

UNIVERSITÉ D'ANGERS

MANAGEMENT DE PROJET

M2 ANALYSE, CONCEPTION ET DÉVELOPPEMENT
INFORMATIQUE

LIDE : Spécifications fonctionnelles

Managers :

Julien FONTAINE Louis
MARCHAND Paulin
VIOLETTE

Développeurs :

Etienne TIGNON
Gabriel MOUGET
Romain GRELIER
Romane POIRIER

16 octobre 2018



1 Le projet

Dans le cadre de la formation de Master 2 ACDI à l'Université d'Angers, une unité d'enseignement transversale avec les étudiants de M1 est proposée. Dans ce cadre nous devons mener un projet de développement dans lequel les étudiants de M2 (groupe de 2 ou 3 étudiants) sont les chefs de projets des étudiants de M1 (groupe de 4 ou 5) qui sont les développeurs. Il s'agit du sujet n°3 : Environnement de développement en ligne.

L'objectif de ce projet est de développer un environnement de développement en ligne accessible depuis un navigateur web classique (l'application doit au moins fonctionner sur Firefox et Chrome). Cet environnement doit permettre l'édition, la compilation et l'exécution de code simple directement depuis le navigateur sans avoir à installer un quelconque outil sur les postes des utilisateurs. Toute la partie concernant la compilation et l'exécution du code est donc effectuée sur un serveur et le résultat rendu à l'utilisateur dans son navigateur. Tout ceci doit être transparent pour l'utilisateur.

Les cibles de l'application sont principalement les étudiants de L1 de l'Université d'Angers. Dans le cadre des enseignements ils sont amenés à faire du développement. Dans la mesure où ces étudiants peuvent ne pas disposer d'un environnement de développement sur leur poste personnel, ils doivent pouvoir continuer leur projet en dehors de l'université. La cible peut être poussée jusqu'aux étudiants de L3 voire M1 si les fonctionnalités proposées par l'application sont suffisamment avancées pour justifier une utilisation par ces étudiants.

L'application fut commencée l'année dernière. Dans l'état actuel, l'application est une preuve de concept. Des changements techniques sont nécessaires pour faire bien fonctionner l'application, mais les bases de l'implémentation sont posées.

Les fonctionnalités de l'application actuelles sont les suivantes :

- Plusieurs langages de programmation doivent pouvoir être compilés dans l'application pour que les étudiants puissent s'en servir dans leurs différentes Unité d'enseignement. L'ajout d'environnement d'exécution (pour différents langage, différents framework ou bibliothèque) doivent pouvoir être effectué par les administrateurs.
- La coloration syntaxique doit s'appliquer en fonction du langage.
- La compilation et l'exécution doivent s'effectuer sur le serveur, l'utilisateur reçoit la sortie sur son navigateur. L'implémentation technique de cette partie va être revue, afin de régler les pertes de données (cer-

taines partie de la sortie du programme pouvait ne pas être transmise à l'utilisateur dans certaines conditions) qui sont ne sont pas acceptable pour l'utilisateur, et d'améliorer l'expérience utilisateur en proposant une utilisation plus fluide, avec moins de délais entre les actions de l'utilisateur dans lors de l'exécution.

- L'utilisateur doit pouvoir observer les messages d'erreur.

Il est prévue d'ajouter les fonctionnalités suivantes :

- L'auto complétion du code.
- Un espace de stockage personnel pour les étudiants doit être à disposition. Il serait préférable que la sauvegarde des fichiers soit transparentes pour l'utilisateur.
- La gestion de projet contenant de multiple fichiers. Les étudiants ou professeurs ont la possibilité de partager leurs projets, qui peuvent être clonés par d'autre étudiant. Cette fonctionnalité est particulièrement intéressante en TP : un professeur peut créer un projet avec certaines parties laisser à remplir pour l'étudiant, qui peut ensuite faire uniquement le travail demandé, sans perdre de temps sur la mise en place du TP.
- L'utilisateur doit pouvoir utiliser un débogueur simple (ajout de breakpoint dans le code, possibilité de faire du pas à pas dans l'exécution des programme et d'observer l'état de la mémoire), via l'interface graphique.
- Un outil d'analyse du code déposé par les étudiants doit être à disposition des professeurs. Il s'agit de tests simples du code : vérifier que le code compile, tester la sortie du programme selon des entrées données, etc...

2 Personas

La cible principale est l'étudiant de L1. Parmi ces étudiants, on peut distinguer deux types d'étudiant :

- **Les étudiants sans aucune expérience de programmation.** Ils sont probablement à l'aise avec le web, et ont des compétences informatiques limitées à la bureautique. Il n'ont jamais utilisé de véritable IDE. Parmi ces étudiants, certains souhaitent continuer vers l'informatique. L'application se doit d'être facilement abordable pour ces étudiants, mais également de les préparer à l'usage de plus gros IDE

tel qu'Eclipse, les produits JetBrains (IntelliJ, PhpStorm), Visual Studio...

- **Les étudiants ayant une déjà une expérience de programmation** par intérêt personnel, ou alors ayant eu un parcours avec de la programmation, par exemple, une spécialité ISN en bac S. Ces étudiants ont possiblement déjà utilisé un IDE : l'application se doit d'être familière pour ces étudiants, c'est à dire garder les normes d'interface que l'on peut retrouver dans la plupart des IDE (onglets pour les fichiers ouverts, console d'exécution intégré dans l'IDE, symbolique des icônes sur les boutons d'actions ...)

L'autre type d'utilisateurs sont les **professeurs de l'université d'Angers** administrateur de l'application. Ces professeurs sont supposés avoir un bon niveau sur les principes de programmation enseignés aux étudiants, et ont déjà utilisé des éditeurs de texte orientés programmation (geany, vim, emacs...) ou des IDE¹. Les administrateurs ont la possibilité de configurer les langages et environnements d'exécution accessibles depuis l'application, ainsi que la configuration des modèles de fichiers accessibles aux étudiants. L'application doit proposer pour ces utilisateurs une interface d'administration permettant ces tâches.

3 Parcours d'un utilisateurs

3.1 Évolutions de l'interface

Dans l'application actuelle, un utilisateur étudiant à accès à trois interfaces :

- La page d'authentification
- La page de développement
- La page de gestion du profil

L'utilisateur non inscrit a également la possibilité de s'inscrire sur une page accessible depuis la page d'authentification.

Afin d'intégrer la gestion de projets à l'application, une interface permettant à un utilisateur de gérer et d'accéder à ses différents projets sera intégrée. Cette interface sera la page d'accueil pour un utilisateur authentifié. Depuis

1. IDE : *Integrated Development Environment*, ou Environnement de Développement Intégré (EDI) en français, est une application avancée destinée à la programmation

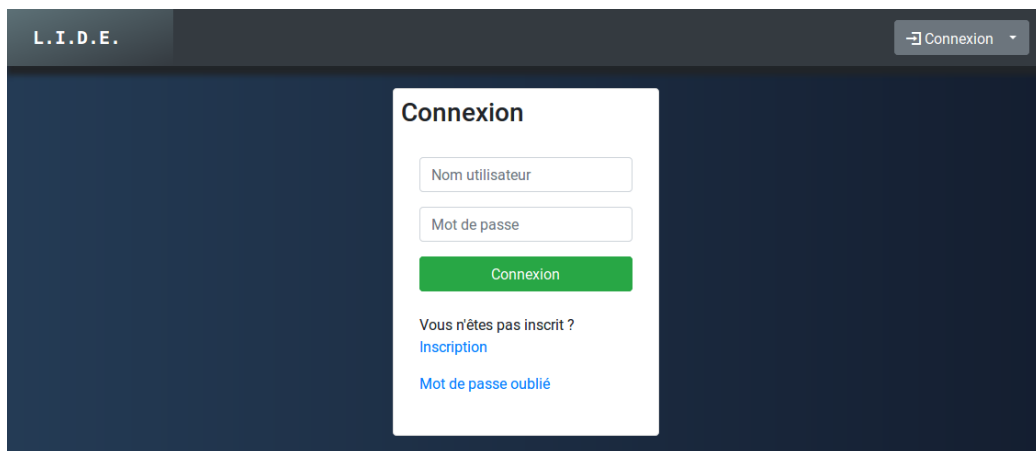


FIGURE 1 – Interface d’authentification

cette page, l’utilisateur pourra sélectionner ou créer un projet à ouvrir dans la page de développement.

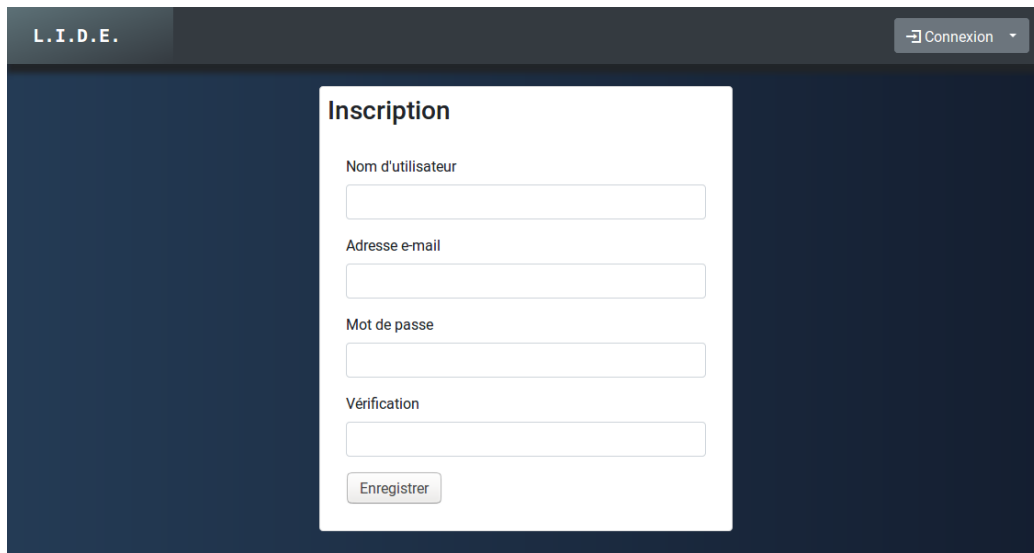
3.2 Interface d’authentification

L’interface d’authentification (voir figure 1) ne changera pas (du moins pour les fonctionnalités, le style visuel pourrait éventuellement changer afin de s’accorder au reste de l’application). Elle est simplement composée d’un formulaire composant d’un champ nom d’utilisateur et d’un champ mot de passe, ainsi que d’un bouton permettant de se connecter.

Un liens permet d’accéder à l’interface d’inscription (voir figure 2). L’inscription demande un nom d’utilisateur et un mot de passe (par lesquels l’utilisateur se connecte), ainsi qu’une adresse mail, sur laquelle est envoyé un mail de confirmation permettant d’activer le compte. L’inscription n’est possible que si l’adresse mail est sur un domaine autorisé par l’application (dans le cas de l’université d’Angers, les adresses en @etud.univ-angers.fr ou @univ-angers.fr).

3.3 Interface de gestion de projet

Cette interface (voir figure 3 et 4) permet à l’utilisateur de visualiser et de gérer ses projets. Elle permet d’accéder à un projet existant, de créer un nouveau projet ou de cloner un projet existant (soit un projet lui appartenant



The image shows a web interface for registration. At the top, there is a dark blue header bar. On the left side of the header, the text 'L.I.D.E.' is displayed. On the right side, there is a button labeled 'Connexion' with a small icon. Below the header, the main content area has a dark blue background. In the center, there is a white rectangular box titled 'Inscription'. Inside this box, there are four text input fields, each with a label above it: 'Nom d'utilisateur', 'Adresse e-mail', 'Mot de passe', and 'Vérification'. Below these fields, there is a button labeled 'Enregistrer'.

FIGURE 2 – Interface d’inscription

ou un projet public d’un autre utilisateur).

Lorsqu’un utilisateur n’a aucun projet créé, nous lui proposons un lien vers un manuel d’utilisation. Un manuel d’utilisation existe déjà la version produite l’an dernier (sous format PDF). Il sera nécessaire d’extraire et de mettre à jour les informations de ce manuel d’utilisation, afin de le proposer sous la forme d’une page web intégré à l’application.

3.3.1 Création d’un nouveau projet

Chaque utilisateur peut créer autant de projet qu’il veut, dans la limite de l’espace de stockage alloué par les administrateurs. Chaque projet est défini par un nom (unique par utilisateur), et d’un environnement d’exécution pour le code. Cet environnement peut être changé plus tard si nécessaire, via l’interface de configuration du projet sur la page de développement. Une fois un projet créer, l’utilisateur est redirigé vers la page de développement.

L’utilisateur peut également créer un projet à partir d’un modèle (c’est à dire un projet existant, créé par lui même ou un autre utilisateur). Pour cela, il doit disposer d’un code correspondant au projet. Il peut ensuite saisir ce code. Après la saisi du code, des informations sur le projet (nom, environnement) lui sont affiché, ou alors si le code donné n’est pas valide, un message

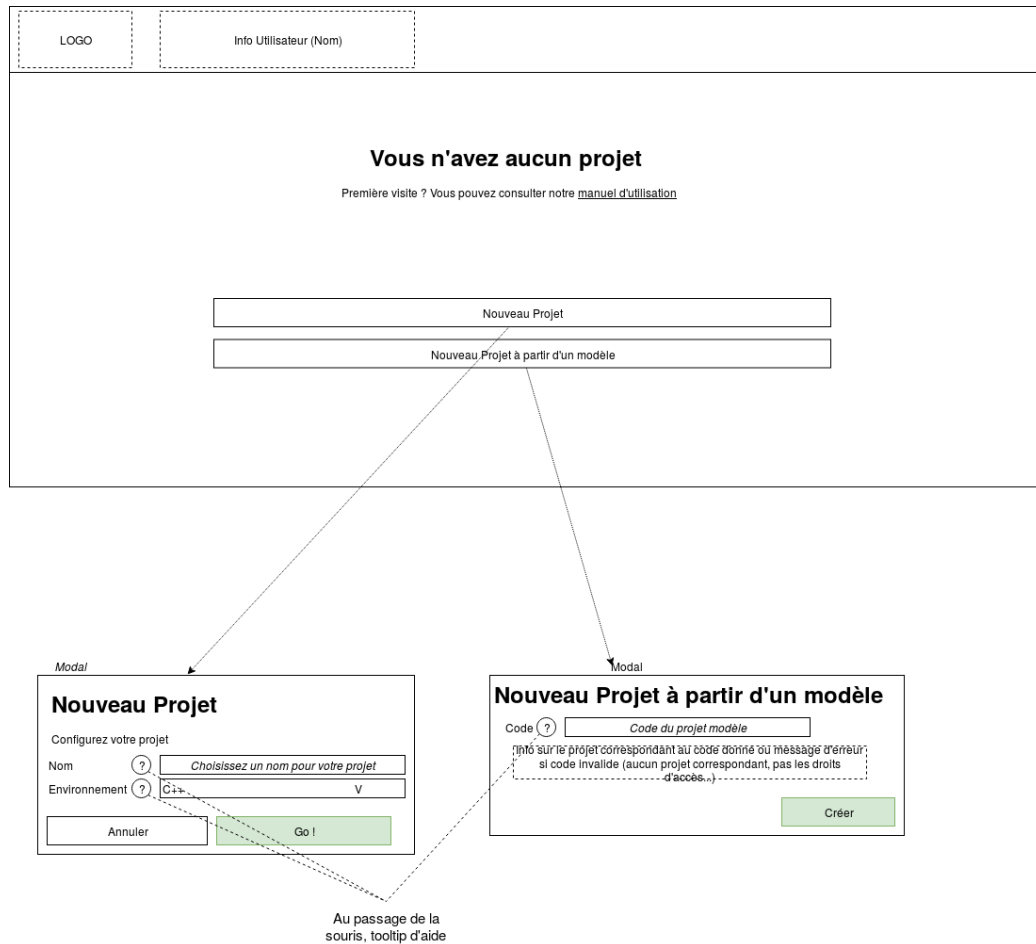


FIGURE 3 – Interface de gestion de projet pour un utilisateur n'ayant aucun projet

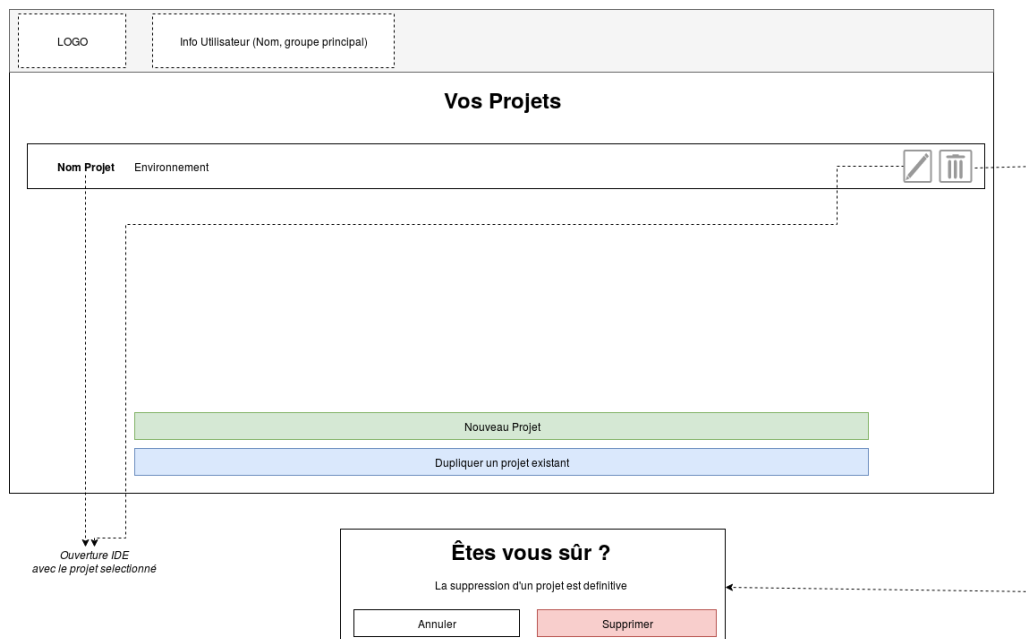


FIGURE 4 – Interface de gestion de projet avec des projets existants

d'erreurs. Si le code est valide, il peut ensuite cloner le projet, puis il est redirigé vers l'environnement de développement.

3.4 L'environnement de développement

L'interface souhaitée pour l'environnement de développement permet d'intégrer la gestion de projet dans l'environnement. Les normes des EDI² existants sont reprises (voir figure 5 : un navigateur de fichier dans un panneau latéral, des menus permettant de créer des fichiers et de configurer l'environnement).

L'utilisateur a la possibilité de masquer la console. Il peut également personnaliser l'affichage (modal configuration de l'affichage) : configuration des couleurs de l'éditeur et de la console, de la taille de la police et également de changer la position du terminal : soit sur le côté, soit en bas (voir figure 6)

L'interface s'adapte au niveau de l'utilisateur. Lors de la création d'un

2. EDI : Environnement de développement intégré

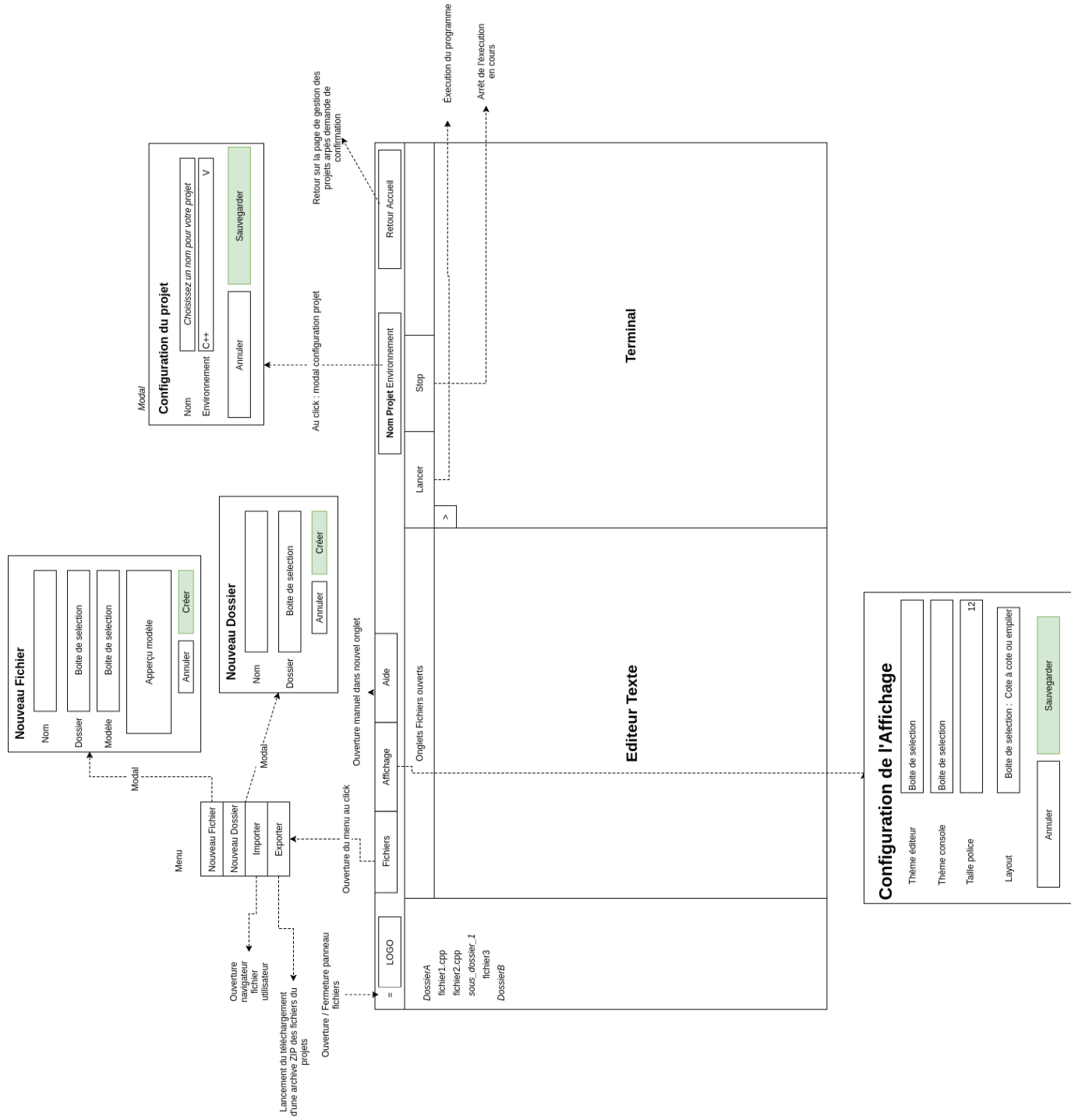


FIGURE 5 – Environnement de développement

environnement, l'administrateur peut rendre disponible ou non des fonctionnalités. Ceci permet de présenter une interface simple au nouveau étudiant, afin qu'ils ne soient pas perdus dans les fonctionnalités. La configuration de l'environnement permet :

- D'activer le support de multiples fichiers : si cette option est désactivée, le panneau latéral permettant de visualiser la liste des fichiers et les onglets des fichiers ouverts disparaît, tout les options du menu de gestion de fichier permettant la création de fichiers et dossier. Seul un fichier est ouvert, et il ne peut pas être fermé. L'import de fichier remplace le contenu du seul fichier ouvert, au lieu de créer un nouveau fichier.
- De rendre disponible ou non le débogueur.

Il faudrait également étudier la possibilité de changer légèrement la disposition et/ou la taille des boutons, afin de mettre en évidence les actions possibles par l'utilisateur. Néanmoins, le passage d'une interface "débutant" à l'interface plus avancée ne doit pas être perturbante.

Lorsque le support de multiples fichiers est activé, l'utilisateur peut créer de nouveaux fichiers. À la création d'un nouveau fichier, il peut définir le nom du fichier, ainsi que choisir un modèle de création du fichier. Les modèles de fichiers, défini par l'administrateur, permettent de créer un fichier avec une base de code déjà présente.

3.5 Interface d’administration

L’interface d’administration ne change pas par rapport à la version actuelle (voir figure 7, à l’exception d’un changement de nomenclature : la catégorie Langage sera renommé Environnement (en effet, les ressources disponibles dans cette catégorie correspondent à des environnements de compilation et d’exécution plutôt qu’à des langages de programmations), et la catégorie Détail Langage sera renommée en Modèle Fichier.

Cette interface est accessible depuis un lien dans la barre de navigation de chaque page, visible uniquement par un utilisateur administrateur authentifié.

Depuis cette interface, un administrateur pourra

- Ajouter, supprimer, modifier, activer et désactiver des environnements.
- Ajouter, supprimer, modifier des modèles de fichiers (qui sont liés à un environnement).
- Voir les utilisateurs inscrits

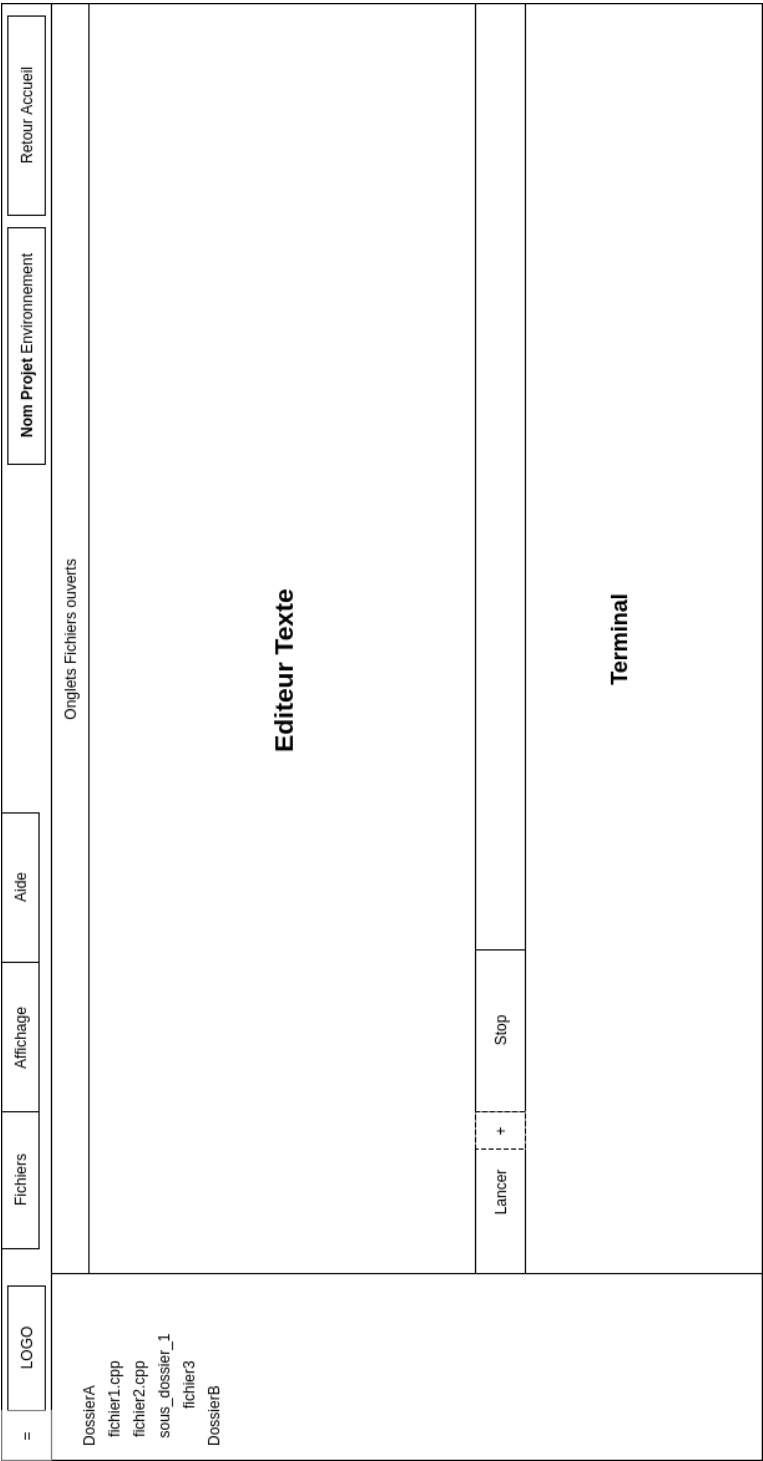


FIGURE 6 – Environnement de développement (terminal en bas)

The screenshot shows the 'LIDE administration' interface. On the left is a dark sidebar with navigation links: 'Langages', 'DetailLangage', 'Serveur', 'User', and 'Retour au site'. The main content area is titled 'Langage' and contains a table with the following data:

ID	Nom	Options	Compilateur	Docker name	Actif	Actions
3	java	AUCUNES	javac	java	NON	Editer Supprimer
2	c	AUCUNES	gcc	gcc	OUI	Editer Supprimer
1	C++	-Wall	g++	gpp	OUI	Editer Supprimer

At the top right of the main area, there is a search bar labeled 'Rechercher' and a button labeled 'Créer Langage'. The user 'admin' is logged in, as indicated in the top right corner.

FIGURE 7 – Interface d’administration actuelle

4 Planning

Suite aux réunions du 3 et 10 octobre, les tâches ont été réparties de la manière suivantes :

- **API Gestion de projet** : Romane Poirier, encadré par Paulin Violette.
- **Déploiement automatisé des environnement** : Romain Grellier, encadré par Julien Fontaine
- **Interfaces EDI et liste des projets** : Gabriel Mouget, encadré par Paulin Violette.
- **Auto complétion** : Etienne Tignon, encadré par Louis Marchand.

Le planning suivant a été établie pour les tâches (voir le diagramme de Gant figure 8) :

Développeur	Tâche	Début	Fin
Romane	API Gestion Projet	15/10	09/11
Gabriel	Interface Accueil Projet	15/10	31/10
Romain	Déploiement Auto. Env.	15/10	26/10
Etienne	Recherche Auto Complétion	15/10	31/10
Romain	Intégration Continue	29/10	02/11
Gabriel	Interface EDI	1/11	21/11
Etienne	Implémentation Auto Complétion	01/11	21/11
Romain	Débogueur	05/11	23/11
Gabriel et Romane	Intégration API /interface	12/11	16/11
Tous	Rédaction rapport	27/11	10/12

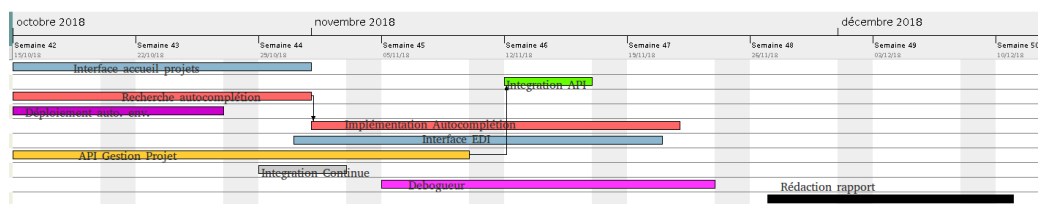


FIGURE 8 – Diagramme de Gant

Le temps non attribué pourra être utilisé pour redistribuer les tâche en fonction de retards éventuelles.