

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid - Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études
Pour l'obtention du Diplôme de Master en Informatique
Option : Génie Logiciel (G.L)

Thème :

CONCEPTION ET REALISATION D'UNE PLATEFORME POUR
L'ANALYSE AUTOMATIQUE EN BOITE NOIRE DES SOLUTIONS DE
T.P POUR LE MODULE DE PROGRAMMATION ORIENTEE-OBJET

Réalisé par :

✓ **SIDIBE Ismaïla**

Soutenu publiquement le 13 Juin 2024 devant le jury composé de :

- Pr. CHIKH Azzedine (Président).
- Dr. CHOUITI Sidi Mohamed (Examineur).
- Dr. SETTOUTI Ahmed Khalid Yassine (Encadrant).

Année universitaire : 2023-2024

Dédicace

Je dédie ce travail à mes parents, dont le soutien inconditionnel et les encouragements constants ont été une source inépuisable de motivation et de force. Leur amour, Leurs sacrifices et leur confiance en moi ont été essentiels pour mener à bien ce projet.

A toute ma famille, qui a toujours été présente pour moi, m'offrant soutien, conseils et réconfort tout au long de ce parcours académique, je vous exprime ma profonde gratitude.

Ce mémoire est le fruit de vos encouragements et de votre amour. Merci du fond du cœur.

Avec toute mon affection.

Remerciements

Je remercie DIEU, de m'avoir permis de réaliser ce travail en santé.

Je souhaite exprimer ma profonde gratitude à toutes les personnes qui m'ont soutenu et aidé durant la réalisation de ce mémoire.

Je remercie le gouvernement algérien pour m'avoir accordé cette bourse d'étude, me permettant ainsi de poursuivre mon projet académique dans les conditions optimales.

Mes remerciements chaleureux vont également à mon encadrant, Docteur SETTOUTI Ahmed Khalid Yassine, pour ses encouragements, son accompagnement, ses précieux conseils et son soutien. Son expertise et sa disponibilité ont été essentiels à la réalisation de ce travail.

Je souhaite aussi remercier tous les professeurs du département d'informatique pour leur enseignement de qualité et leur encouragement tout au long de mon cursus. Leur savoir et leur passion pour l'informatique ont grandement enrichi mon parcours académique.

Je tiens également à exprimer mes sincères remerciements aux honorables membres du jury pour l'honneur qu'il me font en évaluant mon travail.

Enfin, un grand merci à tout le personnel administratif et technique du département d'informatique pour leurs aides précieuses et leur soutien.

A tous, je vous adresse ma plus profonde gratitude.

Table des matières

Table des Figures.....	vi
Liste des Tableaux	viii
Introduction Générale.....	1
Chapitre I : Généralités	4
1.1 Introduction.....	5
1.2 Langage de Programmation Java	5
1.2.1 Un Peu d'historique sur la Programmation.....	5
1.2.2 Définition du langage java	6
1.2.3 Programmation Orientée Objet (P.O.O).....	6
1.3 Analyse de Programme Informatique.....	7
1.3.1 Analyse statique d'un programme informatique	7
1.3.2 Analyse dynamique d'un programme informatique.....	8
1.4 Evaluation Automatique De Code Informatique	8
1.4.1 Test.....	8
1.4.2 Utilisation des Jeux de tests	9
1.5 UML	9
1.5.1 Vues UML	9
1.6 La Méthode de conception Utilisé	10
1.6.1 Etude de faisabilité	11
1.6.2 Spécifications.....	11
1.6.3 Conception Générale	11
1.6.4 Conception détaillé	11
1.6.5 Programmation.....	12
1.6.6 Validation et vérification	12
1.7 Patron d'architecture Utilisé	12
1.7.1 Modèle-Vue-Contrôleur (MVC).....	12
1.8 Conclusion	13
Chapitre II : Travaux similaires	14
2.1 Introduction.....	15
2.2 Présentations de quelques plateformes	15
2.2.1 HacherRank	15
2.2.2 Sphere Online Judge (SPOJ).....	17
2.2.3 HackerEarth	20
2.2.4 CoderByte.....	23

2.3	Etudes comparative des travaux similaires	27
2.3.1	Etude comparative des plateformes.....	27
2.3.2	Etude comparative des travaux dans la littérature	29
2.4	Conclusion	31
Chapitre III : Conception et Spécifications fonctionnelles		32
3.1	Introduction.....	33
3.2	Spécifications fonctionnelles	33
3.2.1	Analyse des besoins.....	33
3.2.2	Acteurs.....	34
3.2.3	Exigences fonctionnelles.....	34
3.2.4	Exigences non fonctionnelles	35
3.3	Conception du système	35
3.3.1	Diagrammes de cas d'utilisation	35
3.3.1.1	Description textuelle des principaux cas d'utilisation	37
3.3.2	Diagrammes de séquence	39
3.3.3	Diagramme de classes.....	43
3.3.4	Diagrammes de Séquence Objet.....	45
3.3.5	Diagramme de classe détaillé du système	48
3.3.6	Diagramme de classe objet	49
3.3.7	Scénario Global pour l'utilisation de notre plateforme	50
3.4	Conclusion	51
Chapitre IV : Présentation du système.....		52
4.1	Introduction.....	53
4.2	Domaines technique et technologiques utilisés.....	53
4.2.1	Options pour les outils de développement.....	53
4.2.2	Sélection des outils et justifications.....	54
4.3	Formation dans le domaine.....	55
4.4	Réalisation des Fichiers attendues.....	56
4.5	Réalisation des Cas de tests.....	57
4.6	Architecture du système	58
4.6.1	Côté enseignant.....	58
4.6.2	Côté étudiant.....	59
4.7	Implémentation de notre plateforme « Test Coding ».....	60
4.7.1	Page d'accueil.....	60
4.7.2	Formulaire d'authentification	61
4.7.3	Côté enseignant.....	62

4.7.4	Côté étudiant(e)	64
4.8	Avantages et bénéfices	69
4.9	Comparaison entre notre projet et les travaux similaires.....	70
Conclusion Générale		72
Bibliographie.....		75

Table des Figures

Figure 1. 1 : Diagramme chronologique représentant l'apparition des langages de programmation.....	6
Figure 1. 2 : La vue « 4+1 » de ph. Kruchten [27]	10
Figure 1. 3 : Waterfall model [28]	10
Figure 1. 4 : Design pattern MVC [73]	13
Figure 2. 1 : Page d'accueil de HackerRank [30].....	16
Figure 2. 2 : Dashboard de HackerRank [30]	17
Figure 2. 3 : Page d'accueil de SPOJ [32].....	18
Figure 2. 4 : Liste des problèmes classique à résoudre sur SPOJ [32]	19
Figure 2. 5 : La spécification du problème à résoudre [32]	19
Figure 2. 6 : Statuts de la compilation du problème [32]	20
Figure 2. 7 : Page d'accueil de HackerEarth [33]	21
Figure 2. 8 : Liste des problèmes basiques sur HackerEarth [33]	22
Figure 2. 9 : Spécifications et soumissions du problème [33]	22
Figure 2. 10 : Résultats de la soumission [33]	23
Figure 2. 11 : Page d'accueil de CoderByte [34]	24
Figure 2. 12 : Spécifications du problème et éditeur de code [34]	25
Figure 2. 13 : Liste des problèmes sur CoderByte [34]	25
Figure 2. 14 : Déroulement du cas de test [34]	26
Figure 2. 15 : Résultats selon les cas de test valide [34]	27
Figure 3. 1 : Diagramme de cas d'utilisation pour l'acteur « Etudiant ».....	36
Figure 3. 2 : Diagramme de cas d'utilisation pour l'acteur « Enseignant ».....	37
Figure 3. 3 : Diagramme de séquence télécharger TP	40
Figure 3. 4 : Diagramme de séquence rendre TP	41
Figure 3. 5 : Diagramme de séquence d'Ajout TP.....	42
Figure 3. 6 : Diagramme de classes système.....	44
Figure 3. 7 : Diagramme de séquence objet d'ajout du travail pratique.....	45
Figure 3. 8 : Diagramme de séquence objet téléchargement du travail pratique.....	46
Figure 3. 9 : Diagramme de Séquence Objet Soumission et Evaluation du TP	47
Figure 3. 10 : Diagramme de classe détaillée du système.....	48
Figure 3. 11 : Diagramme d'Objets	49
Figure 4. 1 : exemple de fichier attendus.....	56
Figure 4. 2 : Exemple de cas de test	57
Figure 4. 3 : Architecture du système côté enseignant.....	58
Figure 4. 4 : Architecture du système côté étudiant.....	59
Figure 4. 5 : Page d'accueil du plateforme Test Coding	60
Figure 4. 6 : Page d'authentification.....	61
Figure 4. 7 : Page de la Liste des tps	62
Figure 4. 8 : Page d'ajout du travail pratique.....	62
Figure 4. 9 : Page de la liste des étudiants.....	63
Figure 4. 10 : Page du profils détails de l'étudiant	64

Figure 4. 11 : Page de modification du profil de l'étudiant	64
Figure 4. 12 : Page des travaux pratiques	65
Figure 4. 13 : Page de soumission du travail pratique	66
Figure 4. 14 : Page de la note.....	67
Figure 4. 15 : Page de soumission avec du nom de fichier incorrecte.....	68
Figure 4. 16 : Page après soumission du nom de fichier incorrect.....	69

Liste des Tableaux

Tableau 2. 1 : Etude comparative des différentes plateformes	28
Tableau 2. 2 Tableau comparative des projets similaires.....	30
Tableau 3. 1 : Description textuelle du cas d'utilisation "Ajouter un T.P"	38
Tableau 3. 2 : Description textuelle du cas d'utilisation Rendre une solution de T.P	39
Tableau 4. 1 : Etude Comparatives de notre plateforme vs plateforme similaires	70

Introduction Générale

L'enseignement universitaire rencontre certaines problèmes tels que les délais serrés pour achever les programmes, le manque de prérequis, l'insuffisance des ressources matérielles. Notamment en ce qui concerne l'enseignement de l'informatique est un domaine relativement nouveau par rapport à d'autres disciplines académiques. Contrairement à des domaines plus établis, comme les mathématiques ou les sciences sociales, l'informatique n'a pas encore accumulé autant d'expérience pédagogique au fil des décennies. Cela signifie que les enseignants d'informatique peuvent souvent se retrouver à développer des méthodes d'enseignement innovantes pour compenser le manque d'expertise éducative acquise au fil du temps. L'informatique est également caractérisée par un rythme d'évolution rapide. Les avancées technologiques et les nouvelles découvertes dans ce domaine se produisent à un rythme vite, ce qui signifie que les enseignants doivent constamment mettre à jour leurs connaissances et leurs programmes d'études pour refléter les développements les plus récents. En plus des problèmes préalablement mentionnés dans l'enseignement universitaire, nous devons également faire face à des défis spécifiques au niveau de l'enseignement de l'informatique, tels que le besoin constant d'adaptation aux avancées technologiques et le développement de nouvelles méthodes pédagogiques pour répondre aux besoins changeants de nos étudiants. Cette nécessité constante de mise à jour soulève également la question des outils et des ressources appropriés pour soutenir efficacement l'enseignement de l'informatique à l'université.

La Programmation Orientée-Object est un module enseigné au niveau du 2^{ème} Année Licence et 2^{ème} Année Ingénieur en Informatique. Des séances de travaux pratique en présentiels sont effectuées pour ce module permettent aux étudiants de mettre en pratique les concepts théoriques abordés en cours. Pendant les séances de Travaux pratique les étudiants sont accompagnés par un ou deux assistant(e)s. Pour chaque séance, les étudiants se voit attribuer une série d'exercice qu'il devrait résoudre avant la fin de la séance. A l'approche de la fin de séances chaque étudiant présent devra retourner sa solution. L'assistant(e) oriente les étudiants au cours de la séance, corrige leurs solutions, les évalue et relève leurs présences. Chaque travail retourné est évalué selon les contraintes qu'il satisfait.

Notre projet va consister à gérer ces séances de travaux pratique à distance en mettant en place un système qui serait capable de tester et d'évaluer les codes des étudiants qui ont été réalisés en se basant sur les exercices des tps donnés d'une manière automatisée. Cependant avec le niveau de difficulté de l'évaluation des Tps de la programmation orientée-objet en présentiels augmente chaque année avec l'augmentation du nombre des étudiants présents dans les salles de TP, les contraintes de temps sont devenues de plus en plus serrées. Avec des groupes pouvant atteindre jusqu'à 40 étudiants, un assistant seul n'a que 4,5 minutes à consacrer à chaque

étudiant durant une séance de TP de 3 heures. Si l'assistant(e) est accompagné par un autre, le taux passe à 9 minutes. Dans les 2 cas, les assistant(e)s doivent expliquer les ambiguïtés, orienter l'étudiant, corriger ses erreurs spécifiques et l'évaluer en moins de 9 minutes (respectivement 4,5 minutes si l'assistant est seul). Face à ce défi croissant, le problème ne risque pas de se régler vu que le recrutement des enseignants se fait de plus en plus rare et les promotions sont de plus en plus grandes (au point de dépasser les capacités physiques théoriques de la faculté et du département). Face à ces différents problèmes rencontrés durant les séances de Tps, les enseignants réagissent souvent de différentes manières. Soit, ils sacrifient la correction, au profit de l'orientation et l'évaluation, Soit, ils laissent l'évaluation pour plus tard (après la séance), afin de pouvoir orienter les étudiants plus efficacement. Dans les 2 cas, l'effort déployé par le(s) assistant(s) n'est ni efficace, ni optimal.

Notre projet vise à créer une plateforme web aidant l'étudiant et l'enseignant durant les séances de travaux pratiques, en corriger automatiquement les solutions retournées par les étudiants, selon des cases de tests spécifiés par l'assistant(e) de TP ou le chargé de module. L'idée est de procéder à l'analyse en boîte noire du code et de vérifier s'ils répondent aux spécifications énoncées.

L'étude de ce projet permet d'offrir aux enseignants une meilleure gestion de temps ainsi facilitant la gestion du travail. Une détection plus facile des erreurs communes (attendues, prévues par les assistants(e)s ou le chargé de module) pour les étudiants. Et une consultation facile de sa note à chaque TP. Pour l'administration (Un apprentissage plus efficace d'une plus grande masse d'étudiants par la masse actuelle des enseignants au niveau du département).

Dans la suite de notre mémoire, nous allons d'abord analyser les solutions existantes dans le même domaine, tout en comparant entre elles. Après, nous allons concevoir, et présenter le prototype fonctionnel du projet. Enfin, nous allons comparer notre application aux solutions existantes, et conclure la thèse.

Chapitre I : Généralités

1.1 Introduction

Dans ce chapitre, nous explorerons les concepts nécessaires à notre projet, qui vise à automatiser l'évaluation des travaux pratiques en informatique à distance, plus spécifiquement dans le cadre du module de Programmation Orientée Objet (P.O.O). Pour atteindre cet objectif, il est essentiel de comprendre plusieurs éléments clés, notamment le langage de programmation spécifique utilisé dans les travaux pratiques de P.O.O dans notre cas, les méthodes d'analyse de code informatique existantes, ainsi que la Méthode de conception que nous adopterons pour la réalisation du projet.

Tout au long de ce chapitre, nous mettrons en avant les définitions essentielles, les concepts de base ainsi que les outils et techniques appropriés pour chaque aspect abordé. Cela établira les bases solides indispensables à une compréhension approfondie et une mise en œuvre réussie de notre projet.

1.2 Langage de Programmation Java

Dans cette section, nous aborderons l'importance de la programmation du langage java dans le cadre de notre projet. Nous commencerons par retracer brièvement l'histoire de la programmation, Ensuite, Nous concentrerons sur les principes fondamentaux du langage java. Le choix pour ce langage de programmation découle de notre projet qui vise à automatiser les travaux pratiques des étudiants en fournissant des code sources écrits spécifiquement en langage java. Ceci dit le projet pourrait facilement être adapter ou dès le début appliqué à d'autres modules pratiques (langage C, Python, PHP, C++...).

1.2.1 Un Peu d'histoire sur la Programmation

L'évolution de la discipline de la programmation au fil des années a été façonnée par des moment clés qui ont marqué son développement. Voici un aperçu des principales dates qui ont contribué à son évolution :

Au début des années 50, les praticiens au domaine de l'informatique proposaient des définitions claires de la programmation [1]. En 1950, Hartree dans [2] définit la programmation comme suit « *Le processus de préparation d'un calcul pour une machine se divise en deux parties, programmer et coder. La programmation est le processus d'élaboration de la séquence d'opérations requise pour réaliser le calcul* ». Wilkes dans [3] donnait une autre définition de la programmation comme suit : « *La séquence d'ordre est connue sous le nom de programme, et la machine l'exécute automatiquement sans intervention de l'utilisateur* ». Ces définitions sont apparues avec l'utilisation en parallèle des ordinateurs dans les calculs. Cela a entraîné

l'apparition des premiers langages de programmation, comme [1] Fortran (Mathematical FORMula TRANslating System) et LISP (LISt Processing Language) [4] [5].

Cependant, au cours des années, et surtout avec l'expansion de l'utilisation de l'ordinateur et de l'invasion dans divers domaines, toute une nouvelle génération de langages est apparue tels Basic (1964), Pascal (1970), C, Prolog (1972) et SmallTalk (1972). Au cours des années 90, avec l'adoption et la popularisation massive de l'orientée objet, des langages tel que java, perl et python ont adoptés les principes fondamentaux du paradigme de programmation orientée-objet [1].

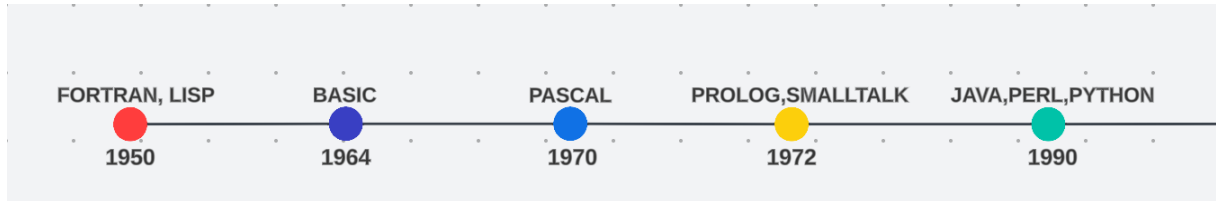


Figure 1. 1 : Diagramme chronologique représentant l'apparition des langages de programmation

1.2.2 Définition du langage java

Java est un langage de programmation de haut niveaux, multiplateforme, orientée-objet créé par James Gosling et Patrick Naughton en 1990, employés de Sun Microsystems [6] [7].

1.2.3 Programmation Orientée Objet (P.O.O)

Le terme P.O.O est un acronyme qui signifie Programmation Orientée-Objet ou Programmation Objet. C'est un paradigme de programmation qui s'appuie sur les concepts d'objets. Un objet est une abstraction d'un élément du monde réel [8] comme une personne, une voiture ou une page de livre [9]. L'objectif de la P.O.O est de faciliter l'écriture et la maintenance du code [10].

Voici quelques langages de programmation qui adoptent le paradigme de la programmation orientée objet [11] :

- Java ;
- C++ ;
- PHP (depuis la version 4) ;
- Ruby ;
- Fortran ;
- Pascal Objet ;
- Smalltalk (totalement objet) ;

1.3 Analyse de Programme Informatique

L'analyse de programme informatique est un domaine de l'informatique qui vise à étudier les propriétés d'un programme informatique. Cela se fait en examinant le code source du programme, sa structure et son comportement. Nous distinguons deux types d'analyse de programme informatique à savoir : *Analyse statique* et *Analyse dynamique* du programme informatique [12].

1.3.1 Analyse statique d'un programme informatique

L'analyse statique consiste à analyser le texte d'un programme pour en extraire de l'information [13]. Cette analyse est effectuée sans exécuter le programme [13]. Le type d'information extraite ainsi que les techniques choisies dépendent de l'utilisation à laquelle l'analyse est destinée [14]. L'analyse statique est utilisée pour repérer des erreurs formelles de programmation, logiques, conception, sécurité, mais aussi pour déterminer la facilité ou la difficulté à maintenir le code [13].

✓ Quelques outils d'analyse du code statique

Ils existent plusieurs outils d'analyse du code statique, chacun offrant des fonctionnalités et des points forts spécifiques. Voici quelques exemples d'outils populaires :

- **Checkmarx** [15].: C'est un outil payant spécialisé dans la sécurité des applications. Détecter les vulnérabilités OWASP¹, les failles de sécurité et les injections SQL, offre des recommandations pour corriger les vulnérabilités identifiées.
- **SonarQube** [16].: Un outil open source (mais payant pour les langages C et C++ d'après [17]) largement utilisé pour la qualité (détection des bugs, des mauvaises pratiques de codage et des doublons) et la sécurité du code (détection des vulnérabilités).
- **Coverity** [18].: C'est un outil propriétaire utilisé pour l'analyse statique de code source C, C++, C#, Java et Javascript.
- **FindBugs** [19].: Est un logiciel libre d'analyse statique de byte code Java. Son but est de trouver des bugs dans les programmes Java en identifiant des patterns reconnus comme étant des bugs [20].

¹ **OWASP** : selon [72] le terme OWASP se définit comme **Open Web Application Security Project** mais actuellement il a été remplacé par **Open Worldwide Application Security Project**, c'est une communauté en ligne travaillant sur la sécurité des applications web.

1.3.2 Analyse dynamique d'un programme informatique

Cette analyse se base sur l'exécution de programme pour l'évaluer, elle est centrée sur le comportement dynamique de programme en prenant en compte plusieurs aspects tels que : la fonctionnalité, l'efficacité, jeux de tests [1] ... etc.

1.4 Evaluation Automatique De Code Informatique

L'évaluation automatique de code informatique est un processus par lequel des programmes informatiques sont utilisés pour analyser, tester et évaluer automatiquement les code. Comme expliqué dans la section 1.2, elle se focalise sur l'analyse du code source. Après cela, elle passe à l'évaluation du travail de l'étudiant à travers des jeux de test automatisés.

1.4.1 Test

« Le test est l'exécution ou l'évaluation d'un système ou d'un composant par des moyens automatiques ou manuels, pour vérifier qu'il répond à ses spécifications ou identifier les différences entre les résultats attendus et les résultats obtenus » [21].

Nous distinguons plusieurs niveau et type de Test :

❑ Niveau de test

- **Test unitaire** : Le plus bas niveau du test, où tous les petits blocs de code sont testés séparément [14].
- **Test système** : Teste le système comme un seul bloc [14]. C'est un processus de test d'un système intégré afin d'évaluer sa conformité aux exigences spécifiées [22].
- **Test d'acceptation** : Effectué par le client pour s'assurer de la conformité au besoin [23].
- **Test de non régression** : Exécuté durant la maintenance, après toute modification, correction ou mise à jour du système, pour vérifier que le système fonctionne toujours correctement. Typiquement, le test de régression est utilisé après l'ajout de nouvelles fonctionnalités à un système déjà testé. Dans ce cas, on répète habituellement les tests exécutés précédemment pour vérifier que les anciennes fonctionnalités fonctionnent [14].
- **Test de surcharge** : consiste à tester le système dans les conditions extrêmes ou bien tester des entrées non-prévues [14].

❑ Type de test

- **Test de boîte noire (ou test fonctionnels)** : Concevoir des tests grâce aux spécifications trouver les jeux de valeurs qui permettront de découvrir les comportements incorrects du système [24].
- **Test de boîte blanche (ou test structurels)** : Cas de tests générés en partant du code, en inspectant les chemins d'exécution du système [23].

1.4.2 Utilisation des Jeux de tests

Afin d'effectuer de manière automatisée l'évaluation du travail d'un étudiant, on se restreint généralement à un certain type de programmes qui lisent des données de travail (les entrées) et écrivent des résultats (la sortie) [25].

À chaque exercice du travail pratique est alors associé un ensemble de tests (le jeu de tests), chacun consistant en une entrée et une sortie de référence. En comparant, pour chaque test, la sortie de référence à la sortie produite par le programme quand on lui donne cette entrée, il est possible de détecter les éventuelles différences, indiquant que le programme n'a pas donné la sortie attendue. En fonction du nombre de tests sur lesquels le programme a donné le bon résultat, il reçoit une note [25]. Si le jeu de tests est bien construit il permet de déterminer de manière très simple et rapide la correction d'un programme.

1.5 UML

Le langage de modélisation unifié (**UML**) (en anglais Unified Modeling Language) est selon **OMG** un langage visuel dédié à la spécification, la construction et la documentation des artefacts d'un système logiciel [26].



Attention ! : UML est un langage...pas une méthode de conception

1.5.1 Vues UML

La représentation du logiciel a différents points de vue [27]:

- **Vue des cas d'utilisation** : vue des acteurs (besoins attendus) [27]
- **Vue logique** : vue de l'intérieur (satisfaction des besoins) [27]
- **Vue d'implémentation** : dépendances entre les modules
- **Vue des processus** : dynamique du système
- **Vue de déploiement** : organisation environnementale du logiciel

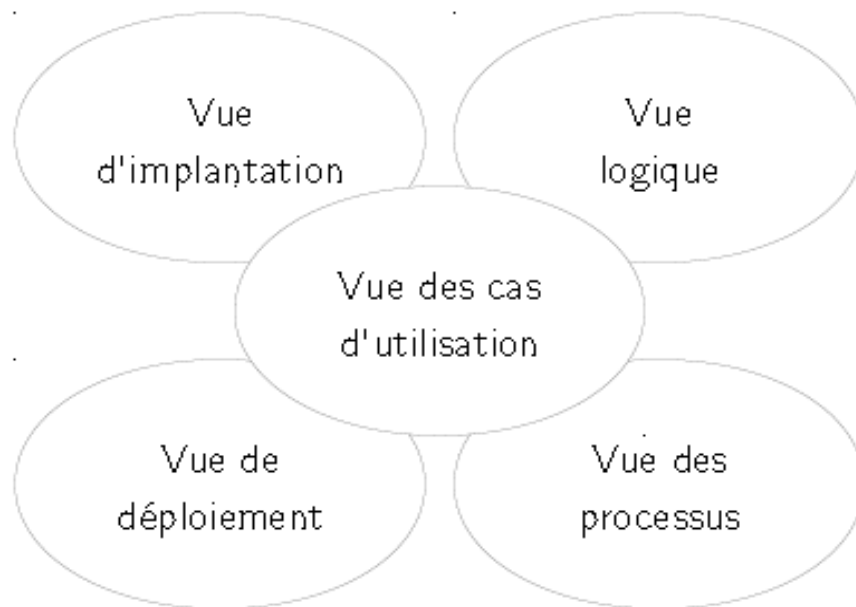


Figure 1. 2 : La vue « 4+1 » de ph. Kruchten [27]

1.6 La Méthode de conception Utilisé

En cascade avec retour en arrière [28]

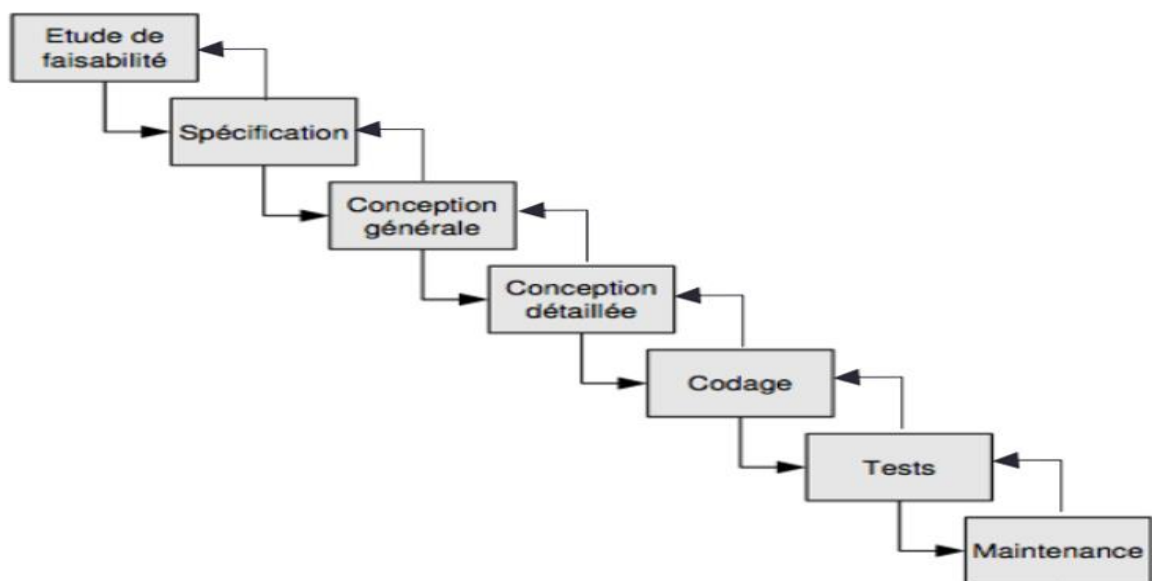


Figure 1. 3 : Waterfall model [28]

La figure 1.3 représente La méthode de conception en cascade avec retour en arrière ou appelé Waterfall model qui a été développé en 1966 et formaliser vers les années 1970 par Winston W. Royce dans son article intitulé « Managing the development of large software systems » [74]. Ce modèle (et ses variantes) décrit le cycle de vie du logiciel selon plusieurs phases séquentielles ; chaque phase est reliée soit à la phase suivante pour représenter l'enchaînement et la progression dans le développement, soit à la précédente pour montrer les corrections pouvant survenir suite à la détection d'erreurs. En principe, les retours arrière se limitent

uniquement à un retour sur la phase immédiatement antérieure. Ainsi, le contrôle imposé à la fin de chaque phase ne favorise pas les changements, et tend à stabiliser rapidement le produit[75].

1.6.1 Etude de faisabilité

Notre projet vise à développer une plateforme de test, mais une étude préalable est nécessaire avant de commencer pour répondre à certaine question :

Pourquoi développer cette plateforme ? Pour faciliter la correction et l'évaluation des travaux pratiques pour les enseignants et assister efficacement les étudiants.

1.6.2 Spécifications

❑ Analyse des besoins :

Comprendre les besoins du client

- Objectifs généraux, environnement du futur système, ressources disponibles, contraintes de performance...
- Fournie par le client (expert du domaine d'application, futur utilisateur...)

❑ Spécification :

- Établir une description claire de ce que doit faire le système (Fonctionnalités détaillées, exigences de qualité, interface...)
- Clarifier le cahier des charges (ambiguïtés, contradictions) en listant les exigences fonctionnelles et non fonctionnelles
- Exigences fonctionnelles :
 - Décrivent les fonctionnalités du système, ce qu'un système doit faire.
- Exigences non fonctionnelles :
 - Définissent les propriétés et contraintes du système.

❑ Diagrammes :

- Diagramme de cas d'utilisation, séquence, classe, séquence objet, classe objet.

1.6.3 Conception Générale

Élaborer une solution concrète réalisant la spécification en utilisant des diagrammes de classe global du système.

1.6.4 Conception détaillé

Détailler le fonctionnement des composants en décomposant la classe système. Et en utilisant des diagrammes objet de séquence et classe.

1.6.5 Programmation

Implantation de la solution conçue. Choix de l'environnement de développement, du/des langage(s) de programmation, de normes de développement...

1.6.6 Validation et vérification

- ❑ Validation : assurer que les besoins du client sont satisfaits (au niveau de la spécification, du produit fini...)
- ❑ Vérification : assurer que le système satisfait sa spécification

1.7 Patron d'architecture Utilisé

1.7.1 Modèle-Vue-Contrôleur (MVC)

[29]Définit en 1979, Le Modèle-Vue-Contrôleur est une architecture logicielle et une méthode de conception logicielle. MVC impose la séparation entre les données, les traitements et la présentation. Le découplage entre la gestion de l'interface utilisateur et la logique propre à l'application peut se faire de différentes manières [29].

- ❑ **Modèle** : contient la logique métier et l'état courant de l'interface durant le cycle dialogue avec l'utilisateur [29].
- ❑ **Vue** : porte toute la logique de présentation. Son travail consiste à afficher les données et à recevoir des interactions de l'utilisateur. La partie vue peut évidemment être composée de plusieurs « fenêtres » graphiques différentes. La vue ne gère pas les interactions avec les utilisateurs mais les délègue au contrôleur [29].

- ❑ **Contrôleur** : reçoit les interactions de l'utilisateur de la vue et les traite en modifiant le modèle. Le code à l'intérieur du contrôleur correspond essentiellement à du code de liaison entre la vue et le modèle [29].

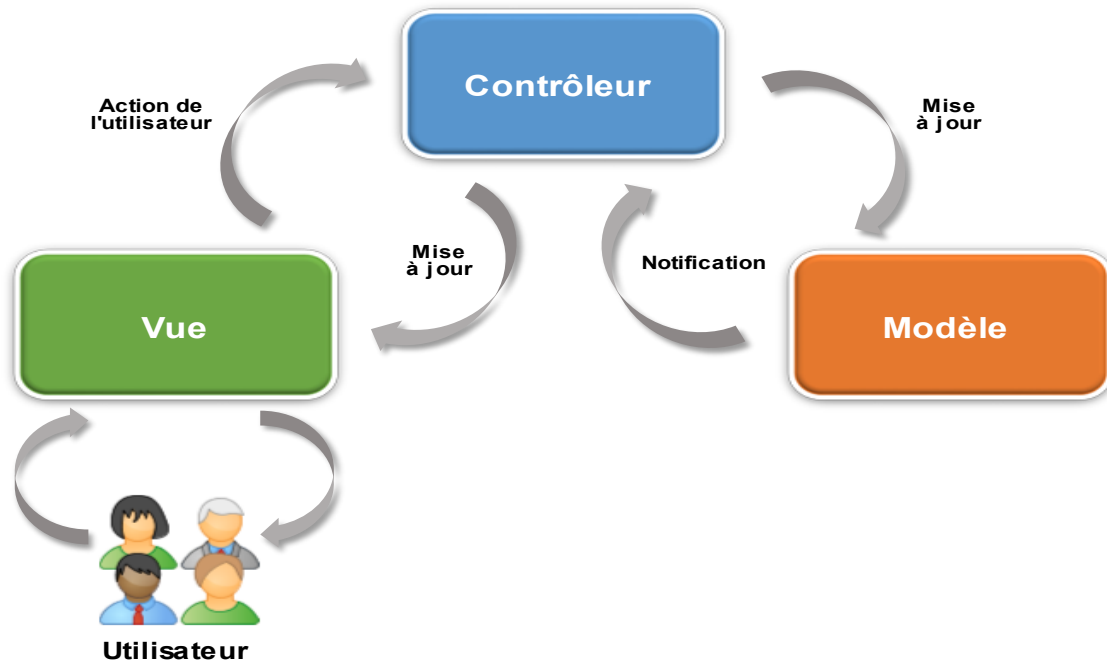


Figure 1. 4 : Design pattern MVC [73]

1.8 Conclusion

Ce chapitre a été essentiel pour clarifier les concepts fondamentaux et les outils indispensables à la réussite de notre projet. En le parcourant, nous avons pris conscience de l'importance de bien comprendre ces éléments pour mener à bien notre travail.

Chapitre II : Travaux similaires

2.1 Introduction

Avant d'entamer le développement de notre solution, il est impératif de faire une étude des travaux dans la recherche scientifique et l'innovation technologique du domaine. À cet effet, nous entreprenons une analyse approfondie des travaux antérieurs qui ont abordé des problématiques similaires ou connexes à la nôtre. Il est essentiel de nous assurer qu'il n'existe pas déjà des solutions satisfaisantes ou des approches novatrices ayant traité notre sujet. Cette étape préliminaire nous permettra d'éviter la duplication inutile d'efforts et de ressources, tout en nous offrant l'opportunité d'apprendre des succès et des échecs des travaux antérieurs. Cette analyse comparative nous permettra aussi d'identifier les solutions existantes et évaluer leurs forces, leurs faiblesses, leurs lacunes, ainsi que leurs opportunités d'innovation. Pour conclure, nous définissons le positionnement de notre travail par rapport aux autres projet concurrentes.

2.2 Présentations de quelques plateformes

Il existe plusieurs plateformes qui utilisent des techniques d'analyse et d'évaluation de programmes informatiques, parmi lesquelles nous pouvons citer :

2.2.1 HacherRank

HackerRank [30] est une plateforme en ligne qui propose une variété de défis de programmation pour les développeurs et une variété de moyens d'évaluation de compétences pour les entreprises en plein recrutement. Elle offre aux entreprises la possibilité de recruter des développeurs en ciblant leurs compétences dans des domaines précis. Par contre, Pour les développeurs, il offre plutôt des services d'entraînement et de promotion de leurs compétences à travers des défis et des publicités. Fondée en 2012, HackerRank a rapidement gagné en popularité en tant que destination de choix pour les programmeurs de tous les niveaux.

La plateforme offre une gamme diversifiée de problèmes de programmation, allant des questions d'algorithmique classiques aux défis plus avancés et couvrent de nombreux domaines de l'informatique tels que l'intelligence artificielle, algorithmes, programmation fonctionnelle, machine Learning etc. Les problèmes de programmation d'HackerRank utilisent une multitude de langages de programmation, y compris Java, C++, Python, JavaScript, et bien d'autres, pour résoudre les défis proposés.

Du côté de l'utilisateur, lorsqu'il soumet une solution à un challenge, sa proposition est notée sur la base du temps mis pour à trouver cette solution mais aussi sur la justesse de la proposition [31]. Les utilisateurs intègrent ainsi le classement mondial de HackerRank et gagnent des badges liés à leurs travaux, ce qui permet de nourrir la compétition entre eux. En plus des programmations individuelles, HackerRank accueille aussi des concours (généralement appelés

« CodeSprints » par le site) au cours desquels les utilisateurs s'affrontent sur les mêmes challenges et dans un temps imparti. Ils sont classés à l'issue du concours. Le côté utilisateur de leur site web est gratuit pour les codeurs [31].

Du côté des entreprises, HackerRank offre des outils puissants pour évaluer les compétences des candidats en programmation de manière objective et efficace. Les entreprises peuvent créer des tests personnalisés adaptés à leurs besoins spécifiques et utiliser les fonctionnalités de la plateforme pour évaluer les performances des candidats de manière automatisée. HackerRank apparaît comme un leader dans son domaine à l'heure de la montée de la gamification dans un univers de la programmation toujours plus compétitif [31].

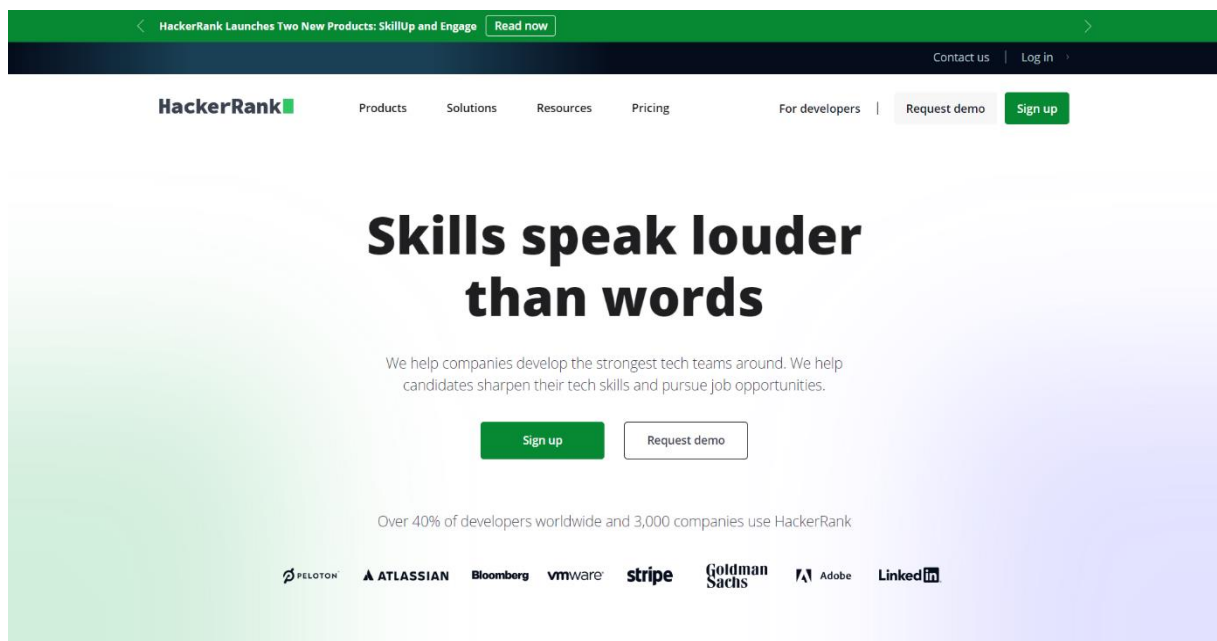


Figure 2. 1 : Page d'accueil de HackerRank [30]

La Figure 2.1 présente la page d'accueil de la plateforme HackerRank, offrant un accès à une multitude de fonctionnalités pour améliorer les compétences en programmation et relever des défis excitants. En examinant cette page, vous trouverez le bouton "Se connecter", permettant aux utilisateurs existants de se connecter facilement à leur compte HackerRank. Pour les nouveaux venus, le bouton "S'inscrire" leur offre la possibilité de créer rapidement un compte et de rejoindre la communauté de développeurs. Cette interface offre également un menu déroulant "Produits" où les utilisateurs peuvent découvrir les solutions d'entreprise proposées par HackerRank pour le recrutement et l'évaluation des compétences. De plus, le menu déroulant "Ressources" propose une multitude de contenus, tels que des articles de blog, des webinaires, des guides de recrutement et des rapports sur les compétences des développeurs.

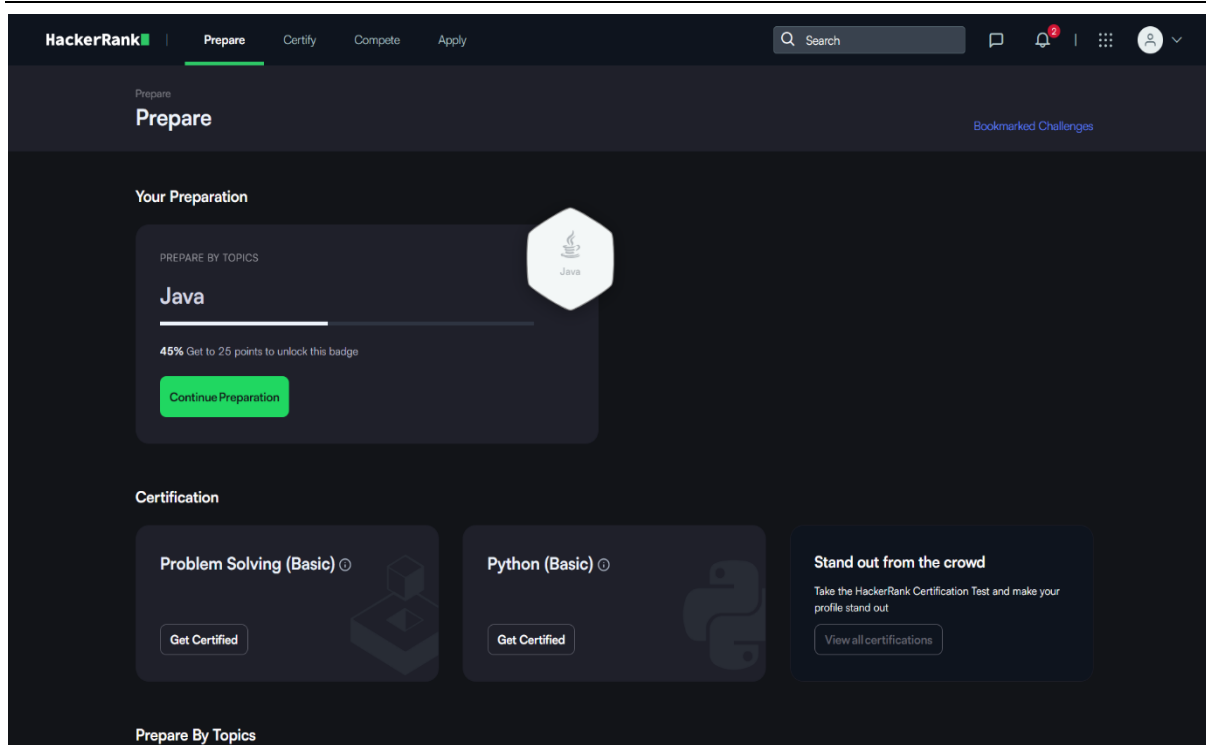


Figure 2. 2 : Dashboard de HackerRank [30]

Sur la figure 2.2, on peut observer la progression des résolutions de problèmes proposés par HackerRank pour le langage Java. Ensuite, il est possible d'accéder à une variété de problèmes à résoudre, adaptés à leur choix de langage.

2.2.2 Sphere Online Judge (SPOJ)

SPOJ (Sphere Online Judge) [32] est un système de jugement en ligne avec plus d'un million d'utilisateurs inscrits et plus de 20 000 problèmes. Les tâches sont préparées par sa communauté de contributeurs de problèmes ou sont extraites de concours de programmation précédents. SPOJ permet aux utilisateurs avancés d'organiser des concours selon leurs propres règles et comprend également un forum où les programmeurs peuvent discuter de la résolution d'un problème particulier.

En plus de la langue anglaise, SPOJ propose également son contenu dans les langues polonaise, portugaise et vietnamienne. La solution aux problèmes peut être soumise dans plus de 40 langages de programmation, y compris des langages ésotériques, via le moteur Sphere. Il est géré par la société polonaise Sphere Research Labs.

Le site Web est considéré à la fois comme un évaluateur automatisé des programmes soumis par les utilisateurs et comme une plateforme d'apprentissage en ligne pour aider les gens à comprendre et à résoudre des tâches computationnelles. Il permet également aux étudiants de comparer des paradigmes et des approches avec une grande variété de langues (Assembleur, C, C++, Fortran, PHP, Prolog...). SPOJ Permettre aux jeunes et aux débutants en programmation

de développer une compréhension des algorithmes, offrir aux étudiants des universités l'opportunité de faire leurs devoirs de manière honnête, approfondie et sans tricher, permettre aux amateurs de programmation fonctionnelle ou orientée objet de résoudre des problèmes de concours dans leur langage de prédilection.

SPOJ est une plateforme polyvalente qui sert à la fois de juge en ligne pour évaluer les compétences en programmation et de ressource d'apprentissage pour les programmeurs de tous niveaux. Avec sa grande bibliothèque de problèmes et son support multilingue, SPOJ est une ressource précieuse pour la communauté des programmeurs.

Topic	Users	R	V	A
How to get LCA accepted in Java?	B S	2	18	14h
Problem with submitting MINSTACK	S N	5	27	3d
Problem with stucking submission	V S	7	55	3d
Getting WA on AKBAR - Akbar , The great. Please help me	C S	4	80	9d
Allow viewing others solutions after solving problem	V A H R	8	3852	14d
Bitmap wa	F S	3	97	31d
How to approach problem VZLA2019K (Koala Fan)? https://www.spoj.com/problems/VZLA2019K/	S	2	123	34d
Pls help me understand the question	M S	2	118	38d
WA in Eko	N	6	351	39d

Figure 2. 3 : Page d'accueil de SPOJ [32]

Dans la figure 2.3, nous avons un aperçu de la page d'accueil de SPOJ, à partir de laquelle les visiteurs peuvent généralement trouver un accès facile à diverses fonctionnalités, telles que la création d'un compte utilisateur pour accéder à des problèmes de programmation, des concours en ligne, des tutoriels et des discussions communautaires. De plus, ceux disposant déjà d'un compte utilisateur peuvent facilement se connecter pour accéder à leurs informations.

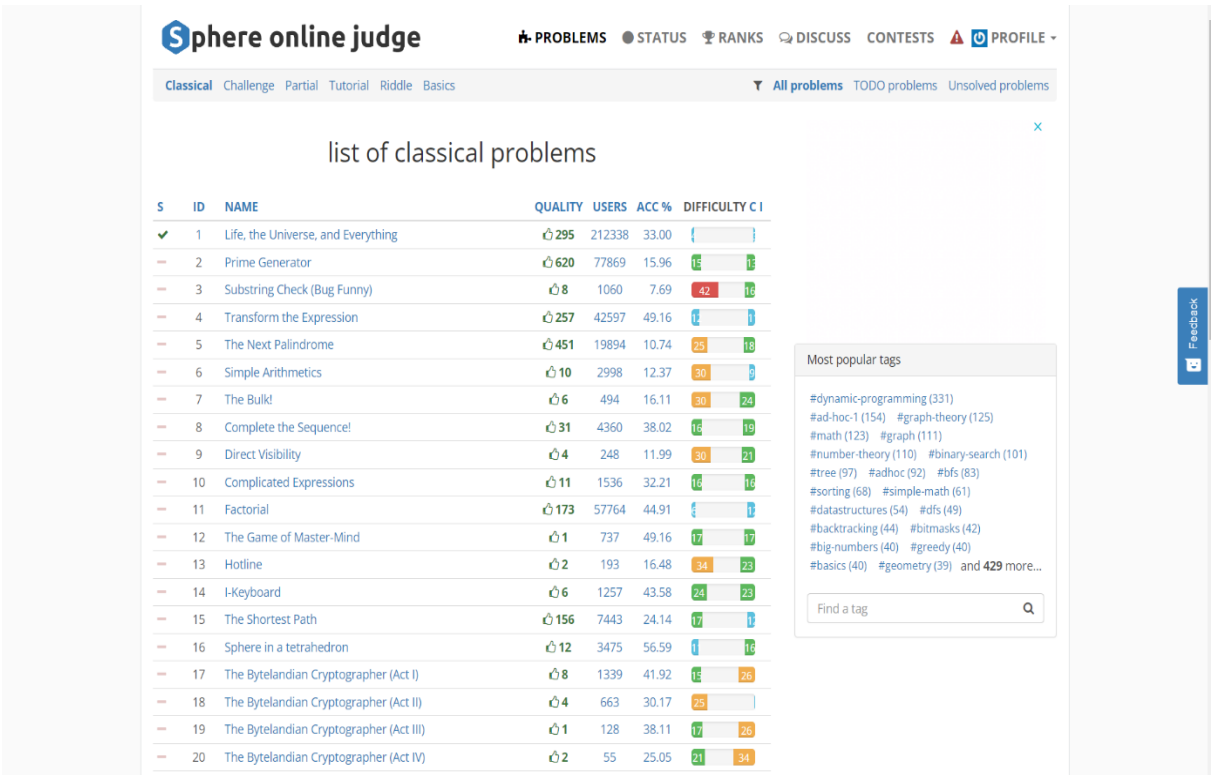


Figure 2. 4 : Liste des problèmes classiques à résoudre sur SPOJ [32]

La figure 2.4 présente la liste des problèmes classiques à résoudre sur SPOJ. Cette liste offre aux utilisateurs une sélection variée de défis de programmation, allant des problèmes de niveau débutant à ceux plus avancés.

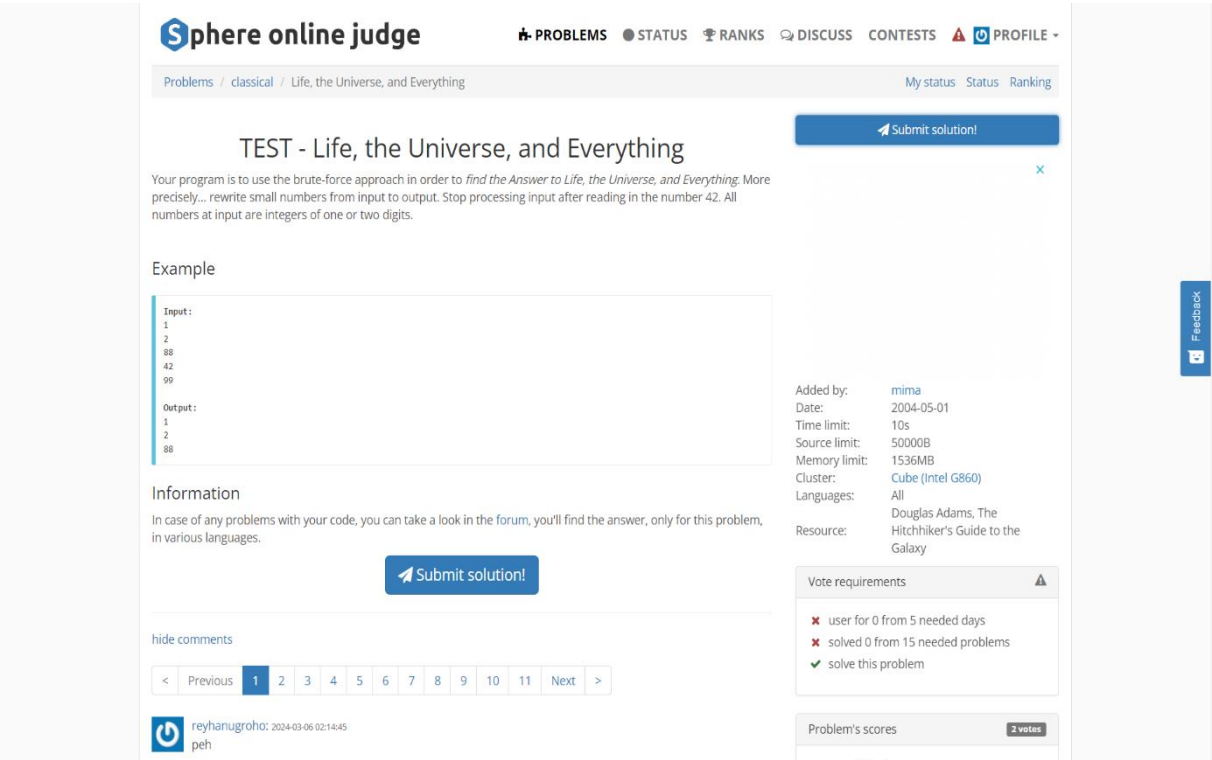


Figure 2. 5 : La spécification du problème à résoudre [32]

Dans la Figure 2.5, nous avons un aperçu de la spécification du problème à résoudre. Cette spécification détaillée pour chaque entrée la sortie attendues, avec des exemples illustratifs.

The screenshot shows the 'judge status' page of the Sphère online judge. It features a navigation bar with links for PROBLEMS, STATUS, RANKS, DISCUSS, CONTESTS, and a PROFILE dropdown. Below the navigation bar is a 'judge status' title and a pagination control showing page 1 of 5. The main content is a table with columns: ID, DATE, USER, PROBLEM, RESULT, TIME, MEM, and LANG. The table lists 18 submissions with various statuses like 'wrong answer', 'accepted', 'compilation error', and 'runtime error'. The table is styled with orange and green rows to indicate different result types.

ID	DATE	USER	PROBLEM	RESULT	TIME	MEM	LANG
32875390	2024-04-05 06:42:41	Cliff	Update the array I	wrong answer	0.18	5.2M	CPP
32875389	2024-04-05 06:42:34	wtf	Word Rings	wrong answer	0.03	5.8M	CPP14
32875386	2024-04-05 06:41:59	vi4deee11	Catapult that ball	wrong answer	0.08	5.2M	CPP14-CLANG
32875384	2024-04-05 06:41:28	Aryan	Catapult that ball	wrong answer	0.05	8.3M	CPP14
32875382	2024-04-05 06:41:04	Aryan	Catapult that ball	compilation error	-	-	ADA95
32875380	2024-04-05 06:40:27	ISMAILA SIDIBE	Life, the Universe, and Everything	accepted	0.01	5.2M	ADA95
32875377	2024-04-05 06:40:12	vi4deee11	Catapult that ball	wrong answer	0.05	5.2M	CPP14-CLANG
32875375	2024-04-05 06:40:00	shreya	Sum 2 digits	wrong answer	0.15	57M	JAVA
32875370	2024-04-05 06:38:20	ISMAILA SIDIBE	Life, the Universe, and Everything	compilation error	-	-	ADA95
32875368	2024-04-05 06:37:53	elkesha	Life, the Universe, and Everything	runtime error (NZEC)	0.17	25M	PYTHON3
32875352	2024-04-05 06:34:48	Cliff	Update the array I	time limit exceeded	-	5.1M	CPP
32875326	2024-04-05 06:29:20	ce978517	Enormous Input Test	compilation error	-	-	ASM64
32875323	2024-04-05 06:29:20	shreya	Sum of two numbers	accepted	0.36	59M	JAVA

Figure 2. 6 : Statuts de la compilation du problème [32]

Dans la figure 2.6, nous visualisons le résultat de la soumission pour connaître si le code soumis par l'utilisateur a été compilé avec succès et s'il réponde aux spécifications énoncées, ou s'il contient des erreurs.

2.2.3 HackerEarth

HackerEarth [33] est une entreprise de logiciels basée à San Francisco qui propose une variété de solutions axées sur le recrutement et le développement des compétences techniques. Fondée en novembre 2012 par Sachin Gupta et Vivek Prakash, tous deux anciens élèves de l'Indian Institute of Technology Roorkee, HackerEarth a commencé comme une plateforme sociale interactive appelée MyCareerStack, visant à aider les étudiants à se préparer aux entretiens techniques.

Par la suite, l'entreprise a évolué vers le développement d'un logiciel d'évaluation technique automatisé, permettant aux organisations de tester les compétences des développeurs à travers des tests de codage. Cette plateforme, appelée HackerEarth Assessments, est certifiée ISO et propose une évaluation basée sur les compétences ainsi que des analyses pour aider les entreprises à embaucher des développeurs qualifiés.

En plus des évaluations techniques, HackerEarth propose également des solutions de recrutement à distance, permettant aux entreprises de mener des entretiens vidéo et d'embaucher des développeurs à distance.

HackerEarth organise également des événements tels que des hackathons, des défis de programmation et des concours de codage, offrant ainsi une plateforme pour les développeurs de mettre en pratique leurs compétences et de se mesurer à d'autres professionnels de l'industrie. En outre, HackerEarth propose une plateforme de pratique qui offre des tutoriels de programmation et des problèmes pratiques couvrant divers sujets tels que les structures de données, les algorithmes, les mathématiques, Python et l'apprentissage automatique. Cette plateforme vise à aider les développeurs à améliorer leurs compétences et à se préparer à des défis techniques.

Enfin, HackerEarth gère un programme d'ambassadeurs étudiants, offrant aux étudiants l'opportunité de créer et de gérer des clubs de programmation dans leurs universités respectives.

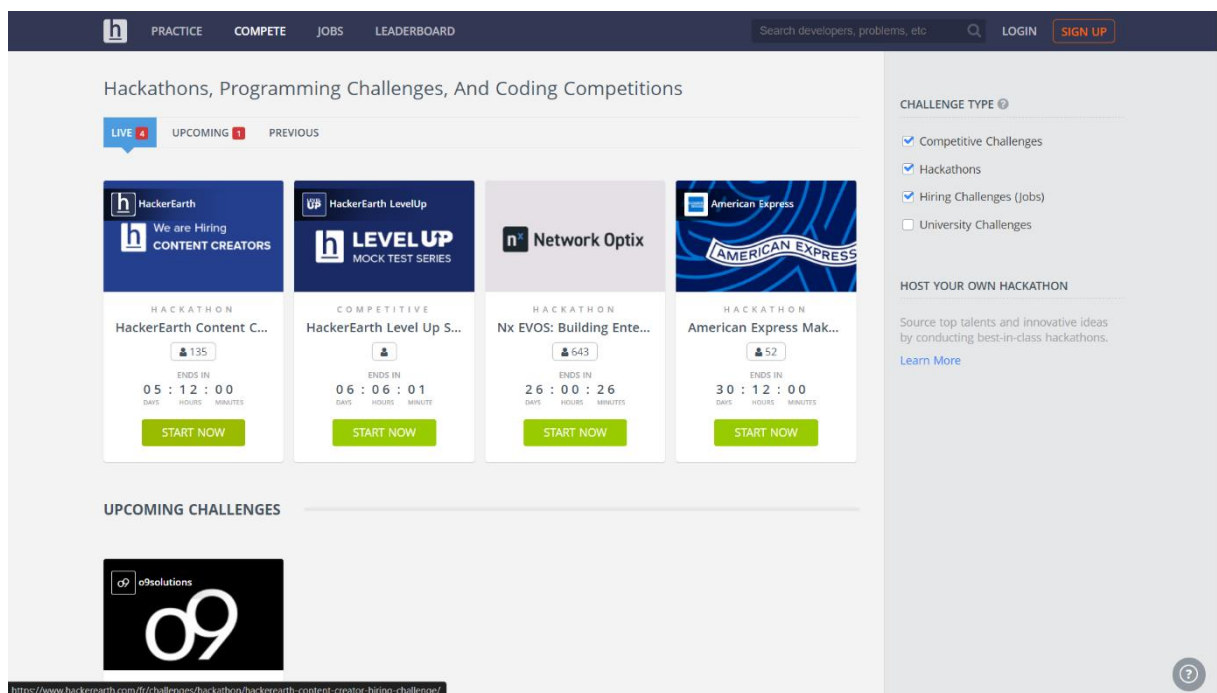


Figure 2. 7 : Page d'accueil de HackerEarth [33]

La page d'accueil de hackerhEarth est présentée sur la Figure 2.7, les utilisateurs ont accès à une gamme de hackathons et ont la possibilité de se connecter ou de créer un compte utilisateur.

The screenshot shows the HackerEarth interface for the 'Basics of Input/Output' track. On the left, a sidebar displays the user's progress (20 points, level 5) and a list of categories: Input/Output, Complexity Analysis, Implementation, Operators, Bit Manipulation, and Recursion. The main content area lists 10 problems, each with its name, attempted count, success rate, and level. The problems are: Favourite Singer (Level: Easy), Maximum borders (Level: Easy), Number of steps (Level: Medium), Zoos (Level: Easy), Divisibility (Level: Easy), Split houses (Level: Easy), Cost of balloons (Level: Easy), Seven-Segment Display (Level: Easy), and Ali and Helping innocent people (Level: Easy). Each problem has a 'Solve Now' button.

Figure 2. 8 : Liste des problèmes basiques sur HackerEarth [33]

Sur la Figure 2.8, Nous trouvons la liste des problèmes basiques a resoudre sur hackerEarth.

The screenshot shows the details of the 'Factorial!!' problem on HackerEarth. The problem description states: 'You have been given a positive integer N . You need to find and print the Factorial of this number. The Factorial of a positive integer N refers to the product of all number in the range from 1 to N . You can read more about the factorial of a number [here](#).' The input format is a single integer N , and the output format is a single line denoting the factorial of the number N . The constraints are $1 \leq N \leq 10$. A sample input of 2 yields a sample output of 2. On the right, a code editor shows a Python 3.8 solution for calculating the factorial of N .

```

1 def factorial(N):
2     result = 1
3     for i in range(1, N + 1):
4         result *= i
5     return result
6
7 # Read input from user
8 N = int(input())
9
10 # Calculate factorial and print the result
11 print(factorial(N))
12

```

Figure 2. 9 : Spécifications et soumissions du problème [33]

Dans la Figure 2.9, nous voyons la spécification détaillée d'un problème à résoudre, présentée aux utilisateurs avec un éditeur de code intégré. Les exigences du problème, y compris les entrées et les sorties attendues, sont clairement décrites, avec des exemples pour illustrer les attentes. Les utilisateurs peuvent développer leur solution directement dans l'éditeur de code ou en uploadant les fichiers solutions et la soumettre ensuite pour évaluation.

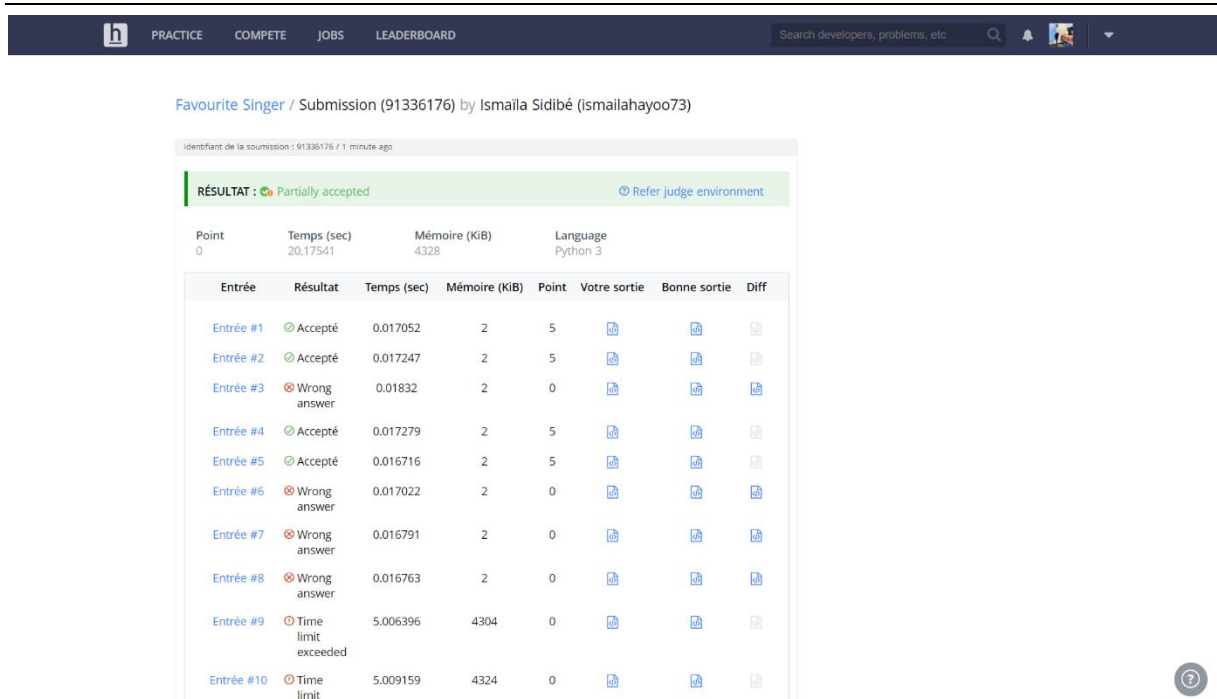


Figure 2.10 : Résultats de la soumission [33]

Dans la Figure 2.10, nous visualisons le résultat du fichier soumis pour le problème sélectionné parmi les problèmes de base. Ce résultat peut inclure divers éléments tels que le score obtenu, les cas de test réussis ou échoués, ainsi que des informations supplémentaires sur la performance de la solution. Cette présentation permet aux utilisateurs de comprendre rapidement l'efficacité de leur solution et éventuellement d'apporter des modifications ou des améliorations si nécessaire.

2.2.4 CoderByte

CoderByte [34] est une plateforme en ligne similaire à HackerEarth, offrant une variété de défis de programmation interactifs pour améliorer vos compétences en codage. Avec des centaines de défis disponibles dans différents langages comme Java, Python, C++, et bien d'autres, chacun adapté à divers niveaux de difficulté, du débutant à l'expert. Un éditeur code intégré permet de tester et de soumettre facilement votre code, avec des commentaires et des solutions disponibles pour chaque défi, facilitant ainsi l'apprentissage par la pratique. De plus, un système de suivi des progrès est disponible pour mesurer votre performance au fil du temps.

En outre, une section dédiée à la préparation aux entretiens techniques propose des exercices similaires à ceux posés lors d'entretiens réels, ce qui est idéal pour se préparer aux défis du monde professionnel de la technologie.

Des défis hebdomadaires encouragent les programmeurs de tous niveaux à mettre régulièrement en pratique leurs compétences et à relever de nouveaux défis.

Une communauté active permet aux utilisateurs de partager leurs solutions, de poser des questions et de discuter de problèmes de programmation avec d'autres membres, favorisant ainsi un environnement d'apprentissage collaboratif. Enfin, des cours en ligne sont disponibles pour approfondir vos connaissances dans des domaines spécifiques de la programmation, couvrant une gamme de sujets allant des structures de données aux frameworks populaires.

En résumé, CoderByte offre une expérience complète et enrichissante pour les programmeurs de tous niveaux, combinant des défis de codage, des outils pratiques, une communauté dynamique et des ressources pédagogiques pour favoriser un développement continu des compétences en programmation.

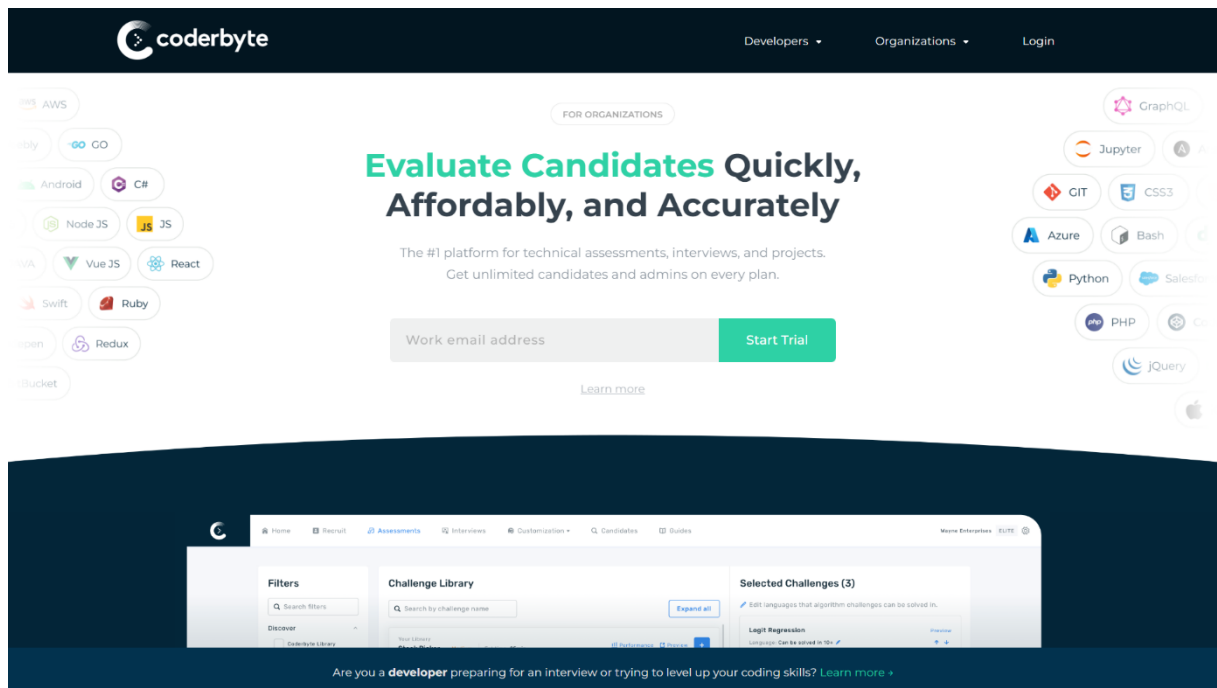


Figure 2. 11 : Page d'accueil de CoderByte [34]

Dans la Figure 2.11, nous avons un aperçu de la page d'accueil de Coderbyte. Cette interface offre plusieurs fonctionnalités permet aux visiteurs de s'inscrire ou de se connecter à un compte existant, facilitant ainsi l'accès aux différents problèmes à résoudre disponibles sur la plateforme. Ainsi que des différents menus en haut de la page pour pouvoir s'inscrire avec le login ou se connecter.

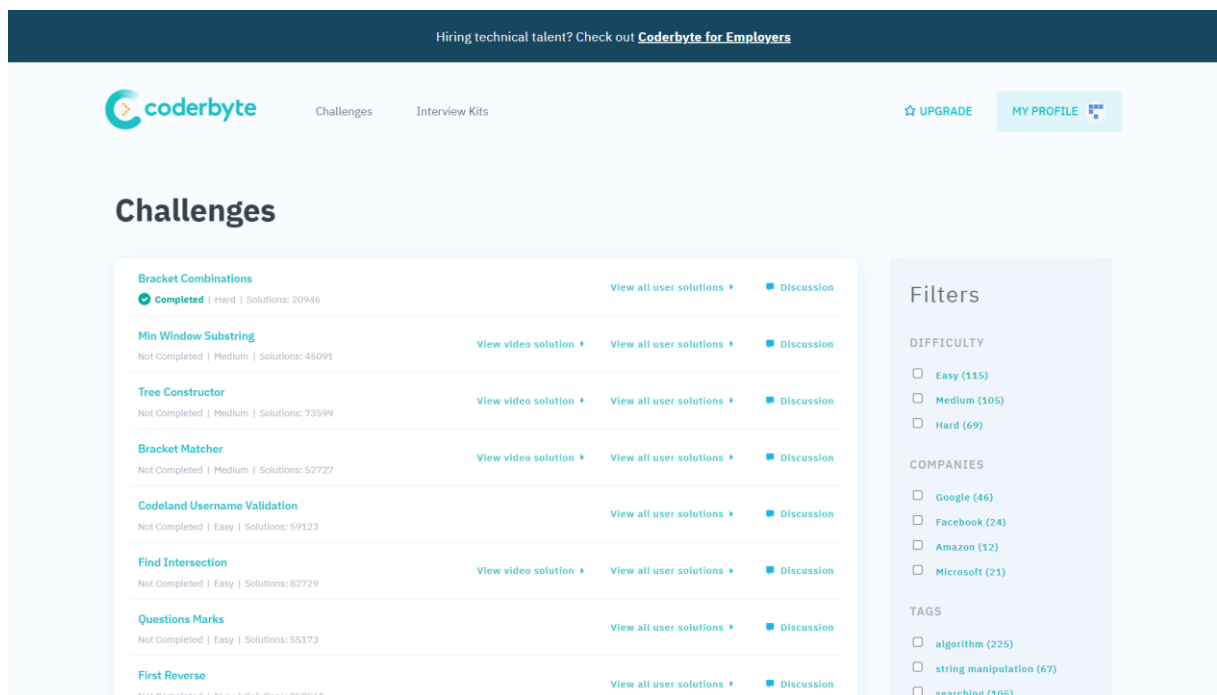


Figure 2. 13 : Liste des problèmes sur CoderByte [34]

La Figure 2.12 présente une liste de problèmes sur la plateforme, offrant une variété de choix avec des niveaux de difficulté indiqués. Les utilisateurs peuvent filtrer les problèmes par entreprise, difficulté ou tags, facilitant ainsi la recherche de problèmes spécifiques selon leurs intérêts ou leurs objectifs.

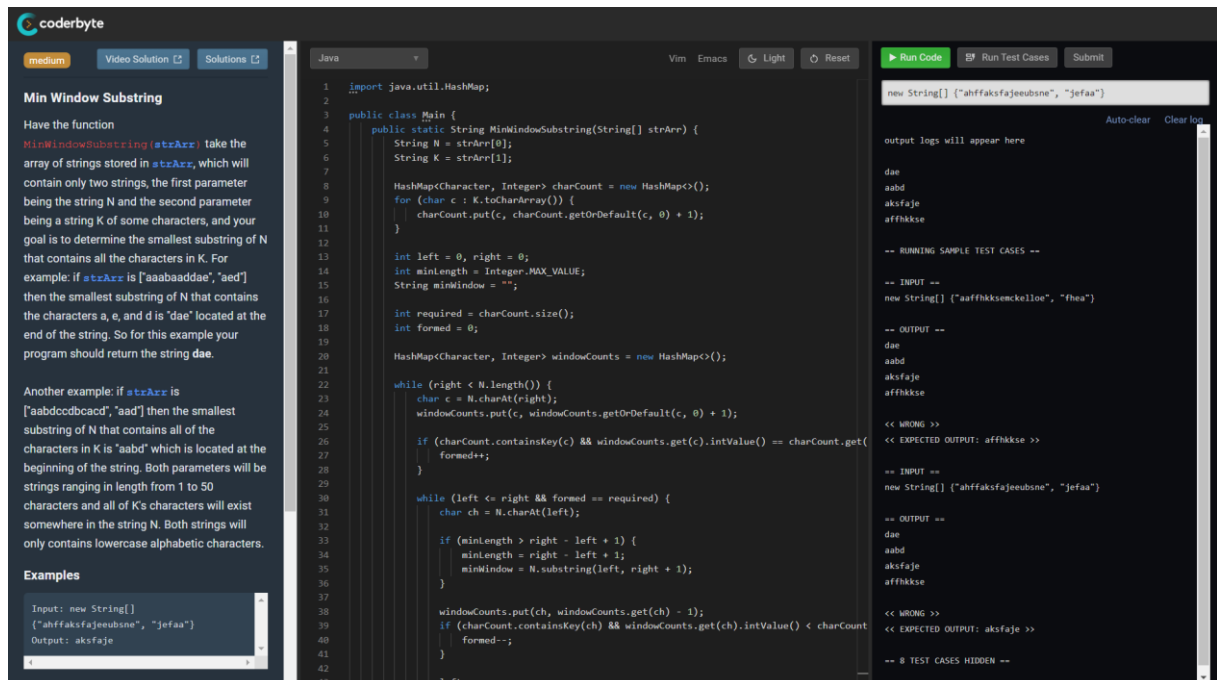


Figure 2. 12 : Spécifications du problème et éditeur de code [34]

Dans la Figure 2.13, une disposition tripartite est présentée pour aider les utilisateurs à résoudre les problèmes de manière interactive. À gauche, la spécification du problème est fournie avec des exemples illustratifs pour clarifier les exigences. Au milieu, un éditeur de code est

disponible pour que les utilisateurs écrivent leur solution. À droite, les utilisateurs peuvent tester leur solution avec des cas de test fournis. Une fois validée, ils peuvent la soumettre pour obtenir leur score.

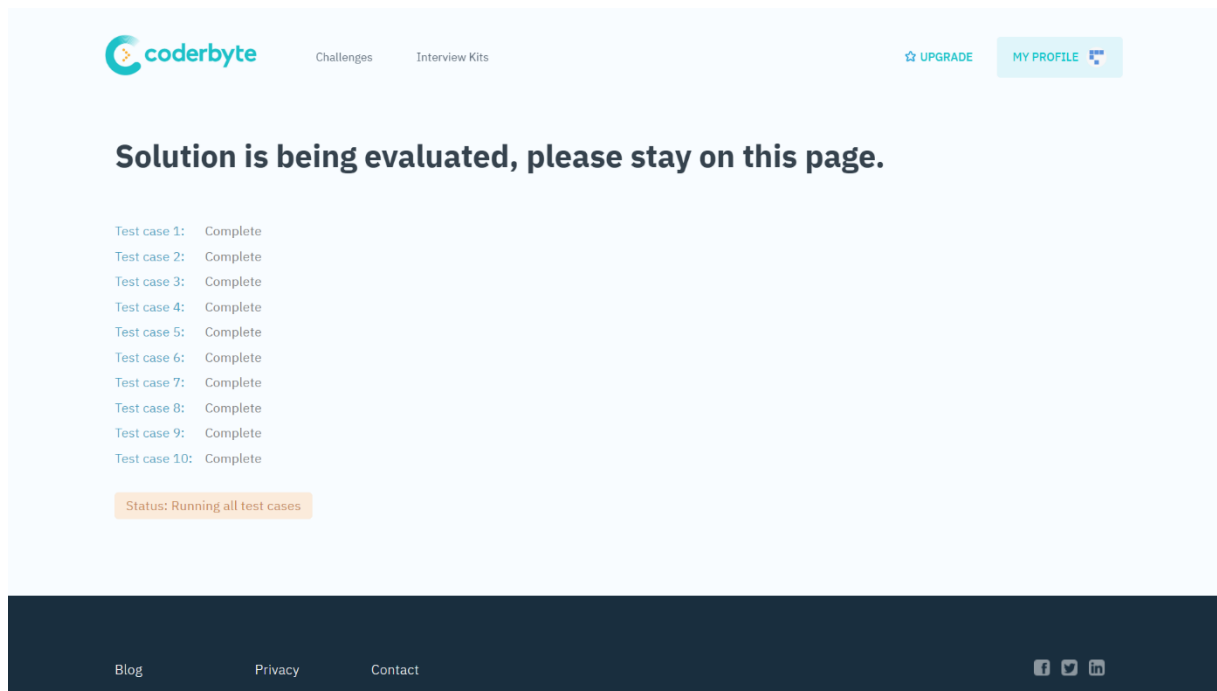


Figure 2. 14 : Déroulement du cas de test [34]

Sur la Figure 2.14, nous observons l'exécution complète de l'ensemble des cas de test associés à un problème donné, pour lequel l'utilisateur a soumis sa solution. Cette visualisation permet de suivre pas à pas le processus de test, montrant si la solution passe ou échoue chaque cas de test individuel, du premier au dernier. Cela offre aux utilisateurs une compréhension claire de la performance de leur solution et de leur score associé.

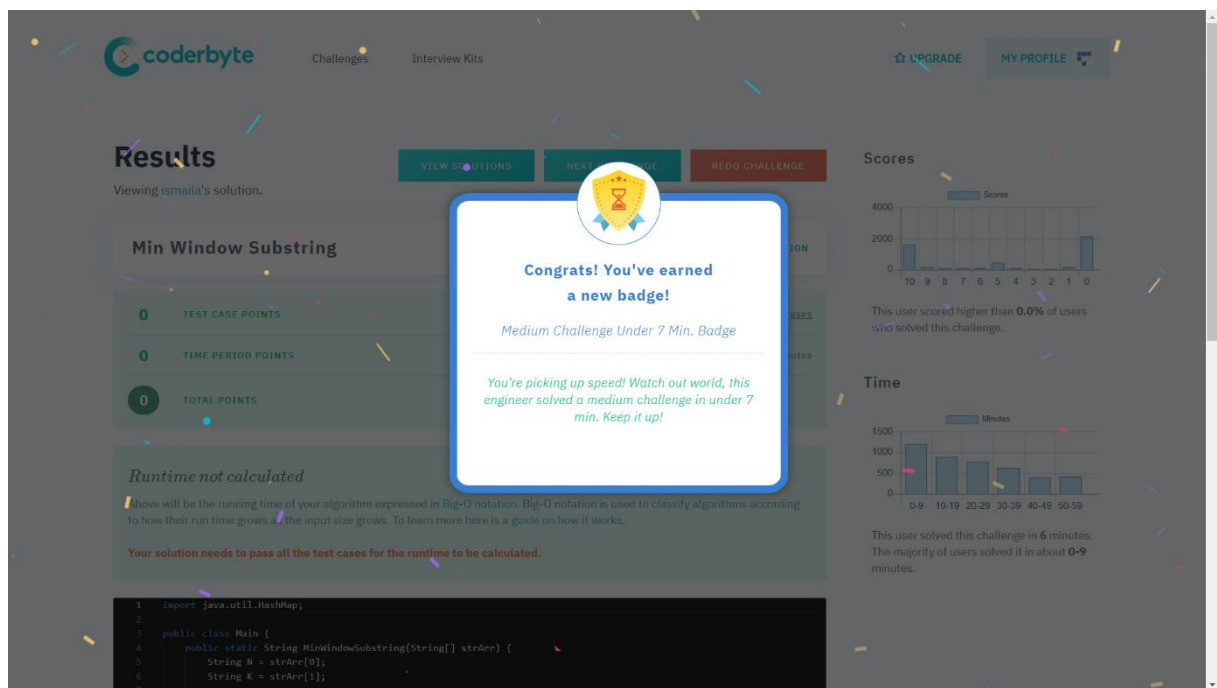


Figure 2. 15 : Résultats selon les cas de test valide [34]

Sur la Figure 2.15, nous découvrons le résultat final obtenu après la soumission de la solution à un problème sélectionné, en tenant compte de la validation des cas de test. Cette visualisation révèle non seulement si la solution a passé ou échoué les cas de test, mais aussi le temps nécessaire à l'exécution du programme. De plus, elle indique le nombre de cas de test non validés. Enfin, cette visualisation attribue à l'utilisateur une note, représentant une évaluation globale de sa performance dans la résolution du problème.

2.3 Etudes comparative des travaux similaires

Dans cette Partie, nous effectuerons une analyse comparative approfondie des plateformes de test de programme informatique mentionnées précédemment (Tableau 2.1) et des mémoires de recherche ou article existants ayant traité des thématiques similaires (Tableau 2.2).

2.3.1 Etude comparative des plateformes

Cette étude compare les quatre plateformes de test de programme informatique populaires cités dans la section *présentations de quelques plateformes* selon les critères d'évaluation :

- **Téléchargements des spécifications des problèmes :**

La plateforme Permet-elle aux utilisateurs de télécharger les spécifications des problèmes sous forme de fichier PDF ou dans un autre format, afin de les consulter hors ligne et de résoudre les problèmes sans avoir besoin d'une connexion Internet continue ?

- **Soumission de Fichiers de solution :**

La plateforme permet-elle de soumettre des solutions en uploadant des fichiers de solution (code) ?

- **Utilisations des jeux de tests :**

La plateforme Utilise-t-elle une variété de jeux de tests pour évaluer les solutions soumises par les utilisateurs ?

- **Délai de soumission :**

Existe-t-il un délai de soumissions pour les problèmes énoncés ?

Les résultats sont présentés sous forme de tableau comparatif, où « Oui » indique que la plateforme répond au critère et « Non » qu'elle ne répond pas au critère.

Critère	HackerRank [30]	Sphere Online Judge [32]	HackerEarth [33]	CoderByte [34]
Téléchargements des spécifications des problèmes	NON	NON	NON	NON
Soumission de fichiers de solution	OUI	OUI	OUI	NON
Utilisation des jeux de test	OUI	OUI	OUI	OUI
Délais de soumissions	Uniquement pour les concours	Uniquement pour les concours	Uniquement pour les concours	Uniquement pour les concours

Tableau 2. 1 : Etude comparative des différentes plateformes existantes

Le tableau 2.1 offre une synthèse des fonctionnalités des plateformes de correction de code en ligne, en mettant l'accent sur des critères clés. Chaque ligne du tableau correspond à un critère d'évaluation spécifique, tandis que les colonnes représentent les différentes plateformes examinées.

Cette comparaison met en évidence les fonctionnalités de chaque plateforme concernant le téléchargement des spécifications des problèmes, la soumission de fichiers de solution, l'utilisation des jeux de tests, ainsi que les délais de soumission aux problèmes. Il est intéressant de noter que malgré les différences notables entre ces plateformes, elles partagent également

des similitudes. Ces points communs découlent des critères d'évaluation choisis, qui mettent en évidence les aspects fondamentaux pour une expérience utilisateur satisfaisante dans le domaine de la programmation et de l'apprentissage.

Bien que les plateformes examinées ne permettent pas le téléchargement des spécifications des problèmes pour une consultation hors ligne, elles offrent toutes la possibilité de soumettre des fichiers de solution, à l'exception de CoderByte. De plus, chacune de ces plateformes intègre l'utilisation de jeux de tests dans leur processus d'évaluation, garantissant ainsi une évaluation précise des solutions soumises. Toutefois, aucune de ces plateformes n'autorise de délai pour la soumission de solutions aux problèmes standard, sauf dans le cadre des compétitions de programmation.

2.3.2 Etude comparative des travaux dans la littérature

Afin de mener une analyse approfondie des travaux existants, nous comparerons ces travaux qui sont les travaux réalisés en Master, et certaine recherche laboratoire, article en fonction de plusieurs critères, tels que la nature des projets, la méthodologie de test utilisée, les résultats obtenus et le contexte dans lequel les travaux ont été réalisés. Cette analyse nous permettra d'identifier les opportunités d'innovation et de différenciation pour notre solution, tout en nous aidant à mieux comprendre les défis et les enjeux du domaine dans lequel nous intervenons.

Organisons ces comparaisons dans un tableau afin de visualiser clairement les différentes approches des travaux :

Critères	Recherche et sélection d'un outil pour les TP de compilation [35]	QueryInspectors [36]	DMOJ [37]	Plate-forme d'entraînement à la programmation [38]	Implémentation d'un LMS d'auto évaluation des devoirs de programmation [39]
Nature des Projets	Etude et sélection du meilleur outil existant pour les TPs de Compilation	Création d'une plateforme en ligne pour l'évaluation automatique des TPs de compilation	Réalisation d'une plateforme en ligne pour l'évaluation des programmes des tps des étudiants	Conception d'une plate-forme en ligne pour l'apprentissage et évaluation automatique de la programmation	Plate-forme web d'auto-évaluations des devoirs des apprenants
Méthodes et outils utilisés	Le générateur d'analyseur javacc avec le processus jjtree	Analyseur lexicale (Flex) et Analyseur syntaxique (GNU Bison)	Standard stdin et stdout d'Unix, Données binaires, VirtualBox	IA, AES (Abstract Execution sequence), embeddings [40]	Techniques d'évaluation automatique des codes en utilisant des jeux de test
Contexte	Les TPs de compilations	Théorie des langages	La programmation compétitive pour le support des travaux pratiques de programmation	Initiation à l'algorithmique et à la programmation	Enseignement à distance dans le domaine de l'informatique
Résultats	Obtention d'un meilleur outil existant pour l'évaluation des TPs de compilation	Validation automatique des programmes en langages C	Validation automatique de solution des tps des étudiants	Validation automatique des programmes écrits en langages Python	Evaluation automatique des devoirs de programmation écrits en langage java

Tableau 2. 2 Tableau comparative des projets similaires

Le Tableau 2.2 offre un aperçu détaillé des critères utilisés pour évaluer ces travaux, de la nature des projets réalisés, des méthodes employées, des contextes dans lesquels ils s'inscrivent, ainsi que des résultats obtenus. Cette analyse permettra aux lecteurs de mieux comprendre les différences et similitudes entre les diverses approches présentées, ainsi que l'impact de ces travaux dans le domaine de l'enseignement de la programmation.

Chaque colonne du tableau représente un travail spécifique entrepris par des individus ou des groupes dans le domaine de vérification et évaluation automatique du code source. Chaque travail est caractérisé par une combinaison de critères, qui déterminent les éléments utilisés pour évaluer les différents travaux présentés, de nature de projet qui décrit le type de projet réalisé dans chaque travail, Les méthodes utilisées détaillent les techniques et approches des fois, voir des moyens comme javacc, Flex... qui ont été employés pour mener à bien les projets, le contexte d'application les domaines dans lesquels les travaux ont été réalisés. Enfin, les Résultats obtenus résume les principales réalisations de chaque projet, soulignant les contributions spécifiques apportées à l'avancement de la vérification et de l'évaluation automatique du code.

Dans le Tableau 2.2, une variété de travaux est répertoriée, tous visant à améliorer l'évaluation et l'apprentissage dans le domaine de la programmation, mais avec des emphases différentes. Certains se contentent de sélectionner un meilleur outil, tandis que d'autres se focalisent sur le développement d'applications les utilisant. Pour l'évaluation automatique des programmes sources des langages C, java et Python, avec des méthodologies et des résultats spécifiques à chaque projet.

2.4 Conclusion

Les études comparatives nous permettent d'identifier les solutions existantes. Grâce à cette démarche, nous serons en mesure de définir le positionnement de notre travail par rapport aux travaux existants. Cela nous offre une perspective précieuse pour situer notre projet dans le paysage de la recherche et de l'innovation, en mettant en évidence ses forces distinctives et son potentiel d'apport de valeur ajoutée par rapport aux autre travaux similaires.

Chapitre III : Conception et Spécifications fonctionnelles

3.1 Introduction

La modélisation de l'architecture de notre plateforme Test Coding nous permet d'identifier en profondeur les différents acteurs (utilisateurs), fonctionnalités, propriétés et contraintes de notre système. Cette étape constitue le fondement sur lequel reposera la mise en œuvre réussie de notre plateforme. Pour parvenir à cet objectif, nous adoptons une approche systématique, en utilisant les principaux concepts de la modélisation logicielle. Cette approche nous permet d'avoir une vision claire et précise de l'architecture de notre plateforme, en définissant les interactions entre les utilisateurs et le système, ainsi que sa structure interne. Plus spécifiquement, nous utilisons divers outils de modélisation, tels que les diagrammes de cas d'utilisation, séquence, classe et d'objet. Les diagrammes de cas d'utilisation nous aident à visualiser de manière détaillée les actions des utilisateurs sur notre plateforme, tandis que les diagrammes de séquence décrivent le déroulement séquentiel des interactions entre les éléments du système. Enfin, les diagrammes de classe nous permettent de modéliser la structure statique de notre système, en identifiant les différentes classes d'objets, leurs attributs et leurs relations. Pour enrichir la modélisation, nous intégrons les diagrammes d'objet pour mieux représenter la structure statique de notre système.

3.2 Spécifications fonctionnelles

3.2.1 Analyse des besoins

La plateforme doit être capable de répondre aux exigences suivantes :

✓ **Gestions des travaux pratiques :**

La plateforme doit permettre à l'acteur enseignant d'ajouter un nouveau travail pratique et à l'acteur étudiant de télécharger le travail pratique disponible sur la plateforme.

✓ **Soumission des solutions :**

Les étudiants doivent pouvoir soumettre leur(s) solution(s) des travaux pratiques via la plateforme.

✓ **Evaluation des solutions soumises :**

La plateforme doit évaluer automatiquement les solutions soumises par les étudiants sans intervention de la part de l'enseignant, en leur attribuant une note en fonction du nombre de cas de test réussis.

✓ **Gestion des utilisateurs :**

La plateforme doit gérer les comptes des étudiants et des enseignants.

3.2.2 Acteurs

Nous distinguons pour notre système trois (3) types d'acteurs qui sont :

- **Enseignant :**

Ajout de travail pratique, spécifications des cas de test et les fichiers attendus pour chaque travail pratique, visionner la liste des étudiants inscrit ainsi que leur note obtenue à chaque travail pratique.

- **Étudiant :**

Télécharger TP, Soumet ses solutions aux travaux pratiques et consulte les résultats du TP soumis.

- **Admin :**

Quant à nous, nous assumons le rôle de l'administrateur, chargé de la gestion globale du système.

3.2.3 Exigences fonctionnelles

- ✓ **Gestion des travaux pratiques**

- Un enseignant peut créer un nouveau travail pratique en spécifiant la date du début, la date de fin, les fichiers cas de test ainsi que les fichiers attendus. Il peut aussi visionner la liste des étudiants inscrits.
- Un étudiant peut télécharger les travaux pratiques disponibles dont le délai n'est pas dépassé.
- La plateforme affiche la liste des travaux pratiques disponibles, avec leur date de début, leur date de fin et le numéro du TP ainsi le Button télécharger TP pour le délais n'est pas dépassé.

- ✓ **Soumission des solutions**

- L'étudiant peut soumettre un fichier contenant sa solution pour un travail pratique spécifique.
- La plateforme accepte les fichiers qu'au format (.java)

- ✓ **Évaluation des solutions**

- La plateforme évalue automatiquement les solutions soumises par les étudiants en fonction des cas de test et les fichiers attendus définis par l'enseignant.
- Pour chaque cas de test, la plateforme utilise les données d'entrée spécifiées pour comparer la sortie produite par la solution de l'étudiant avec la sortie attendue.

- ✓ **Gestion des utilisateurs**

- La plateforme permet la connexion des enseignants et des étudiants.

-
- Chaque utilisateur dispose d'un compte unique identifié par son email et son mot de passe.

3.2.4 Exigences non fonctionnelles

✓ **Performance**

- La plateforme doit être capable de gérer un nombre important d'utilisateurs et de travail pratique.

✓ **Sécurité**

- La plateforme doit être sécurisée contre les accès non autorisés et les modifications frauduleuses des données.
- Les mots de passe des utilisateurs doivent être stockés de manière sécurisée (c'est-à-dire hachage de mot de passe).

✓ **Ergonomie**

- L'interface de la plateforme doit être intuitive et facile à utiliser pour tous les utilisateurs du site web.

3.3 Conception du système

3.3.1 Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation décrivent les fonctionnalités du système du point de vue de l'utilisateur. Ils montrent comment les différents types d'acteurs (utilisateur) interagissent avec le système et quelles actions ils peuvent effectuer.

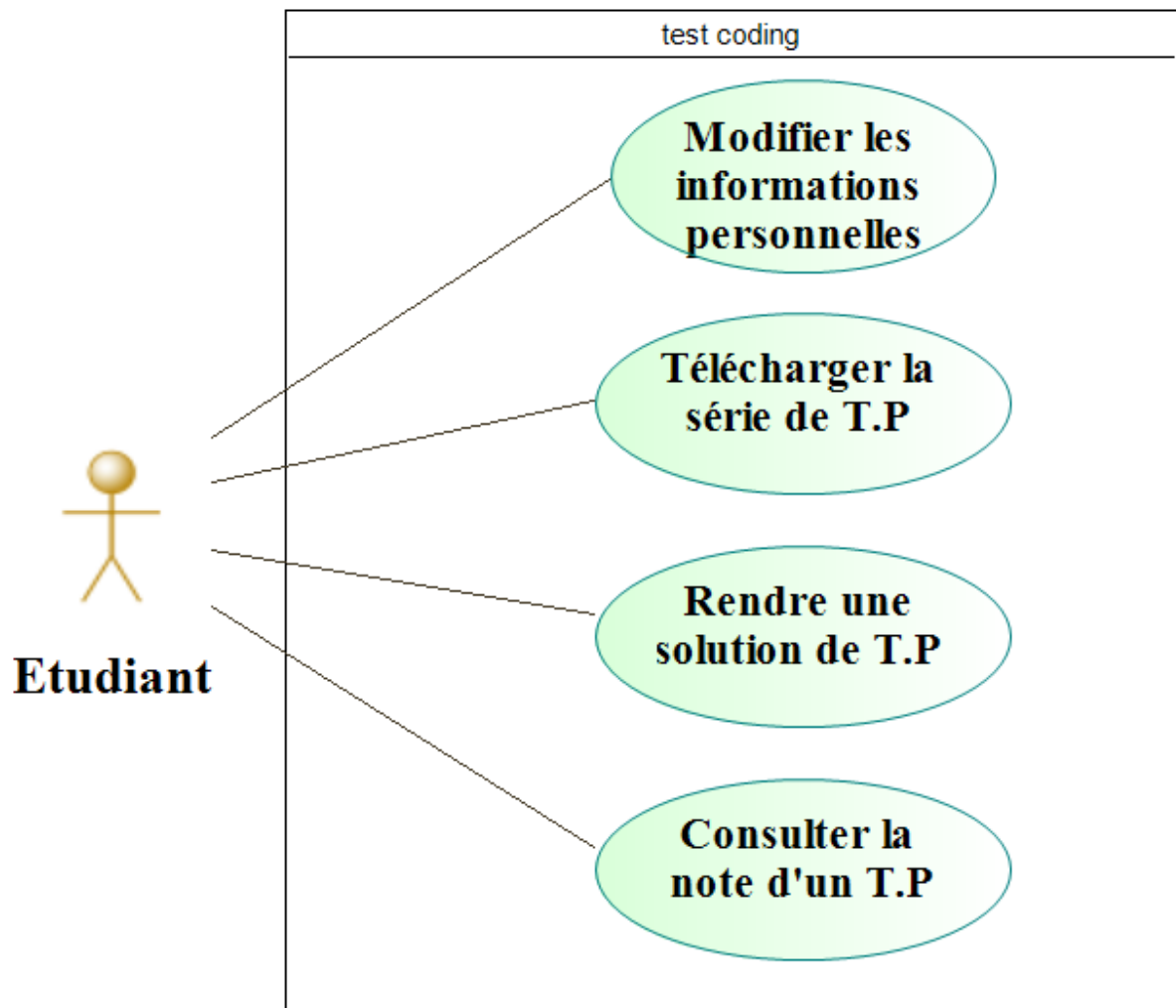


Figure 3. 1 : Diagramme de cas d'utilisation pour l'acteur « Etudiant »

Dans la Figure 3.1, l'acteur étudiant peut effectuer des actions telles que Modifier ses informations personnelles, télécharger la série de TP disponible, rendre une solution de T.P et enfin consulter la note obtenue a un Travail Pratique rendus.

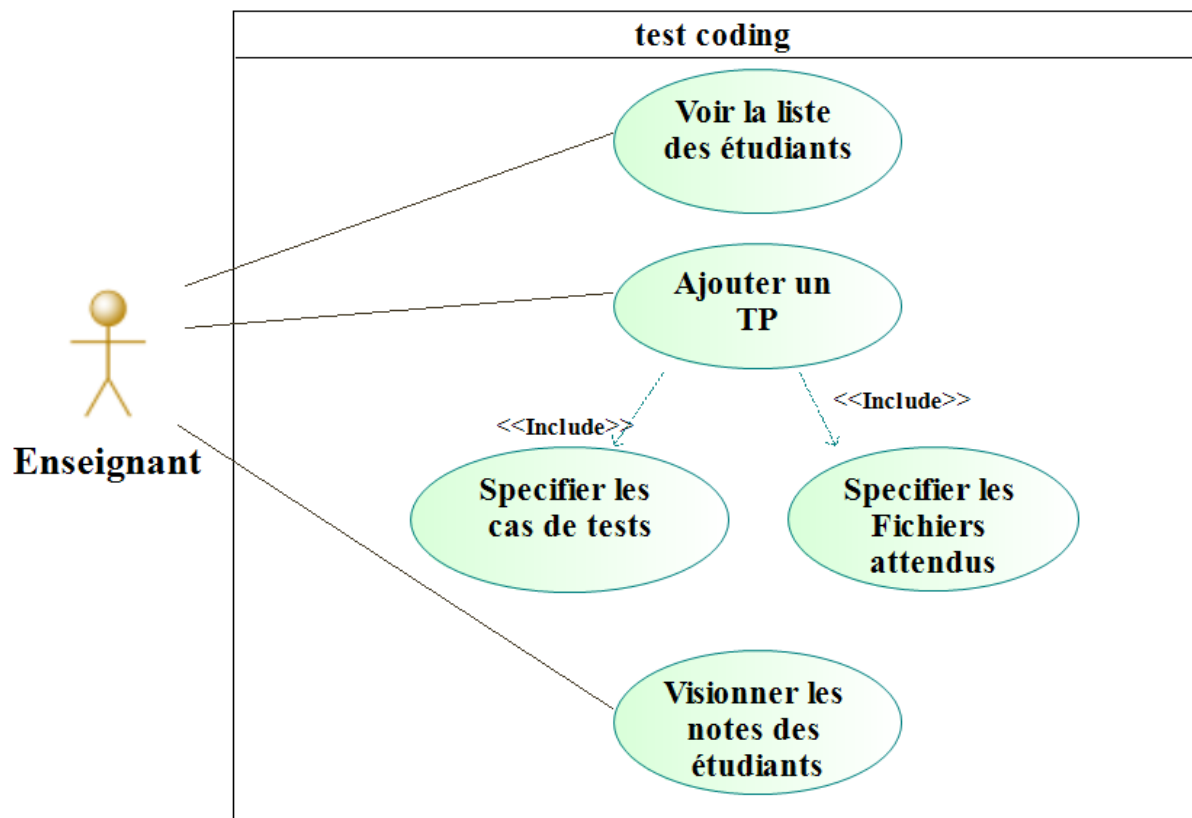


Figure 3. 2 : Diagramme de cas d'utilisation pour l'acteur « Enseignant »

La Figure 3.2 explique comment l'acteur enseignant peut effectuer des actions telles que l'ajout de nouveau T.P, visionner la liste des étudiants inscrits ainsi que leurs notes obtenues aux solutions du travail pratique soumis.

Les diagrammes de cas d'utilisation ont permis de décrire les fonctionnalités offertes à chaque utilisateur. Ils constituent une représentation visuelle claire et concise des interactions entre les utilisateurs et le système.

3.3.1.1 Description textuelle des principaux cas d'utilisation

Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation [41]. Il est recommandé de rédiger une description textuelle du cas d'utilisation nécessaire pour comprendre la séquence d'action exécuté dans le cas d'utilisation de bout en bout. C'est une forme informelle pour la compréhension d'un cas d'utilisation, n'est pas formalisée par UML, C'est une sorte de documentation des cas d'utilisation [42]. Voir ci-dessous les descriptions dans le tableau des cas d'utilisation principale.

Nom du cas d'utilisation N°1	Ajouter un T.P
Acteur Principal	Enseignant
Descriptions	Ce cas d'utilisation permet à un enseignant d'ajouter un travail pratique (T.P) sur la plateforme. L'enseignant peut déposer la série de T.P, spécifie les cas de test, les fichiers attendus associe au T.P, les dates de début et fin.
Préconditions	L'utilisateur doit être authentifié en tant qu'enseignant.
Démarrage	L'enseignant choisit l'option "Ajouter un T.P"
Scénario nominal	1. L'enseignant sélectionne "Ajouter un TP" sur la plateforme. 2. Le système affiche un formulaire pour ajouter un T.P. 3. L'enseignant dépose la série de T.P. 4. L'enseignant spécifie les cas de tests et les noms du fichiers attendus pour le TP. 5. L'enseignant spécifie une date de début et fin pour le TP. 6. L'enseignant soumet le formulaire en cliquant sur le bouton Ajouter. 7. Le système vérifie que toutes les informations requises sont présentes et valides. 8. Le système enregistre le TP avec toutes les informations fournies dans la base de données. 9. L'enseignant voit un message de confirmation et retourne à la page des T.P ajoutés.
Scénario alternatifs	Si l'enseignant ne remplit pas l'une des cases dans le formulaire d'ajout de travail pratique à partir du point 3 du scenario nominal, Le système informe l'enseignant que les informations sont incomplètes et le scenario reprend au point 3 du scenario nominal.

Tableau 3. 1 : Description textuelle du cas d'utilisation "Ajouter un T.P"

Nom du cas d'utilisation N°2	Rendre une solution de T.P
Acteur Principal	Etudiant
Descriptions	Ce cas d'utilisation permet à un étudiant de soumettre sa solution pour un travail pratique (T.P.) via la plateforme TestCoding. L'étudiant peut télécharger les fichiers de solutions et recevoir une confirmation de soumission.
Préconditions	L'utilisateur doit être authentifié en tant qu'étudiant et avoir accès au T.P à rendre.
Démarrage	L'étudiant choisit l'option "soumettre T.P"
Scénario nominal	1. L'étudiant sélectionne "Rendre une solution de T.P." dans le système. 2. Le système affiche une liste des T.P. disponibles pour soumission. 3. L'étudiant sélectionne le T.P. à rendre. 4. Le système affiche un formulaire de soumission de solution. 5. L'étudiant télécharge le fichier de solution. 6. L'étudiant soumet le formulaire. 7. Le système enregistre la solution et affiche un message de confirmation de soumission. 8. L'étudiant sera redigé vers la page de consultation de note obtenue.
Scénario alternatifs	3.a L'étudiant décide de ne pas soumettre de solution et quitte le formulaire. 6.a formulaire incorrecte si l'étudiant ne remplit pas une des cases alors le scénario reprend au point 5

Tableau 3. 2 : Description textuelle du cas d'utilisation Rendre une solution de T.P

3.3.2 Diagrammes de séquence

Les diagrammes de séquence système permettent d'illustrer de manière chronologique les interactions entre les différents objets d'un système, entre ces objets et les acteurs externes. Ils expliquent en détail la transmission des messages entre les objets et les acteurs, et fournissent une vue temporelle de ces interactions. Grâce à ces diagrammes, il est possible de décrire la mise en œuvre des cas d'utilisation sur le système illustré par le diagramme de classe. Ils fournissent une vision interne du fonctionnement du système, en soulignant l'état du système à

un moment donné et en décrivant des scénarios spécifiques. On présente les échanges de messages de manière chronologique, illustrant la transmission des messages entre les acteurs et le système et entre les différents objets du système [43]. Nous avons identifié plusieurs cas d'utilisation clés pour lesquels nous avons créé des diagrammes de séquence système. Parmi ces cas d'utilisation, nous avons choisi de mettre en avant le processus de téléchargement et de soumission des TP pour l'acteur étudiant, ainsi que le processus d'ajout de TP pour l'acteur enseignant. Ces cas ont été sélectionnés en raison de leur importance dans le fonctionnement global de notre système (qui font partie des exigences fonctionnelles). Ces diagrammes permettent de visualiser les interactions entre les différents objets du système. Voir Les figures Ci-dessous qui présentent visuellement le déroulement des interactions entre les acteurs et le système.

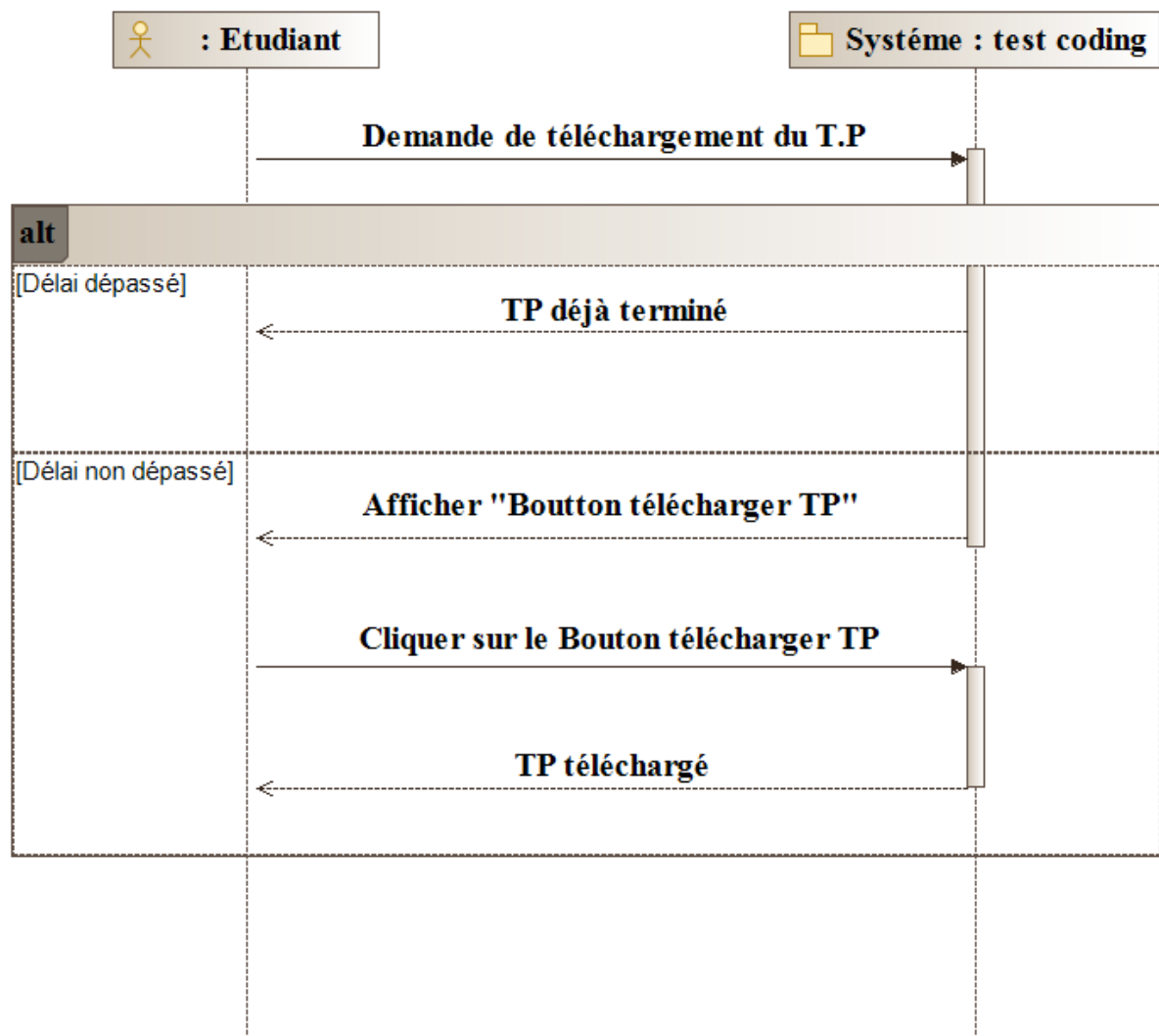


Figure 3. 3 : Diagramme de séquence télécharger TP

La Figure 3.3 décrit le processus de téléchargement d'une série de T.P par l'acteur étudiant, suivi des différents scénarios.

- L'Etudiant fait une demande au système pour télécharger la série de T.P,
- Le système vérifié si le délai de la série de T.P est dépassé ou non,
- Dans le cas où le délai est dépassé le système affiche un message d'erreur indiquant que le TP est déjà terminé,
- Dans le cas contraire Le système affiche le bouton "Télécharger la série de T.P" pour permettre à l'étudiant de télécharger le T.P,
- L'étudiant clique sur le bouton pour lancer le téléchargement du TP,
- Le système confirme le téléchargement et affiche un message de réussite.

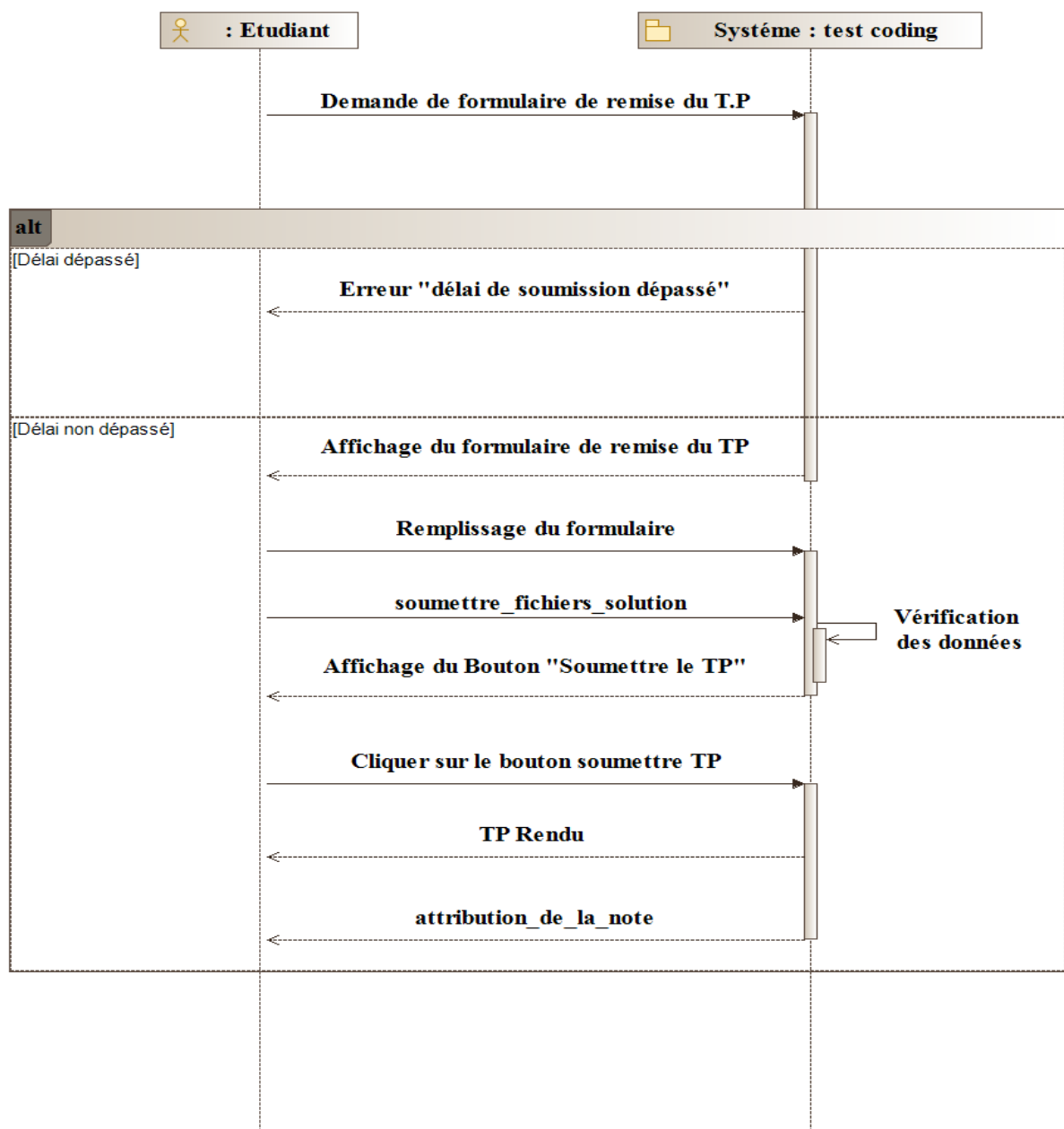


Figure 3. 4 : Diagramme de séquence rendre TP

La figure 3.4 décrit les étapes qu'un étudiant doit suivre pour qu'il soumet un TP avec des scénarios variés détaillés comme suite :

- L'étudiant fait une demande pour soumettre le TP,
- Le système fait une vérification sur la date de la soumission ,
- Si la date limite est dépassé le système affiche un message d'erreur indiquant que le délais de soumission du T.P est dépassé et que le TP ne peu plus être rendu,
- Sinon le système affiche un formulaire de remise de TP,
- L'étudiant remplit le formulaire avec les informations du rendu : ensemble des fichiers de solution du T.P.
- L'étudiant clique sur le bouton Soumettre la solution T.P ,
- Le système fait une vérification si les fichiers rendus correspondre au fichier attendus et qu'ils requièrent toutes les conditions.
- Après vérification, le système affiche un message de confirmation du TP rendu.
- Et lancer le processus de correction du fichier jusqu'à l'attribution de la note en fonction du cas de test passés.

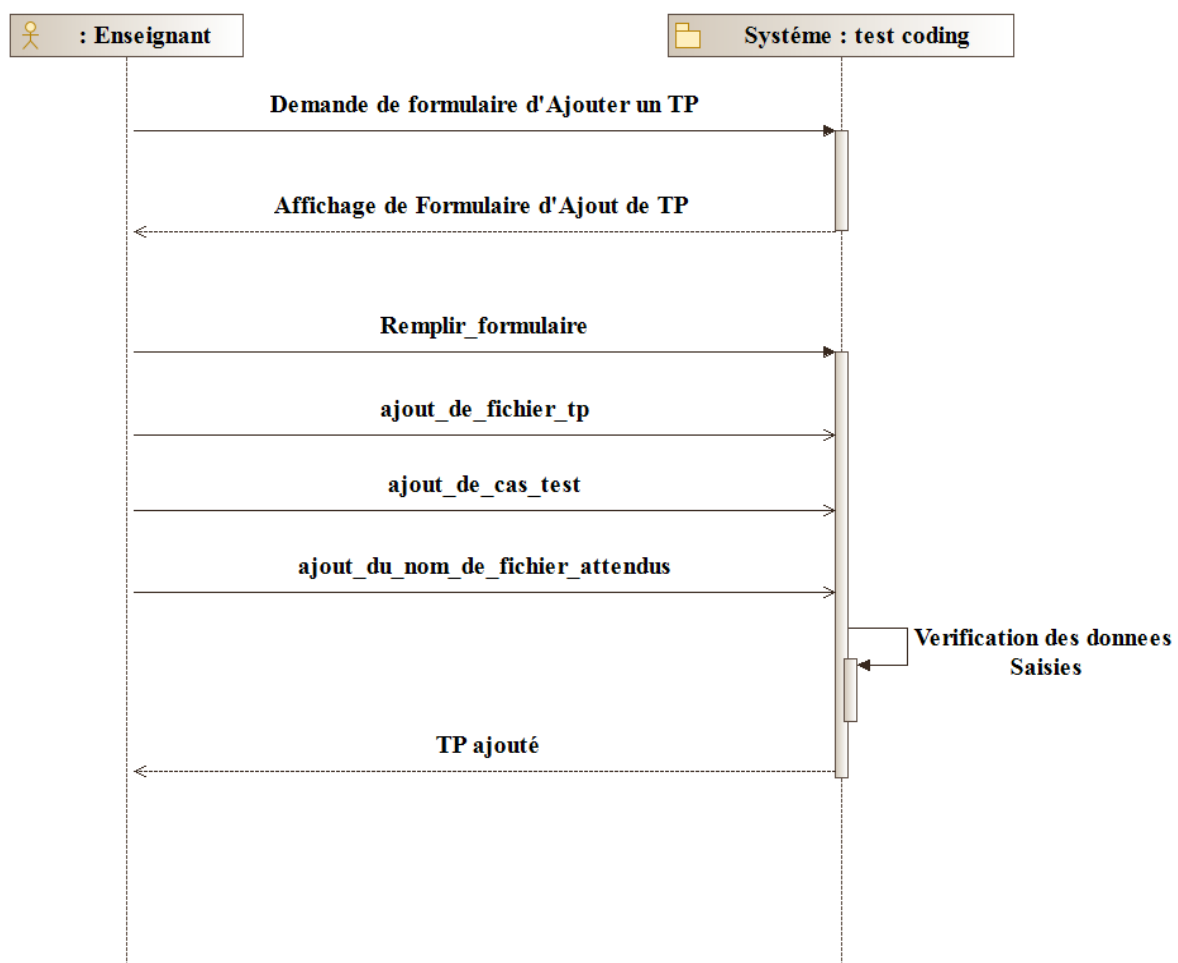


Figure 3. 5 : Diagramme de séquence d'Ajout TP

Dans la Figure 3.5, nous observons comment un enseignant procède à l'ajout d'un TP,

- L'enseignant fait une demande d'ajout de nouveau TP,
- Le système affiche le formulaire d'ajout de TP ,
- L'enseignant remplit le formulaire avec les informations du TP :
 - Date de debut et fin ,
 - La série de TP , Les fichier attendus pour le TP, et les cas de tests associé au TP
 - et le renvoyer au système
- Le système vérifie les données saisies ,
- Si les données sont valides le système enregistre le TP et affiche un message de confirmation d'ajout du T.P,
- Si les données sont invalide le système affiche un message en indiquant les champs erronés

Les diagrammes de séquence ont permis de visualiser le déroulement séquentiel des interactions entre les différents éléments du système pour des cas d'utilisation clés.

3.3.3 Diagramme de classes

Les diagrammes de classes sont une représentation visuelle d'un modèle de classe [44], chaque classe est composée d'attributs, de méthodes et de relations entre les différentes classes, clarifiant ainsi les responsabilités de chaque objet du système [44].

Nous avons identifié plusieurs classes fondamentales, notamment la classe Users et la classe système (Test Coding). Ces classes sont dotées d'attributs spécifiques.

Les diagrammes de classes jouent un rôle crucial dans la conception d'un système bien organisé et modulaire. Ils garantissent que chaque classe a un rôle clair et bien défini, facilitant ainsi le développement, la maintenance et l'extension du système dans le temps. Voir la Figure 3.6.

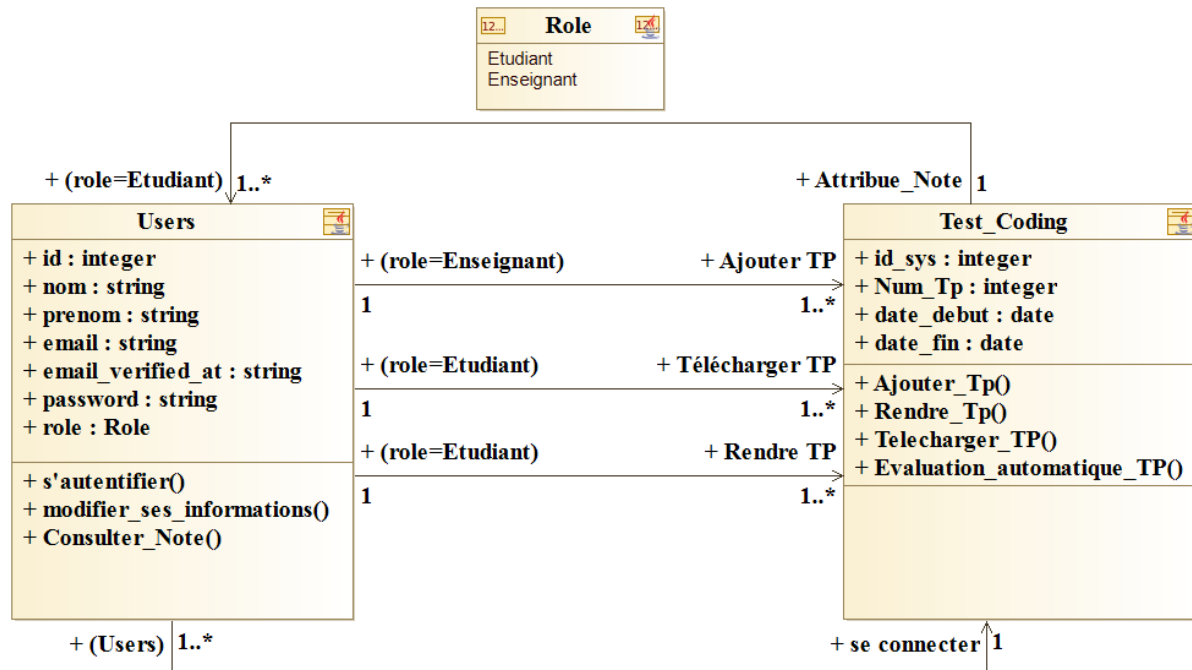


Figure 3. 6 : Diagramme de classes système

Sur la Figure 3.6, nous avons un aperçu du diagramme de classes générales de notre système. Il met en évidence deux classes principales : la classe "Users" et la classe "TestCoding". Chacune de ces classes possède des attributs et des méthodes spécifiques. Dans la classe "Users", nous trouvons un attribut nommé "role", associé à un type "Role", représentant une classe. Cette classe "Role" peut prendre deux valeurs possibles : "Etudiant" ou "Enseignant". Cette distinction permet de différencier les utilisateurs en fonction de leur statut dans le système. Les utilisateurs connectés en tant qu'enseignants ont la possibilité d'ajouter des travaux pratiques sur la plateforme "TestCoding" et peuvent également se connecter. Quant aux utilisateurs connectés en tant qu'étudiants, ils peuvent télécharger des séries de travaux pratiques depuis la plateforme, se connecter et soumettre leurs solutions aux travaux pratiques. Les diagrammes de classe ont permis de définir la structure statique du système en identifiant les classes d'objets, leurs attributs et leurs relations. Ils constituent une base solide pour la mise en œuvre du système et garantissent sa cohérence et sa modularité.

3.3.4 Diagrammes de Séquence Objet

La différence entre le diagramme de séquence système et objet réside dans le fait que le séquence objet offre une vue interne du système, permettant de visualiser le déroulement des procédures et des fonctions, et il est utilisé principalement dans le cadre de la conception [45]. Les diagrammes de séquence objet sont représenté sur les figures ci-dessous.

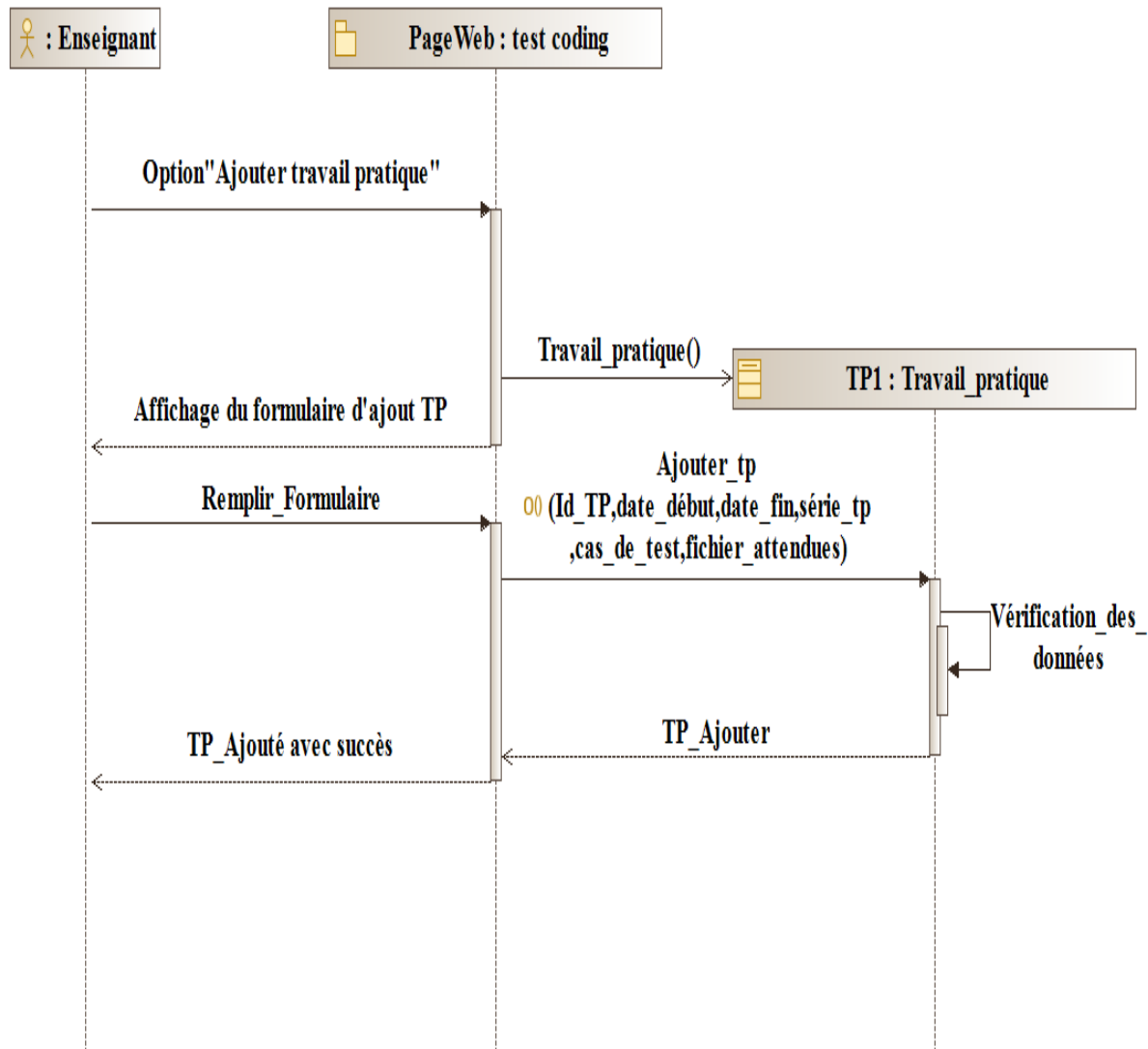


Figure 3. 7 : Diagramme de séquence objet d'ajout du travail pratique

La Figure 3.7 présente le diagramme de séquence détaillé pour l'ajout d'un travail pratique par un enseignant sur une plateforme web. Le processus commence avec la sélection de l'option d'ajout de travail pratique par l'enseignant, suivi par l'affichage et la complétion d'un formulaire sur la page web. Un nouvel objet "Travail_pratique" TP1 est ensuite créé avec les informations fournies. La méthode appropriée est appelée pour ajouter le travail pratique, suivie par une vérification des données. Si les données sont validées, le travail pratique est ajouté avec succès au système, et une notification de réussite est envoyée à l'enseignant.

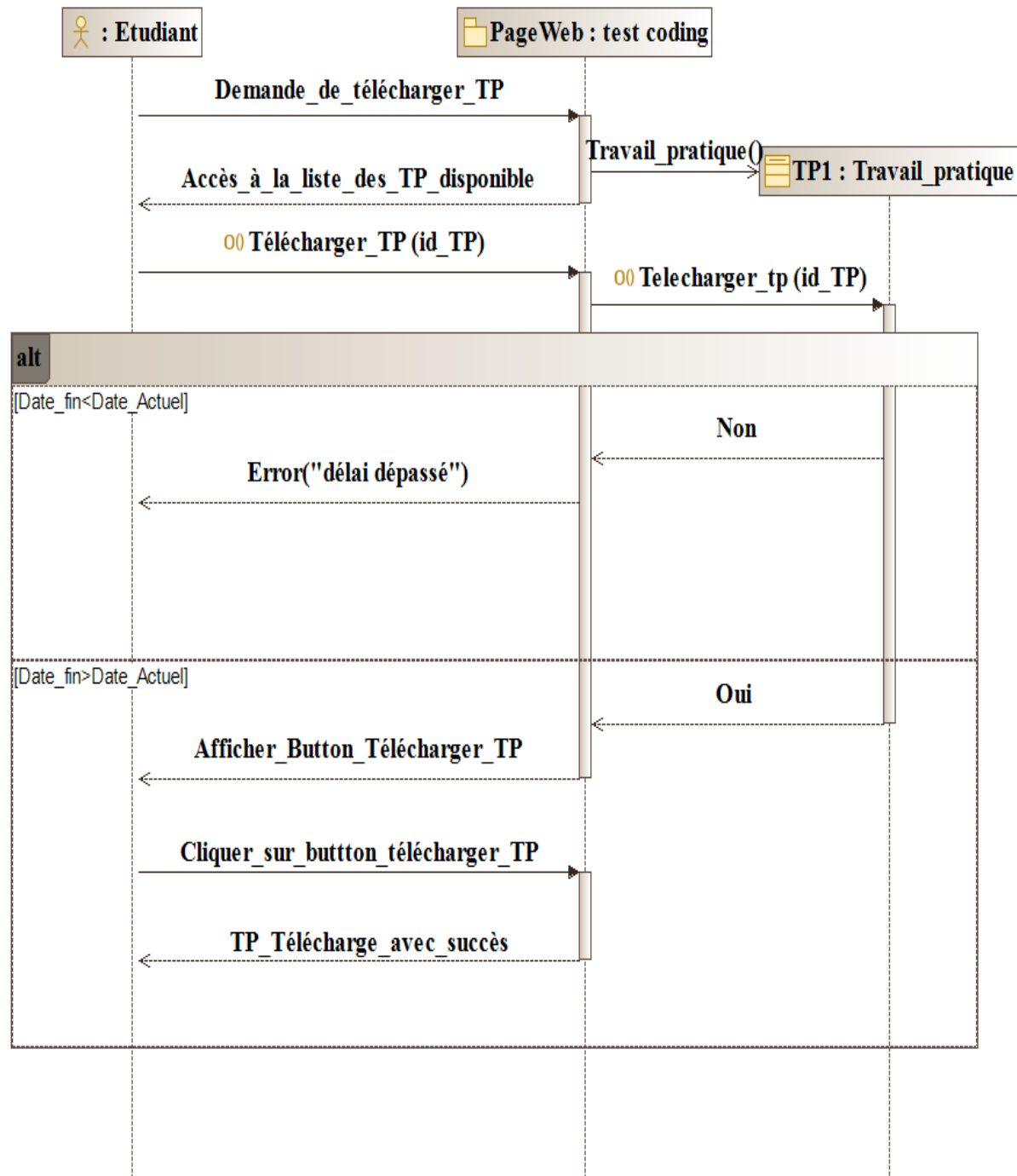


Figure 3. 8 : Diagramme de séquence objet téléchargement du travail pratique

Le diagramme de séquence dans la Figure 3.8 illustre le processus de téléchargement d'un travail pratique par un étudiant sur une plateforme web. L'étudiant initie une demande de téléchargement, et un nouvel objet "Travail_pratique" TP1 est créé. Il accède ainsi à la liste des travaux pratiques disponibles. En sélectionnant un travail pratique spécifique, la plateforme vérifie si le délai n'est pas dépassé. Si la date limite est dépassée, un message d'erreur "délai dépassé" est affiché. Sinon, le bouton de téléchargement apparaît, et l'étudiant peut cliquer dessus pour télécharger le travail pratique avec succès.

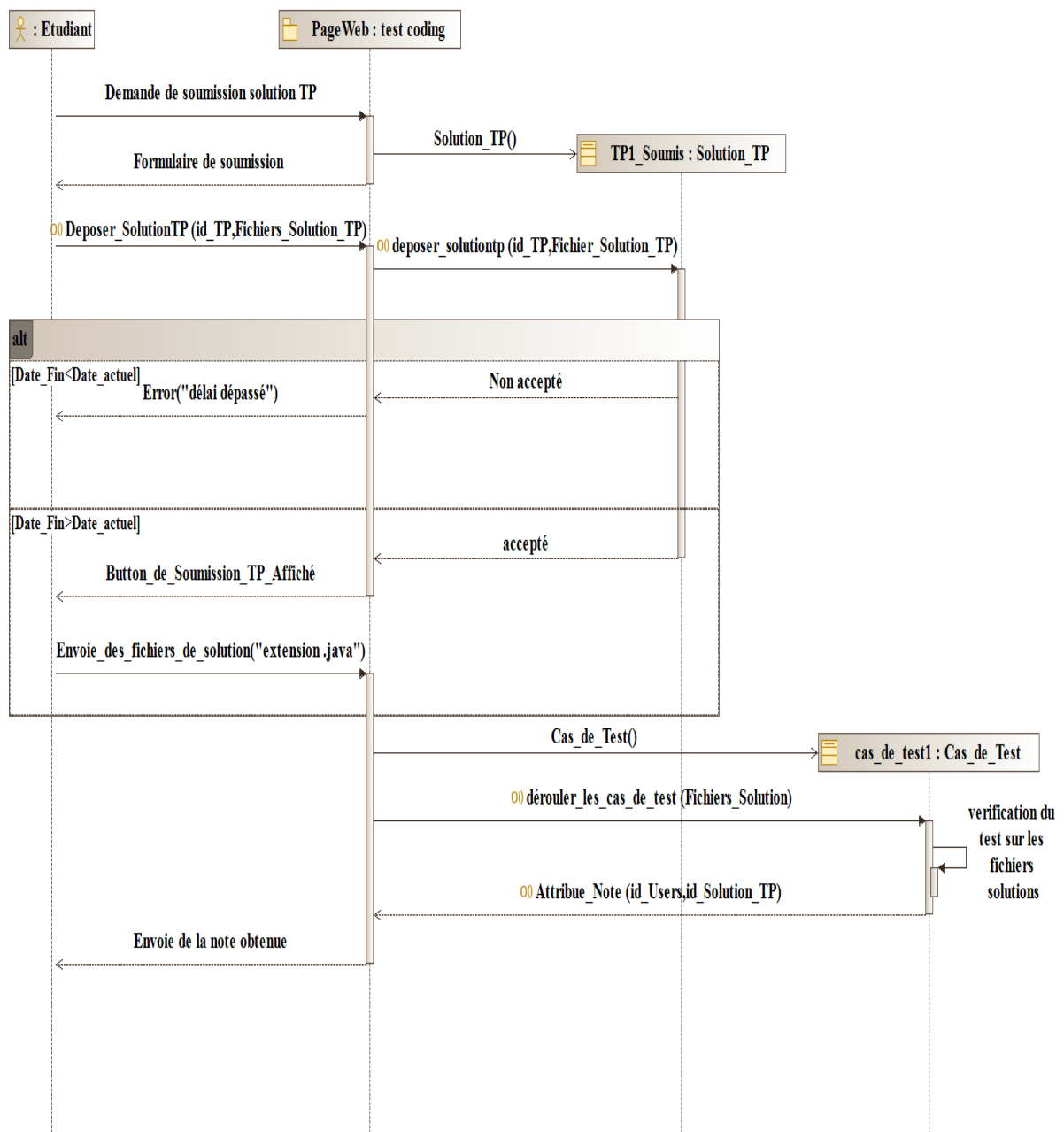


Figure 3. 9 : Diagramme de Séquence Objet Soumission et Evaluation du TP

La figure 3.9 illustre le processus de soumission et d'évaluation d'un travail pratique rendu sur la plateforme. L'étudiant accède à la page de soumission après la création de l'objet TP1_soumis de la classe Solution_TP. Il doit ensuite sélectionner le fichier contenant l'ensemble de ses solutions et cliquer sur le bouton "Soumettre" si la date limite de soumission n'est pas dépassée. La plateforme crée alors l'objet cas de test associé au TP et l'exécute sur les fichiers de solution de l'étudiant. Les cas de test sont des scripts qui vérifient si la solution de l'étudiant répond aux exigences du travail pratique. Une fois les cas de test exécutés, la plateforme attribue une note à la solution de l'étudiant après l'interaction entre la page web et l'objet cas de test. Cette note est basée sur le nombre de cas de test réussis.

3.3.5 Diagramme de classe détaillé du système

Le diagramme de classe détaillé représente une décomposition plus approfondie de la classe système figurant dans le diagramme de classe général déjà établi. Il offre une vue plus détaillée de la structure du système en identifiant et en décrivant les différentes sous-classes possibles, ainsi que les relations et les interactions entre elles. Cette représentation permet une compréhension plus précise des composants et des fonctionnalités du système, facilitant ainsi sa conception, son développement et sa maintenance... [44]

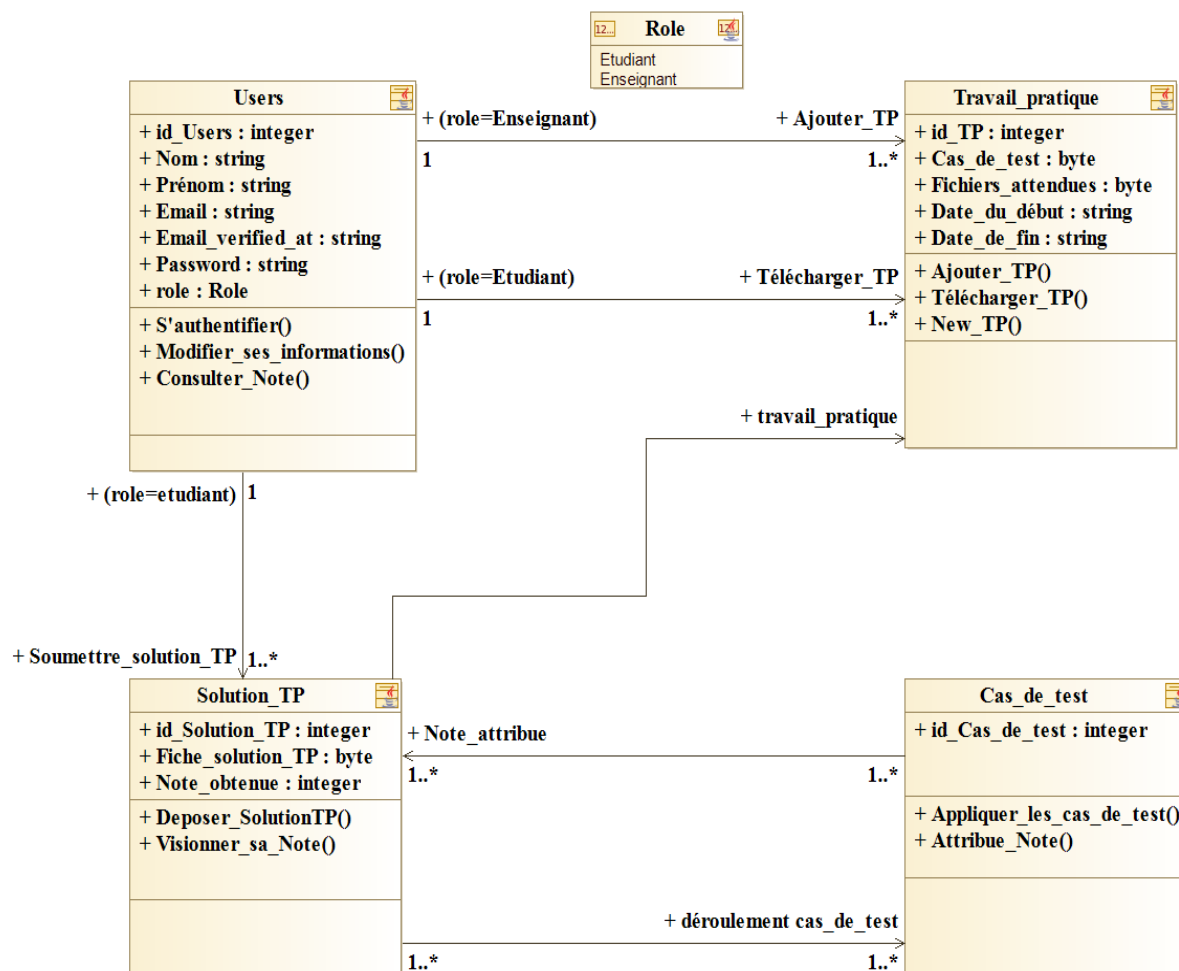


Figure 3. 10 : Diagramme de classe détaillée du système

La Figure 3.10 illustre le diagramme de classes détaillée pour le système de gestion des travaux pratiques (T.P). Les utilisateurs, divisés en étudiants et enseignants, possèdent des attributs. Les enseignants peuvent ajouter des travaux pratiques, que les étudiants peuvent télécharger. Les étudiants soumettent ensuite leurs solutions de TP, qui sont évaluées via des cas de test. Chaque solution de TP est associée à des cas de test spécifiques, permettant l'évaluation et l'attribution de notes. Ce diagramme montre les interactions et relations entre les différentes entités du système, incluant les rôles des utilisateurs et les processus de soumission et d'évaluation des TP.

3.3.6 Diagramme de classe objet

Il Représente des objets (c'est-à-dire instances de classes) et leurs liens (c'est-à-dire instances de relations). Ce diagramme est Utilisé pour illustrer le modèle de classes et prendre une image du système à un moment donné [46]. Le diagramme de classe système modélise les règles tandis que Le diagramme d'objet modélise des faits [46].

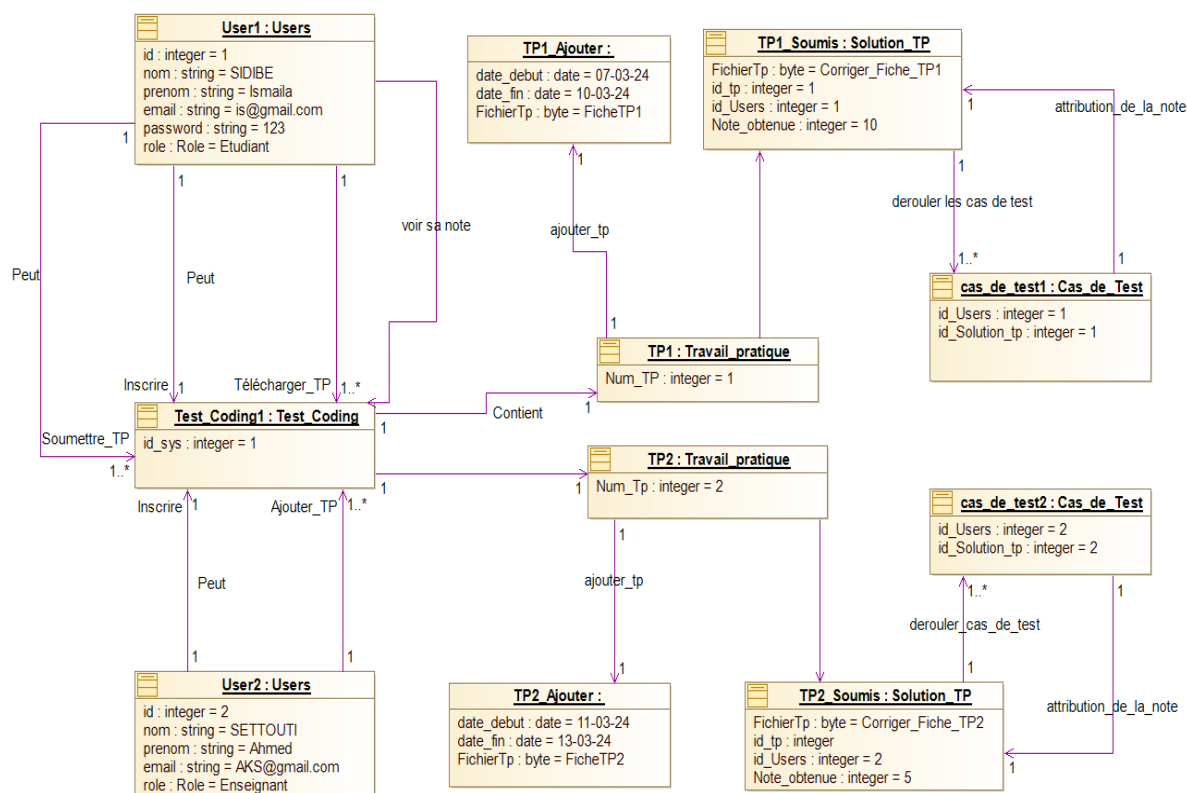


Figure 3. 11 : Diagramme d'Objets

La Figure 3.11 présente un diagramme d'objets pour le système de gestion des travaux pratiques (T.P), illustrant les interactions entre les utilisateurs (un étudiant nommé Sidibe et un enseignant nommé Monsieur Settouti), les T.P, les solutions soumises et les cas de test. Sidibe, l'étudiant, peut s'inscrire, télécharger des TP (TP1 et TP2), et soumettre ses solutions pour évaluation. Settouti, l'enseignant, peut inscrire des utilisateurs et ajouter des TP au système. Les TP téléchargés par les étudiants sont gérés par le système de *TestCoding1*, qui contient les TP ajoutés et les solutions soumises. Les cas de test (cas de test1 et cas de test2) sont utilisés pour évaluer les solutions soumises, dérouler les tests et attribuer des notes. Ce diagramme montre comment les TP sont ajoutés par l'enseignant, téléchargés par l'étudiant, soumis pour évaluation, et évalués à l'aide de cas de test, décrivant ainsi un exemple détaillé des interactions dans un système de gestion de T.P.

3.3.7 Scénario Global pour l'utilisation de notre plateforme

L'utilisation de notre plateforme test Coding implique une série d'interactions dynamiques entre les utilisateurs (étudiants et enseignants) et le système. Ce scénario global guide à travers les différentes étapes, offrant un aperçu complet du parcours utilisateur :

- L'étudiant accède à la plateforme de test des solutions TP.
- L'étudiant se connecte à son compte en utilisant ses identifiants personnels.
- Après s'être authentifié avec succès, l'étudiant explore les différentes fonctionnalités disponibles sur la plateforme, telles que la consultation de son profil, la modification de ses informations, le téléchargement du TP disponible, la soumission des Tps et la consultation de ses notes.
- L'étudiant consulte son profil pour vérifier ses informations personnelles.
- Si nécessaire, l'étudiant modifie ses informations personnelles.
- En modifiant le(s) champs concernés et cliquer sur le bouton "Save" Pour effectuer la mise à jour de ses profils.
- Ensuite, l'étudiant peut télécharger le TP pour le résoudre.
- Pour télécharger le TP l'étudiant cliquer sur le bouton "télécharger TP".
- En cas de dépassement de la date limite le système affiche le message "TP terminé".
- Dans le cas échéant il permet le téléchargement du TP disponible.
- Après avoir travaillé sur sa solution, l'étudiant soumet son TP.
- Pour la soumission du TP le système vérifié si le délai n'est pas dépassé il affiche le formulaire de remise du TP.
- Une fois le formulaire rempli l'étudiant soumet le TP et le système effectuer une vérification sur le formulaire avant de le valider.
- Mais dans le cas où la date limites est dépassé alors dans ce cas il est impossible de soumettre le TP.
- Sinon le serveur lance l'analyse automatique du code source sur le fichier que l'étudiant a soumis.
- Enfin, l'étudiant Peut consulter ses notes pour le TP soumis.
- L'enseignant se connecte à la plateforme en utilisant ses identifiants de connexion.
- Après authentification, l'enseignant accède à son compte.
- Depuis son espace, l'enseignant peut effectuer plusieurs actions telles que :
 - Consulter la liste des Etudiants.
 - Ajouter un nouveau TP, en spécifiant une date de début, une date de fin ainsi que les cas de tests possibles et les noms des fichiers attendus.

-
- L’enseignant peut ajouter un TP, en cliquant sur le bouton ajouter TP et en remplissant le formulaire avec les détails du nouveau TP.
 - Après avoir enregistré les informations, le système ajoute le TP à la base de données.
 - Une fois que l'enseignant a terminé ses actions, il peut choisir de se déconnecter.

Le scénario global décrit en détail les interactions entre les utilisateurs et la plateforme, illustrant les différentes fonctionnalités disponibles pour chaque type d'utilisateur. Cette description détaillée permet de visualiser le fonctionnement concret de la plateforme et de comprendre les différentes actions possibles pour chaque utilisateur.

3.4 Conclusion

La conception de l'architecture de la plateforme Test Coding s'est révélée être une étape fondamentale pour garantir sa réussite. En adoptant une approche méthodique et en utilisant des outils UML pertinents, nous avons acquis une vision claire et précise de la structure et du fonctionnement du système. Ces efforts ont permis une analyse approfondie des interactions entre les utilisateurs et la plateforme, clarifiant ainsi les besoins et les attentes de chaque acteur impliqué. En définitive, cette modélisation rigoureuse a constitué le fondement solide sur lequel reposent la conception, le développement et la mise en œuvre réussie de la plateforme.

Chapitre IV : Présentation du système

4.1 Introduction

La mise en œuvre de notre plateforme TestCoding implique l'utilisation d'outils et de technologies spécifiques pour assurer la réalisation et le bon fonctionnement du site web. Pour ce faire, nous avons examiné plusieurs solutions répondant à nos besoins, afin de faire des sélections. Notre choix s'est fondé sur une évaluation des critères d'expérience utilisateur, de performance et de facilité d'utilisation. Dans ce chapitre, nous détaillerons notre démarche de sélection, en présentant les technologies retenues, et en expliquant le fonctionnement de la plateforme. De plus nous procéderons à une comparaison approfondie avec les solutions existantes ainsi que les capture d'écran du site web réalisé.

4.2 Domaines technique et technologiques utilisés

4.2.1 Options pour les outils de développement

Pour la conception et la réalisation de notre Système, plusieurs outils sont envisageables, répartis en différentes catégories :

☐ **Diagrammes UML :**

Pour la modélisation de l'architecture du système, nous pourrions utiliser des logiciels tels que Modelio [47], draw.io, ArgoUML [48], Entreprise Architect [49], Lucidchart [50], StarUML [51], MagicDraw [52], jetUML [53]...

☐ **Framework frontend :**

Concevoir une interface utilisateur réactive et conviviale, l'utilisation de frameworks est essentielle. Parmi ces outils, on trouve des frameworks spécialisés tels que : Bootstrap [54], Semantic UI [55], foundation [56], React [57], Vue.js [58], Angular [59] .

☐ **Framework backend :**

Pour la mise en œuvre de la logique métier et des fonctionnalités côté serveur, des Framework comme Laravel [60], Django [61] pourrait être appropriés.

☐ **Bases de données :**

Pour le stockage et la gestion des données, des solutions telles que MySQL [62], PostgreSQL [63], SQL lite [64]...ainsi que d'autres bases de données relationnelles et non relationnelles (NoSQL) telles que MongoDB [65], peuvent être utilisées.

❑ Editeur de code :

Il existe une multitude d'éditeurs de code parmi lesquels nous pouvons citer Visual Studio Code [66], Sublime Text [67], Notepad++ [68] et NetBeans [69] ... chacun offrant ses propres fonctionnalités et avantages pour répondre aux besoins variés des utilisateurs.

4.2.2 Sélection des outils et justifications**❑ Présentation des outils utilisés :****❖ Modelio :**

[47]Modelio, développé par Modeliosoft et basé à Paris, en France, est un outil UML open-source polyvalent. Il est utilisé pour la conception de systèmes logiciels, l'ingénierie système, l'architecture d'entreprise et la modélisation des processus métier.

❖ Visual studio code :

[66]Visual studio code (VS code) est un éditeur de code source gratuit, open-source, multiplateforme et développé par Microsoft, supportant une dizaine de langage.

❖ Laragon :

[70]Laragon est un environnement de développement universel portable, isolé, rapide et puissant pour Windows. Il est conçu pour simplifier la configuration et la gestion des environnements de développement web locaux, et il est livré avec un ensemble d'outils et de services qui sont couramment utilisés dans le développement web, tels qu'Apache, PHP et MySQL. Il comprend également des outils supplémentaires tels que Node.js, Composer et Git.

❖ Bootstrap :

[54]Bootstrap, développé par twitter. Est une collection d'outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc.) de sites et d'applications web. C'est un ensemble qui contient des codes HTML (HyperText Markup Language) et CSS (Style Sheet Cascading), des formulaires, boutons, outils de navigation et autres éléments interactifs, ainsi que des extensions JavaScript en option. C'est l'un des projets les plus populaires sur la plate-forme de gestion de développement GitHub.

❖ Laravel :

[60]Laravel est un framework PHP open-source et gratuit, spécialement conçu pour simplifier et accélérer le développement d'applications web complexes. Respectant le principe du modèle-vue-contrôleur (MVC) et entièrement élaboré en programmation orientée objet (POO), il offre une structure claire et élégante pour organiser votre code de manière efficace. Créé par Taylor Otwell en juin 2011, Laravel est devenu un choix privilégié pour de nombreux développeurs grâce à sa facilité d'utilisation et ses fonctionnalités avancées.

❑ Justifications :

Nous choisirons Modelio pour la modélisation de notre architecture logicielle. Nous avons déjà acquis de l'expérience avec ce logiciel lors de notre troisième année de licence, dans le cadre du module de génie logiciel. Le choix d'utiliser Laravel et Bootstrap repose sur notre expérience préalable avec ces technologies. Lors du module de web avancé avec Monsieur Tadlaoui au cours de notre précédent cursus en master 1, nous avons travaillé sur des projets Laravel qui nous ont permis de bien maîtriser ce framework. Cette familiarité avec Laravel, combinée à notre connaissance de Bootstrap, nous assure que nous pourrions créer une interface utilisateur moderne, réactive et esthétiquement agréable pour notre plateforme. Enfin, pour faciliter notre processus de développement, on choisira Laragon qui offre un environnement de développement complet, intégrant les outils de gestion de la base de données.

4.3 Formation dans le domaine

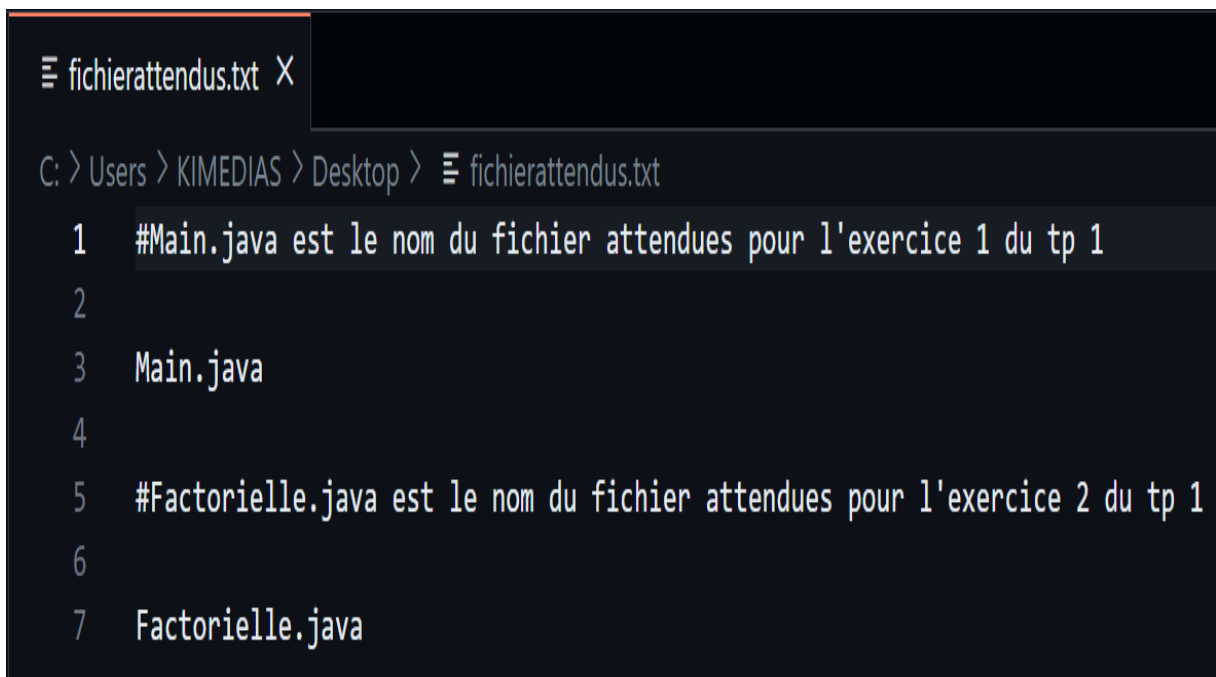
Pendant notre formation en Master 2 génie logiciel, durant l'année académique 2023-2024, nous avons suivi le module de validation et vérification (V&V) sous la direction de l'assistante de Madame Seladji, Madame Wahiba Bachiri. Ce module nous a offert l'opportunité de concevoir des cas de tests en nous appuyant sur les spécifications fournies. Cela comprenait notamment des tests en boîte noire pour évaluer le comportement global du code source. Nous avons également réalisé des tests en boîte blanche pour analyser sa structure interne. En plus de

cela, des tests d'intégration ont été effectués pour vérifier la compatibilité et l'interaction entre les différentes parties du système. Enfin, des tests de développement dirigé ont été utilisés pour valider les fonctionnalités. Dans le cadre de ce module, nous avons concrétisé ces concepts à travers divers exercices et travaux pratiques, les codes sources à tester étaient écrits en langage Java. Nous avons élaboré des plans de test, identifié des cas de test pertinents, et utilisé des outils ainsi que des techniques de test pour évaluer la conformité des codes sources aux spécifications fonctionnelles et assurer leur bon fonctionnement.

Bien que notre expérience dans ce domaine puisse être limitée, cette introduction nous a permis de comprendre certains aspects fondamentaux des tests visant à vérifier l'exécution du code source.

4.4 Réalisation des Fichiers attendus

Vérifier si l'étudiant a respecté le nom du fichier requis tel qu'indiqué dans les spécifications sur la série de T.P.

A screenshot of a text editor window titled 'fichierattendus.txt'. The window shows a list of expected files for two exercises. Line 1: '#Main.java est le nom du fichier attendus pour l'exercice 1 du tp 1'. Line 2: (empty). Line 3: 'Main.java'. Line 4: (empty). Line 5: '#Factorielle.java est le nom du fichier attendus pour l'exercice 2 du tp 1'. Line 6: (empty). Line 7: 'Factorielle.java'. The editor has a dark background with light-colored text.

```
1 #Main.java est le nom du fichier attendus pour l'exercice 1 du tp 1
2
3 Main.java
4
5 #Factorielle.java est le nom du fichier attendus pour l'exercice 2 du tp 1
6
7 Factorielle.java
```

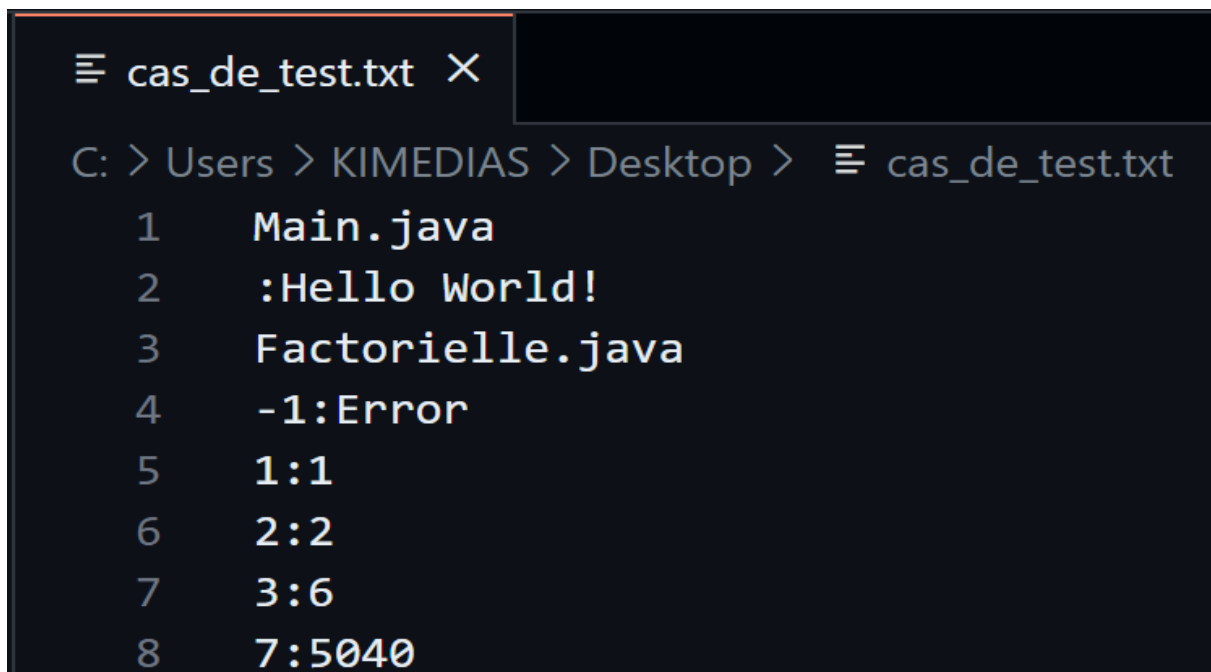
Figure 4. 1 : exemple de fichier attendus

La Figure 4.1 représente un fichier au format texte qui contient une liste de noms de fichiers attendus, chaque nom étant sur une ligne distincte. Ces noms de fichiers correspondent aux programmes que nous souhaitons tester.

4.5 Réalisation des Cas de tests

Le test est l'exécution ou l'évaluation d'un système, d'un programme informatique ou d'un composant par des moyens automatiques ou manuels, pour vérifier qu'il répond à ses spécifications ou identifier les différences entre les résultats attendus et les résultats obtenus [21].

Tester, c'est exécuter le programme dans l'intention d'y trouver des anomalies ou des défauts [71]. La figure ci-dessous représente un exemple de cas de test que nous avons utilisé au cours du projet.



```
cas_de_test.txt X
C: > Users > KIMEDIAS > Desktop > cas_de_test.txt
1    Main.java
2    :Hello World!
3    Factorielle.java
4    -1:Error
5    1:1
6    2:2
7    3:6
8    7:5040
```

Figure 4. 2 : Exemple de cas de test

La figure 4.2 représente un fichier de cas de test divisé en deux parties distinctes, chacune dédiée au test d'un fichier spécifique : "Main.java" et "Factorielle.java". Dans la première partie, le cas de test pour "Main.java" est défini. Cette section commence par spécifier le nom du fichier attendu, suivi de la description du cas de test. Le programme "Main.java" doit afficher "Hello World!" en sortie et ne nécessite pas d'entrée. La seconde partie du fichier contient plusieurs cas de test pour le fichier "Factorielle.java". Chaque cas de test consiste en une paire d'entrée-sortie attendue, où l'entrée représente une valeur donnée au programme et la sortie la valeur attendue. Par exemple, lorsque l'entrée est "0", la sortie attendue est "1", et lorsque l'entrée est "2", la sortie attendue est "2". En résumé, ce fichier de cas de test fournit une méthode structurée pour vérifier le fonctionnement des programmes "Main.java" et "Factorielle.java". Chaque cas de test spécifie clairement les conditions d'entrée et les résultats attendus. Si les programmes produisent les sorties attendues pour chaque cas de test, cela

indique qu'ils fonctionnent correctement. Il est essentiel de noter que la notation avec les deux points (:) dans chaque cas de test permet une spécification claire de l'entrée fournie au programme et de la sortie attendue. Le serveur lit et récupère pour chaque fichier attendu le cas de test associé en récupérant l'entrée et la sortie pour les stocker en base de données.

4.6 Architecture du système

4.6.1 Côté enseignant

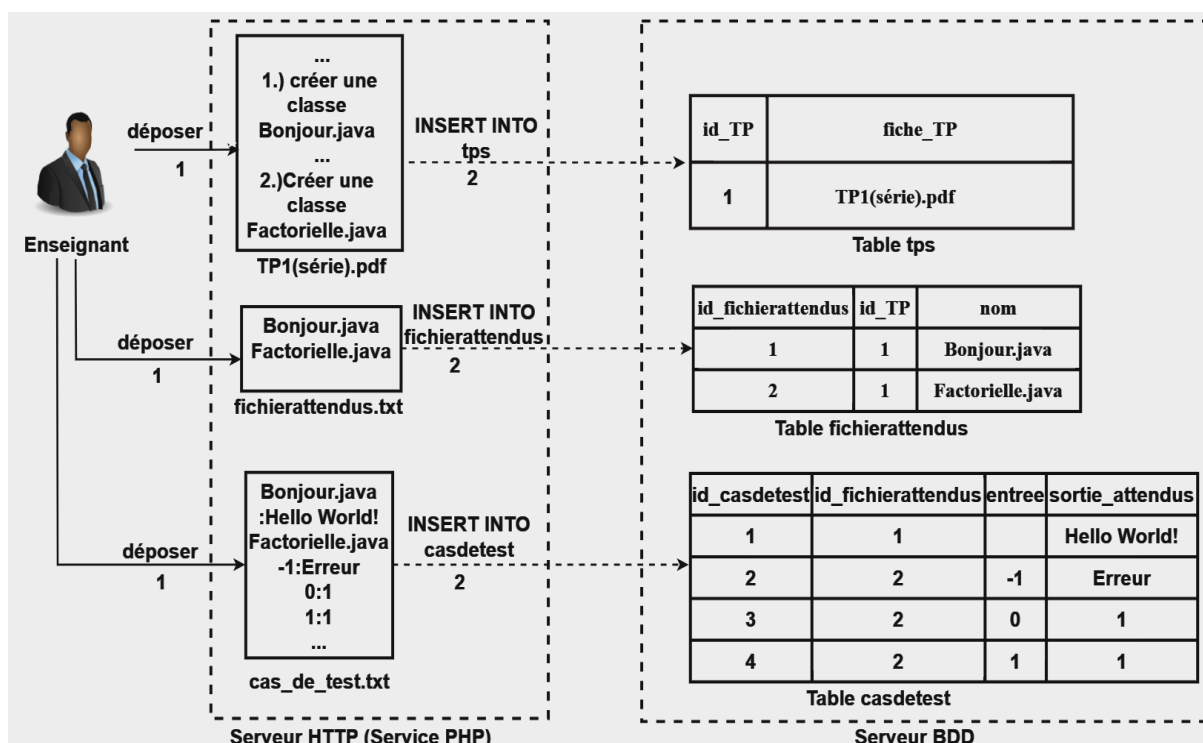


Figure 4.3 : Architecture du système côté enseignant

La figure 4.3 illustre le processus d'ajout du Travail Pratique sur la plateforme TestCoding, géré par un serveur HTTP (Service PHP) et un serveur de base de données (BDD). D'abord, l'enseignant dépose un document de spécifications de TP, "TP1(série).PDF", sur le serveur http, spécifiant les tâches à réaliser, comme la création des classes "Bonjour.java" et "Factorielle.java". Ce fichier est inséré dans la table "tps" de la BDD. Ensuite, l'enseignant dépose "fichierattendus.txt", contenant les fichiers attendus ("Bonjour.java" et "Factorielle.java"). Ces informations sont insérées dans la table "fichierattendus" de la BDD, chaque fichier étant lié à l'ID du TP correspondant. Enfin, l'enseignant dépose "cas_de_test.txt", contenant des cas de test pour les classes soumises par les étudiants, comprenant des entrées et des sorties attendues. Ces cas de test sont insérés dans la table "casdetest" de la BDD, avec des références aux fichier attendus. Tous ces données insérées permettent d'évaluer automatiquement les soumissions des étudiants grâce à l'analyse des cas de test sans intervention continue de l'assistant.

4.6.2 Côté étudiant

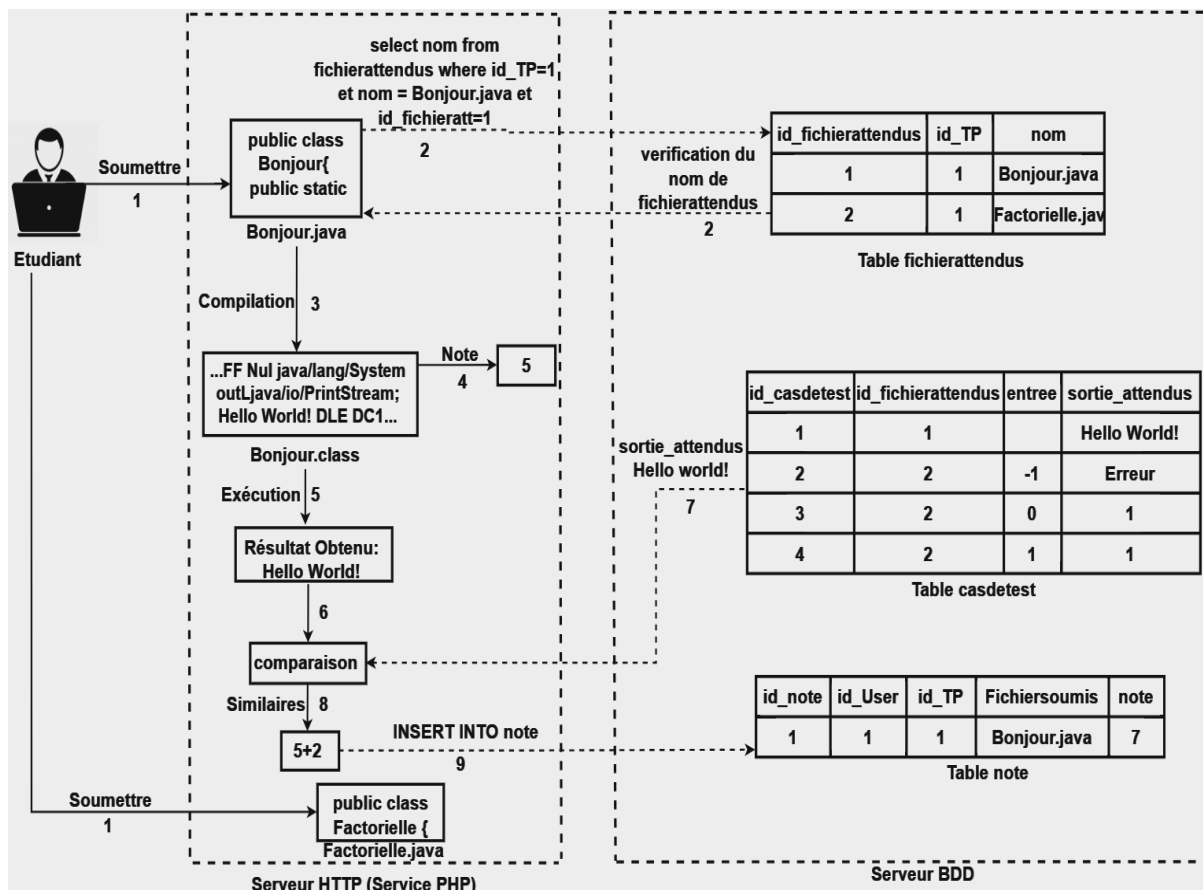


Figure 4. 4 : Architecture du système côté étudiant

La Figure 4.4 décrit le processus de soumission de la solution par les étudiants et le traitement effectué côté backend. L'étudiant soumet le fichier "Bonjour.java" via le Serveur HTTP (Service PHP), qui vérifie le nom du fichier soumis en le comparant aux fichiers attendus stockés dans la table "fichierattendus" de la BDD. Une requête SQL est exécutée pour confirmer que le fichier soumis correspond à "Bonjour.java" (attendu pour le TP en cours). Si le fichier est correct, le serveur procède à la compilation de "Bonjour.java" en "Bonjour.class". Si la compilation est réussie, une note est octroyée. Après cela, le serveur exécute le fichier compilé et compare sa sortie à la sortie attendue (c'est-à-dire « Hello World! ») définie dans la table "casdetest" de la BDD. Si la sortie est correcte, une note (dans notre contexte, c'est 2) est attribuée au fichier soumis et cette note est insérée dans la table "notes". L'étudiant soumet également "Factorielle.java", qui suit un processus similaire de vérification, compilation, exécution et notation. Les résultats des cas de test pour le "Factorielle.java" sont vérifiés et notés en fonction des entrées et des sorties attendues définies dans la table "casdetest". En résumé, la Figure 4.4 présente un processus automatisé de soumission, vérification, compilation, exécution et notation des travaux pratiques, permettant une évaluation efficace et sans intervention continue des assistants de travaux pratiques.

4.7 Implémentation de notre plateforme « Test Coding »

4.7.1 Page d'accueil

L'utilisateur peut accéder à la plateforme en cliquant sur le bouton "Connexion", ce qui ouvrira une nouvelle page pour lui.



Figure 4. 5 : Page d'accueil du plateforme Test Coding

La Figure 4.5 illustre la page d'accueil de notre plateforme TestCoding. À gauche et au centre, un bouton "Connexion" permet aux utilisateurs de se connecter à leur compte. La plateforme met en avant ses partenaires, représentés par des logos d'entreprises et plateformes connues telles que HackerRank, LeetCode, Google, Coursera. La navigation est simplifiée avec un lien "Contact" en haut à droite pour permettre aux utilisateurs de contacter les administrateurs de la plateforme en cas de problème avec le système. Cette page est animée, indiquant une interface moderne et interactive.

4.7.2 Formulaire d'authentification

C'est la page qui permet à un utilisateur déjà inscrit de se connecter et d'accéder à son menu.

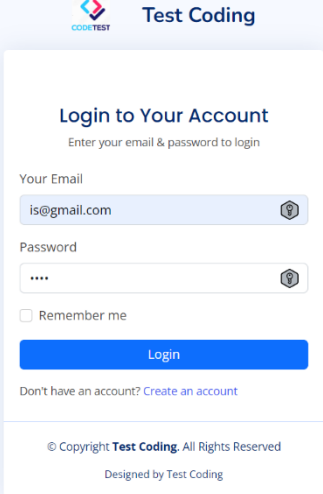


Figure 4. 6 : Page d'authentification

La figure 4.6 présente le formulaire d'authentification du système "Test Coding". Ce formulaire comprend un logo et le nom du système en haut, suivis de champs pour entrer l'adresse e-mail et le mot de passe, avec l'e-mail préremplie en tant qu'exemple. Il inclut également une case à cocher "Remember me" une option pour se souvenir de l'utilisateur sur cet appareil, un bouton de connexion bleu "Login" pour soumettre ses informations et se connecter, et un lien pour créer un compte pour les nouveaux utilisateurs. En bas de la page, on trouve les informations de copyright et de conception, indiquant que le design a été réalisé par "Test Coding". L'interface est claire et simple, facilitant la connexion des utilisateurs à leurs comptes.

4.7.3 Côté enseignant

❑ Page des Travail pratique disponibles

TP ajouter avec success!

Travaux Pratique

Home / Travaux Pratique

+ Ajouter Travaux Pratique

10 entries per page

N°TP	Date Debut	Date Fin	Actions
1	2024-04-01	2024-04-08	Voir Supprimer
2	2024-04-08	2024-04-15	Voir Supprimer
3	2024-04-15	2024-04-27	Voir Supprimer
4	2024-04-21	2024-04-23	Voir Supprimer
5	2024-04-21	2024-04-22	Voir Supprimer
6	2024-04-15	2024-04-30	Voir Supprimer

Showing 1 to 1 of 1 entries

Figure 4. 7 : Page de la Liste des tps

Figure 4.7 illustre la page de liste des travaux pratique (T.P) sur la plateforme TestCoding. Cette page permet à un enseignant de visualiser et de gérer tous les TP qu'il a créés. L'enseignant peut visualiser les informations de chaque TP, telles que le numéro, la date de début et fin.

❑ Formulaire d'ajout du travail pratique

Ajouter Travail Pratique

Home / Travaux Pratique / Ajouter Travail Pratique

Date Debut: 15/04/2024

Date Fin: 30/04/2024

Fiche TP: Choisir un fichier TP1 De POO.pdf

Fichier Attendus: Choisir un fichier fichierattendus.txt

Cas de test: Choisir un fichier cas_de_test.txt

[Ajouter](#)

© Copyright Test Coding. All Rights Reserved
Designed by Test Coding

Figure 4. 8 : Page d'ajout du travail pratique

Nous voyons dans la figure 4.8 le formulaire d'ajout du TP, rempli à titre d'exemple avec tous les détails nécessaires pour l'ajout d'un nouveau travail pratique. Ce formulaire permet à un enseignant de spécifier les informations suivantes : la date de début et de fin du travail pratique, la série de TP à déposer, les cas de test requis et les fichiers attendues. L'exemple rempli dans le formulaire illustre comment un enseignant peut utiliser ce système pour organiser efficacement les travaux pratiques, en veillant à ce que toutes les informations nécessaires soient fournies de manière exhaustive et cohérente.

❑ Page Liste des étudiants inscrits

The screenshot displays the 'Liste des Etudiants' page. At the top, there's a header with the 'Test Coding' logo, a search bar, and a user profile 'SETTOUTI Ahmed Khaled Yassine'. The left sidebar contains a 'Menu' section with links to 'Travaux Pratique', 'La note des etudiants', and 'Liste des Etudiants'. The main content area shows a table of students. The table has columns for 'Nom', 'Prenom', 'Email', and 'Actions'. There are 10 entries, each with a 'Supprimer' button. Below the table, it says 'Showing 1 to 10 of 10 entries'.

Nom	Prenom	Email	Actions
SIDIBE	Ismaila	is@gmail.com	Supprimer
DIARRA	Djenebou	dd@gmail.com	Supprimer
DIONE	Brehima	bd@gmail.com	Supprimer
KEITA	Makan	mk@gmail.com	Supprimer
TRAORE	Moussa	mt@gmail.com	Supprimer
COULIBALY	Mohamed	mc@gmail.com	Supprimer
DEMBELE	Salia	sd@gmail.com	Supprimer
SIDIBE	Chiaka	cs@gmail.com	Supprimer
DIARRA	Tata	td@gmail.com	Supprimer
TALL	Cheick Sadibou	ct@gmail.com	Supprimer

Figure 4. 9 : Page de la liste des étudiants

La figure 4.9 représenté la liste complète du nom des étudiants inscrites, visible uniquement pour les enseignants.

4.7.4 Côté étudiant(e)

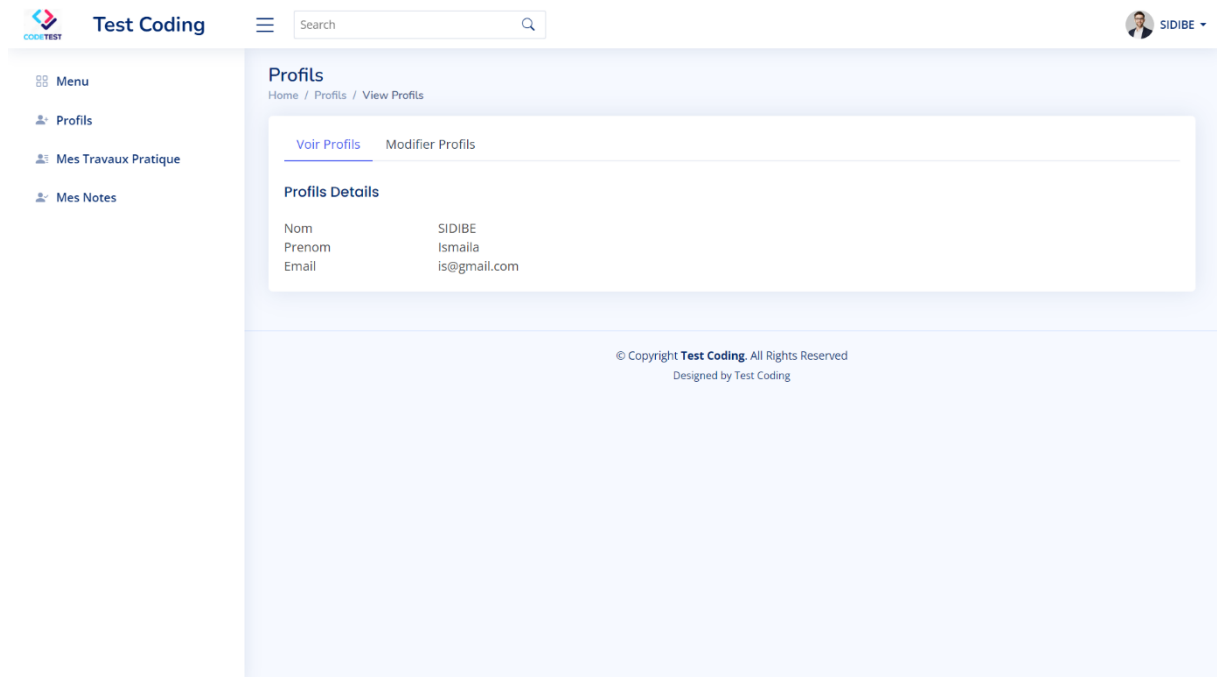
❑ Page du profils détails :

Figure 4. 10 : Page du profils détails de l'étudiant

Dans la Figure 4.10, nous voyons l'interface de la section des profils de l'étudiant sur la plateforme TestCoding. Cette section permet aux étudiants connecter de visualiser et de modifier leurs informations personnelles. L'écran affiché montre les détails du profil de l'utilisateur, incluant le nom, le prénom et l'email. Cette disposition permet à l'étudiant de vérifier et de mettre à jour leur information personnelles.

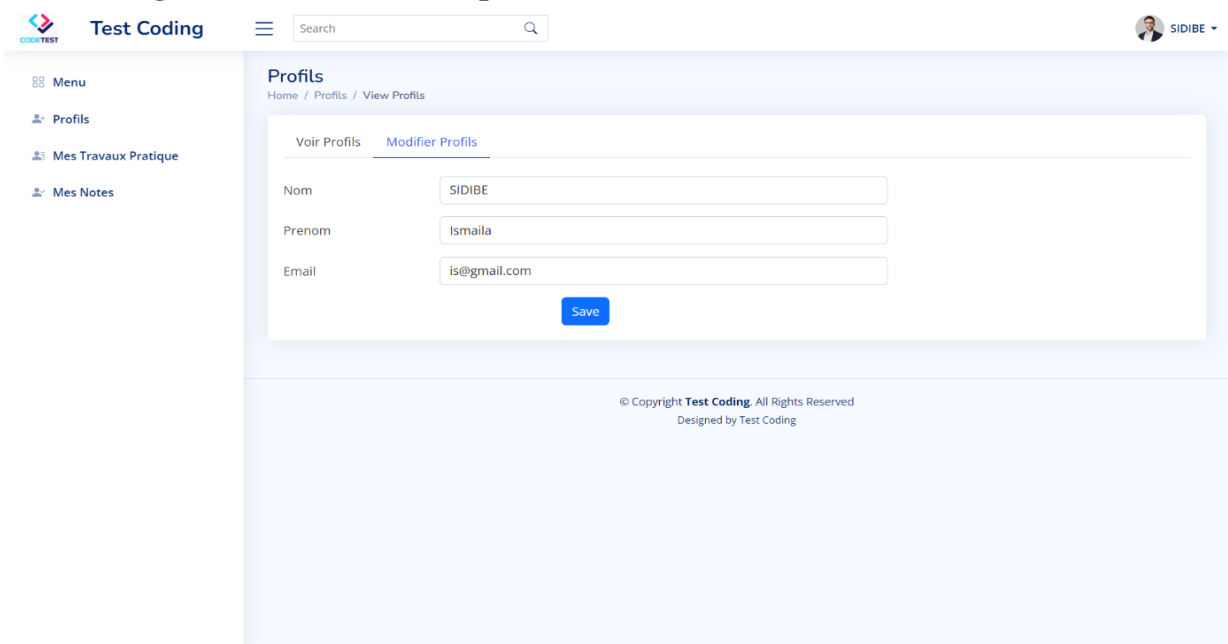
❑ Page de modification du profil :

Figure 4. 11 : Page de modification du profil de l'étudiant

Dans la figure 4.11, nous voyons l'interface de modification des profils de l'étudiant sur la plateforme. L'écran affiché montre un formulaire avec des champs modifiables pour le nom, le prénom et l'email de l'utilisateur. Une fois les modifications terminées, un bouton Save en bleu permet de sauvegarder les informations mises à jour.

❑ **Page des travail pratique ajouté par enseignant :**

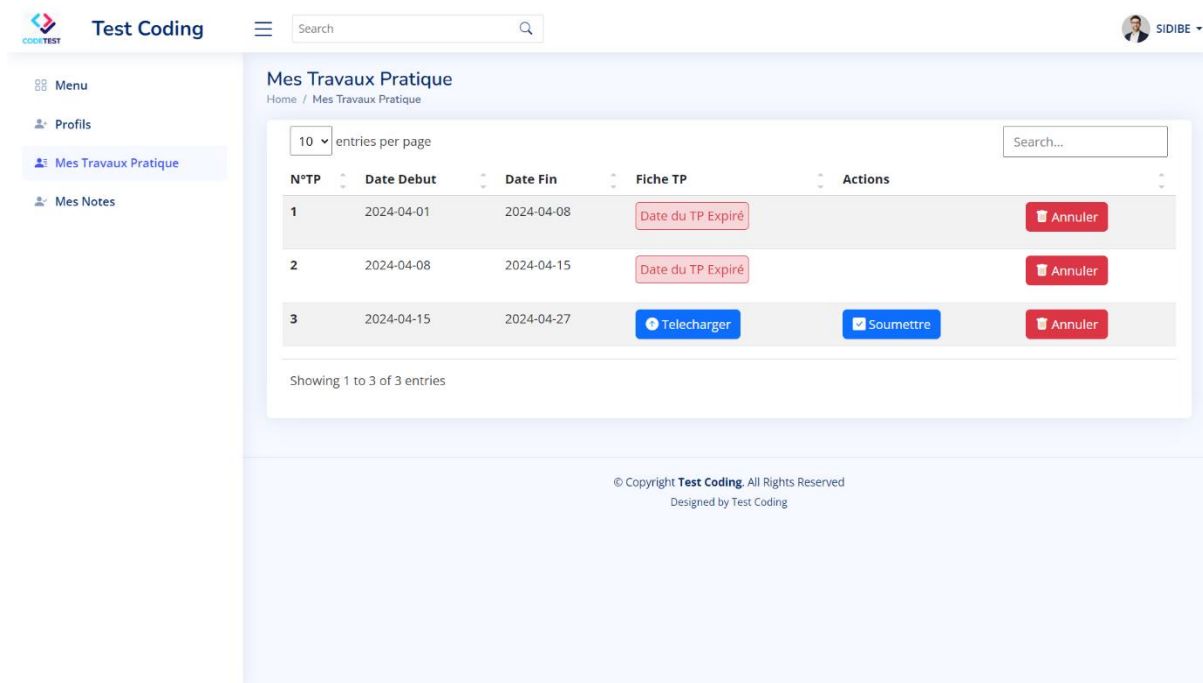


Figure 4. 12 : Page des travaux pratiques

La Figure 4.12 présente les travaux pratique disponible sur la plateforme TestCoding. Cette section permet aux étudiants de visualiser, l'écran affiche un tableau listant les TP avec leur date et fin ainsi que des options pour chaque TP. Dans cet exemple, trois TP sont listés le premier et le deuxième TP sont expirés car le délai du travail a déposé est dépasser tandis que le troisième TP est en cours, avec des butons télécharger permettant aux étudiants de télécharger la série de TP ainsi que le bouton soumettre pour rendre leur solution au travail pratique.

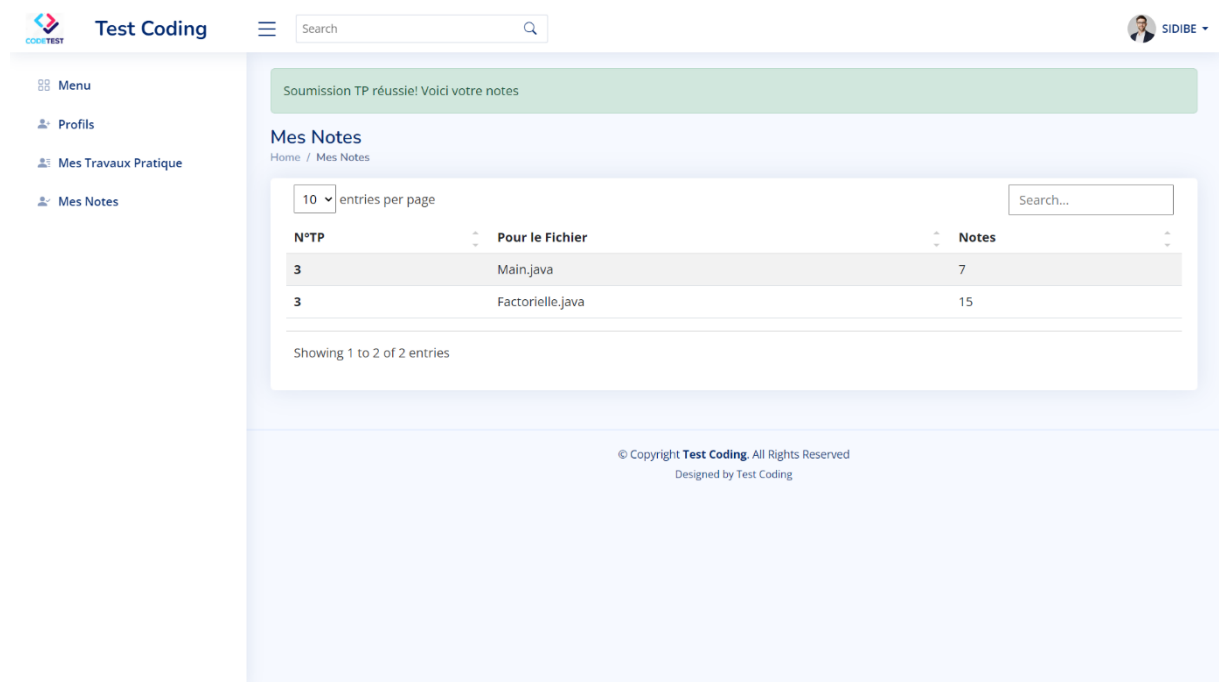
❑ Page du formulaire de soumission Travail pratique :

Figure 4. 13 : Page de soumission du travail pratique

Figure 4.13 montre la page de soumission du travail pratique sur le site TestCoding. La page, intitulée Soumettre TP, permet aux étudiants de télécharger des fichiers solutions pour la correction. Deux champs sont visibles pour choisir des fichiers, intitulés "Fiche Corriger TP", avec des fichiers "Main.java" et "Factorielle.java" déjà sélectionnés. Un bouton "Soumettre" se trouve en dessous pour envoyer les fichiers.

❑ Page de la note après soumission :

Pour l'évaluation du fichier soumis, les notes sont distribuées en fonction de deux critères principaux : la compilation du fichier et le résultat du cas de test. En cas de compilation réussie du fichier, une note de 5 est octroyée. De plus, chaque cas de test passé avec succès ajoute 2 points à la note. Cependant, en cas d'échec de la compilation, une note de 0 est attribuée. La figure ci-dessous représente les notes obtenues pour des fichiers soumis qui passent tous les cas de test.



The screenshot shows the 'Test Coding' website interface. At the top, there is a navigation bar with the 'Test Coding' logo, a search bar, and a user profile icon labeled 'SIDIBE'. A left sidebar contains a menu with options: 'Menu', 'Profils', 'Mes Travaux Pratique', and 'Mes Notes'. The main content area features a green confirmation banner at the top stating 'Soumission TP réussie! Voici votre notes'. Below this, the 'Mes Notes' section is displayed, showing a table of submitted files. The table has columns for 'N°TP', 'Pour le Fichier', and 'Notes'. Two entries are visible: 'Main.java' with a score of 7 and 'Factorielle.java' with a score of 15. A footer at the bottom of the page reads '© Copyright Test Coding. All Rights Reserved. Designed by Test Coding'.

N°TP	Pour le Fichier	Notes
3	Main.java	7
3	Factorielle.java	15

Figure 4. 14 : Page de la note

Figure 4.14 illustre la page de résultats de soumission de travaux pratiques sur le site "Test Coding". Après avoir soumis un travail pratique, un message de confirmation indique "Soumission TP réussie ! Voici votre note". La section "Mes Notes" liste les fichiers soumis avec leurs notes respectives. Deux entrées sont visibles : "Main.java" avec une note de 7 et "Factorielle.java" avec une note de 15.

❑ Page de la soumission avec des nom de fichier Incorrect :

The screenshot displays the 'Soumettre TP' (Submit Assignment) page in the Test Coding application. The page has a light blue header with the 'Test Coding' logo and a search bar. A sidebar on the left contains a menu with options: 'Menu', 'Profils', 'Mes Travaux Pratique', and 'Mes Notes'. The main content area is titled 'Soumettre TP' and includes a breadcrumb trail: 'Home / Mes Travaux / Formulaire Soumission'. Below this, there are two rows for file selection, each labeled 'Fiche Corriger TP'. The first row shows a button 'Choisir un fichier' and the text 'Main.java'. The second row shows a button 'Choisir un fichier' and the text 'fact.java', which is highlighted with a red rectangle. A blue 'Soumettre' button is located at the bottom right of the form. At the bottom of the page, there is a copyright notice: '© Copyright Test Coding. All Rights Reserved. Designed by Test Coding'.

Figure 4. 15 : Page de soumission avec du nom de fichier incorrecte

Figure 4.15 montre une interface utilisateur de TestCoding pour la soumission de travaux pratique. Dans cette interface, deux champs sont disponibles pour la sélection de fichiers sous l'étiquette "Fiche Corriger TP". Le premier champ montre que le fichier "Main.java" a été sélectionné correctement. Cependant, le deuxième champ, encadré en rouge, affiche le fichier "fact.java", ce qui indique une erreur dans la sélection du fichier à soumettre.

❑ Page après soumission de nom du fichier incorrect :

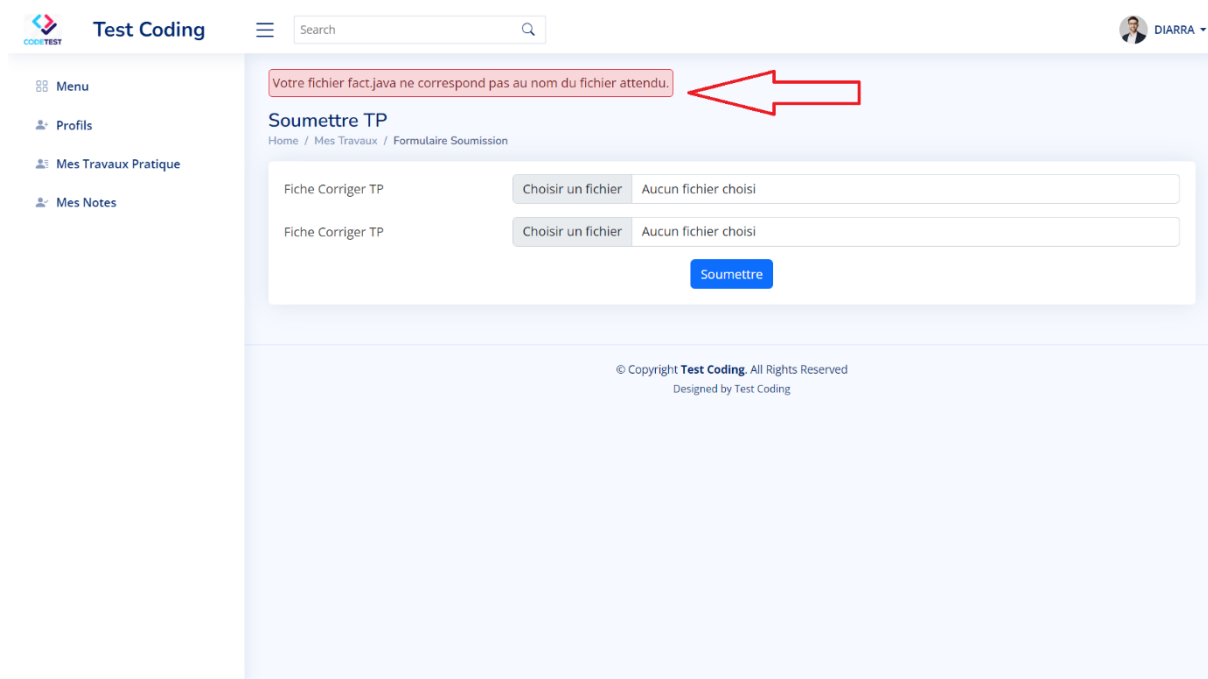


Figure 4. 16 : Page après soumission du nom de fichier incorrect

La Figure 4.16 montre la page de soumission de travaux pratiques sur le site Test Coding après qu'un utilisateur a tenté de soumettre un fichier incorrect. En haut de la page, un message d'erreur en rouge indique : "Votre fichier fact.java ne correspond pas au nom du fichier attendu." Ce message signale que le fichier soumis ne correspond pas aux critères de nommage attendus par le système. Le formulaire "Soumettre TP" affiche deux champs de fichier, tous deux marqués "Aucun fichier choisi" après la tentative de soumission incorrecte. Cela suggère que le système a réinitialisé les sélections de fichiers suite à l'erreur. Un bouton bleu "Soumettre" est positionné en dessous des champs de fichier pour permettre une nouvelle tentative de soumission.

4.8 Avantages et bénéfices

Notre plateforme TestCoding offre une série d'avantages significatifs pour les étudiants et les enseignants. Tout d'abord, elle allège considérablement la charge de travail pour les assistants de TP. En outre elle optimise leur temps pour orienter les étudiants et note leur présence. L'un des principaux points forts de notre plateforme est son système d'évaluation rapide et précise avec l'utilisation des cas de test qui couvre le maximum de cas possible pour détecter les erreurs dans leur code, de plus il facilite la détection des erreurs dans le code et offre une consultation plus facile des notes obtenues à chaque travail pratiques.

4.9 Comparaison entre notre projet et les travaux similaires

Pour mon projet, il est essentiel de réaliser une comparaison approfondie avec des travaux similaires afin d'évaluer la pertinence et l'originalité de mon travail de recherche. Après avoir examiné en détail plusieurs projets précédents dans le même domaine, cette comparaison a pour objectif de mettre en évidence les avantages et les inconvénients de chaque approche. Grâce à l'analyse de projets ayant des objectifs, des méthodes et des résultats similaires, je peux mieux placer mon projet dans le contexte actuel de la recherche et du développement. Il est crucial de passer par cette dernière étape de comparaison afin de déterminer si mon projet apporte une contribution significative et originale par rapport aux projets déjà en place.

Critère	HackerRank	Sphere Online Judge	HackerEarth	CoderByte	TestCoding
Téléchargements des spécifications des problèmes	NON	NON	NON	NON	OUI
Soumission de fichiers de solution	OUI	OUI	OUI	NON	OUI
Utilisation des jeux de test	OUI	OUI	OUI	OUI	OUI
Délais de soumissions	Uniquement pour les concours	Uniquement pour les concours	Uniquement pour les concours	Uniquement pour les concours	OUI

Tableau 4. 1 : Etude Comparatives de notre plateforme vs plateforme similaires

Dans l'analyse présentée dans le Tableau 4.1, aucune des plateformes examinées ne répond au critère de téléchargement des spécifications des problèmes, ce qui permettrait aux utilisateurs de visualiser les exigences du problème à résoudre hors ligne. De ce fait, une connexion internet est toujours requise pour accéder aux spécifications du problème. En revanche, ma plateforme, TestCoding, permet à ses utilisateurs d'accéder aux spécifications du problème sans nécessiter de connexion internet. De plus, ma plateforme impose une date limite de soumission pour chaque problème, ce qui incite les utilisateurs à travailler sur le problème de manière plus diligente. En comparaison, les autres plateformes ne requièrent pas de délai de soumission sauf au niveau des compétitions ou concours organiser sur les plateformes, ce qui peut encourager un retard prolongé voire l'abandon du problème. À l'exception de Coderbyte, toutes les autres plateformes permettent la soumission de fichiers. Elles utilisent tous également des cas de test pour évaluer le travail en fonction des tests réussis.

Notre projet est en accord avec ceux présentés dans le tableau 2.2 du chapitre II section études comparatives des travaux dans la littérature, qui concernent la mise en place de plateformes web pour l'évaluation et la correction des programmes sources. Cependant, il se distingue du travail de OUAHRANI et AOUN [35], qui se concentre sur la sélection des meilleurs outils existants pour automatiser la correction des TPs de compilation. En comparaison avec le projet de DARI [39] notre approche se rapproche davantage, notamment en ce qui concerne l'utilisation de cas de test et la correction des programmes sources en langage Java. Notre projet se distingue également par son contexte plus spécifique, axé sur les travaux pratiques de programmation orientée objet. Contrairement à la plupart des autres projets qui abordent d'autres aspects à l'exception du projet de MEDJAHED et BOUTI [36] qui est dans le contexte de travaux pratique mais du module de compilation. En termes de résultats attendus, notre projet vise à fournir une évaluation automatique des travaux pratiques écrits en langage Java, ce qui diffère des autres projets qui se concentrent sur d'autres langages de programmation tels que le langage C et Python.

Conclusion Générale

Dans ce projet, nous avons entamé par une exploration des travaux similaires afin de nous assurer qu'il n'existe pas déjà des solutions satisfaisantes ou des approches novatrices qui ont abordé des problématiques similaires. Cette étape a permis de situer notre travail dans contexte des études existantes. Avant de commencer la conception et la réalisation de notre système, nous avons établi un cahier des charges contenant toutes les exigences fonctionnelles et non fonctionnelles nécessaires, répondant aux besoins des futurs utilisateurs de notre système. Cette phase nous a permis de cerner précisément l'objectif de notre projet et de comprendre les besoins de chaque utilisateur.

Nous avons ensuite cherché à avoir une vision claire et précise de l'architecture en utilisant des outils UML et une méthodologie de conception, avec la possibilité de revenir en arrière pour réviser les différents diagrammes du système. Après avoir obtenu une vision claire de notre système à développer, nous avons procédé à sa mise en place. Nous avons créé une interface utilisateur avec Bootstrap et un backend développé grâce au Framework PHP Laravel. Pour faciliter l'utilisation et la mise en place de la base de données, nous avons utilisé l'environnement de développement Laragon, et avons réussi à mettre en place notre système sous le nom de TestCoding.

Au cours de la réalisation de ce projet, nous avons rencontré plusieurs difficultés, notamment l'usage des cas de test pour évaluer les programmes informatiques et leur intégration au système pour automatiser l'évaluation, ainsi que la formalisation des besoins à partir des retours d'expérience. La recherche de cas de test pertinents pour juger équitablement le travail des étudiants a également été un défi, nécessitant une couverture exhaustive des spécifications données dans la série de TP. De plus, la difficulté de faire communiquer PHP avec Java pour compiler l'ensemble des fichiers de solution a été un obstacle majeur.

Malheureusement, tous les objectifs fixés pour ce projet n'ont pas pu être atteints en raison du manque de temps. Par exemple, nous avions prévu d'effectuer les tests en boîte blanche, mais notre travail se contente uniquement des tests en boîte noire. De plus, notre système devait créer lui-même sa propre classe de test et ajouter les cas de test ainsi que les fichiers attendus grâce à une interface graphique, mais ces objectifs n'ont pas pu être réalisés.

Pour les projets futurs, nous envisageons plusieurs améliorations. Nous espérons réaliser l'analyse des tests en boîte blanche et prendre en charge d'autres langages de programmation. De plus, nous prévoyons intégrer des outils permettant d'effectuer la détection de plagiat entre les codes soumis par les étudiants et la génération des commentaires pour les erreurs rencontrées lors de la compilation des fichiers des étudiants. Cela permettra d'éviter les copies colle du code

entre les étudiants, et de les orienter plus efficacement. De plus, les aidants à corriger les erreurs dans leurs code soumise. En somme, ce projet a posé des bases solides pour des développements futurs.

Bibliographie

- [1] K. Boussaha, «L'évaluation de l'apprenant dans les environnements d'apprentissage de TéléTPs (Thèse),» Bibliothèque centrale de l'université de Annaba, Annaba. Retrieved May 9, 2024 , 2015, [En ligne]. Available: <https://biblio.univ-annaba.dz/wp-content/uploads/2019/02/These-Boussaha-Karima.pdf>. [Accès le 31 Mars 2024].
- [2] H. R. Douglas, «Calculating instruments and machines,» *Cambridge University Press*, 1950. [Accès le 31 Mars 2024].
- [3] M. V. Wilkes, «Automatic digital computers,» *Cambridge University Press*, 1956. [Accès le 31 Mars 2024].
- [4] A. F. Blackwell, «What is Programming?,» .PPIG, Brunel University, London, UK, 2002. [Accès le 31 Mars 2024].
- [5] N. Guibert, L. Guittet et P. Girard, «A STUDY OF THE EFFICIENCY OF AN ALTERNATIVE PROGRAMMING PARADIGM TO TEACH THE BASICS OF PROGRAMMING,» *LISI/ENSMA, 86961 Futuroscope Chasseneuil cedex, France*, 2005. [Accès le 1 Avril 2024].
- [6] «What is Java? - Java Programming Language Explained - AWS,» Amazon Web Service, [En ligne]. Available: <https://aws.amazon.com/what-is/java/>. [Accès le 1 avril 2024].
- [7] «Java (langage),» 13 Février 2024, Foundation Wikimédia. [En ligne]. Available: [https://fr.wikipedia.org/wiki/Java_\(langage\)](https://fr.wikipedia.org/wiki/Java_(langage)). [Accès le 1 avril 2024].
- [8] M. s. Chouiti, «Programmation oriente objet classes et objet,» University of Tlemcen, Tlemcen, Algérie, 2020-2021. [Accès le 30 Mars 2024].
- [9] «Programmation orientée objet,» 7 Février 2024, Foundation Wikimédia. [En ligne]. Available: https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet. [Accès le 28 Avril 2024].
- [10] «Programmation Orientée Objet (POO) : tout Savoir,» IA School, 28 Septembre 2023. [En ligne]. Available: [https://www.intelligence-artificielle-school.com/ecole/technologies/programmation-orientee-objet/#:~:text=La%20programmation%20orient%C3%A9e%20objet%20\(POO,et%20la%20maintenance%20du%20code..](https://www.intelligence-artificielle-school.com/ecole/technologies/programmation-orientee-objet/#:~:text=La%20programmation%20orient%C3%A9e%20objet%20(POO,et%20la%20maintenance%20du%20code..) [Accès le 28 Avril 2024].
- [11] «List of object-oriented programming languages,» 23 Août 2023, Foundation Wikimédia. [En ligne]. Available: https://en.wikipedia.org/wiki/List_of_object-oriented_programming_languages. [Accès le 30 Mars 2024].

- [12] «Program analysis,» 16 Mars 2023, Foundation Wikimédia. [En ligne]. Available: https://en.wikipedia.org/wiki/Program_analysis. [Accès le 30 Mars 2024]
- [13] G. Yves, «Utilisation de l'analyse statique comme outil d'aide au développement,» Longueuil, Université de Sherbrooke, Québec, Canada, 2008. [Accès le 2 avril 2024]
- [14] A.-H.-A. Fatima zahrae, C. Fatima-Zahra et E. Zainab, «Technique de Test par Mutation,» Université Ibn Tofail, Academia.edu, Maroc, Kenitra, 2016. [En ligne]. Available: https://www.academia.edu/24959130/Test_mutational. [Accès le 27 avril 2024].
- [15] Checkmarx, «Application Security Testing Tool: Software Security Testing Solutions: Checkmarx,» [En ligne]. Available: <https://checkmarx.com/>. [Accès le 2 Avril 2024].
- [16] SonarQube, «Code Quality Tool & Secure Analysis with SonarQube,» [En ligne]. Available: <https://www.sonarsource.com/products/sonarqube/>. [Accès le 2 Avril 2024].
- [17] «Analyse statique de programmes,» 3 Mars 2024, Foundation Wikimédia. [En ligne]. Available: https://fr.wikipedia.org/wiki/Analyse_statique_de_programmes. [Accès le 2 Avril 2024].
- [18] Coverity, «Coverity Static Analysis Software,» [En ligne]. Available: <https://www.synopsys.com/software-integrity/static-analysis-tools-sast/coverity.html>. [Accès le 2 Avril 2024].
- [19] «FindBugs™ - Find Bugs in Java Programs,» [En ligne]. Available: <https://findbugs.sourceforge.net/>. [Accès le 2 Avril 2024].
- [20] «Findbugs,» [En ligne]. Available: <https://www.wikiwand.com/fr/FindBugs>. [Accès le 2 Avril 2024].
- [21] «Standard Glossary of Software Engineering Terminology(ANSI),» in IEEE Std 610.12-1990, pp. 1-84, 31 Déc. 1990, doi: 10.1109/IEEESTD.1990.101064. [En ligne]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=159342&isnumber=4148>. [Accès le 19 Avril 2024].
- [22] «IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries,» in IEEE std 610, New York, pp. 1-217, 18 Jan. 1990, doi: 10.1109/IEEESTD.1991.106963. [En ligne]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=182763&isnumber=4683>. [Accès le 19 Avril 2024].

- [23] S. Mihaela, «Méthode de test,» UFR d'informatique Paris 7, LIAFA, Paris,175 rue Chevaleret, Bureau 6A7,[En ligne].Available: https://www.irif.fr/~sighirea/cours/methtest/c_test_1.pdf. [Accès le 27 Avril 2024].
- [24] Y. Seladji, «Concevoir des tests grâce aux spécifications,» University of Tlemcen, Tlemcen, Algérie, 2017. [Accès le 20 Avril 2024].
- [25] L. Février, «Analyse automatique de codes informatiques pour l'aide à l'enseignant,» Laboratoire LIP6, 4 place Jussieu, 75005 Paris, France, 2011. [En ligne]. Available: <https://www.academia.edu/download/31632885/actesFinauxAvecCouv.pdf#page=47>. [Accès le 15 Mars 2024]
- [26] Object Management Group, «Welcome to UML web site,» Version 2.5.1. [En ligne]. Available: <https://www.uml.org/>. [Accès le 10 Avril 2024].
- [27] M. S. Tani, «Chapitre 2 : UML : Unified Modeling Language,»University of Tlemcen, Tlemcen, Algérie, 2021.
- [28] M. S. Tani, «Introduction au génie logiciel,»University of Tlemcen, Tlemcen, Algérie, octobre 2021.
- [29] A. S. n. AMRAOUI, «Patrons d'architecture,»University of Tlemcen, Tlemcen, Algérie, 2022-2023. [Accès le 3 Avril 2024].
- [30] HackerRank, «Online Coding Tests and Technical Interviews,» [En ligne]. Available: <https://www.hackerrank.com/>. [Accès le 04 Avril 2024].
- [31] «HackerRank,» 6 Mai 2024, Foundation Wikimédia. [En ligne]. Available: <https://fr.wikipedia.org/wiki/HackerRank>. [Accès le 4 Avril 2024].
- [32] Sphere Research Labs, «Become a true programming master,» November 2012. [En ligne]. Available: <https://www.spoj.com/>.
- [33] HackerEarth, «Helping 7.6M+ developers level up and 1000+ recruiters hire top tech talent.,» [En ligne]. Available: <https://www.hackerearth.com/>. [Accès le 05 Avril 2024].
- [34] CoderByte, «Evaluate Candidates Quickly,Affordably, and Accurately,» 2012. [En ligne]. Available: <https://coderbyte.com/>. [Accès le 06 Avril 2024].
- [35] O. F. Zohra et A. Ikram, «Recherche et sélection d'un outil pour les travaux pratiques du compilation(Mémoire),» University of Tlemcen, TLEMEN, Algérie, 2021. [Accès le 17 Février 2024]

-
- [36] M. Manel et B. Nesrine, «Réalisation d'un site web pour la vérification automatique des solutions du tp de compilation (Mémoire),» University of Tlemcen, TLEMCEN, Algérie, 2023. [Accès le 17 Février 2024].
- [37] T. Raffailac, F. Chavanon et R. Vuill, «Mise en place d'une plateforme de programmation competitive comme support de TDs,» [En ligne]. Available: https://scholar.google.com/scholar?hl=fr&as_sdt=0%2C5&q=Mise+en+place+d%27une+plateforme+de+programmation+comp%C3%A9titive+comme+support+de+TDs&btnG=. [Accès le 17 Février 2024].
- [38] G. Cleuziou, . F. Flouvat, . M. Exbrayat, J. Robert et R. Thion, « *Apprentissage de représentations pour l'enseignement de la programmation : une approche centrée enseignant*, » HAL, 2021. [Accès le 17 Février 2024]
- [39] K. Dari, «Implémentation d'un LMS d'auto évaluation des devoirs de programmation(Mémoire),» University of Tlemcen, Tlemcen, Algérie, 2021. [Accès le 19 Février 2024]
- [40] G. Cleuziou et F. Flouvat, «*Learning student program embeddings using abstract*, » Paris: HAL, 2022, p. 1.
- [41] «Chapitre 3 Diagramme de cas d'utilisation (vue fonctionnelle),» Université Badji Mokhtar-Annaba, Annaba, Algérie. [En ligne]. Available: <https://elearning-facsci.univ-annaba.dz/mod/resource/view.php?id=1092>
- [42] A. Souissi, «conception orientée objet (UML) - Chapitre 2,» Institut supérieur des Etudes Technologiques (ISET-Tunisie), Tunis, Tunisie ,2017. [En ligne]. Available: <https://fr.slideshare.net/slideshow/chap2-diagramme-des-cu/72382901>
- [43] L. Delphine, «UML COUR 5 Diagramme de séquence,» Laboratoire de Recherche Informatique de France (LRI.fr), Paris, 2016-2017.
- [44] A. Chikh, «Object Oriented Analysis and Design (M1GL) classes and Associations,» University of Tlemcen, Tlemcen, Algérie, 2022-2023.
- [45] M. S. Tani, «Diagramme de séquence objet,» University of Tlemcen, Tlemcen, Algérie, 2021.
- [46] M. S. Tani, «Diagramme d'objet,» University of Tlemcen, Tlemcen, Algérie, 2021.
- [47] Modeliosoft, «Modelio Open Source - UML and BPMN free modeling tool,» version 4.1 . [En ligne]. Available: <https://www.modelio.org/index.htm>. [Accès le 15 Avril 2024].
- [48] «List of Unified Modeling Language tools,» 08 - 02 - 2024, Foundation Wikimédia. [En ligne]. Available: https://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools. [Accès le 19 février 2024].
- [49] Sparx Systems, «Outils de modélisation UML pour les métiers, les logiciels, les systèmes et l'architecture,» [En ligne]. Available: <https://sparxsystems.com/>. [Accès le 19 Février 2024].
- [50] Lucid Software Inc., «Lucidchart,» [En ligne]. Available: <https://www.lucidchart.com/pages/>. [Accès le 19 Février 2024].
-

-
- [51] MKLab Co., Ltd., «StarUML,» [En ligne]. Available: <https://staruml.io/>. [Accès le 19 Février 2024].
- [52] MagicDraw, «MagicDraw,» 06 Décembre 2023. [En ligne]. Available: <https://en.wikipedia.org/wiki/MagicDraw>. [Accès le 19 Février 2024].
- [53] «JetUML,» 21 Décembre 2023, Foundation Wikimédia. [En ligne]. Available: <https://en.wikipedia.org/wiki/JetUML>. [Accès le 19 Février 2024].
- [54] Twitter, «Bootstrap · The most popular HTML, CSS, and JS library in the world.,» [En ligne]. Available: <https://getbootstrap.com/>. [Accès le 19 Février 2024].
- [55] Semantic UI, «Semantic UI User Interface is the language of the web,» [En ligne]. Available: <https://semantic-ui.com/>. [Accès le 19 février 2024].
- [56] ZURB,«Foundation: The most advanced responsive front-end framework in the world,» [En ligne]. Available: <https://get.foundation/>. [Accès le 19 Février 2024].
- [57] Facebook, Inc, «React,» [En ligne]. Available: <https://react.dev/>. [Accès le 19 Février 2024].
- [58] Vue.js, «Vue.js - The Progressive JavaScript Framework | Vue.js,» [En ligne]. Available: <https://vuejs.org/>. [Accès le 19 Février 2024].
- [59] Google, «Angular,» [En ligne]. Available: <https://angular.io/>. [Accès le 19 Février 2024].
- [60] T. Otwell, «The PHP Framework For Web Artisans (Laravel),» juin 2011. [En ligne]. Available: <https://laravel.com/>. [Accès le 19 Février 2024].
- [61] Django Software Foundation, «Django Project,» [En ligne]. Available: <https://www.djangoproject.com/>. [Accès le 19 Février 2024].
- [62] Oracle Corporation, «MySQL,» [En ligne]. Available: <https://www.mysql.com/>. [Accès le 19 Février 2024].
- [63] PostgreSQL Global Development Group, «PostgreSQL,» [En ligne]. Available: <https://www.postgresql.org/>. [Accès le 19 Février 2024].
- [64] D. Richard Hipp, «SQLite Home Page,» [En ligne]. Available: <https://www.sqlite.org/>. [Accès le 19 Février 2024].
- [65] MongoDB Inc, «The Developer Data Platform,» [En ligne]. Available: <https://www.mongodb.com/>. [Accès le 15 Avril 2024]
- [66] Microsoft, «Visual Studio Code - Code Editing. Redefined,» [En ligne]. Available: <https://code.visualstudio.com/>. [Accès le 15 Avril 2024].
- [67] J. Skinner, «the sophisticated text editor for code, markup and prose,» [En ligne]. Available: <https://www.sublimetext.com/>. [Accès le 15 Avril 2024].
- [68] D. Ho, «Notepad++: A free source code editor and notepad replacement,» [En ligne]. Available: <https://notepad-plus-plus.org/>. [Accès le 15 Avril 2024].
- [69] Sun Microsystems, «Apache Netbeans,» [En ligne]. Available: <https://netbeans.apache.org/front/main/index.html>. [Accès le 15 Avril 2024].
-

- [70] Laragon, «portable, isolated, fast & powerful universal development environment for PHP, Node.js, Python, Java, Go, Ruby.,» 25 Avril 2019. [En ligne]. Available: <https://laragon.org/index.html>. [Accès le 15 Avril 2024].
- [71] M. Glenford, B. TOM et S. Corey, « The Art of Software Testing, » 1946, -3rd ed. p. cm. [En ligne]. Available: http://tailieuso.udn.vn/bitstream/TTHL_125/9062/3/TheArtOfSoftwareTesting.TT.pdf. [Accès le 24 Mai 2024].
- [72] «Open Web Application Security Project,» 23 Janvier 2024, Fondation Wikimédia. [En ligne]. Available: https://fr.wikipedia.org/wiki/Open_Web_Application_Security_Project. [Accès le 2 Avril 2024].
- [73] «Modèle-Vue-Contrôleur,» 30 Avril 2024, Fondation Wikimédia. [En ligne]. Available: <https://fr.wikipedia.org/wiki/Mod%C3%A8le-vue-contr%C3%B4leur>. [Accès le 8 Mai 2024].
- [74] Royce, W. W. (1970). Managing the Development of Large Software Systems. Proceedings of IEEE WESCON, 1-9. [Accès le 16 juin 2024].
- [75] Boehm B.W., Software Engineering Economics, Englewood Cliffs N.J., Prentice Hall, 1981. [Accès le 16 juin 2024].

Résumé

Notre travail se concentre sur la mise en place d'un système permettant d'évaluer automatiquement les solutions de travaux pratiques soumises par les étudiants. Ce système s'applique dans le cadre du module de programmation orientée-objet. L'évaluation suit les spécifications définies dans la série de TP. Le code source n'est pas examiné directement. Ainsi, elle vise également à réduire la charge de travail pour les enseignants, et orienter plus efficacement les étudiants. Cela étant possible grâce l'analyse en boîte noire des codes des étudiants. En plus de cela, elle permet de les évaluer, sans intervention continuelle de la part de l'assistant durant la séance.

Mots-clés : évaluation automatique ; Analyse de code ; Java ; analyse en boîte noire ; site web ; travaux pratique.

Abstract

Our work focuses on setting up a system to automatically evaluate practical work solutions submitted by students. This system is applied as part of the object-oriented programming module. The assessment follows the specifications defined in the TP series. The source code is not examined directly. Thus, it also aims to reduce the workload for teachers, and to guide students more effectively. This is possible thanks to black-box analysis of student codes. In addition to this, it allows them to be evaluated, without continuous intervention on the part of the assistant during the session.

Keywords: automatic evaluation; Analyze the code; Java; black box analysis; website; practical work.

ملخص

يركز عملنا على إنشاء نظام للتقييم التلقائي لحلول العمل العملية المقدمة من الطلاب. يتم تطبيق هذا النظام كجزء من وحدة . لا يتم فحص كود المصدر مباشرة. وبالتالي، فإنه TP البرمجة الموجهة للكائنات. يتبع التقييم المواصفات المحددة في سلسلة يهدف أيضًا إلى تقليل عبء العمل على المعلمين، وتوجيه الطلاب بشكل أكثر فعالية. وهذا ممكن بفضل تحليل الصندوق الأسود لأكواد الطلاب. بالإضافة إلى ذلك، فإنه يسمح بتقييمها، دون تدخل مستمر من جانب المساعد أثناء الجلسة.

الكلمات المفتاحية: التقييم التلقائي؛ تحليل الكود جافا؛ تحليل الصندوق الأسود؛ موقع إلكتروني؛ أعمال تطبيقية