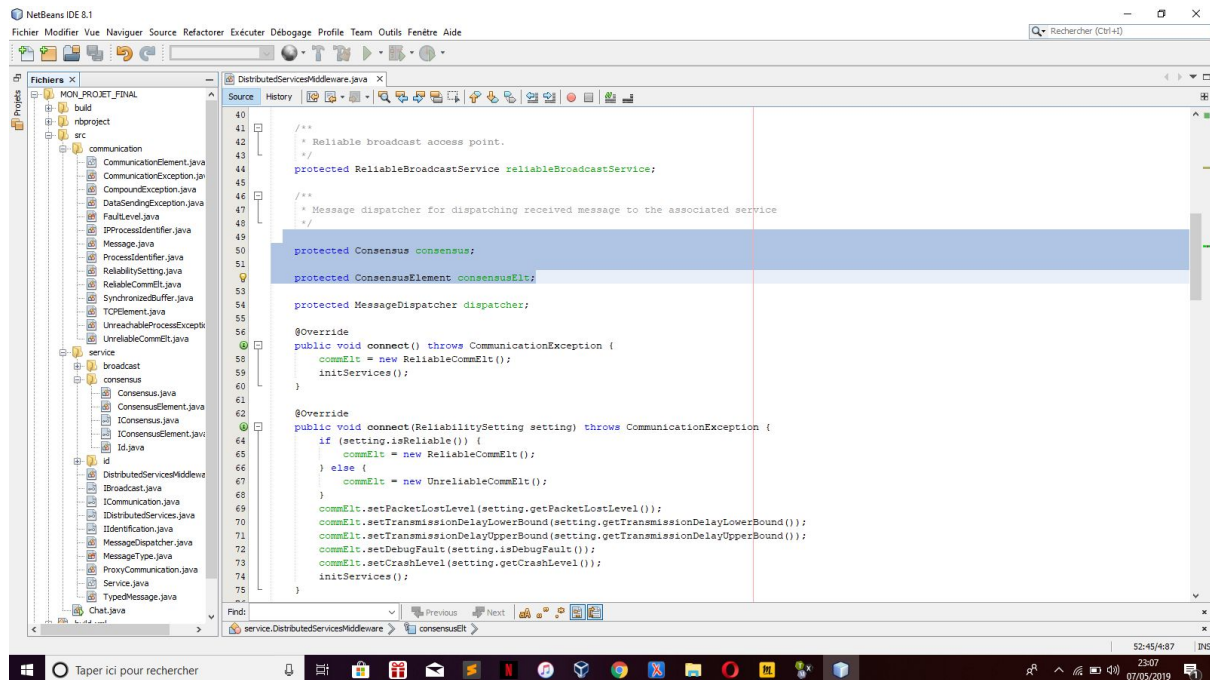


# ***RAPPORT PROJET SPÉCIFICATION DES SYSTÈMES DISTRIBUÉS***

Auteurs : Bagré Gnebehi & Kouamé-Kodia N'tanouan

## **Création du sous package *service.consensus*, des classes et interfaces pour gérer le service *Consensus***



IConsensusElement()  
IConsensus();  
et leur implémentation

ConsensusElement();  
Consensus();

## Présentation de la classe ***DistributedServicesMiddleware***

```
protected Consensus consensus;  
  
protected ConsensusElement consensusElt;  
consensus.initialize(dispatcher, commElt, MessageType.CONSENSUS);  
consensus.setIdentificationService(idService);  
consensus = new Consensus();
```

## Ajout d'un nouveau type dans ***MessageType***

```
public enum MessageType {  
    NONE,  
    IDENTIFICATION,  
    BASIC_BROADCAST,  
    RELIABLE_BROADCAST,  
    CONSENSUS  
}
```

## Modification de ***DistributedServices***

```
public IConsensus getConsensusService();
```

## **Modification du Chat pour lancer le Consensus**

```
consensusService = services.getConsensusService();

consensusService.initConsensus(true);

if (message.equals("consensus")) {
    System.out.println("Faites votre vote
!");
    if(message.equals("true")){

consensusService.initConsensus(decision);
        ConsensusElement elt = new
ConsensusElement();
        consensusService.register(elt);
    }
    else
    {
        System.out.println("Veuillez Saisie
true ou false !");
    }
}
```

## **Fonctionnement de notre Projet**

-> On lance **IdentificationServer.java** puis Chat.java  
Si un processus veut rejoindre le *consensus* il saisie le mot "consensus" , il rejoint automatiquement le service.  
Il saisie true ou false. Tous les processus votent et à la fin on affiche la décision majoritaire.

```

int nbT = 0;
    int nbF = 0;
    boolean decision = false;
    String message = null;
    BufferedReader input = new BufferedReader(new
InputStreamReader(System.in));
    consensusService.initConsensus(true);

    while (true) {
        // read the user entry
        System.out.println("Commande : Saisir
'consensus' pour rejoindre le service");
        System.out.println("Votre message ('end' pour
finir) :");
        try {
            message = input.readLine();
        } catch (IOException ex) {
            System.err.println("Mauvaise Saisie : " +
ex);
            System.exit(2);
        }

        if (message.equals("consensus")) {
            System.out.println("Faites votre vote
!");
            if(message.equals("true")){

                nbT = nbT +1;

consensusService.initConsensus(decision);
                ConsensusElement elt = new
ConsensusElement();
                consensusService.register(elt);
            }
            else
            {
                nbF = nbF + 1;
                System.out.println("Veuillez Saisie
true ou false !");
            }
        }
    }

```

```

        System.out.print(" --> Notification aux
utilisateurs ... ");
        try {
            broadcastService.broadcast("a été
ajouté au consensus !");
        } catch (CommunicationException ex) {
            Logger.getLogger(Chat.class.getName()).log(Level.SEVERE,
            null, ex);
        }

        consensusService.initConsensus(true);
    }

    // if end, disconnect from the system and
    exit the JVM
    if (message.equals("end")) {
        if(nbT>nbF){

            System.out.println("La décision
prioritaire est True !");
            decision = true;
        }
        else{

            System.out.println("La décision
prioritaire est False !");
            decision = false;

        }
        System.out.println("A la prochaine !");
        services.disconnect();
        System.exit(0);
    }

    // broadcast the message
    try {
        System.out.print(" --> Envoi message ...
");
        broadcastService.broadcast(message);
    }

```

```
        broadcastService.broadcast(decision);
        System.out.println("done");
    } catch (CommunicationException ex) {
        System.err.println(" *** communication
problem: " + ex);
    }
}
```