

Chapitre III : MÉTHODOLOGIE

III.1 Introduction

Après avoir lu et comprendre les différentes littératures qui s'inscrivent dans le même cadre de notre étude dans le deuxième chapitre, ce chapitre expose la méthodologie suivie pour mener à bien cette recherche. Il vise à présenter de manière claire et structurée les différentes approches et techniques adoptées afin de modéliser et d'assurer la fiabilité des résultats obtenus.

III.2 Méthodes et techniques

III.2.1 Méthodes

Dans cette étude, nous avons choisi d'utiliser la méthode UP (Unified Process) et langage de modélisation UML. Ces démarches vont nous aider à bien organiser nos idées et à représenter clairement le fonctionnement de la plateforme. Grâce à UML, nous pourrions dessiner des schémas qui montrent comment les artisans et les clients vont interagir avec le système. Cela rendra la conception plus simple à comprendre et plus facile à réaliser. Le processus UP nous guide aussi étape par étape jusqu'à la création du projet.

Le processus unifié (UP) se divise en quatre grandes phases : **l'initialisation**, **l'élaboration**, **la construction** et **la transition**. Dans la phase d'initialisation, nous allons définir les besoins principaux et les objectifs du projet. Ensuite, lors de l'élaboration, nous allons analyser les exigences en profondeur et établir l'architecture de base du système. La phase de construction consistera à développer les fonctionnalités principales du système selon l'architecture définie, tandis que la phase de transition nous permettra de tester le système, de corriger les éventuels défauts et de le déployer auprès des utilisateurs. Cette approche nous garantit une progression claire et structurée du projet, en réduisant les risques à chaque étape (PHILIPPINES, 2024).

L'utilisation d'UML va nous permettre de représenter graphiquement tous les aspects importants du système. Nous utiliserons des **diagrammes de cas d'utilisation** pour identifier les interactions entre les utilisateurs (artisans, clients, administrateur) et le système. Des **diagrammes de classes** nous aideront à représenter la structure des données, tandis que des **diagrammes de séquence** montreront comment les différentes opérations se déroulent dans le temps. Grâce à ces outils, nous aurons une vision précise du fonctionnement du système avant même de commencer le développement, ce qui facilitera la communication avec les parties prenantes et la validation du projet à chaque étape (Riccio, 2007).

III.2.2 Techniques et Outils

III.2.2.1 Technique

Dans le cadre de cette étude, nous avons utilisé deux techniques principales pour analyser les besoins réels des artisans et des clients dans la ville de Butembo : **l'observation** et **les entretiens directs**. Ces méthodes nous ont permis de mieux comprendre le fonctionnement actuel du secteur et d'identifier les véritables problèmes rencontrés sur le terrain.

1. Observation des méthodes traditionnelles de recherche d'artisans

Nous avons d'abord effectué une observation directe sur le terrain afin d'identifier comment les habitants de Butembo recherchent habituellement des artisans (menuisiers, mécaniciens, etc.). Cette technique nous a permis de constater que la majorité des citoyens se fient au -à-oreille, aux recommandations de voisins, ou à des pancartes affichées dans les rues **Par exemple**, dans le quartier Vungi, nous avons observé qu'un grand nombre de personnes s'adressaient à un même artisan suite à des recommandations personnelles, ce qui montre une forte dépendance à la connaissance de proximité. Ce mode de fonctionnement limite la visibilité des autres artisans, même compétents.

2. Entretiens avec le Bureau urbain des Cultures et des Arts

Ensuite, nous avons mené des entretiens directs avec des responsables du Bureau urbain des Cultures et des Arts à Butembo. L'objectif était de mieux comprendre l'organisation actuelle du secteur artisanal dans la ville, les mesures mises en place pour soutenir les artisans, ainsi que les difficultés rencontrées. **Par exemple**, lors d'un échange avec un agent de ce bureau, nous avons appris que malgré l'existence d'une base de données interne d'artisans enregistrés, celle-ci n'est pas accessible au public et reste très peu utilisée pour la mise en relation entre artisans et clients. Cette information a confirmé le besoin d'une plateforme plus ouverte et dynamique.

III.2.2.2 Outils

Pour la réalisation de cette plateforme plusieurs outils matériels et logiciel ont été mobiliser :

1.Vscode

Dans cette étude, nous allons utiliser Visual Studio Code comme environnement de développement. Ce choix s'explique par sa simplicité, sa légèreté et ses nombreuses fonctionnalités utiles pour le développement web. Grâce à ses extensions, nous allons pouvoir configurer facilement notre espace de travail pour écrire du code en HTML, CSS, JavaScript et PHP. L'éditeur propose des outils comme l'autocomplétion, la coloration syntaxique, le

terminal intégré et le débogueur, qui vont nous permettre de coder plus efficacement et de tester directement certaines fonctionnalités de notre plateforme. Par exemple, nous allons utiliser VS Code pour concevoir les interfaces de la plateforme et pour vérifier la communication avec la base de données locale pendant le développement(Viot, 2023).

2. Xampp

Dans cette étude, nous allons utiliser XAMPP comme serveur local pour exécuter notre application web. XAMPP regroupe les outils essentiels au développement web, notamment Apache pour le serveur web, MySQL pour la gestion de base de données, ainsi que PHP et phpMyAdmin. Grâce à cet environnement, nous allons pouvoir tester notre plateforme en local, sans avoir besoin d'un hébergement en ligne pendant la phase de développement. Par exemple, nous allons créer une base de données avec phpMyAdmin pour stocker les informations des utilisateurs et des artisans, et nous allons utiliser Apache pour simuler le comportement du site tel qu'il sera en ligne. XAMPP facilitera donc toutes les étapes de test, de correction et d'amélioration de notre application(N'KPONON, 2023).

3. PHP

Dans cette étude, nous allons utiliser le langage PHP pour développer la partie dynamique de notre plateforme. PHP va nous permettre de gérer les échanges entre l'utilisateur et le serveur, comme l'enregistrement des comptes, la connexion, ou la gestion des données des artisans et des clients. Grâce à PHP, nous allons aussi pouvoir interagir avec la base de données MySQL afin d'enregistrer, afficher, modifier ou supprimer des informations selon les actions de l'utilisateur. Par exemple, lorsque qu'un client remplira un formulaire pour contacter un artisan, le script PHP traitera les données et les enregistrera dans la base de données. Ce langage jouera donc un rôle essentiel dans le bon fonctionnement du système(N'KPONON, 2023).

III.3 MODELISATION DE LA SOLUTION

III.3.1 EXPRESSION DES BESOINS

Dans la méthode UP, l'expression des besoins est une étape essentielle qui permet de comprendre ce que les utilisateurs attendent réellement du système à développer. Elle consiste à recueillir et formuler de manière claire les différentes attentes fonctionnelles (comme la création d'un compte, la recherche d'un artisan, l'envoi de messages, etc.) et non fonctionnelles (comme la rapidité, la sécurité, ou la simplicité d'utilisation). Cette phase sert de base à tout le projet, car elle permet d'éviter les erreurs d'interprétation et d'orienter correctement la conception(Shankar et al., 2020). Dans notre étude, nous allons nous appuyer sur des observations du terrain et des entretiens réalisés à Butembo pour identifier

précisément les besoins des artisans et des clients, afin de proposer une plateforme qui répond à leurs réalités.

III.3.2 **Cahier de charger**

Le **cahier des charges** est un document très important dans un projet. Il explique clairement ce que doit faire le projet, les besoins des utilisateurs, et les règles à suivre. Ce document sert de guide pendant toute la durée du projet. Il décrit ce que le système doit pouvoir faire (comme les fonctionnalités) et ce qu'il doit respecter (comme la sécurité, la rapidité ou la facilité d'utilisation). Le cahier des charges aide à éviter les erreurs, à mieux comprendre les attentes des utilisateurs et à s'assurer que tout est fait correctement et dans les temps(Coulon, 2019).

Identification du projet

Notre travail porte sur le développement d'une plateforme de mise en relation entre artisan et client d'une manière simple et sécurisé. Il reprend les besoins ci-dessous :

1. Besoin fonctionnel

Une fois réaliser, ce système sera à mesure de :

- Visiter site
- Inscrire les artisans et client : le système va permettre de créer un compte personnel en remplissant un formulaire d'inscription.
- Publication d'une demande : le système va permettre à un client de publier une demande de service destinée aux artisans
- Publier un Avis d'un client : le système va permettre à un client d'évaluer et de laisser un avis sur un artisan après la réalisation d'un service.
- Messagerie entre artisan et client : le système doit permettre aux clients et aux artisans de communiquer entre eux via une messagerie interne, afin de clarifier les demandes, fixer des rendez-vous ou négocier les conditions.

2. Besoin opérationnel ou non fonctionnel

- **Sécurité** :Pour garantir la **confidentialité des données** et protéger l'accès aux fonctionnalités de la plateforme, un **système d'authentification sécurisé** est mis en place. Seuls les utilisateurs autorisés (artisans, clients et administrateurs) peuvent accéder à leur espace personnel.
- **Performance** : assurer une performance élevée même avec un grand nombre d'utilisateur en utilisant une BD relationnelle

- **Compatibilité** : compatible avec tout le système d'exploitation
- **Ergonomie** : notre plateforme est facile à utiliser

3. Choix technique

- **Méthode de modélisation** : UP
- **Langage de modélisation** : UML
- **SGBD** : MySQL
- **Architecture** : Client-serveur 3/3
- **Langage de programmation** : PHP
- **Pour les interfaces** : Bootstrap, HTML, CSS

III.3.3 Identification des acteurs et leurs rôles

un acteur désigne le rôle assumé par une entité extérieure qu'il s'agisse d'un utilisateur, d'un appareil, d'un équipement ou d'un autre système qui établit une interaction directe avec le système analysé (Hadzovic et al., 2021).

Les acteurs de notre plateforme sont :

- Artisans
- Client
- Internaute
- Administrateur

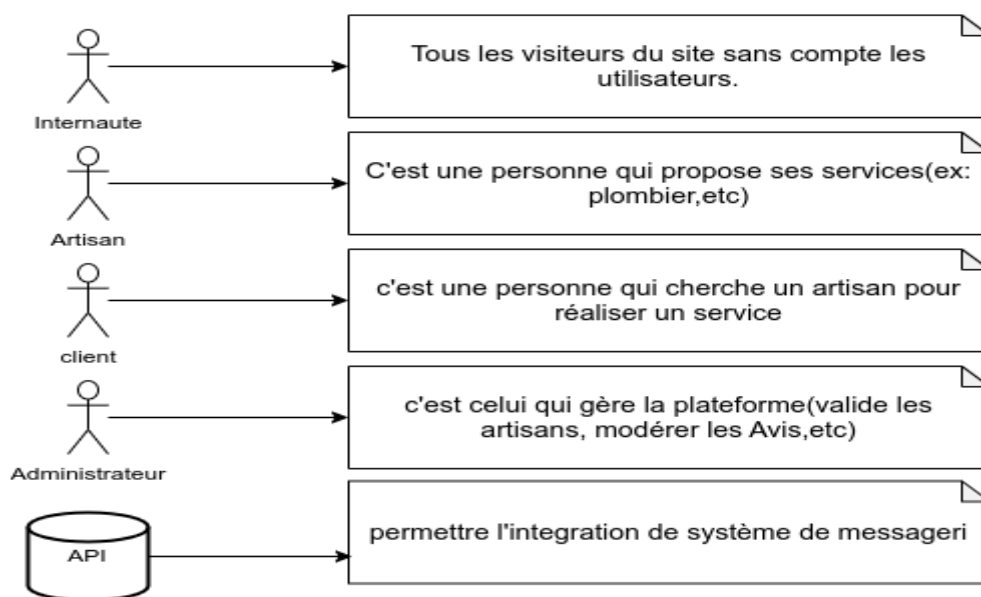


Figure 1

Identification des acteur et leur rôles

III.3.1 Modélisation des contextes

La modélisation des contextes consiste à représenter de façon structurée l'environnement dans lequel un système évolue. Elle permet d'identifier les éléments externes et internes qui influencent ou interagissent avec le système. Cette démarche facilite la compréhension des besoins réels et des contraintes du projet. Elle sert aussi à définir les limites du système et à préciser les échanges avec les entités externes. Grâce à cette modélisation, les analystes peuvent anticiper les comportements attendus. Elle est souvent représentée sous forme de diagrammes comme les diagrammes de contexte ou d'acteurs(Mukoma et al., 2024).

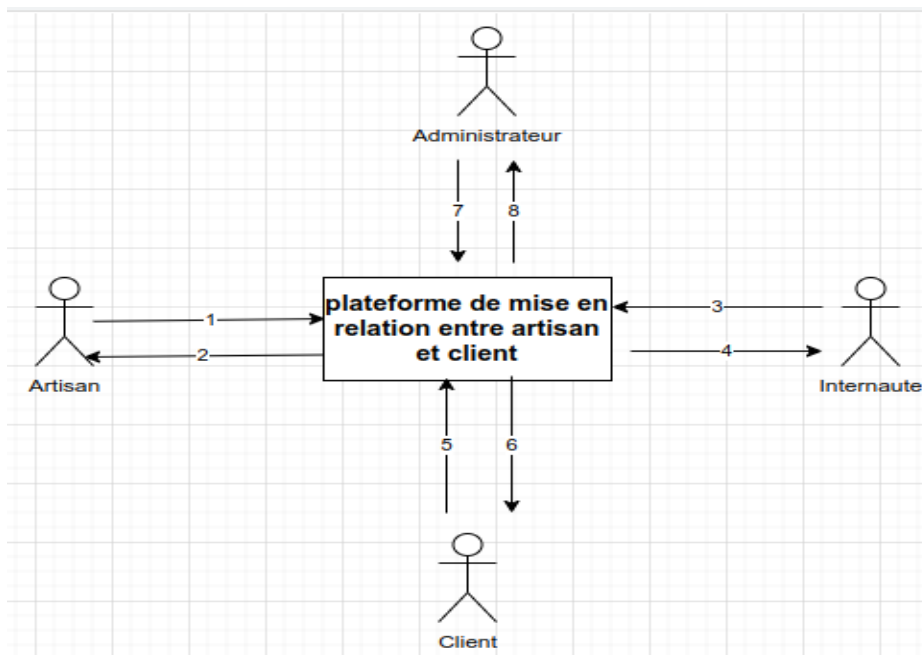


Figure 2 Modélisation des contextes

Légende :

1. L'artisan envoie ses informations d'inscription vers la plateforme : lors de l'inscription (nom, postnom, etc)
2. La plateforme renvoie à l'artisan une confirmation d'inscription ou de validation
3. l'internaute émet une requête de consultation d'un profil
4. la plateforme lui retourne les profils des artisans
5. Le client envoie une demande de service via la plateforme : après s'être connecté, il effectue cette opération.
6. La plateforme renvoie au client un message de l'artisan qui a accepté ce service la première fois
7. L'administrateur envoie des données de gestion vers la plateforme

8. La plateforme retourne des rapport des diffèrent profils

III.4 **MODELISATION DYNAMIQUE DU SYSTEME**

Dans cette partie, il s'agira de la clarification du diagramme des cas d'utilisation, de la description textuelle de différents cas, des diagrammes de séquences, d'activité, de package, d'état transition, et la matrice de validation.

III.4.1 **Identification des cas d'utilisation**

L'identification des cas d'utilisation, consiste à définir les interactions entre les utilisateurs et le système pour atteindre des objectifs spécifiques. Chaque cas d'utilisation formalise une exigence fonctionnelle du système, en décrivant les actions à entreprendre pour satisfaire un besoin. Ce processus permet de structurer les attentes des utilisateurs et les contraintes du système, facilitant ainsi la communication entre les parties prenantes. Il sert de base à la conception et au développement du système(Mukoma et al., 2024).

Cas d'utilisation identifiés pour notre système :

1. **Visiter le site**
2. **S'inscrire**
3. **S'authentifier**
4. **Consulter publication**
5. **Valider inscription**
6. **Modérer envi**
7. **Gérer compte**
8. **Envoyer message**
9. **Publier une demande**

III.4.2 **Diagramme de cas d'utilisation**

Dans l'article « *Use Case Diagrams in Support of Use Case Modeling* » les **cas d'utilisation** sont définis comme des descriptions structurées des interactions possibles entre des acteurs (utilisateurs ou systèmes externes) et le système étudié, dans le but d'atteindre un objectif précis (Siau & Cao, 2019)

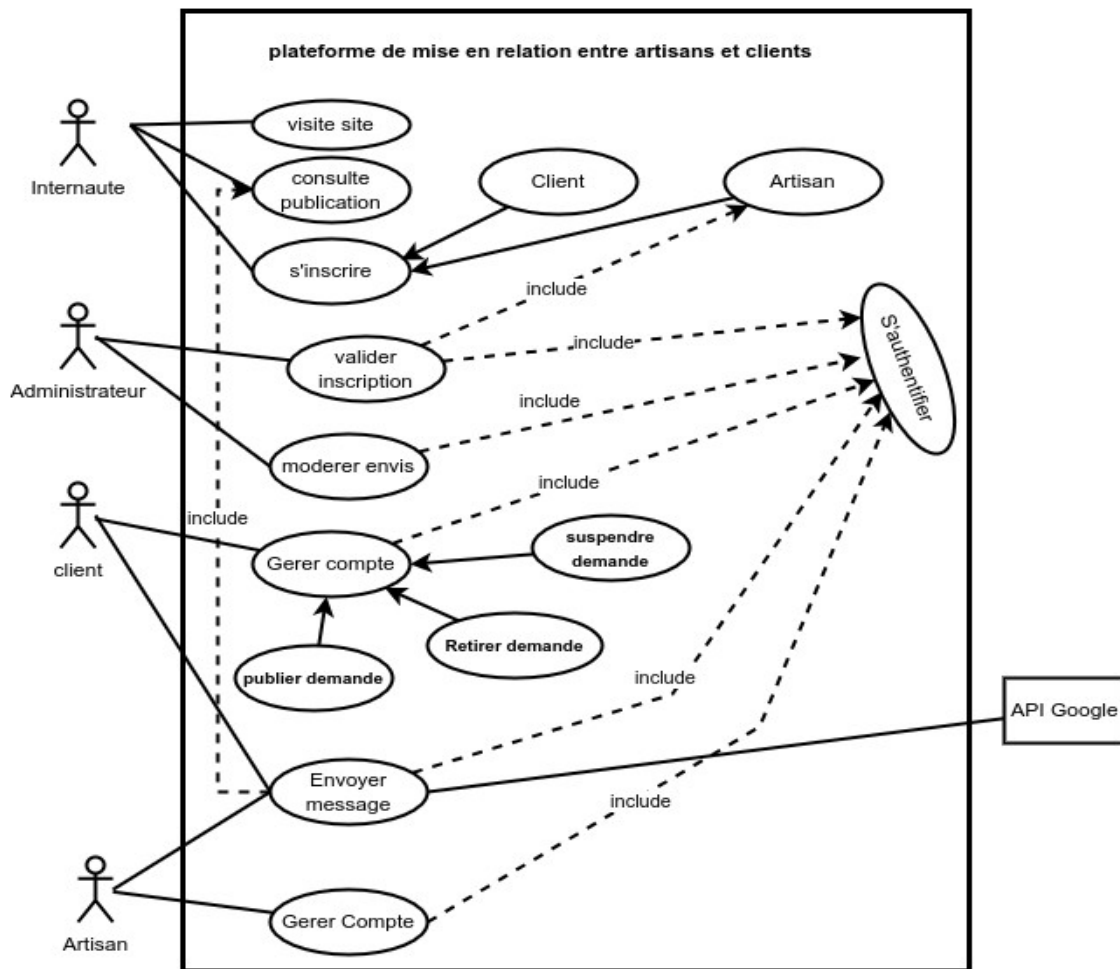


Figure 3 Diagramme de cas d'utilisation

III.4.2.1 Structuration des CAU(Description)

Dans ce qui suit, nous décrirons de façon détaillée certains cas d'utilisation identifiés précédemment en recensant de façon textuelle toutes les interactions entre

a) Description textuelle du cas « s'authentifier »

Identification du CAU

Nom : Authentification

But : donner aux utilisateur la possibilité de se connecter au système ;

Acteurs : Artisan, Admirateur,client ;

Date de création : 10/05/2025 ;

Version : 0.1

Responsable : morisho nyembo delphin

Séquencement

Précondition : acceder à la plateforme dans le navigateur web

Enchainement Nominal

1. Afficher la page d'accueil avec le menu d'authentification ;
2. clique sur le menu d'authentification
3. Afficher la page d'authentification ;
4. Remplir le formulaire qui se présente ;
5. Après clic de validation ;
6. Vérification des données saisies

Enchainement Alternatif.

A.4. Affichage du message d'erreur

le système demande de saisir les information à nouveau

Post-Condition

Accès à la page d'accueil de l'utilisateur

b) Description textuelle du cas s'inscrire**Identification du CAU**

Nom : s'inscrire

But : donner aux utilisateur la possibilité d'avoir un compte au système ;

Acteurs : internaute ;

Date de création : 10/05/2025 ;

Version : 0.1

Responsable : morisho nyembo delphin

Séquencement

Précondition : accéder à la plateforme dans le navigateur web

Enchainement Nominal

1. Affiche la page d'accueil + menu s'inscrire
2. L'internaute clique sur le menu s'inscrire
3. Le système affiche le formulaire d'inscription
4. L'internaute complete c'est formulaire et valide
5. Vérification de donnée saisi

Enchainement Alternatif.

A.4. Affichage du message d'erreur que le nom d'utilisateur existe déjà

Le système demande de ressayer à nouveaux

Post-Condition

Création du compte

c) Description textuelle du cas « valider inscrit »

Identification du CAU

Nom : valider inscription

But : c'est cas permet à l'administrateur de vérifier les informations dès l'artisan si il est bien artisan (certificat, ...) ;

Acteurs : Admirateur ;

Date de création : 10/05/2025 ;

Version : 0.1

Responsable : morisho nyembo delphin

Séquencement

Précondition : Etre connecte au système.

Enchaînement Nominal

1. L'administrateur clique sur le menu gérer utilisateur puis sur artisanal
2. Le système affiche la liste des artisans
3. L'administrateur clique sur le menu profil non valider
4. Le système affiche tout le profil non valider
5. L'administrateur sélectionne un profil
6. Le système affiche les information consternant cette artisan (certificat, etc.)
7. Vérifie si les informations sont correctes
8. Validation du profil

Enchaînement Alternatif.

A.4. Profil non conforme

Envoyer message à l'artisan lui informent la non validation

Post-condition

Envoyer message à l'artisan lui informent la validation

d) Description textuelle du cas « modérer envi »

Identification du CAU

Nom : modérer envi

But : assurer un équilibre et en limitant les excès ;

Acteurs : Admirateur ;

Date de création : 10/05/2025 ;

Version : 0.1

Responsable : morisho nyembo delphin

Séquencement

Précondition : Être connecte au système.

Enchaînement Nominal

1. L'administrateur clique sur le menu modérer l'envi
2. Le système affiche la liste de profil+leur envi
3. Administrateur sélectionne le profil
4. Le système affiche le profil+ l'envi
5. Vérifie si l'envie et correcte
6. Publie l'envi

Enchaînement Alternatif.

A.4. Rejeter l'envie

Post-condition

Valider l'envi

e) Description textuelle du cas « publier demande »**Identification du CAU**

Nom : publier demande

But : donner aux client la possibilité de publier une demande ;

Acteurs : client ;

Date de création : 10/05/2025 ;

Version : 0.1

Responsable : morisho nyembo delphin

Séquencement

Précondition : Précondition : Être connecte au système.

Enchainement Nominal

1. Clique sur le menu publier une demande
2. Le système affiche le formulaire de publication de la demande
3. Compléter le formulaire
4. Le système retourne le message de la confirmation de la publication

Post-Condition

Publication de la demande

III.4.3 Diagramme de séquences

L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés (kabunga & kambale, 2024).

A) Diagramme de séquence du cas « s'authentifier »

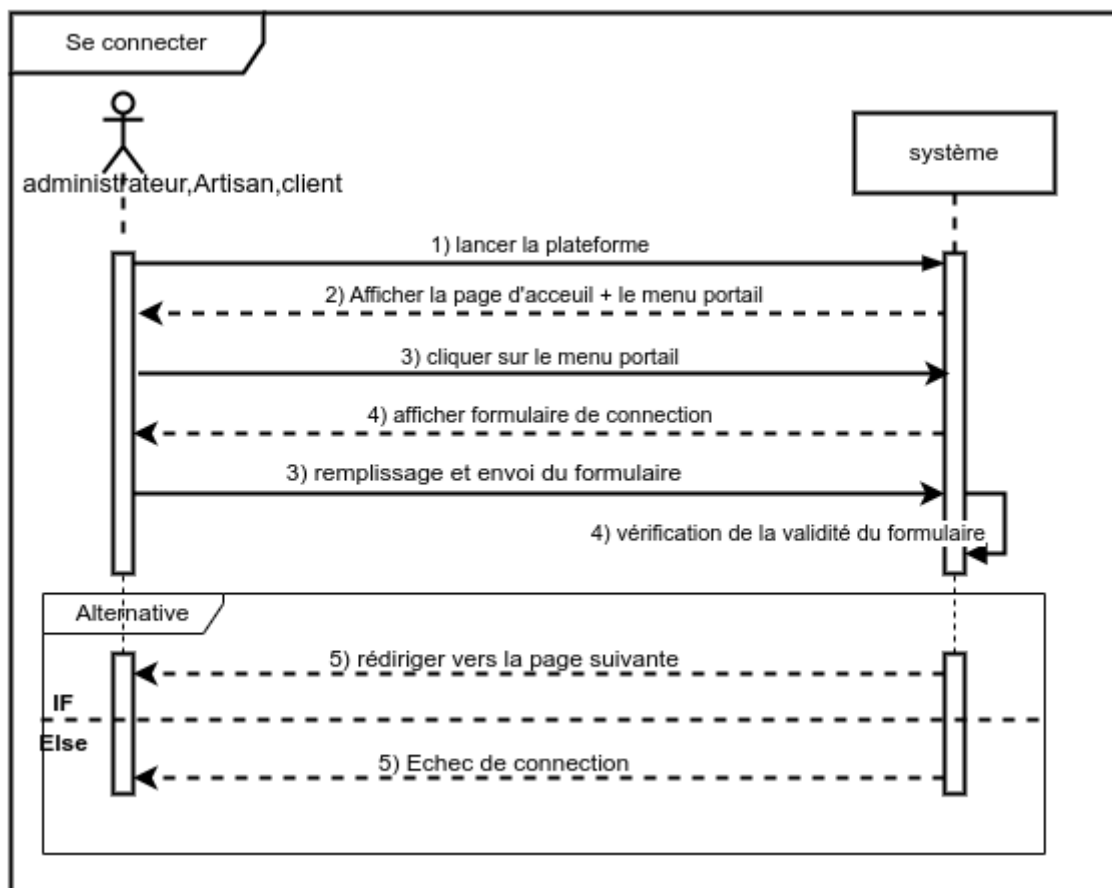


Figure 4 Diagramme de séquences

B) Diagramme de séquence du cas « s'inscrire »

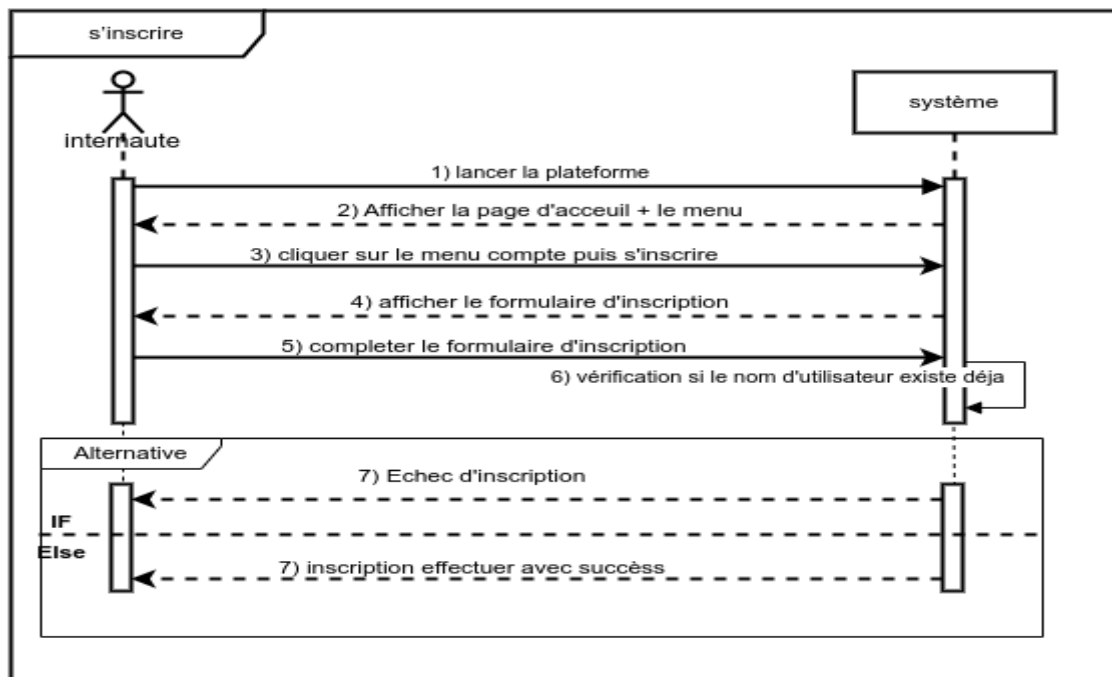


Figure 5 Diagramme de séquence du cas « s'inscrire »

C) Diagramme de séquence du cas « valider inscription »

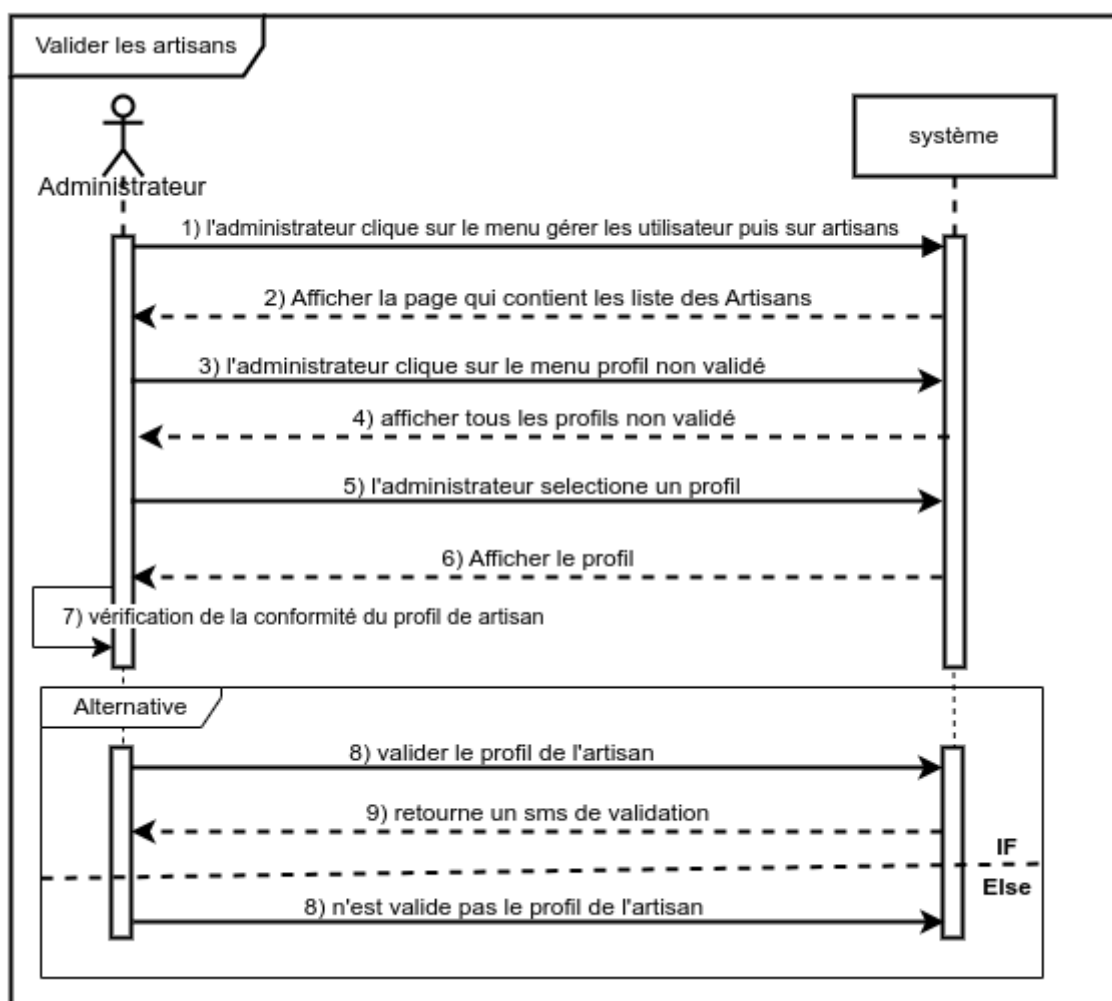


Figure 6 Diagramme de séquence du cas « valider inscription »

D) Diagramme de séquence du cas « publier demande »

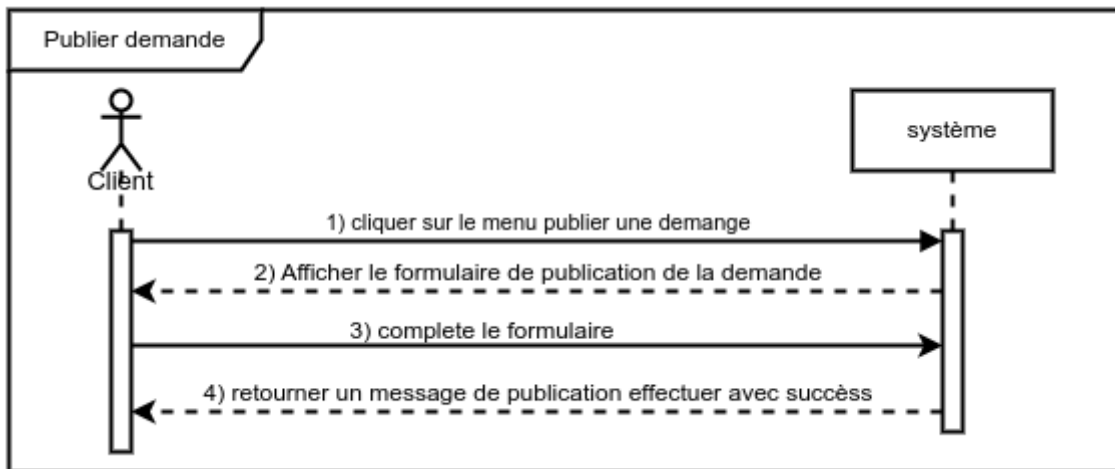


Figure 7 Diagramme de séquence du cas « publier demande »

E) Diagramme de séquence du cas « modérer un envi »

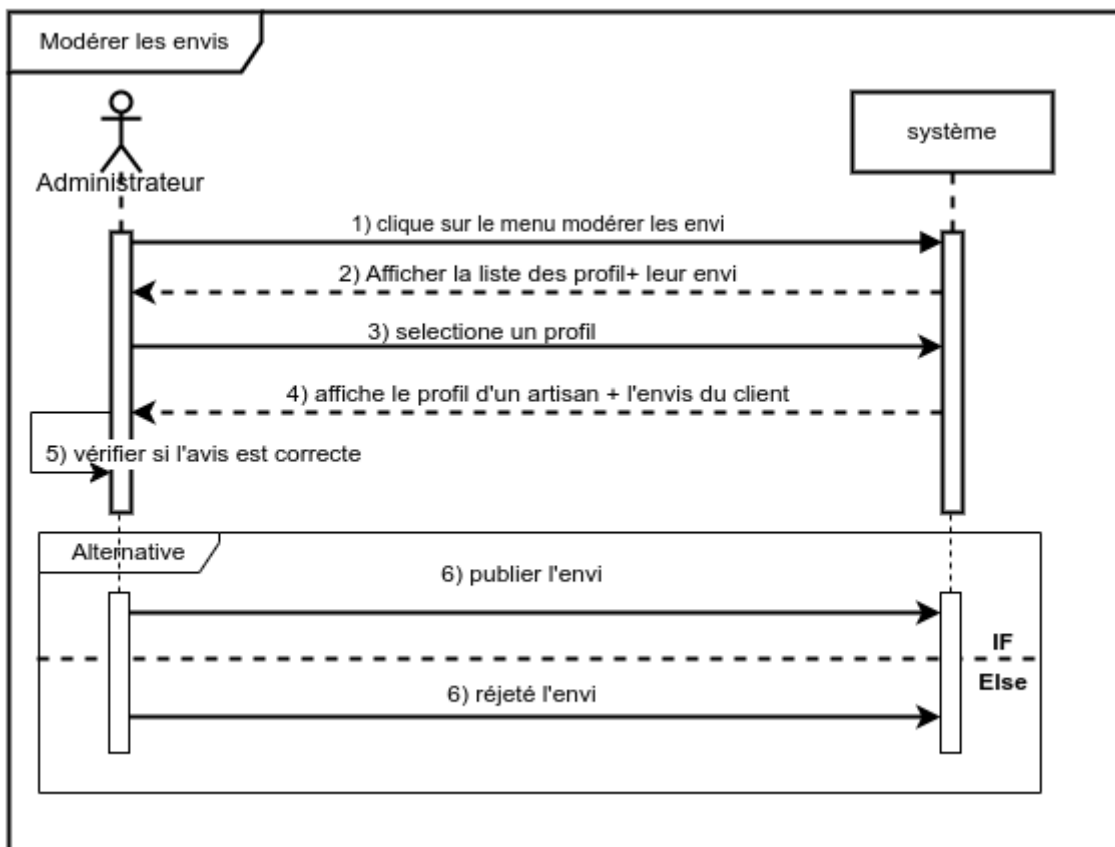


Figure 8 Diagramme de séquence du cas « modérer un envi »

F) Diagramme de séquence du cas « publier envi »

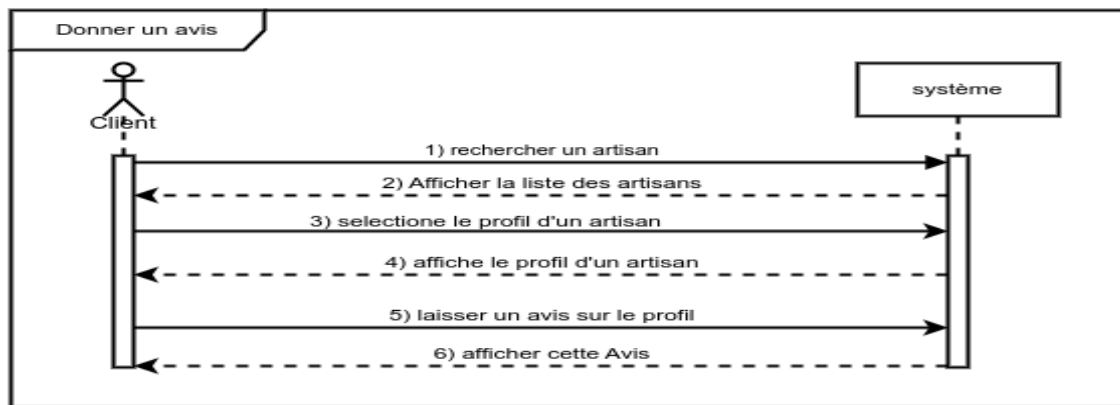


Figure 9 Diagramme de séquence du cas « publier envi »

III.4.3 Diagramme d'activité

Sert à développer et analyser le scénarios d'utilisation du système d'une façon graphique (kabunga & kambale, 2024).

A. Diagramme « d'activité du cas s'authentifier »

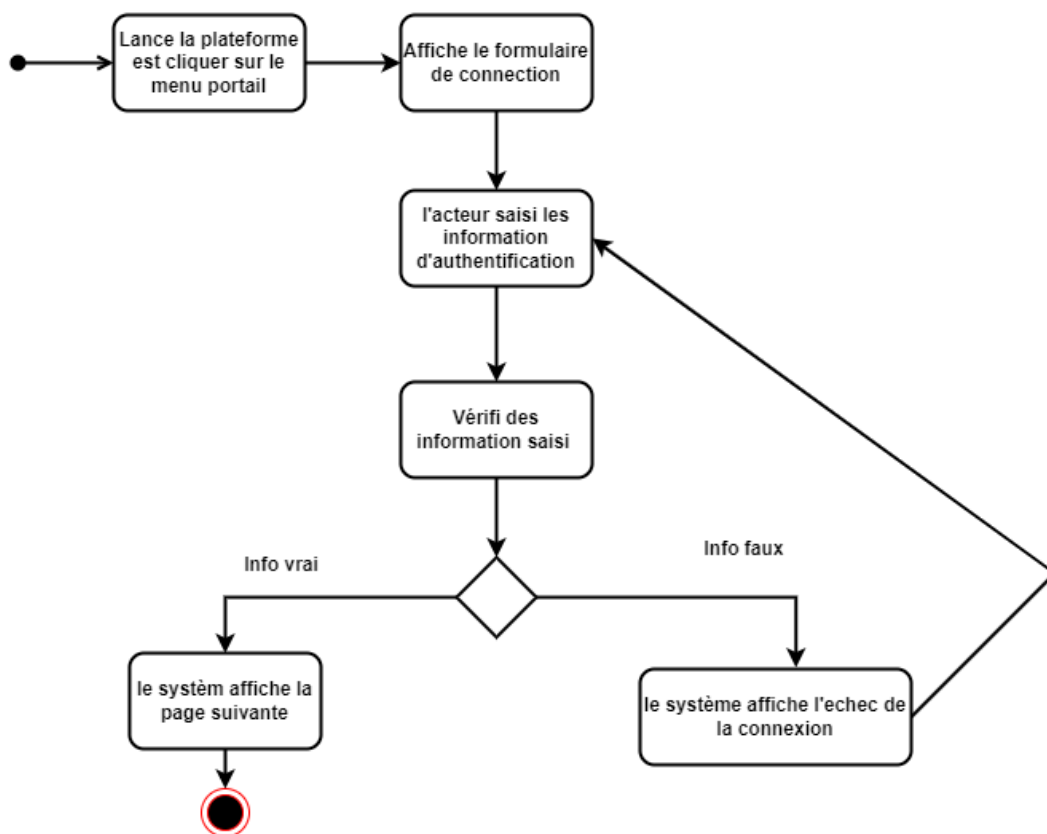


Figure 10. Diagramme « d'activité du cas s'authentifier »

B. Diagramme « d'activité du cas s'inscrire »

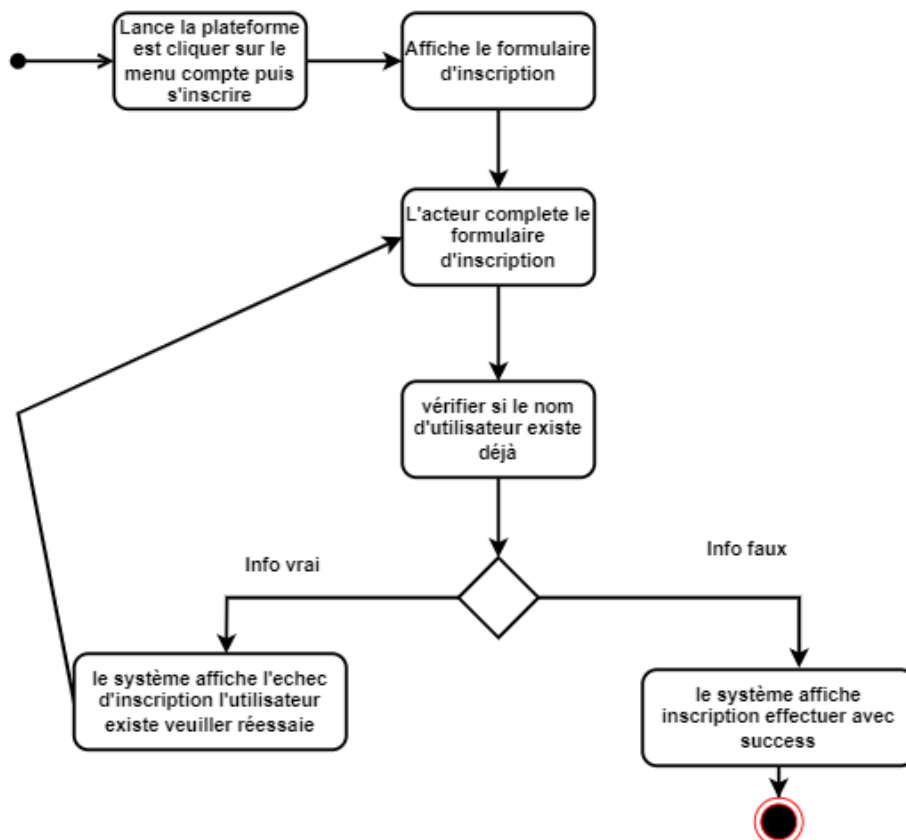


Figure 11. Diagramme « d'activité du cas s'inscrire »

C. Diagramme d'activité « valider les inscription »

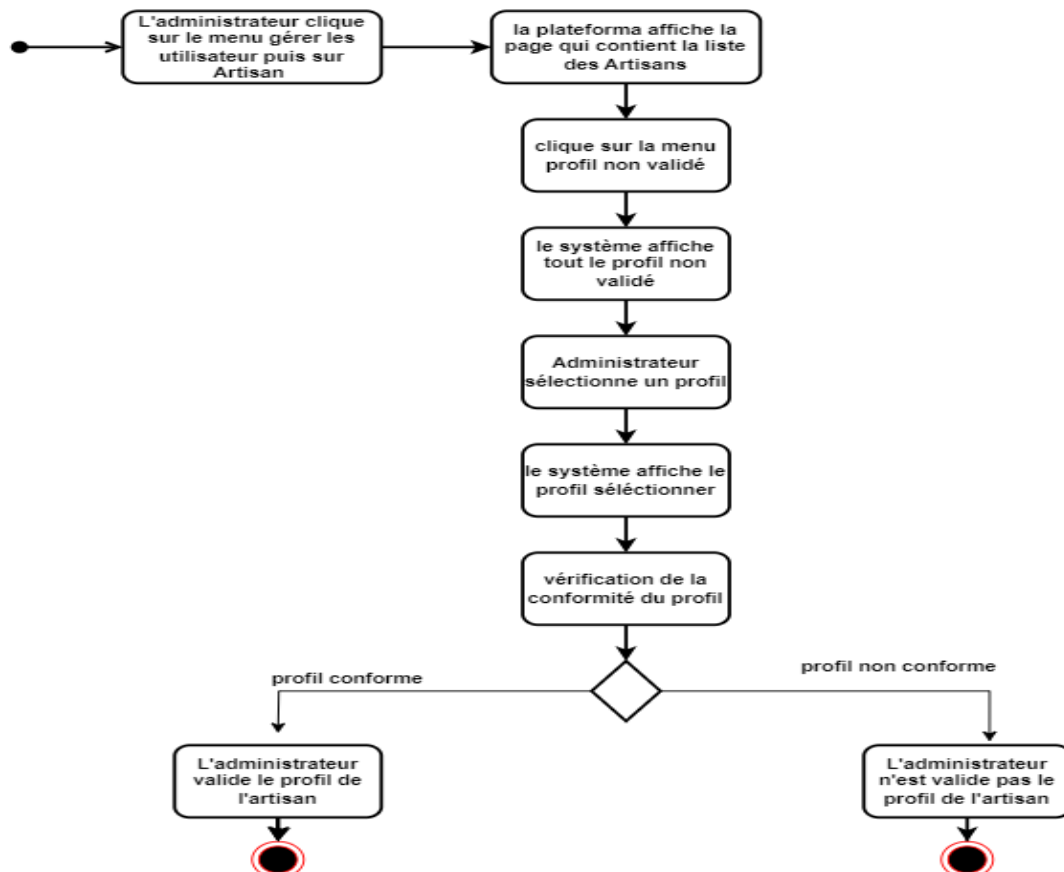


Figure 12 Diagramme d'activité « valider les inscription »

D. Diagramme d'activité « publier demande »

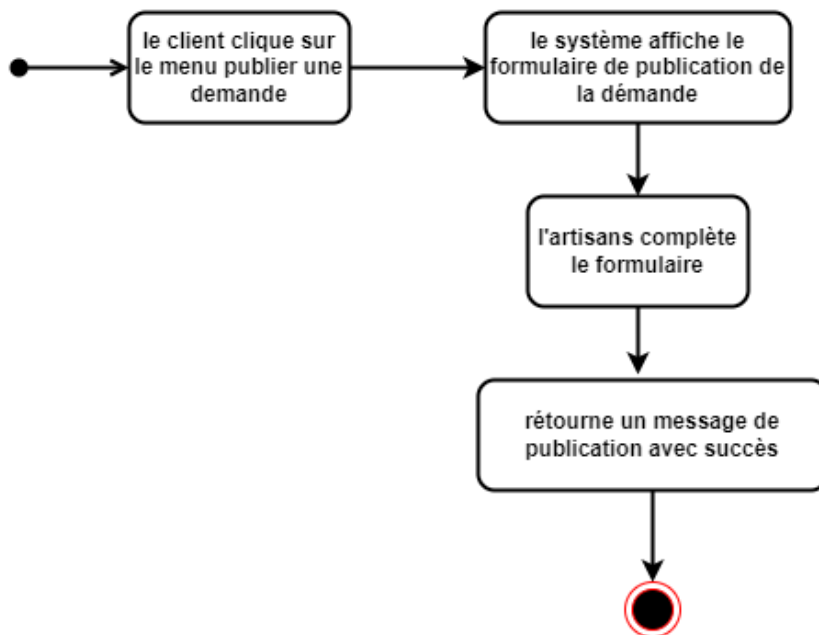


Figure 13. Diagramme d'activité « publier demande »

E. Diagramme d'activité « modérer un envi »

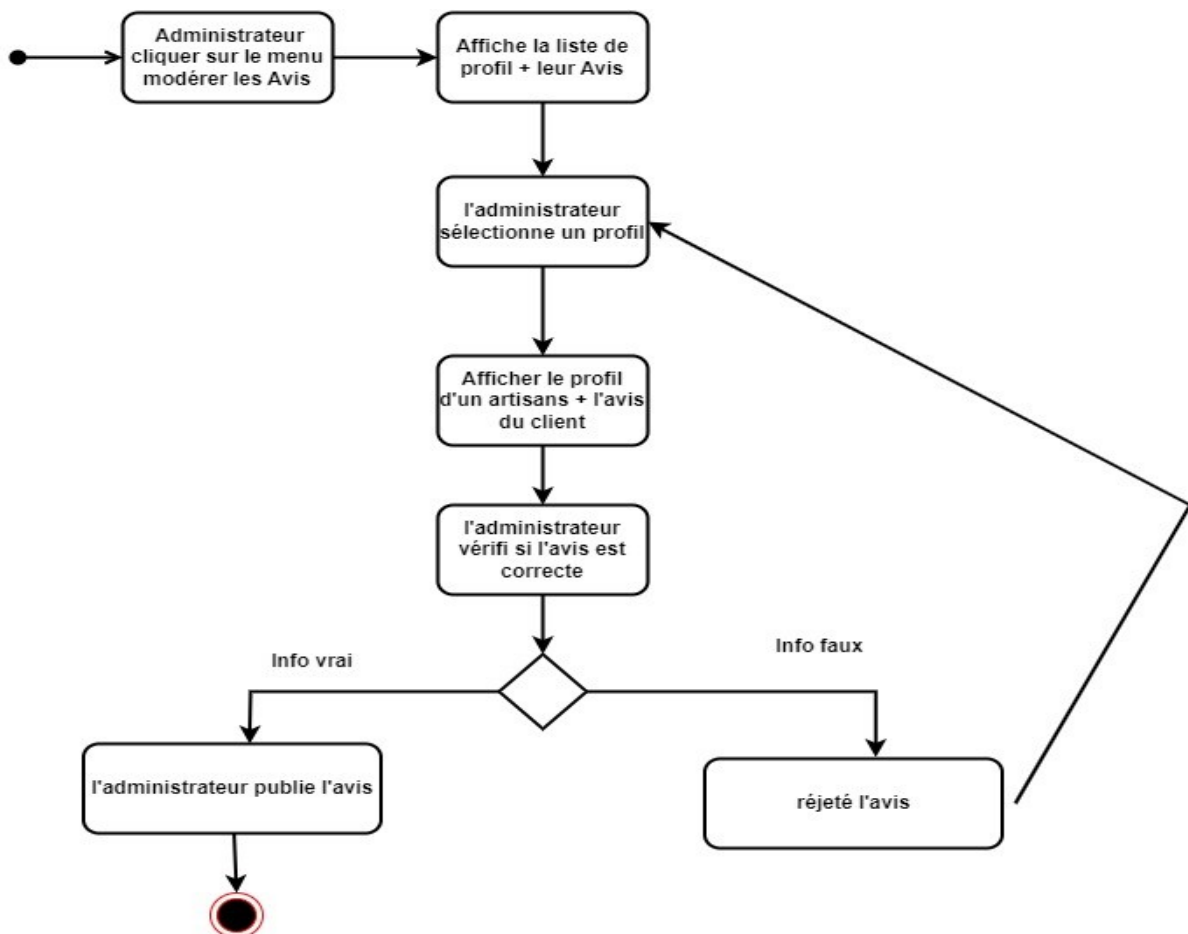


Figure 14. Diagramme d'activité « modérer un envi »

F. Diagramme d'activité « Publier envi »

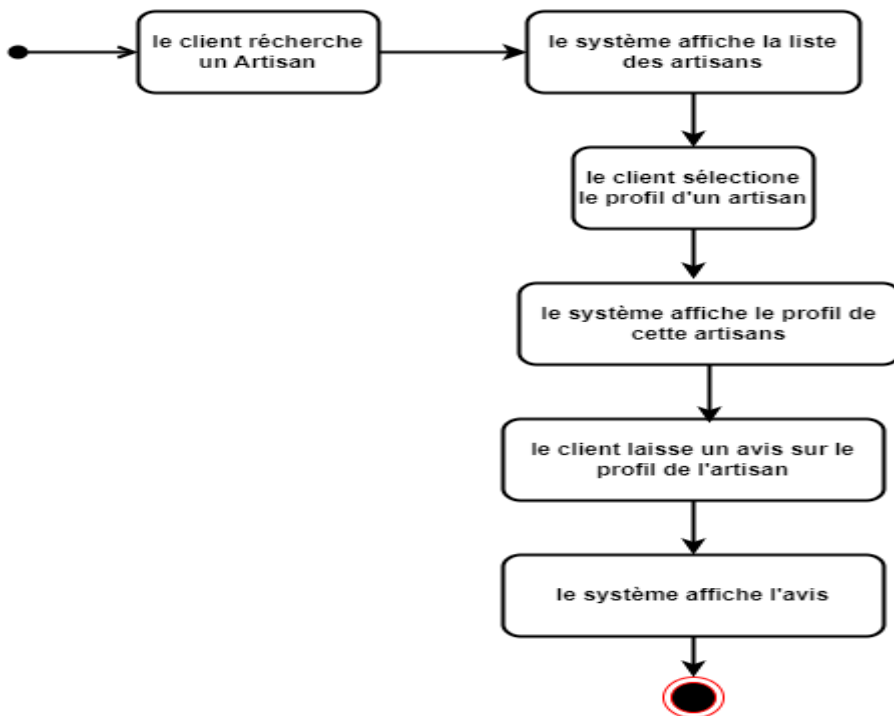


Figure 15. Diagramme d'activité « Publier envi »

III.4.4. Diagramme d'état transition

Les diagrammes d'états-transitions permettent de modéliser le comportement dynamique d'un objet via un automate à états finis, décrivant les séquences d'états et les réactions à des événements (Lano & Kolahdouz-Rahimi, 2011).

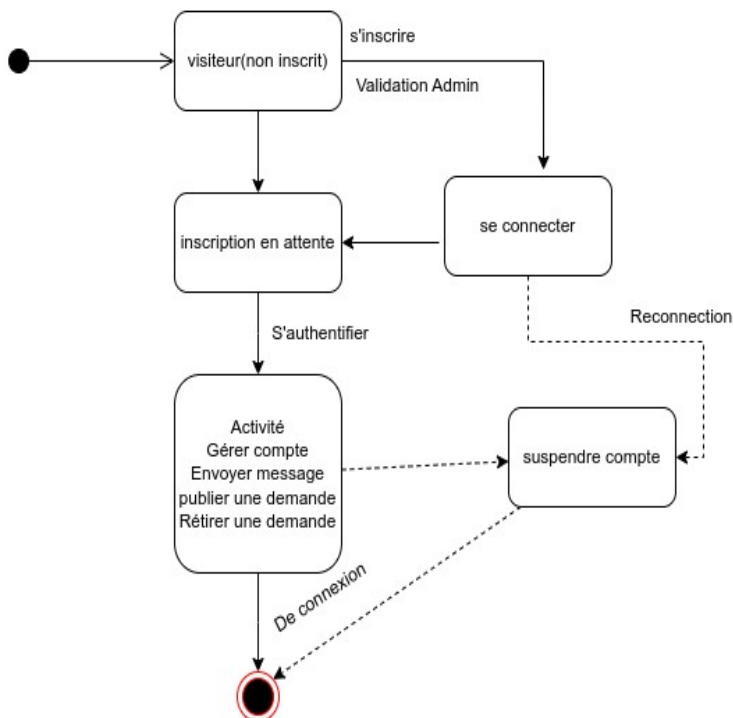


Figure 16 Diagramme d'état transition

III.4.5 Diagramme de Package (Cas d'utilisation)

Le diagramme de package en UML est une structure permettant d'organiser et de regrouper des éléments du modèle, tels que des cas d'utilisation, des classes ou des sous-systèmes, en packages ou modules logiques (Alturas, 2023).

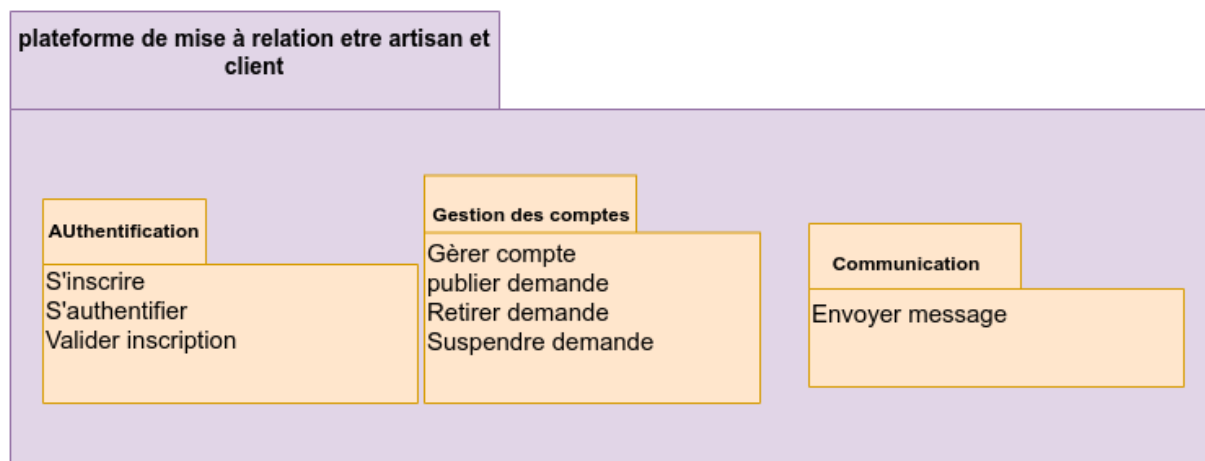


Figure 17 Diagramme de Package (Cas d'utilisation)

III.4.6 Matrice de validation des cas d'utilisations

Une matrice de validation des cas d'utilisation est un outil utilisé dans l'ingénierie logicielle pour vérifier que tous les cas d'utilisation définis dans un système ont été correctement implémentés et testés(Alturas, 2023).

	Visite site	s'inscrire	s'authentifier	Consulter publication	Valider inscription	Modérer envie	Gérer compte	Envoyer message	Publier demande
Visite site									
Inscrire les artisan et le client									
Publication d'une demande									
Publier un envie d'un client									
Messagerie entre artisan et client									

Tableau 1 Matrice de validation des cas d'utilisations

III.5 MODÉLISATION STATIQUE DU SYSTEME

Dans cette partie nous allons décrire la structure du système en termes d'éléments statiques comme les classes, objets et les relations entre eux.

III.5.1 Conception du modèle Conceptuel des Données (Diagramme des Classes)

Les diagrammes de classes sont l'un des types de diagrammes UML les plus utiles, car ils décrivent clairement la structure d'un système particulier en modélisant ses classes. Ce diagramme constitue l'un des diagrammes pivots et essentiels dans la modélisation. En effet, il permet de décrire les différentes classes qui, de surcroît, donnent naissance à des objets ayant la même structure de données (mêmes noms et mêmes attributs), mais qui se distinguent par les valeurs respectives de leurs attributs. Néanmoins, ils jouissent des mêmes opérations de la classe génératrice (Claude, 2018).

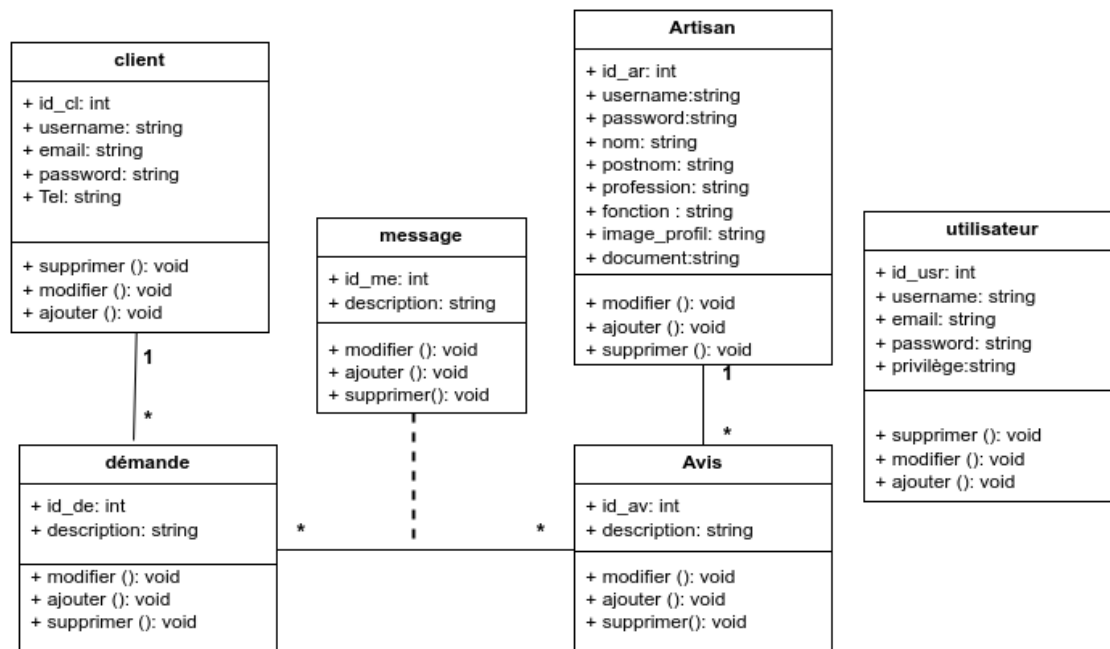


Figure 18 Diagramme des Classes

III.5.2 Conception du Modèle L

le schéma relationnel est défini dans le cadre du processus de transformation d'un modèle conceptuel en un modèle logique relationnel (Teorey et al, 1986).

Demande(id_de,description,#id_cl)

Artisan(id_ar,username,password,nom,postnom,profession,fonction,image_profil,document)

client(id_cl,username,email,password,Tel)

Avis(id_av,description,#id_ar)

message(id_me,description,#id_de,#id_av)

utilisateur(id_usr,username,email,password,privilège)

III.5.3 Diagramme d'objets

Représente des instances spécifiques de classe(objets) à un moment donné. Il montre comment les objets sont interconnectés, leurs valeurs actuelles, et leurs relations.

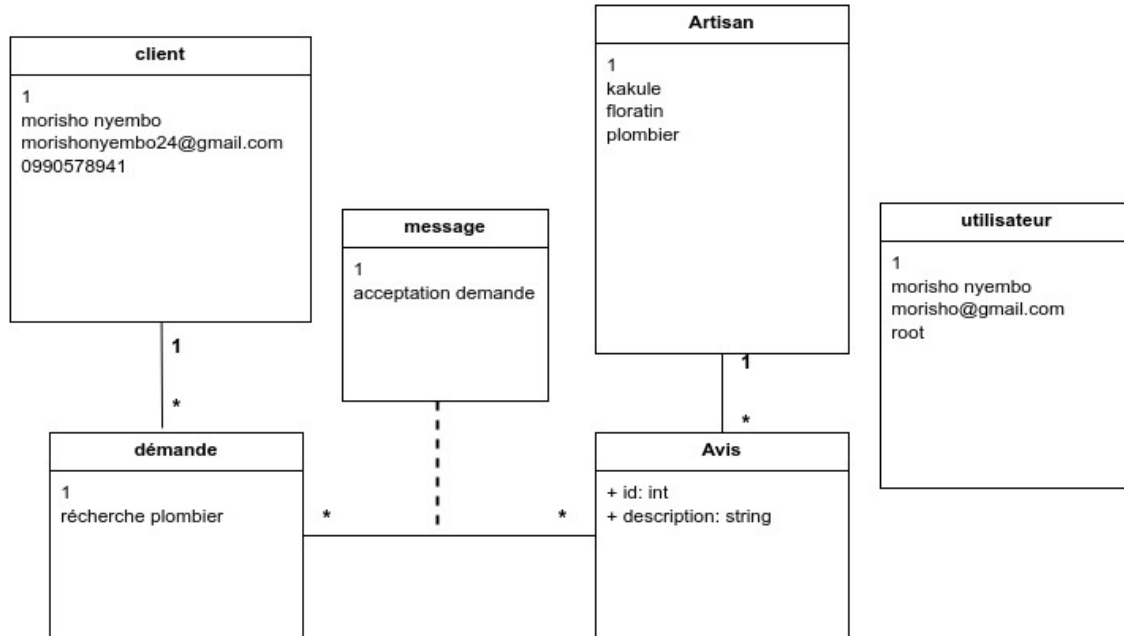


Figure 19 Diagramme d'objets

III.5.4 Conception des interfaces

La conception des interfaces en UML consiste à modéliser formellement les composants et les interactions d'une interface utilisateur à travers des extensions spécifiques des diagrammes UML. Cette modélisation intègre à la fois la structure statique des éléments graphiques (tels que fenêtres, boutons, menus) et leur comportement dynamique face aux actions de l'utilisateur. L'objectif est d'assurer une représentation claire et cohérente de l'interface, facilitant ainsi la communication entre les développeurs et la cohérence avec l'architecture logicielle sous-jacente(Campos & Nunes, 2005).

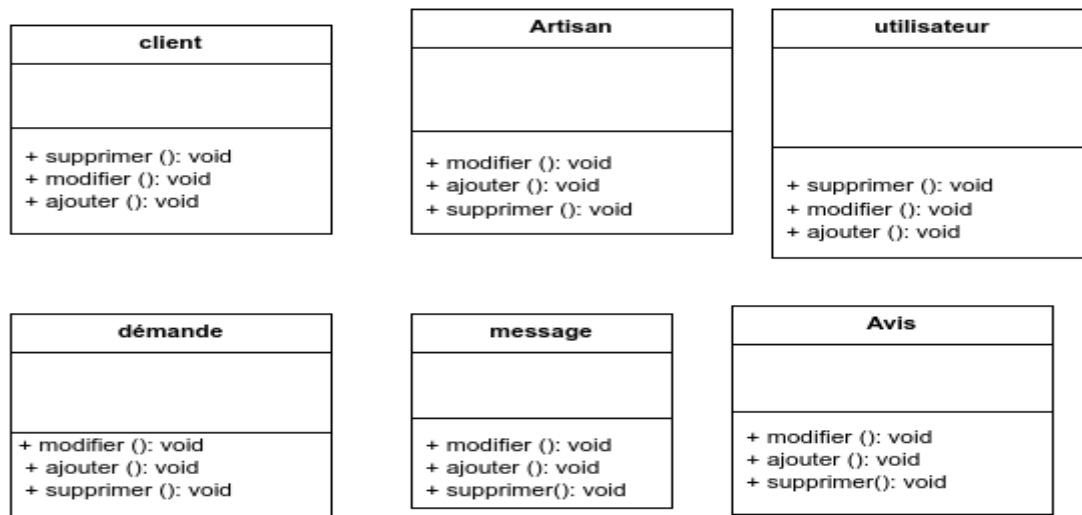


Figure 20 Conception des interfaces

III.5.5 Diagramme de Déploiement

Le diagramme de déploiement UML est une représentation structurelle qui décrit l'architecture physique d'une application, en précisant les relations entre les composants logiciels et matériels, ainsi que l'évolution de l'application en termes de réseau et de communication (Kabunga Kitengera & Kambale Kasambya, 2024).

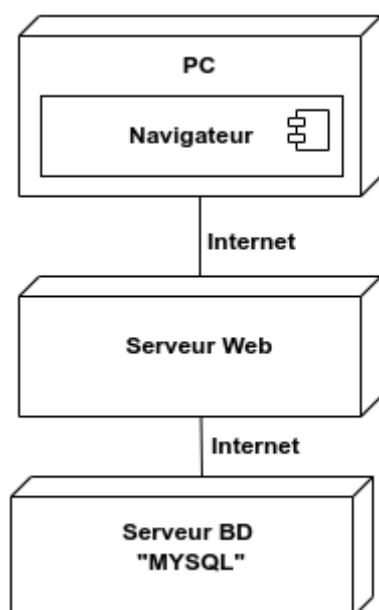


Figure 21 Diagramme de Déploiement

III.6. Conclusion

Dans ce chapitre, nous avons fait la modélisation du système futur en nous servant du langage de modélisation UML piloté par les cas d'utilisations. Enfin, ce chapitre nous a permis de préparer la phase de réalisation qui concrétisera tout ce qui a été présenté jusque-là.

Bibliographie

1. Siau, K., & Cao, Q. (2019). Use case diagrams in support of use case modeling: Deriving requirements and system understanding. *Journal of Systems and Software*, 149, 1–14. <https://doi.org/10.1016/j.jss.2019.110429>
2. Riccio, P.-M. (2007). Une méthodologie unifiée pour la construction de systèmes de qualité dans le domaine des TIC. *Communication et organisation. Revue scientifique francophone en Communication organisationnelle*, 15, Article 15. <https://doi.org/10.4000/communicationorganisation.2188>
3. (PDF) DEVELOPMENT OF EMPLOYEE ATTENDANCE AND MANAGEMENT SYSTEM USING QUICK RESPONSE (QR) CODE IN SORSOGON STATE UNIVERSITY, CASTILLA CAMPUS, PHILIPPINES. (2024). *ResearchGate*. <https://doi.org/10.46827/ejes.v9i11.4545>
4. Viot, D. (2023). *Méthodes Informatiques*. Sorbonne Université, LPTMC. <https://www.lptmc.jussieu.fr/user/viot/COURS/MI2023.pdf>
5. N'KPONON, C. R. (2023). *MÉTHODOLOGIE AGILE UNIFIÉE ET APPORTS À LA MISE EN ŒUVRE D'UN PROJET INFORMATIQUE : CAS DU LOGICIEL DE GESTION DES APPRENANTS AU CENTIC* [Mémoire de Licence, Université des Sciences et Technologies du Bénin (USTB)]. <https://d1wqtxts1xzle7.cloudfront.net/103950283/MemoireLicence-libre.pdf>
6. Shankar, P., Morkos, B., Yadav, D., & Summers, J. D. (2020). Towards the formalization of non-functional requirements in conceptual design. *Research in Engineering Design*, 31(4), 449-469. <https://doi.org/10.1007/s00163-020-00345-6>
7. Coulon, A. (2019, février). *Le cahier des charges d'un système d'information*. Espaces numériques. <https://espaces-numeriques.org/wp-content/uploads/2019/02/158p38.pdf>
8. Hadzovic, S., Mrdovic, S., & Radonjic, M. (2021). Identification of IoT Actors. *Sensors*, 21(6), Article 6. <https://doi.org/10.3390/s21062093>
9. Mukoma, G., & Mbaya Musaka, D. (2024). *Développement d'une plateforme web d'évaluation des enseignements par les étudiants en République Démocratique du Congo*. Zenodo. <https://doi.org/10.5281/zenodo.14232064>
10. Kabunga Kitengera, C., & Kambale Kasambya, M. (2024). *Développement d'une plateforme web d'évaluation des enseignements par les étudiants en République démocratique du Congo*. *Revue Internationale Multidisciplinaire Etincelle*, 25(2). <https://doi.org/10.61532/rime252117>

11. Lano, K., & Kolahdouz-Rahimi, S. (2011). *Solving the TTC 2011 Model Migration Case with UML-RSDS*. In *Proceedings of the 33rd International Conference on Software Engineering (ICSE)* (pp. XYZ–XYZ). IEEE.
<https://doi.org/10.1109/ICSE.2011.102>
12. Alturas, B. (2023). Connection between UML use case diagrams and UML class diagrams: a matrix proposal. *International Journal of Computer Applications in Technology*, 72(3), 161–168. <https://doi.org/10.1504/IJCAT.2023.133294>
13. Claude, P. (2018). *Introduction à la modélisation UML*. Éditions Techniques.
14. Teorey, T. J., Yang, D., & Fry, J. P. (1986). A logical design methodology for relational databases using the extended entity-relationship model. *ACM Computing Surveys*, 18(2), 197–222. <https://doi.org/10.1145/7474.7475>
15. Campos, P. F., & Nunes, N. J. (2005). A UML-based tool for designing user interfaces. In N. Jardim Nunes, B. Selic, A. Rodrigues da Silva, & A. Toval Alvarez (Eds.), *UML Modeling Languages and Applications* (Vol. 3297, pp. 321–334). Springer.
https://doi.org/10.1007/978-3-540-31797-5_32

