

# Rapport TP: « EventManadg mentProject »

INFO641

BALLAZ Corentin – BURILLON William

2018/2019



**POLYTECH<sup>®</sup>**



**UNIVERSITÉ  
SAVOIE  
MONT BLANC**

## I. Introduction

Ce rapport présente le rendu du projet « EventManadgmentProject » réalisé en cours d'INFO641 sur 4 séances de 4h.

Rappelons rapidement le sujet : l'objectif est de réaliser une application style messagerie instantanée qui permet à plusieurs bavards (utilisateurs) de discuter entre eux. Ceci se fait par l'intermédiaire d'un concierge, qui gère tous les bavards et à la vue sur tous les messages. On peut retrouver plusieurs concierges avec plusieurs bavards, en sachant que les bavards peuvent discuter entre eux que s'ils appartiennent au même concierge. La communication entre les utilisateurs se fera en utilisant le mécanisme d'évènements (justification du titre du projet). Le gestionnaire aura la charge de récupérer un message envoyé et de le transmettre aux autres bavards.

Dans ce rapport, nous présenterons la conception globale du système en mettant en avant nos choix de conception. Nous verrons ensuite les fonctionnalités implémentées qui tournent, puis celle imaginées mais qui ne tournent pas où qui ne fonctionnent pas comme souhaitées. Nous présenterons enfin la démarche pour utiliser le programme, et pour terminer un bilan personnel sur ce projet.

## II. Conception globale du système

Le concept de ce système est l'implémentation et l'utilisation des interfaces afin de gérer des messages entre plusieurs personnes.

Dans le fonctionnement du modèle, nous pouvons distinguer deux classes principales : les Bavards (utilisateurs) et les concierges (administrateurs/gestionnaires). Ces deux classes implémentent toutes les deux l'interface PapotagesListeners. En effet, ces deux classes vont recevoir des messages (PapotageEvent), seulement ils ne vont pas les gérer de la même façon. Quand un concierge va recevoir un PapotageEvent, il va le transmettre à tous les bavards qui sont abonnés à lui, excepté la source. Alors que lorsque le Bavard va recevoir le PapotageEvent, il va le stocker dans une liste de message reçu. De plus, le Bavard sera capable de générer un PapotageEvent.

A la base, le sujet que nous devions faire était de créer un système avec un seul concierge gérant des bavards. Nous avons décidé d'aller un peu plus loin et de créer un système permettant d'accueillir plusieurs concierges. De ce fait, le concierge est "le routeur" permettant de permuter les différents messages seulement aux Bavards qui sont abonnés à lui. En outre, notre système permet de créer différents groupes de discussion, permettant ainsi de discuter qu'avec les personnes concernées par un message.

Au niveau de l'interface graphique de ce système, nous avons décidé d'utiliser Java Swing. Il est constitué d'un JFrame principale permettant la gestion globale du système. Ce JFrame possède différents JPanels, relatifs à l'action désiré (souscription, connexion ou gestion).

A l'exception du JPanel de souscription, les autres JPanel, une fois le formulaire remplis, génère un nouveau JFrame relatifs au concierge ou au Bavard qui s'est connecté. Le fonctionnement des différents panels sera détaillé dans la section suivante.

Globalement, nous avons voulu développer une interface graphique la plus intuitive possible et qui ressemble à ce qui se fait actuellement, sans réellement d'innovation.

### III. Fonctionnalités

#### a. Fonctionnalités implémentées qui fonctionnent

Voici la liste des différentes fonctionnalités implémentées dans notre programme :

- Interface d'accueil avec 3 boutons : « Subscribe », « Login », « Conciergerie ».
- Dans l'interface « Subscribe », possibilité d'ajouter un bavard : on demande son nom, un mot de passe, sa date de naissance ainsi que le concierge auquel il est relié (il faut sélectionner un concierge qui existe).
- Dans l'interface « Login », on peut rentrer son nom et son mot de passe pour se connecter. Une fenêtre s'ouvre dans laquelle on retrouve la liste des messages reçus par le bavard, où on peut regarder un message en particulier. On peut également envoyer un message avec un objet et du texte. Un bouton Refresh est disponible en bas de la boîte de réception afin de mettre à jour la messagerie, et un bouton d'envoi est disponible en bas de la saisie de texte pour envoyer un message.
- On peut donc ouvrir une fenêtre de messagerie instantanée pour chaque bavard qui existe. Si on ferme une fenêtre utilisateur, on peut l'ouvrir de nouveau et tous les messages sont toujours là.
- Dans l'interface « Conciergerie », on retrouve une première fenêtre pour se connecter en tant que concierge. On a alors accès à la liste des bavards qui appartient au concierge. On peut en sélectionner un et le supprimer.

#### b. Fonctionnalités imaginées mais non réalisées

Voici la liste des différentes fonctionnalités imaginées et des améliorations possibles à mettre en œuvre si nous avons eu plus de temps :

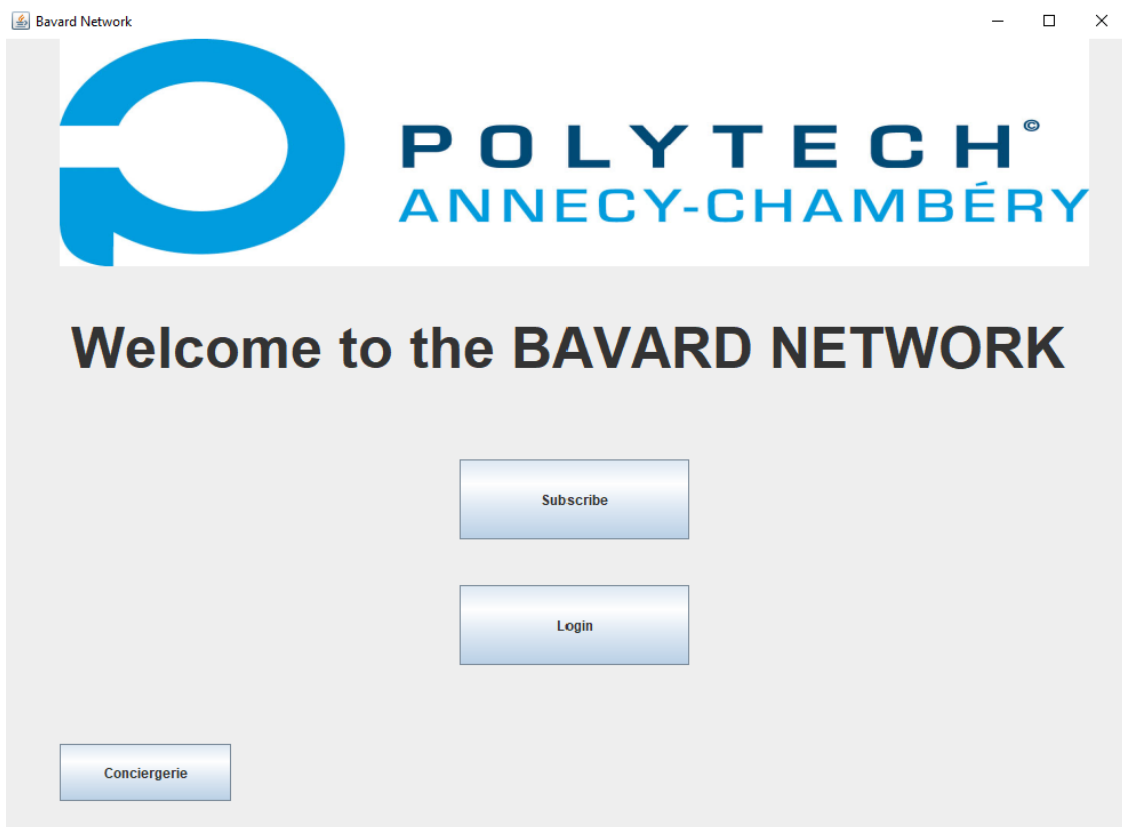
- Mettre en place dans l'interface des utilisateurs une partie où l'on peut voir le nom des utilisateurs connectés (qui appartiennent au même concierge) et ceux qui existent mais qui ne sont pas connectés (même principe qu'une messagerie instantanée).
- Faire la même chose mais dans l'interface des concierges où un concierge peut voir les messages de tous les bavards qu'il contient.
- Sauvegarder tous les messages des différents bavards appartenant aux différents concierges dans un fichier (peut-être .txt). On pourrait donc récupérer tous les messages existant après avoir fermé la fenêtre principal (avec

l'avancement actuel, lorsqu'on ferme la fenêtre principale, on perd tous les messages qui étaient écrit avant).

- Changer les caractères dans les zones de saisie de mot de passe par des astérisques.
- Modifier l'affichage graphique afin de rendre l'interface plus « jolie » (surtout sur la partie boîte de réception et envoi de messages, possibilité d'utiliser la bibliothèque graphique java fx).

#### IV. Fonctionnement et utilisation du programme

Lorsqu'on lance le programme, on se retrouve avec la fenêtre d'accueil suivante :



On a trois boutons disponibles. « Subscribe » permet d'ajouter un nouvel utilisateur (nom, mot de passe, date de naissance et concierge auquel il appartient). On a cette fenêtre ci :

Bavard Network

## Sign Up

**Name :**

**Password :**

**Birthday :**

**Caretaker :**   

Coco

Bubu

On clique sur le bouton « Submit » et l'utilisateur est ajouté. On clique sur le bouton « Back » pour revenir au menu. On a ensuite le bouton « Login » qui permet de se connecter en tant qu'utilisateur. On rentre son nom et son mot de passe comme ci-dessous :

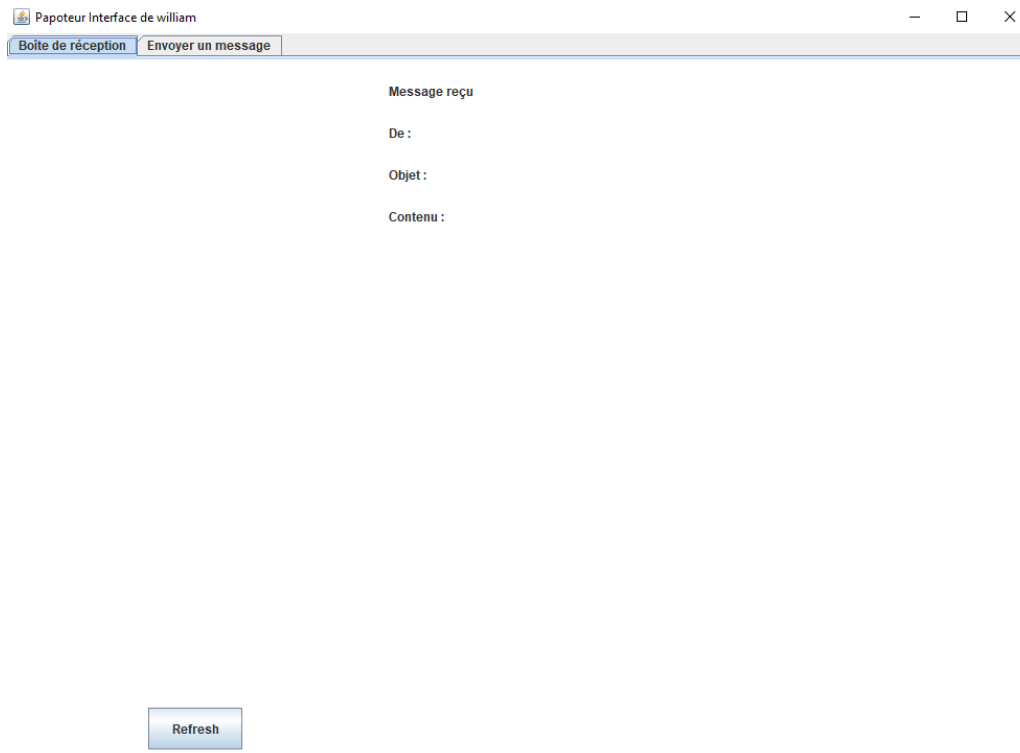
Bavard Network

## Sign in

**Name :**

**Password :**

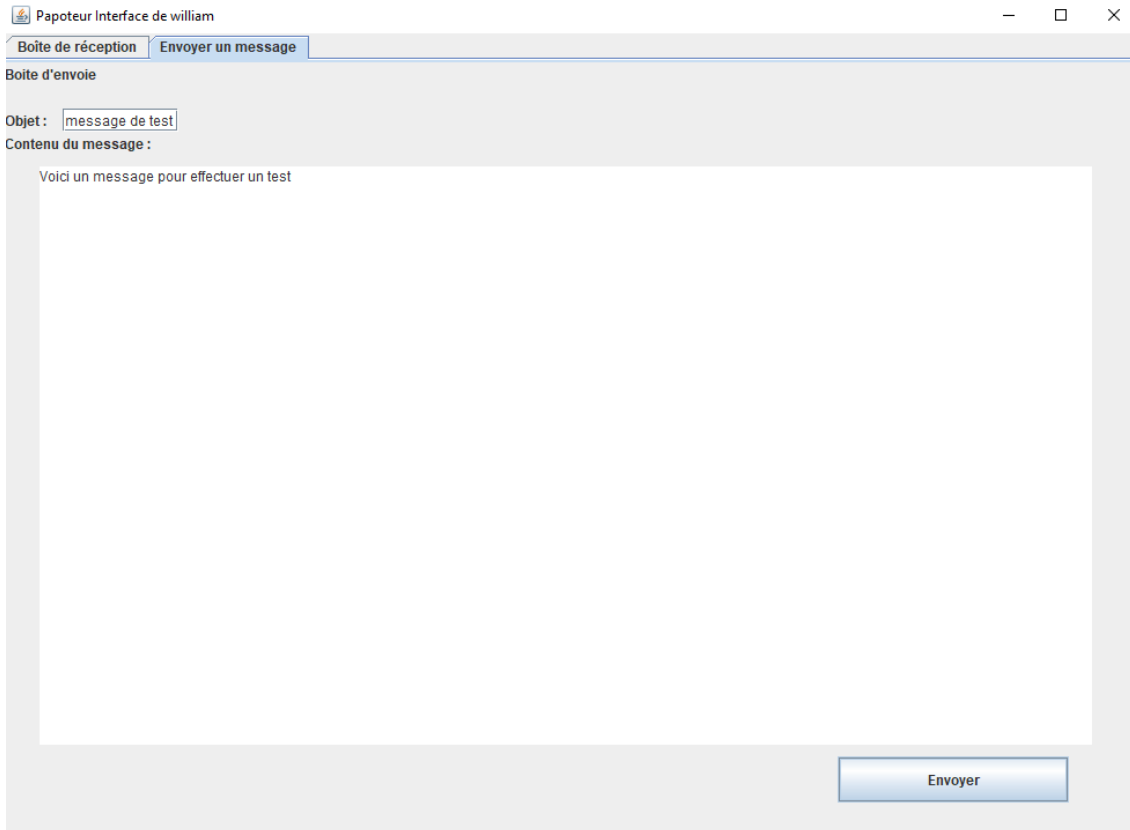
En cliquant sur le bouton « Valider », l'utilisateur est connecté et sa fenêtre s'ouvre :



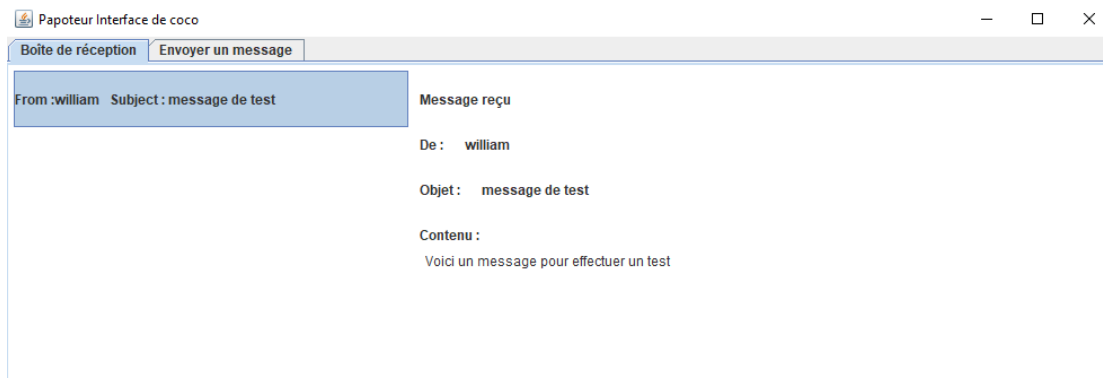
On retrouve ici une fenêtre avec deux onglets : une boîte de réception des messages des autres utilisateurs appartenant au même concierge, et un onglet envoyer un message. Si un autre utilisateur envoie un ou plusieurs messages, il suffit de cliquer sur le bouton « refresh » en bas pour les retrouver dans la boîte de réception :



On va ensuite pouvoir envoyer un message aux autres utilisateurs appartenant au même concierge en allant dans l'onglet « envoyer un message ». On retrouve un champ de saisie de texte pour l'objet du message, puis pour le corps du message, comme ceci :



On va pouvoir envoyer un message en appuyant sur le bouton « envoyer ». Les autres utilisateurs connectés vont donc recevoir le message :



## V. Bilan personnel sur le projet

### a. Analyse des principaux apprentissages

Ce TP nous a permis de mettre en place ce qui a été appris au niveau du traitement des événements : la gestion des événements, des écouteurs, et des relations entre tous ces éléments. De plus, l'aspect graphique du rendu est un point que nous avons tout au long du projet essayé d'améliorer. Nous avons beaucoup étudié le modèle MVC l'année précédente, et ce TP fut un bon moyen pour nous de remettre en œuvre toutes ces connaissances.

## b. Organisation du binôme

Au niveau de l'organisation, nous avons tout d'abord travaillé à deux sur la première séance, pour construire la base du modèle à événements. Etant donné que c'était la première fois que nous l'étudions, nous avons préféré travailler dessus à deux de manière qu'il soit plus clair et mieux compris.

Ensuite, pour le temps restant, il nous restait la partie interface graphique et la gestion des écouteurs. Comme nous ne partions pas de rien, nous avons partagé les tâches : William a travaillé sur les premières fenêtres d'accueil (fenêtre d'accueil, subscribe, login et conciergerie) et Corentin a travaillé sur la fenêtre dédiée aux bavards (boite de réception et envoie des messages).