

# TE Génie Logiciel - GEN

Nom :

Date : 25 Mai 2016

1	2	3	4	5	6	7	8	
8	0	2	2	2	1.5	4	9.5	23.0
11	2	2	2	2	2.5	8	7	36.5

## EXERCICE 1 – 11 POINTS

Dresser le modèle de domaine UML décrivant les informations manipulées par une agence de voyage qui répond à des demandes de clients en leur proposant des hôtels susceptibles de répondre aux critères de la demande.

La société maintient une clientèle où chaque client est enregistré avec son nom et son email.

Elle maintient également une liste d'hôtels où chaque hôtel est également enregistré avec son nom et son email.

Enfin, elle maintient une liste de compagnies d'aviation, caractérisées chacune uniquement par un nom.

Les demandes de clients, propositions faites en retour, modifications de demandes, annulations, décisions finales sont enregistrés sous la forme d'événements, caractérisés chacun par une date et le client concerné.

En particulier:

- Une demande est caractérisée par un texte (critères de la demande), une date de début et une date de fin.
- Les modifications de demandes sont de simples demandes, si ce n'est qu'elles se rattachent en plus à la demande initiale.
- A chaque demande (ou modification de demande) correspondra une proposition faite en retour qui indiquera la liste des hôtels conseillés, avec pour chacun d'eux une note de 1 à 5, correspondant au "degré de matching" avec la demande.
- Une annulation se rattache à la demande initiale.
- Une décision finale de client doit comprendre une sélection de deux d'hôtels au plus, indiquant pour chacun d'eux s'il s'agit du premier ou du deuxième choix. Une décision se rattache à la demande initiale.

Une fois qu'un client a pris sa décision finale, les réservations sont opérées et enregistrées dans le système, en enregistrant pour chacune:

- l'hôtel
- les dates d'arrivée et de départ
- les deux compagnies d'aviation utilisées pour le départ et l'arrivée
- le client concerné

- A retenir la notion de scalabilité en réseau pour les vendredi examens.  
pour la négociation du contenu, le serveur fait

### Remarques générales concernant les exercices de modélisation

1. Ne pas rajouter d'informations qui ne se trouvent pas exprimées dans l'énoncé.
2. Toutefois, si l'énoncé vous paraît ambigu (cela peut arriver..), vous pouvez le compléter. Indiquez-le par une remarque.
3. Utiliser l'**agrégation** ou la **composition** partout où cela s'y prête. Accompagner votre schéma d'une justification (pourquoi agrégation plutôt que composition).
4. Nommez les associations simples
5. Utiliser la « **Généralisation-Spécialisation** » partout où cela s'y prête.
6. Utiliser des **associations ternaires** partout où cela s'y prête.
7. Indiquer la cardinalité de manière explicite (ne pas utiliser de valeurs par défaut)
8. Compléter votre schéma de modélisation par la liste des **contraintes** que la syntaxe des diagrammes ne permet pas d'exprimer.

### EXERCICE 2 – 2 POINTS

Méthodes agiles : Que signifie « Le processus de développement doit être adaptatif? »

0/2 Cela signifie que le projet doit pouvoir accepter des modifications au cours du développement

Non. Il s'agit de pouvoir adapter la méthode de travail pour gagner en efficacité, et non pas

### EXERCICE 3 – 2 POINTS

XP : Si l'équipe change (départs, nouveaux engagements), doit-on revoir la vélocité? (justifier)

2/2 oui, car la vélocité <sup>représente</sup> la charge de travail réalisable pour une équipe donnée, ainsi si cette équipe change, elle change aussi <sup>d'adapter le développement aux différents changements</sup>

### EXERCICE 4 – 2 POINTS

- 1 Point par réponse fausse

Méthode XP : Quelles sont les tâches confiées au client ?

- 2/2
- La spécification des tests fonctionnels ✓
  - La rédaction du plan d'iteration ✓
  - La rédaction des scénarios ✓

- connaître aussi

CRUD - Create (post)

Read (get)

update (put)

delete (delete)

il faut aussi connaître  
les requête et reponses

HTTP

Page 3

### EXERCICE 5 - 2 POINTS

Méthode XP : Avec XP, quand on écrit la spécification d'un logiciel, doit-on penser aux tests fonctionnels ? Justifiez votre réponse.

oui, car lors de la réalisation des itérations, ce sont ces tests qui sont réalisés et lorsqu'à la fin les résultats des tests sont positifs, cela permettra de savoir qu'on a réussi l'itération

### EXERCICE 6 - 2.5 POINTS

- 1 Point par réponse fausse

Méthode XP : Le refactoring d'un fragment de code intervient à quel moment ? Cochez les propositions qui vous paraissent correctes.

- 1.5 / 2.5
- ☒ Avant la spécification des tests fonctionnels correspondants
  - ☒ Dès lors que le client souhaite une amélioration de la conception du fragment de code
  - ☒ Dès lors que le client change d'avis vis à vis d'une fonctionnalité pour laquelle le fragment de code joue un certain rôle
  - ☒ Dès lors que le fragment de code donne satisfaction (du point de vue de ses résultats)
  - ☒ Dès lors que le fragment de code est soupçonné d'un délit de YAGNI

---

**EXERCICE 7 – 8 POINTS**

Vous allez modéliser le fonctionnement d'un processus d'enquête administrative en dessinant le **diagramme d'activités** correspondant. Ce diagramme d'activités met en jeu 3 couloirs d'activités, celui du Responsable administratif, celui de l'Analyste, et celui du Service Postal de l'entreprise.

C'est le responsable administratif (« RA ») qui, au départ, lance l'enquête.

Une fois l'enquête lancée, le RA confie l'enquête à l'analyste. Ce dernier conçoit l'enquête et la retourne au RA. Le RA fait imprimer les formulaires d'enquête et prépare la liste des personnes qui seront interviewées pour l'enquête.

Les formulaires et la liste des personnes sont alors confiées par le RA au Service Postal de l'entreprise (« SP »).

Pendant que le SP fait son travail (voir plus loin : envoi des formulaires et réception des réponses brutes), le RA opère les réservations auprès des différents médias qui publieront les résultats de l'enquête. En parallèle à ces réservations, le RA attend les réponses brutes recueillies par le SP.

Une fois que le RA a reçu les réponses brutes du SP et qu'il a opéré toutes les réservations auprès des médias, il ne lui reste plus qu'à publier les résultats. L'opération est alors terminée.

Le rôle du SP – dès qu'on lui confie les formulaires et la liste des personnes – consiste à distribuer les formulaires auprès des personnes sélectionnées (elles devront répondre dans un délai de 30 jours), et à recueillir les réponses brutes. Dès que les 30 jours sont écoulés, le SP communique toutes les réponses brutes recueillies au RA et archive l'enquête.

***Note technique** - Toutes les communications d'informations sont opérées de manière **asynchrone** (on n'attend pas la réponse).*

### EXERCICE 8 : 7 POINTS – DIAGRAMME DE SEQUENCE

Vous allez dessiner le diagramme de séquence à même de modéliser l'interaction (échange de messages) entre les différents objets concernés par l'exécution de la méthode **handleRequest** du contrôleur général d'une application WEB. Cette méthode analyse la requête http reçue d'un client, effectue le traitement demandé, et retourne une réponse au client (code html).

Le diagramme de séquence fera intervenir les 3 classes **GeneralController**, **SpecificController** et **Response**.

La communication des paramètres ne sera pas montrée dans le diagramme.

```
class GeneralController {
    public void handleRequest(HTTPRequest req) {
        Response resp = new Response();
        Controller ctr = getSpecificController(req);
        ctr.handle(req, resp);
        if (resp.OK())
        {
            sendResponse(resp);
        }
    }

    private Controller getSpecificController (HTTPRequest req){
        // Analyse l'url de la requête, crée une nouvelle instance d'un contrôleur spécifique
        // à même de prendre en charge l'action demandée par le client
        [..code non présenté..]
        unController = new SpecificController();
        return unController ;
    }

    private void sendResponse (Response resp){
        // Pour renvoyer la réponse (code HTML) au client
        [..code non présenté..]
    }
}
```

- Les

x Resolution de nom

```

class SpecificController {
    public SpecificController() {
        [...code non présenté...]
    }

    public void handle(HTTPRequest req, Response resp) {
        // Analyse de la requête et construction de la réponse (code html)
        [... construction de la réponse, enregistrée dans la variable t (String),
            code non présenté ...]
        resp.setResponse(t);
    }
}

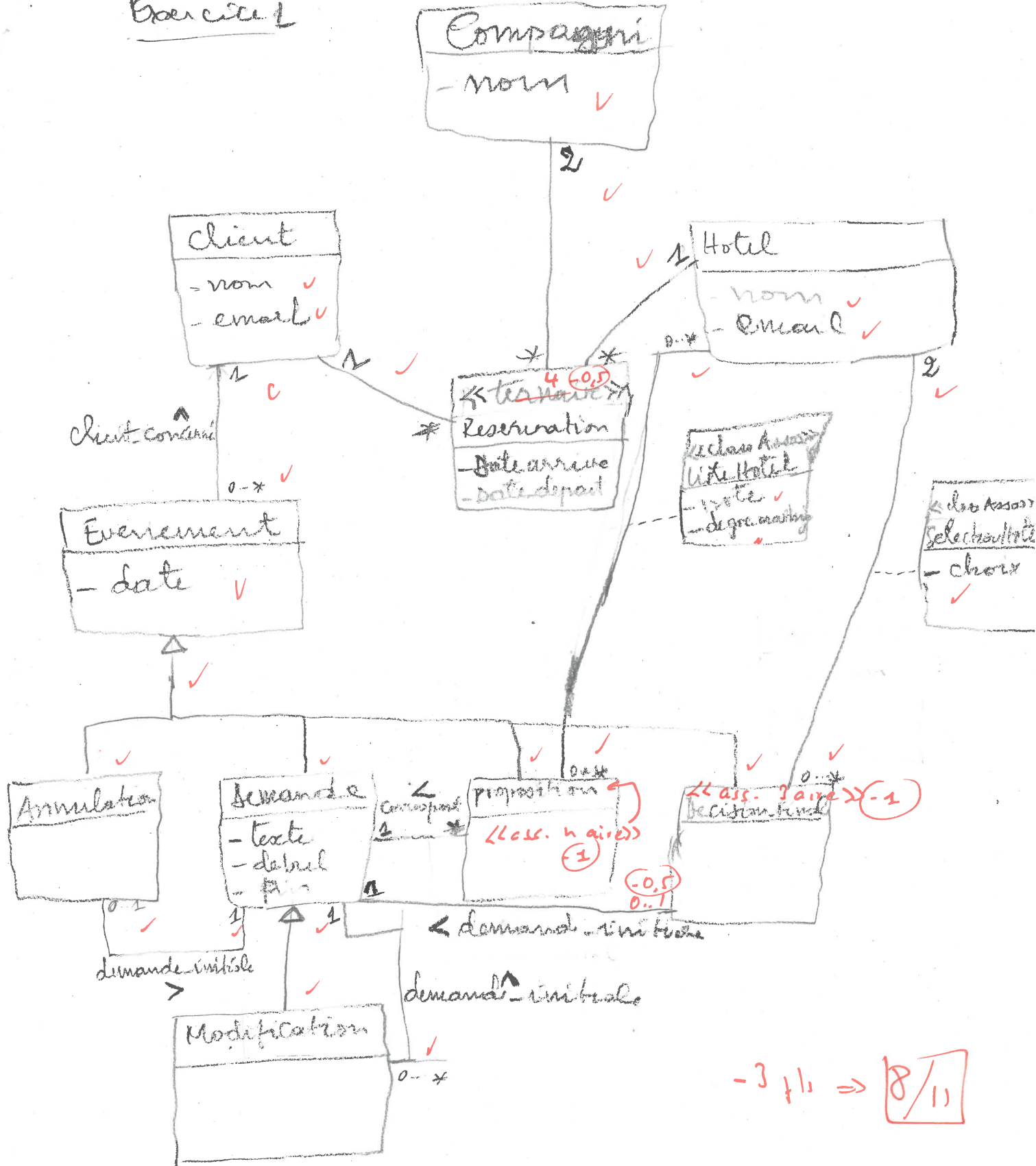
class Response {
    public Response() {
        [...code non présenté...]
    }

    public void setResponse(String html) {
        // Construction de la réponse à partir du texte html passé en paramètre
        [...code non présenté...]
    }

    public boolean ok() {
        // Retourne true si la réponse est prête à être envoyée au client, false sinon.
        [...code non présenté...]
    }
}

```

# Exercice 1



C.1 : la date d'arrivee doit être différent de la date d'arrivée

C.2 : une choix dans une selectionHotel ne peut être que soit premier ou deuxième

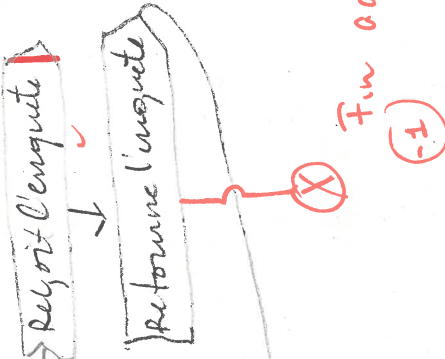
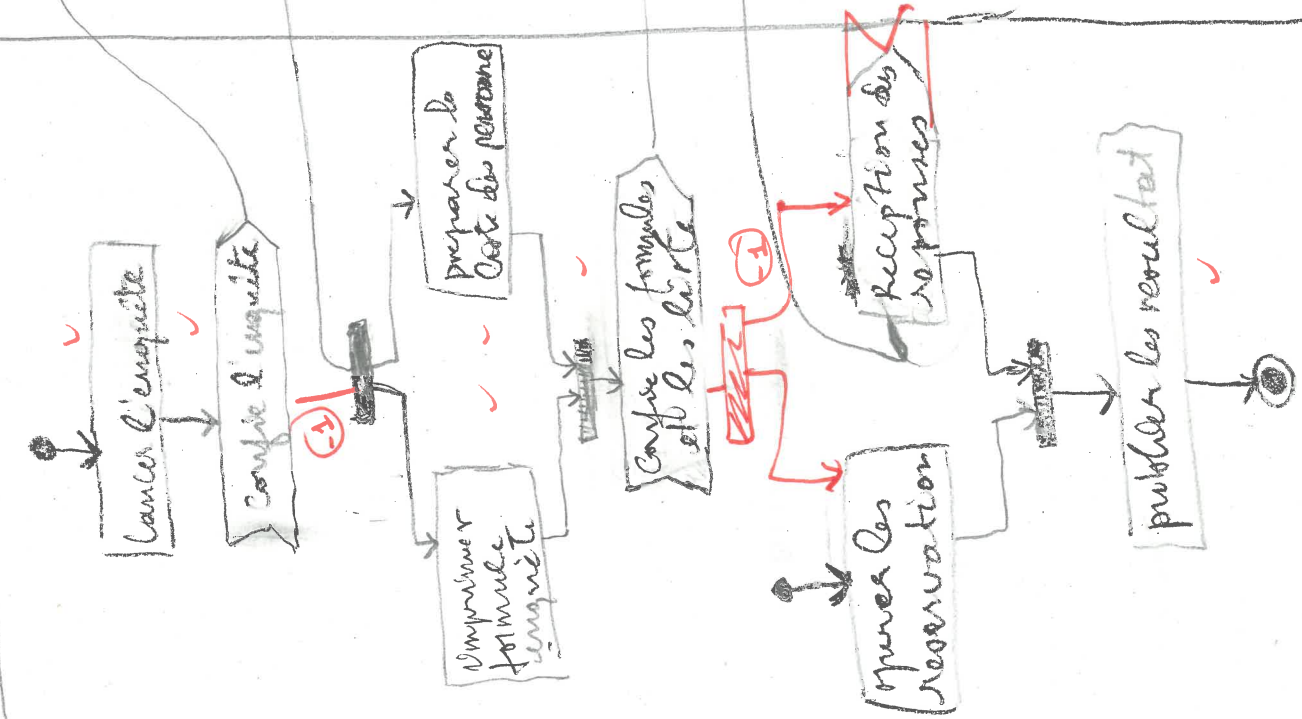
C.3 : une demande initiale pour les modification, peut pas être une modification.

C.4 : la note doit varier entre 1 et 5 inclus









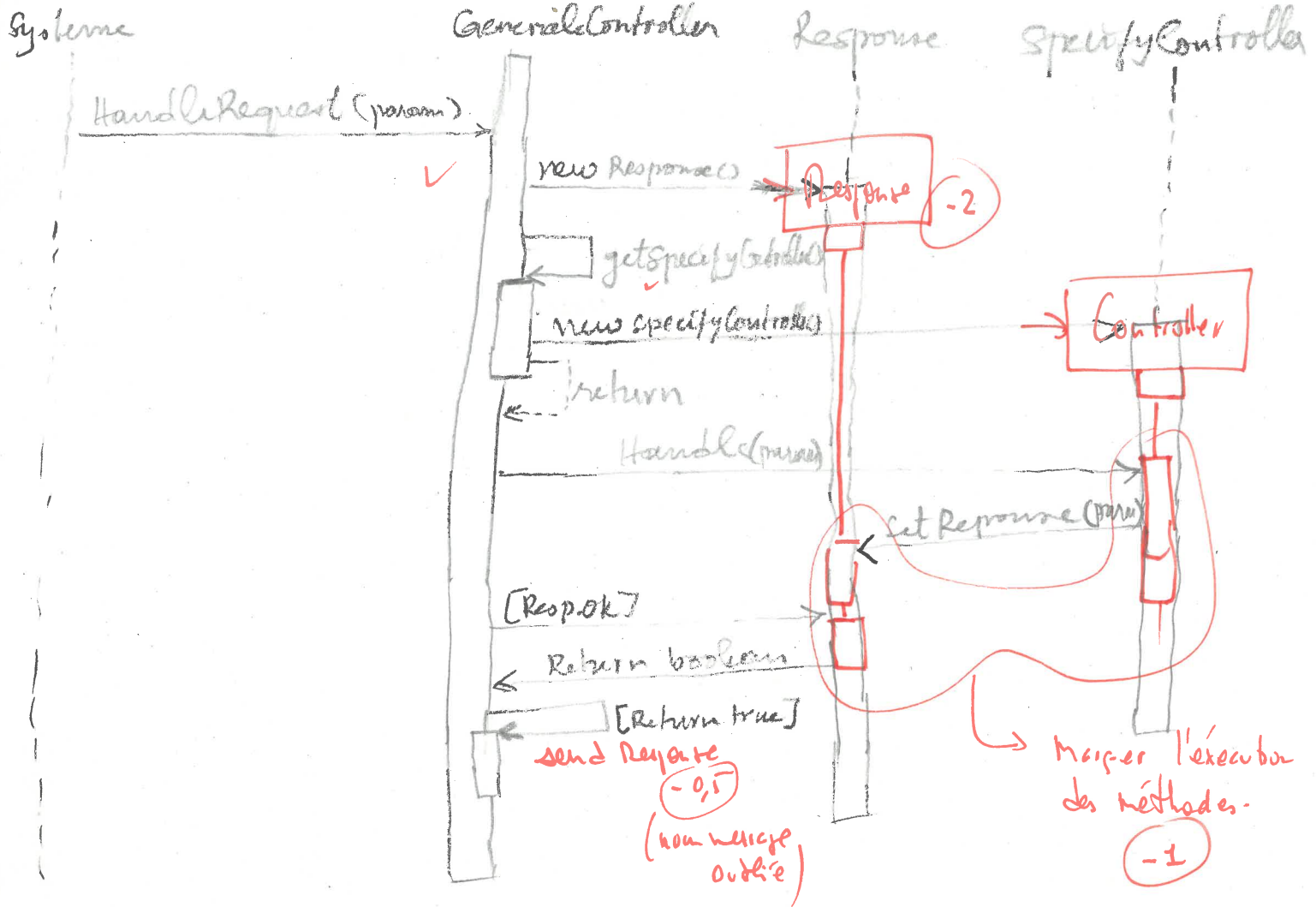
-1  
30j

Diagrammer:  
 Fin processus (O)  
 Fin activité (X)

-4 => 4/8



# Exercice 8



$$-3,5 \Rightarrow \boxed{3,5/7}$$

