

DOSSIER JAVA

Projet Multi-User Dungeon

*A l’attention de M. Teyssié et M. Torguet*

**Rendu 1**

30 janvier 2019

BOURAKKEDI – Lapeyrade – OLIVIER

M1 STRI INGE UPSSITECH

118 Route de Narbonne, 31062 Toulouse

Sommaire

[Présentation 2](#_Toc535998923)

[Étape 1 : 2](#_Toc535998924)

[Introduction 2](#_Toc535998925)

[Conception 3](#_Toc535998926)

[Diagramme de cas d’utilisation 3](#_Toc535998927)

[Diagramme de classe participantes 4](#_Toc535998928)

[Diagrammes de séquence 5](#_Toc535998929)

[Résultats 8](#_Toc535998930)

[Commentaires 9](#_Toc535998931)

# Présentation

Le but de ce projet est de créer un jeu de rôle massivement multi-joueur de type MUD (Multi-User Dungeon) en Java avec les Sockets TCP, UDP et/ou RMI. Nous avons personnellement choisi d’utiliser seulement RMI.

Par soucis de granularité, la réalisation de ce projet a été diviser en 5 étapes majeures que l’organisation de ce dossier reprendra. Nous nous attacherons pour chacune d’entre d’elles de les présenter avant de fournir la conception et le code source leur correspondant.

# Étape 1 :

## Introduction

La première étape est la base du programme, elle comprend la mise en place d’un serveur de Donjon et de discussion auquel plusieurs joueurs pourront se connecter en simultanée. Le Donjon est un damier composé de pièces dans lequel les joueurs pourront se déplacer dans les pièces nord, sud, est et ouest.

Après s’être connecté en entrant le nom de leur personnage, les joueurs arrivent dans la pièce de départ et les joueurs seront prévenu si d’autres personnages sont dans la même pièce ou arrivent dans celle-ci. Ils pourront communiquer avec ces derniers via le serveur de discussion.

Notre implémentation se fera donc entièrement sur le principe de RMI (Remote Method Invocation) où les joueurs (clients) se connecteront aux différents serveurs.

## Conception

### Diagramme de cas d’utilisation

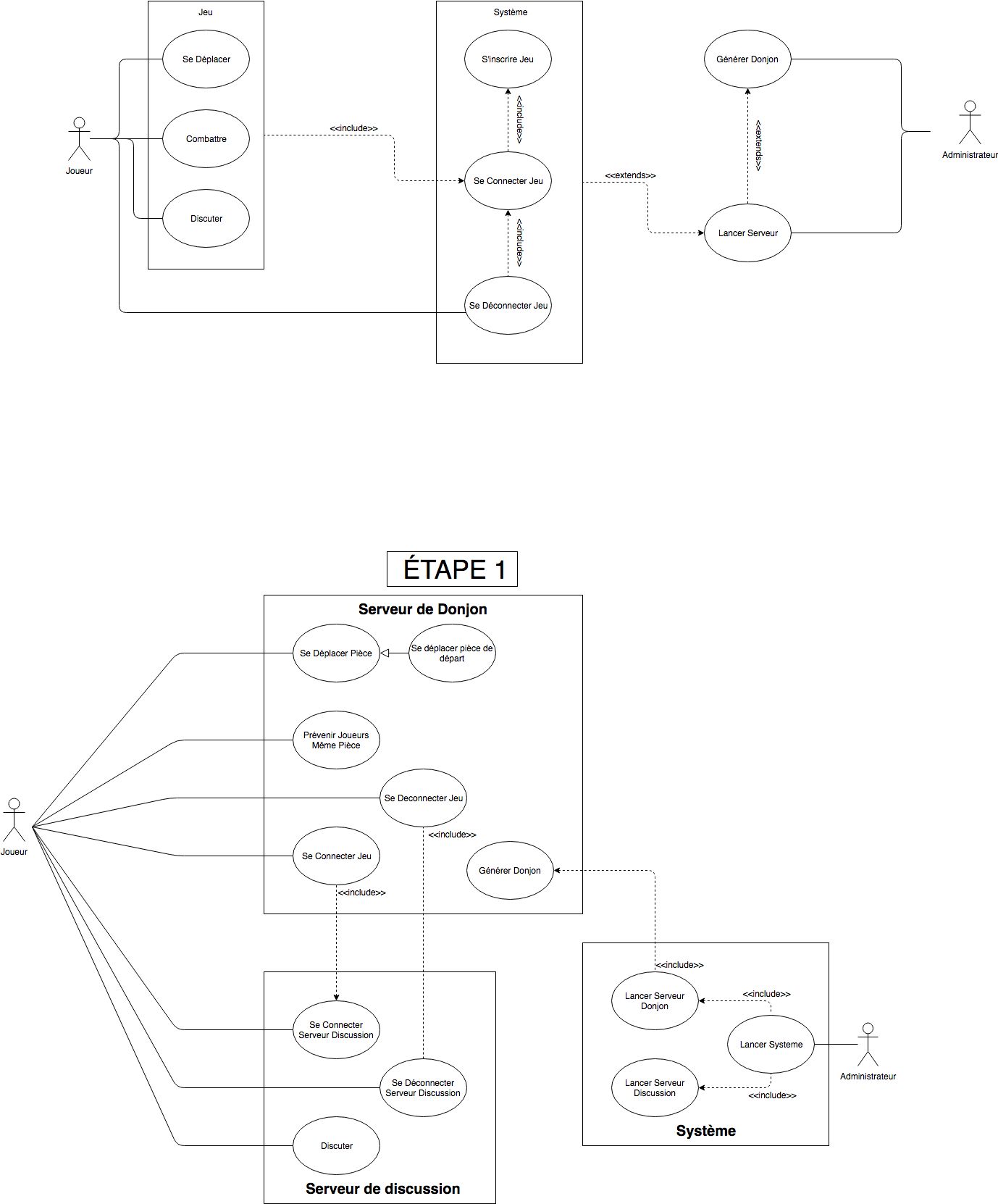
Voici le diagramme de cas d’utilisation de l’étape 1 :

Figure 1: Diagramme de cas d'utilisation de l'étape 1

Nous détaillerons respectivement en diagramme de séquence les cas d’utilisation suivant :

* ***Cas « Se déplacer pièce »***
* ***Cas « Se connecter jeu »***
* ***Cas « Discuter »***
* ***Cas « Lancer Système »***

Ces derniers reprennent en effet les différentes fonctionnalités principales de la première étape.

### Diagramme de classe participantes

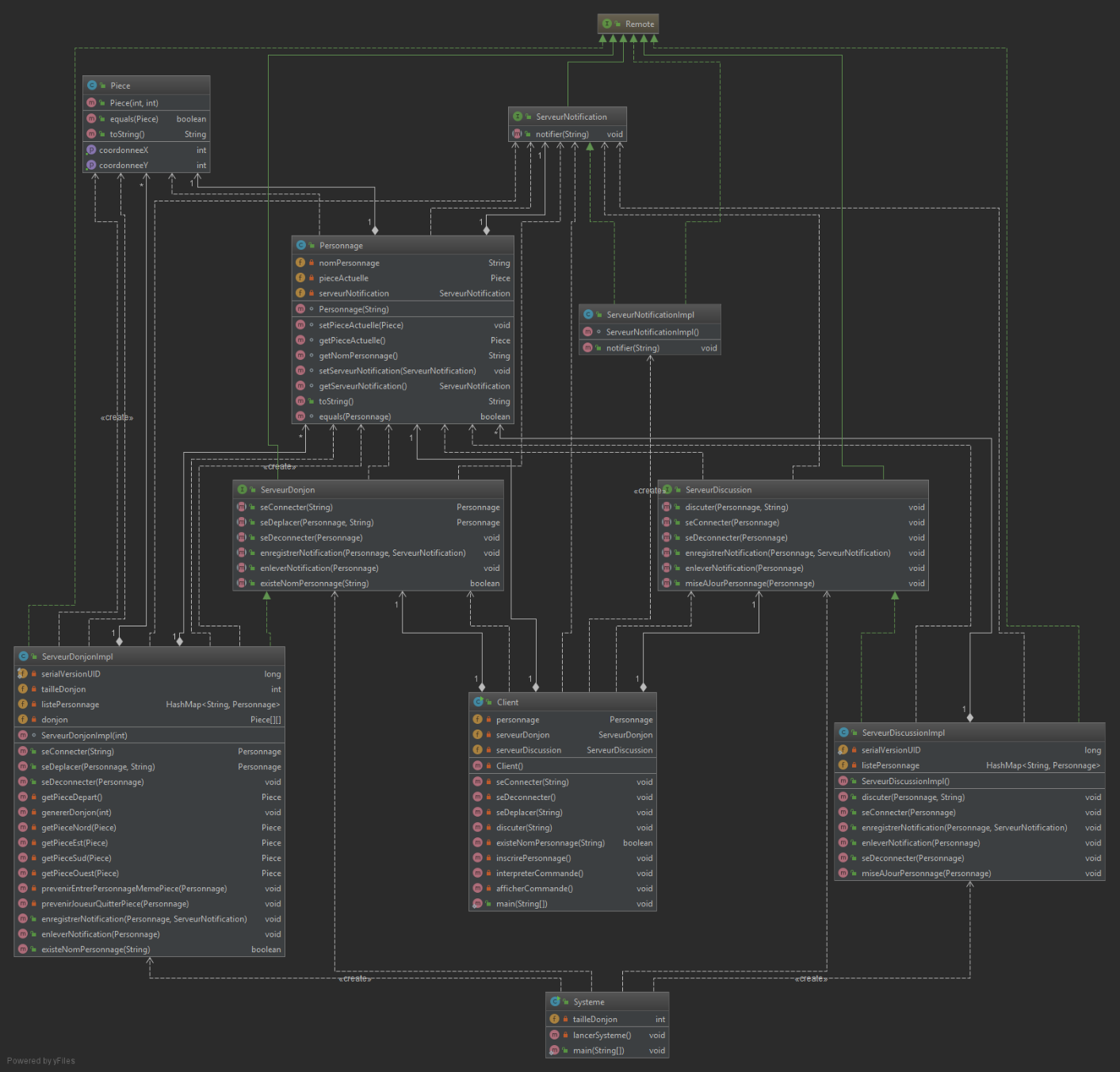
 Voici le diagramme de classe participantes de notre étape 1 illustrant notre implémentation de notre Diagramme de cas d’utilisation. On y retrouve les différentes classes et leurs relations respective. On remarque la présence de l’objet « Remote » nécessaire à l’implémentation de RMI.

Figure 2: Diagramme de classes participantes de l'étape 1

### Diagrammes de séquence

#### **Cas « Se déplacer pièce »**

Figure 3: Diagramme de séquence du cas "Se déplacer"

#### **Cas « Se connecter jeu »**

Figure 4: Diagramme de séquence du cas "Se connecter"

#### **Cas « Discuter »**

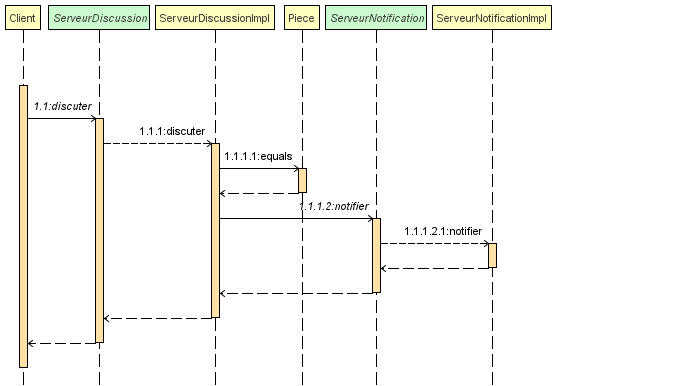
***Cas « Lancer Système »***

Figure 5: Diagramme de séquence du cas "Discuter"

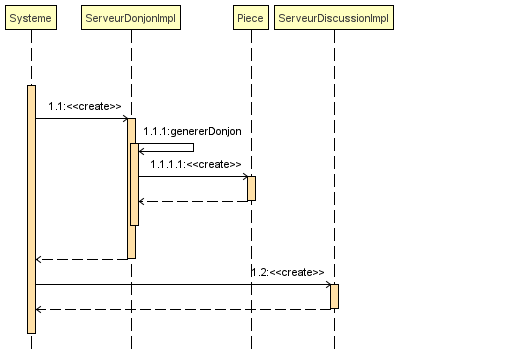
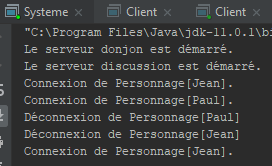


Figure 6: Diagramme de séquence du cas "Lancer Système"

## Résultats



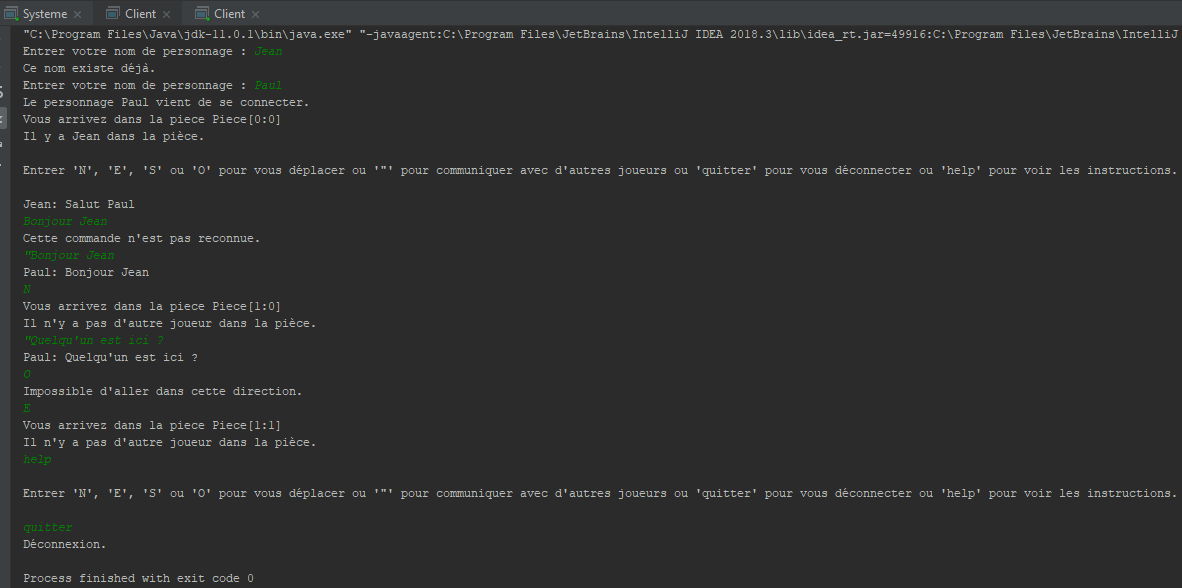
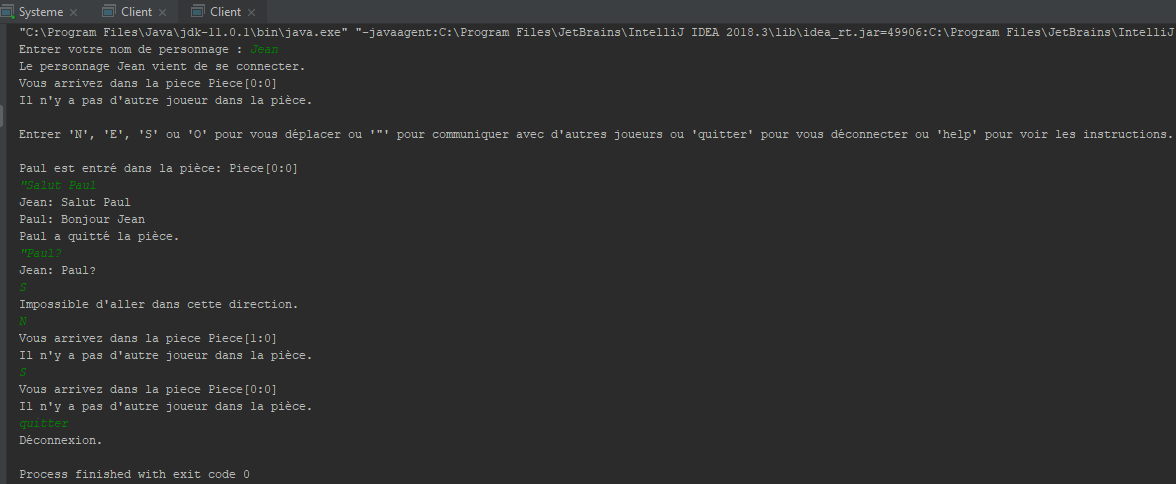


Figure : Capture d'une exécution d'un premier client de l'étape 1

Figure : Capture d'une exécution d'un second client de l'étape 1

Figure : Capture d'une exécution du Système de l'étape 1

### Commentaires

Sur la Figure 7, nous pouvons observer que nous pouvons connecter plusieurs personnages en simultanées. De plus, lorsqu’un personnage se déconnecte, il peut être reconnecter avec le même nom.

Sur la Figure 8, nous voyons que le peronnage « Jean » est connecté dans la pièce de départ « 0,0 ». Quand, un second personnage « Paul » se connecte dans la même pièce de départ, le premier est bien prévenu. Les deux personnages peuvent alors communiquer entre eux, Jean reçoit bien le message de Paul. De même quand, ce dernier quitte la pièce, le personnage restant est notifié.

Après avoir changé de pièce, Paul ne peut plus recevoir de message envoyé par Jean comme nous pourvons le vérifier Figure 9.

Concernant les directions, la pièce de départ se trouvant à l’extrémité sud-ouest du damier, le personnage ne peut aller au Sud ni à l’est depuis la pièce de départ, et un message l’en notifie bien. Si le personnage va dans une direction où une pièce existe, le serveur lui notifie de la pièce où il arrive et si d’autre joueurs sont présents.

Enfin si le joueur entre la commande « quitter », il est bien déconnecter du serveur et le processus est bien terminé. On peut alors se reconnecter avec son nom, et seulement alors. Comme nous pouvons le voir au déut de la Figure 9, Jean étant connecté, on ne peut pas se connecter avec son nom.

On observe de plus sur cette dernière capture que pour communiquer avec d’autres personnages, le joueur doit bien précéder son message par « ‘’ ». Sinon, le client est notifié que la commande n’est pas reconnue.

Enfin, nous pouvons constater que lorsque le joueur entre la commande « help », les commandes possibles lui sont bien rappelés.

[Figure 1: Diagramme de cas d'utilisation de l'étape 1 3](file:///C:\Users\sylva\Dropbox\upssitech\m1\java\projet_mud_m1\docs\rendu_bourakkadi_lapeyrade_olivier_v1.docx#_Toc535998914)

[Figure 2: Diagramme de classes participantes de l'étape 1 4](file:///C:\Users\sylva\Dropbox\upssitech\m1\java\projet_mud_m1\docs\rendu_bourakkadi_lapeyrade_olivier_v1.docx#_Toc535998915)

[Figure 3: Diagramme de séquence du cas "Se déplacer" 5](file:///C:\Users\sylva\Dropbox\upssitech\m1\java\projet_mud_m1\docs\rendu_bourakkadi_lapeyrade_olivier_v1.docx#_Toc535998916)

[Figure 4: Diagramme de séquence du cas "Se connecter" 6](file:///C:\Users\sylva\Dropbox\upssitech\m1\java\projet_mud_m1\docs\rendu_bourakkadi_lapeyrade_olivier_v1.docx#_Toc535998917)

[Figure 5: Diagramme de séquence du cas "Discuter" 7](file:///C:\Users\sylva\Dropbox\upssitech\m1\java\projet_mud_m1\docs\rendu_bourakkadi_lapeyrade_olivier_v1.docx#_Toc535998918)

[Figure 6: Diagramme de séquence du cas "Lancer Système" 7](file:///C:\Users\sylva\Dropbox\upssitech\m1\java\projet_mud_m1\docs\rendu_bourakkadi_lapeyrade_olivier_v1.docx#_Toc535998919)

[Figure 8: Capture d'une exécution d'un premier client de l'étape 1 8](file:///C:\Users\sylva\Dropbox\upssitech\m1\java\projet_mud_m1\docs\rendu_bourakkadi_lapeyrade_olivier_v1.docx#_Toc535998920)

[Figure 9: Capture d'une exécution d'un second client de l'étape 1 8](file:///C:\Users\sylva\Dropbox\upssitech\m1\java\projet_mud_m1\docs\rendu_bourakkadi_lapeyrade_olivier_v1.docx#_Toc535998921)

[Figure 7: Capture d'une exécution du Système de l'étape 1 8](file:///C:\Users\sylva\Dropbox\upssitech\m1\java\projet_mud_m1\docs\rendu_bourakkadi_lapeyrade_olivier_v1.docx#_Toc535998922)