CAHIER DES CHARGES FONCTIONNEL

Javora

VENTURA Alexandre

BORDES Yohan

VARANGE Jeremy

LACASTE Noah

16 Octobre 2020

Dépôt GIT : https://dwarves.iut-fbleau.fr/git/bordes/ProjetTuteure/

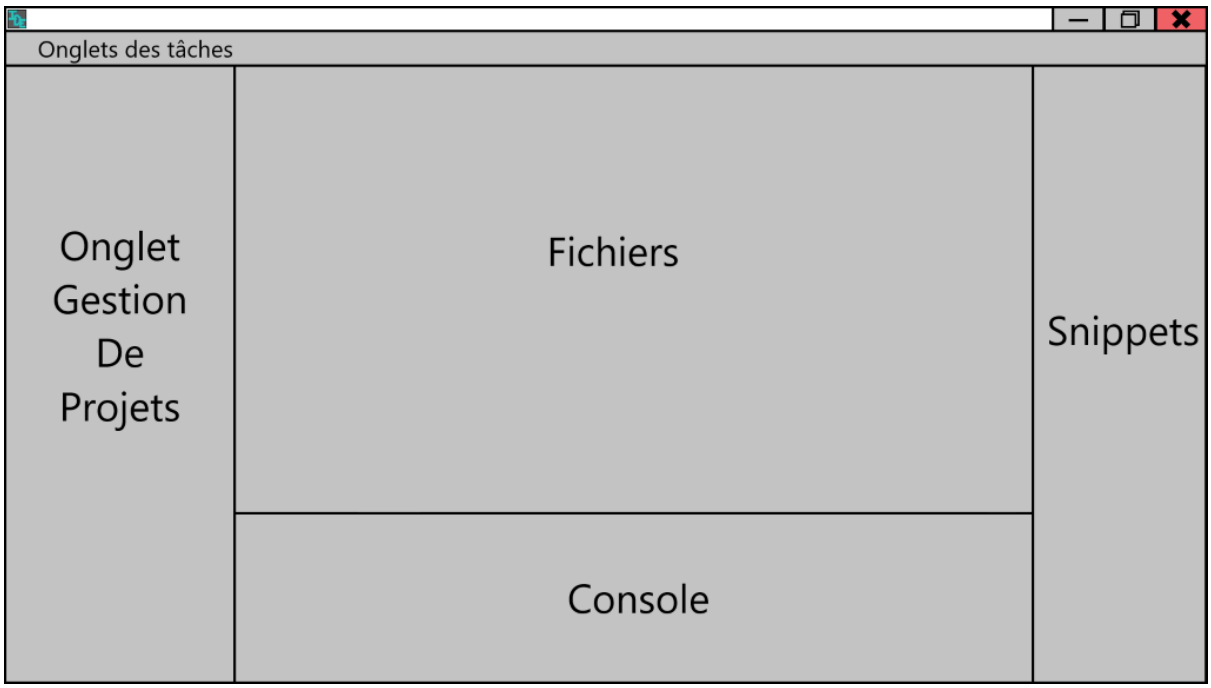
Sommaire

Présentation du logiciel

Un IDE, ou environnement de développement, est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels. Il comporte traditionnellement un éditeur de texte et un compilateur. Nous avons envisagé d'imaginer et de développer notre propre IDE pour répondre aux besoins que les environnements populaires ne peuvent satisfaire, c'est-à-dire un IDE où l'on peut intégrer le code de nos professeurs simplement tout en suivant leur méthodologie d'organisation et de rédaction. Notre application permettra d'écrire et exécuter le langage JAVA.

On cible ici comme principaux utilisateurs les développeurs, notamment ceux de l'IUT. Pour se faire cet outil devra être ergonomique, intuitif et simple d'utilisation et cela, même pour un débutant. Le design de l'user interface est donc une partie importante du projet. De plus l'objectif de l'application est aussi d'être assez complet pour pouvoir être utilisé tout en prenant de bonnes habitudes de rédaction de code grâce à des raccourcis mais aussi avec une gestion de projet adéquate.

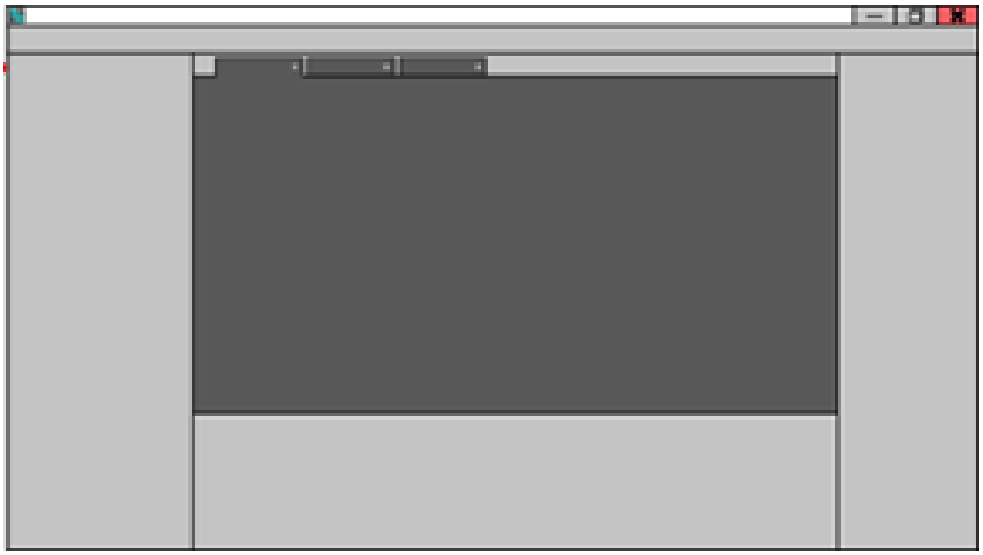
L'application ne devra pas donc être trop lourde car ceci viendrait directement affecter l'efficacité avec laquelle on peut coder si l'IDE à des problèmes de ralentissement, de bugs, …
L'aspect général du logiciel correspondra donc à ceci :



Spécifications des fonctionnalités

Créer, lire et modifier un fichier :

Il s'agit d'une fonctionnalité vitale pour un environnement de développement. On doit pouvoir ouvrir un fichier et modifier son contenu. Pour cela, le fichier sera lu par le logiciel et affiché dans le bloc central de la fenêtre. L'utilisateur doit cependant être capable de modifier plusieurs fichiers en même temps. Cela sera rendu possible via la mise en place d'un système d'onglets comme illustré dans ce wireframe.



Il sera possible de créer un fichier via le sous-menu « new » du menu file ou encore du menu affiché par un clic droit dans le bloc consacré à la gestion de projet. De plus, lors de la création d'un fichier, il sera possible de sélectionner le code de base qu'il contiendra via des cases à cocher dans une boite de dialogue.

Compiler :

C’est une des fonctionnalités les plus importantes. Pour pouvoir compiler un programme, il est nécessaire d’ouvrir notre invite de commande puis d’écrire les différentes commandes. Pour y remédier, l’IDE intégrera un bouton « compiler » dans sa barre d’outils ainsi que dans les menus (clic droit, barre de menu). Lors du clic sur ce bouton, le logiciel appellera le compilateur installé sur la machine avec la commande nécessaire afin de transformer le projet en un exécutable. Cette fonctionnalité augmente le confort d’utilisation de l’IDE ainsi que l’efficacité du développeur.

Terminal :

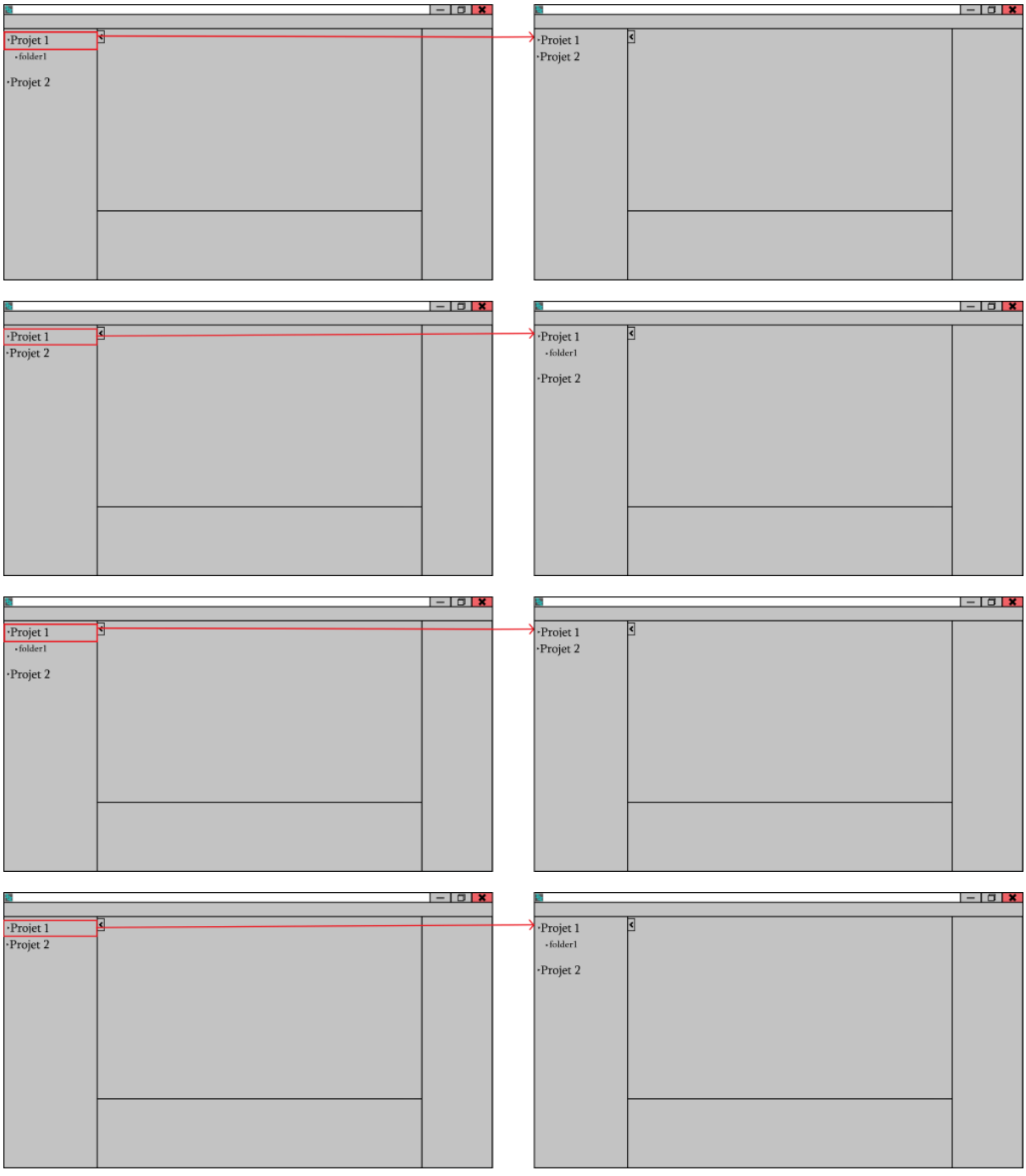
Dans le bloc situé en bas de la fenêtre, il sera possible d’ouvrir des terminaux sur lequel les projets en cours de développement s’exécuteront. Il sera possible d’y ouvrir deux types de terminaux :

- Les terminaux libres : dans lequel on peut exécuter les commandes que l’on veut.
- Les terminaux protégés : dans lequel il ne sera possible que d’exécuter des projets enregistrés dans l’IDE.

Afin de gérer la multiplicité des terminaux, un système d’onglet sera également présent comme illustré dans ce wireframe.

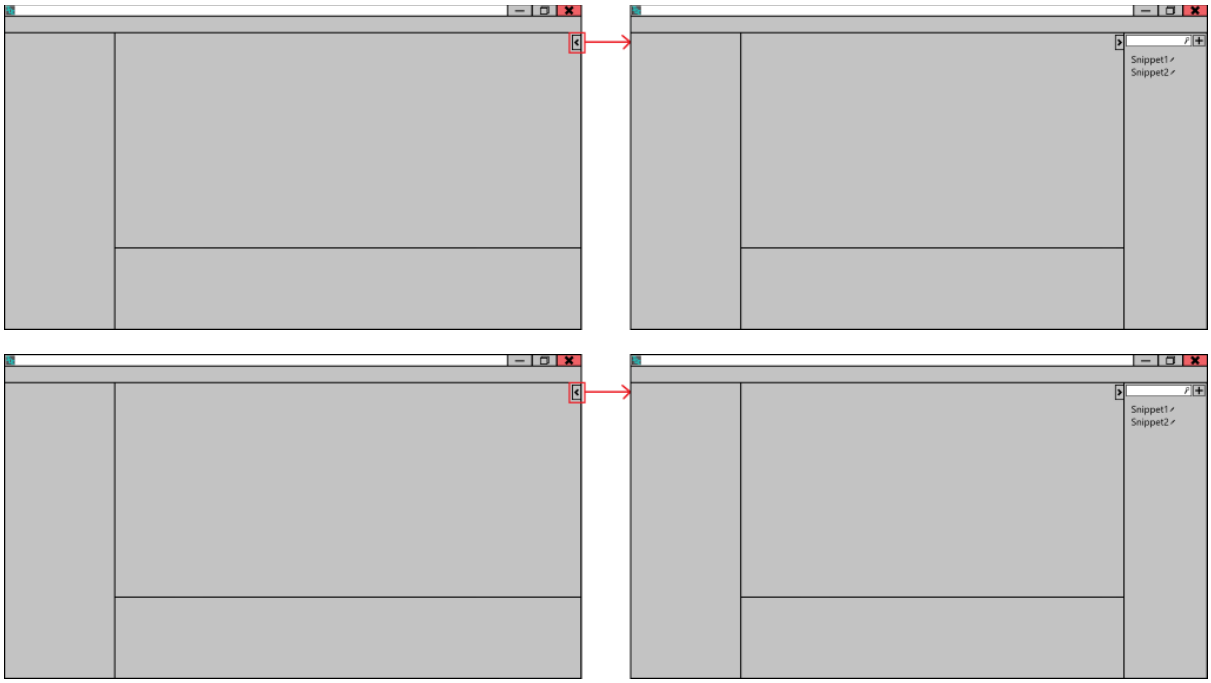
Onglets gestion de projet :

L’application doit pouvoir ouvrir tous les fichiers concernant un même projet ainsi que de gérer leur arborescence c’est-à-dire créer, supprimer voire renommer ces fichiers et dossier en plus d’en faire une représentation dans l’IDE.



Snippets :

De nombreuses portions de codes sont régulièrement utilisées par les développeurs telles que l'affichage du contenu d'un tableau, l'affichage d'une fenêtre… Pour éviter de perdre du temps, l'IDE permettra l'intégration de snippets. Un snippet est une petite portion réutilisable de code. L'application devra être capable d'en ajouter de nouveaux à sa bibliothèque et de les intégrer au code en train d'être édité simplement.



Coloration syntaxique :

La coloration syntaxique permet d'améliorer la lisibilité du code en mettant en évidence sa structure. Un environnement de développement se doit d'en faire usage pour rendre le développement plus efficace. Les commentaires seront colorés en vert et les mots clé, en bleu et en rouge.

Liste de suggestions :

Aussi appelé auto-complétion, la liste déroulante d'options en fonction des caractères saisie mais aussi du type de variable saisie est d'une grande aide pour un développeur. En effet, le langage JAVA possède de nombreuses méthodes que l'utilisateur pourra utiliser sans devoir regarder la documentation. Cet éditeur devrait aussi posséder cet outil.

Synchroniser avec git :

Pour pouvoir synchroniser un projet avec git, il faut ouvrir gogs sur Internet, puis créer un répertoire et écrire de nombreuses commandes pour le configurer. A l'aide d'un simple bouton, l'IDE gérera toutes ces manipulations automatiquement. Dans l'idéal il devrait être capable de gérer les différentes versions du projet en train d'être réalisé en regardant si des fichiers ont été modifiés.

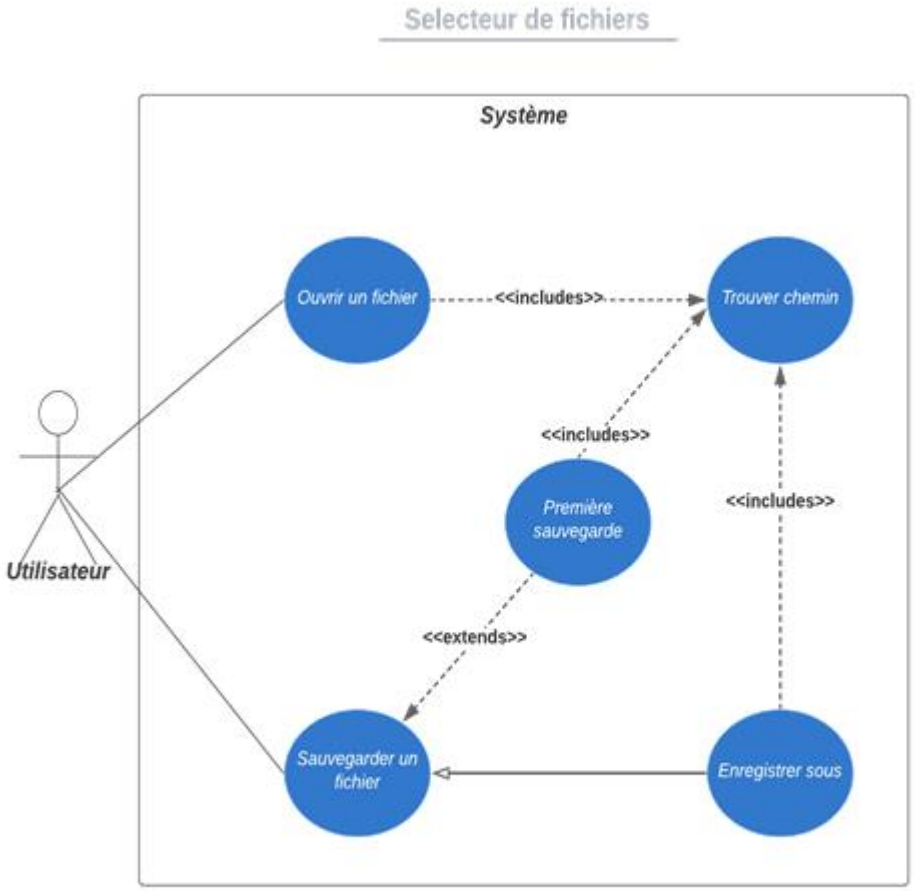
Raccourcis clavier :

Les raccourcis claviers sont essentiels pour écrire efficacement, bon ombre d'entre eux seront donc naturellement présents dans un IDE.

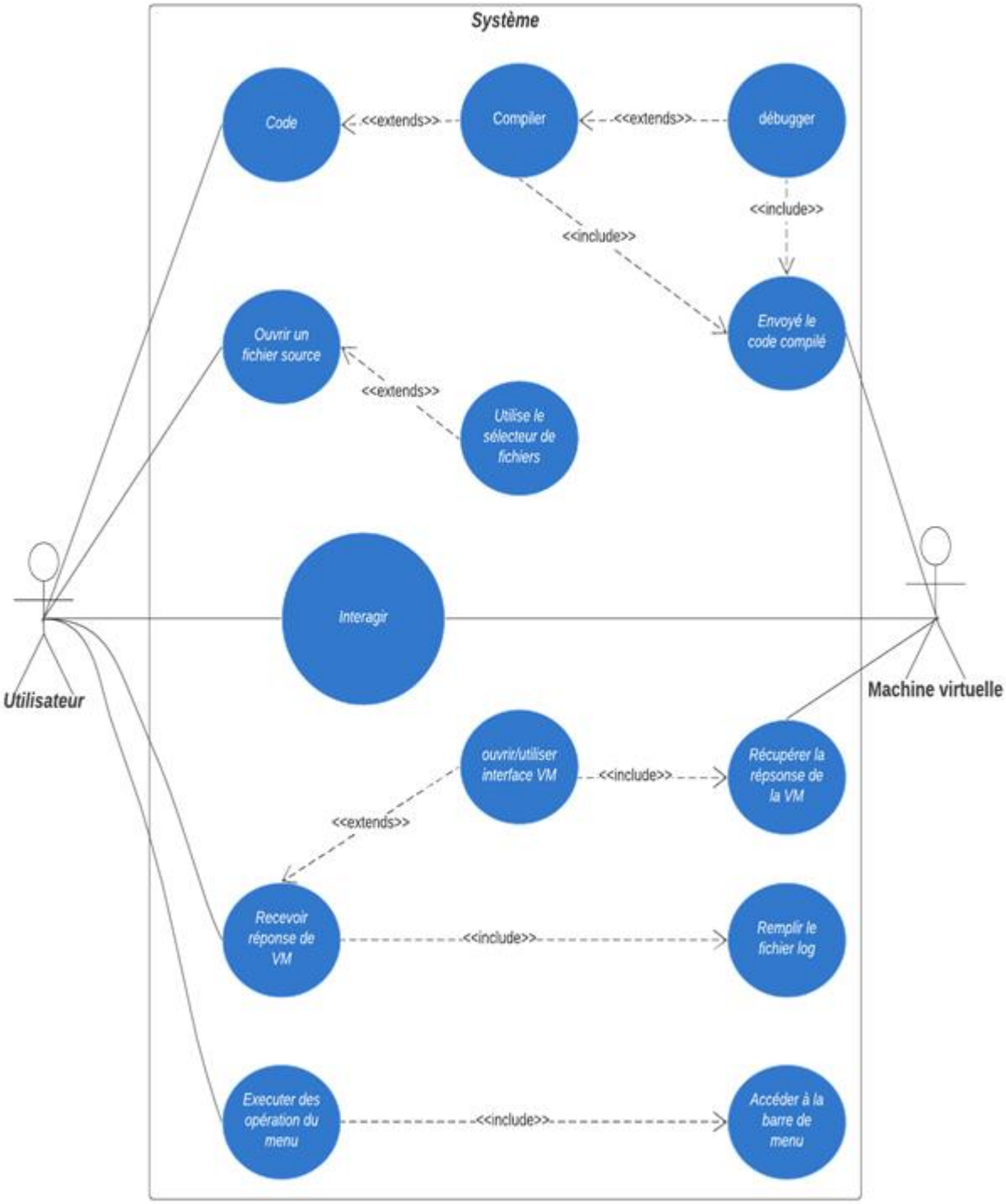
Thèmes personnalisés :

L'intégration d'un thème sombre et clair est primordiale pour le confort des yeux et une lisibilité agréable de l'application.

Diagramme de cas d’usage



IDE


Système
Code
<<extends>>
Compiler
<<extends>>
débugger
<<include>>
<<include>>
Envoyé le code compilé
Ouvrir un fichier source
<<extends>>
Utilise le sélecteur de fichiers
Interagir
Utilisateur
Machine virtuelle
ouvrir/utiliser interface VM
<<include>>
Récupérer la répsonse de la VM
<<extends>>
Recevoir réponse de VM
<<include>>
Remplir le fichier log
Executer des opération du menu
<<include>>
Accéder à la barre de menu

Priorisation des objectifs

Afin de définir la priorité des objectifs de ce projet, nous allons utiliser la méthode MosCoW. La méthode MoSCoW est une technique visant à prioriser des besoins ou des exigences avec : M pour must (vital), S pour should (essentiel), C pour could (confort), W pour won't (probablement jamais livré mais envisageable).

Must	Should	Could	Would
- Lecture - Ecriture - Compilation	- Onglet de gestion de projet - Snippets - Coloration - Auto-complétion	- Raccourcis clavier - Raccourcis clavier	- Génération de makefile -Transformation du code en diagramme -Synchronisation avec le git