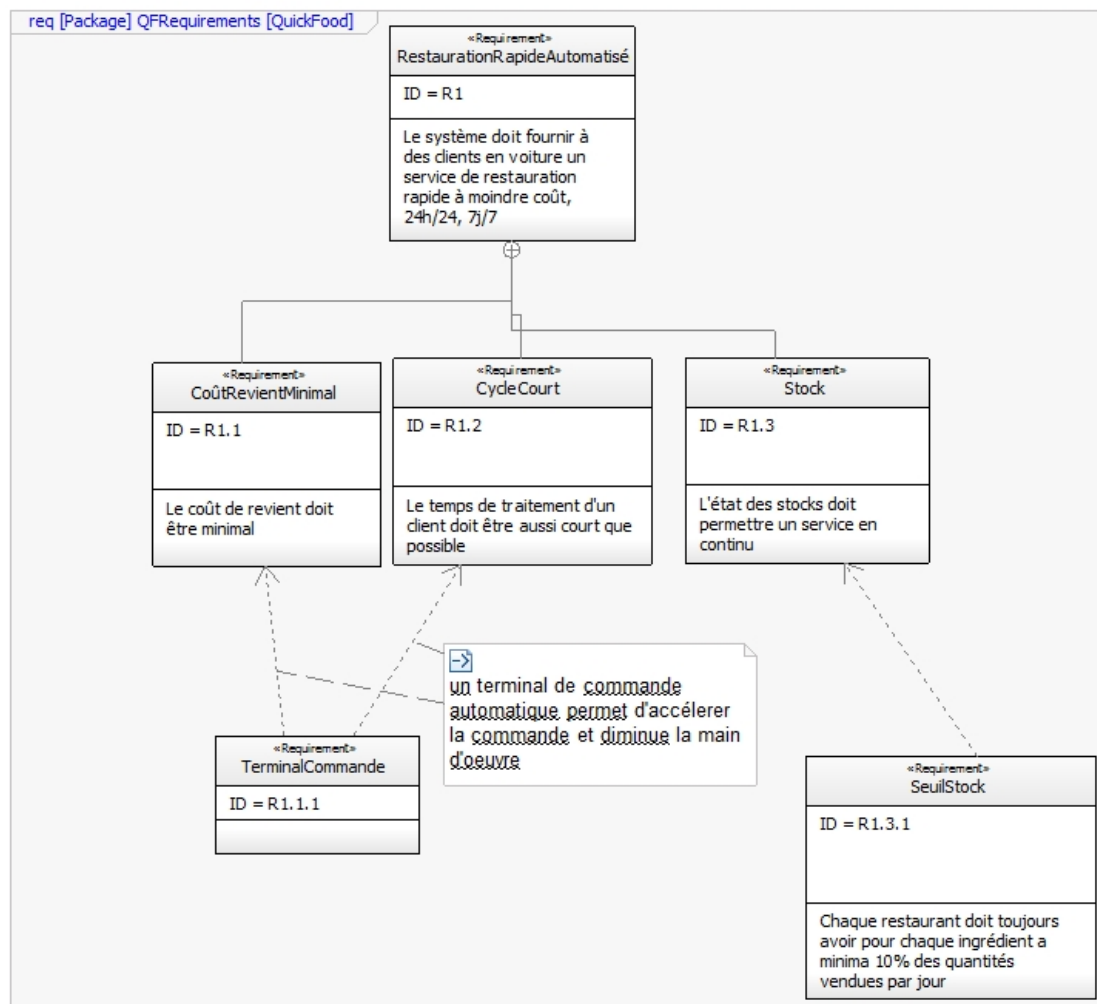


On considère une entreprise de restauration rapide qui propose un service de vente aux clients en voiture. Cette entreprise, appelée QuickFood, possède plusieurs restaurants et plusieurs entrepôts d'approvisionnement. Le principe commun pour tous les restaurants est le suivant : chaque client qui arrive en voiture saisit sa commande sur un terminal électronique (2 formules possibles A ou B), le client roule ensuite jusqu'à un terminal de paiement pendant que sa commande est préparée. Une fois que la commande est payée et préparée, le client peut disposer de sa commande et partir. Dès lors qu'une commande est prise, le restaurant vérifie l'état des stocks pour ses ingrédients et, en cas de besoin, commande à un entrepôt. Chaque restaurant est caractérisé par un profit journalier qui est fonction du nombre de formules vendues. La formule A a un profit de 2€, la formule B 3,5€.



## Questions

1. Quels sont les types de dépendance entre TerminalCommande-CycleCourt et TerminalCommande-CoûtRevientMinimal ? **(2points)**
2. Quel est le type de dépendance entre SeuilStock et Stock ? **(1point)**
3. Modifiez le diagramme de la figure 1 pour prendre en compte le fait que chaque restaurant doit être approvisionné par un entrepôt. **(1point)**
4. Définir un diagramme d'activités pour spécifier le flot des traitements. **(3points)**
5. Que faut-il faire pour exprimer le fait que si un paiement a échoué la commande n'est pas délivrée ? **(2points)**
6. Définir un diagramme de définition de blocs. **(3points)**
7. Que faut-il faire dans ce bdd pour comptabiliser le nombre de clients servis ? **(1point)**
8. Que faut-il faire dans ce bdd pour prendre en compte des restaurants qui propose une carte de formules plus étendues (4 menus au lieu de 2). **(1point)**
9. Exprimer par un diagramme paramétrique le bénéfice journalier des restaurants. **(3points)**
10. Soient les classes modelica suivantes : **(3points)**

```
connector OutPort
output Boolean signal;
end OutPort;
```

```
connector OutNumber
output Integer number;
end OutNumber;
```

```
class GenerationClients
OutPort outPort ;
OutNumber outNumberA,outNumberB ;
...//partie générations aléatoires
end GenerationClients ;
```

On suppose que la classe GenerationClients génère des événement aléatoires de type booléens pour simuler l'arrivée de clients (vraie si un client est là, faux sinon) et des nombres de menus (A ou B) commandés par respectivement les variables outPort, outNumberA et outNumberB.

Ecrire une classe Restaurant, avec tous les éléments nécessaires (y compris leurs définitions sous forme de classes ou dérivés de classes si nécessaire), pour que Restaurant soit connectée à GenerationClients et qui permette de calculer le nombre de clients et de repas servis.

Connecter GenerationClients et Restaurant