {e? 🡪mat ?} |

* **ChangerNom**

|  |
| --- |
| Δétudiant  e ?: [etudiants]  n ?: string |
| Nom’= nom + {e ? 🡪n ?} |

* **ChangerPre**

|  |
| --- |
| Δétudiant  e ?: [etudiants]  p ?: string |
| prenom’= prenom + {e ? 🡪p ?} |

* **ChangerMat**

|  |
| --- |
| Δétudiant  e ?: [etudiants]  mat?: String |
| matricule’= matricule + {e ? 🡪mat?} |

* **ChangerFil**

|  |
| --- |
| Δétudiant  e ?: [etudiants]  f?: string |
| Filière’= Filière + {e ? 🡪f?} |

* **ChangerSexe**

|  |
| --- |
| Δétudiant  e ?: [etudiants]  s?: String |
| Sexe’= sexe + {e ? 🡪s?} |

* **ChangerNiv**

|  |
| --- |
| Δétudiant  e ?: [etudiants]  niv?: String |
| niveau’= niveau + {e ? 🡪niv?} |

* **ChangerDateN**

|  |
| --- |
| Δétudiant  e ?: [etudiants]  daten?: String |
| DateNaissance’= DateNaissance + {e ? 🡪daten?} |

* **changerLieuN**

|  |
| --- |
| Δétudiant  e ?: [etudiants]  lieun?: String |
| LieuNaissance’= LieuNaissance + {e ? 🡪lieun?} |

1. **MODELES**
2. **Modèle Conceptuel de Données (MCD)**
3. **Modèle Logique de données (MLD)**
4. **Modèle Physique de Données (MPD)**
5. **DIAGRAMME DE CAS D’UTILISATION**
6. **DESCRIPTIONS TEXTUELLE DES CAS D’UTILISATION**

* Cas d’utilisation **: Se connecter**

|  |  |
| --- | --- |
| **Objectif** | Permet à l’administrateur de se connecter à l’application |
| **Acteur** | Administrateur |
| **Pré condition** | Il doit être administrateur |
| **Post condition** | Fermer le programme |
| **Scénario normal** | * Entrer son adresse Email, son prénom et son mot de passe ; * Cliquer sur connecter * Ouverture de la page principale |
| **Scénario alternatif** | * L’administrateur entre un faut mot de passe * Un message d’erreur s’il n’est pas enregistré dans la base de donnée comme enseignant |

* Cas d’utilisation : **Ajouter étudiant**

|  |  |
| --- | --- |
| **Objectif** | Permet d’ajouter un étudiant à la filière |
| **Acteur** | L’administrateur |
| **Pré condition** | Etre connecté en tant qu’administrateur |
| **Post condition** | Cliquer sur le bouton **ajouter** |
| **Scénario normal** | * Insérer le nom * Insérer le prénom * Insérer le sexe * Insérer la date de naissance * Insérer le lieu de naissance * Insérer la filière * Insérer le niveau * Insérer le matricule |
| **Scénario alternatif** | * L’administrateur doit ajouter un étudiant * Message d’erreur pour matricule incorrect * Message d’erreur pour nom incorrect * Message d’erreur pour champ vide |

* Cas d’utilisation : **Supprimer étudiant**

|  |  |
| --- | --- |
| **Objectif** | Supprimer un étudiant de la base de données |
| **Acteur** | L’administrateur |
| **Pré condition** | * Etre connecté à l’application * L’étudiant à supprimer doit exister au préalable |
| **Post condition** | Fermer la page |
| **Scénario normal** | * Entrer le matricule de l’étudiant à supprimer * Appuyer sur supprimer |
| **Scénario alternatif** | * L’utilisateur entre un faux matricule * Un message d’erreur est affiché * Cliquer sur le bouton supprimer * Message d’erreur sur l’absence du matricule dans le champ |

* Cas d’utilisation : **Modifier informations**

|  |  |
| --- | --- |
| **Objectif** | Changer une donnée sur un étudiant |
| **Acteur** | L’administrateur |
| **Pré condition** | * Etre connecté en tant qu’Administrateur * L’étudiant dont les informations doivent être supprimées doit exister dans la base de données |
| **Post condition** | * Cliquer sur le bouton modifier * Fermer l’application |
| **Scénario normal** | * Se connecter * Sélectionner le matricule de l’étudiant concerné * Changer les informations * Cliquer sur **MODIFIER** |
| **Scénario alternatif** | * Sélectionner le matricule * Message d’alerte pour champ incorrect |

* Cas d’utilisation : **ajouter une note**

|  |  |
| --- | --- |
| **Objectif** | Ajouter une ou plusieurs notes dans les champs d’un étudiant |
| **Acteur** | L’administrateur |
| **Pré condition** | * Etre connecté en tant qu’Administrateur * L’étudiant doit exister dans la base de données |
| **Post condition** | * Cliquer sur le bouton **Ajouter** * Fermer l’application |
| **Scénario normal** | * Se connecter * Sélectionner le matricule de l’étudiant concerné * ajouter les informations * cliquer sur **Ajouter** |
| **Scénario alternatif** | * Sélectionner le matricule * Message d’alerte pour champ incorrect |

* Cas d’utilisation : **Modifier une note**

|  |  |
| --- | --- |
| **Objectif** | Changer la note d’un étudiant |
| **Acteur** | L’administrateur |
| **Pré condition** | * Etre connecté en tant qu’Administrateur * L’étudiant dont la note doit être modifié doit exister dans la base de données |
| **Post condition** | * Cliquer sur le bouton modifier * Fermer l’application |
| **Scénario normal** | * Se connecter * Sélectionner le matricule de l’étudiant concerné * Changer la note * Cliquer sur **MODIFIER** |
| **Scénario alternatif** | * Sélectionner le matricule * Message d’alerte pour champ incorrect |

* Cas d’utilisation : **Consulter notes**

|  |  |
| --- | --- |
| **Objectif** | Consulter les différentes notes |
| **Acteur** | L’administrateur |
| **Pré condition** | * Etre connecté en tant qu’Administrateur * Les étudiants dont les informations doivent être consultées doivent exister dans la base de données |
| **Post condition** | * Cliquer sur le bouton modifier * Fermer l’application |
| **Scénario normal** | * Se connecter * Cliquer sur consultation * Choisir l’unité d’enseignement |
| **Scénario alternatif** | * Cliquer sur consultation * Message d’alerte pour champ incorrect |

* Cas d’utilisation : **Calculer la moyenne**

|  |  |
| --- | --- |
| **Objectif** | Calculer la moyenne d’un étudiant |
| **Acteur** | L’administrateur |
| **Pré condition** | * Etre connecté en tant qu’Administrateur * L’étudiant dont la moyenne doit être calculé doit exister dans la base de données * L’étudiant doit avoir des notes valides |
| **Post condition** | * Fermer l’application |
| **Scénario normal** | * Cliquer sur moyenne |
| **Scénario alternatif** | * Cliquer sur moyenne * Message d’erreur si les notes sont invalides |

**CONCLUSION**

Il était question pour nous de réaliser un cahier d’analyse et de conception afin de mener à bien la création d’un logiciel de gestion des notes. Ce qui nous a emmené à établir une méthode Z, un modèle conceptuel, logique et physique en utilisant merise sur le logiciel Power Designer. Aussi, nous avons produit un diagramme de cas d’utilisation suivi de la description détaillée.