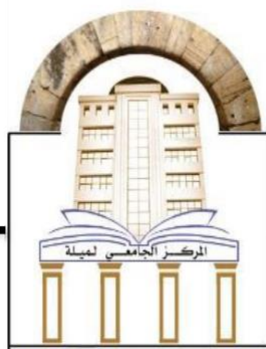


الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique



Centre Universitaire Abdelhafid Boussouf – Mila
Institut des Mathématiques et informatique
Département d'Informatique

Mémoire préparé pour obtenir le diplôme de Master en Informatique

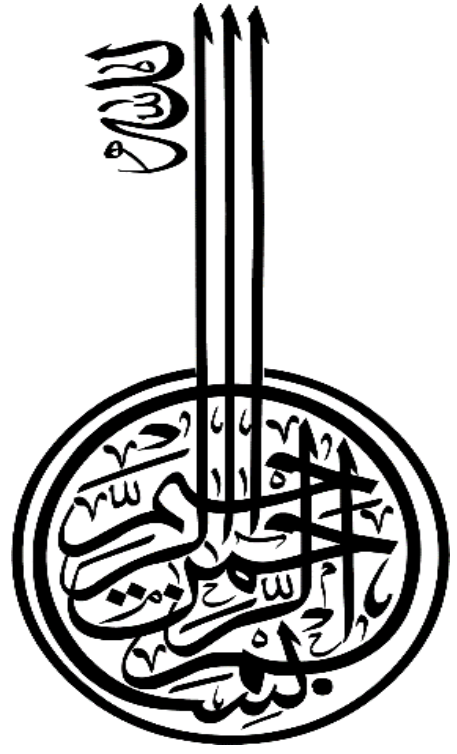
Option : Sciences et Technologies de l'Information et de la
Communication (STIC)

Thème :
**Migration d'une application open source vers les Lignes
de Produits Logiciels.**

Réalisé par :
Boualita Nada
Hebbachi Asma

Soutenue devant le jury :
Président : Dr. Guettiche Mourad
Encadré par Dr. Meghzili Said
Examineur : Dr. Aouag Mouna

Année Universitaire : 2023/2024



Remerciements

On remercie Dieu le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire. Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans l'aide et l'encadrement de **Dr.Said Meghzili** on le remercie pour la qualité de son encadrement exceptionnel, pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.

Nos remerciement s'adresse également à tout nos professeurs pour leur générosités et la grand patience dont ils ont su faire preuve malgré leurs charge académiques et professionnelles.

En conclusion, nous remercions toujours Dieu à tout moment et à toutes les étapes vie.

Merci et nous vous souhaitons tout le meilleur avec nos meilleurs vœux.

اهداء

" بسم الله خالقي و ميسر امري وعصمة امري
لك كل الحمد و الامتنان "

اهدي هذا النجاح لنفسى اولا ثم الى كل من سعى معي لإتمام هذه المسيرة
دمتم لي سنداً لا عمر له ...

من كلكه الله بالهيبة و الوقار... الى من احمل اسمه بكل فخر ... الى من حصد الاشواك من دربي ليمهد لي طريق العلم
بعد فضل الله ما انا فيه يعود ل ابي "رحمه الله" الرجل الذي سعى طوال حياته لكي نكون افضل منه
ابي الغالي

الى ملاكي في لحياة الى معنى الحب و الى معنى الحنان و التفاتي .. الى بسمه الحياة و سر الوجود
الى من كان دعاؤها سر نجاحي و حنانها بلسم جراحي
التي كانت لي الام و الاخت و الصديقة داعمي الاول ووجهتي التي استمد منها القوة
التي لن اوفيتها حقها مهما فعلت حبيبي ورفيقة دربي
امي الحبيبة

الى مصدر قوتي .. الداعمين الساندين .. ارضي الصلبة و جداري المتين .. الى من راهنو على نجاتي
و يذكروني بمدى قوتي و استطاعتي الذين لا يحبطوني و يؤمنوا بشجاعتى مهما ضعفت و ارتخيت واقفين خلفي
مثل ظلا مهما كثرت تخبطاتي .. الى من بذلوا جهدا في مساعدتي و كانوا عوننا و سندا الى الشموع التي تنير طريقي
اخواني و اخواتي

الى سندي الثاني في الحياة الى من كان عوضا لي و ملا فراغ حياتي .. الى ابي الثاني الذي
كان داعمي و سندي في غياب ابي شكرا لك
عمي الغالي

و لا انسى صديقاتي و اختاي رفيقات الروح الذين شاركوني خطوات الطريق بحلوها ومرها الى من هونوا تعب الطريق
الى من شجعوني على المثابرة و اكمال المسيرة الى رفقاء السنين ممتنة لكم
احلام و ايناس

الى من بداء معها مسيرتي و انهيناها معا الى من كانت لي بمثابة الاخت و عشت معها كل تفاصيل مسيرتي
الى تقاسمت معها كل ايامي الجامعية و تعبها و سنينها ها نحن ذا ننهي المشوار معا
شكرا لأنك كنتي معي في هذا الطريق و اتمنا ان تكوني في اعلى المراتب
اسماء

لله الشكر كله ان وفقني لهذه اللحظة .. فالحمد لله رب العالمين و الصلاة و السلام على نبيه الكريم .

خريجتكم : ندى

"وَ آخِرُ دَعْوَاهُمْ أَنِ الْحَمْدُ لِلَّهِ رَبِّ الْعَالَمِينَ"

”

من قال أنا لها ... "نالها"

و أنا لها و إن أبت رَغَمًا عنها أتيت بها .

نلتها و عانقت اليوم مجدًا عظيمًا لم يكن الحلم قريبًا ولا الطريق سهلاً و لكن ... وصلت .
الحمد لله حباً و شكرًا و إمتنانا الحمد لله الذي بفضلها ها أنا اليوم أنظر الى حلم طال إنتظاره و قد أصبح واقعًا أفتخر به.

أهدي بكل حبِّ ثمرة تخرجي الى ...

الى نفسي العظيمة التي تحملت كل العثرات و أكملت رغم الصعوبات
الى من أحمل إسمه بافتخار الى من كان نور دربي و من علمني دون انتظار أبي الغالي أدامك الله لي .
الى أملي في الحياة و قرة عيني الى من كان دعائها سر نجاحي أمي الحبيبة حفظك الله و رعاك .
الى من قيل فيهم : "سنشد عضدك بأخيك" من دامت لي أياديهم وقت ضعفي الى ضلعي الثابت و
أمان قلبي .. إخوتي (نجلاء، صفاء).

الى من كاتفتني و نحن نشق الطريق معًا نحو النجاح في مسيرتنا العلمية ...رفيقة دربي : ندى.
الى كل من أعطاني يد العون من قريب أو من بعيد و ساعدني في هذا المشوار.

هباشي أسماء

تلخيص

تعد إعادة استخدام البرامج مفهومًا أساسيًا ومستخدمًا على نطاق واسع في تطوير أنظمة الكمبيوتر LPL. (خطوط إنتاج البرمجيات) هو نموذج هدفه الرئيسي هو تطوير أنظمة البرمجيات التي تشكل جزءًا من نفس المجال. بالإضافة إلى ذلك، تهدف LPLs إلى تلبية متطلبات الأسواق ذات النمو المرتفع من خلال تطوير أنظمة برمجية من موارد قابلة لإعادة الاستخدام بدلاً من البدء من الصفر. في هذا العمل، نقترح طريقة لترحيل تطبيق إدارة مشروع مفتوح المصدر إلى خطوط إنتاج البرامج باستخدام منصة Mobioos Forge. وبالإضافة إلى ذلك، فإننا نقدم امتدادًا لهذا التطبيق عن طريق إضافة وحدة جديدة لإدارة المشروع.

الكلمات المفتاحية: إدارة المشاريع، التباين، خطوط إنتاج البرمجيات، Mobioos Forge

Résumé

La réutilisation de logiciels est un concept essentiel et largement utilisé dans le développement des systèmes informatiques. Les LPL (lignes de Produits Logiciels) sont un paradigme dont l'objectif principal est de développer des systèmes logiciels faisant partie du même domaine. En plus, les LPL visent à répondre aux demandes des marchés à forte croissance en développant des systèmes logiciels à partir de ressources réutilisables plutôt qu'en partant de zéro. Dans ce travail, nous proposons une approche de migration d'une application open source sur la gestion des projets vers les Lignes de Produits Logiciels en utilisant la plateforme Mobioos Forge. En plus, nous proposons une extension pour cette application en ajoutant un nouveau module de gestion des projets.

Mots clés : Gestion de projets, Variabilité, lignes de Produits Logiciels, Mobioos Forge.

Abstract

Software reuse is an essential and widely used concept in the development of computer systems. LPL (Software Product Lines) is a paradigm whose main objective is to develop software systems that are part of the same domain. Additionally, LPLs aim to meet the demands of high-growth markets by developing software systems from reusable resources rather than starting from scratch. In this work, we propose an approach for migrating an open source project management application to Software Product Lines using the Mobioos Forge platform. In addition, we offer an extension for this application by adding a new project management module.

Keywords: Project management, Variability, Software Product lines, Mobioos Forge.

Table des Matières

Remerciements	i
Dédicaces	ii
Abstract	iv
Résumé	iv
Table des Matières	vii
Table des Figures	x
Liste des Tableaux	xii
Introduction générale	1
1. Présentation de l'organisme d'accueil	6
1.1. Introduction	5
1.2. Présentation d'Algérie Télécom	5
1.3. Apparition	6
1.4. Présentation de la direction opérationnelle de Mila	6
1.4.1. La sous-direction Technique (SD TECHNIQUE).....	7
1.4.1.1. Département Planification et Suivi	7
1.4.1.2. Département Réseau Accès	8
1.4.1.3. Département Réseau Transport	8
1.4.1.4. Centre Opérationnel Énergie et Environnement	9
1.5. Les Volets d'Activité Essentiels	9
1.5.1. Pose de Canalisations	9
1.5.2. Pose de Fibre Optique.....	10
1.5.3. Pose d'ODN (Optical Distribution Network).....	10
1.5.4. Pose de CDN (Content Delivery Network)	10
1.5.5. Pose de Raccordement des Abonnés	10
1.6. La gestion des projets dans Algérie Télécom	11
1.7. Conclusion.....	13
2. Ligne de produit logiciel	15
2.1. Introduction	15
2.2. Définition	15

Table des matières

2.3. Environnement pour l'ingénierie des lignes de produits logiciels	15
2.3.1. Ingénierie de domaine.....	16
2.3.1.1. Analyse de domaine	16
2.3.1.2. Modélisation de domaine.....	16
2.3.1.3. Implantation de domaine	17
2.3.1.4. Test de domaine.....	17
2.3.2. Ingénierie d'application	17
2.3.2.1. Exigences d'application	17
2.3.2.2. La modélisation de l'application	17
2.3.2.3. Implantation d'application	17
2.3.2.4. Test de l'application	17
2.4. Exemples des lignes de produits logiciels.....	19
2.4.1. Sandwicheries	18
2.4.2. L'application de Bank Web.....	19
2.4.3. Téléphones mobiles Nokia	20
2.5. Apports de l'utilisation des lignes de produits logiciels	21
2.5.1. Réutilisation accrue du code	21
2.5.2. Meilleure qualité logicielle.....	21
2.5.3. Rapidité sur le marché	21
2.5.4. Réduction de l'effort de maintenance	22
2.5.5. L'amélioration de l'estimation des coûts	22
2.6. Conclusion	23
3. Migration de système vers les lignes de produits logiciels	25
3.1. Introduction	25
3.2. Plateforme Mobioos forge	25
3.2.1. Modèle de fonctionnalités	26
3.2.2. Mappage des fonctionnalités (feature mapping)	26
3.2.3. Personnalisation	28
3.3. La génération des lignes de produit logicielles à partir de la gestion des projets	29
3.3.1. Modelé de fonctionnalité	29
3.3.2. Les Marqueurs et cartes	31
3.3.3. Les variantes dérivées.....	33
3.4. Conclusion	36
4. Extension du système.....	38
4.1. Introduction	38
4.2. Architecture microservices	38
4.2.1. Définition	38

Table des matières

4.2.2. Exemple d'une architecture Microservice	38
4.2.3. Les avantages des microservices	39
4.3. Le choix de la méthodologie	39
4.3.1. Rappel sur le processus unifié	40
4.3.1.1. Définition du processus unifié	40
4.3.1.2. Les caractéristiques du processus unifié	40
4.3.2. Rappel sur le langage UML	41
4.3.2.1. Définition d'UML	41
4.3.2.2. Définition d'un diagramme	41
4.4. Conception du système	42
4.4.1. La gestion des projets en Algérie	42
4.4.2. Architecture logicielle du système	44
4.4.2.1. Ajout d'un nouveau microservice au système	44
4.4.3. Diagramme de cas d'utilisation	45
4.4.4. Diagramme de classes	46
4.4.5. Le passage du diagramme de classe au modèle relationnel	47
4.5. Implémentation du système	47
4.5.1. Présentation des outils et de langages utilisés	47
4.5.2. Description du système	50
4.5.3. Les interfaces	51
4.5.3.1. L'Interface de l'application de base avant modification	51
4.5.3.2. L'Interface de L'application après modification	52
4.6. Conclusion	54
Conclusion générale.....	56
Bibliographie.....	57

Table des Figures

Figure 1.1 Algérie Telecom	6
Figure 1.2 Organigramme de La sous-direction Technique.....	7
Figure 1.3 Organigramme de Le Département Planification et Suivi.....	7
Figure 1.4 Organigramme de Le Département Réseau Accès	8
Figure 1.5 Organigramme de Le Département Réseau Transport.....	8
Figure 1.6 Organigramme de Le Centre Opérationnel Énergie et Environnement	9
Figure 1.7 Organigramme de La gestion de projet chez Algérie Télécom.....	11
Figure 2.1 Ingénierie de domaine et ingénierie d'application.....	16
Figure 2.2 Le configurateur de Sandwich	18
Figure 2.3 Bank Web	19
Figure 2.4 Téléphones mobiles Nokia	20
Figure 2.5 temps de mise sur le marché avec et sans l'ingénierie de lignes de produits	22
Figure 3.1 Le processus de Mobioos Forge pour les lignes de produits logiciels	25
Figure 3.2 Modèle de fonctionnalités de Notepad créé avec MF	26
Figure 3.3 Exemples de marqueurs et de cartes dans MF	27
Figure 3.4 Mappage de fonctionnalité	28
Figure 3.5 L'éditeur de configuration dans MF	29
Figure 3.6 Le modèle de fonctionnalités résultant de gestion des projets.	30
Figure 3.7 Quantité de marqueurs et de cartes dans Project Management SPL	31
Figure 3.8 Éditeur de personnalisation de la variante test 1	34
Figure 3.9 L'Interface de résultat de la variante test 1	34
Figure 3.10 Éditeur de personnalisation de la variante test 2	35
Figure 3.11 L'Interface de résultat de la variante test 2	35
Figure 3.12 Éditeur de personnalisation de la variante test 3	36
Figure 3.13 L'Interface de résultat de la variante test 3	36
Figure 4.1 Découpage d'une application en petites services.....	38
Figure 4.2 Exemple d'une architecture microservices.....	39
Figure 4.3 Déroulement de processus unifié	41
Figure 4.4 Exemple d'une décision de clôture d'un projet	42
Figure 4.5 Décision d'intégration d'un projet dans la nouvelle loi.....	43
Figure 4.6 Diagramme de cas d'utilisation de microservice gestion des projets actualisés	45
Figure 4.7 Diagramme de classes de microservice gestion des projets actualisées	46
Figure 4.8 Logo du framework spring boot.....	47
Figure 4.9 Logo du langage de programmation java	48
Figure 4.10 Logo de Framework Angular	48

Table des figures

Figure 4.11 Logo de Framework Mobioos Forge	48
Figure 4.12 Logo de Framework Visual Studio Code	49
Figure 4.13 Logo de Framework Node.js	49
Figure 4.14 Logo du gestionnaire des packages NPM	49
Figure 4.15 Logo du système de la base de données	50
Figure 4.16 Architecture de système	51
Figure 4.17 L'Interface de l'application de base la Page d'accueil	51
Figure 4.18 L'Interface de l'application de base d'ajouter projet	52
Figure 4.19 L'Interface de L'application après modification la page d'accueil	52
Figure 4.20 L'Interface de L'application après modification d'ajouter projet.....	53
Figure 4.21 L'Interface de détail d'un projet.....	53
Figure 4.22 L'Interface la liste des projets	54

Liste des Tableaux

Tableau 3.1 Métriques concernant l'activité de mappage des fonctionnalités32
Tableau 3.2 Métriques concernant quelques variantes générées.....33

Introduction générale

Introduction générale

Dans la dernière décennie, plusieurs entreprises ont commencé un processus de migration de leurs applications en les Lignes de Produits Logiciels (LPL). Ce processus vise à tirer parti des avantages offerts par les LPL, tels que la réutilisabilité, la flexibilité et l'efficacité dans le développement de logiciels. En plus, les LPL visent à répondre aux demandes des marchés à forte croissance en développant des systèmes logiciels à partir de ressources réutilisables plutôt qu'en partant de zéro. En outre, dans le secteur industriel, de grandes sociétés telles que Renault, Toshiba, BMW et Nokia ont opté pour les LPL afin d'améliorer leurs services.

Dans ce contexte, nous avons choisi une application open source dans le domaine de la gestion des projets comme étude de cas et l'entreprise Algérie Télécom comme lieu de notre stage. Après notre étude, nous avons remarqué qu'il y a un changement dans la réglementation de la gestion des projets. Plus précisément, le législateur algérien a promulgué la loi organique n° 18-15 relative aux lois de finances, qui vise à réformer le cadre budgétaire et comptable et à opérer une transformation majeure dans la gestion des deniers publics, à travers la promulgation d'un arsenal de textes législatifs et réglementaires.

Dans ce travail, nous proposons une approche de migration d'une application open source sur la gestion des projets vers les Lignes de Produits Logiciels en utilisant la plateforme *Mobioos Forge*. En plus, nous proposons une extension pour cette application en ajoutant un nouveau module de gestion des projets conformément à Loi organique 18-15 qui est destiné aux Institutions publiques de nature managériale et institutions similaires. Cette extension permettra aux responsables de contrôler et de superviser toutes les opérations et les aide également à prendre les décisions appropriées.

Ce mémoire est organisé comme suit. Il comprend quatre (04) chapitres:

Le premier chapitre offre une vue d'ensemble de l'institution où nous avons effectué un stage pratique, incluant son historique, ses missions.

Le deuxième chapitre présente les concepts fondamentaux des lignes de produits logiciels, leur importance, et comment elles sont structurées pour maximiser la réutilisabilité et l'efficacité. Ce chapitre met en lumière les aspects théoriques et pratiques des LPL.

Le troisième chapitre détaille le processus de migration d'une application open source sur la gestion des projets vers les Lignes de Produits Logiciels et présente des étapes clés tout en fournissant des statistiques détaillées sur cette transition.

Introduction générale

Le quatrième chapitre présente l'architecture microservices adoptée pour notre projet actualisé, la conception du système à travers divers diagrammes comme ceux de cas d'utilisation, de classe, ainsi que le modèle relationnel. En plus, Il met en lumière les outils, les langages de programmation, l'environnement de développement utilisés, et présente quelques interfaces clés de l'application.

Chapitre 01 : Présentation de l'organisme d'accueil

Chapitre 01: présentation de l'organisme d'accueil

1. Présentation de l'organisme d'accueil :

1.1. Introduction :

Au cours de notre stage chez Algérie Télécom, nous avons exploré la gestion de projet dans le contexte des télécommunications en Algérie. Cette expérience nous a permis de comprendre les défis spécifiques et les pratiques de gestion de projet au sein de l'organisation.

Dans ce chapitre, D'abord, nous présentons Algérie Télécom, leurs directions, départements. Ensuite, nous décrivons les volets d'Activité Essentiels d'Algérie télécom.

Enfin, nous expliquons le processus de la gestion des projets dans Algérie télécom.

1.2. Présentation d'Algérie Télécom :

Algérie Telecom, est une société par actions à capitaux publics opérant sur le marché des réseaux et services de communications électroniques.

Sa naissance a été consacrée par la loi 2000/03 du 5 août 2000, relative à la restructuration du secteur des Postes et Télécommunications, qui sépare notamment les activités Postales de celles des Télécommunications.

Algérie Telecom est donc régie par cette loi qui lui confère le statut d'une entreprise publique économique sous la forme juridique d'une société par actions SPA.

Entrée officiellement en activité à partir du 1er janvier 2003, elle s'engage dans le monde des Technologies de l'Information et de la Communication avec trois objectifs:

- ❖ Rentabilité
- ❖ Efficacité
- ❖ Qualité de service

Algérie Telecom est leader sur le marché Algérien des télécommunications qui connaît une forte croissance. Offrant une gamme complète de services de voix et de données aux clients résidentiels et professionnels.

Cette position s'est construite par une politique d'innovation forte adaptée aux attentes des clients et orientée vers les nouveaux usages.

Chapitre 01: présentation de l'organisme d'accueil

Son ambition est d'avoir un niveau élevé de performance technique, économique, et sociale pour se maintenir durablement leader dans son domaine, dans un environnement devenu concurrentiel.

Son souci consiste, aussi, à préserver et développer sa dimension internationale et participer à la promotion de la société de l'information en Algérie.

1.3. Apparition :

Il est situé à centre-ville de la wilaya de Mila Rue 1 Novembre 1945 vers route Zeghaia



Figure 1.1 Algérie Telecom

1.4. Présentation de la direction opérationnelle de Mila :

La Direction Opérationnelle (DO) est constituée de trois sous-directions, un service de la sûreté, et un chargé de la communication :

- ❖ Sous-direction Technique.
- ❖ Sous-direction Commerciale.
- ❖ Sous-direction Fonctions Support.

Chapitre 01: présentation de l'organisme d'accueil

1.4.1. La sous-direction Technique (SD TECHNIQUE) :

La Sous-direction Technique est constituée de trois départements et d'un centre énergie et environnement :

- Département Planification et Suivi (Dpt.PS)
- Département Réseau Accès (Dpt.RA)
- Département Réseau Transport (Dpt.RT)
- Centre Technique Énergie et Environnement (CTEE)

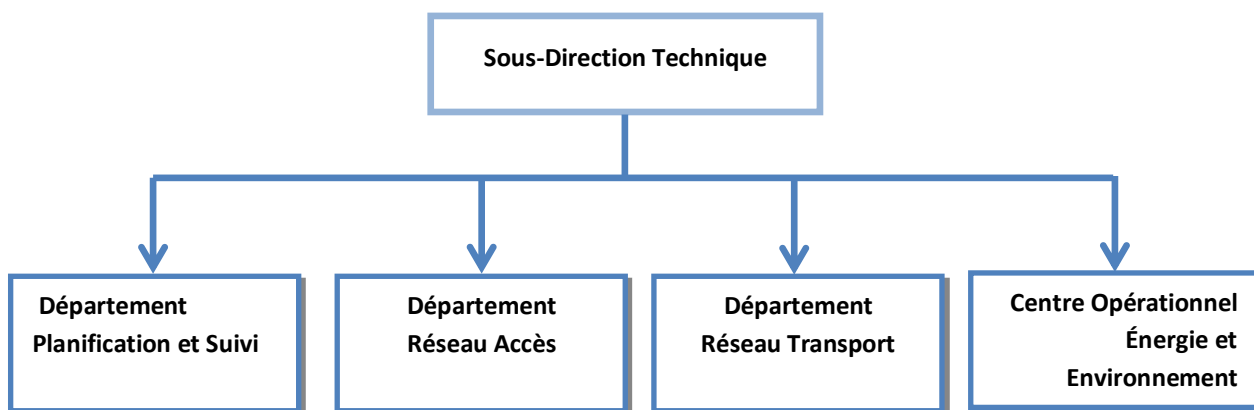


Figure 1.2 Organigramme de La sous-direction Technique

1.4.1.1. Département Planification et Suivi :

Le Département Planification et Suivi est composé TROIS (3) Services :

- Service Développement Réseau et Implémentation
- Service Étude et Ingénierie
- Service Base de Données Réseau et Indicateurs (QoS)

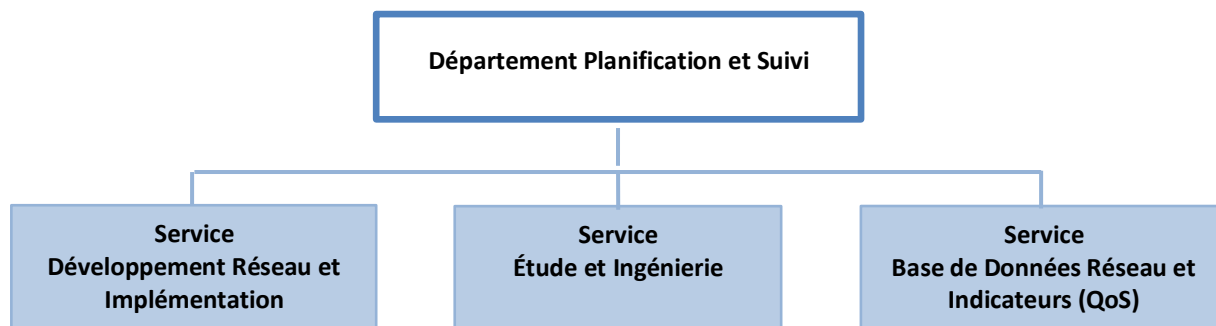


Figure 1.3 Organigramme de Le Département Planification et Suivi

Chapitre 01: présentation de l'organisme d'accueil

1.4.1.2. Département Réseau Accès :

Le Département Réseau Accès est composé TROIS (3) Services :

- Service Réseau RADIO
- Service Équipements Accès
- Service Réseau Urbain

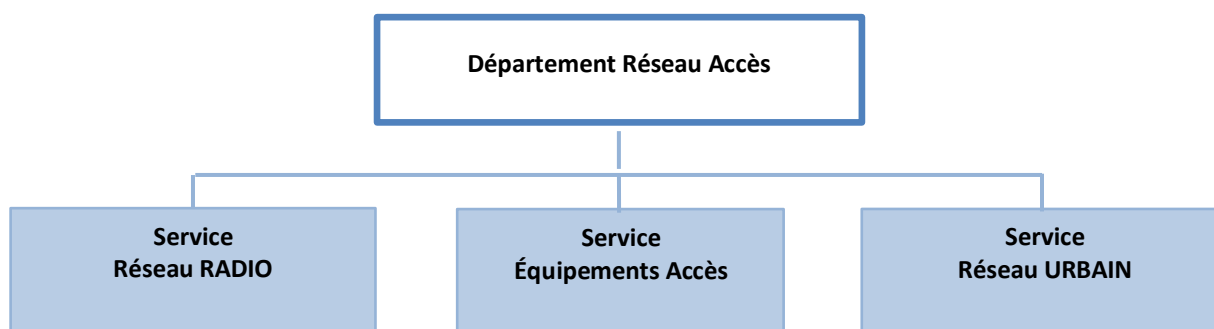


Figure 1.4 Organigramme de Le Département Réseau Accès

1.4.1.3. Département Réseau Transport :

Le Département Réseau Transport est composé TROIS (3) Services :

- Service Réseau Radio FH
- Service Équipements Transport
- Service Réseau Câbles FO

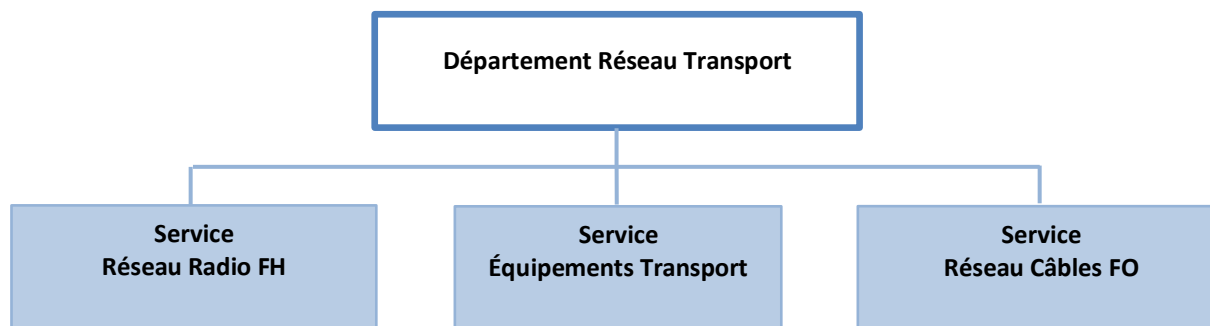


Figure 1.5 Organigramme de Le Département Réseau Transport

1.4.1.4. Centre Opérationnel Énergie et Environnement :

Le Centre Opérationnel Énergie et Environnement est composé de TROIS (3) Cellules :

- Cellule Énergie
- Cellule Équipements
- Cellule Environnement

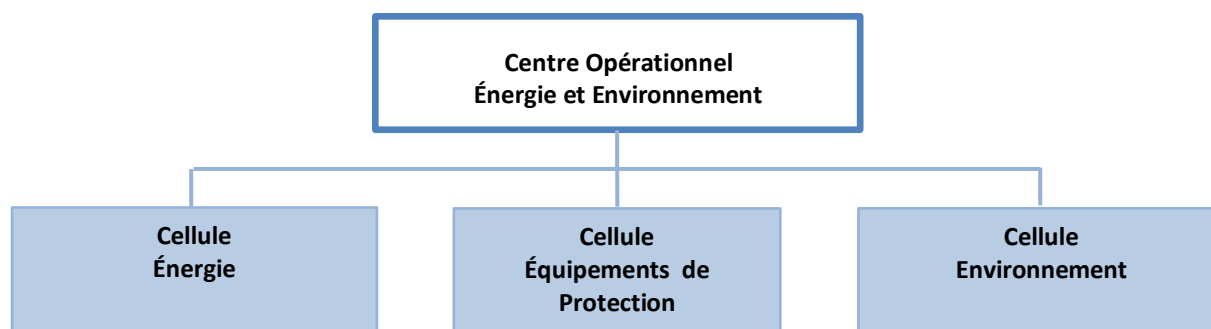


Figure 1.6 Organigramme de Le Centre Opérationnel Énergie et Environnement

1.5. Les Volets d'Activité Essentiels:

Algérie Telecom opère dans plusieurs domaines:

- la pose de canalisation
- la pose de fibre optique, la pose d'ODN (Optical Distribution Network)
- la pose de CDN (Content Delivery Network)
- la pose de raccordement des abonnés pour offrir des services de télécommunication avancés.

1.5.1. Pose de Canalisation :

Cela implique la mise en place d'infrastructures souterraines ou aériennes pour abriter les câbles de communication. Ces canalisations servent de chemins physiques pour les câbles, assurant une protection et une organisation adéquate.

1.5.2. Pose de Fibre Optique :

L'installation de câbles à fibre optique, qui transmettent des données sous forme de signaux lumineux. Cette technologie permet des débits de données élevés et une meilleure qualité de transmission par rapport aux câbles traditionnels en cuivre.

1.5.3. Pose d'ODN (Optical Distribution Network) :

Le réseau de distribution optique est mis en place pour gérer la répartition des signaux optiques vers les abonnés. Cela inclut l'installation d'équipements tels que les répartiteurs optiques.

1.5.4. Pose de CDN (Content Delivery Network) :

La mise en place d'un réseau de diffusion de contenu pour optimiser la distribution des données et améliorer les performances, notamment la diffusion de médias, la mise en cache et la réduction de la latence.

1.5.5. Pose de Raccordement des Abonnés :

La pose de raccordement des abonnés englobe la dernière étape cruciale, assurant la connexion

Directe des utilisateurs finaux au réseau de télécommunication. Cela peut comprendre le raccordement des lignes de fibre optique à des points d'accès spécifiques dans les domiciles ou les entreprises. De manière plus détaillée, dans le contexte de déploiements :

FTTH (Fibre to the Home), la fibre optique est directement acheminée jusqu'au domicile de l'abonné, garantissant une connectivité optimale.

FTTC (Fibre to the Curb), la fibre optique est étendue jusqu'à des points de raccordement optique stratégiques, puis le raccordement final s'effectue en utilisant les lignes de cuivre existantes vers les domiciles.

Chapitre 01: présentation de l'organisme d'accueil

1.6. La gestion des projets dans Algérie Télécom :

La gestion de projet chez Algérie Télécom est un processus crucial qui nécessite une planification minutieuse et une coordination efficace pour assurer le déploiement réussi des infrastructures de télécommunication. Cette gestion comprend plusieurs étapes, allant de l'étude initiale des besoins à la réalisation concrète des projets sur le terrain.

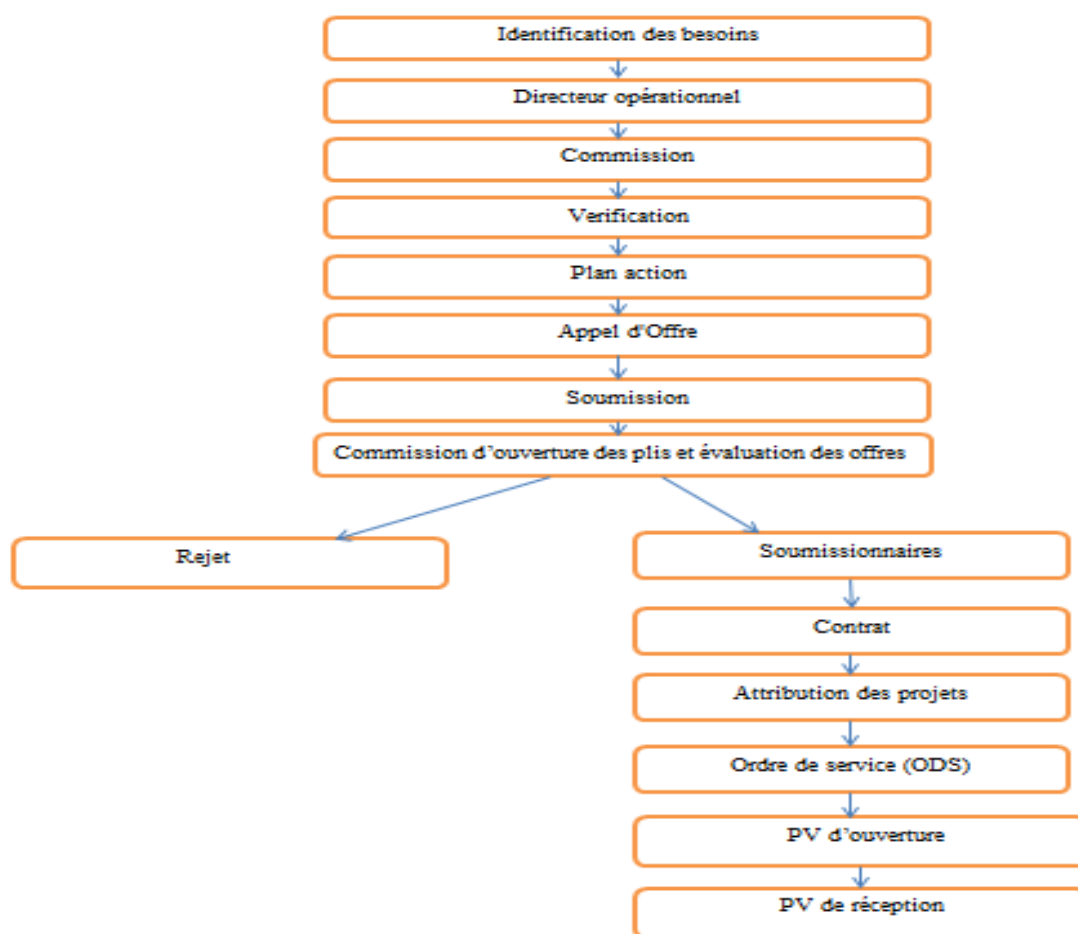


Figure 1.7 Organigramme de La gestion de projet chez Algérie Télécom

Chapitre 01: présentation de l'organisme d'accueil

Étape 1 : Analyse et Planification de la Demande:

- Étude de l'évolution de la demande des services (voix et data) par commune et Daïra.
- Planification de l'infrastructure télécoms nécessaire pour répondre viablement et fiablement aux demandes des clients, en élaborant un plan de charges fondé sur une co-collaboration avec les départements techniques Réseaux (Accès, Transport, Core et Énergie) et commerciaux au niveau de directeur opérationnel.
- Identification des besoins de projets.

Étape 2 : Établissement des Procédures et Appel d'Offres

- Le directeur opérationnel créer une commission pour la vérification des besoins (DEPRA).
- Élaboration du plan d'action 2024.
- La Direction opérationnelle de télécommunication (DOT) annonce un appel d'offre.
- La participation des soumissionnaires qualifiés selon des conditions.
- Composition d'une commission d'ouverture des plis et évaluation des offres (COPEO).

Étape 3 : Évaluation des Soumissions

- La commission fait l'acceptation ou le rejet du dossier soumissionnaires

Étape 4 : Conclusion des Contrats

- DOT fait un contrat d'adhésion avec les cocontractants

Étape 5 : Attribution des Projets et Début des Travaux

- L'attribution des projets aux entreprises (les cocontractants) fait dans le département réseaux d'accès (DRA)
- L'établissement d'une ordre de service (ODS) pour les travaux (numéro, nom de l'entreprise, délais de projet, date ...)
- Signature d'ODS par le DO
- Signature d'ODS par l'entreprise
- DRA lancer le projet avec un PV d'ouverture (processus verbal d'ouverture)
- L'entreprise fait une demande de réception au DOT
- DRA fait un PV de réception sans réserve

1.7. Conclusion:

Dans ce chapitre, nous avons présenté la direction opérationnelle d'Algérie télécom Mila et notamment le département réseau accès de la Sous-Direction technique, lieu de notre stage. Ensuite, nous avons présenté les volets d'Activité Essentiels d'Algérie télécom. Enfin, nous avons expliqué le processus de la gestion des projets dans Algérie télécom.

Chapitre 2 : Lignes de produits Logiciels

Chapitre 02 : Lignes de produits logiciels

2. Lignes de produits logiciels :

2.1. Introduction :

Les lignes de produits logiciels (LPL) représentent une approche puissante dans le domaine du développement logiciel, offrant des avantages significatifs en termes de réutilisation, de qualité, de rapidité sur le marché, de réduction de l'effort de maintenance et d'amélioration de l'estimation des coûts. Cette méthodologie repose sur la création de systèmes logiciels à partir d'un ensemble commun de fonctionnalités partagées, répondant ainsi aux besoins spécifiques d'un marché ou d'un domaine particulier.

Dans ce chapitre, premièrement, nous définissons les lignes de produit logiciels, Ensuite, nous présentons l'environnement pour l'ingénierie des lignes de produit logiciel, Après nous donnons des exemples réels des lignes de produit logiciel, Enfin, nous identifions les apports de l'utilisation des lignes de produit logiciel.

2.2. Définition :

Une ligne de produits logiciels est un ensemble des systèmes partageant des caractéristiques communes répondent à des besoins spécifiques dans un domaine spécifique [1]. La meilleure définition du terme LPL est fournie par le Software Engineering Institute de l'université Carnegie Mellon :

“A software product line (SPL) is a set of software-intensive systems that Share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way”.

2.3. Environnement pour l'ingénierie des lignes de produits logiciels :

Le paradigme de l'ingénierie des lignes de produits logiciels est séparé en deux processus [2]: l'ingénierie de domaine et l'ingénierie d'application (voir **figure 2.1**). L'ingénierie de domaine implique le développement et la construction des ressources qui seront utilisées pour créer des produits. D'autre part, l'ingénierie d'application se concentre sur l'utilisation de ces ressources pour construire un produit ou une application spécifique une fois que les points de variation ont été identifiés et définis [3].

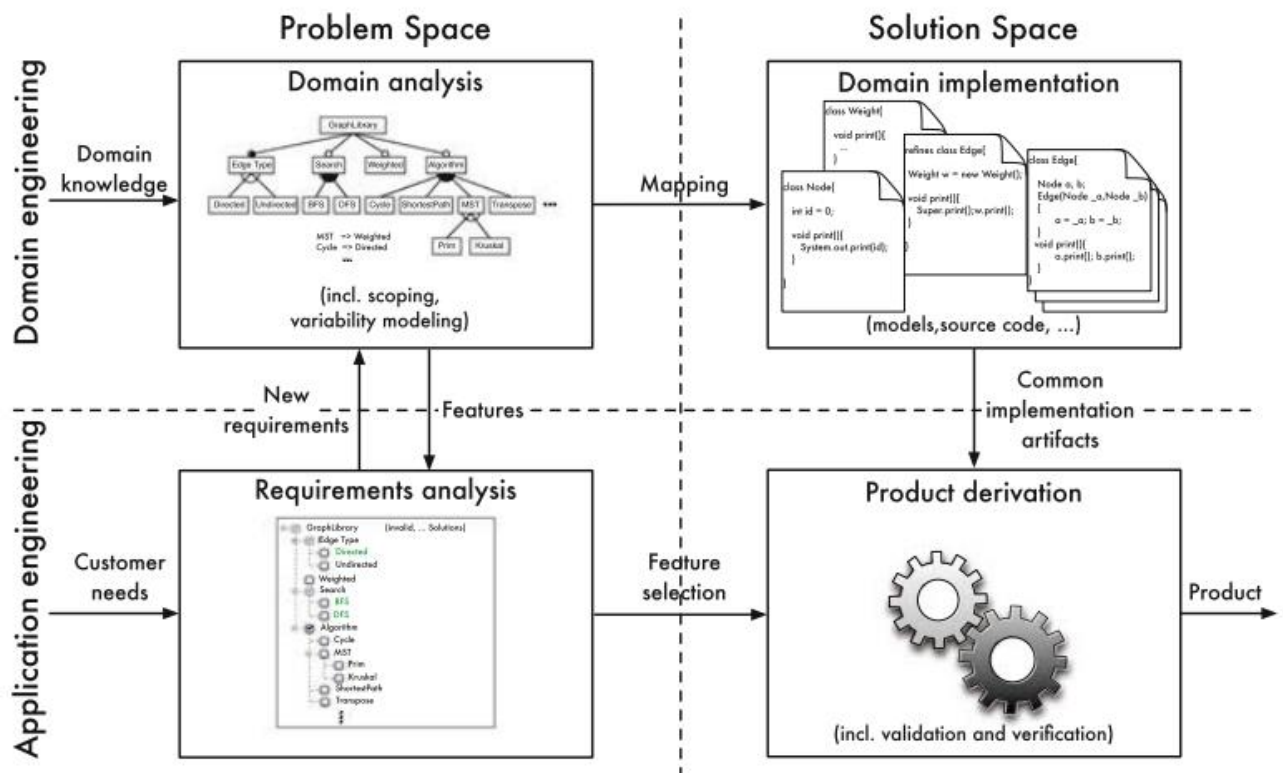


Figure 2.1 Ingénierie de domaine et ingénierie d'application [4]

2.3.1. Ingénierie de domaine :

Cette phase permet de gérer une ligne de produits logiciels dans son ensemble plutôt que comme une collection de produits distincts traités individuellement. La ligne de produits logiciels doit ainsi inclure les besoins spécifiés par toutes les catégories d'utilisateurs ciblées [3].

L'ingénierie de domaine inclut quatre activités : l'analyse de domaine, la modélisation de domaine, l'implémentation de domaine et le test de domaine :

2.3.1.1. Analyse de domaine :

L'analyse de domaine englobe l'ingénierie des exigences du domaine. En effet, les exigences sont analysées pour déterminer celles qui sont communes à tous les produits et celles qui sont spécifiques à des produits particuliers [3].

2.3.1.2. Modélisation de domaine :

L'objectif de la modélisation de domaine est de développer une architecture logicielle générique pour la ligne de produits logiciels. Cette architecture inclut un arbre de composants génériques qui capture les points communs et les variabilités de chaque LPL. La variabilité identifiée lors de

l'analyse de domaine doit être spécifiquement détaillée dans l'architecture de la LPL [5].

Chapitre 02 : Lignes de produits logiciels

2.3.1.3. Implantation de domaine :

L'implémentation d'application implique la mise en œuvre de l'architecture spécifique définie lors de la conception d'application, transformant ces conceptions en composants réels produits pendant la phase d'ingénierie de domaine, destinés à la construction d'un produit spécifique [5].

2.3.1.4. Test de domaine :

Cette tâche implique la validation et la vérification des composants réutilisables obtenus après l'étape précédente [4].

2.3.2. Ingénierie d'application :

L'ingénierie d'application consiste à utiliser les résultats de l'ingénierie de domaine pour construire un produit spécifique [1].

Tout comme l'ingénierie de domaine, l'ingénierie d'application se compose de quatre activités principales : la spécification des exigences de l'application, la modélisation de l'application, l'implémentation de l'application et les tests de l'application.

2.3.2.1. Exigences d'application :

Dans la phase d'expression des exigences d'application, les exigences de produit issues de l'ingénierie de domaine sont dérivées en prenant des décisions ou des choix concernant les points de variation afin de formuler les exigences spécifiques à un produit particulier [5].

2.3.2.2. Modélisation de l'application :

La modélisation de l'application facilite la création de l'architecture du produit à partir de l'architecture globale définie lors de la modélisation du domaine. Pour cela, les éléments essentiels du modèle sont choisis et intégrés [3].

2.3.2.3. Implémentation d'application :

Cette activité implique la concrétisation du produit final souhaité. Les composants correspondant aux parties sélectionnées lors de l'étape précédente sont ensuite assemblés. [3].

2.3.2.4. Test de l'application :

Le test d'application comprend les activités nécessaires pour valider et vérifier une application par rapport à ses spécifications [6].

Chapitre 02 : Lignes de produits logiciels

2.4. Exemples des lignes de produits logiciels :

2.4.1. Sandwicheries:

Dans le domaine des sandwicheries, une approche intéressante et réussie dans l'industrie de la restauration rapide consiste à offrir une ligne de produits où les clients peuvent créer leur sandwich selon leurs préférences. Plutôt que de proposer des sandwiches prédéfinis, ces établissements offrent une variété d'options, comprenant différents types de pain, de garnitures, d'accompagnements, de sauces et d'épices. Cette approche offre une grande flexibilité aux clients tout en simplifiant le processus pour le producteur. Plutôt que de gérer des commandes spécifiques à chaque fois, le producteur se concentre sur la préparation d'éléments de base comme des tranches de tomate, des viandes précuites et des sauces, qui peuvent ensuite être combinés selon les choix des clients. Certains établissements proposent même des outils comme des configurateurs papier pour aider les clients à composer leur sandwich idéal. [4].

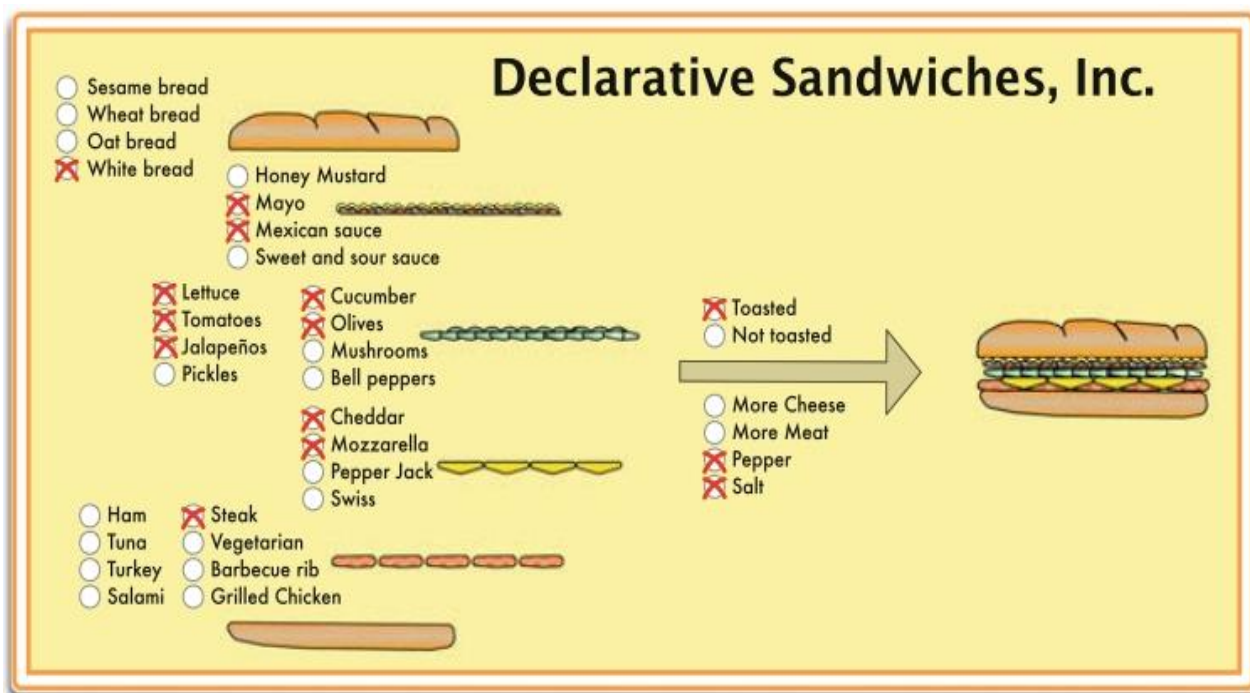


Figure 2.2 Le configurateur de Sandwich [4]

2.4.2. L'application de Bank Web :

La ligne de produit logiciel est importante dans le contexte de Bank Web car elle permet de gérer efficacement les différentes fonctionnalités et variations de l'application. En intégrant une approche de ligne de produit logiciel, l'application Bank Web peut offrir une flexibilité accrue pour répondre aux besoins des utilisateurs et s'adapter aux évolutions du marché. Par exemple, différentes versions de l'application peuvent être développées pour prendre en charge des fonctionnalités spécifiques à chaque type de carte de crédit ou à chaque service bancaire, tout en partageant un noyau commun de fonctionnalités. Cela permet une gestion efficace du développement, de la maintenance et de la personnalisation de l'application, ce qui peut conduire à une meilleure satisfaction des clients et à une plus grande compétitivité sur le marché.

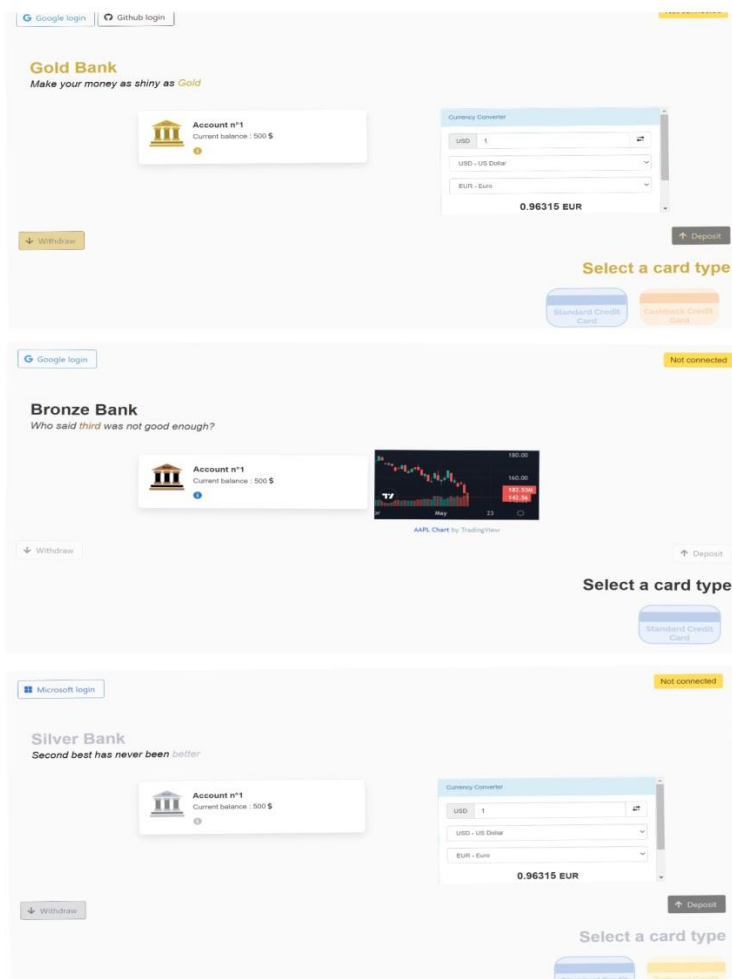


Figure 2.3 Bank Web [27].

Chapitre 02 : Lignes de produits logiciels

2.4.3. Téléphones mobiles Nokia :

Lignes de produits avec 25-30 nouveaux produits par an contre 5 par an à l'origine. Parmi les produits, il y a [10]:

- Nombre variable de clés
- Différentes tailles d'affichage
- Divers ensembles de caractéristiques
- 58 langues prises en charge
- 130 pays desservis
- Protocoles multiples
- Besoins de rétrocompatibilité
- Caractéristiques configurables
- Les besoins en matière de comportement des produits
- Changements après la sortie du produit



Figure 2.4 Téléphones mobiles Nokia [10]

2.5. Apports de l'utilisation des lignes de produits logiciels :

L'institut d'Ingénierie Logicielle (Software Engineering Institute) énumère les avantages suivants associés aux lignes de produits logiciels :

2.5.1. Réutilisation accrue du code :

La réutilisation a été définie par Kreuger comme [7]:

“La réutilisation de logiciels est le processus de création de systèmes logiciels à partir de logiciels existants plutôt que de construire des systèmes logiciels à partir de zéro. ”

Les lignes de produits logiciels capitalisent sur cette approche en permettant une réutilisation efficace des composants et des fonctionnalités partagées à travers une ligne de produits. Cette réutilisation réduit de manière significative le temps et les efforts nécessaires au développement, car les équipes n'ont pas besoin de concevoir chaque aspect d'un produit à partir de rien. En utilisant des composants préexistants, les développeurs peuvent se concentrer sur l'intégration de ces éléments dans de nouveaux produits, ce qui accélère le processus de développement et améliore l'efficacité globale du projet.

2.5.2. Meilleure qualité logicielle :

Les artefacts sur la plateforme sont examinés et testés dans de nombreux produits. Ils doivent prouver leur bon fonctionnement dans plus d'un type de produit. Cette assurance qualité approfondie implique une probabilité significativement plus élevée de détecter des défauts et de les corriger, augmentant ainsi la qualité de tous les produits [8].

2.5.3. Rapidité sur le marché :

La réduction des délais de mise sur le marché et des coûts de développement s'accompagne d'une amélioration de la qualité des produits [3].

La **figure 2.5** illustre la réduction du temps nécessaire pour la mise sur le marché en utilisant ou non l'ingénierie des lignes de produits. Dans le cas d'un produit unique, le temps de mise sur le marché reste constant. Cependant, pour l'ingénierie des lignes de produits, il est crucial de prendre en compte le temps nécessaire à la construction des éléments communs à tous les produits. Cependant, ce temps est compensé par la réutilisation des éléments communs [3].

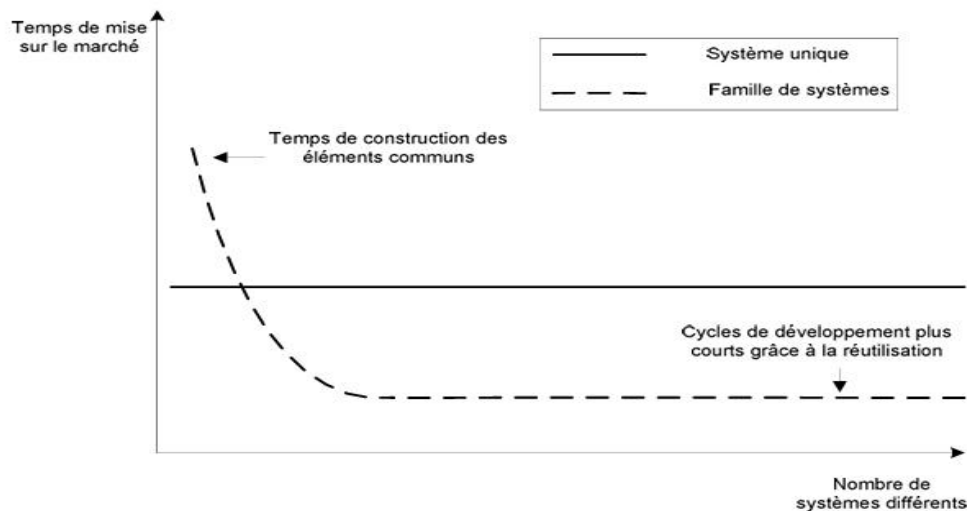


Figure 2.5 temps de mise sur le marché avec et sans l'ingénierie de lignes de produits [3]

2.5.4. Réduction de l'effort de maintenance :

Lorsque qu'un artefact de la plateforme est modifié, par exemple pour corriger une erreur, les changements peuvent être propagés à tous les produits dans lesquels l'artefact est utilisé. Cela peut être exploité pour réduire l'effort de maintenance. Dans le meilleur des cas, le personnel de maintenance n'a pas besoin de connaître tous les produits spécifiques et leurs composants, réduisant ainsi également l'effort d'apprentissage. Cependant, étant donné que les artefacts de la plateforme sont modifiés, tester les produits est encore inévitable. Cependant, la réutilisation des procédures de test est également au cœur de l'ingénierie des lignes de produits et contribue à réduire l'effort de maintenance. Les techniques utilisées en ingénierie des lignes de produits rendent un système plus facilement maintenable comme indiqué dans [Coplien 1998] : *"Les mêmes techniques de conception qui conduisent à une bonne réutilisation conduisent également à l'extensibilité et à la maintenabilité dans le temps"* [2].

2.5.5. L'amélioration de l'estimation des coûts :

Les lignes de produits logiciels peuvent permettre des économies de coûts à long terme en raison de la réutilisation, de la modularité et d'une gestion améliorée des variantes, ce qui réduit les dépenses liées au développement, à la maintenance et au support logiciel.

2.6. Conclusion :

Dans ce chapitre, nous avons défini les lignes de produit logiciel, Ensuite, nous avons présenté l'environnement pour l'ingénierie des lignes de produit logiciel, Après cela, nous avons donné des exemples réels de lignes de produit logiciel, Enfin, nous avons identifié les apports de l'utilisation des lignes de produit logiciel, montrant comment cette approche peut améliorer l'efficacité et la flexibilité du développement logiciel.

Chapitre 03 :

Migration du système vers

Les lignes de produits logiciels

3. Migration du système vers les lignes de produits logiciels :

3.1. Introduction

Dans ce chapitre, nous présentons la plateforme *Mobioos Forge* qui permet la création des lignes de produits logiciels grâce à ses trois principes fondamentaux suivants: le modèle de fonctionnalités, le mappage des fonctionnalités et la personnalisation. Ensuite, nous décrivons notre approche permettant la génération des lignes de produits logiciels à partir d'une application open source.

3.2. Plateforme Mobioos forge :

Mobioos forge (MF) est une extension pour Visual Studio Code (VScode). MF s'aligne avec le processus LPL, mais propose une vision orientée vers la migration, permettant aux développeurs de créer des LPL à partir de code source d'application individuelle [11].

Les LPLs reposent sur trois principes fondamentaux : le modèle de fonctionnalités (Feature model), le mappage des fonctionnalités (Feature Mapping) et la personnalisation (Customization) [12].

Dans ce qui suit, nous présentons ces trois principes :

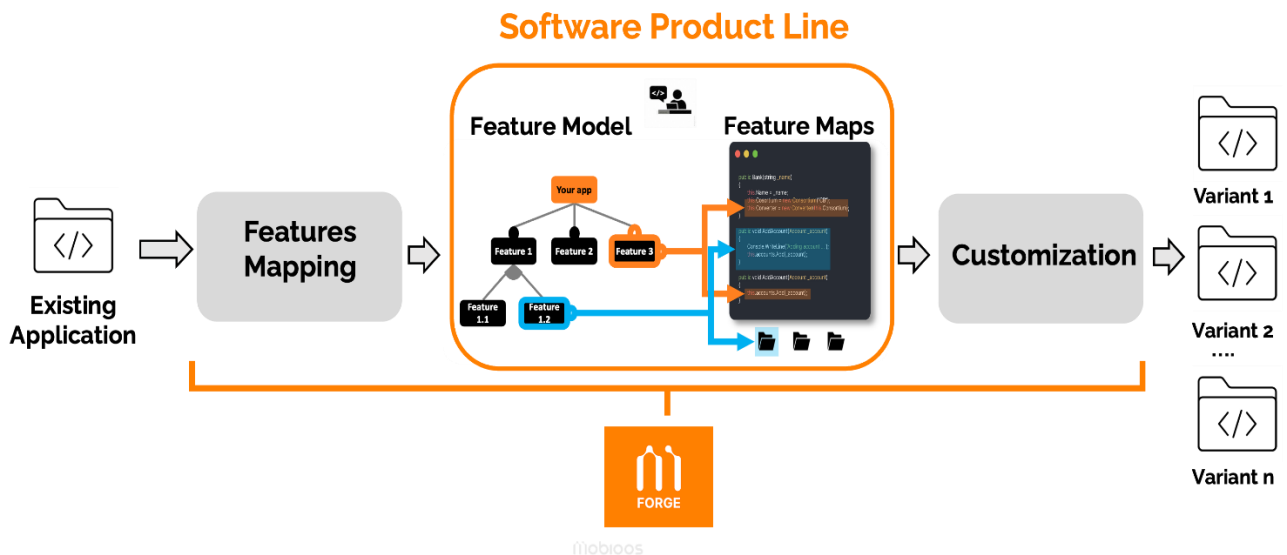


Figure 3.1 Le processus de Mobioos Forge pour les lignes de produits logiciels

Chapitre 03 : Migration du système vers les lignes de produits logiciels

3.2.1. Modèle de fonctionnalités:

La première étape pour mettre en œuvre les lignes de produits logiciels consiste à décrire explicitement les fonctionnalités métier de la famille d'applications à l'aide de ce que l'on appelle des Modèles de Fonctionnalités. Les Modèles de Fonctionnalités (FM) sont une notation populaire capable de décrire un domaine d'application donné en termes de fonctionnalités métier [12].

Le modèle de fonctionnalités fournit une notation graphique en forme d'arbre. La **figure 3.2** montre le concepteur de modèle de fonctionnalités intégré de MF utilisé pour concevoir le modèle de fonctionnalités de l'application Notepad. Ce modèle de fonctionnalités contient 8 fonctionnalités, y compris la fonctionnalité racine nommée Notepad [11].

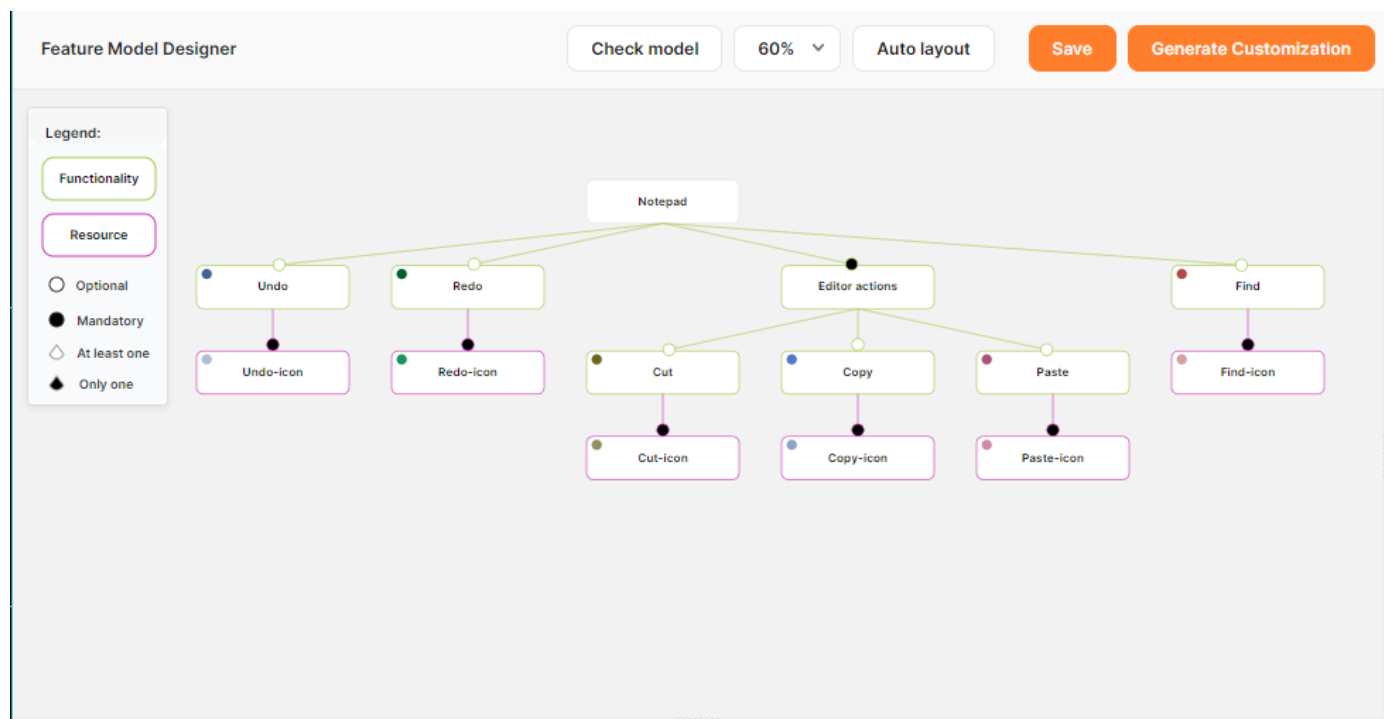


Figure 3.2 Modèle de fonctionnalités de Notepad créé avec MF

3.2.2. Mappage des fonctionnalités (feature mapping) :

La deuxième étape pour implémenter une ligne de produits logiciels consiste à mapper ces fonctionnalités sur le code source. L'objectif est de créer un pont entre les fonctionnalités du domaine et le code source. Le mappage des fonctionnalités vise à localiser, pour chaque fonctionnalité, les fragments de code qui l'implémentent. L'approche la plus couramment utilisée pour le mappage des fonctionnalités repose sur l'annotation ou la décoration du code source pour mettre en évidence le lien avec les fonctionnalités [12].

Chapitre 03 : Migration du système vers les lignes de produits logiciels

Les marqueurs MKR sont fournis par les développeurs en fonction de leur connaissance du code source et des fonctionnalités identifiées. En fonction de la granularité du mappage, MF propose deux types de marqueurs [11] :

- **Les marqueurs de code** Ces modèles sont appliqués aux fragments de code. Ils associent le fragment de code sélectionné à une fonctionnalité spécifique.
- **Les files markers** qui associent un fichier/dossier entier à une fonctionnalité spécifique.

La Figure 3.3 améliore les exemples de marqueurs et de cartes présents dans l'application *Notepad*. Dans cette figure, nous pouvons voir à gauche un exemple de marqueur de fichier ajouté au fichier *RedoAction.java* ainsi que des cartes à l'intérieur du fichier *Notepad.java*. Sur le côté droit de la figure, plusieurs cartes sont affichées dans l'éditeur *VScode*. Elles sont distinguables par les couleurs appliquées dans l'éditeur.



Figure 3.3 Exemples de marqueurs et de cartes dans MF

Une fois qu'un marqueur est ajouté, l'activité de cartographie des fonctionnalités assiste les développeurs avec un processus semi-automatique qui infère des cartes, c'est-à-dire des fragments de code et des fichiers/dossiers appartenant à la fonctionnalité à partir du marqueur initial. Pour que MF calcule des cartes à partir d'un marqueur, le marqueur doit être ajouté à une déclaration (déclaration de classe, déclaration de méthode, etc.). Une fois qu'un marqueur est ajouté, le développeur doit valider/supprimer les cartes fournies par MF. La validation d'une

Chapitre 03 : Migration du système vers les lignes de produits logiciels

carte implique de confirmer sa position dans le code source ou de la modifier pour couvrir une zone différente du code source. Une fois que la carte est validée, MF calcule d'autres cartes potentielles à l'intérieur du code source qui pourraient être associées à la fonctionnalité concernée. Il est important de noter qu'une validation de carte peut entraîner la validation automatique d'autres cartes. Si la zone de la carte nouvellement validée couvre d'autres cartes liées à la même fonctionnalité, ces cartes seront automatiquement validées. Ainsi, le nombre de validations requis diminue, ce qui entraîne une réduction de la charge de travail du développeur [11].

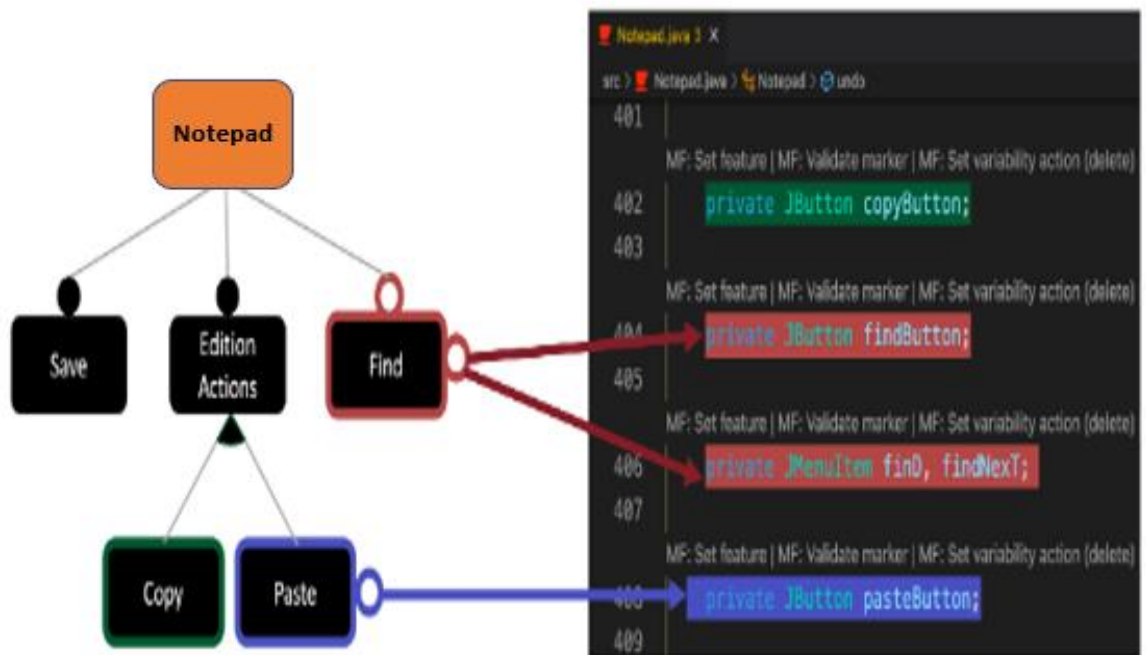


Figure 3.4 Mappage des fonctionnalités

3.2.3. Personnalisation :

La personnalisation implique de générer de nombreuses variantes à partir de la même famille. Cela peut comprendre l'activation ou la désactivation de certaines fonctionnalités que l'utilisateur souhaite avoir sur une variante spécifique comme illustré dans les figures suivantes [12].

La Figure 3.5 montre l'éditeur de configuration décrit intégré dans MF. Cette configuration a été utilisée sur le SPL *Notepad*. Comme vous pouvez le voir, seules les fonctionnalités *Refaire* (*Redo*) et *Annuler* (*Undo*) sont sélectionnées.

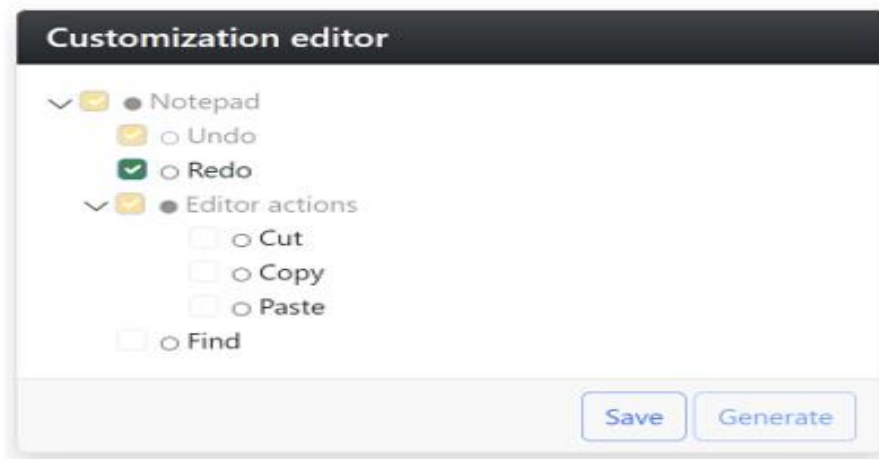


Figure 3.5 L'éditeur de configuration dans MF

3.3. La génération des lignes de produits logiciels à partir de la gestion des projets :

Dans le contexte de la migration des LPL utilisant les capacités de MF, nous avons choisi une application open-source existante relative à la gestion de projets comme étude de cas illustrative. Cette application, développée avec *Java/Maven*. Cette application permet le suivi du développement local au niveau du ministère de l'enseignement supérieur et la recherche scientifique. Cette application contient les fonctionnalités suivantes : la gestion des utilisateurs, l'authentification, le suivi des projets. Dans ce qui suit, nous présentons notre approche de génération des lignes de produits logiciels à partir de la gestion des projets en détail.

3.3.1. Le modèle de fonctionnalité :

La figure 3.6 montre le modèle des fonctionnalités conçu pour la gestion des projets. Ce modèle inclut plusieurs fonctionnalités clés liées à la gestion de projets. Il comprend la gestion des utilisateurs (user management), une fonctionnalité optionnelle qui concerne la gestion des comptes utilisateurs et leurs rôles. La fonction *login* (elle est quant à elle obligatoire) permettant aux utilisateurs d'accéder au système en saisissant leurs identifiants. Parmi les autres fonctionnalités de base on trouve la gestion de projet (*Project management*), également optionnelle, englobant les sous-fonctionnalités suivantes : *progress, lot, market, advancement*.

Le modèle inclut également la fonctionnalité optionnelle *PEC* pour le suivi des projets en cours, ainsi que la fonctionnalité *evolution* qui suit l'évolution du secteur de l'enseignement supérieur.

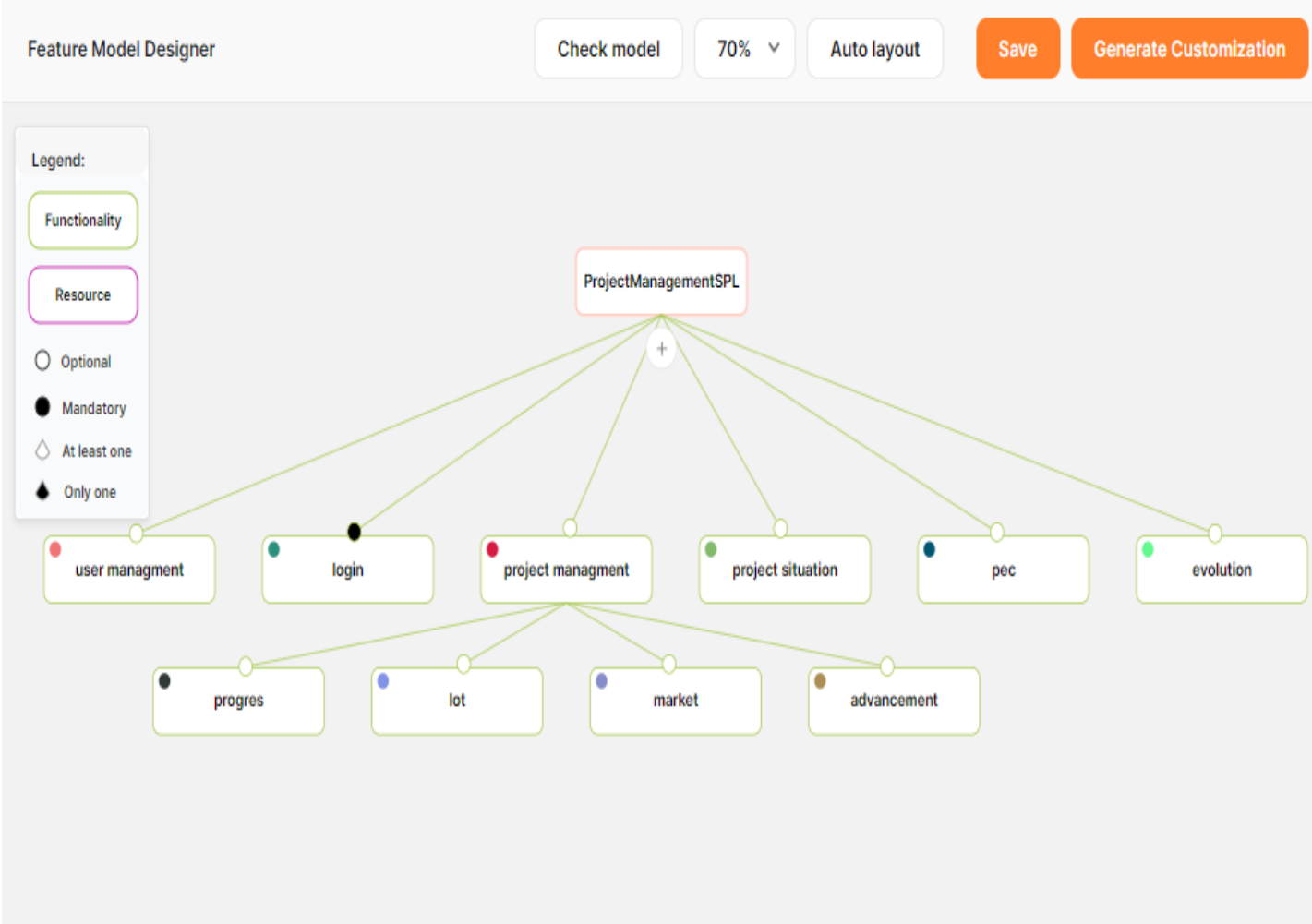


Figure 3.6 Le modèle de fonctionnalités résultant de gestion des projets.

3.3.2. Marqueurs et cartes (Markers and Maps) :

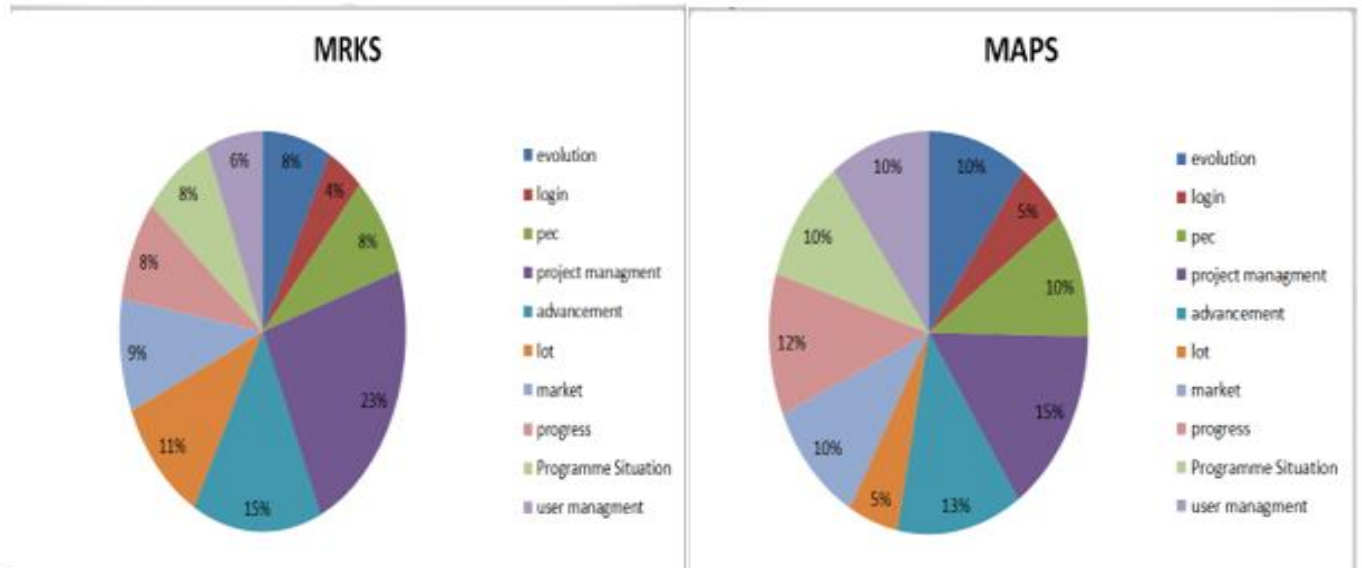


Figure 3.7 Quantité de marqueurs et de cartes dans *Project Management SPL*

L'analyse des résultats de la Figure 3.7 présentés révèle que la fonctionnalité de gestion de projet nécessitait un pourcentage élevé de marqueurs pour être entièrement cartographiée dans le code, représentant 23% du total des marqueurs ajoutés. Cependant, le nombre de cartes associées à cette fonctionnalité était légèrement inférieur, soit 15%, ce qui témoigne de l'efficacité de MF dans la facilitation du mappage des fonctionnalités dans le code source. D'autre part, la fonctionnalité *Login* nécessitait un pourcentage plus faible de marqueurs (4%), mais le pourcentage de cartes générées était légèrement supérieur (5%), indiquant que les cartes associées à cette fonctionnalité pourraient avoir été cartographiées avec moins d'efficacité. La fonctionnalité *evolution* nécessitait un pourcentage moyen de marqueurs et de cartes, avec respectivement 8% et 10%, suggérant une cartographie adéquate et organisée de cette fonctionnalité dans le code.

L'importance des cartes générées à partir des marqueurs pour faciliter de la compréhension et de l'analyse du code, ainsi que pour l'identification des emplacements liés à chaque fonctionnalité, ce qui contribue à économiser du temps et des efforts pour les développeurs et à améliorer la qualité de l'application.

Chapitre 03 : Migration du système vers les lignes de produits logiciels

Feature	LOCs	Percentage of Mapped LOCs	Impacted Files	Maps / Marker
Login	35363	74.55%	261	0.44
User Management	715	1.51%	12	0.57
Project Management	2380	5.02%	28	0.44
Progress	1370	2.89%	14	0.5
Lot	514	1.08%	9	0.13
Market	1632	3.44%	39	0.4
Advancement	1377	2.90%	15	0.31
Programme Situation	1229	2.59%	15	0.47
Pec	1404	2.96%	18	0.44
Evolution	1453	3.06%	15	0.47

Tableau 3.1 Métriques concernant l'activité de mappage des fonctionnalités.

Ce tableau fournit des données sur diverses fonctionnalités en termes de lignes de code (LOCs), pourcentage de LOCs mappées, fichiers impactés et cartes par marqueur.

La fonctionnalité "Login" possède le plus grand nombre de lignes de code avec 35 363 LOCs, représentant 74,55 % du nombre total de LOCs mappées. En revanche, la fonctionnalité "Lot" a le plus petit nombre de lignes de code avec 514 LOCs, représentant 1,08 % du nombre total de LOCs mappées. Les LOCs des autres fonctionnalités varient entre 715 et 2380, représentant entre 1,51 % et 5,02 % du nombre total de LOCs mappées.

La fonctionnalité "Login" domine avec près des trois quarts du nombre total de LOCs mappées (74,55 %). Les autres fonctionnalités représentent chacune une petite fraction du nombre total de LOCs mappées, la fonctionnalité *Project Management* étant la plus élevée parmi elles avec 5,02 %.

En ce qui concerne les fichiers impactés, nous remarquons que la fonctionnalité *Login* a le plus grand nombre de fichiers (261 fichiers), tandis que la fonctionnalité *Lot* a le moins de fichiers (9 fichiers). Les autres fonctionnalités ont un nombre significativement inférieur de fichiers, allant de 12 fichiers (*User Management*) à 39 fichiers (*Market*).

En examinant les cartes par marqueur, nous constatons que la fonctionnalité "*User management*" a la valeur la plus élevée égale 0,57, tandis que la fonctionnalité "*Lot*" a la valeur la plus basse égale 0,13. Les valeurs pour les autres fonctionnalités vont de 0,31 (*Advancement*) à 0,50 (*Progress*).

Chapitre 03 : Migration du système vers les lignes de produits logiciels

Nous remarquons que la fonctionnalité "Login" possède le plus grand nombre de LOCs et de fichiers impactés, ce qui indique son implémentation étendue dans la base de code. Bien que la fonctionnalité "User management" ait un nombre relativement faible de LOCs, elle contient un nombre relativement élevé de cartes par marqueur. Les fonctionnalités "Project management" et "Market" ont un nombre de LOCs relativement élevé mais impactent un nombre moyen de fichiers comparé à la fonctionnalité "Login". La fonctionnalité "Lot" a le plus petit nombre de LOCs et impacte le moins de fichiers parmi les fonctionnalités non liées à la gestion des utilisateurs.

3.3.3. Variantes Dérivées:

Le tableau ci-dessous présente un ensemble de variantes dérivées de l'application *Project Management SPL*, avec leurs fonctionnalités activées, ainsi que le nombre de lignes de code pour chaque variante et le nombre de lignes supprimées par rapport au total des lignes de code de l'application d'origine.

Variants (enabled features)	LOCs	Deleted LOCs
Project Management SPL (original application)	47391	0
Login	35319	12072
<i>Test 1: Login, User Management</i>	36050	11341
<i>Test 2: Login, Project Management, Lot, Market</i>	39840	7551
<i>Test 3: Login, Evolution, Pec, Programme Situation</i>	39371	8020

Tableau 3.2 Métriques concernant quelques variantes générées.

La variante Test 1 :

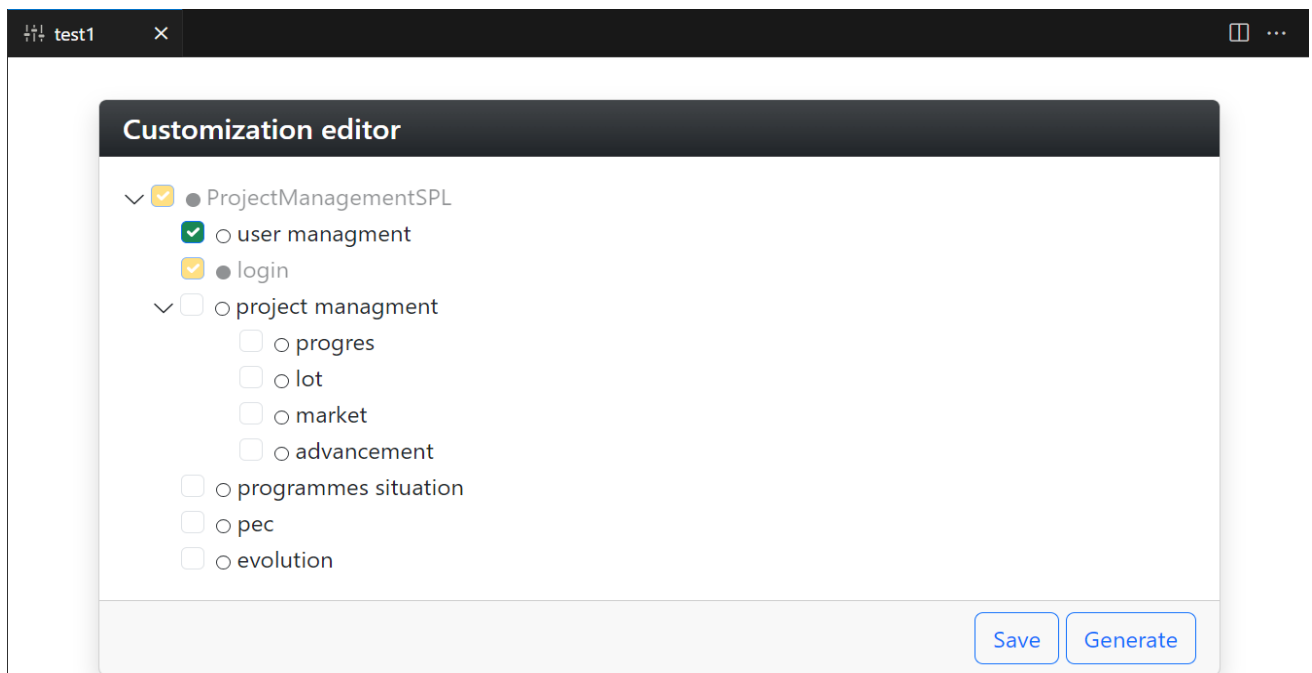


Figure 3.8 Éditeur de personnalisation de la variante test 1



Figure 3.9 L'Interface de résultat de la variante test 1

Chapitre 03 : Migration du système vers les lignes de produits logiciels

La variante Test 2 :

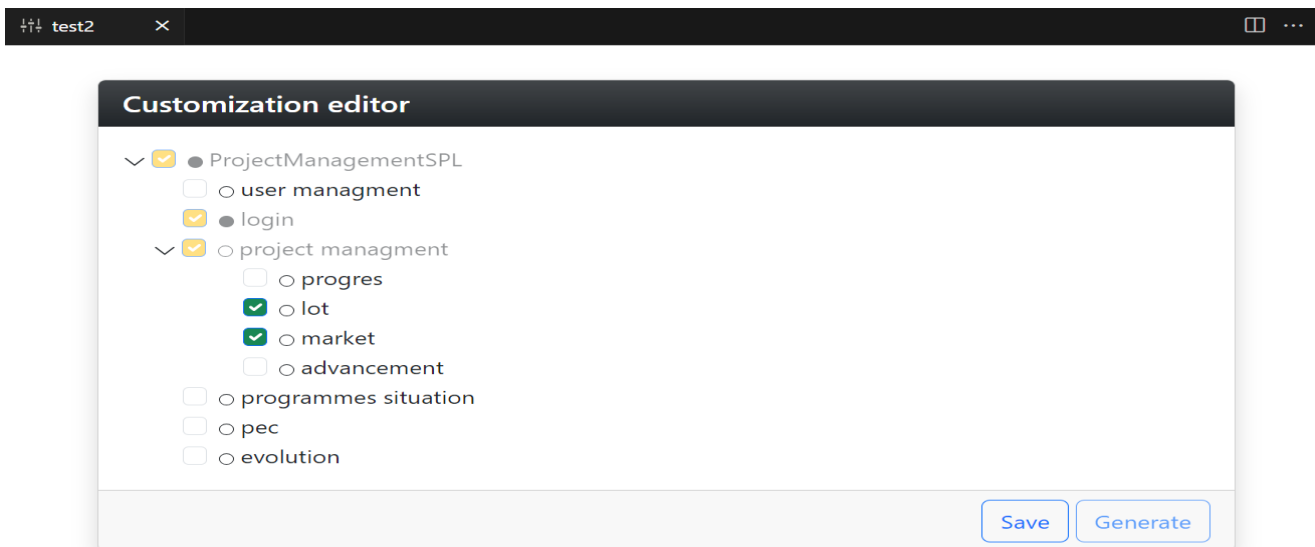


Figure 3.10 Éditeur de personnalisation de la variante test 2



Figure 3.11 L'Interface de résultat de la variante test 2

La variante Test 3 :

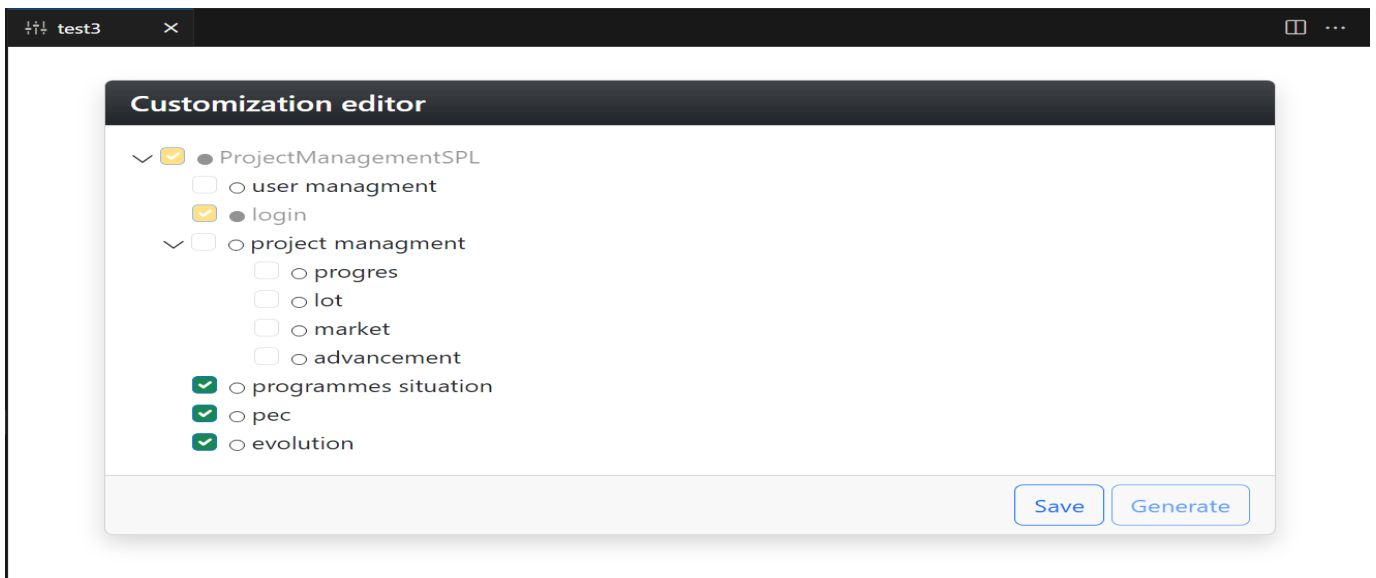


Figure 3.12 Éditeur de personnalisation de la variante test 3



Figure 3.13 L'Interface de résultat de la variante test 3

3.4. Conclusion :

Dans ce chapitre, nous avons introduit la plateforme Mobioos Forge, Ensuite nous avons présenté la génération des lignes de produit à partir d'une application de la gestion des projets.

Chapitre 04 :

Extension du système

Chapitre 04 : Extension du système

4. Extension du système :

4.1. Introduction

Dans ce chapitre, nous commençons par définir les microservices. Ensuite, nous présentons le processus unifié (UP) ainsi que le langage de modélisation UML (Unified Modeling Language). Après cela, nous procédons à la conception du système. Enfin, nous terminons par l'implémentation du système.

4.2. Architecture microservices :

4.2.1. Définition :

Les microservices représentent à la fois une architecture et une approche logicielle moderne où le code d'application est divisé en petits composants autonomes et faciles à gérer. Chaque microservice peut utiliser sa propre base de données, et peut être développé à l'aide d'un langage [13].

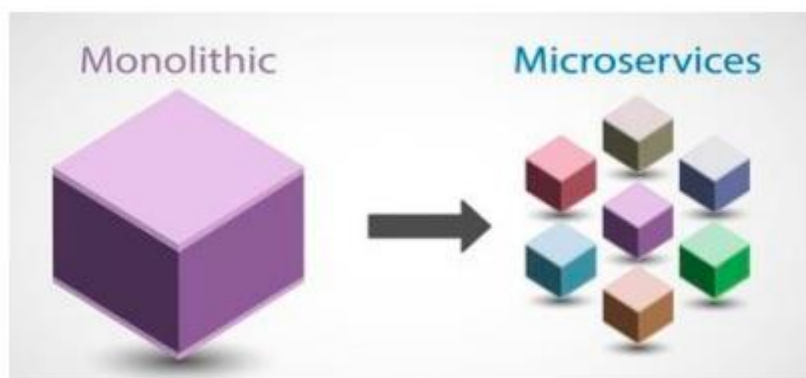


Figure 4.1 Découpage d'une application en petites services [13]

4.2.2. Exemple d'une architecture Microservice :

La **Figure 4.2** montre l'architecture d'une application de gestion de restaurant basée sur Microservices. L'application est implémentée à l'aide de Spring Boot et se compose de modules Ci-dessous : vendeur de tickets, statistiques d'accès au restaurant, étudiants et contrôle d'accès Dans le restaurant. De plus, chaque module utilise sa propre base de données [13].

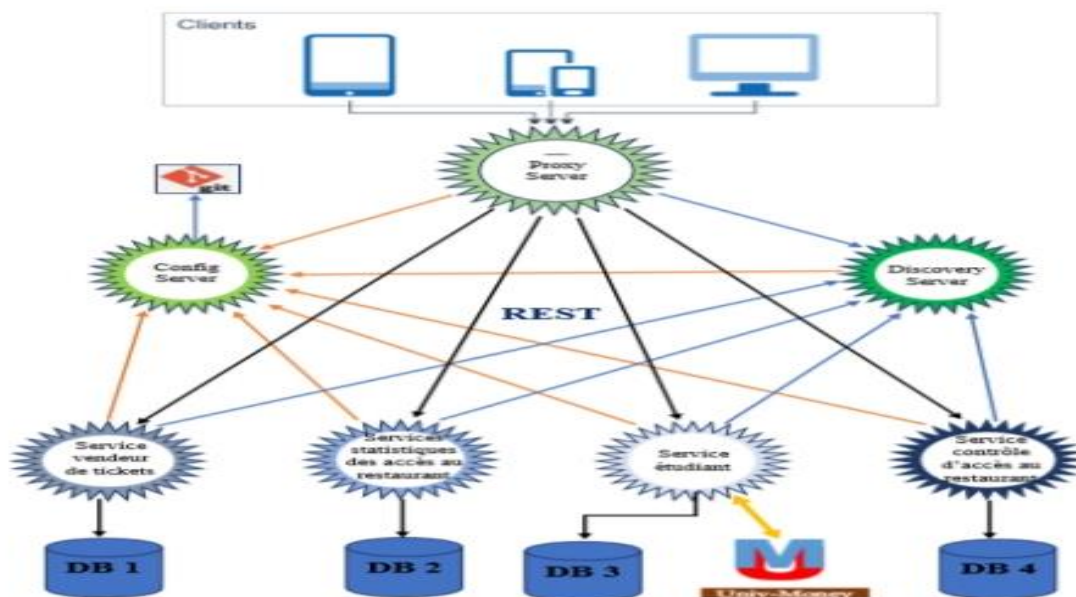


Figure 4.2 Exemple d'une architecture microservices

4.2.3. Les avantages des microservices :

Vous trouverez ci-dessous les avantages des microservices :

- Chaque microservice peut utiliser sa propre base de données [13].
- Chaque service est conçu pour un ensemble de capacités et se concentre sur la résolution d'un problème spécifique. En d'autres termes, un microservice doit accomplir une fonctionnalité unique et complète du système global [14].
- Les fonctionnalités de l'application sont réparties sur plusieurs services. Lorsqu'un microservice échoue, les fonctionnalités fournies par d'autres microservices continuent disponibles [13].
- Tout problème ou défaillance qui survient dans un microservice affecte uniquement ce microservice, sans avoir d'impact sur l'ensemble de l'application [15].

4.3. Le choix de la méthodologie :

Un certain nombre de méthodes de développement logiciel reposent sur UML, telles que UP et TUP. Parmi ces méthodes, nous avons choisi la méthode UP (Unified Process).

4.3.1. Rappel sur le processus unifié :

4.3.1.1. Définition du processus unifié :

Le processus unifié est un processus de développement logiciel itératif, axé sur l'architecture, guidé par des cas d'utilisation et orienté vers la réduction des risques. Il s'agit d'un cadre de processus adaptable à une vaste gamme de systèmes logiciels, à divers domaines d'application, à différents types d'entreprises, à divers niveaux de compétences et à des entreprises de toutes tailles [13].

4.3.1.2. Les caractéristiques du processus unifié :

Tout processus UP se distingue par les caractéristiques suivantes [16]:

- **UP est itératif et incrémental :**

Le projet est découpé en itérations ou étapes à court terme, ce qui permet un meilleur suivi de l'avancement global.

À la fin de chaque itération, une partie exécutable du système final est produite de manière incrémentielle (voir **Figure 4.3**)

- **UP est centré sur l'architecture:**

Tout système complexe doit être divisé en composants modulaires pour en faciliter l'entretien et l'évolution. Cette architecture doit être modélisée en UML plutôt que simplement documentée par du texte.

- **UP est guidé par les cas d'utilisation d'UML:**

La réalisation du projet est orientée par les besoins et les exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits de manière précise et priorisés.

- **Piloté par les risques :**

Les causes majeures d'échec d'un projet logiciel doivent être identifiées et écartées en priorité [13].

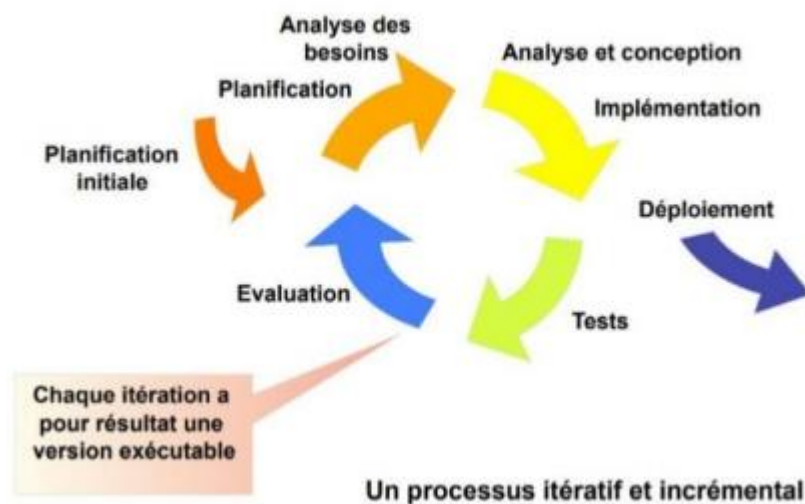


Figure 4.3 déroulement de processus unifié [16]

4.3.2. Rappel sur le langage UML :

4.3.2.1. Définition d'UML :

L'UML (Unified Modeling Language) est un langage de modélisation à la fois graphique et textuel. Il permet de comprendre et de décrire les besoins, d'identifier et de documenter les systèmes, de concevoir des architectures logicielles, de formuler des solutions et de communiquer différentes perspectives [17].

4.3.2.2. Définition d'un diagramme :

Un diagramme UML est une représentation graphique qui illustre un aspect particulier du modèle. Chaque type de diagramme UML possède sa propre structure et une sémantique spécifique [17].

Pour modéliser notre projet, nous avons intéressé à deux types de diagrammes : le diagramme de cas d'utilisation et les diagrammes de classes

Chapitre 04 : Extension du système

Rappel n° de l'opération: N1 011 044 01 4045 000 045 22 029

C - ECHancier PREVISIONNEL DES PaiEMENTS (10° DA)

ج - جدول الزمني التقديري للتدفقات (10° ج)

الجمع « MONTANT en DINAIRS الفياتر »	سنة 2022	سنة 2023	سنة 2024	سنة 2025		
DEVIS DIRECTS						
TOTAL المجموع	0	40 000	59 000	0		

العنوان: 3

رقم العملية: N1 011 044 01 4045 000 045 22 029

مبلغ التمويل الذاتي: /

معلومات أخرى: MI/2022/DI/82, 04/01/2022

توزيع رخصة الإقتراف حسب النصوص: 10° DA

N°	Lots	Montant
Lot N°01	Réalisation de 06 classes -A-	26 506
Lot N°02	Réalisation de 06 classes -B-	27 564
	CTC	2 453
	Etude	3 480
	Suivi	722
	Frais de publicité ANEP	
MONTANT DE L'OPERATION NON VENTILLE		36 085
Total		99 000

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLICUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

ولاية النعامة

مديرية البرمجة و متابعة العمليات

WILAYA DE NAAMA

DIRECTION DE LA PROGRAMMATION ET DE SUIVI BUDGETAIRES

N° 03 /DPSB/2024

Le 02/04/2024

N° Decision d'intégration MF /2023/DIN/20 du 06/09/2023

2023/09/06 / 20 / 1 / 2023 / م / 2023 / م / 2023 / م / 2023 / م

N° Fixe 22 45 02 014

رقم التثبيت

مجموعة البرامج	الترتيب الوظيفية
البرامج	التعليم
البرنامج الفرعي	التعليم التقني و الإبتدائي
النشاط (sous action)	النشاط غير معترف الوحدة
Titre 3	تلفقات الاستثمار
N° de l'opération	N1 011 044 01 4045 000 045 22 029

مقرر إعادة الترتيب: X

إن مسؤول النشاط

بمقتضى القانون العضوي رقم 18-15 المؤرخ في 22 ذي الحجة عام 1439 الموافق 2 سبتمبر سنة 2018 المتعلق بقرائن العمالة، العمل والمهنة

بمقتضى المرسوم التنفيذي رقم 20-403 المؤرخ في 14 جندب الأول عام 1442 الموافق 29 ديسمبر 2020، الذي يحدد شروط إنتاج وتسجيل البرامج العمل والمهنة

بمقتضى المرسوم التنفيذي رقم 20-404 المؤرخ في 14 جندب الأول عام 1442 الموافق 29 ديسمبر 2020، الذي يحدد إجراءات تسجيل وتكوين الاضغاث العمالية

بمقتضى القرار رقم 424 المؤرخ في 28 ديسمبر 2023، العمل والتمه نظر رقم 124 المؤرخ في 15 ابريل 2022 الذي يحدد الاضغاث الفرعية لتفقت وكذا ترميز الضمانات حسب الطريقة الاقتصادية لأجاء ميزانية الدولة

بمقتضى القرار رقم 03 المؤرخ في 11 يناير 2023 الذي يحدد إجراءات إنتاج وتسجيل عماليات الاستثمار العمومي للدولة بعنوان برنامج

بمقتضى المنشور رقم 1564 المؤرخ في 2023/03/06 المتعلق بإعداد وتنفيذ التسيير الميزانياتي لتفقت الاستثمار

بمقتضى مقرر معلق من الإجماع لسنة 2023 كوزارة العمالية «شعبوية العمدة لتسيير» رقم 309 المؤرخ في 10 سبتمبر 2023

بمقتضى مقرر تغيير هيئة لجنة رقم 213/م/م/2023 المؤرخ في 2023/12/17

بمقرر ما يلي

DECIDE

المادة 1- يتم إعادة ترتيب العملة المشار اليها أعلاه والتي جاء تحديدا لتدبرمج هذا المقرر طبقا لاستخراج معلق من الإجماع المذكور أعلاه

ARTICLE 1- L'opération dont le N° est indiqué ci dessous et dont le libellé suit est Reclassee par la présente décision conformément à l'extrait de l'annexe de la décision d'intégration suscité

Etude, suivi, réalisation et équipement d'une école primaire type 2 à la cité HR route ABK commune de Naama

المادة 2- قدر كلفة العملة المصنفة بالمرفق: 99 000 000,00

ARTICLE 2- Le coût de l'opération est de: 99 000 000,00 DA

المادة 3- يوجد كل من التعديل هيكلية للتدفق، نوع التمويل والجدول الزمني التقديري للتدفقات على التوالي في المصارف أ، ب، ج والشعبوية

ARTICLE 3- La structure du coût, la nature du financement et l'échéancier prévisionnel des paiements sont détaillés respectivement dans les tableaux A, B et C suivants.

Figure 4.5 Décision d'intégration d'un projet dans la nouvelle loi

Dans le cadre de notre travail, nous allons mettre en place une extension du système basée sur la nouvelle loi. Cette nouvelle procédure vise à améliorer l'organisation et la traçabilité de nos projets. Chaque code d'opération sera structuré de manière à refléter des informations spécifiques et pertinentes, telles que "portefeuille de programmes", "programme", "sous-programme" et "action (sous-action).

Chapitre 04 : Extension du système

4.4.2. Architecture logicielle du système:

L'architecture logicielle représente de manière symbolique et schématique les composants distincts d'un ou de plusieurs systèmes informatiques, ainsi que leurs relations et interactions mutuelles. Pour ce projet, nous avons utilisé l'architecture Microservices. Les microservices permettent de développer des applications modulaires et indépendantes, offrant ainsi une flexibilité et une adaptabilité accrues au code.

4.4.2.1. Ajout d'un nouveau microservice au système :

Le principe est d'ajouter un nouveau microservice à un système déjà découpé en trois parties distinctes :

- Une partie dédiée à la gestion des statistiques.
- Une partie pour la gestion de projets.
- Une partie pour la gestion des utilisateurs.

En plus de ces trois microservices principaux, nous avons ajouté un nouveau microservice pour améliorer encore d'avantage la fonctionnalité du système nommé *Gestion projet actualisée*. Ce Microservice (*Gestion projet actualisée*) sera constitué de :

- Page pour consulter la liste des projets.
- Page d'ajout d'un nouveau projet.
- Page pour modifier les informations d'un projet.
- Page pour consulter les projets en cours et les projets clôturés.
- Page d'ajout d'un nouvel avancement pour un projet.
- Page pour consulter tous les avancements pour chaque projet.
- Page pour modifier les informations d'un avancement.
- Page pour consulter tous les lots pour chaque projet de chaque entité.
- Page d'ajout d'un nouvel lot pour un projet.
- Page pour modifier les informations d'un lot.
- Page pour consulter tous les marchés pour chaque projet.
- Page d'ajout d'un nouvel marché pour un projet.
- Page pour modifier les informations d'un marché.

Chapitre 04 : Extension du système

4.4.3. Diagramme de cas d'utilisation :

Les diagrammes de cas d'utilisation modélisent le comportement d'un système et permettent de capturer ses exigences. Ils décrivent les fonctionnalités générales et la portée du système, ainsi que les interactions entre le système et ses acteurs. Ces diagrammes identifient ce que le système fait et comment les acteurs l'utilisent, sans toutefois révéler le fonctionnement interne du système [13].

Pour ce projet, chaque microservice est accompagné d'un diagramme de cas d'utilisation. La **figure 4.6** représente spécifiquement le diagramme de cas d'utilisation du microservice *Gestion projet actualisée*.

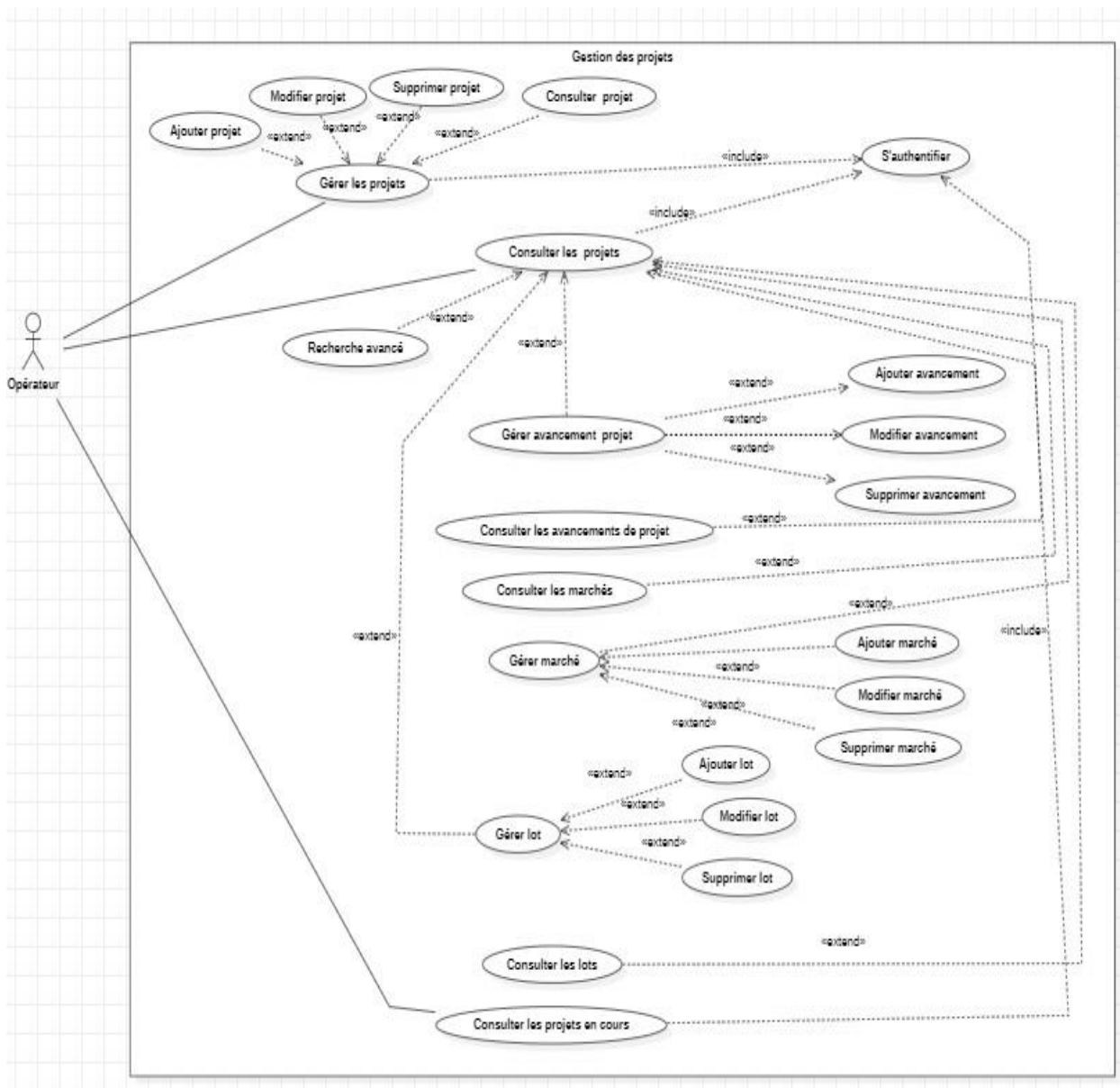


Figure 4.6 Diagramme de cas d'utilisation de microservice gestion des projets actualisés

Chapitre 04 : Extension du système

4.4.4. Diagramme de classes:

Les diagrammes de classes sont parmi les types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes, ses attributs, ses opérations et les relations entre ses objets. Les diagrammes de classes suivants illustrent respectivement le modèle adopté pour ce projet [13].

Pour ce projet, chaque microservice est accompagné d'un diagramme de classes. La **figure 4.7** représente spécifiquement le diagramme de cas d'utilisation du microservice *Gestion projet actualisée*.

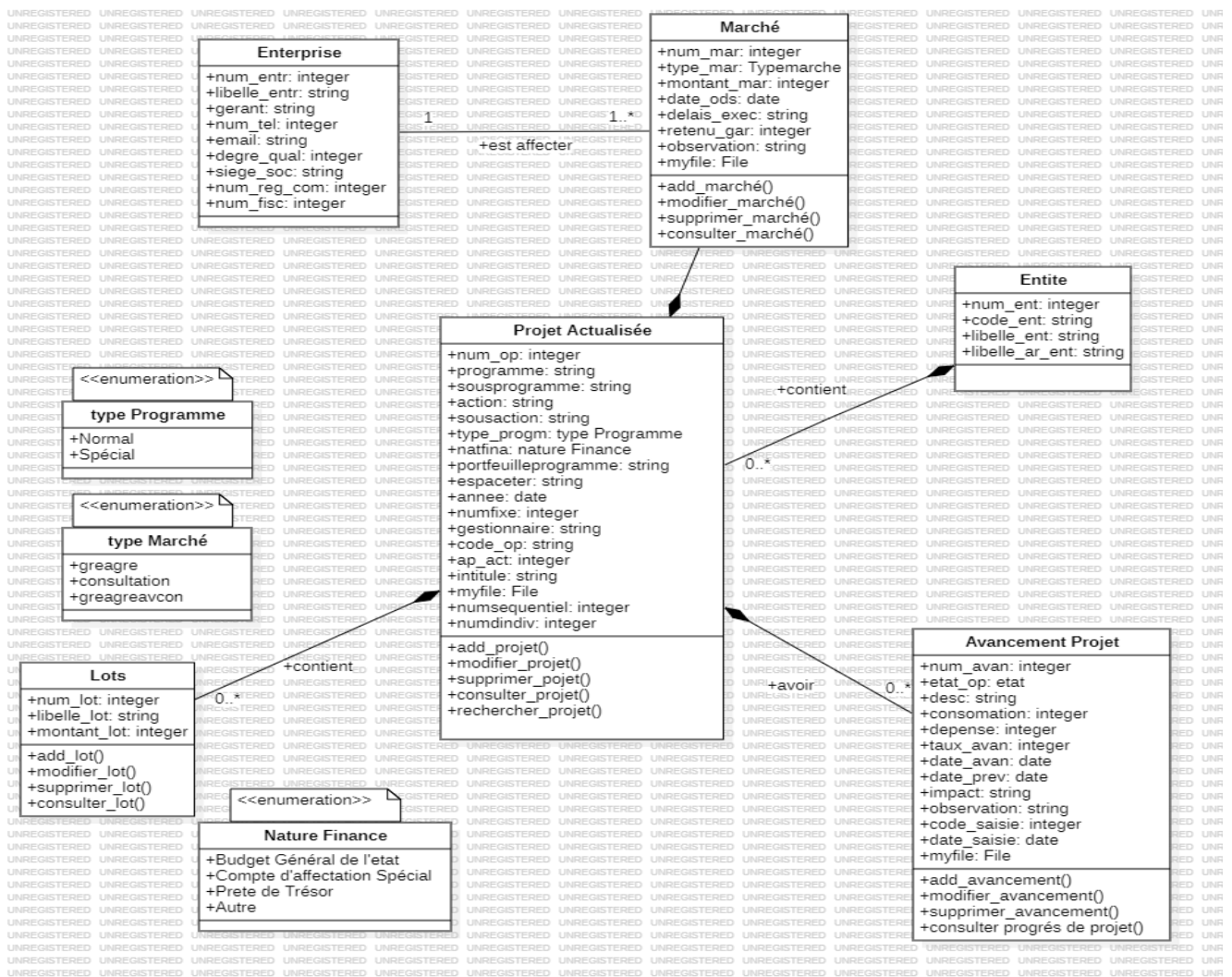


Figure 4.7 Diagramme de classes de microservice gestion des projets actualisées

4.4.5. Le passage du diagramme de classe au modèle relationnel :

❖ Les tables de la base de données :

En se basant sur les règles ci-dessus, nous avons converti les classes et leurs associations, à des tables dans la base donnée. Les tables générées sont :

Newprojet (num_op, programme, sousprogramme, sousaction, action, type_progm, natfina, portefeuilleprogramme, espaceter, annee, numfixe , gestionnaire , code_op, ap_act , Intitule , entite , myfile , numsequentiel , numdindiv, # num_ent) ,

Avancement Projet (num_avan, etat_op , desc, consommation, depense , taux_avan , date_avan, date_prev , impact , observation , date_saisie, code_saisie, myfile , # num_op),

Marché (num_mar, type_mar, desc_mar, date_ods, montant_mar, delais_exec, retenu_gar, observation , myfile ,#num_op, #num_entr),

Entreprise (num_entr , libelle_entr , gerant , num_tel , email , degre_qual ,siege_soc , num_reg_com , num_fisc),

Lots (num_lot , libelle_lot , montant_lot , # num_op) ,

Entite (num_ent, code_ent, libelle_ent, libelle_ar_ent).

4.5. Implémentation du système :

4.5.1. Présentation des outils et de langages utilisés :

Nous avons exploité une variété d'outils logiciels afin de mener bien notre projet. Ces divers outils nous ont facilité la création de l'ensemble des diagrammes présentés dans ce document et l'implémentation des différents programmes informatiques.

➤ Spring Boot:



Figure 4.8 Logo du framework spring boot

C'est un Framework qui permet de démarrer rapidement le développement d'applications [19].

➤ **Java :**



Figure 4.9 Logo du langage de programmation java

La technologie Java définit à la fois un langage de programmation orienté objet et une plateforme informatique [20].

➤ **Angular:**



Figure 4.10 Logo de Framework Angular

Angular est un Framework côté client, open source, basé sur Type Script, et codirigé par l'équipe du projet Angular à Google et par une communauté de particuliers et de sociétés. Il permet la création d'applications Web et plus particulièrement de ce qu'on appelle des SPA

(*Single Page Applications*) des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités [21].

➤ **Mobioos Forge :**



Figure 4.11 Logo de Framework Mobioos Forge

Mobioos Forge vous permettra de piloter vos projets d'implémentation logicielle en ajoutant une vue cartographique fonctionnelle [22].

➤ Visual Studio Code:



Figure 4.12 Logo de Framework Visual Studio Code

Visual Studio Code est un éditeur de code source léger mais puissant qui s'exécute sur votre bureau et est disponible pour Windows, macOS et Linux. Il est livré avec une prise en charge intégrée de JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages et environnements d'exécution (tels que C++, C#, Java, Python, PHP, Go, .NET) [23].

➤ Node.js :



Figure 4.13 Logo de Framework Node.js

C'est un environnement d'exécution single-thread, open source et multi plate-forme permettant de créer des applications rapides et évolutives côté serveur et en réseau [24].

➤ Npm :



Figure 4.14 Logo du gestionnaire des packages NPM

NPM est un gestionnaire de packages pour le langage de programmation JavaScript. Il s'agit du gestionnaire de packages par défaut pour l'environnement d'exécution JavaScript Node.js. Il se compose d'un client de ligne de commande, également appelé *npm*, et d'une base de données en ligne de packages publics et privés payants, appelée registre *npm* permettant à des milliers de développeurs d'accélérer leurs développements [25].



Figure 4.15 Logo du système de la base de données

PostgreSQL est un SGBD relationnelle objet open source avec plus de 35 ans de développement actif qui lui a valu une solide réputation de fiabilité, de robustesse des fonctionnalités et de performance [26].

4.5.2. Description du système :

Nous avons étendu une application Web existante. L'architecture générale de cette application se compose de trois parties principales, comme le montre la **figure 4.16** : la partie front-end, la partie back-end et les bases de données.

- **Partie front-end** : développée avec le framework Angular.
- **Partie back-end** : nous avons modifié cette partie en ajoutant un microservices, et nous avons utilisé Spring Boot, Spring Cloud et Spring Security.
- **Bases de données** : pour la gestion des bases de données, nous avons utilisé *PostgreSQL*.

Après avoir terminé l'application des LPL sur le système, nous avons ajouté un nouveau microservice. Cette application contient 7 microservices. Nous avons implémenté le microservice *Newprojet*.

Le microservice *Zuul* est un Api gateway permet la communication entre le front-end et les microservice *Newprojet*, *projet*, *Statistiques* et *User*.

Le microservice *Config* est utilisé pour la configuration de l'application.

Le microservice Eureka permet l'enregistrement des microservices connectés.

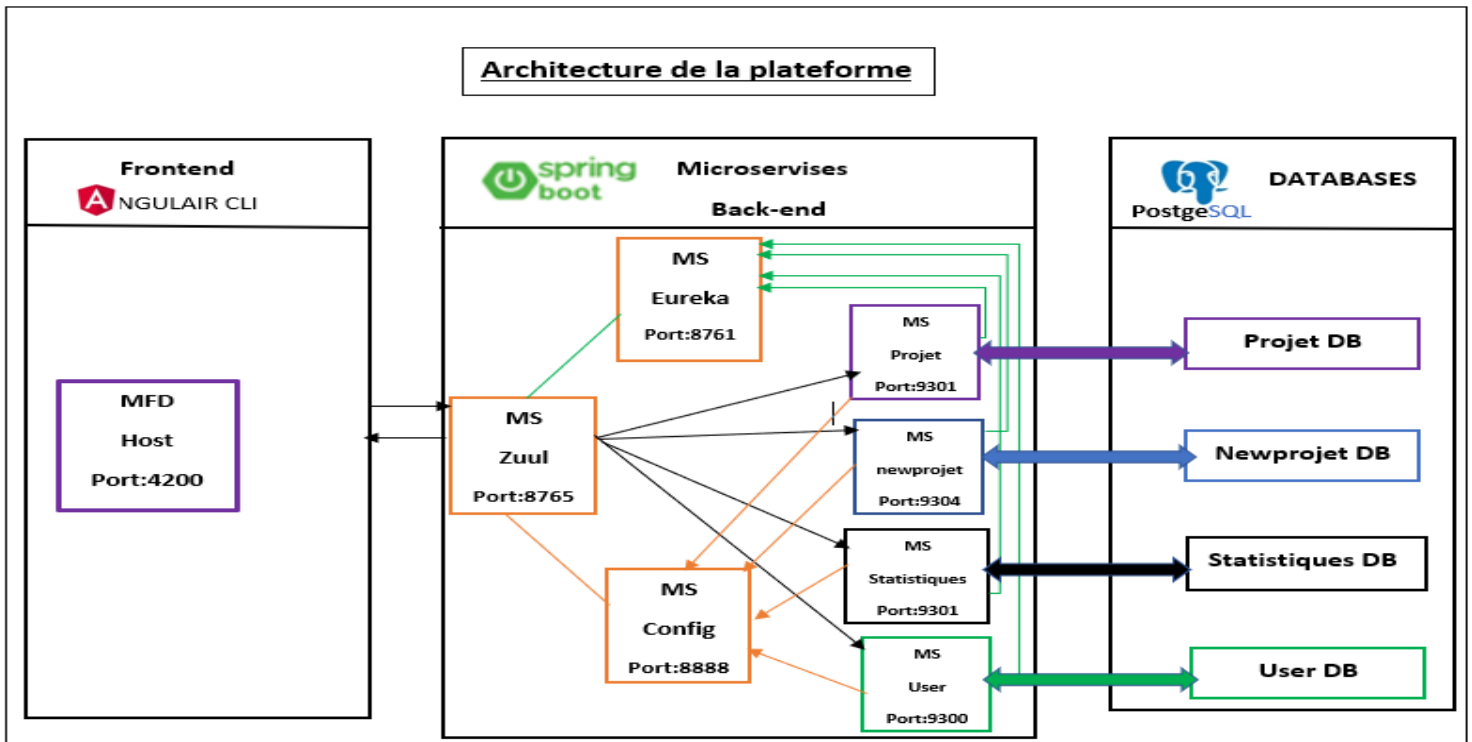


Figure 4.16 Architecture du système

4.5.3. Les interfaces

Dans cette partie, nous présentons les interfaces graphiques de notre application.

4.5.3.1. L'Interface de l'application de base avant modification :

- L'interface page d'accueil :



Figure 4.17 L'Interface de l'application de base la Page d'accueil

Chapitre 04 : Extension du système

- L'interface ajouter projet :

Ajouter Projet

Type de budget* Type Opération* Secteur*

Sous-secteur* Chapitre Article

Intitulé*

Code Opération* Date d'inscription* (mm/dd/yyyy)

AP actuelle*

Gestionnaire/Maitre de l'ouvrage*

Localisation

AP + notification(Fichier PDF) *

Figure 4.18 L'Interface de l'application de base d'ajouter projet

4.5.3.2. L'Interface de L'application après modification:

- L'interface page d'accueil :



Figure 4.19 L'Interface de L'application après modification la page d'accueil

Chapitre 04 : Extension du système

- L'interface ajouter projet :

The screenshot shows the 'Ajouter Projet' interface. The form includes the following fields and sections:

- Portefeuille programme:** education national
- Programme*:** education
- Sous-programme*:** Enseignement préparatoire et primaire
- Action:** L'activité n'est que floue
- Sous action:**
- Nature du financement*:** Budget general de l'état
- Type programme*:** Normal
- Intitulé*:** Acheter du matériel spécial
- Code Opération*:** 03/2024
- Année d'invalidisation*:** 02/04/2024
- Numéro Fixe*:** 224502014
- Numéro Séquentiel*:** N101104401404500004522029
- AP actuelle*:** 9900
- Gestionnaire/Maitre de l'ouvrage*:** Direction de la Programmation et du Suivi Budgétaire
- Espace territorial:** Naama

Buttons at the bottom: Enregistrer (green), Annuler (red).

PDF document on the right: AP + notification(Fichier PDF) *

Figure 4.20 L'Interface de L'application après modification d'ajouter projet

- L'interface de détail d'un projet :

The screenshot shows the 'Liste des projets' interface. It includes a search bar and a table of projects. The detailed view on the right provides the following information:

Détail du projet: Télécharger AP

- Programme:** éducation
- Sous-programme:** Enseignement préparatoire et primaire
- Action:** L'activité n'est que floue
- Sous action:** null
- Code:** 03/2024
- Numéro Séquentiel:** N101104401404500004522029
- Numéro Fixe:** 224502014
- Intitulé:** Acheter du matériel spécial
- Date :** 2024-02-04T08:00:00.000+00:00
- AP Actuelle:** 9900
- Nature du financement:** PSD **Type Programme:** Construction
- Gestionnaire:** Direction de la Programmation et du Suivi Budgétaire
- Portefeuille programme:** Naama
- Espace territorial:** éducation national

Buttons: + Lot (blue)

Libellé lot	Montant	Action
-------------	---------	--------

Figure 4.21 L'Interface de détail d'un projet

Chapitre 04 : Extension du système

- L'interface de la liste des projets :

Navigation: Accueil | Gestion projets | Gestion projets actualisés | Etat PEC | Evolution | Situation programmes | aya | Déconnexion

Liste des projets

Rechercher par titre Recherche

Code Opération	Intitulé	AP	annee Inscription
03/2024	Acheter du matériel ...	9900	2024-02-04

S.V.P cliquer sur un projet...

« 1 »

tems : 3

Figure 4.22 L'Interface de la liste des projets

4.6. Conclusion :

Dans ce chapitre, nous avons présenté une introduction approfondie sur les microservices, le processus unifié (UP) et le langage de modélisation UML. Nous avons également exploré la conception du système. Enfin, nous présentons l'implémentation du système.

Conclusion générale

Conclusion générale

Dans ce mémoire, nous avons migré une application open source de gestion de projets vers les Lignes de Produits Logiciels en utilisant la plateforme *Mobioos Forge*. De plus, nous avons ajouté un nouveau module de gestion de projets conforme à la *Loi organique 18-15* en développant un nouveau microservice. L'objectif de cette application est de permettre aux responsables de contrôler et de superviser toutes les opérations et de les aider également à prendre les décisions appropriées.

Afin de réaliser notre projet, nous avons commencé par étudier le domaine de la gestion de projet en effectuant un stage pratique chez *Algérie Télécom* et en explorant également l'application open source. Ensuite, nous avons maîtrisé les concepts fondamentaux des *Lignes de Produits Logiciels* ainsi que la plateforme *Mobioos Forge*. Puis, nous avons généré les Lignes de Produits Logiciels à partir d'une application open source dédiée à la gestion de projets et avons discuté les étapes clés en fournissant des statistiques détaillées sur cette transition. Finalement, nous avons étendu cette application et l'avons adaptée aux évolutions de la réglementation en matière de gestion de projets en ajoutant un nouveau microservice. Plus précisément, nous avons présenté la nouvelle architecture microservices adoptée pour notre projet actualisé, ainsi que la conception du système à travers divers diagrammes, tels que les cas d'utilisation, les classes et le modèle relationnel. En plus, nous avons présenté les outils, les langages de programmation, l'environnement de développement utilisés et quelques interfaces clés de l'application.

Ce travail a été pour nous une expérience très enrichissante, nous permettant d'approfondir nos connaissances en développement d'applications basées sur les Lignes de Produits Logiciels et l'architecture Microservices.

Dans un travail futur, nous planifions de déployer ce système dans les institutions algériennes. Nous planifions également de développer un nouveau module intelligent permettant l'évaluation d'un marché public avant de soumettre son dossier au contrôleur budgétaire en se basant sur des techniques d'apprentissage automatique.

Bibliographie

- [1] T. ZIADI, Manipulation de Lignes de Produits en UML, Institut de Formation en Sciences de l'Ingénieur de l'Université de Rennes 1, 2004.
- [2] Klaus Pohl, Günter Böckle, and F. Van, Software product line engineering : foundations, principles, and techniques. Berlin ; New York: Springer, 2005.
- [3] T. Ben Rahouma Aouina, composition des modèles de lignes de produits logiciels, université paris-sud école doctorale d'informatique, 2012.
- [4] S. Apel, D. Batory, and C. Kastner, Feature-Oriented Software Product Lines. 2013.
- [5] Y. Xu and B. Traverson, "Procédés de réutilisation pour les lignes de produits logiciels," *Revue des Sciences et Technologies de l'Information - Série ISI: Ingénierie des Systèmes d'Information*, vol. 13, no. 2, pp. 33–54, 2008.
- [6] Pohl.K , Böckle.G et Van Der Linden.F, Software Product Line Engineering: Foundations, Principles, and Techniques, Springer,2005.
- [7] C. W. Krueger, Software reuse, *ACM Computing Surveys*, vol. 24, no. 2, p. 131–183, juin 1992.
- [8] Pohl.K , Böckle.G et Van Der Linden.F, Software Product Line Engineering: Foundations, Principles, and Techniques, Springer,2005.
- [9] D. M. Weiss and C. Tau, Software Product-line Engineering. Addison-Wesley Professional, 1999.
- [10] P. Clements and J. Bergey, "The U.S. Army's Common Avionics Architecture System (CAAS) Product Line: A Case Study," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, CMU/SEI-2005-TR-019, Sep. 2005.
- [11] K. Ghallab, T. Ziadi, and Z. Chalal, "Migrating Individual Applications into Software Product Lines using the Mobioos Forge Platform," in *Proceedings of the Asia Pacific Software Engineering Conference*, Taipei, Taiwan, Dec. 2023, pp. 100–110.
- [12] Forge Documentation, documentation.mobioos.ai. <https://documentation.mobioos.ai/?id=what-is-mobioos-forge>.
- [13] M.Babacar DIAGO ,2022. Automatisation du système de restauration de l'université Assane Seck de

Bibliographie

Ziguinchor, de la vente des tickets au contrôle des accès aux restaurants, mémoire de fin d'étude Master (Génie Logiciel), Université Assane Seck de Ziguinchor.

- [14] AMZALI Mouloud et ABBASSEN Mouhcene ,2023. Revolutionizing Job Seeking: Development of a Microservices-Based Platform with Domain-Driven Design, Mémoire préparé en vue de l'obtention du diplôme de Master, Option Networks and Distributed Systems (RSD), University of Abdelhamid Mehri – Constantine 2.
- [15] Boucheloukh Ghazlane et Sekfali Aya ,2023. Conception et Réalisation d'une Plateforme d'information pour le Suivi du développement du Secteur de l'enseignement Supérieur, Mémoire préparé en vue de l'obtention du diplôme de Master, Option Informatique, Sciences et Technologies de l'Information et de la Communication (STIC), Centre Universitaire Abdellhafid Boussouf - Mila.
- [16] Boumeshal Yahya et Meskine Mourad ,2022. Conception et Réalisation d'un Système d'Information Sécurisé basé sur la Blockchain, Mémoire préparé en vue de l'obtention du diplôme de Master, Option Informatique, Sciences et Technologies de l'Information et de la Communication (STIC), Centre Universitaire Abdellhafid Boussouf - Mila.
- [17] Pascal Roques; Franck Vallée édition Eyrolles; UML 2 en action: De l'analyse des besoins à la conception.
- [18] Architecture microservices. (s. d.). linkedin.com
- [19] *Spring Boot*. (s. d.). Spring Boot. <https://projects.spring.io/spring-boot/>.
- [20] « Java : définition ». JDN : E-business, FinTech, Big Data, IoT, tendances média, décideurs... <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203555-java-definition/>.
- [21] Contributeurs aux projets Wikimedia. « Angular — Wikipédia ». Wikipédia, l'encyclopédie libre. <https://fr.wikipedia.org/wiki/Angular>.
- [22] MOBIOOS FORGE: <https://documentation.mobioos.ai/>
- [23] Visual Studio Code : <https://code.visualstudio.com/docs> .
- [24] « Node.js ». Node.js. <https://nodejs.org/en/>.
- [25] « npm Docs ». Npm Docs. <https://docs.npmjs.com/> .
- [26]« PostgreSQL ». PostgreSQL. <https://www.postgresql.org/>.
- [27] <https://github.com/Mobioos/Forge-BankWeb>.

