

# Rapport Général

Iteration 1 : Du 25/05/2020 au 02/06/2020

## Cahier des charges

Nous avons d'abord défini le cahier de charges pour bien cerner le besoin des différents utilisateurs du système, définir les différentes fonctions principales et fonctions contraintes du système et enfin, envisager différentes réponses à ces besoins.

CF : cahier\_des\_charges.odt

## Maquettes

Nous avons ensuite fait plusieurs maquettes pour représenter le système que l'on voudrait mettre en place, avec chacun sa représentation.

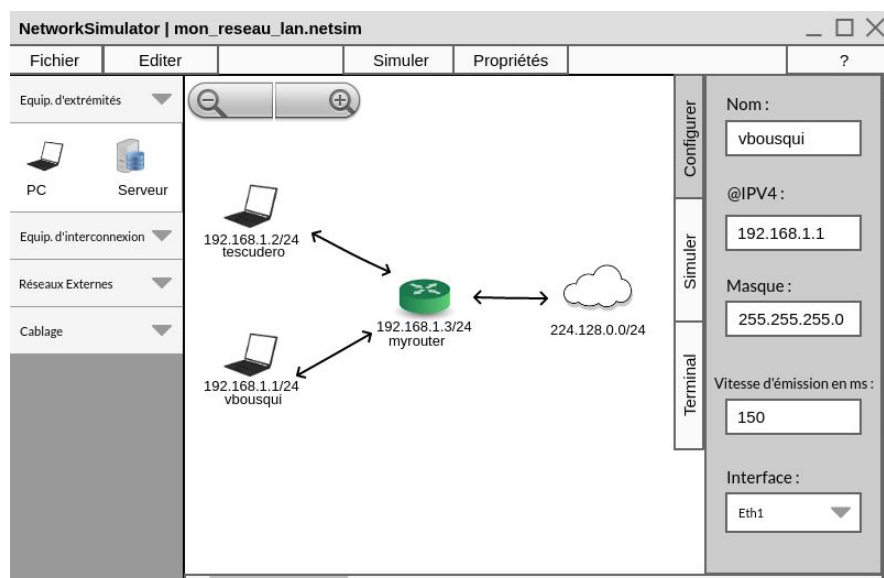
CF : maquette\_ALFRED.epgz

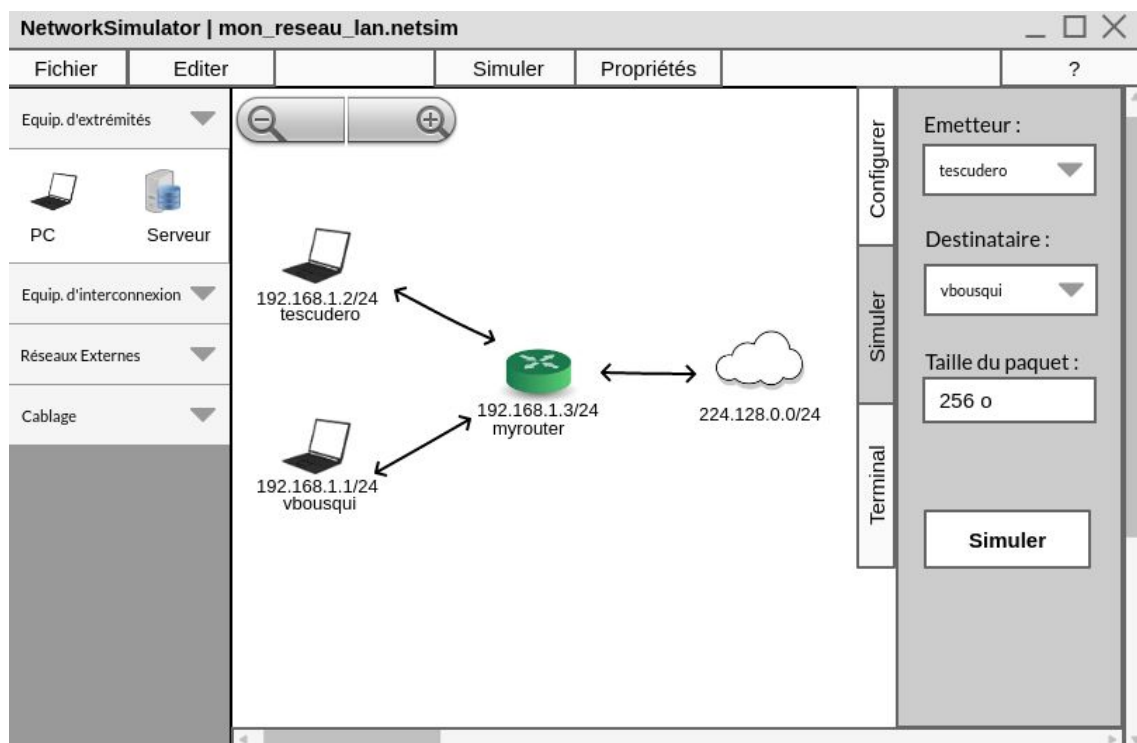
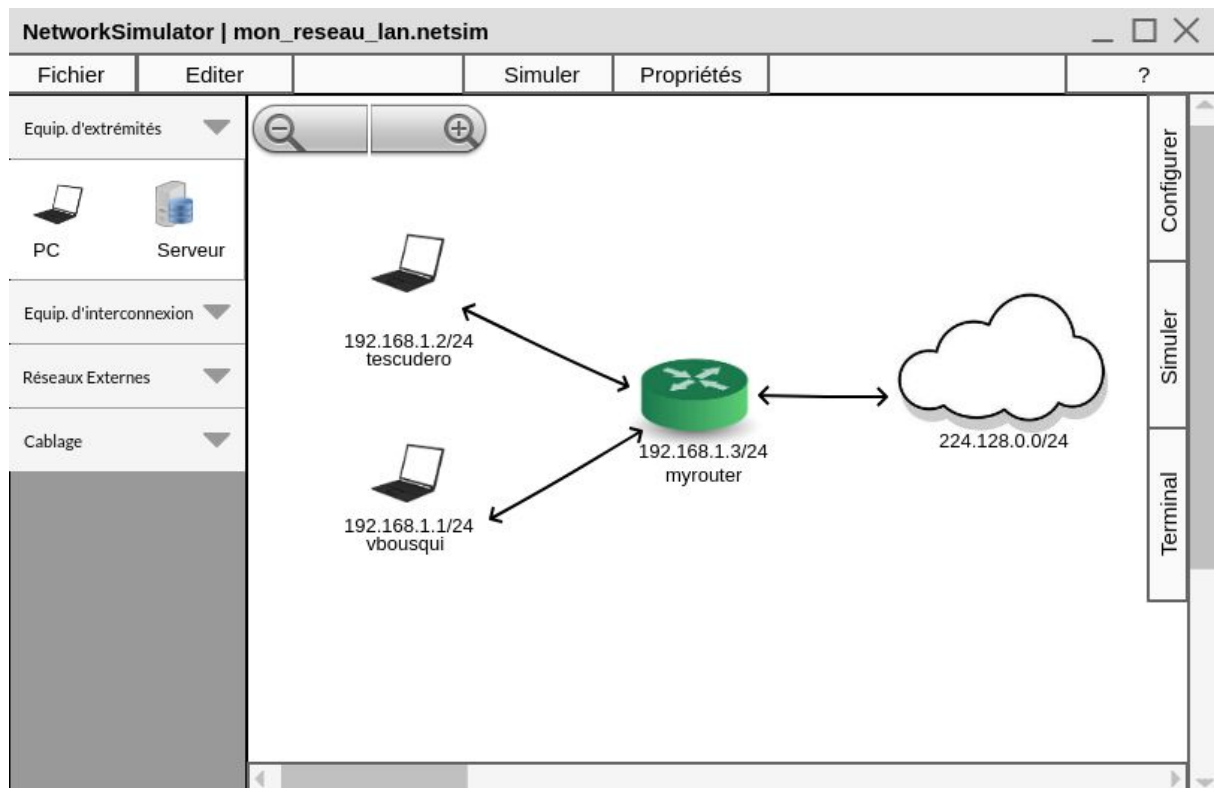
CF : maquette\_BOUSQUIE.epgz

CF : maquette\_TEISSEDRE.epgz

Les maquettes ont été réalisées avec le logiciel PENCIL.

Nous nous sommes ensuite mis d'accord sur une maquette regroupant les avantages de chacune des précédentes :





```
mon_reseau_lan $ > ls
vbousqui
tescudero
myrouter

mon_reseau_lan $ > cd tescudero

mon_reseau_lan / tescudero > ipconfig 192.168.1.2 255.255.255.0

mon_reseau_lan / tescudero > ip
192.168.1.2 /24

mon_reseau_lan / tescudero > ping vbousqui

Ok .

mon_reseau_lan / tescudero > ping mteissedre

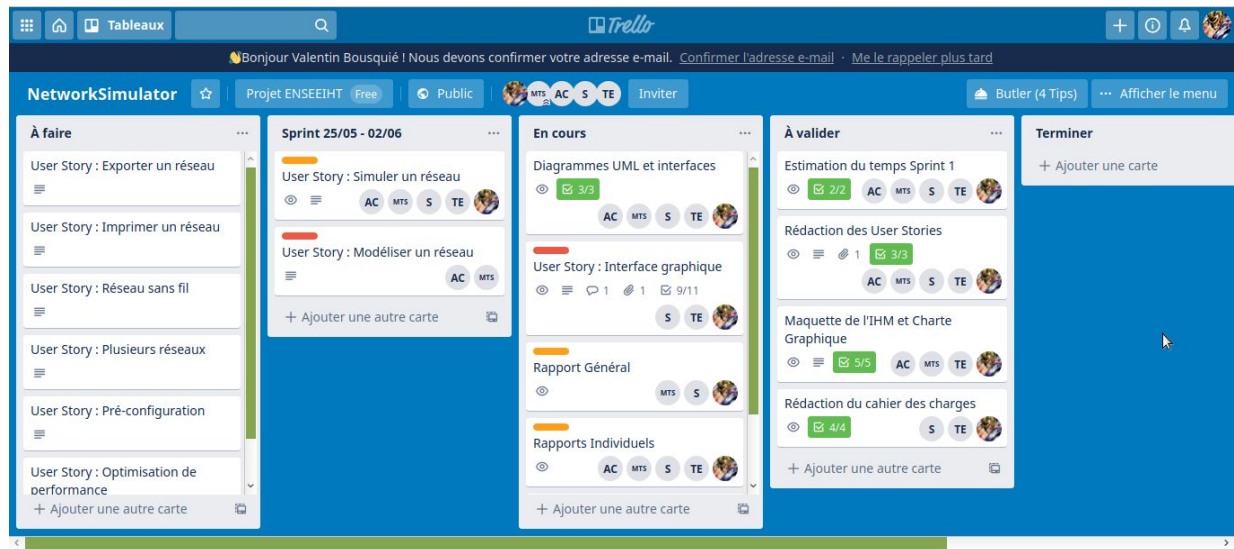
Réseau Inconnu.
```

Propriétés mon_reseau_lan.netsim	
Nombre d'équipements :	<input type="text" value="3"/>
@ de réseau:	<input type="text" value="192.168.1.0"/>
Masque de réseau :	<input type="text" value="255.255.255.0"/>
Nombre de sous-réseaux:	<input type="text" value="0"/>
@ de sous-réseaux:	<input type="text"/>
	<input type="text"/>
<input type="button" value="Ok"/>	

# Gestion de projet

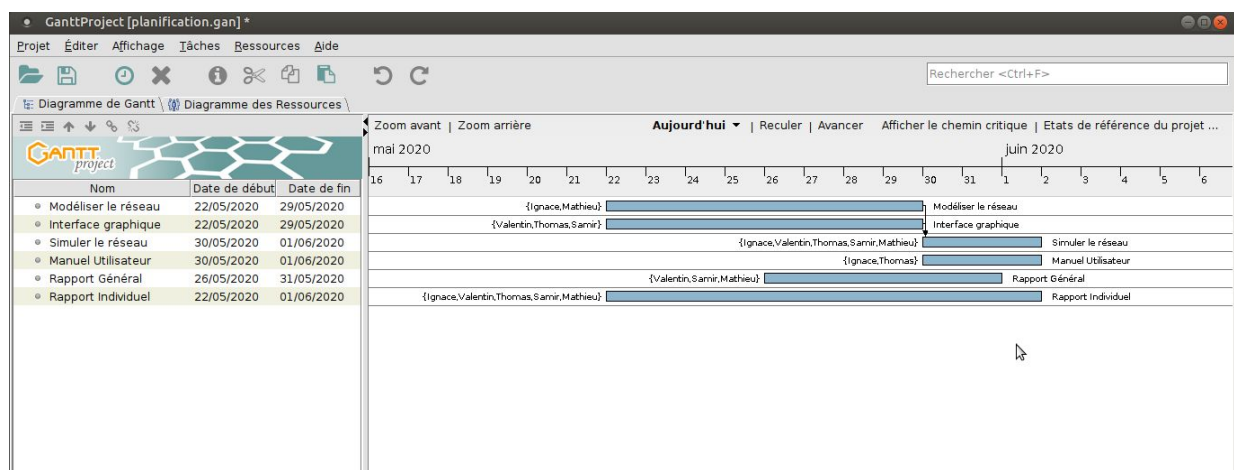
## Outil

Pour organiser notre projet nous avons mis en place un tableau de bord sur Trello. Ainsi on a pu décomposer le projet en sous-tâches, grâce aux users stories. Cet outil nous permet de mettre à jour l'avancement d'une tâche grâce à des TODO List.



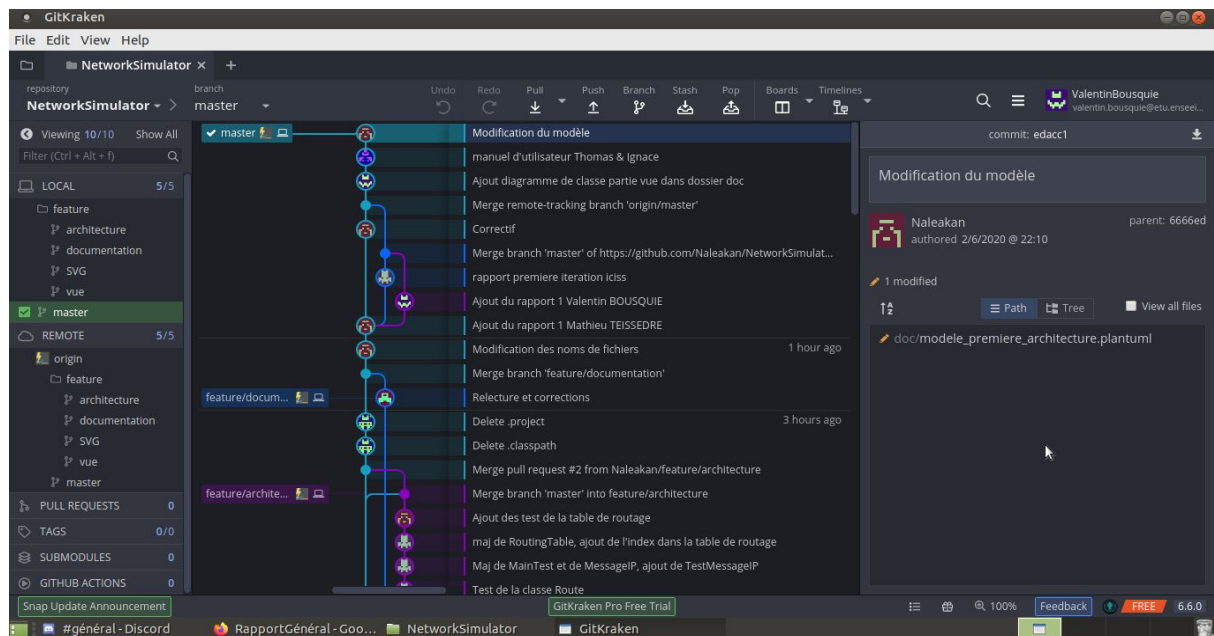
## Diagramme de Gantt

Pour nous aider à planifier le sprint nous avons décidé de créer un diagramme de Gantt avec l'outil GanttProject.



## Suivi de version

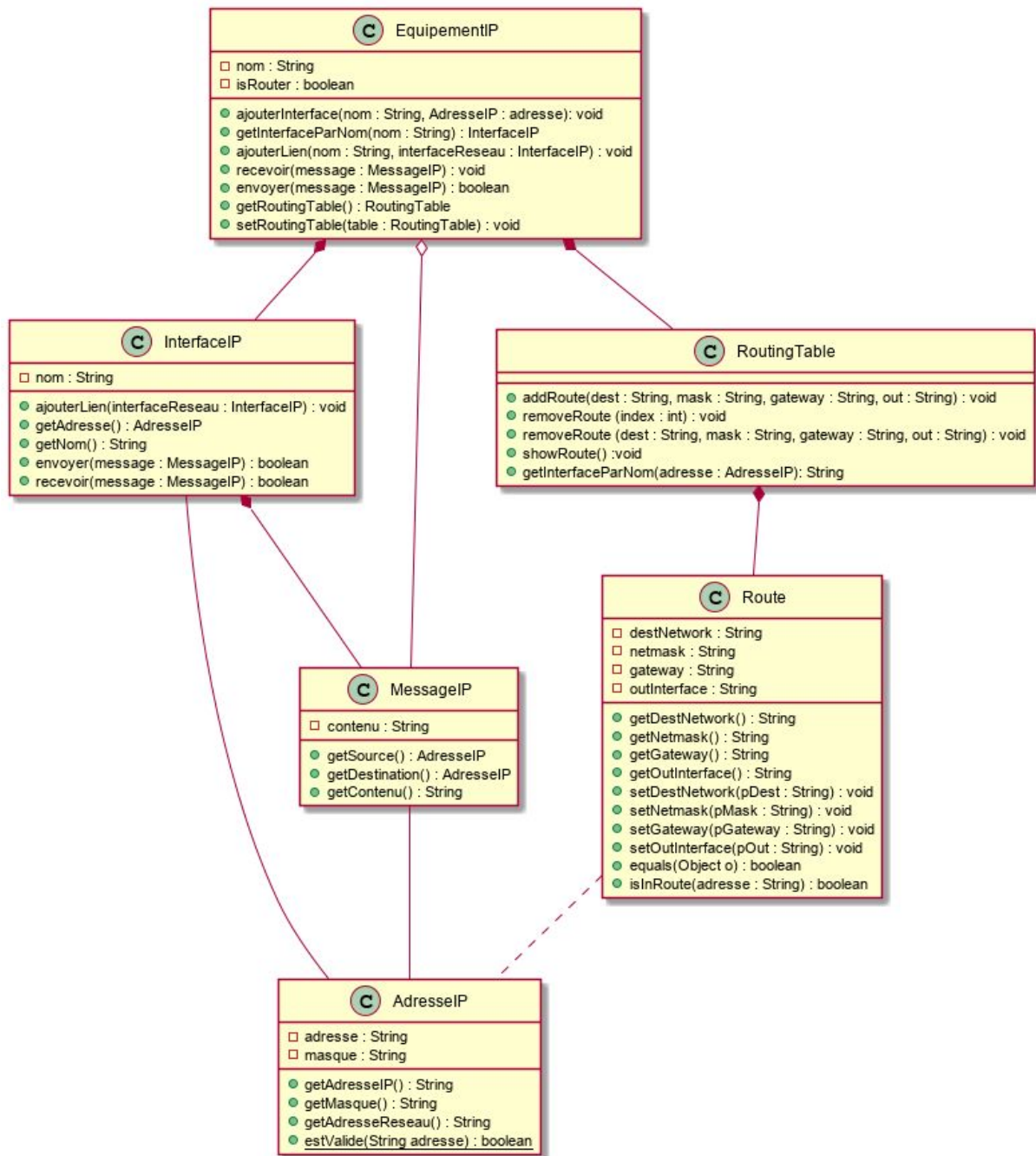
Pour gérer la problématique du développement à plusieurs nous avons décidé de mettre en place un remote sur GitHub. Ensuite grâce à git nous avons pu créer différentes branches en fonction des fonctionnalités. Pour faciliter l'utilisation de git nous avons utilisé un outil graphique permettant d'afficher l'arbre des commits. Il s'agit de GitKraken :



Cet outil graphique a permis de mieux appréhender le système de branche de git. Il permet également de mieux visualiser les fichiers qui ont été modifiés avant de commit.

# Conception du modèle

## Diagramme de classe



Nous avons d'abord commencé à faire un diagramme de classe composé d'interfaces pour représenter le réseau de manière générique, mais nous nous sommes rendus compte que cette conception était bancal. Nous avons donc décidé de nous limiter au niveau IP.

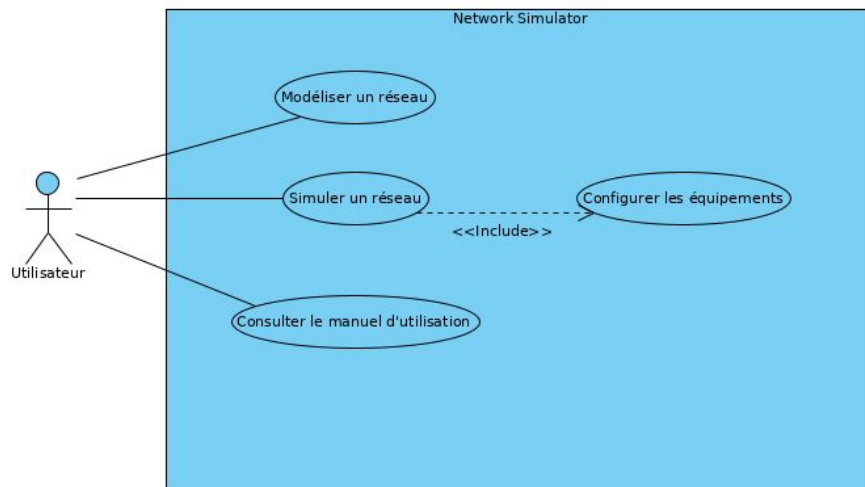
Chaque équipement possède des interfaces et une table de routage. La table de routage permet pour une adresse IP donnée, de trouver une interface qui permet d'y accéder (si elle existe). Les messages IP contiennent les différentes données qui vont être transmises.

Chaque classe a été implémentée en Java, testée avec JUnit et différents tests d'intégration ont été réalisés pour vérifier que toutes les méthodes se coordonnent bien.

Il faudra ajouter des liens entre les interfaces pour les connecter (et ne pas les interconnecter directement)

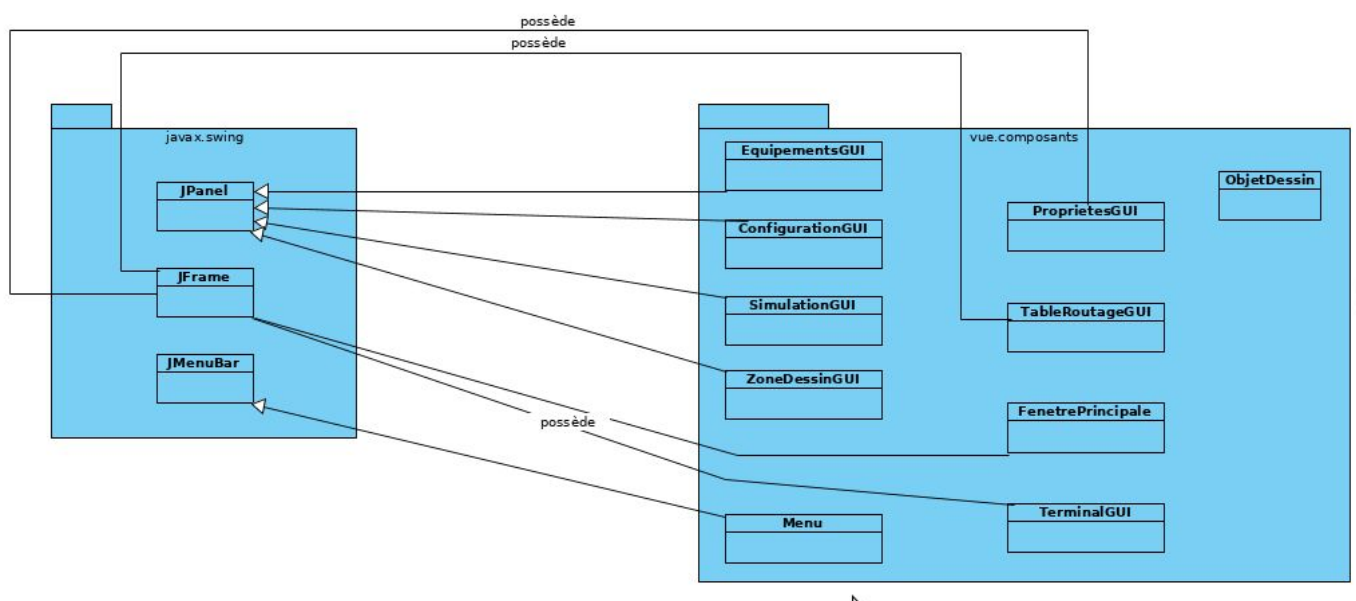
# Conception de la Vue

## Diagramme de cas d'utilisation



Ce diagramme d'utilisation a été réalisé au début du sprint, il représente les fonctionnalités auxquelles un utilisateur aurait accès à la fin de la première itération.

## Diagramme de classes de la Partie Vue





Ce diagramme de classes représente les composants graphiques de notre application. Il est simplifié et met en avant les principales relations entre les composants et les classes fournies par le package javax.swing.

Nous avons décidé de décomposer notre fenêtre principale en plusieurs composants graphiques. Le BorderLayout a été choisi pour le management de la disposition de cette fenêtre. Ainsi chaque position du layout est remplie avec un composant qui hérite (spécialise) la classe JPanel.

La partie “Vue” a été bien avancée au cours de ce sprint. En effet, la grande majorité des composants graphiques a été développée. Au cours du 2nd sprint, les contrôleurs seront approfondis pour relier “Vue” et “Modèle”.

## Bilan 1er Sprint

Le premier sprint nous a permis de définir les grandes lignes de notre application, que ce soit de la représentation du réseau au niveau Vue, que de sa simulation au niveau Modèle.

Ce sprint nous a aussi permis de mettre en place les outils nécessaires pour travailler ensemble efficacement.

Il reste encore beaucoup de choses à améliorer pour le prochain sprint, comme le terminal pour envoyer des commandes, les liens entre les objets graphiques, les envois de messages ou la configuration des équipements.