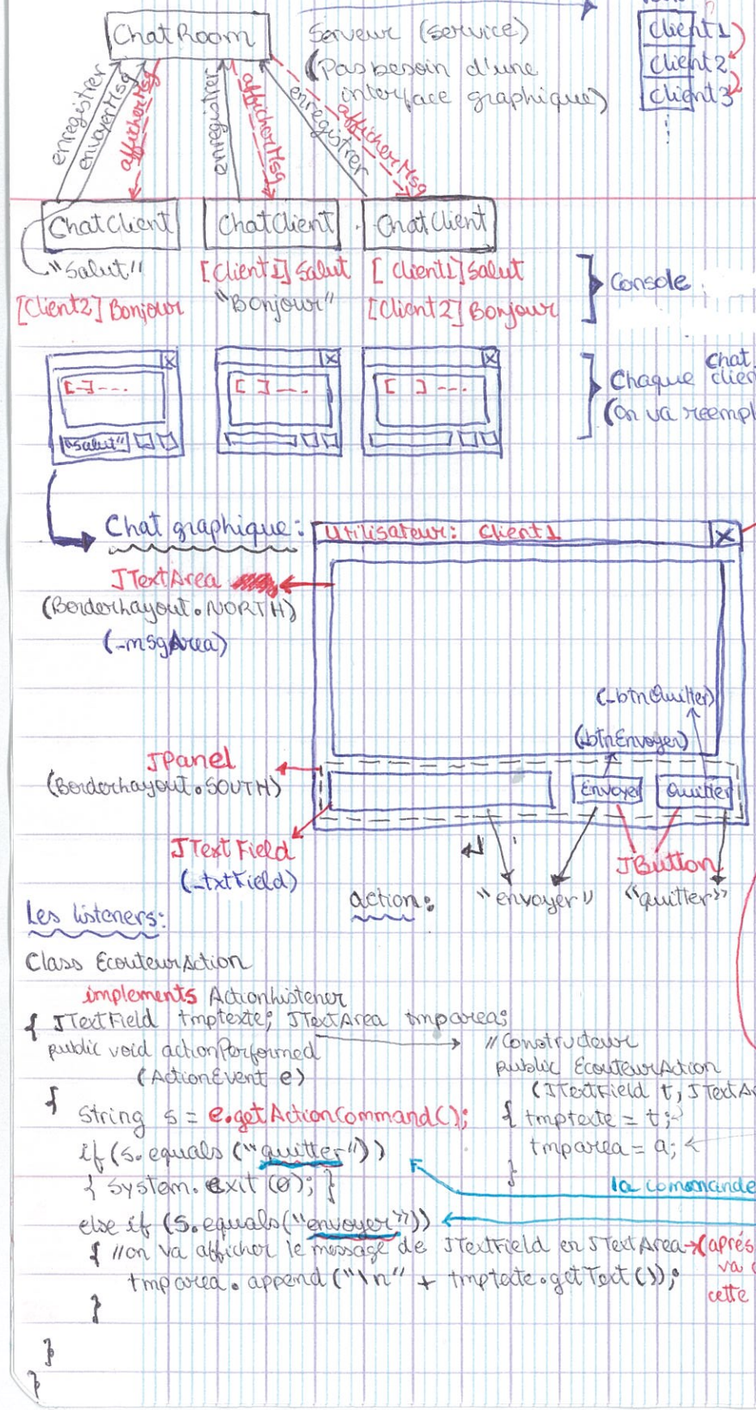



Class GUIClientSwing extends JFrame

```
//Attributs
JTextArea -msgArea;
...
//Constructeur
public GUIClientSwing(String name)
{ super("Utilisateur: " + name);
  this.add(-msgArea, BorderLayout.NORTH);
  ...
  //les listeners.
}
//Pour tester:
public static void main (String args[])
{ JFrame fenetre = new GUIClientSwing("Client1");
}
}
```

Les listeners:

```
Class EcouteurAction
   implements ActionListener
{ JTextField tmptexte; JTextArea tmparea;
  public void actionPerformed
        (ActionEvent e)
  { String s = e.getActionCommand();
    if (s.equals("quitter"))
    { System.exit(0); }
    else if (s.equals("envoyer"))
    { //on va afficher le message de JTextField en JTextArea
      tmparea.append("\n" + tmptexte.getText());
    }
  }
}
```

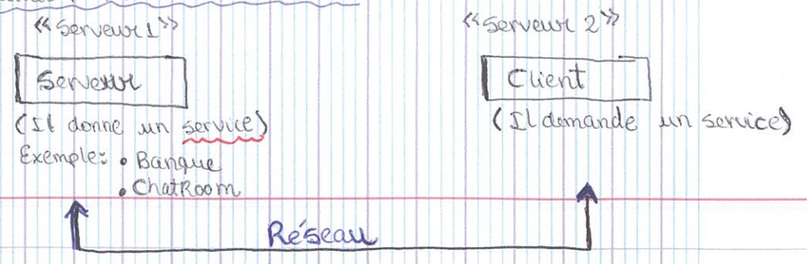
```
//Constructeur
public EcouteurAction
  (JTextField t, JTextArea a)
{ tmptexte = t;
  tmparea = a;
}
```

la commande (après on va changer cette logique)

```
ActionListener -eA = new EcouteurAction
  (-txtField, -msgArea);
-btnQuitter.setActionCommand("quitter");
-btnQuitter.addActionListener(-eA);
-btnEnvoyer.setActionCommand("envoyer");
-btnEnvoyer.addActionListener(-eA);
-txtField.setActionCommand("envoyer");
-txtField.addActionListener(-eA);
```

le même écouteur

## RMI: Pour utiliser les objets à distance




### Serveur:

① Il faut créer les interfaces.
Par exemple: Banque, Compte

② Il faut implementer les interfaces (pas 1)
Par exemple: BanqueImpl, CompteImpl

③ RMI ne comprends pas BanqueImpl.java
Il faut créer les fichier stub et skeleton des classes qui DONNENT le service.
rmic BanqueImpl → Classe BanqueImpl donne le service, par contre CompteImpl ne donne pas un service. BanqueImpl utilise CompteImpl seulement
Résultat: un fichier BanqueImpl_stub.java

④ La classe "Serveur" pour publier le service

```
obj = new BanqueImpl();
Naming.rebind("nomService", obj);
```

annuaire RMI → nomService → obj
(le objet est gardé et attends qu'un client l'utilise)

⑤ Démarrer le RMI
rmiregistry &

⑥ Executer le «Serveur»
java -D.java.security.policy=java.policy Serveur
(Classe (pas 4))
option pour donner une politique sécurité
java.policy: fichier qui donne la politique de sécurité

### Client:

⑦ La classe "Client" pour chercher et demander le service.