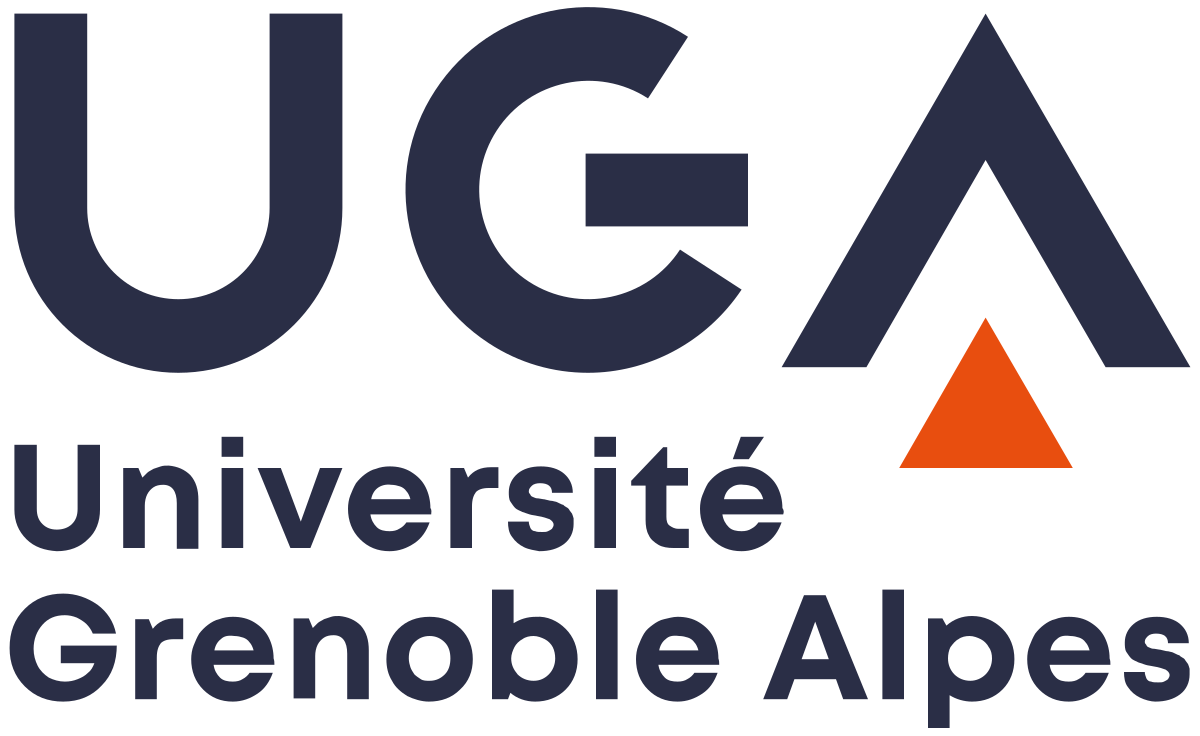
L2 MIASHS - UE Gestion de projet

**Rapport de gestion de projet :**

**Evaluation de plateformes de correction d’exercices de programmation**



<https://www.freepik.com/free-photos-vectors/informatics>

FAURE Lora-Line

MAATA Mohamed Taha

MO’MANTE SOH Micelle-Ange

OUCOUC Hafid

PELLEGRIN Noémie Année universitaire 2019-2020

# Charte anti-plagiat

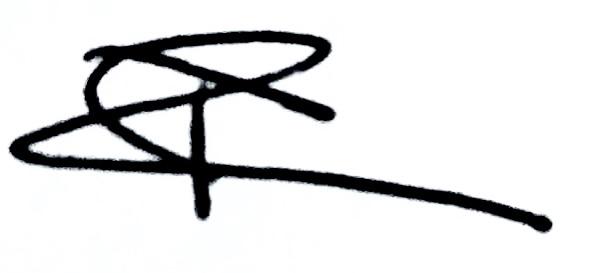
Nous, sous-signé.e.s FAURE Lora-Line, MAATA Mohamed Taha, MO’MANTE SOH Micelle-Ange, OUCOUC Hafid, PELLEGRIN Noémie, avons pris connaissance des obligations décrites dans la charte anti-plagiat, et nous engageons à nous y conformer strictement.

Date et signatures :

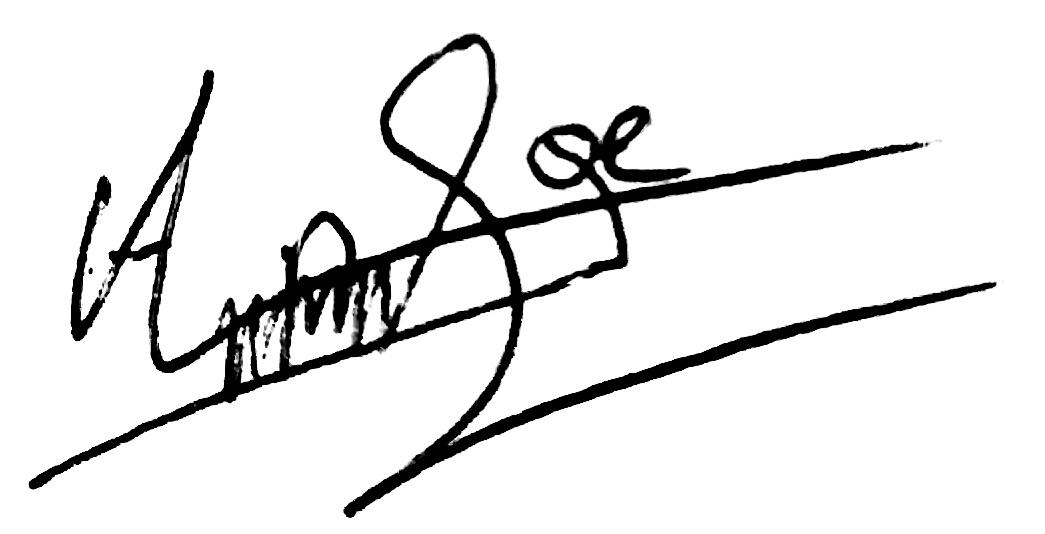
le 12/05/2020



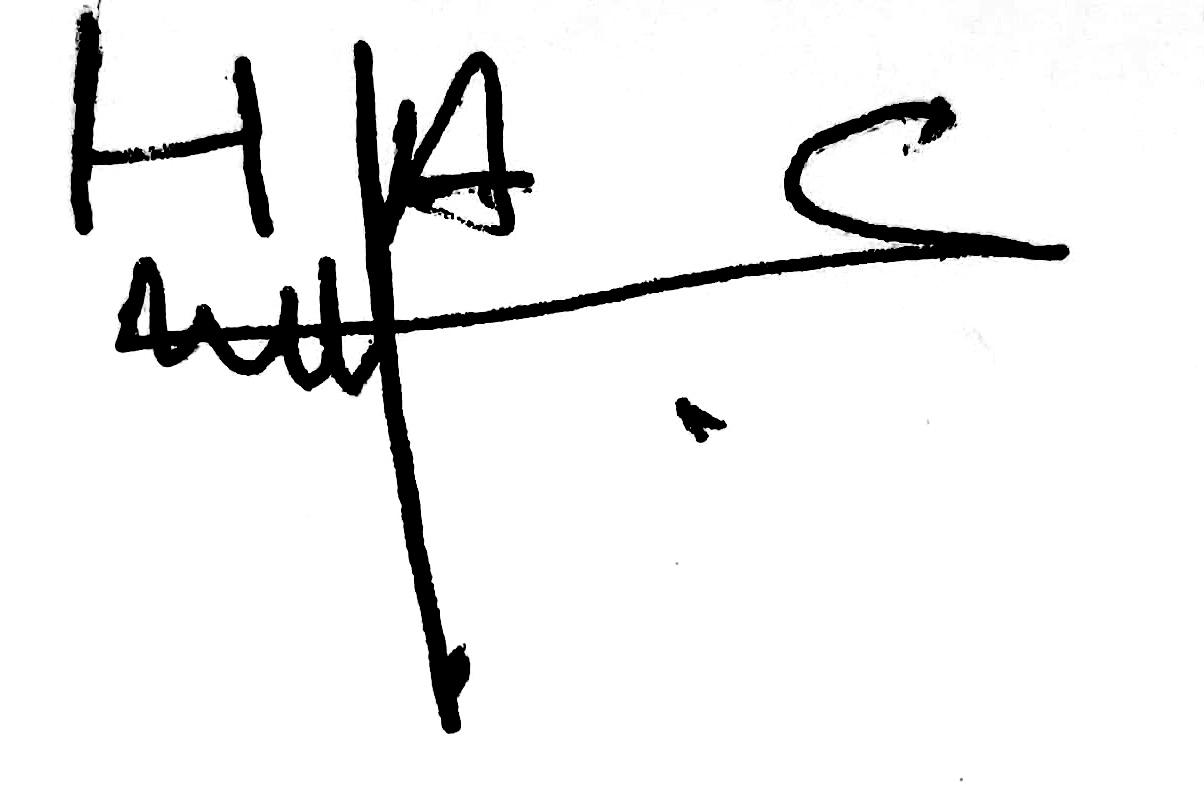
Lora-Line FAURE



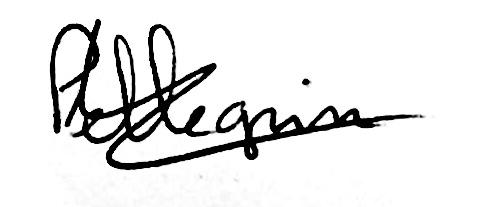
Mohamed Taha MAATA



Micelle-Ange MO’MANTE SOH



Hafid OUCOUC



Noémie PELLEGRIN

Sommaire

**1.** [**Introduction**](#_f3kmw3t6mas8) **3**

1.1. [Contexte du projet](#_2yfzl1om1eod) 3

1.2. [Objectif et étapes intermédiaires](#_83xf21hz4jza) 3

**2.** [**Missions**](#_h54253l2cliw) **4**

2.1. [Démarrage du projet et outils techniques](#_7dft9wt4cspj) 4

2.2. [Organisation du groupe](#_1zt7q4w5vesm) 4

2.3. [Elaboration du cahier des charges](#_9ysjq3abeihq) 5

2.4. [Recherche des solutions existantes et précision de la demande](#_jgw9esozregp) 7

2.5. [Réalisation d’un outil de comparaison des solutions envisagés](#_wg0fv1rn06dy) 7

2.6. [Identification des besoins des étudiants](#_4xevwrncx2w2) 9

2.7. [Analyse des solutions envisagées et hiérarchisation](#_4vusssn2q1ws) 10

**3.** [**Conclusion**](#_brm9b8omn7q9) **13**

[**Bibliographie/webographie**](#_r1r0l8otxyh2) **15**

[**Annexe 1 - Extrait du journal de bord**](#_jz4lzcxb25mg) **16**

[**Annexe 2 - Echéancier**](#_gimx9xo3q7ip) **17**

[**Annexe 3 - Matrice RACI**](#_w9hiw3m7ch7n) **18**

[**Annexe 4 - Cahier des charges**](#_2gj6q468a1q1) **19**

[**Annexe 5 - Grille critériée**](#_r0yc45erovih) **26**

[**Annexe 6 - Questionnaire fourni aux L1 MIASHS**](#_qrrlmavz5cdm) **29**

[**Annexe 7 - Hiérarchie des solutions trouvées**](#_uid0wix1i46s) **30**

# Introduction

## Contexte du projet

Ce projet s’inscrit dans le cadre de l’UE Gestion de projet, réalisé en deuxième année de licence Mathématiques et Informatique Appliquées aux Sciences Humaines et Sociales (MIASHS) à l’Université Grenoble Alpes (UGA), sous la direction des UFR SHS (Sciences Humaines et Sociales) et FEG (Faculté d’Economie de Grenoble). Cet enseignement vise à initier les étudiants au travail en entreprise à travers la réalisation d’un projet et de sa gestion tout au long d’un semestre. Il aide aussi à favoriser la communication entre les différentes parties du projet, que ce soit au sein du groupe à travers un réel travail d'équipe, ou avec le maître d’ouvrage via les multiples réunions organisées. De plus, la mise en place d’une gestion de projet agile permet aux étudiants de s’adapter en permanence et de mettre en place des démarches qui leur serviront plus tard, dans leur vie professionnelle.

Notre groupe de projet est constitué de Lora-Line FAURE, Mohamed Taha MAATA, Micelle-Ange MO’MANTE SOH, Hafid OUCOUC, Noémie PELLEGRIN.

Notre professeur référent est Mr Daniel BARDOU, responsable informatique de la licence MIASHS à l’Université Grenoble Alpes.

## Objectif et étapes intermédiaires

L’objectif du projet est d’analyser les plateformes de correction d’exercices de programmation existantes et de proposer les meilleures d’entre elles au maître d’ouvrage. À plus long terme, cette plateforme permet aux étudiants de L1 MIASHS de s’auto-entraîner sur des exercices de programmation supplémentaires, afin de renforcer ou d’améliorer leurs connaissances dans ce domaine.

Nous avons donc identifié les différentes étapes permettant d’atteindre cet objectif, qui seront développées tout au long du rapport. Ces étapes sont les suivantes :

* Mettre en commun les informations sur la gestion de projet et choisir les outils adaptés.
* Organiser le temps et répartir le travail entre les membres du groupe.
* Elaborer un cahier des charges.
* Rechercher les plateformes et outils déjà existants.
* Déterminer les besoins et attentes des étudiants de L1 MIASHS.
* Créer une grille de comparaison des différentes plateformes.
* Analyser les différents outils trouvés grâce à la grille de comparaison et les hiérarchiser.

# Missions

## Démarrage du projet et outils techniques

Tout d’abord, le travail a consisté à prendre contact avec notre maître d’ouvrage et à mettre en commun les différentes informations obtenues par les membres du groupe, sur la gestion de projet et les méthodes agiles (Manager GO!, 2019). Le choix de mettre en place une gestion agile a été fait rapidement, afin de permettre une constante adaptation aux besoins et attentes du maître d’ouvrage, ainsi que d’éviter une mauvaise gestion du temps en établissant des contraintes temporelles pour chaque étape du projet. Dans ce cadre, des rendez-vous réguliers toutes les deux semaines ont été fixés avec le maître d’ouvrage, pour montrer l’évolution de notre travail et pour échanger sur les points positifs, négatifs ou les lacunes de ce dernier. Ceci nous a permis de répondre au mieux à la demande du maître d’ouvrage et de répartir les ressources disponibles de manière efficace.

Ensuite, plusieurs outils de communication et de travail collaboratif ont été choisis afin de pouvoir rester en contact, travailler et aider facilement chaque membre du groupe. Notre choix s’est porté sur la plateforme Slack car elle nous a paru intéressante pour identifier les différents sujets de discussion du projet, faciliter le partage des ressources pour un thème donné et permettre de communiquer directement avec le groupe. De plus, un dossier partagé à tous les membres sur Google Drive a été créé, parce qu’il nous a semblé être le meilleur moyen de travailler ensemble sur divers documents. L’outil de communication instantanée Messenger a été notre principal outil de communication à distance, permettant de garder le contact et d’instaurer des réunions rapides régulièrement, environ une toutes les deux semaines. Également, un journal (cf. Annexe 1) contenant tous les comptes-rendus de réunion a été rédigé tout au long du projet pour garder une trace des décisions importantes, des informations échangées et du travail à réaliser. Il nous a été bénéfique pour observer l’évolution de notre projet et d’identifier les difficultés rencontrées lors de ce dernier.

## Organisation du groupe

Après les questions d’ordre matériel, celles propres à la gestion du temps et de la répartition du travail entre les membres a été étudiée. Pour gérer au mieux le temps à disposition et le respect de la date limite du 6 mai, nous avons décidé d’élaborer un échéancier sous forme de diagramme de Gantt (cf. Annexe 2). Ce format nous a paru le plus efficace pour représenter de manière claire et adaptée les tâches à réaliser et de comparer facilement l’avancée du travail avec les prévisions faites au début du projet. La création de l’échéancier n’a pas été difficile à réaliser grâce aux expériences précédentes du groupe, mais elle a nécessité la réalisation de plusieurs étapes et un suivi régulier de l’avancée du projet.

Dans ce cadre, une liste des tâches à réaliser a été établie par chaque membre du groupe puis mise en commun, dans le but de définir les différentes étapes à réaliser. Puis, nous avons fait le choix de répartir ces tâches entre les membres du groupe en fonction de l’affinité de chacun, certains préférant travailler plus activement sur l’aspect du projet que sur l’aspect de sa gestion et inversement. Elle nous a paru essentielle afin de favoriser l’expérience préalable de chacun, de ne pas perdre du temps à tous travailler sur une même tâche et de définir un cadre de travail précis. Pour effectuer cette répartition, nous avons élaboré une matrice RACI (Responsable, Acteur, Consulté, Informé) (cf. Annexe 3), qui définit pour chaque tâche les participants actifs ou ceux évaluant et donnant leur avis sur le travail effectué (Bachelet, 2020). Cette dernière n’a pour autant pas été suivie car des changements de tâches sont apparus et chaque personne a choisi à chaque réunion la tâche qu’il souhaitait continuer ou entreprendre. Toutefois, la répartition des tâches allouées à la gestion de projet d’une part, et celles concernant le projet d’autre part ont été effectuées par les mêmes personnes afin que chacun soit l’équivalent d’un expert dans un certain type de tâche. Ainsi, nous avons pu assurer une certaine efficacité dans notre travail et avoir un retour par les pairs intéressant à exploiter car il représentait un point de vue nouveau ou différent.

Enfin, nous avons estimé le temps nécessaire pour chaque tâche et leur ordre chronologique pour mettre en forme l’échéancier. Deux stratégies ont été suivies, l’une consistant à partir de la date du début jusqu’à la fin, l’autre étant la stratégie inverse, dans le but d’observer les périodes cruciales du projet et celles où une certaine marge de manœuvres était présente. Par la suite, l’échéancier a régulièrement été mis à jour, afin de prendre en compte notre avancée réelle et d’évaluer le temps que nous avions à disposition, par rapport aux prévisions faites auparavant. En terme de modifications, l’ajout de nouvelles tâches, la suppression d’autres et le changement de l’ordre dans lequel elles devaient être réalisées ont été les plus importantes. Par exemple, notre idée de départ était de faire tester les plateformes trouvées aux étudiants pour analyser leurs retours et permettre une sélection du meilleur outil à proposer au maître d’ouvrage. Cependant, le manque de temps et la difficulté de mettre en place ceci, nous a fait réfléchir sur une autre manière d’impliquer les étudiants de L1. Ceci a mené à l’élaboration tardive d’un questionnaire, retardant la fin de la création de la grille critériée. Toutefois, toutes les tâches que nous estimons principales ont été réalisées et la date limite fixée a été respectée.

Concernant la gestion du groupe, nous avons fait le choix de ne pas avoir de leader, chaque membre étant acteur au même niveau dans le projet. Néanmoins, un chef de projet a été nommé toutes les deux semaines, afin de rappeler les objectifs à réaliser et les délais fixés par l’échéancier. Son but était d’impliquer tous les membres dans le projet, de faciliter la communication du groupe et avec le professeur référent, d’assurer le maintien des délais et l’écriture des compte-rendus de réunion. Malgré une rotation quasi inexistante de ce rôle entre les membres, les personnes l’ayant assumé ont garanti la continuité du travail de tous les membres, notamment lors de la période de crise sanitaire, qui a rendu plus difficile le maintien de la communication et de la motivation au sein du groupe.

## Elaboration du cahier des charges

L’un des premiers objectifs que nous avions à poursuivre était la création d’un cahier des charges (cf. Annexe 4). Pour commencer, des recherches ont été faites pour comprendre la manière de le construire et les différentes parties qui le composent. Cette étape n’a pas difficile à réaliser au vu des nombreuses informations disponibles sur ce sujet (Bigand, 2013 ; Drack, 2020 ; Granger, 2019). Toutefois, au début du projet, le sujet n’a pas été clairement défini par le maître d’ouvrage et nous n’avions pas identifié précisément ses attentes concernant le travail demandé. En effet, nous avons envisagé de créer une plateforme de correction d’exercice, en s’appuyant sur les outils utilisés par les plateformes déjà existantes. C’est pourquoi, nous avons eu des difficultés à commencer l’écriture du cahier des charges, les objectifs à suivre étant imprécis et inadéquats par la suite.

Les rendez-vous réguliers avec le maître d’ouvrage, nous ont fait comprendre plus en détail sa demande et nous ont amené à redéfinir l’objectif principal à atteindre, consistant en la comparaison de plateformes existantes. Nous avons également fixé de nouveaux objectifs intermédiaires et identifié les cibles visées et les fonctions possédées par ces plateformes. Pour cela, nous avons utilisé des outils tels que le diagramme de la bête à corne ou le diagramme pieuvre.

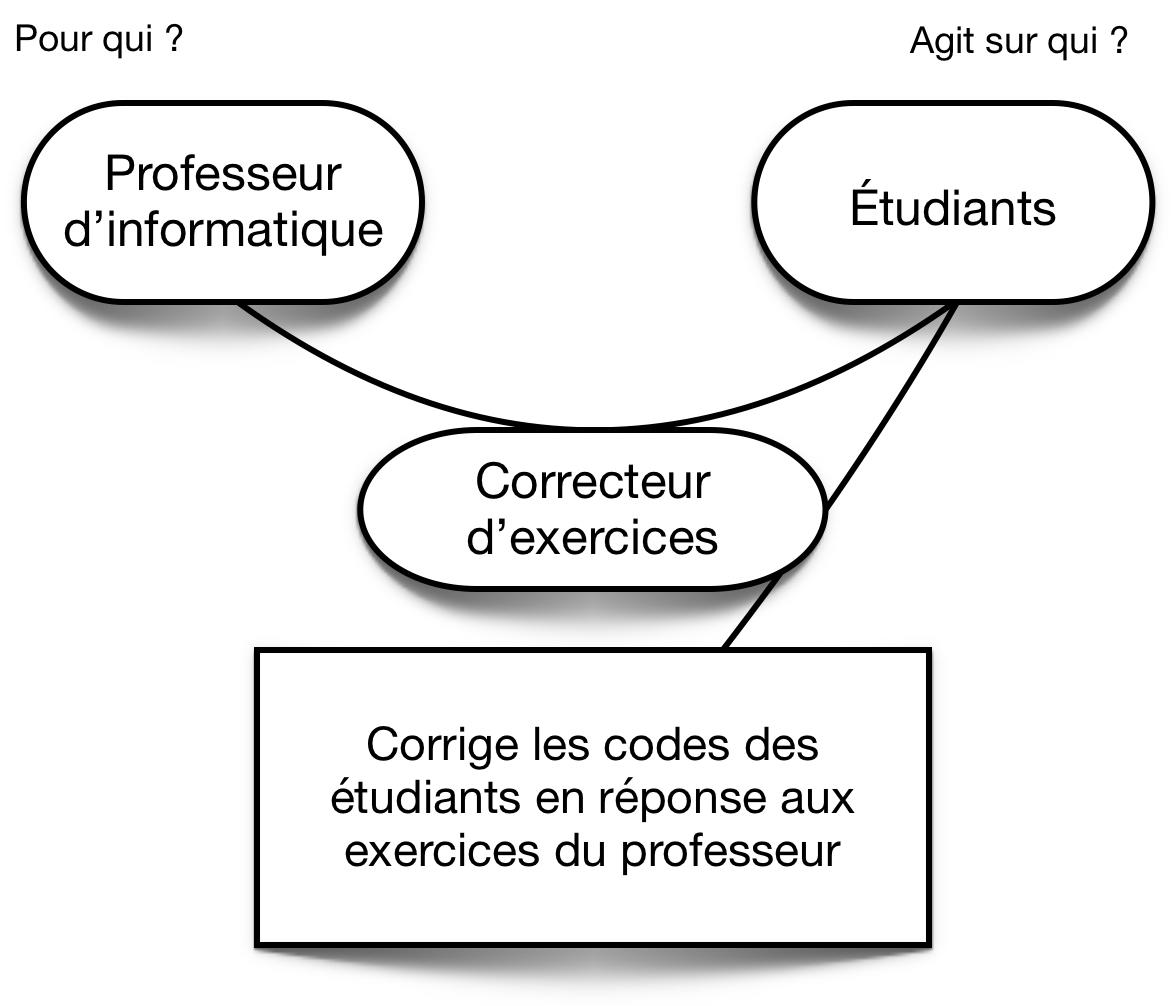


Figure 1 : diagramme bête à cornes

Le premier diagramme (cf. Figure 1) consiste à trouver les besoins auxquels répond l’outil en définissant ses utilisateurs, ce sur quoi il agit et dans quel but il est utilisé. De cette manière, une vision claire des cibles et de leurs besoins est présente, ce qui favorise la recherche de solutions adaptées à ces dernières.

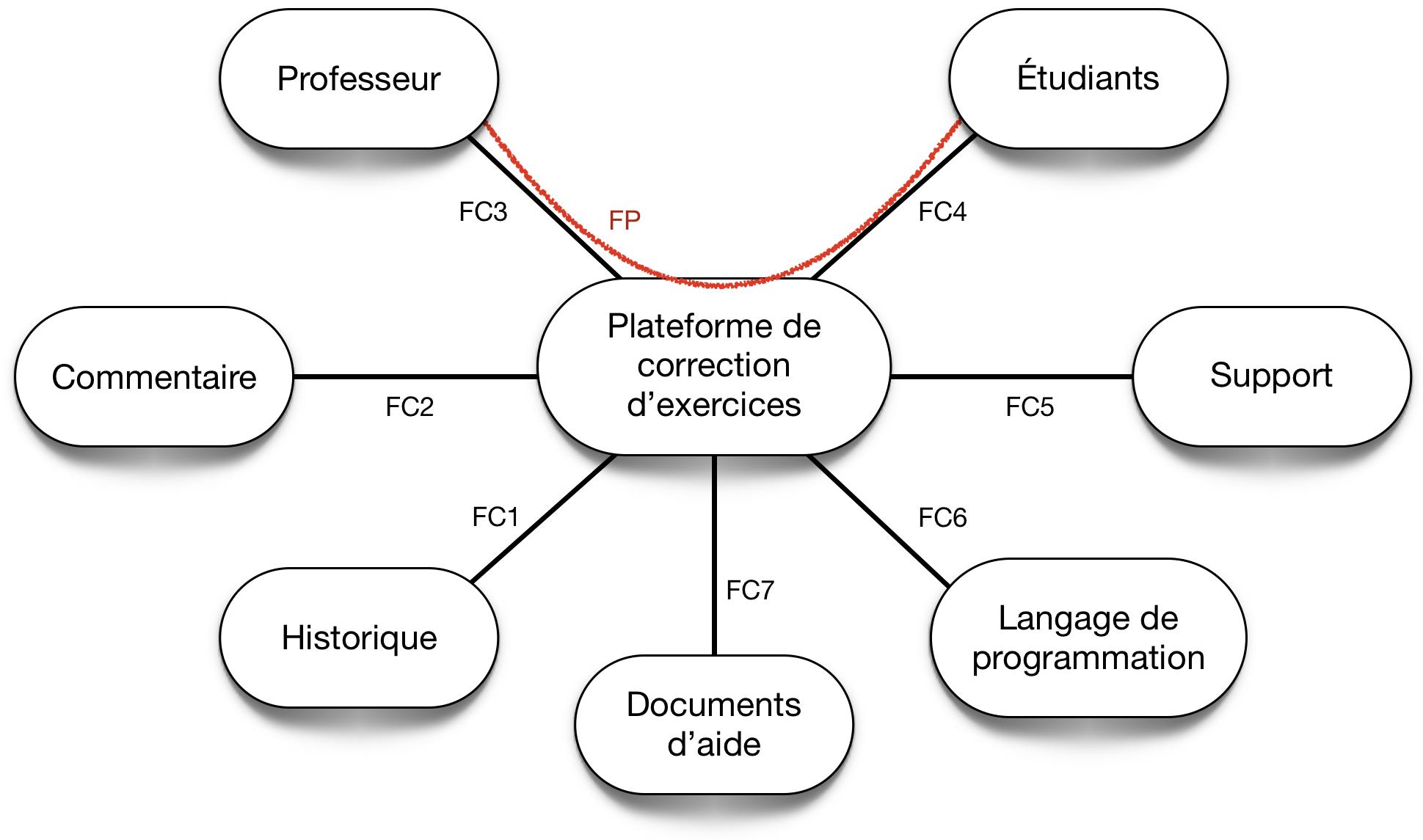


Figure 2 : diagramme pieuvre

Énoncé des fonctions principale et contraintes:

| FP | Permettre au professeur de donner des exercices aux élèves |
| --- | --- |
| FC1 | Proposer un historique des solutions |
| FC2 | Pouvoir poster un commentaire sous une solution |
| FC3 | Les solutions postées par les étudiants sont consultables par le professeur |
| FC4 | L’étudiant peut rédiger sa correction sur la plateforme |
| FC5 | Être accessible via plusieurs outils technologiques et numériques pour permettre à tous les étudiants d’y avoir accès |
| FC6 | Posséder les langages de programmation correspondent à ceux du cours |
| FC7 | Proposer des documents d’aide avec des rappels de cours |

Le deuxième diagramme (cf. Figure 2) permet d’exprimer les différentes fonctions reliant directement l’outil à l’utilisateur ou bien reliant deux acteurs à travers son utilisation (Bigand, 2013). Nous avons pu grâce à lui, identifier la majorité des fonctionnalités, commencer à entrevoir la manière de les décrire et certaines contraintes associées à leur utilisation.

Nous nous sommes aussi appuyés sur les recherches effectuées lors de la première semaine et l’expérience de certains membres du groupe sur certaines plateformes pour comprendre quelles étaient les fonctionnalités principales de ces plateformes. Cependant, nous avons rencontré quelques difficultés car nous n’avions pas pris en compte les fonctions dont les étudiants voulaient bénéficier en utilisant une telle plateforme. Ceci a contribué à retarder la description complète des fonctionnalités à prendre en compte. De plus, lors de nos échanges avec le professeur référent, il nous est apparu que la description des différentes fonctions était peu compréhensible et facile. Ainsi, nous avons décidé de rajouter des illustrations de tests, réalisés par les membres du groupe lors de l’analyse des différentes plateformes, accompagné d’un texte explicatif.

Enfin, les ressources techniques, contraintes et livrables ont été facilement définis et présentés. Ces parties ont néanmoins été revues ou complétées au cours du temps, pour s’adapter au mieux aux objectifs visés et à l’évolution du travail. Notamment, nous avons choisi de rassembler les différentes contraintes ensemble et d’ajouter une partie concernant le budget futur éventuellement requis par certaines plateformes.

## Recherche des solutions existantes et précision de la demande

Avant le début du travail, nous avons été confrontés à une difficulté : la non attribution du sujet à notre première rencontre avec le maître d'ouvrage qui nous a demandé de chercher un sujet par nous même, une recherche qui n'a pas été facile. Ensuite, le maître d'ouvrage nous a attribué un thème restant cependant vague et mal compris par nous. La tâche assignée par le professeur a été d’analyser cette demande en nous donnant comme point de départ de “trouver les outils déjà existants qui permettent de mettre en place puis de corriger des exercices de programmation”.

La première étape conduisant à la réalisation de cette tâche s’est reposée sur des recherches webographiques individuelles. Chaque membre du groupe a été amené à trouver au moins deux plateformes (site web, application) répondant à la demande du professeur et à les analyser. Le but de l'analyse était de faire une brève description de la plateforme, de donner ses avantages et ses inconvénients en termes de langages supportés, de travail de préparation vis à vis de l’enseignant, du prix, etc. Puis, nous avons mis en commun toutes les solutions trouvées sous forme de grille, avant de les remettre à notre professeur référent. Il a pu ainsi les commenter et nous donner les grandes lignes du projet durant notre 2eme entretien.

La deuxième phase a consisté à trouver les outils ou les méthodes utilisés par les plateformes que nous avions trouvé pour évaluer et corriger les exercices. Cette tâche a été difficile car nous avons accentué nos recherches dans des outils de création de plateformes de correction automatique, ce qui demandait un niveau de connaissances parfois trop élevé en informatique. Les différentes solutions ont été envoyées au maître d’ouvrage pour avoir son analyse à la prochaine réunion. Au final, son retour nous a permis de nous rendre compte que nos recherches n’étaient pas été bien centrées et que notre analyse était quelques fois erronée.

A la suite du recadrage de notre maître d'ouvrage, la demande était plus claire et nous avons décidé de réaliser une analyse approfondie des plateformes trouvées. Celle-ci permettant de réaliser la tâche qui nous a été attribuée, consistant à comparer les différentes plateformes d’exercice de programmation existantes. Cette analyse s’est faite sous forme de grille en fonction de critères définis au cours de nos recherches et de nos échanges avec le professeur référent. Tout au long du projet, les résultats de nos différentes recherches sont mis en commun, puis envoyés au professeur référent avant chaque rendez vous pour qu'il les analyse et nous donne son point de vue lors notre réunion.

## Réalisation d’un outil de comparaison des solutions envisagés

Afin d’évaluer les différentes plateformes trouvées, l’idée de réaliser une grille de comparaison est apparue très rapidement. Nous avons fait le choix de créer une grille critériée car certains membres du groupe avaient déjà une certaine expérience et des ressources sur lesquelles s’appuyer pour réaliser cet outil, grâce aux projets réalisés lors des deux semestres de cette année universitaire (Tardif, 2013). En outre, il paraît être le plus efficace pour effectuer un comparatif entre plusieurs solutions car il permet d’analyser la pertinence de chacune et de les comparer à l’aide d’un outil de travail commun. Le maître d’ouvrage avait également suggéré la création d’un format de grille multi-critères possédant plusieurs niveaux avec des descriptions détaillées pour chacun d’eux. La création de la grille nous a aussi permis d’identifier clairement et précisément les besoins des cibles visées, ce qui assure une certaine pertinence de l’analyse des solutions les plus adaptées.

Les différents critères pris en compte ont d’abord été choisis en fonction de la demande du maître d’ouvrage. Plusieurs fonctions essentielles de la plateforme sont apparues au cours des rendez-vous réguliers avec celui-ci. Notamment, les fonctions suivantes : la facilité de mise à disposition des exercices, la possibilité d’avoir accès à un historique des solutions proposées, les langages de programmation pris en compte ou encore l’accessibilité de la plateforme. Cependant, les attentes des étudiants n’avaient pas été considérées comme pertinentes au début du projet car nous supposions que le professeur faisait cette demande en ayant connaissance de ces dernières. De ce fait, la première grille proposée ne répondait pas entièrement aux différents usages possibles des plateformes d’exercices.

Après plusieurs semaines et un rendez-vous avec le maître d’ouvrage, nous avons choisi d’établir un questionnaire pour connaître les fonctionnalités auxquelles les étudiants voulaient avoir accès sur ce type de plateforme. Les résultats, présentés dans la partie 2.6., nous ont donc fourni les autres critères nécessaires à la création de la grille. Enfin, d’autres échanges avec le maître d’ouvrage nous ont permis de finaliser le choix de tous les critères.

Ensuite, le choix du nombre de niveau s’est fait naturellement, en considérant qu’un nombre inférieur n’aurait pas eu un grand intérêt car il ne permettait pas de comparer efficacement la pertinence des sites entre eux. De plus, un nombre supérieur de niveaux était envisageable mais aurait nécessité de distinguer plus précisément les différents niveaux, ce qui nous paraissait difficile à mettre en place pour une majorité des critères. Par exemple, c’est le cas de la langue utilisée par la plateforme ou de son attractivité.

Concernant la description détaillée des critères, nous nous sommes appuyés sur l’ensemble des solutions déjà trouvées pour permettre d’identifier quels éléments étaient mesurables quantitativement ou qualitativement, afin de situer le niveau atteint par chaque solution. Par exemple, le critère sur la saisie du code a été construit en analysant les présentations des exercices sur les différentes plateforme lors de leur utilisation à la manière d’un étudiant. En outre, la description des niveaux a aussi évoluée au cours du temps à l’aide des retours faits par le maître d’ouvrage, notamment sur la langue utilisée sur la plateforme. Ce dernier estimant que le français 0n’était pas une absolue nécessité.

Enfin, les dernières modifications mineures effectuées sur la grille ont été mises en place lors de son utilisation effective pour analyser les sites. Certains niveaux ne fournissaient pas une description assez claire ou détaillée de la description d’une fonctionnalité. Par exemple, le critère d’utilisation de la plateforme varie beaucoup selon les plateformes, certaines étant accessibles à tout public et proposant directement des exercices, alors que d’autres sont uniquement développées pour un public universitaire et assurent au professeur de poster ses exercices directement sur la plateforme. La grille critériée finale utilisée pour l’analyse est présentée dans l’annexe 5.

## Identification des besoins des étudiants

Ce projet est destiné à deux cibles, les professeurs d’informatique, ainsi que les étudiants de L1 MIASHS. Son but est d’aider ces derniers à faciliter l’apprentissage des notions du cours et des stratégies à mettre en place lors de la programmation dans un langage de programmation donné. Pour cela, la plateforme doit disposer de diverses caractéristiques telles que :

* la présence d’un forum de discussion pour demander et recevoir l’aide des autres étudiants;
* la possibilité d’ajouter un commentaire sur des solutions proposées précédemment par d’autres étudiants;
* un système permettant de lancer des défis/concours entre étudiants;
* l’éventualité de recevoir une note comme évaluation pour avoir donné une solution à un exercice.

Cependant, certaines sont plus importantes que d’autres, selon l’objectif poursuivi par les utilisateurs de ces plateformes. C’est pour cette raison que nous avons décidé de demander directement aux L1 MIASHS les fonctionnalités auxquelles ils aimeraient avoir accès en utilisant une telle plateforme. Nous avons créé un questionnaire succinct (cf. Annexe 6), afin de recueillir le plus de participations possibles. Il comporte une courte introduction qui présente les grandes lignes de notre projet et deux questions. La première vise à vérifier l’intérêt des étudiants dans la mise en place d’une telle plateforme et la seconde leur demande de sélectionner au maximum trois fonctionnalités parmi les 10 proposées, qui leur semblent être les plus importantes. Cette seconde question nous a permis d’avoir une idée concernant les aspects les plus importants à prendre en compte lors nos recherches ultérieures et déterminer les critères de sélection de la grille d’analyse, ainsi que leur importance respective.

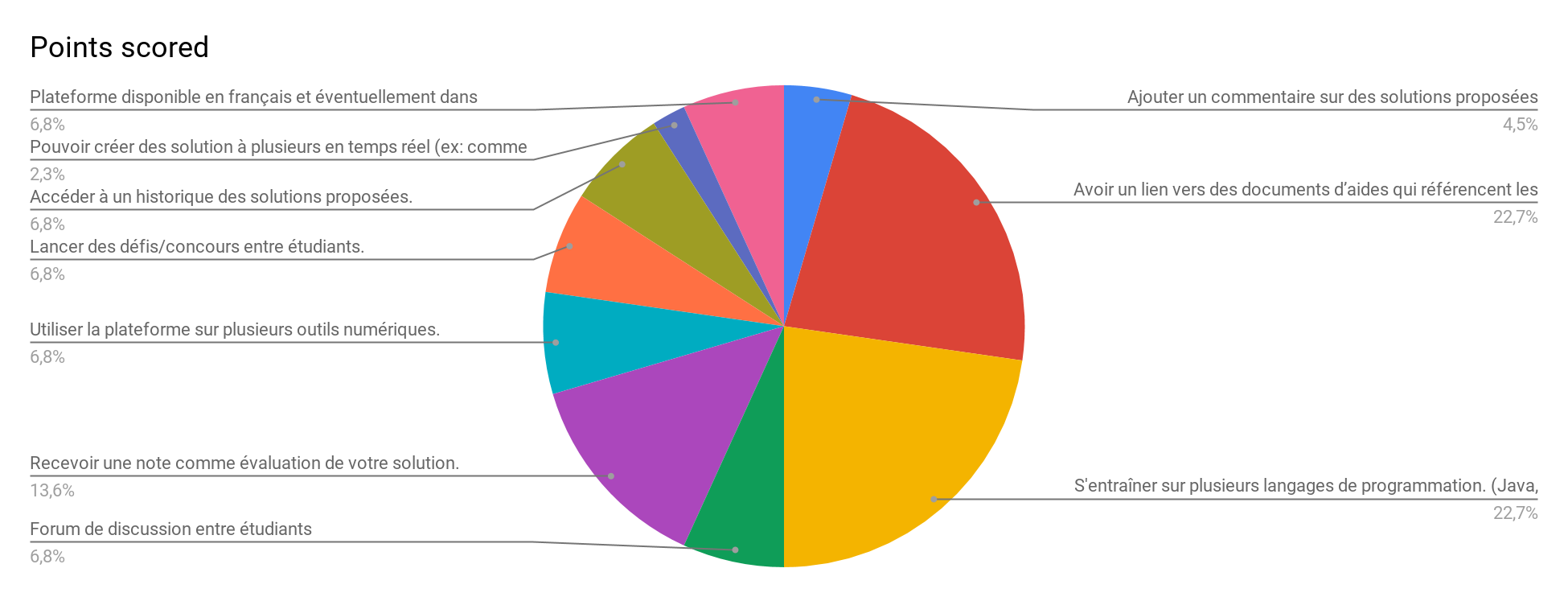
En ce qui concerne la mise à disposition de ce questionnaire, il a été publié sur la page Facebook des L1 MIASHS Grenoble 2019-2020 le 20/03/2020, ouvert pendant 2 semaines. Les résultats sont visibles sur le diagramme ci-dessous (voir Figure 2), qui présente les fonctionnalités proposées et leur pourcentage de sélection par les étudiants ayant répondu au sondage. 

Figure 2: Diagramme de résultats du questionnaire soumis aux L1 MIASHS.

A l’issue des deux semaines nous avons recueilli 16 réponses, soit moins de 15% de la promotion des L1. Nous pouvons attribuer ce faible taux de réponse au manque de relance de la part du groupe pour inciter les étudiants à participer. De plus, nous n’avons pas pris en compte les personnes n’étant pas inscrites dans le groupe Facebook de la promotion, ce qui a en partie réduit la visibilité de notre questionnaire à tous le public concerné. Afin d’avoir plus de réponses, il aurait été intéressant de prendre contact avec le secrétariat de la licence MIASHS, pour demander l’envoi d’un mail à tous les étudiants de L1 ou bien de mettre le lien vers le questionnaire sur leur groupe de discussion Discord. Nos résultats ne sont donc pas réellement représentatifs des attentes de l’ensemble des étudiants, mais ces réponses nous ont permis de dégager quelques fonctionnalités majeures.

Deux choix se démarquent des autres, la présence de documents d’aide et la possibilité de s’entraîner sur plusieurs langages. Ainsi, nous estimons que ces deux caractéristiques doivent être respectées par toutes les solutions que nous avons trouvées, ce qui est effectivement le cas mais de manière plus ou moins efficace et intuitive. En revanche, dans la situation où une telle plateforme serait utilisée en cours d’informatique, les langages utilisés resteraient ceux de la licence (Java et Scheme en L1). On remarque aussi que plus de 13% des réponses sont en faveur de l’obtention d’une note pour certains exercices effectués en ligne. Nous considérons donc ce critère comme optionnel mais qui peut être un atout.

## Analyse des solutions envisagées et hiérarchisation

Lors de cette étude de l'existant, nous avons soumis à évaluation sept sites internet et deux applications mobiles. Nous avons aussi décrit rapidement les autres solutions techniques envisagées. Les plateformes choisies pour cette analyse proviennent de nos expériences personnelles précédentes en tant qu’étudiant et d’une recherche spécifique à ce projet, plus centrée sur les critères qui ont été considérés comme pertinents.

Nous avons débuté nos analyses par une description succincte des caractéristiques de chaque plateforme, afin d’avoir une idée générale de ce en quoi elles consistent. Puis, nous avons développé et élargi notre base d’informations concernant ces plateformes et avons orienté nos efforts sur les sites et applications car en ce qui concerne les solutions techniques, elles nous apparaissent pour la plupart irréalistes ou irréalisables. Parmi ces dernières, seule la solution proposant la création d’un site web complet avec des cours et énoncés d’exercices dans lequel serait intégré un compilateur sous forme d’API (Application Programming Interface) est satisfaisante. Mais elle dépasse nos compétences et connaissances en informatique.

Pour comprendre et faire un rapport des fonctionnalités et caractéristiques spécifiques à chaque site et application, nous avons dû les tester nous même. Cette tâche a été facilitée par le fait que certains membres du groupe ont déjà eu recours à plusieurs de ces solutions dans un cadre scolaire ou personnel afin de s’entraîner à programmer, et sont donc familiers avec leurs particularités.

Au final, après discussion avec le maître d’ouvrage, notre choix de présentation de cette analyse s'est porté sur une description des solutions sous forme de paragraphes, suivi de leur évaluation à l’aide de la grille critériée (cf. livrable). Les paragraphes permettent de décrire avec plus de liberté les fonctionnalités, pour les étudiants et le professeur, présentes dans la grille d’analyse, tout en ajoutant les éléments qui nous semblent pertinents mais qui n’entrent pas dans les critères d’analyse et de sélection. La grille, quant à elle, permet une présentation plus concise et organisée des points forts et points faibles de chaque solution. Ceci permet de faciliter la lecture des analyses faites et d’identifier les sites les plus adéquats. De plus, cette analyse nous permet d’observer que, parmi notre échantillon de plateformes, toutes ont des avantages qui les démarquent des autres. Notamment, l’application Swift Playgrounds qui est la plus attractive pour les débutants, mais qui ne répond à quasiment aucun autre critère. Nous l’avons tout de même ajoutée à notre étude car elle donne un exemple d’interface attractive et une façon de coder facilitée et originale.

Selon les critères et nos appréciations, nous avons réalisé un classement des 9 solutions, afin de présenter clairement les plateformes qui paraissent les plus adaptées à la demande du maître d’ouvrage (cf. Annexe 7). Néanmoins, l’ordre dans le classement final ne correspond pas expressément au nombre de critères par plateforme qui correspondent aux catégories bien, passable ou mauvais, il est aussi influencé par l’importance de ces différents critères. Nous estimons que certaines fonctionnalités sont indispensables à l’usage d’une plateforme par rapport à d’autres plus superflues, car le but de ces plateformes reste avant tout l’apprentissage de langages de programmation. Par exemple, nous pensons que la possibilité de proposer des exercices a un poids supérieur à la mise à disposition de la plateforme. De plus, le classement prend en compte les critères paraissant importants aux yeux des cibles visées, le maître d’ouvrage ainsi que les étudiants. Pour autant, notre classement est seulement présenté à titre indicatif, le choix de la plateforme finale revenant au professeur.

Pour conclure, les deux plateformes nous paraissant les plus adaptées à la demande du projet sont Qualified et Virtual Programming Lab, bien que nous n’ayons pas eu l’occasion de tester nous-mêmes Qualified. En effet, cette plateforme est payante et notre demande de test est restée sans réponse. Toutefois, la description qui en est faite est suffisamment complète pour l’analyser selon nos critères. Nous comprenons que cette plateforme permet de résoudre les exercices proposés par le professeur comme dans le compilateur Éclipse (ou autre logiciel de programmation selon les langages), tout en lui offrant la possibilité de voir le processus qui a mené à telle ou telle solution pour chaque étudiant. Virtual Programming Lab correspond aussi très bien aux attentes du professeur et des élèves, notamment via des fonctionnalités similaires à celles de Qualified. Elle se démarque surtout au niveau de sa mise à disposition facile car c’est la seule plateforme qui peut être installée directement sur moodle. En ce qui concerne les plateformes ne permettant pas de proposer des exercices, France IOI donne tout de même un bon exemple de plateforme de programmation en ligne performante et fonctionnelle.

# Conclusion

Tout d’abord, à l’aide de cet enseignement, nous avons pu découvrir en quoi consistait la gestion de projet, ses différentes étapes, ses difficultés, ainsi que de connaître et appliquer les méthodes agiles. Nous avons eu une première expérience positive à exploiter plus tard dans un cadre universitaire ou professionnel. De notre point de vue, le bilan général du travail réalisé, tant au niveau de la gestion de projet que du projet, est plutôt positif. Notamment, nous pensons avoir essayé de mettre en place une stratégie agile de gestion de projet et avoir atteint l’objectif du projet, au travers des différentes étapes accomplies.

Concernant la gestion de projet, les différents outils techniques, nous ont permis d’établir une communication simple, fluide et relativement régulière au sein du groupe. De plus, ils ont garanti une visibilité sur le travail fourni par chacun et ont facilité l’intervention à différentes échelles de tous les membres. L’expérience personnelle de chaque membre a contribué à une répartition du travail adaptée et un échange sur les méthodes de travail à mettre en place. De plus, le rôle de chef de projet, même s’il n’a pas fonctionné de manière optimale, a renforcé l’investissement et la motivation de tous les membres impliqués dans le projet.

Ensuite, nous avons appris à élaborer des outils de gestion de projet tels que le diagramme de Gantt ou la matrice RACI, et à les exploiter. Cependant, le second n’a servi que peu et a constitué une charge de travail supplémentaire. Quelques difficultés sont aussi apparues dans notre gestion du temps. La mauvaise identification des tâches et leur agencement dans l’échéancier nous ont légèrement retardé par rapport aux délais prévus. Toutefois, nous avons réussi à nous adapter en allouant pour chaque tâche le temps dont nous disposions, afin de toutes les mener à terme, sans compromettre la qualité globale de notre travail.

La création du cahier des charges a été longue et compliquée à cause d’une mauvaise considération des objectifs, des cibles et de leurs attentes. Toutefois, les échanges réguliers avec le maître d’ouvrage et la consultation des étudiants de L1 nous ont été bénéfiques pour élaborer un cahier des charges clair et adapté.

Enfin, d’un point de vue fonctionnel, nous n’avons pas pu fournir rapidement une grille d’évaluation réellement fonctionnelle, compte tenu des changements d’objectifs. Néanmoins, nous avons pu démontrer notre avancée régulière en présentant les différentes solutions trouvées et les idées ou propositions de critères à insérer dans notre grille.

Concernant le projet, nous avons rendu une évaluation et une comparaison des différentes plateformes d’exercices de programmation existantes. Notre difficulté majeure a été la recherche des critères pertinents dans notre grille, en considérant d’un côté l’utilisation du professeur et d’un autre celle des élèves. Néanmoins, à l’aide des tests de différents sites trouvés au préalable, nous avons pu leur proposer de choisir parmi certaines fonctionnalités, celles paraissant essentielles à l’usage d’une telle plateforme. Le faible taux de réponses, ne nous permet pas de savoir précisément si les fonctionnalités que nous avons considérées sont réellement celles principales. Toutefois, elles nous ont fourni une base de travail exploitable pour la création de notre grille et l’analyse des plateformes. De plus, nous nous sommes appuyés sur notre expérience d’étudiant pour estimer si les plateformes testées nous paraissait facile d’utilisation et adaptées à des étudiants. Le détail des différents niveaux a été réalisé sans trop de difficultés.

Enfin, le livrable final donné au maître d’ouvrage a permis de répondre à l’objectif principal, de manière détaillée et précise à l’aide de la description des sites, tout en résumant les informations trouvées dans la grille, aidant le professeur dans son choix d’outil. Nous lui avons également proposé une hiérarchie des solutions pour faire un bilan simple de toutes les solutions dont il dispose.

Certains compléments auraient pu être ajoutés dans notre livrable, notamment en augmentant le nombre de niveaux pour certains critères ou en proposant une notice d’installation de la meilleure solution dont nous disposons. De plus, un pré-test auprès des étudiants de L1 aurait également été bénéfique afin d’avoir un feedback sur leur utilisation réelle. Une dernière amélioration aurait pu être la prise en compte des étudiants de L2 et L3, afin d’avoir un panel plus large de réponses et de solliciter leur expérience en tant qu’étudiant. Par exemple, ils pourraient apporter des informations sur d’éventuelles lacunes ou points faibles dans les connaissances et compétences apprises, engendrant des difficultés d’apprentissage dans la suite de leur cursus d’informatique. Un éventuel projet futur pourra peut-être se consacrer à la mise en place et la gestion d’une telle plateforme et de l’évaluation de ses effets positifs pour les étudiants.

# 

# Bibliographie/webographie

Bachelet, R. (2019, septembre 11). L’essentiel des outils d’organisation de projet. Consulté le 28 avril 2020, à l’adresse <https://gestiondeprojet.pm/bases-organisation-projet/>

Bachelet, R. (2020, janvier 13). Conception et planification de projet : outils avancés. Consulté le 28 avril 2020, à l’adresse <https://gestiondeprojet.pm/planification-projet/>

Bigand, M. (2013, juin 16). Cours d’analyse fonctionnelle, établissement du cahier des charges – MOOC Gestion de Projet. Consulté le 28 avril 2020, à l’adresse <https://gestiondeprojet.pm/analyse-fonctionnelle/>

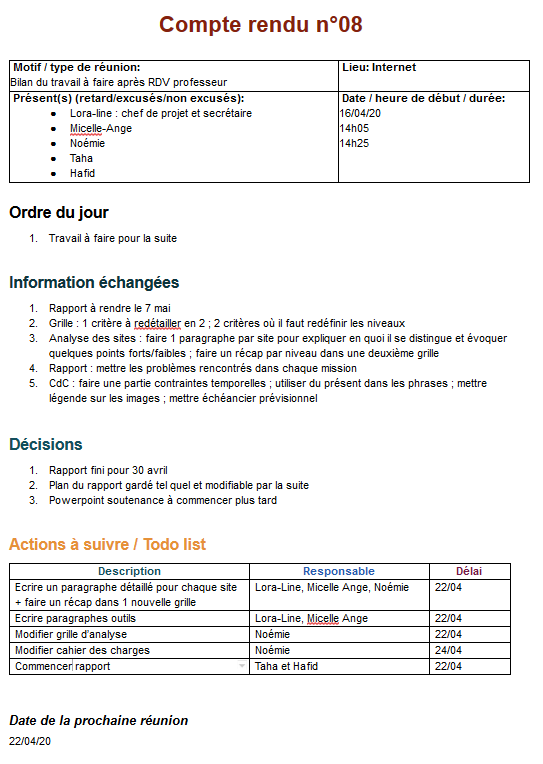
Drack, A. (2020, février 18). Cahier des charges : définition, étapes et exemples, on vous dit tout. Consulté le 28 avril 2020, à l’adresse <https://www.appvizer.fr/magazine/operations/gestion-de-projet/cahier-des-charges#etape-n-1-l-analyse-fonctionnelle>

Granger, L. (2019, novembre 22). Exemple de cahier des charges pour un projet simple. Consulté le 28 avril 2020, à l’adresse <https://www.manager-go.com/gestion-de-projet/dossiers-methodes/elaborer-un-cdc>

Manager GO! (2019, octobre 25). Méthodes Agiles dans les projets : l’essentiel + illustration Scrum. Consulté le 28 avril 2020, à l’adresse <https://www.manager-go.com/gestion-de-projet/methodes-agiles.htm>

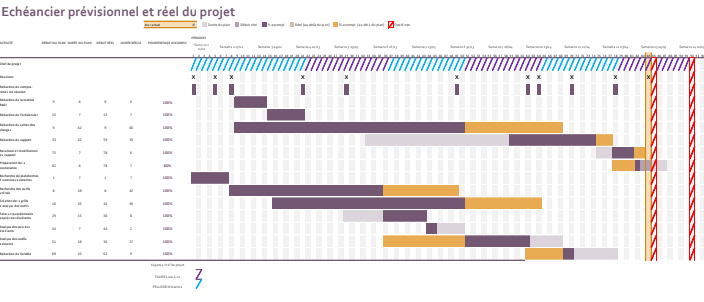
Tardif, J., & Côté, R. (2013, juillet 15). Élaboration d’une grille d’évaluation. Consulté le 28 avril 2020, à l’adresse <http://pedagogie.uquebec.ca/outils/elaboration-dune-grille-devaluation>

# Annexe 1 - Extrait du journal de bord

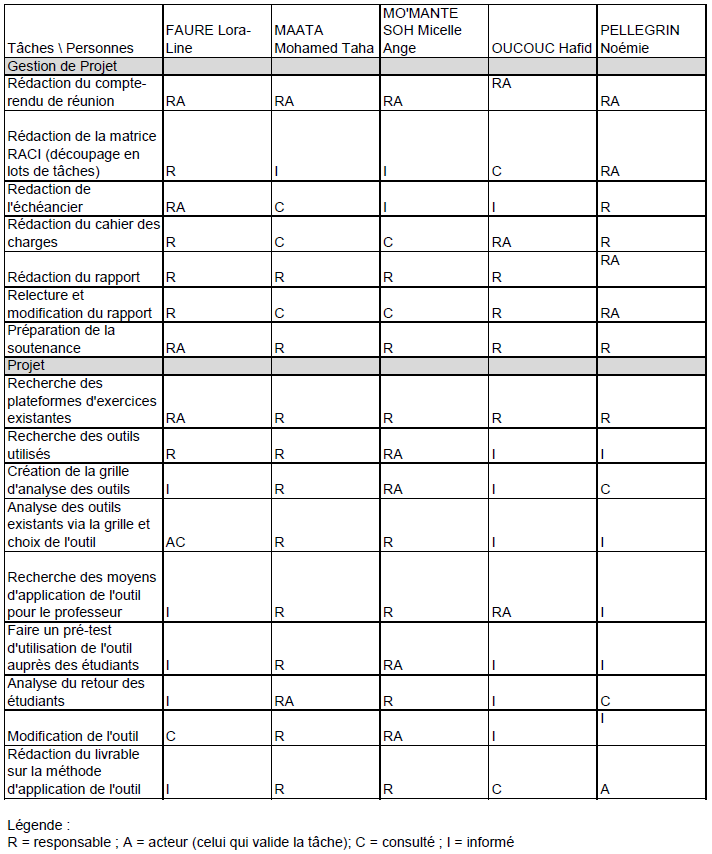


Format du compte-rendu de réunion provenant du site de Bachelet (2019).

# Annexe 2 - Echéancier



# Annexe 3 - Matrice RACI



# 

# Annexe 4 - Cahier des charges



L2 MIASHS - Année universitaire 2019/2020

**UE Gestion de projet :**

**Cahier des charges**

FAURE Lora-Line, MAATA Mohamed Taha, MO’MANTE SOH Micelle-Ange, OUCOUC Hafid, PELLEGRIN Noémie

# Sommaire

**1.** [**Présentation du projet**](#_ht15git33l5x) **2**

1.1 [Contexte et objectif général](#_vnhm3vbtxq79) 2

1.2 [Objectifs](#_pgi7xeo7e8mw) 2

1.3 [Périmètre](#_iv4yhe708ulj) 2

1.4 [Cibles](#_yrlt2osbvqer) 2

**2.** [**Description fonctionnelle et technique**](#_roklallshfqa) **3**

2.1 [Fonctionnalités](#_rj7yyp8zrc0e) 3

2.2 [Ressources techniques](#_8103pzjc5bpa) 4

2.3 [Contraintes](#_jl5ug9j9z439) 4

**3.** [**Organisation du projet et livrables**](#_p4yke76wst1r) **5**

3.1 [Livrables du projet](#_rvpjlvl356l) 5

3.2 [Budget](#_5jodz983r9l9) 5

3.3 [Délais](#_ffg8syjjjbcs) 5

[**Annexe**](#_ot10uyfydno) **- Echéancier prévisionnel 6**

# 

# Présentation du projet

## Contexte et objectif général

Ce projet s’inscrit dans le cadre de l’UE Gestion de projet, réalisé en deuxième année de licence MIASHS. Celui-ci consiste en la recherche d’un outil d'entraînement aux langages de programmation. Cette demande a été faite par un professeur d’informatique, afin d’enrichir l’enseignement qu’il propose à ses étudiants. Cet outil permet à ces derniers de s’auto-entraîner sur des exercices supplémentaires et d’approfondir des notions abordées en classe.

## Objectifs

Plusieurs objectifs plus spécifiques sont alors poursuivis lors de ce projet :

* Rechercher sur des moteurs de recherches, magasin d’applications, sites spécialisés et projets informatiques, 5 à 10 outils d’entraînement aux langages de programmation.
* Déterminer les 2 à 3 besoins principaux des étudiants de L1 MIASHS, concernant l’usage d’un outil d’aide à l’enseignement en informatique. Ils seront trouvés grâce à un questionnaire accessible en ligne, depuis la page facebook de la promotion.
* Créer une grille de comparaison des outils à l’aide de 5 à 10 critères et 3 niveaux d’analyse pertinents. Ils seront définis à l’aide des résultats du sondage et des échanges réguliers avec le maître d’ouvrage.
* Analyser et hiérarchiser les solutions envisagées avec la grille construite.
* Rédiger le rapport d’activité ainsi que le livrable final, un compte-rendu de l’analyse effectuée, contenant la proposition de la solution répondant le mieux à tous les critères.

## Périmètre

Ce projet est effectué à la demande du maître d’ouvrage, Mr Daniel BARDOU, responsable informatique de la licence MIASHS à l’Université Grenoble Alpes.

L’équipe responsable du projet est constituée de 5 étudiants en Licence 2 MIASHS : FAURE Lora-Line, MAATA Mohamed Taha, MOMANTE SOH Micelle-ange, OUCOUC Hafid, PELLEGRIN Noémie.

## Cibles

Le projet est à destination d’un public universitaire, un professeur d’informatique, qui vise à mettre à disposition un outil d’aide à ses étudiants de L1 MIASHS. Ces derniers auront alors la possibilité d’utiliser l’outil pour s’entraîner individuellement sur des exercices supplémentaires.

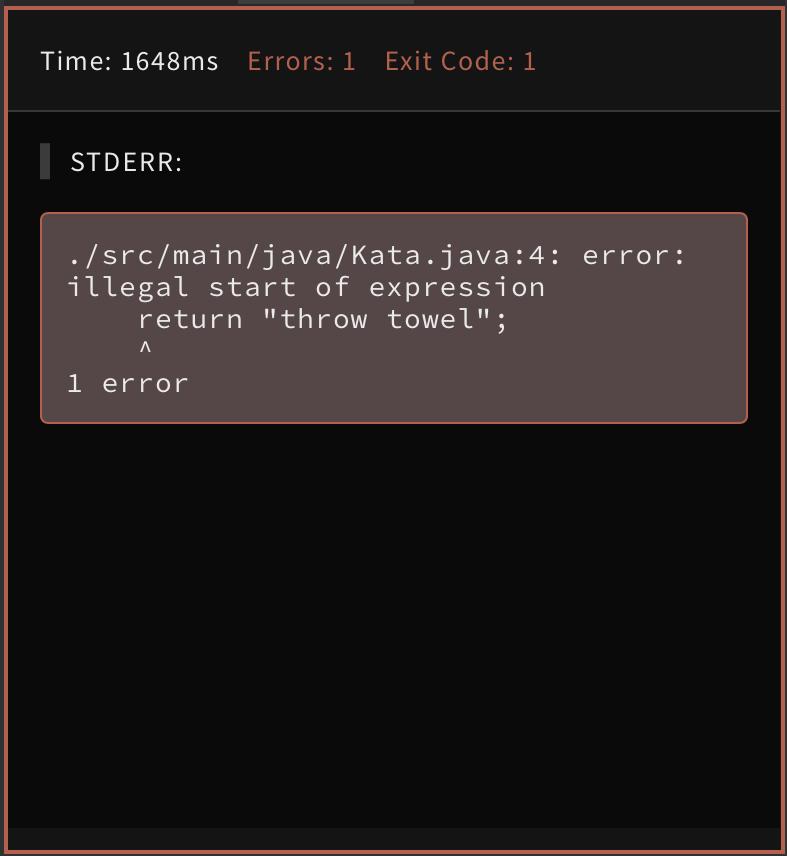
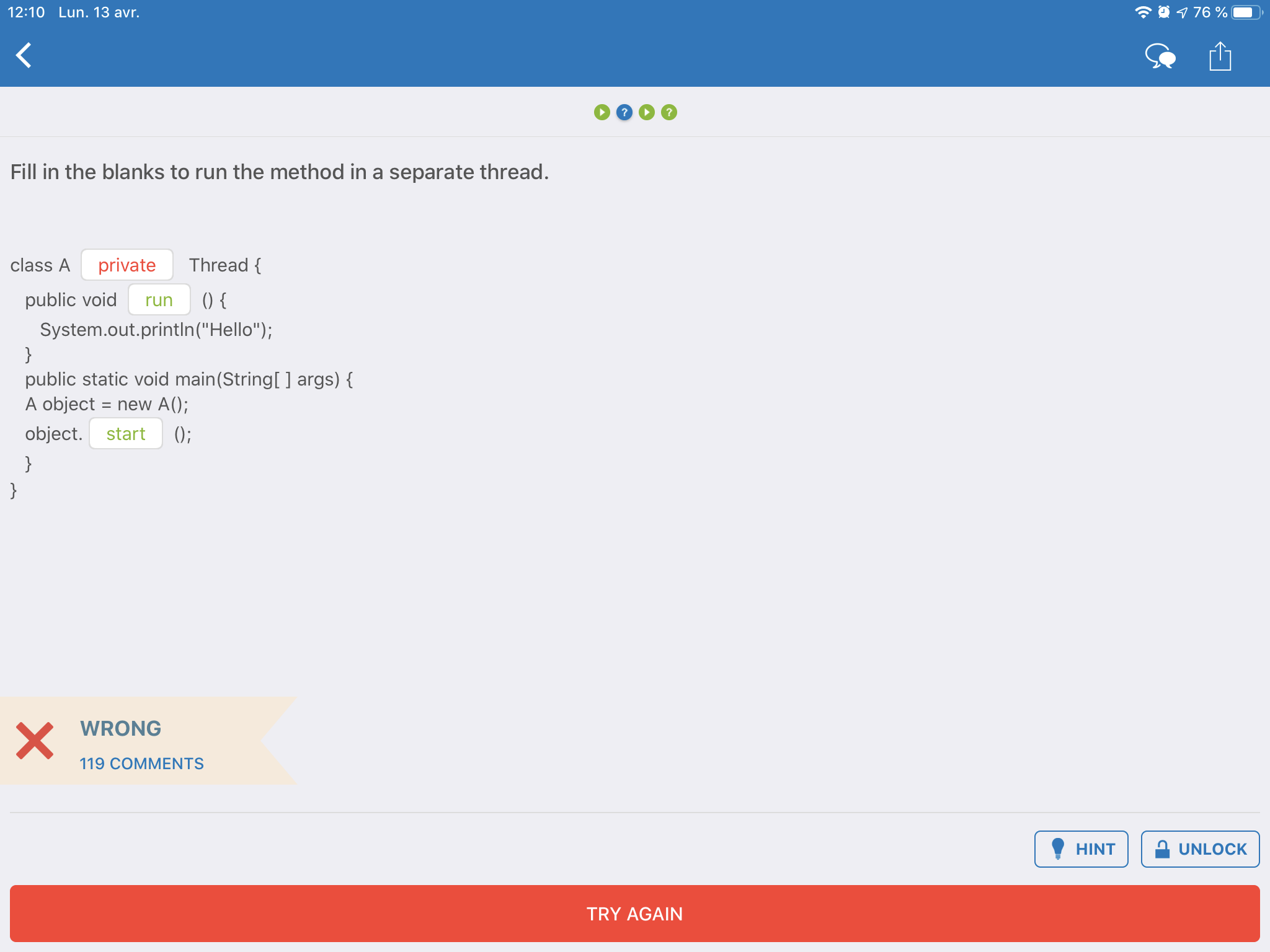
# Description fonctionnelle et technique

## Fonctionnalités

Suite aux entretiens avec le maître d’ouvrage et une analyse des besoins des étudiants, plusieurs fonctionnalités du produit ont été identifiées.

Parmi les fonctions principales de l’outil, l’une d’elles est de proposer des exercices de programmation à ses utilisateurs. Le but est de proposer aux étudiants un accès à des exercices postés par le professeur. Pour cela, un menu ou une présentation des exercices postés doivent apparaître. Pour le professeur, une page ou une commande pour déposer des fichiers d’exercices sont requis. Les contraintes sont le type et le volume des fichiers que peut accepter l’outil. Également, au minimum un langage de programmation doit être considéré, le langage java.

Ensuite, la deuxième fonctionnalité essentielle est la correction des réponses fournies aux exercices. L’outil doit valider ou rejeter la solution proposée pour un exercice donné. Ainsi, un espace doit permettre à l’étudiant de poster sa solution, puis après envoi, l’affichage d’une boîte de dialogue lui fournit un retour positif ou négatif. Pour illustrer cette fonctionnalité, deux exemples sont fournis (voir Figures 1 et 2).

Figure 1 : capture d’écran de la plateforme Sololearn. Figure 2 : capture d’écran de la plateforme

Elle représente un exemple d’exercice déjà proposé Codewars. Elle illustre la réponse fournie à

auquel l’étudiant peut répondre. Des cases à remplir un étudiant. Dans ce cas, une erreur est

sont disponibles pour accueillir sa solution. L’étudiant affichée, ainsi que son type et la ligne à

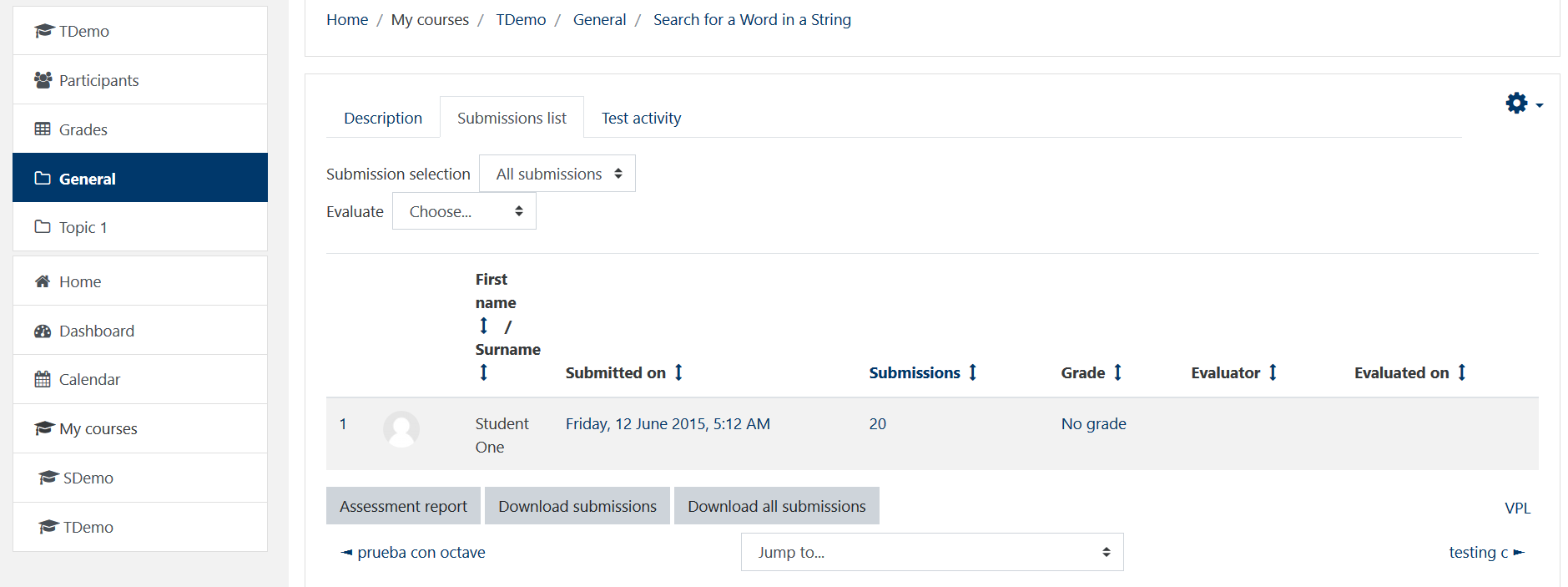
vérifie la validité de sa solution en considérant la couleur laquelle elle apparaît.

des mots affichés (rouge = incorrect, vert = correcte).

Puis, une autre fonction importante à considérer est celle de la mise à disposition de l’outil. Ceci exprime l’idée que l’outil doit être facilement accessible au professeur et mis à disposition des étudiants. L’accès à ce dernier est effectué par un lien ou un programme intégré sur le site du professeur ou sur la plateforme de cours en ligne de l’université, moodle. La contrainte majeure est alors le niveau d’accessibilité de l’outil. Celui-ci pouvant être utilisé sur divers outils technologiques : smartphone, tablette, ordinateur, … Ainsi que sur plusieurs navigateurs web. De plus, son fonctionnement doit respecter la charte d'utilisation des outils numériques de l’université.

Une autre fonctionnalité plus secondaire est à prendre en compte. Tout d’abord, un système d’aide à la résolution des exercices. Celui-ci comprend des fiches conseils ou des ressources d’aide à la programmation, comme des liens vers des sites web spécifiques au langage étudié.

Enfin, une dernière fonction, optionnelle, est la présence d’un historique des solutions proposées par les étudiants. Le professeur peut visualiser, dans une fenêtre ou un autre outil graphique, les solutions pour chaque exercice mis en ligne. Une éventuelle contrainte serait la gestion de cet historique, déléguée au professeur ou bien empêchant une prise en main aisée et rapide de l’outil. Un exemple concret est donné par la figure 3.

Figure 3 : capture d’écran de la plateforme Virtual Programming Lab. Une liste contenant le nom, la date, le nombre de solutions proposées et la note obtenue par l’élève est présenté au professeur. Cet historique existe indépendamment pour chaque exercice.

## Ressources techniques

La recherche et la mise en place de l’outil nécessite l’utilisation de ressources numérique et technologiques telles que les ordinateurs, smartphones, tablettes. Ensuite, pour assurer la communication et le travail de groupe des outils de travail collaboratifs et de communication numériques ont été utilisés. Enfin, pour la mise en place du questionnaire à destination des L1 MIASHS, l’application Google Form a été choisie. Celle-ci a permis de mettre à disposition rapidement et facilement le questionnaire aux étudiants.

## Contraintes

Tout d’abord, des contraintes d’organisation de l’équipe projet apparaissent à cause des semaines d’interruptions pédagogiques du 20 février et du 20 avril. Ensuite, la semaine du 9 mars 2020 a lieu quelques partiels de mi-semestre. De plus, la fermeture de l’université et le début de la période de confinement du 16 mars nécessitent une réorganisation du mode de fonctionnement de l’équipe et de la tenue des rendez-vous avec le maître d’ouvrage.

Ensuite, des contraintes liées aux ressources techniques sont envisagées. La communication entre les membres du groupe et le travail en commun requiert l’utilisation d’outils en ligne. C’est pourquoi, du matériel technologique personnel et universitaire tels que des ordinateurs, smartphones ou tablettes, ayant accès facilement au réseau internet sont nécessaires. De plus, l’usage de plateformes et ressources numériques et la navigation sur internet doivent être effectués en toute sécurité, en prenant en compte les droits et devoirs liés à leur utilisation et à leur accès. Ceci doit respecter les lois en vigueur, décrites dans le RGBD.

Enfin, des contraintes temporelles plus spécifiques à chaque objectif ont été fixées, dans le but de respecter les délais. Ainsi, ces dernières sont détaillées ci-dessous :

* Une période de 4 semaines a été allouée à la recherche des outils.
* Le questionnaire transmis aux étudiants est disponible pendant 2 semaines. L’analyse des résultats se déroule la semaine suivante.
* Pour la création de la grille critériée, une période de 6 semaines a été envisagée.
* Ensuite, 3 semaines sont consacrées à l’analyse des solutions considérées.
* Enfin, une plage d’une semaine est dédiée à la rédaction du livrable proposant les solutions.
* La rédaction du rapport d’activité est prévue sur toute la durée du projet.

# Organisation du projet et livrables

## Livrables du projet

Pour répondre à la demande effectuée, un compte-rendu écrit sera rendu. Il contiendra une grille de comparaison des différentes solutions envisageables assorties de leur analyse détaillée. De plus, il présentera le classement des solutions trouvées pour aboutir sur la proposition finale d’un outil. En option, une notice d’installation de la meilleure solution pourra être ajoutée.

En outre, pour répondre à l’évaluation de l’UE gestion de projet, un rapport d’activité de 15 pages est fourni. Celui-ci présente l’analyse du travail effectué par l’ensemble du groupe lors de la réalisation du projet. Egalement, il justifie les choix mis en oeuvre concernant la gestion agile du projet.

Enfin, dans le cadre d’une soutenance orale, une présentation visuelle avec des diapositives sera effectuée. Celle-ci sera un support pour la présentation du projet, de sa gestion ainsi que la justification de la démarche suivie par le groupe projet.

## Budget

La recherche des solutions et leur analyse s’effectuant grâce à des outils gratuits, accessibles librement et en ligne, aucun budget n’a été alloué à ce projet. Cependant, la solution finale proposée au maître d’ouvrage pourra nécessiter un abonnement ou un achat payant.

## Délais

La date de rendu final du projet est le 7 mai 2020. De plus, une soutenance orale aura lieu durant la semaine du 11 mai 2020. Afin de permettre une gestion du temps adaptée, un diagramme de Gantt a été réalisé (voir annexe).

# Annexe - Echéancier prévisionnel

L’échéancier prévisionnel à suivre a été construit selon un diagramme de Gantt, présenté ci-dessous (Figure 4).

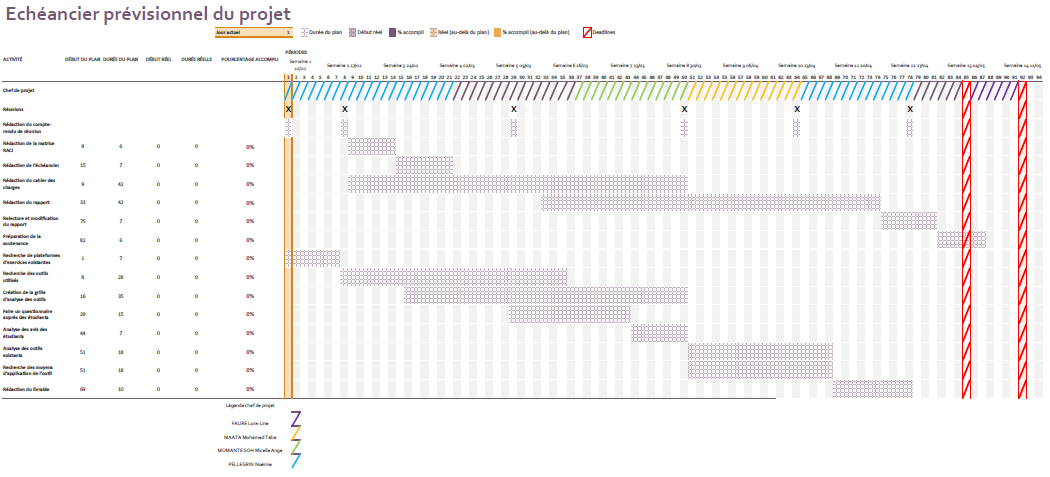


Figure 4 : échéancier prévisionnel du projet.

# 

# Annexe 5 - Grille critériée

| Etudiant | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Critères/Niveaux | | 1. Insuffisant | | | 2. Passable | | 3. Correct | |
| Accès à des documents d'aide | | Il n'est pas possible d'avoir accès à des documents d'aide, l’étudiant doit chercher par lui-même ou se référer à la page web de son professeur ou ses cours | | | Des documents ou références d'aides sont accessibles mais seulement dans une page qui y est dédiée spécifiquement. Elle n'est pas accessible depuis l'exercice en cours de réalisation. | | La page d'accueil ou la page sur laquelle l'exercice est proposé comporte des liens vers des sites web ou des fichiers d'explication de fonctions principales. | |
| Capacité d'auto-évaluation | | Les élèves ont seulement un avis binaire par rapport à leur solution. Ils ne savent pas où est le problème, ou bien à quel niveau ils ont réussi l'exercice. | | | La réponse à la solution proposée est binaire, mais une indication est donnée concernant la position de l'erreur. | | Les élèves obtiennent une note correspondant à la solution qu'ils proposent. Une indication est donnée concernant la partie du code qui ne fonctionne pas. | |
| Communication entre étudiants et/ou avec le professeur | | Il n'y a pas de commentaires ou de forum relié à la plateforme, les étudiants ne peuvent pas s'entraider. | | | Un outil de discussion entre étudiants existe mais il n'est pas relié à un exercice spécifique. Ou il existe un outil de discussion uniquement entre le professeur et 1 élève. | | Un forum ou une discussion de groupe est présent sur chaque exercice. Les étudiants peuvent poser des questions ou répondre à d'autres. De plus, les étudiants ont la possibilité d'envoyer directement un message à leur professeur. | |
| Saisie du code, de la solution | | Le fichier contenant la solution de l'élève, créé sur une application extérieure à la plateforme, doit être envoyé sur la plateforme pour être analysé par cette dernière. | | | Pour écrire son code, il est nécessaire de fournir à l'étudiant des lignes de codes ou des fichiers qu'il doit mettre en amont de ce qu'il écrit. | | Il est possible de copier son code java et le coller sur la plateforme pour l'exercice demandé ou de rédiger directement la solution. L'étudiant peut ensuite lancer la compilation et l'exécution de son programme. | |
| Bonus : difficulté des exercices déjà présents | | Les personnes débutantes peuvent difficilement faire les exercices proposés. Ou il n'y a pas plusieurs niveaux de difficulté proposés. | | | Les exercices présents sur la plateforme sont accessibles à tous les niveaux d'élèves, des débutants comme des étudiants ayant des connaissances dans le langage de programmation. | | Au fur et à mesure de la progression des élèves, des exercices correspondant à leur niveau sont proposés pour qu’ils puissent s'améliorer pas à pas. | |

| Professeur | | | | | |
| --- | --- | --- | --- | --- | --- |
| Critères/Niveaux | 1. Insuffisant | | 2. Passable | | 3. Correct |
| Feedback vers les étudiants | Le professeur ne peut pas commenter les solutions données par les élèves. | | Le professeur peut envoyer un message personnel à un étudiant, mais pas spécifiquement pour une solution particulière. | | Les solutions reçues peuvent être commentée par le professeur. Ou un commentaire général peut être envoyé à tous les étudiants au sujet d'un exercice. |
| Historique des solutions | Il n'y a pas d'historique des solutions proposées par les élèves. | | Le professeur reçoit chaque correction indépendamment, il n'y a pas d'historique réel. Ou l'accès à l'historique n'est pas intuitif. | | Un onglet ou une commande facilement identifiable sur la page d'accueil permet d'accéder à l'historique des solutions proposées. |
| Mise à disposition de la plateforme | Nécessite un processus d'installation d'un logiciel ou d'un plug-in pour les étudiants. Il ne fonctionne pas sur smartphone et tablette. Il n'est pas compatible avec tous les navigateurs. | | Accessible via un site internet externe. Son usage n'est pas possible sur l'un des deux outils : smartphone ou tablette. Un navigateur n'est pas compatible avec la plateforme. | | Mis en place et accessible depuis moodle ou le site du professeur, tous les navigateurs et les outils numériques peuvent l'utiliser. |
| Utilisation de la plateforme | Il faut effectuer plusieurs manipulations avant de poster ses exercices. Ou la plateforme ne permet pas de créer ou proposer des exercices. | | Quelques modifications des paramètres de la plateforme doivent être effectuées avant de pouvoir déposer les exercices. Ou les exercices ne sont pas postés uniquement à destination des étudiants mais à un public plus large. | | Le professeur peut déposer ses fichiers d'exercices directement sur la plateforme et les rendre accessible aux étudiants. |

| Outil | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Critères/Niveaux | 1. Insuffisant | | 2. Passable | | 3. Correct | |
| Attractivité et affordance | La plateforme n'utilise pas de couleurs, polices ou type d'affichage pour distinguer les éléments sur une page ou suggérer la fonction de chaque élément. Son contenu n'est pas attractif. | | La présentation globale utilise peu d'outils graphiques qui rendent la navigation agréable. Seules 1 ou 2 fonctionnalités sont précisément identifiables et suggèrent leur utilisation ou leur fonctionnement. | | La mise en page présente divers composantes (couleur, forme, taille police, ..) pour attirer le regard de l'utilisateur ou distinguer des composantes ou des fonctionnalités différentes. La majorité des fonctionnalités sont identifiables facilement. | |
| Diversité des langages de programmation | Un ou plusieurs langages sont proposés mais pas java. | | 2 à 3 langages sont proposés, dont un est java, mais il n'y a pas scheme. Ou il n'y a que java de disponible. | | 4 langages ou plus sont proposés. Les langages java et scheme sont présents. Éventuellement, les langages prolog, SQL sont proposés aussi. | |
| Langue utilisée | Seule une seule langue est disponible, ce n'est ni de l'anglais, ni du français. Ou la plateforme est partiellement traduite en anglais. | | Seules 2 ou 3 langues sont disponibles, dont l'anglais. | | La plateforme est disponible dans plusieurs langues (> 3) dont entièrement en français ou en anglais. | |
| Prix | L'utilisation ou la mise en place de la plateforme nécessite un abonnement ou un prix très important. | | Un abonnement ou un prix moyen est demandé. | | Un investissement financier minime voir nul est requis. | |

# Annexe 6 - Questionnaire fourni aux L1 MIASHS

# 

# 

# Annexe 7 - Hiérarchie des solutions trouvées

| Classement | Nom de la plateforme |
| --- | --- |
| 1 | Virtual Programming Lab |
| 2 | Qualified |
| 3 | France IOI |
| 4 | Codewars |
| 5 | Sololearn |
| 6 | W3Schools |
| 7 | Coding Dojo |
| 8 | Swift Playgrounds |
| 9 | JUnit |