

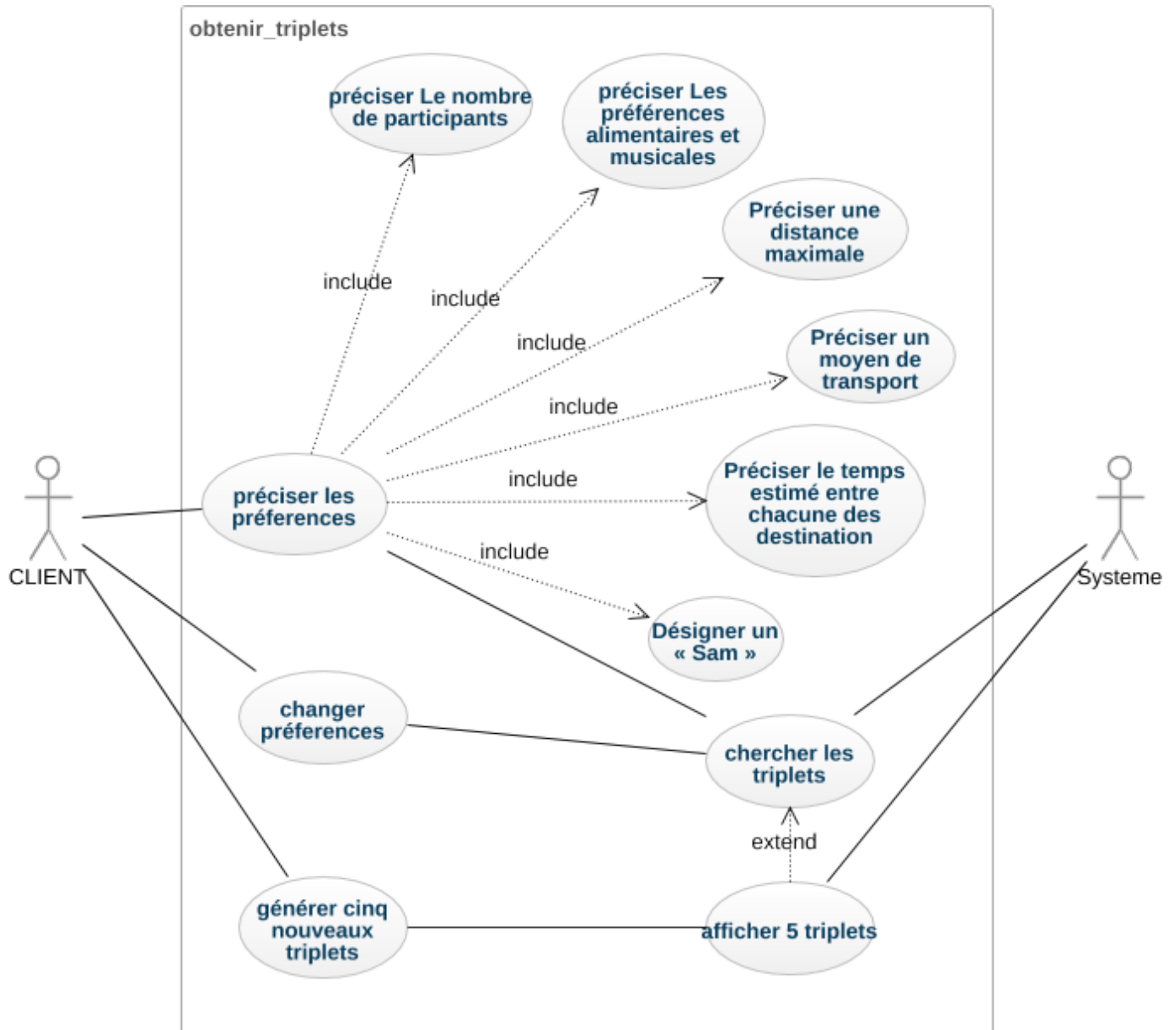
# Sortir à Paris

## Cahier des charges (version 2)

### Analyse

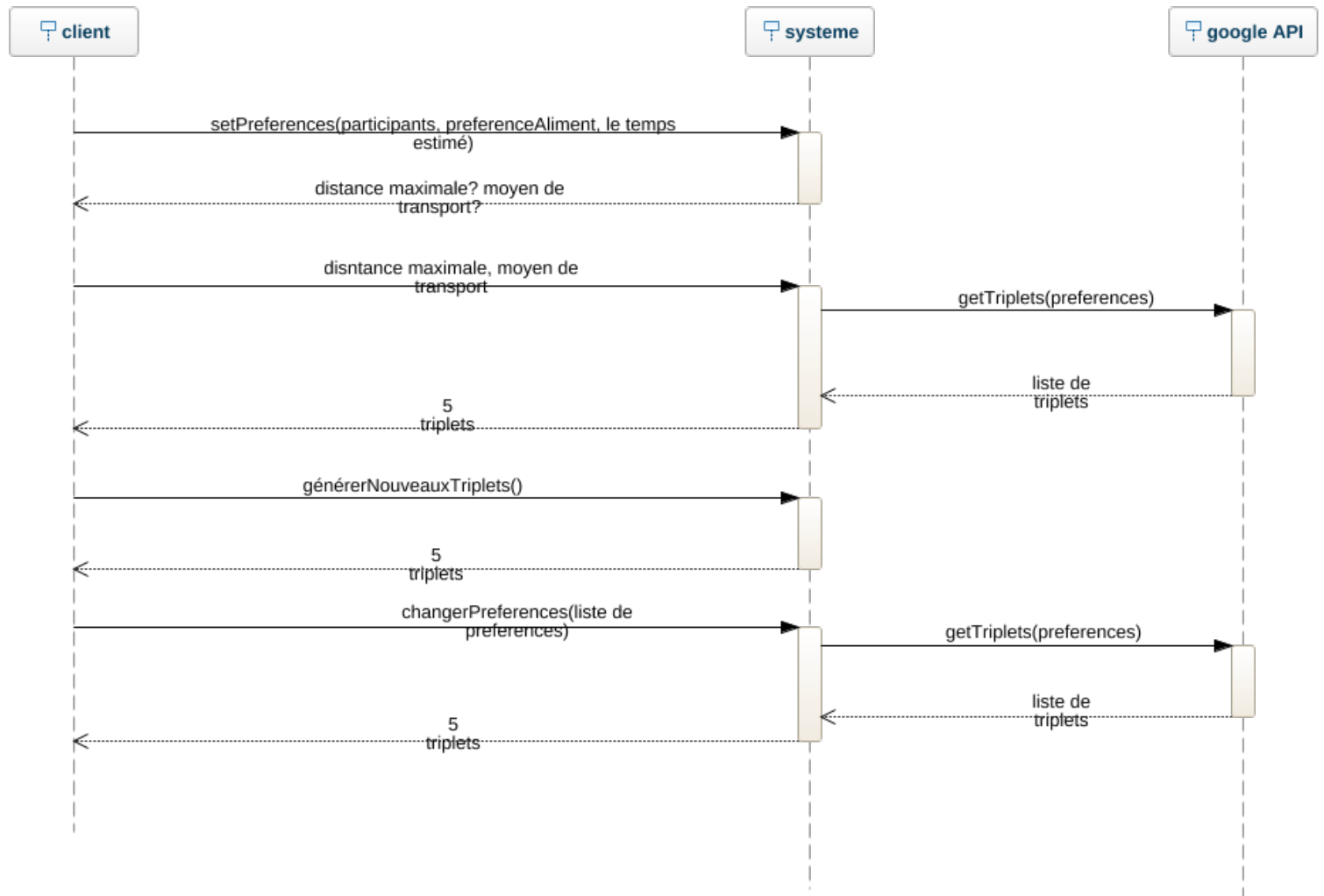


### Diagramme de cas d'utilisation 1 : Obtenir les triplets.



**Explications :**

Pour obtenir un triplet, le client peut effectuer trois actions. Il peut préciser les préférences pour sa soirée, et ce, à travers de nombreux détails représentés par les “include” dans notre diagramme. Il peut également modifier ces préférences. Quand les préférences sont notifiées, le système doit afficher et proposer cinq triplets. L'utilisateur peut à tout moment décider de générer cinq nouveaux triplets.

**Diagramme de séquence 1 : Obtenir les triplets.**

**Explications :**

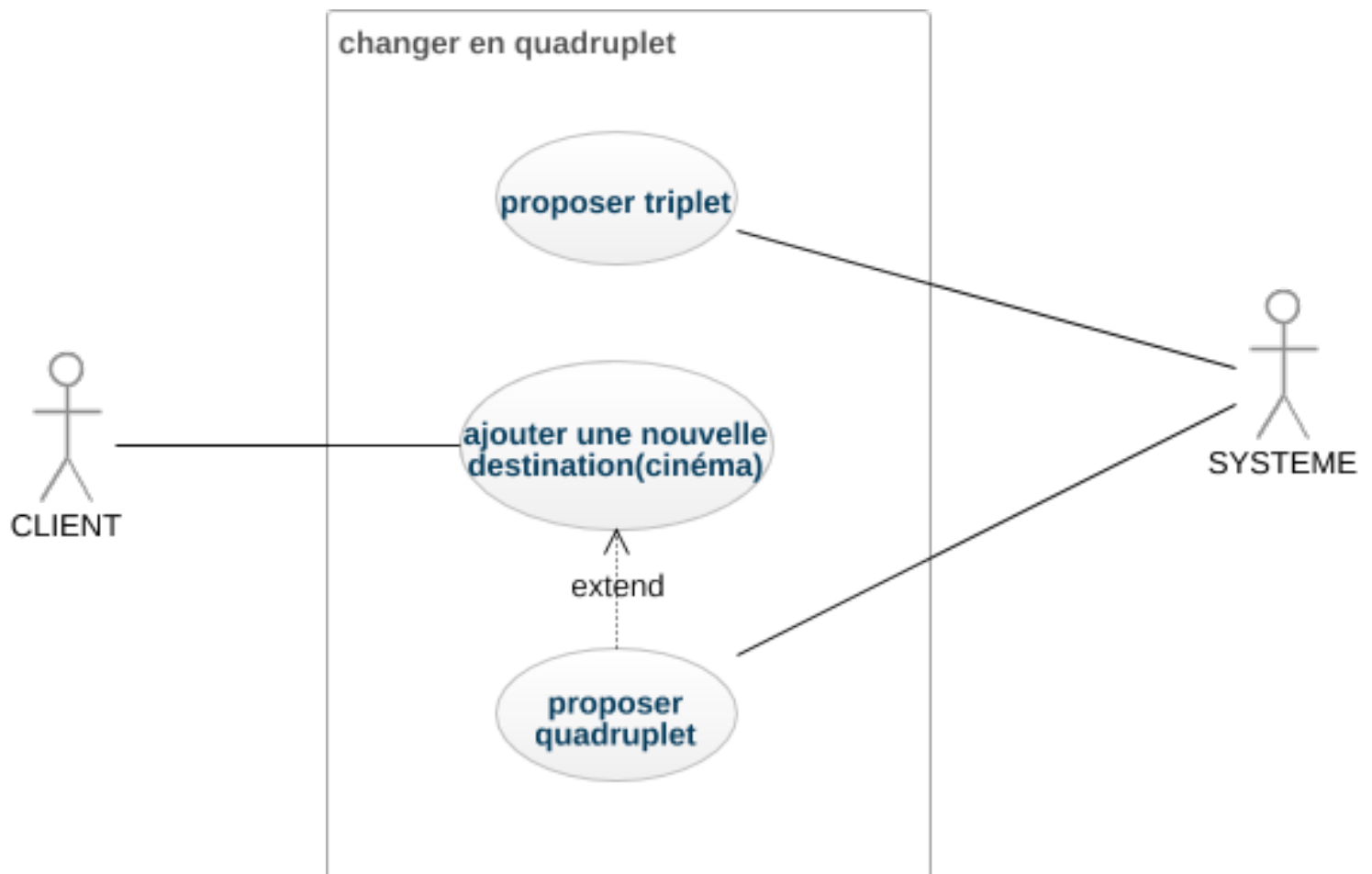
Le diagramme de séquence ci-dessus précise le diagramme de cas d'utilisation expliqué précédemment. Quand l'utilisateur spécifie ces premières préférences va lui demander si il souhaite des préférences supplémentaires qui seront facultatives. Une fois les préférences spécifiées, le système va interroger l'api google pour récupérer les triplets. De même pour changer les préférences. Si l'utilisateur souhaite charger de nouveaux triplets, le système n'a pas besoin d'interroger à nouveau l'api google car un certain nombre de triplets seront stockés en avance.

**Scénario nominal :**

1. Le client spécifie ses préférences pour la soirée.
2. Le système demande des renseignements facultatifs supplémentaires.
3. Le client spécifie les paramètres supplémentaires.
4. Le système interroge l'API google.
5. Le système retourne cinq triplets.
6. Le client consulte les triplets.

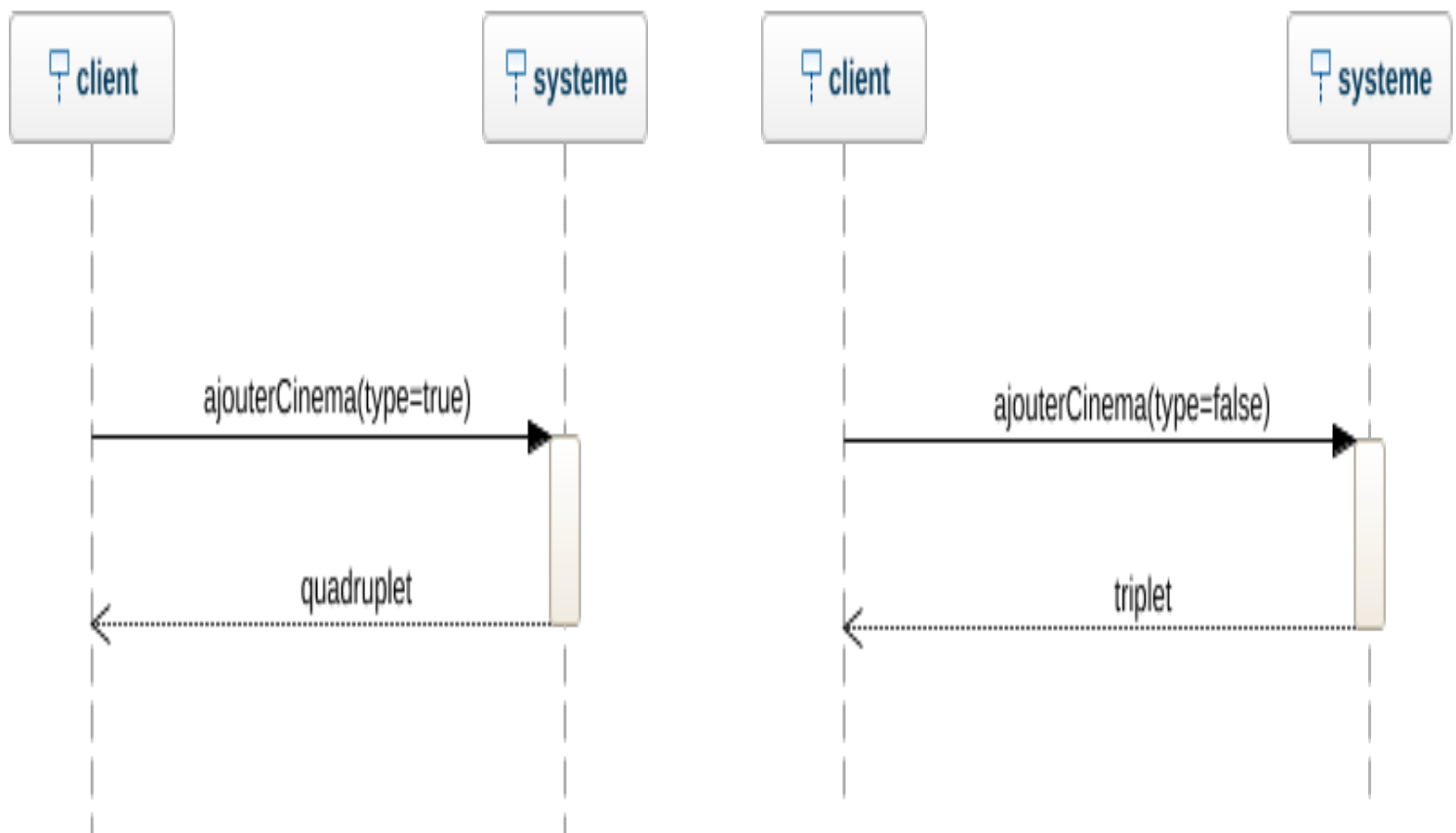
**Scénarios alternatifs :**

- 6.a Le client charge cinq nouveaux triplets.
- 6.b Le client modifie les paramètres. (retour à l'étape 4)

**Diagramme de cas d'utilisation 2 : Changer en quadruplet.**

**Explications :**

Le client peut avant de choisir ses préférences, décider d'ajouter une destination ainsi demander au système de générer non pas des triplets, mais des quadruplets. Cette nouvelle destination (un cinéma) vient compléter les trois autres si l'option est choisie.

**Diagramme de séquence 2 : Changer en quadruplet.**



**Explications :**

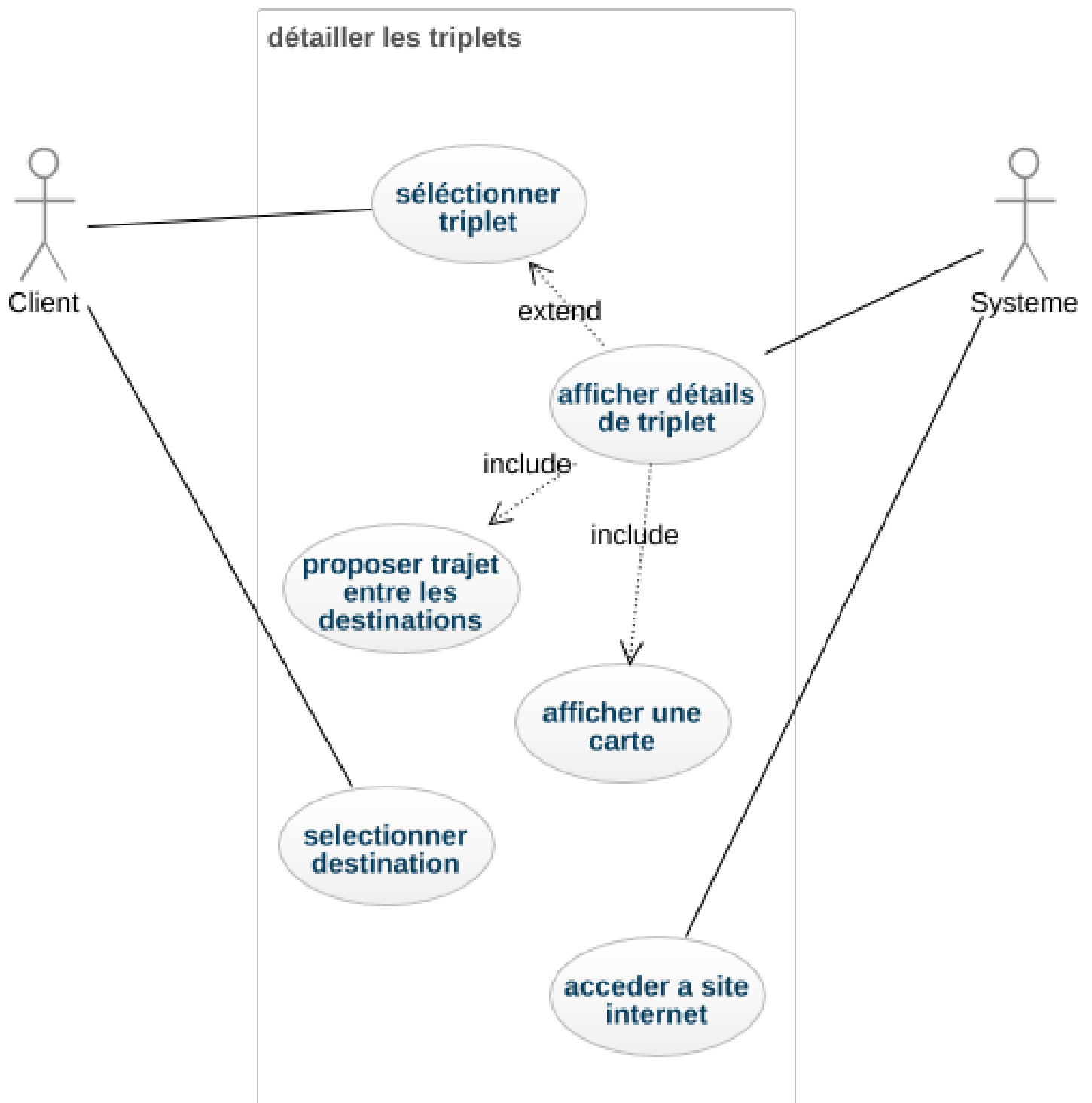
Le diagramme de séquence (2) ci-dessus précise le diagramme de cas d'utilisation (2) expliqué précédemment. De plus, tout ce qui a été défini dans le diagramme de séquence 1 se déroule au milieu de l'opération spécifiée ici. En effet, l'utilisateur décide d'ajouter une nouvelle sortie ou non, puis, avant de retourner des triplets/quadruplets, toutes les actions spécifiées dans le diagramme 1 sont exécutées. Ce diagramme sert simplement à représenter le changement de type de retour si l'option est choisie.

**Scénario nominal :**

1. Le client choisit l'option quadruplet
2. Le système effectue les opérations du diagramme 1.
3. Le système retourne un quadruplet.

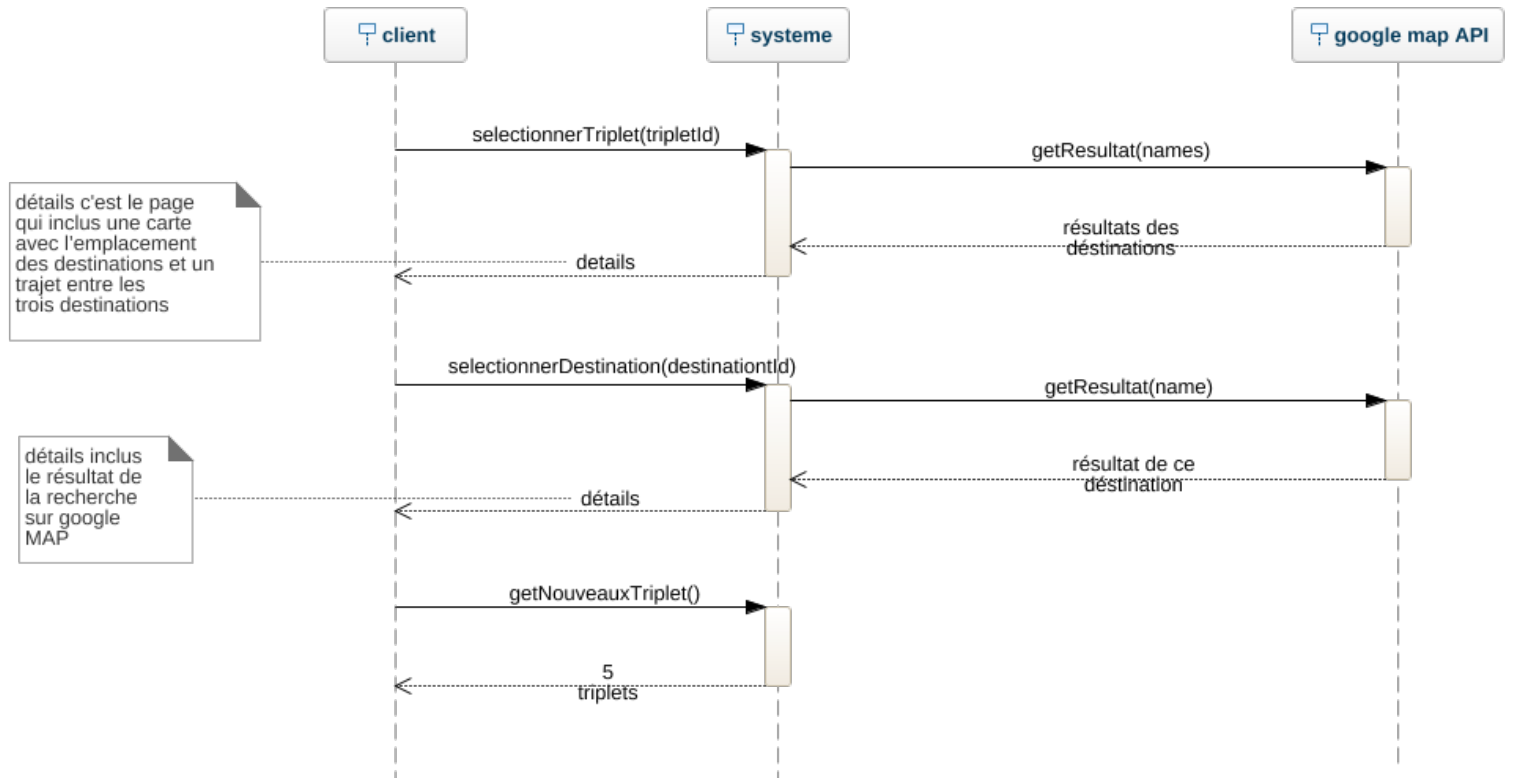
**Scénarios alternatifs :**

- 1.a Le client choisit l'option triplet (retour aux actions du diagramme 1).

**Diagramme de cas d'utilisation 3 : Détailler les triplets.**

**Explications :**

Pour obtenir des détails sur les 5 triplets proposés, le client peut sélectionner un des triplet pour obtenir des détails sur celui-ci (ces détails sont spécifiés dans le digramme). Il peut également obtenir des détails sur une destination précise, et notamment accéder (si il existe) à son site internet dont l'accès est assuré par le système.

**Diagramme de séquence 3: Détailler les triplets.**

**Explications :**

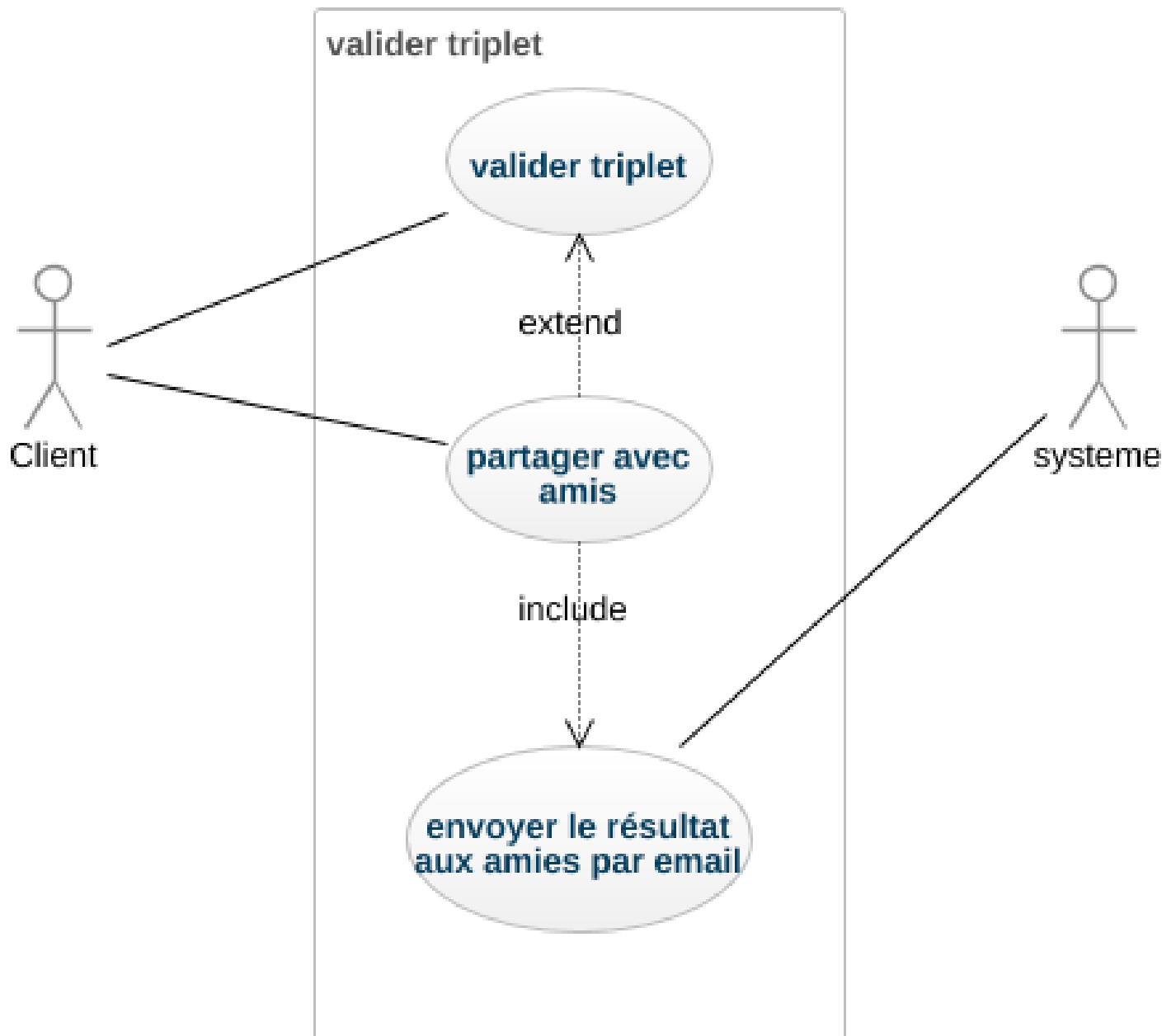
Le diagramme de séquence (3) ci-dessus précise le diagramme de cas d'utilisation (3) expliqué précédemment. Un utilisateur peut sélectionner un triplet et le système en affiche les détails à travers une page spécifique. Sur cette page, le client peut sélectionner une destination précise du triplet pour obtenir des détails sur celle-ci (son site internet notamment). L'utilisateur peut toujours générer cinq nouveaux triplets. Le client peut également valider un triplet (voir diagrammes suivants)

**Scénario nominal :**

1. Le client sélectionne un triplet parmi les cinq proposés.
2. Le système affiche les détails du triplet.
3. Le client sélectionne une destination.
4. Le système affiche les détails de la destination.
5. Le client veut accéder au site internet de la destination.
6. Le système ouvre une nouvelle page avec le site de la destination.
7. Le client retourne sur la page de détails du triplet et le valide.

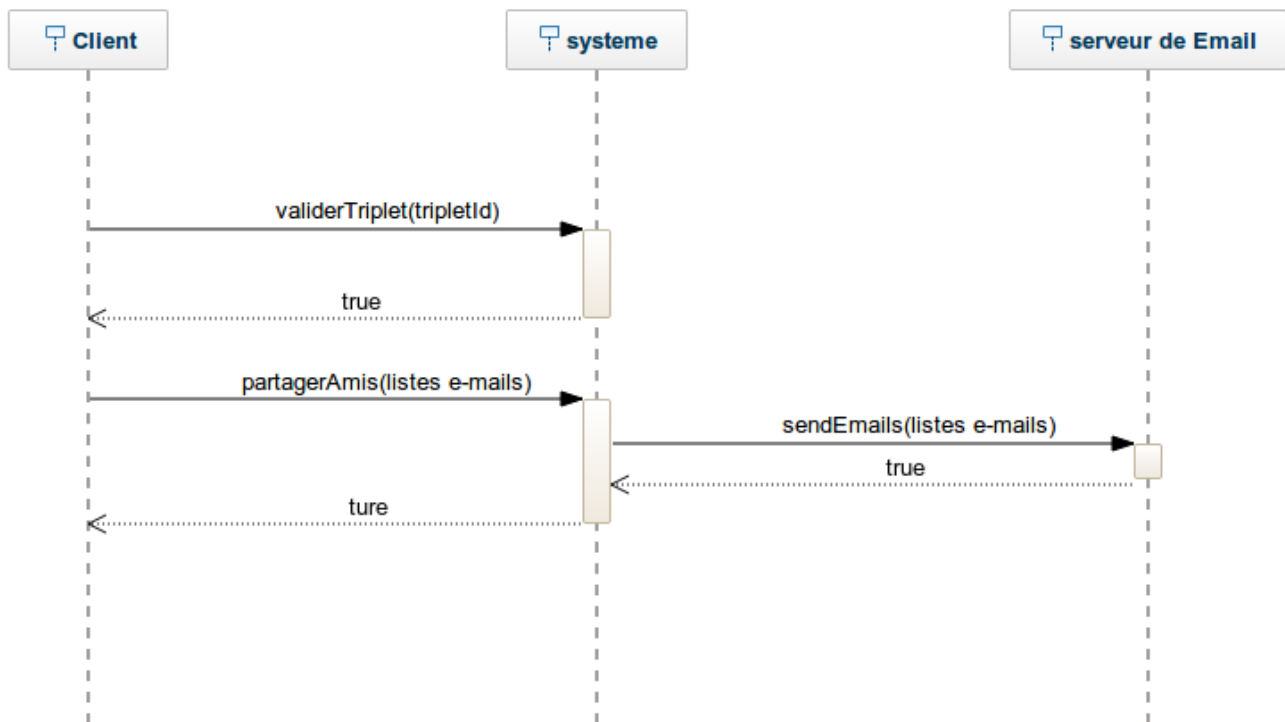
**Scénarios alternatifs :**

- 1.a Le client charge cinq nouveaux triplets.
- 3.a Le client retourne en arrière (retour à l'étape 1).
- 3.b Le client valide le triplet.
- 5.a Le client retourne en arrière (retour à l'étape 3).

**Diagramme de cas d'utilisation 4 : Valider triplet.**

**Explications :**

Le client peut valider un triplet parmi ceux proposés. Il peut également, si le triplet est validé, partager les détails de la soirée ainsi créée avec ses amis par mail. Si il décide de partager avec ses amis, le système envoie un mail à tous les participants avec les détails de la soirée.

**Diagramme de séquence 4 : Valider triplet.**



**Explications :**

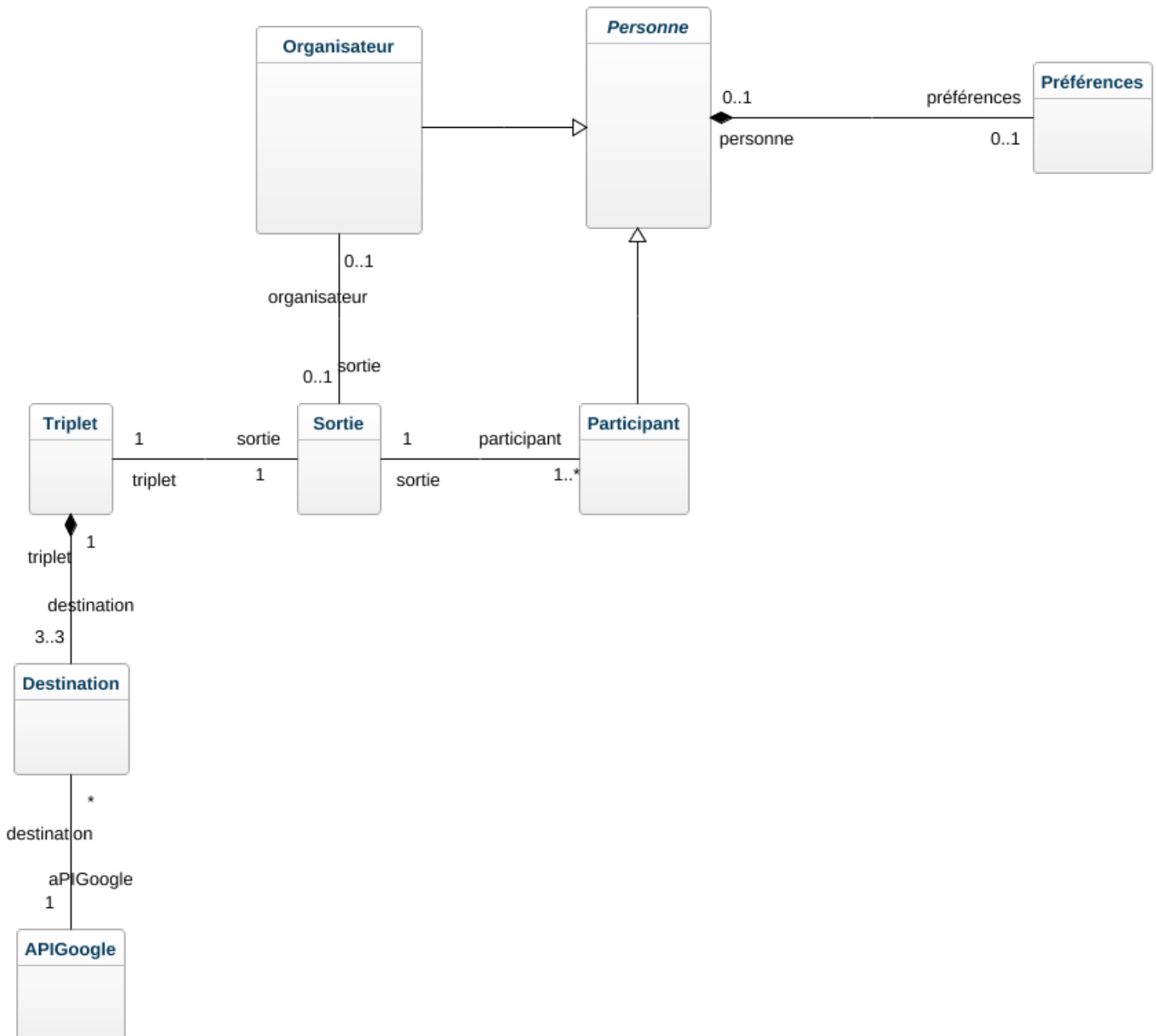
Le diagramme de séquence (4) ci-dessus précise le diagramme de cas d'utilisation (4) expliqué précédemment. Un utilisateur peut décider de valider un triplet. Il a ensuite la possibilité de partager sa soirée avec les participants. Le système procède donc à l'envoi d'un mail « type » à tous les participants de la soirée (les adresses mails sont connues par le système car spécifiées lors de la création des participants).

**Scénario nominal :**

1. Le client valide un triplet
2. Le système affiche une confirmation et propose un partage.
3. Le client décide de partager avec ses amis.
4. Le système envoie un mail aux participants et confirme à l'utilisateur.

**Scénarios alternatifs :**

- 2.a Le client annule sa validation.

**Diagramme de classe simplifié.**

**Explications :**

Le diagramme de classe simplifié ci-dessus présente les entités concernées et les relations entre les différentes classes .

- La classe « Préférence » est constituée des préférences des différents participants. Par exemple les préférences alimentaires ou musicales.
- La classe « Personne » est une classe abstraite qui contient les paramètres communs des participants et de l'organisateur (En effet, l'organisateur participe également à la soirée). Par exemple les paramètres, nom , prénom, l'adresse mail. Il nous semblait pertinent de créer dans le cas présent une classe abstraite dont les deux classes « participant » et « organisateur » pourront hériter.
- La classe « Participant » hérite de le classe « personne » .
- La classe « Organisateur » hérite également de la classe « personne » avec des paramètres et fonctions supplémentaires afin de gérer et organiser la soirée au mieux.
- La classe « APISGoogle » représente la classe qui servira d'interface et de lien entre notre application et l'API Google nécessaire à son fonctionnement.
- La classe « Destination » contient les données récupérées par la classe « APISGoogle ». Elle contient donc les différentes informations des destinations récupérées.
- La classe « Triplet » contient une collection de destination (trois dans notre diagramme mais cela pourra être 4 comme expliqué plus haut).
- La classe « Sortie » est constituée de participants, d'un triplet et d'un organisateur. Elle représente le résultat final de la sortie proposée par notre application.

Nous avons prévu, pour notre projet, de suivre un modèle MVC que nous détaillerons avec un autre diagramme dans la partie conception de notre cahier des charges.