



Filière :

Développement des systèmes d'information

Rapport de Mini projet

Sous thème :

La Gestion des matériels

Chez ONOUSC

Réalisé par :

Ziyad Aya

Membre de jury :

M. ALhaiane Hamid

Année scolaire : 2024/2025

Remerciement

En préambule à ce rapport, nous souhaitons adresser ici tous nos remerciements aux personnes qui nous ont apporté leur aide et qui ont ainsi contribué à l'élaboration de ce rapport.

Nous tenons à remercier dans un premier temps, toute l'équipe pédagogique de BTS KENITRA, et les intervenants professionnels responsables de la formation développement des systèmes d'informations, pour avoir assuré la partie théorique et pratique et de celle-ci.

Nous exprimons notre sincère gratitude au membre du jury M. Alhaiane hamid qui a accepté d'évaluer ce projet de fin d'étude.

Chapitre 1 : Contexte générale du projet.

I. Introduction :

« Le projet est un effort complexe pour atteindre un objectif bien spécifique, devant respecter un échéancier et un budget »

L'étude du projet est une démarche stratégique visant à organiser le bon déroulement d'un projet et d'assurer la conduite de toutes les phases qui le constituent.

Une étude complète et efficace conduit généralement à la réussite d'un projet. Cette étude fera donc l'objet de notre premier chapitre qui sera consacré à la présentation du projet ainsi que la définition de notre langage et méthodologie de développement.

1. Objectifs de(s) Mini-projet(s) :

- Application des connaissances : le mini-projet vise à permettre aux étudiants d'appliquer les connaissances théoriques acquises tout au long de leur formation à des problématiques concrètes. Il s'agit de mettre en pratique les compétences et les concepts étudiés dans un contexte réel.
- Approfondissement d'un sujet : le projet de fin d'étude offre aux étudiants l'opportunité d'approfondir un domaine spécifique qui les intéresse. Il peut s'agir d'une discipline particulière, d'une question de recherche, d'un problème technique ou d'un sujet d'étude choisi par l'étudiant lui-même.

2. Présentation de l'entreprise :

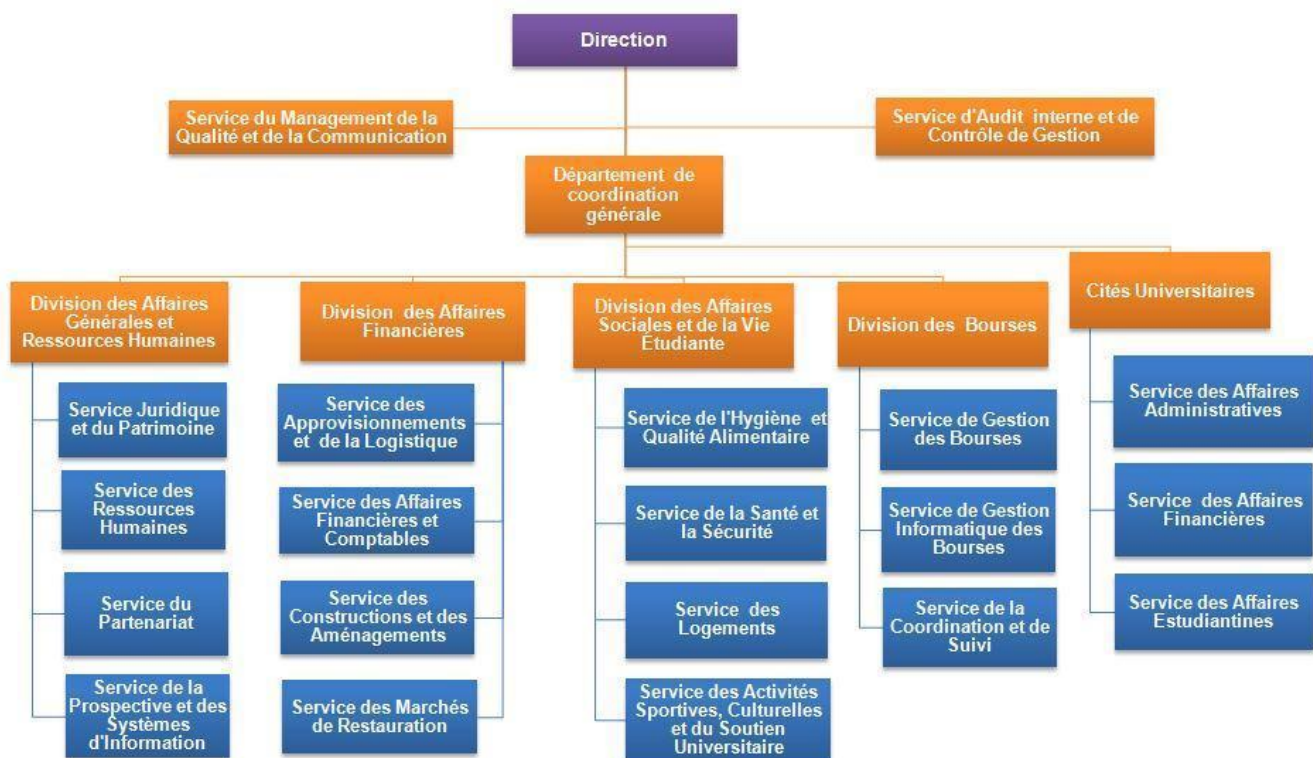
L'Office National des Œuvres Universitaires, Sociales et Culturelles est une institution créée par le [Dahir N° 1.01.205 du 30 Août 2001](#). Il bénéficie de l'autonomie administrative et financière et est placé sous la tutelle du Ministre de l'Enseignement supérieur, de la Recherche scientifique et de l'Innovation. Parmi ses principales missions telles qu'elles sont précisées dans l'article 3 du Dahir de sa création sont :

- Le versement aux étudiants des bourses d'études dont les crédits budgétaires sont mis à sa disposition par les départements ministériels intéressés, dans les conditions qui seront fixées par voie réglementaire.
- La contribution à la promotion du sport universitaire national.
- La prise de participations, conformément à la législation en vigueur, dans toutes entreprises entrantes, par leur objet, dans le cadre de ses activités.
- Le développement d'activités universitaires afin de renforcer et compléter la formation des étudiants dispensée dans les établissements universitaires.

3. Organigramme :

L'organigramme de l'Office a été visé en Mai 2016. Il comporte au niveau de l'administration centrale :

- 1 département de la coordination générale.
- 4 divisions.
- 18 services.



II. Cahier des charges :

1. Présentation du projet :

Mon projet est né afin de simplifier non seulement le processus de vérification de la connectivité des matériels, mais aussi la possibilité de créer une demande de fourniture en adressant un fournisseur bien spécifique. Ce qui permet de minimiser le temps et l'effort fournis par l'admin.

2. Problématique :

Il est à noter que la vérification des appareils et la demande de fourniture est toujours manuelle, ce qui nécessite des efforts considérables de notre part en vue d'aider l'admin.

Pour combler cette lacune, l'admin recommande la remise en place d'urgence d'une application informatique desktop pour améliorer ses méthodes de vérification et demande de nouveau matériel.

3. Fonctionnalités attendues :

Notre application doit répondre aux exigences suivantes :

- L'administrateur doit pouvoir se connecter en saisissant un identifiant et un mot de passe.
- Il doit exister un espace où l'administrateur peut vérifier la connectivité des matériels.
- L'administrateur doit pouvoir ajouter et mettre à jour les commandes.

III. Planning et conduite du projet :

1. Gestion de projet :

La gestion de projet est une démarche visant à organiser de bout en bout le bon déroulement d'un projet. Concourant à un même projet.

Lorsque la gestion de projet porte sur un ensemble de projets concourant à un même objectif, on parle de gestion de programme.

Le Management de projet est un management qui se caractérise par :

- L'irréversibilité des opérations des participants.
- Un fort degré de liberté des actions des participants.
- Une organisation vouée à être évolutive et temporaire.

2. Estimation de la charge : COCOMO est un modèle permettant de définir une estimation de l'effort à fournir dans un développement logiciel et la durée que ce dernier prendra en fonction des ressources allouées.

Estimation de KLOC Pour un projet simple :

- **Cas d'utilisation simples :** 50 à 100 lignes de code par cas

- Si on suppose qu'il y a 4 cas d'utilisation, et qu'ils sont relativement simples, on pourrait avoir par exemple **300 lignes de code** (soit 0,3 KLOC).

Calcul de la charge et du temps de développement :

1. **Charge** = $2,4 \times (\text{KLOC})^{1,05}$

Pour **KLOC = 0,3**:

Charge = $2,4 \times (0,3)^{1,05} \approx 2,4 \times 0,295 \approx 0,708$ mois

2. **Temps de développement (TDEV)** = $2,5 \times (\text{Charge})^{0,38}$

Pour **Charge = 0,708** :

TDEV = $2,5 \times (0,708)^{0,38} \approx 2,5 \times 0,868 \approx 2,17$ mois

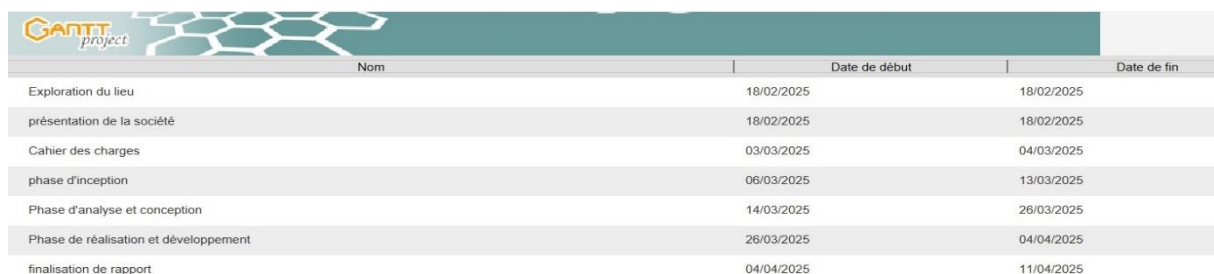
Résultat : Charge $\approx 0,71$ mois TDEV $\approx 2,17$ mois

3. Diagramme de Gantt :

Le diagramme de Gantt est un outil utilisé (souvent en complément d'un réseau PERT) en ordonnancement et gestion de projet et permettant de visualiser dans le temps les diverses tâches liées composant un projet (il s'agit d'une représentation d'un graphe connexe, value et orienté). Il permet de représenter graphiquement l'avancement du projet.

Dans mon projet j'ai utilisé l'outil Gantt Project.

Diagramme de Gantt :



| Nom | Date de début | Date de fin |
|---------------------------------------|---------------|-------------|
| Exploration du lieu | 18/02/2025 | 18/02/2025 |
| présentation de la société | 18/02/2025 | 18/02/2025 |
| Cahier des charges | 03/03/2025 | 04/03/2025 |
| phase d'inception | 06/03/2025 | 13/03/2025 |
| Phase d'analyse et conception | 14/03/2025 | 26/03/2025 |
| Phase de réalisation et développement | 26/03/2025 | 04/04/2025 |
| finalisation de rapport | 04/04/2025 | 11/04/2025 |

IV. Langage et méthodologie de conception :

Une méthodologie de développement est un cadre utilisé pour structurer, planifier et contrôler le développement d'une application. C'est le fait de modéliser un système avant sa réalisation pour bien comprendre son fonctionnement et assurer sa cohérence.

Un modèle est ainsi un facteur de réduction des couts et des délais. Il est donc indispensable pour assurer un bon niveau de qualité de produit dont la maintenance est efficace. Les méthodes agiles sont des méthodes adaptées pour les projets complexes ou de taille moyenne qui durent longtemps et qui

nécessitent l'intervention d'une grande équipe de développement qui reste tout au long du projet en interaction directe avec le client pour résoudre les tâches complexes et réagir contre l'évolution permanente du contexte. Vu que ce n'est pas le cas de notre projet, les méthodes agiles sont à écarter. Recours aux Processus Unifiés.

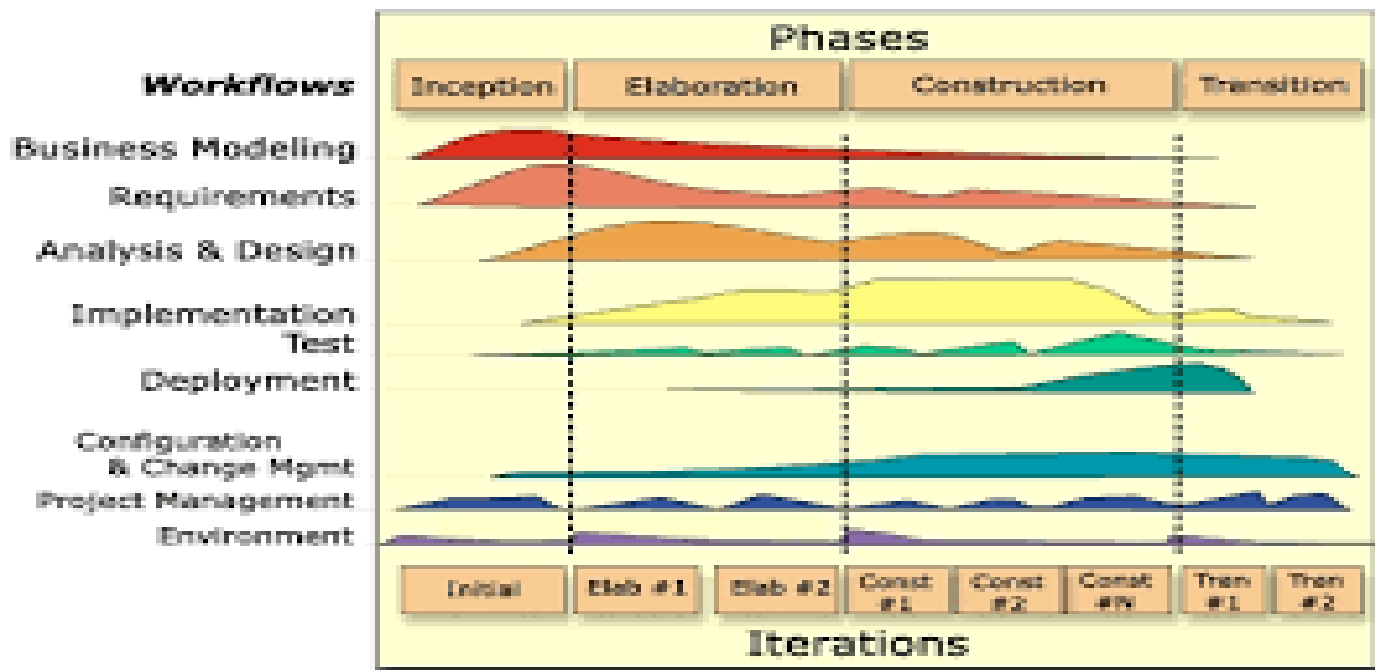
✚ Le Processus Unifié (UP) est un processus de développement logiciel organisée suivant les quatre phases suivantes :

1. La phase d'initialisation permet de décider la poursuite du projet ou son arrêt.
 2. La phase d'élaboration poursuit trois objectifs principaux en parallèle : (identifier et décrire, construire, lever les risques).
 3. La phase de construction consiste à concevoir et implémenter l'ensemble des éléments opérationnels.
 4. En fin, la phase de transition permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux.
- Les implémentations du Processus Unifié sont représentées dans le tableaux suivant :

| Les principes fondamentaux De Processus Unifié (UP) | RUP | 2TUP: Two Tracks Unified Process |
|---|--|---|
| | Le Rational UNIFIED Process (RUP) est l'une des plus célèbres implémentations de la méthode PU. Elle permet de donner un cadre au développement logiciel, répondant aux exigences fondamentales préconisées par les créateurs d'UML : -Une méthode de développement doit être centrée sur l'architecture logicielle. - Elle doit être itérative et incrémentale. | Il propose un cycle de développement en Y, qui dissocie les aspects techniques des aspects fonctionnels. Il commence par une étude préliminaire qui consiste essentiellement à identifier les acteurs, les messages, produire le cahier des charges et modéliser le contexte (le système est une boîte noire, les acteurs l'entourent et sont reliés à lui). le processus s'articule ensuite autour de trois phases essentielles : une branche fonctionnelle et une phase de réalisation. |

Nous adaptons alors le processus RUP pour la suite de notre travail.

1. Mise en pratique du RUP :



- Le développement d'un logiciel se fait bien souvent de manière itérative jusqu'à conduire à des livrables incrémentaux du système. Cette méthode va alors servir à organiser la modélisation de la partie développement. En effet, au-delà de l'aspect théorique, c'est la mise en œuvre de la méthode RUP qui est intéressante pour la qualité du travail. Les plus de la méthode RUP, ce sont ses revues fréquentes avec toutes les parties prenantes : le produit est ainsi construit en sécurité. Chaque itération est considérée comme un livrable et l'on peut revenir sur les phases précédentes et réadapter le logiciel. Le retour utilisateur prend une grande place dans la construction.

2. Le formalisme UML : UML est considéré comme le langage standard de conception orienté objet, il est un formalisme et pas une méthode. Il s'en suit qu'il définit un ensemble d'éléments de modélisation et une notation graphique pour modéliser les systèmes et ne décrit pas les étapes à suivre pour le faire.

Conclusion : Dans ce premier chapitre, nous avons fait une étude de l'existant nous permettant de bien entamer notre projet. Nous avons comparé les différentes méthodes de développement afin de clôturer ce chapitre par une présentation de la méthodologie RUP que nous allons adopter pour le développement de notre projet.

Le chapitre suivant sera consacré à l'analyse de notre projet et à la capture des besoins.

**“Requirements Is about understanding what a software system shall do
& user expectations “**

Chapitre 2 : Phase d’Inception

I. Introduction : Cette phase consiste à comprendre le contexte du système. Il s’agit de déterminer les fonctionnalités et les acteurs les plus pertinents, de préciser les risques les plus critiques et d’identifier les cas d’utilisation initiaux. Ceci dit, notre description va sembler trop détaillée pour une première phase de processus.

II. Capture des besoins :

1. Définition des besoins fonctionnels : Il s’agit des fonctionnalités du système. Ce sont les besoins spécifiant un comportement d’entrée/sortie du Système.

Le système à concevoir doit permettre à l’utilisateur d’effectuer les opérations suivantes :

S’authentifier : Pour accéder aux différentes fonctionnalités, l’admin doit être authentifié par la saisie d’un identifiant et un mot de passe unique.

Vérifier la connectivité d’un appareil : Après une authentification passée, l’admin peut vérifier la connectivité d’un appareil en entrant une adresse IP et cliquant sur un bouton dédié, une réponse (est générer après le pingue de cette adresse) indique si le matériel à tester est connecter ou non.

Gérer des commandes de fourniture : Un admin authentifié peut créer (ou modifier) des commandes de fourniture pour remplacer le manque des matériels irréparables, au premier temps une vérification des champs faite avant l’affichage d’un message de confirmation de l’opération souhaitée.

| Référence | Fonction |
|-----------|-------------------------------------|
| R1 | Authentifier |
| R2 | Vérifier la connectivité |
| R3 | Créer une commande de fourniture |
| R4 | Modifier une commande de fourniture |

On peut utiliser un SYSML Pour modéliser les exigences :

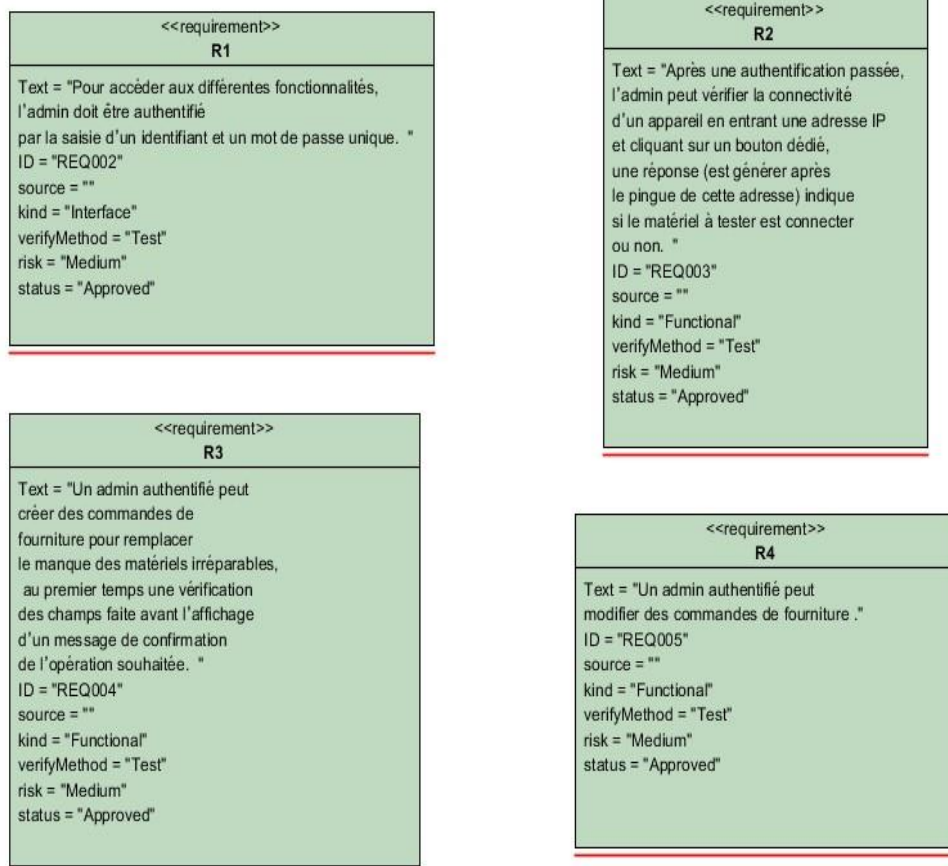


Figure 1 : SYSML des exigences

III. identification des acteurs :

- Admin : celui qui est chargé de la vérification de la connectivité des matériels, gestion des commandes de fournitures.

3. Identification des cas d'utilisation :

A partir des exigences précédentes nous pouvons tirer les cas d'utilisation suivants :

- UC1 : vérifier la connectivité d'un matériel
- UC2 : créer une demande de fourniture
- UC3 : modifier une demande de fourniture
- UC4 : Authentification

4. Regroupement des exigences fonctionnelles par intentions des acteurs :

| Exigence Fonctionnelle | Intention d'acteur | Acteur |
|---|--|--------------|
| R1 : Pour accéder aux différentes fonctionnalités l'Admin doit être authentifié par la saisie d'un identifiant et un mot de passe unique. | S'authentifier | Admin |
| R2 : Après une authentification passée, l'admin peut vérifier la connectivité d'un appareil en entrant une adresse IP et cliquant sur un bouton dédié, une réponse (est générée après le ping de cette adresse) indique si le matériel à tester est connecté ou non. | Vérifier la connectivité d'un matériel | Admin |
| R3 – R4 : Un admin authentifié peut gérer des commandes de fourniture pour remplacer le manque des matériels irréparables, au premier temps une vérification des champs faite avant l'affichage d'un message de confirmation de l'opération souhaitée. | Créer une commande de fourniture. | Admin |
| | Mettre à jour une commande de fourniture. | Admin |

4. Diagramme de cas d'utilisation :

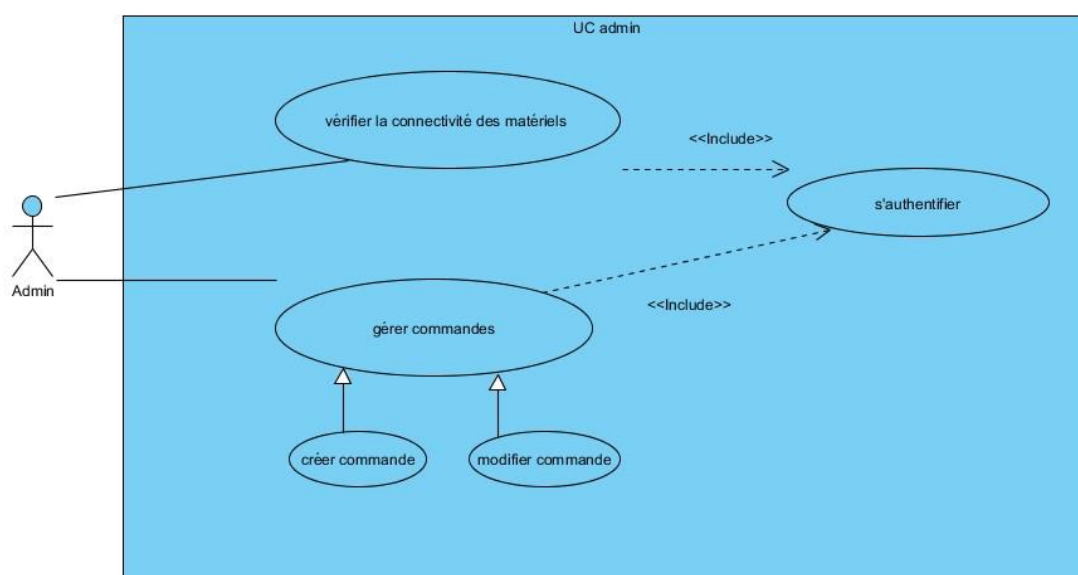


Figure 2 : use case diagramme

Chapitre 3 : Phase d'Analyse et de conception

- I. Introduction :** Cette phase poursuit la tâche entamée dans la phase d'Inception, cette phase vise à identifier et décrire la majorité des besoins utilisateurs, construire l'architecture de base du système et lever les risques majeurs du projet.

Au niveau de cette phase, j'analyse les différents cas d'utilisation puis j'essaie de les détailler pour comprendre chaque fonctionnalité attendue du système en examinant tous les scénarii comptés des scénarios exceptionnels et des erreurs.

Classement des cas d'utilisation : Pour classer les cas d'utilisation, il est important de décider qu'elles sont les cas les plus prioritaires et qu'elles sont les cas qui présente de grands risques techniques.

| Cas d'utilisation | Priorité | Risque |
|--|----------|--------|
| UC1 : authentification | Basse | Moyen |
| UC2 : Vérifier la connectivité d'un matériel | Haute | Haut |
| UC3 : Créer une commande de fourniture. | Moyenne | Moyen |
| UC4 : Traiter une demande de fourniture | Moyenne | Moyen |

Les cas d'utilisations qui ont une priorité élevée et un risque élevé sont placés dans les premières itérations, on obtient ainsi le classement suivant :

| Cas d'utilisation | Priorité | Risque | Itération |
|--|----------|--------|-----------|
| UC1 : authentification | Basse | Moyen | 4 |
| UC2 : Vérifier la connectivité d'un matériel | Haute | Haut | 1 |
| UC3 : Créer une commande de fourniture. | Moyenne | Moyen | 2 |
| UC4 : Traiter une demande de reproduction | Moyenne | Moyen | 3 |

Itération numéro 1 : UC1 : Vérifier la connectivité d'un matériel

Description Textuelle :

| | |
|-----------------------|---|
| Nom | Vérifier la connectivité des matériels |
| Objectif | Permettre à l'Admin d'effectuer une vérification de la connectivité des matériels. |
| ID | 1 |
| Acteurs principaux | Admin |
| Date | 01/03/2025 |
| Responsable | Aya |
| Statut | Proposé |
| Version | V1.0 |
| Précondition | Admin authentifié |
| Scénario nominal | <ol style="list-style-type: none">1. SYSTEM affiche la liste des matériels2. Admin saisit l'adresse IP de l'un des matériels3. SYSTEM vérifie les données saisies.4. SYSTEM affiche les résultats de l'opération.5. Fin |
| Scénario alternatif : | - indiqué l'obligation de remplir le champs |
| Post condition : | Résultat fournie |

6. Diagramme de Séquence : Pour le cas d'utilisation de l'itération 1 :

| |
|---|
| 1.admin saisit l'adresse IP du matériel |
| 2. SYSTEM vérifie la connectivité du matériel |
| 2.1 if matériel connected |
| 2.1.1 SYSTEM affiche un message de confirmation |
| 2.2 else |
| 2.2.1 SYSTEM affiche un message de la non-connectivité |
| end if |

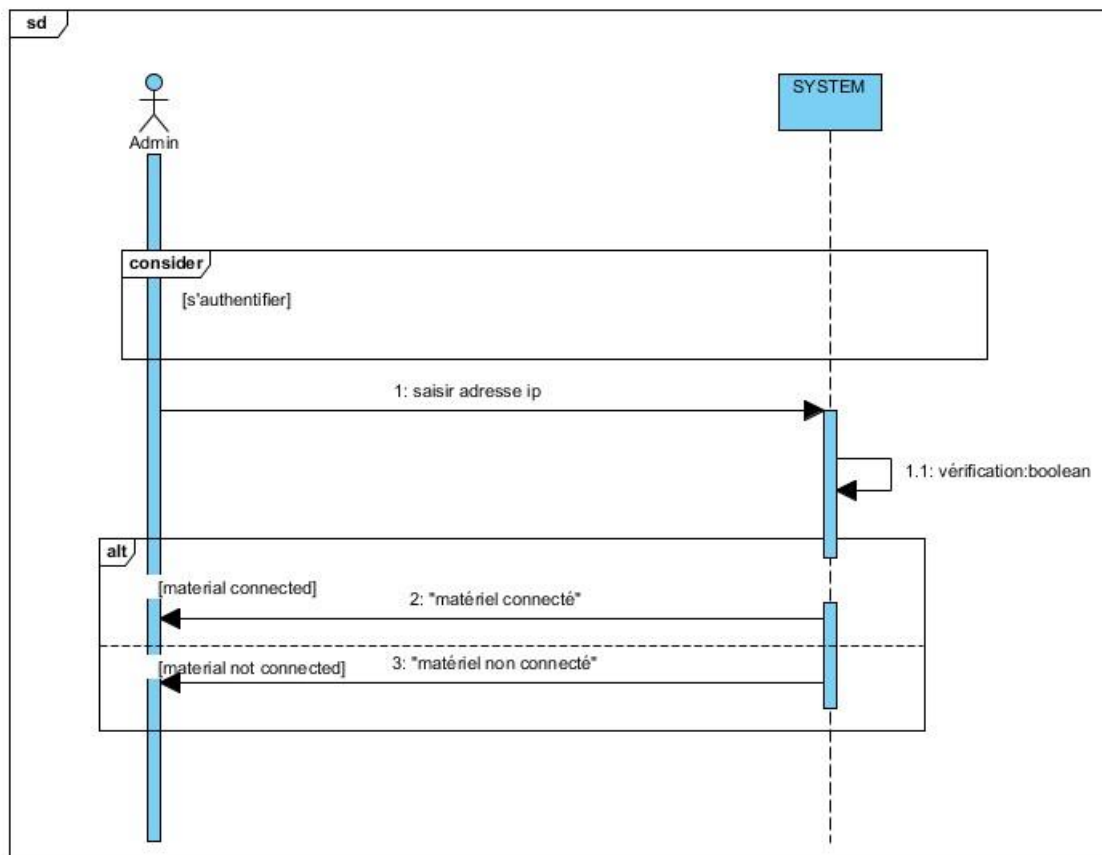


Figure 3 : SD : vérifier la connectivité des matériels

- 7. Maquette :** l'interface comporte une fonctionnalité de saisir une adresse IP et un bouton "vérifier" pour traiter l'information entrée.

rechercher un matériel par son adresse ip :

| idmat | nom | ip | marque |
|-------|-----|----|--------|
| | | | |
| | | | |
| | | | |

entrer une adresse ip :

Figure 4 : prototype de l'interface admin relative à uc1

8. Trouver les concepts du Domain :

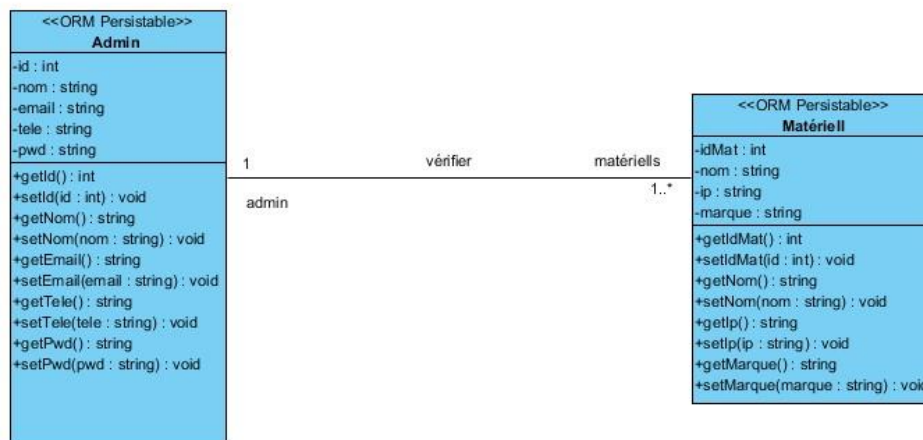


Figure 5 : CD : vérifier la connectivité des matériels

-Il s'agit d'un modèle de données pour un système de vérification de la connectivité des matériels où un admin peut faire des vérifications liées à un certain matériel.

9. Diagramme de classes participantes :

Dans un diagramme de classe participante, on trouve trois types de classes qui sont :

1- **Control Class** contiennent la logique métier pour gérer les interactions entre les classes entité et les classes frontières. Elles sont responsables de la coordination du flux de contrôle dans le système.

2- **ENTITY Class** représentent des objets ou des concepts du monde réel qui ont une existence et une identité persistante dans le système. Elles encapsulent généralement les données et le comportement lié à ces objets.

3- **BOUNDARY Class** agissent comme une interface entre le système et son environnement externe, elles sont responsables de la collecte des entrées de l'utilisateur et de l'affichage des résultats.

-Les classes de contrôle coordonnent ces interactions en utilisant les classes frontières pour communiquer avec l'utilisateur et les classes entité pour accéder et manipuler les données.



Figure 6 : DCP effectuer une vérification de connectivité d'un matériel

10. Diagramme d'Interaction :

Après avoir trouvé les classes participantes, nous allons établir un diagramme d'interaction (Une variante du diagramme de séquence) pour modéliser les interactions entre ces objets qui participe à la réalisation du cas d'utilisation étudié.

Le diagramme d'interaction pour le UC1 se présente comme suit :

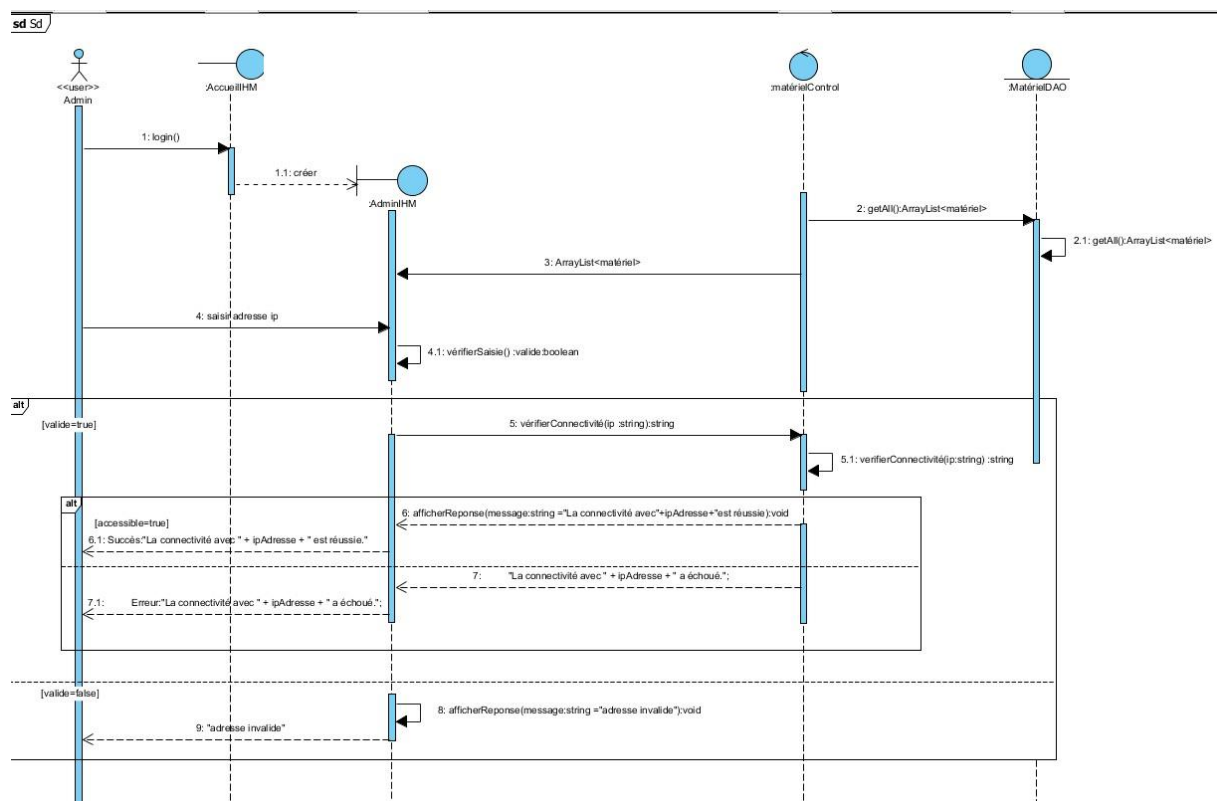


Figure 7 : diagramme d'interaction UC1

Itération numéro 2 : UC2 : Créer une demande de fourniture

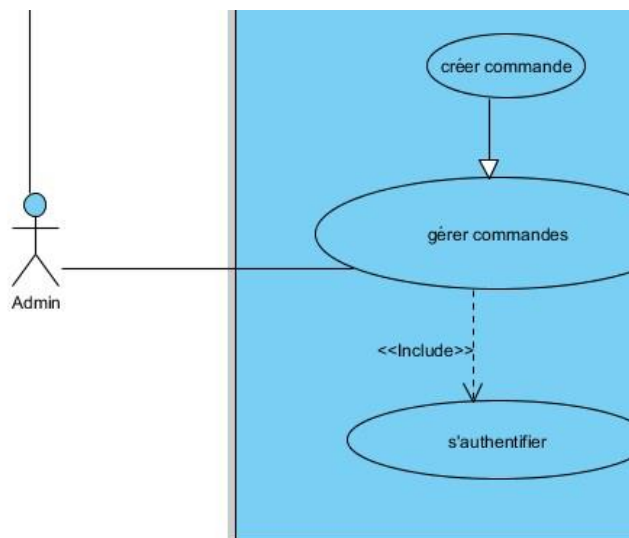


Figure 8 : Cas d'utilisation 2

Description Textuelle :

| | |
|-----------------------|---|
| Nom | Créer une demande de matériel |
| Objectif | Permettre à l'Admin d'effectuer une demande de matériel |
| ID | 1 |
| Acteurs principaux | Admin |
| Date | 01/03/2025 |
| Responsable | Aya |
| Statut | Proposé |
| Version | V1.0 |
| Précondition | Admin authentifié |
| Scénario nominal | <ol style="list-style-type: none">1. Admin saisit les informations du matériel2. SYSTEM vérifie les données saisies.3. SYSTEM ajout la demande4. Fin |
| Scénario alternatif : | 3-a. SYSTEM affiche un message d'erreur |
| Post condition : | Demande ajoutée. |

1. Diagramme de séquence : Flow of events :

| |
|---|
| 1. Admin s'authentifier |
| 2. Admin clique sur onglet mes demandes |
| 3.Admin saisit les informations de la demande |
| 4.SYSTEM vérifier les données |
| 4.1 if données valid |
| 4.1.2 SYSTEM ajout la nouvelle demande |
| 5 else |
| 5.1 SYSTEM affiche message d'erreur |
| end if |

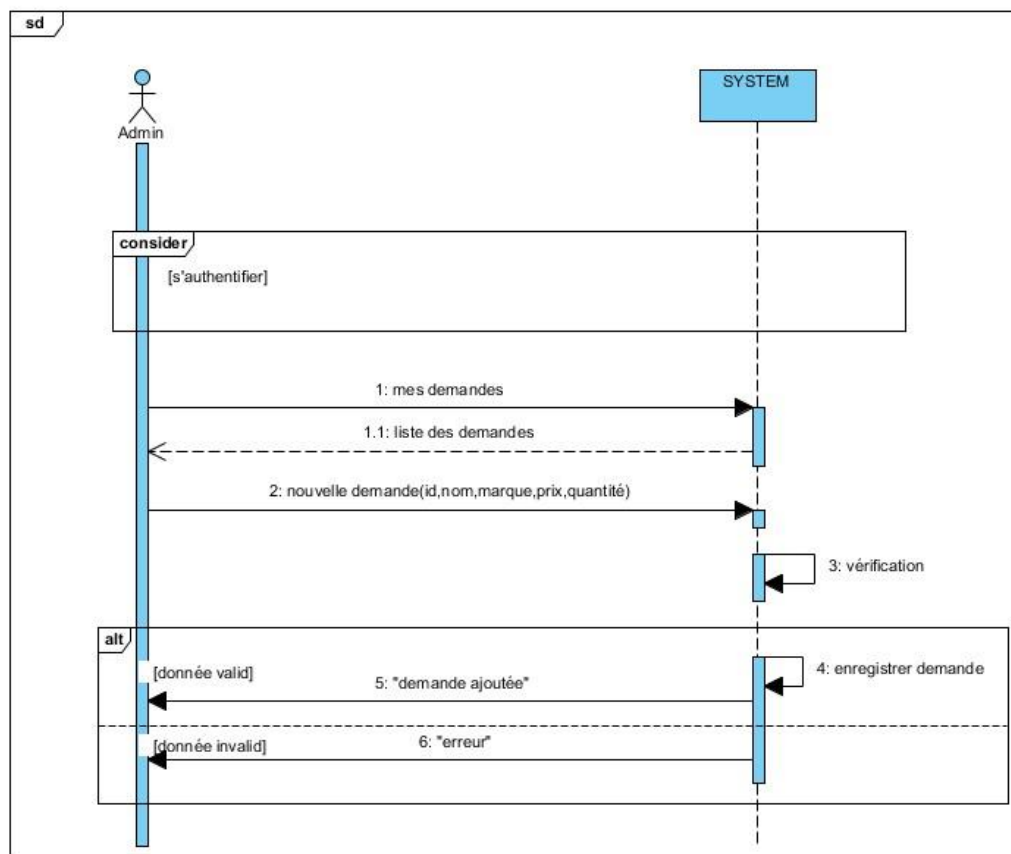


Figure 9 : SD : effectuer demande des matériels

2. Maquette :

Nouvelle Commande

ID Commande :

Nom Matériel :

Marque :

Prix :

Quantité :

Ajouter Ligne

| ID Cmd | Nom | Marque | Prix | Quantité |
|--------|-----|--------|------|----------|
|--------|-----|--------|------|----------|

Figure 10 : CD : effectuer demande des matériels

3. Trouver le concept du domaine :

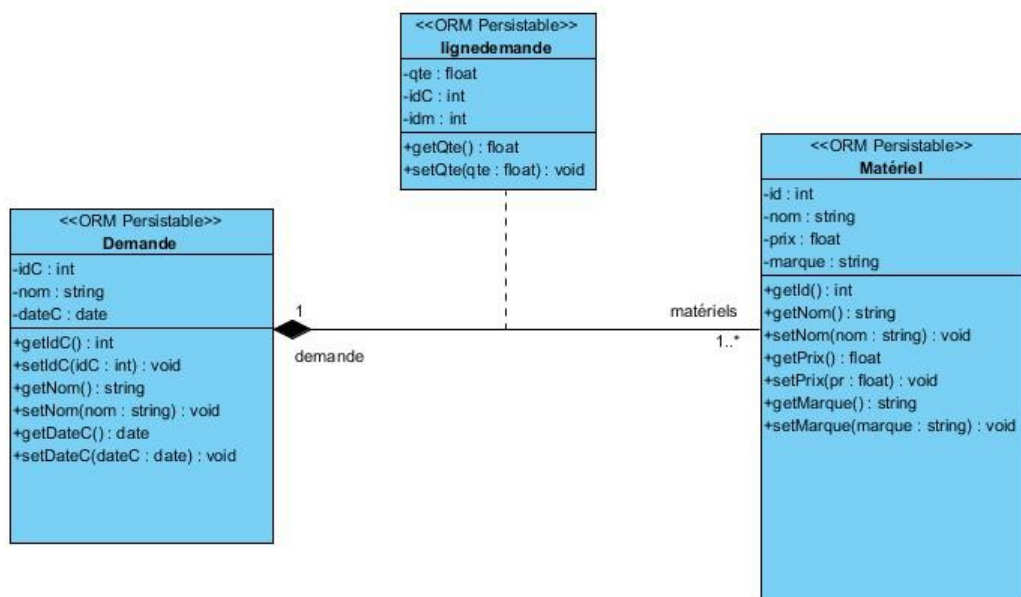


Figure 11 : CD : effectuer demande des matériels

4. Diagramme des classes participantes :

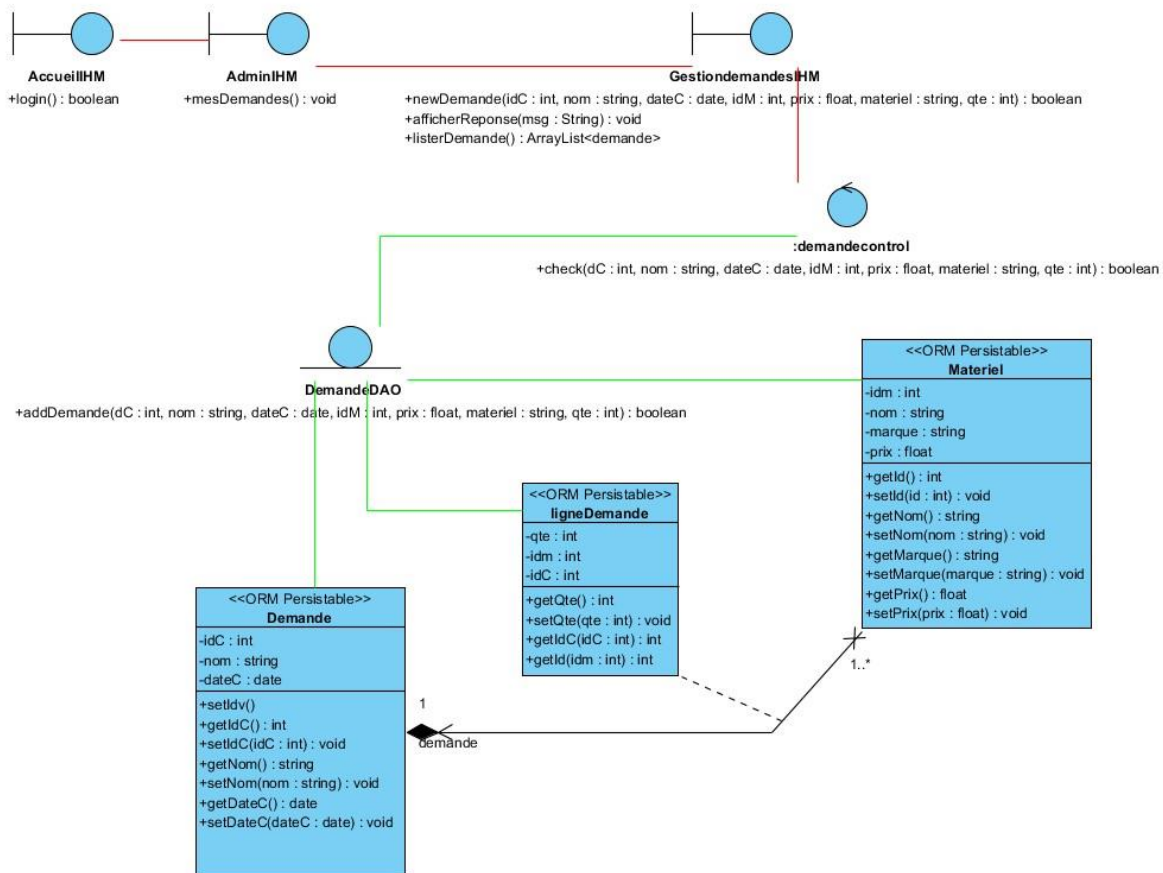


Figure 11 : DCP effectuer une demande d'un matériel

5. Diagramme de navigation : Dans l'UC2, nous avons Trouver 3 IHM, Pour réaliser le cas d'effectuer demande de matériel, l'admin passe de sa page d'accueil **Accueil IHM** vers son espace **Admin IHM**, en cliquant sur 'mes demandes' une page nommée **Gestion Demande IHM** va contenir le formulaire de demande de matériel à remplir.

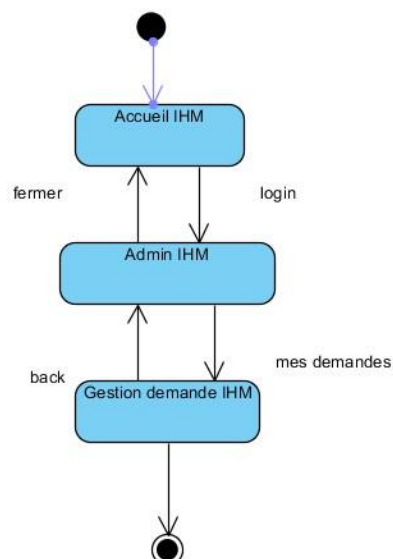


Figure 12 : Nav créer demande :

6. Diagramme d'interaction :

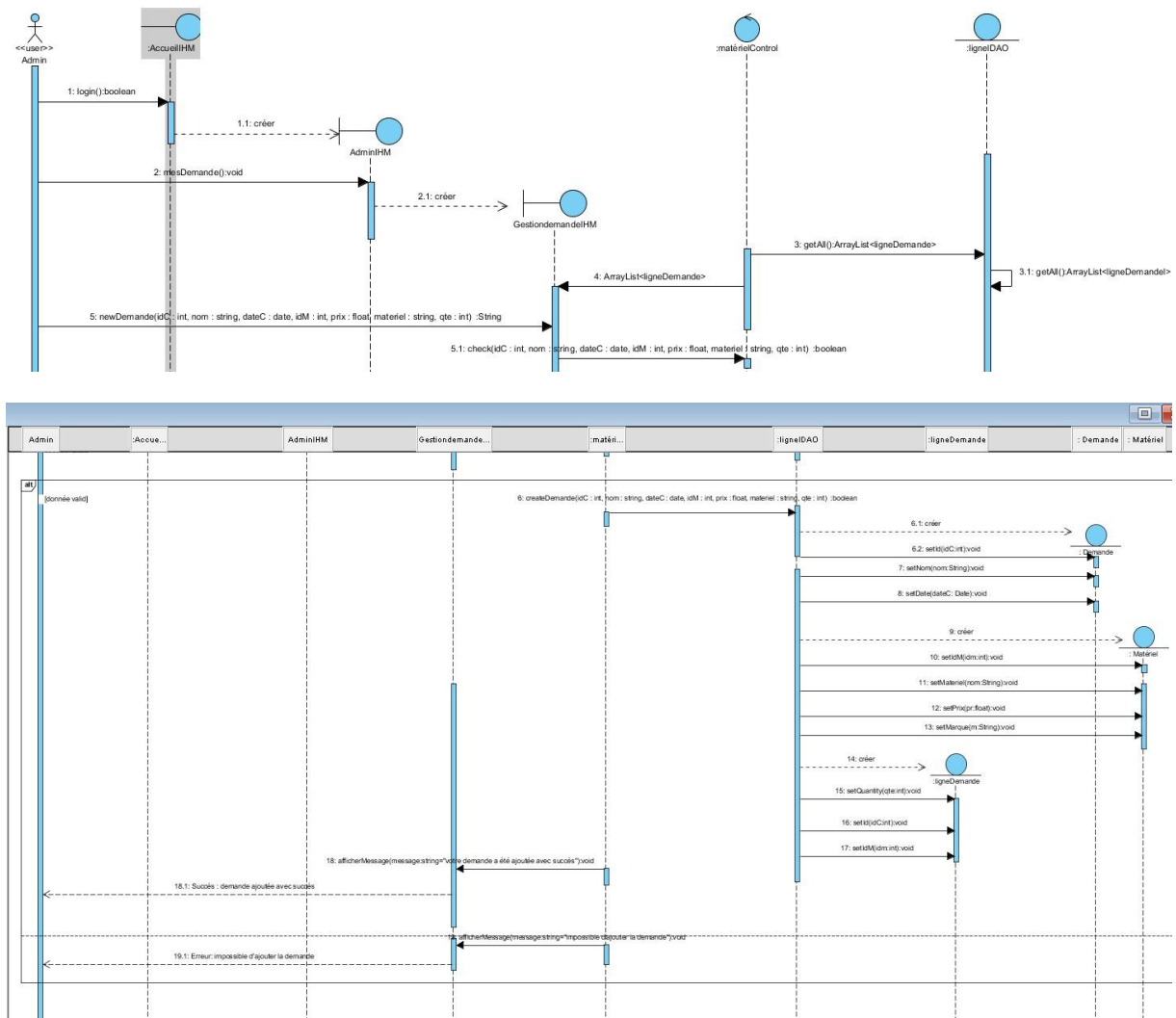


Figure 13 : SD : effectuer demande des matériels

Itération numéro 3 : UC3 : modifier une demande de fourniture

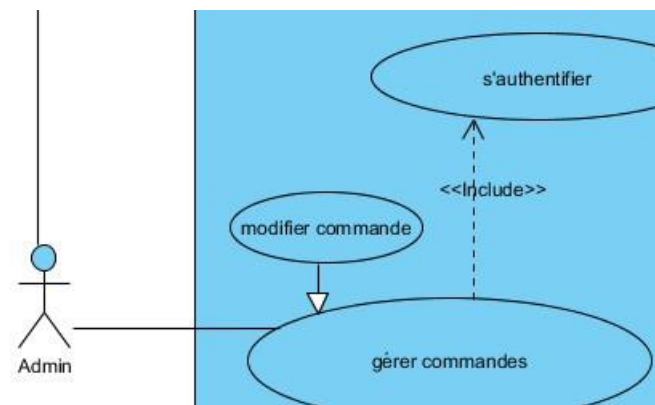


Figure 14 : cas d'utilisation 3

Description Textuelle :

| | |
|-----------------------|---|
| Nom | Mettre à jour une demande de matériel |
| Objectif | Permettre à l'Admin de modifier une demande de matériel |
| ID | 1 |
| Acteurs principaux | Admin |
| Date | 01/03/2025 |
| Responsable | Aya |
| Statut | Proposé |
| Version | V1.0 |
| Précondition | Demande à modifier est sélectionnée |
| Scénario nominal | <ol style="list-style-type: none">1. Admin saisit les nouvelles informations du matériel2. SYSTEM vérifie les données saisies.3. SYSTEM modifie la demande4. Fin |
| Scénario alternatif : | 3-a. SYSTEM affiche un message d'erreur |
| Post condition : | Demande modifiée. |

1. Diagramme de séquence : Flow of events :

| |
|--|
| 1. Admin sélectionne la demande à modifier |
| 2. Admin entre les nouvelles informations |
| 3. SYSTEM vérifier les données |
| 3.1 if données valid |
| 3.1.2 SYSTEM modifier demande |
| 5 else |
| 5.1 SYSTEM affiche message d'erreur |
| end if |

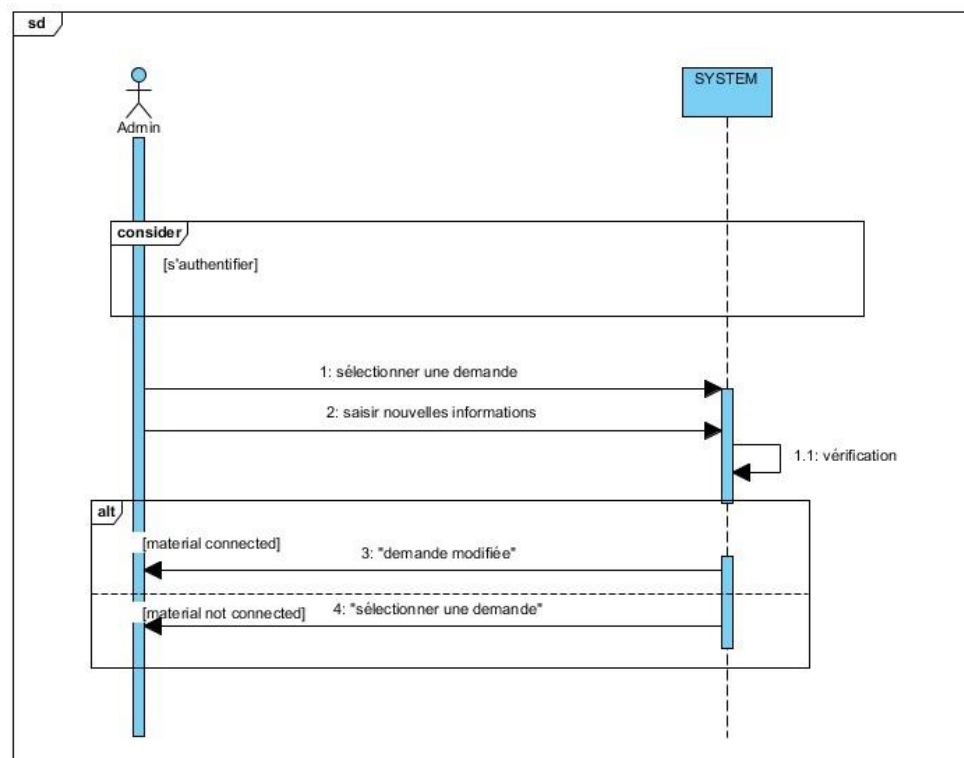


Figure 15 : SD : effectuer demande des matériels

2. Trouver le concept du domaine :

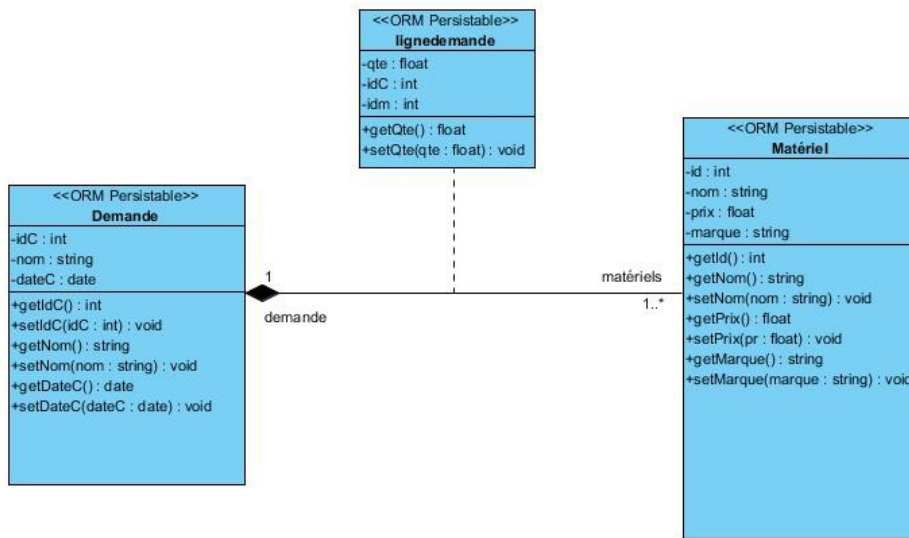


Figure 16 : CD : modifier demande des matériels

4. Diagramme des classes participantes :

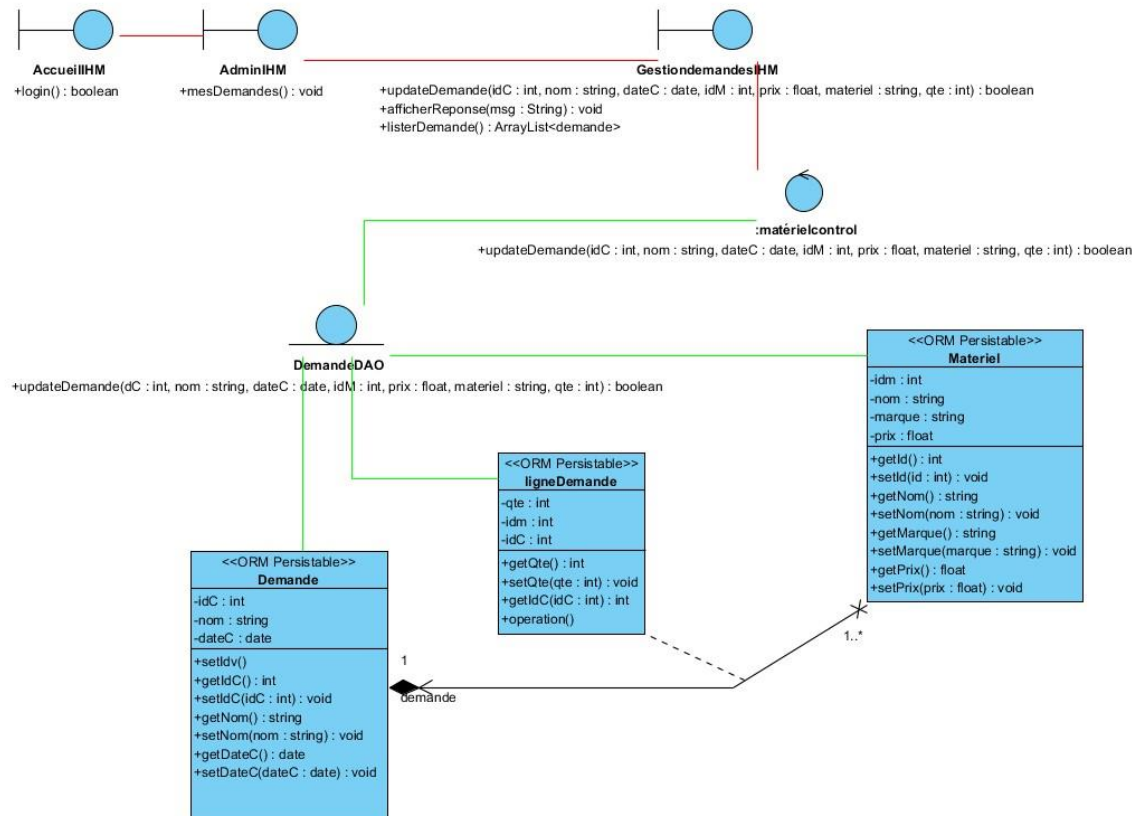


Figure 17 : CP : demande des matériels

Chapitre 4 : Phase de réalisation

Introduction : la phase qui suit une conception est éventuellement l'implémentation. Après les trois chapitres précédents, nous arrivons à rassembler nos informations et réaliser notre système, nous commençons ce chapitre par décrire l'environnement de développement matériel et logiciel que nous avons adopté. Nous présentons ensuite quelques captures d'écrans du logiciel fourni.

1. Environnement de développement :

Dans cette partie, nous présentons les différents outils matériels et logiciels nécessaires pour le développement de notre application.

a) Environnement matériel :

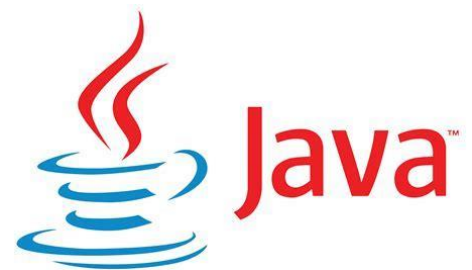
Pour la réalisation de ce mini-projet on a disposé d'un ordinateur DELL caractérisé par :

- Processeur : Intel(R) CORE(TM) i7 2.90 GHz
- Mémoire : 8 Go de RAM
- Disque dur : 256Go SSD
- Système d'exploitation : Windows 10

b) Environnement logiciel :

Dans ce qui suit, nous présentons l'environnement logiciel utilisé pour mener à terme ce sujet.

- ✚ Java est un langage de programmation orienté objet, créé par Sun Microsystems (maintenant propriété d'Oracle Corporation) et publié pour la première fois en 1995. Il est conçu pour être portable et exécutable sur n'importe quelle plateforme grâce à la machine virtuelle Java (JVM), ce qui signifie que le code Java peut fonctionner sur différents types de systèmes sans modification.



- ✚ XAMPP est un package logiciel libre et multiplateforme, qui inclut les applications nécessaires pour créer un environnement de développement web local. Son nom est un acronyme pour Cross-Platform (multiplateforme) .







- ✚ PhpMyAdmin est un logiciel libre écrit en PHP qui a pour mission de s'occuper de l'administration d'un serveur de base de données MySQL. Vous pouvez utiliser phpMyAdmin pour réaliser la plupart des tâches d'administration, ceci incluant la création de base de données, l'exécution de demandes, et l'ajout de comptes utilisateur.

2. Présentation de l'application :

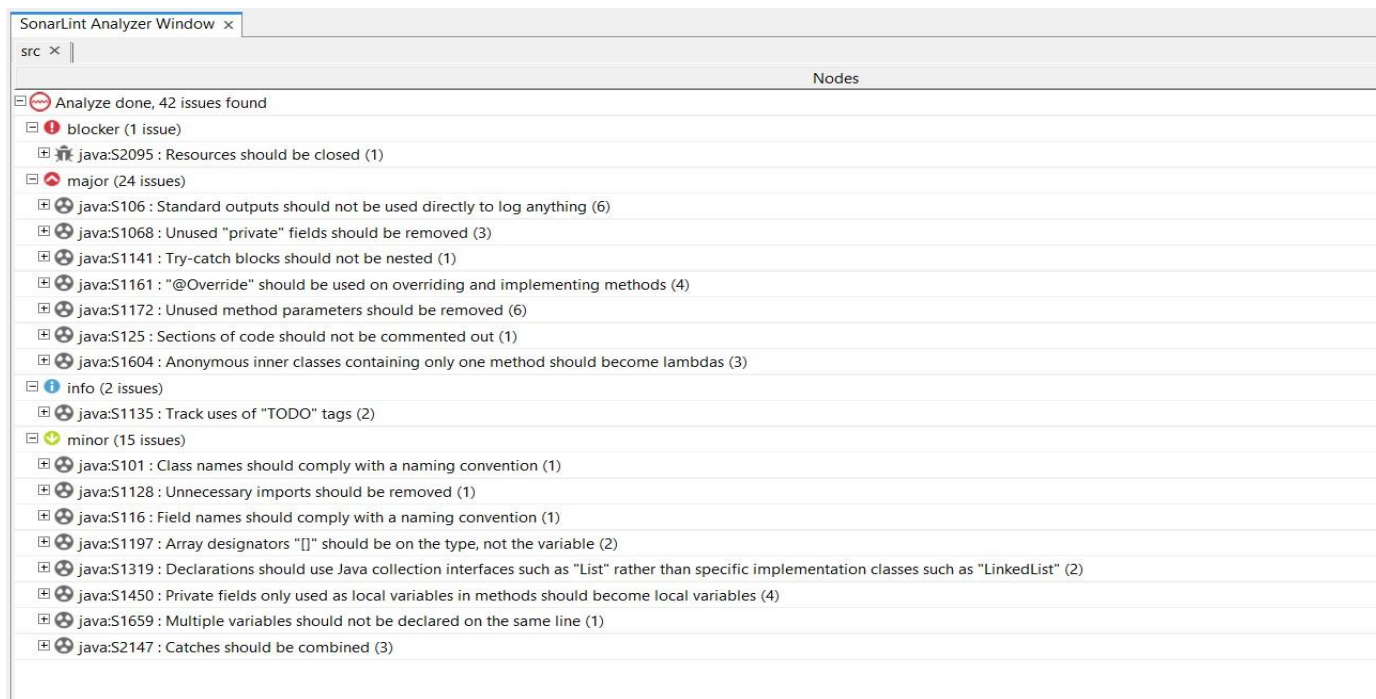
a) Capture d'écran des tâches (Task Manager) :

Dans cette section, nous avons intégré un gestionnaire de tâches qui permet de suivre l'avancement des différentes tâches liées au projet. Vous trouverez ci-dessous une capture d'écran représentant l'interface du *Task Manager*, où sont listées les tâches à accomplir, leur description, ainsi que leur location. Cette fonctionnalité vise à améliorer la gestion du projet et à assurer un suivi en temps réel de son évolution.

| Action Items × | | | |
|---|---|----------------------|---|
| | Description | File | Location |
|  | TODO : tester la connectivité avec la base de données | MaterielDAO.java | ...jects/Miniprojet/src/dao/MaterielDAO.java.2 |
|  | TODO : tester la classe Materiel | Materiel.java | ...sProjects/Miniprojet/src/entity/Materiel.java |
|  | TODO : implémenter la classe MaterielDAOTest | MaterielDAOTest.java | ...s/Miniprojet/test/dao/MaterielDAOTest.java |
|  | TODO : implémenter la classe | MaterielTest.java | ...jects/Miniprojet/test/entity/MaterielTest.java |

b) Utilisation de SONARLINT pour l'Analyse du Code :

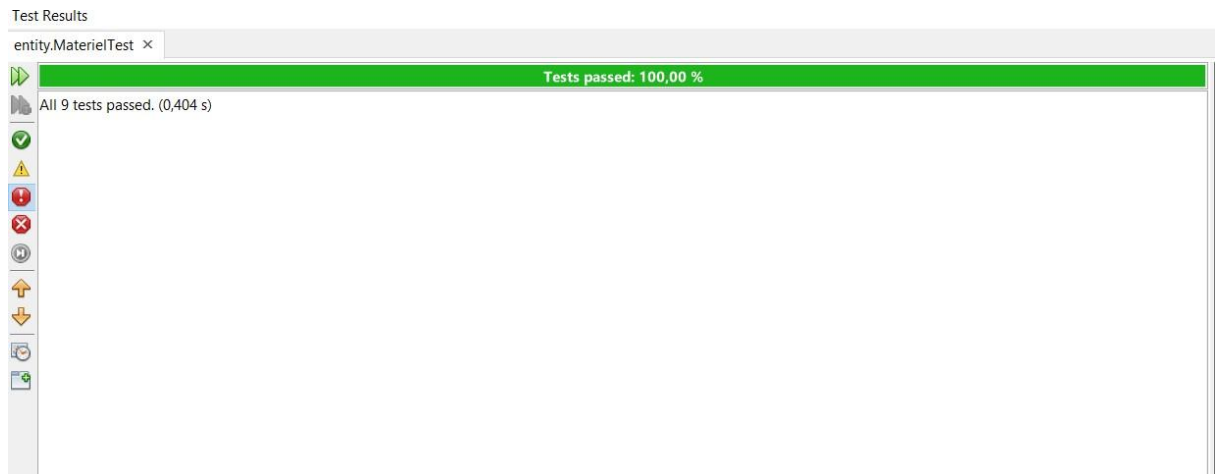
Dans le cadre de l'amélioration de la qualité du code, nous avons utilisé SONARLINT, un plugin intégré dans l'environnement de développement NetBeans, pour analyser le code source du projet. SONARLINT est un outil d'analyse statique qui détecte les problèmes potentiels dans le code, notamment en ce qui concerne les bonnes pratiques, la sécurité et la performance.



c) Tests Unitaires avec JUnit :

En programmation informatique, le test unitaire est un procédé permettant de s'assurer du fonctionnement correct d'une partie déterminée d'un logiciel ou d'une portion d'un programme (appelée « unité » ou « module »).

Voici un petit exemple de test :



d) Captures d'écrans de l'application :

1- Page d'authentification

A screenshot of a web application's authentication page. The browser window title is 'Authentification'. The page has a light gray background. It contains two input fields: 'Nom d'utilisateur :' with the value 'Admin' and 'Mot de passe :' with masked characters '.....'. Below these fields are two buttons: a blue 'Se connecter' button and a light blue 'Annuler' button.

Nom d'utilisateur :

Mot de passe :

Se connecter

Annuler

2- Page de vérification de la connectivité des matériels

rechercher un matériel par son adresse ip :

| idmat | nom | ip | marque |
|--------|---------------|---------------------|---------|
| MAT001 | 192.168.1.10 | Ordinateur Portable | Dell |
| MAT002 | 192.168.1.11 | Imprimante Laser | HP |
| MAT003 | 10.0.0.1 | Switch Réseau | Cisco |
| MAT004 | 172.16.0.5 | Serveur Fichier | Lenovo |
| MAT005 | 192.168.0.254 | Routeur | TP-Link |

entrer une adresse ip :

Mes demandes

3- Page de création des commandes de fourniture

Gestion Simplifiée des Commandes

Nouvelle Commande

ID Commande :

Nom Matériel :

Marque :

Prix :

Quantité :

Ajouter Ligne Modifier Ligne

| ID Cmd | Nom | Marque | Prix | Quantité |
|--------|-----|--------|------|----------|
|--------|-----|--------|------|----------|

Conclusion :

La réalisation de ce projet nous a permis de développer notre rigueur, notre esprit d'analyse, ainsi que notre capacité à modéliser des situations réelles en solutions informatiques. Les contraintes techniques rencontrées ont été surmontées grâce à une recherche continue et une bonne collaboration, ce qui renforce notre autonomie et notre compétence dans le domaine. Il constitue également une base solide pour des projets plus complexes à l'avenir.

Merci à Monsieur Hamid Alhaine.

