

Assistant évènementiel de sorties à Paris : Cahier de conception

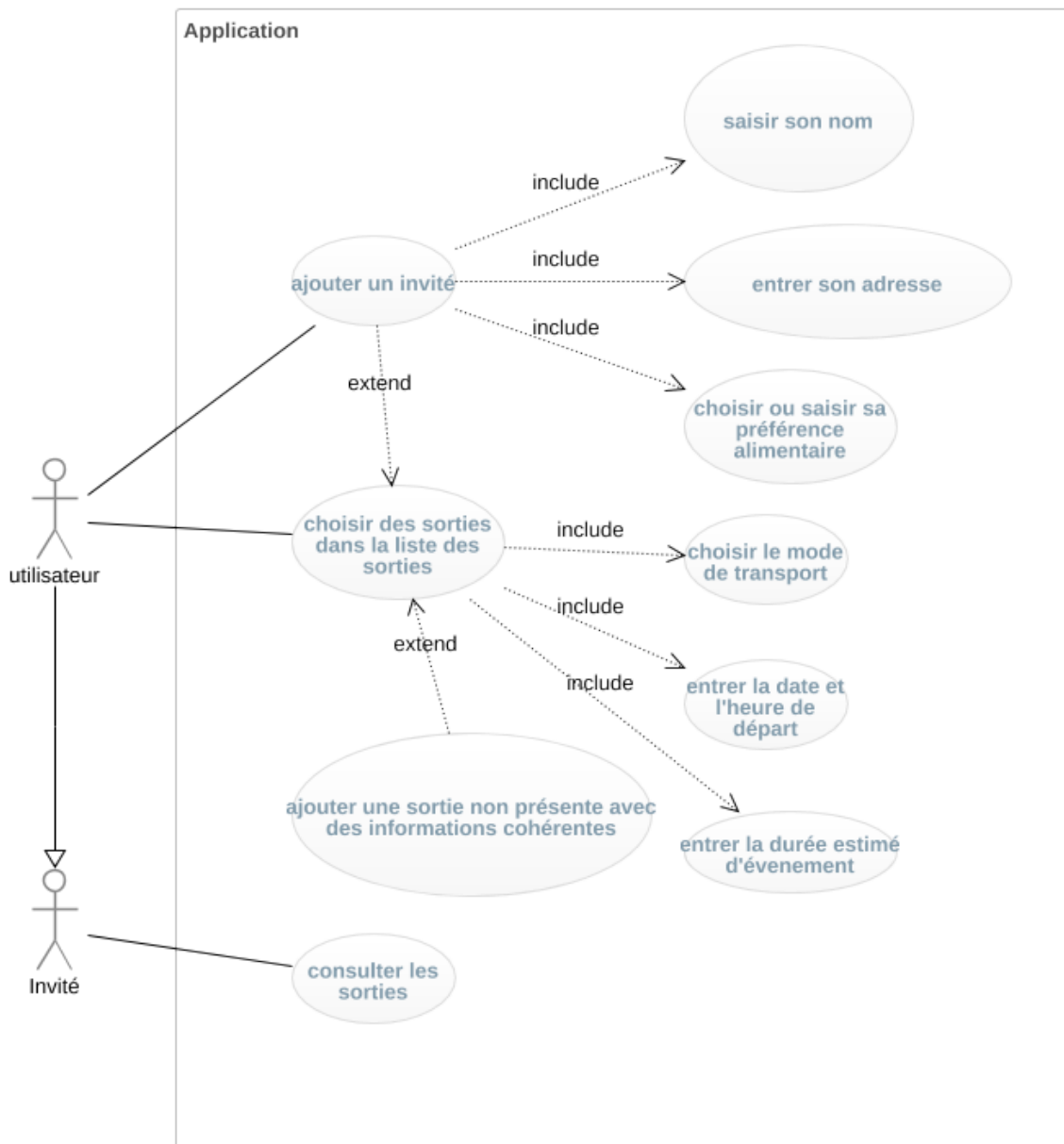
Binello Arthur, Mo Rui, Gan Boyuan

Table des matières :

1. Diagramme de cas d'utilisation	2
1. Diagramme	2
2. Explications	3
2. Diagrammes de séquence et fonctions utilisées	4
1. Ajouter un invité	4
a. Diagramme	4
b. Scénario	5
2. Choisir des sorties dans la liste des sorties	6
a. Diagramme	6
b. Scénario	7
3. Ajouter une sortie non présente avec informations cohérentes	8
a. Diagramme	8
b. Scénario	9
4. Consulter les sorties	10
a. Diagramme	10
b. Scénario	10
3. Diagramme de classe	12
1. Diagramme	12
2. Explications	12

1. Diagramme de cas d'utilisation

1. Diagramme



2. Explications

Les acteurs :

- Invité : Peut seulement réaliser l'action consulter les sorties.
- Utilisateur : Peut réaliser toutes les actions du système.

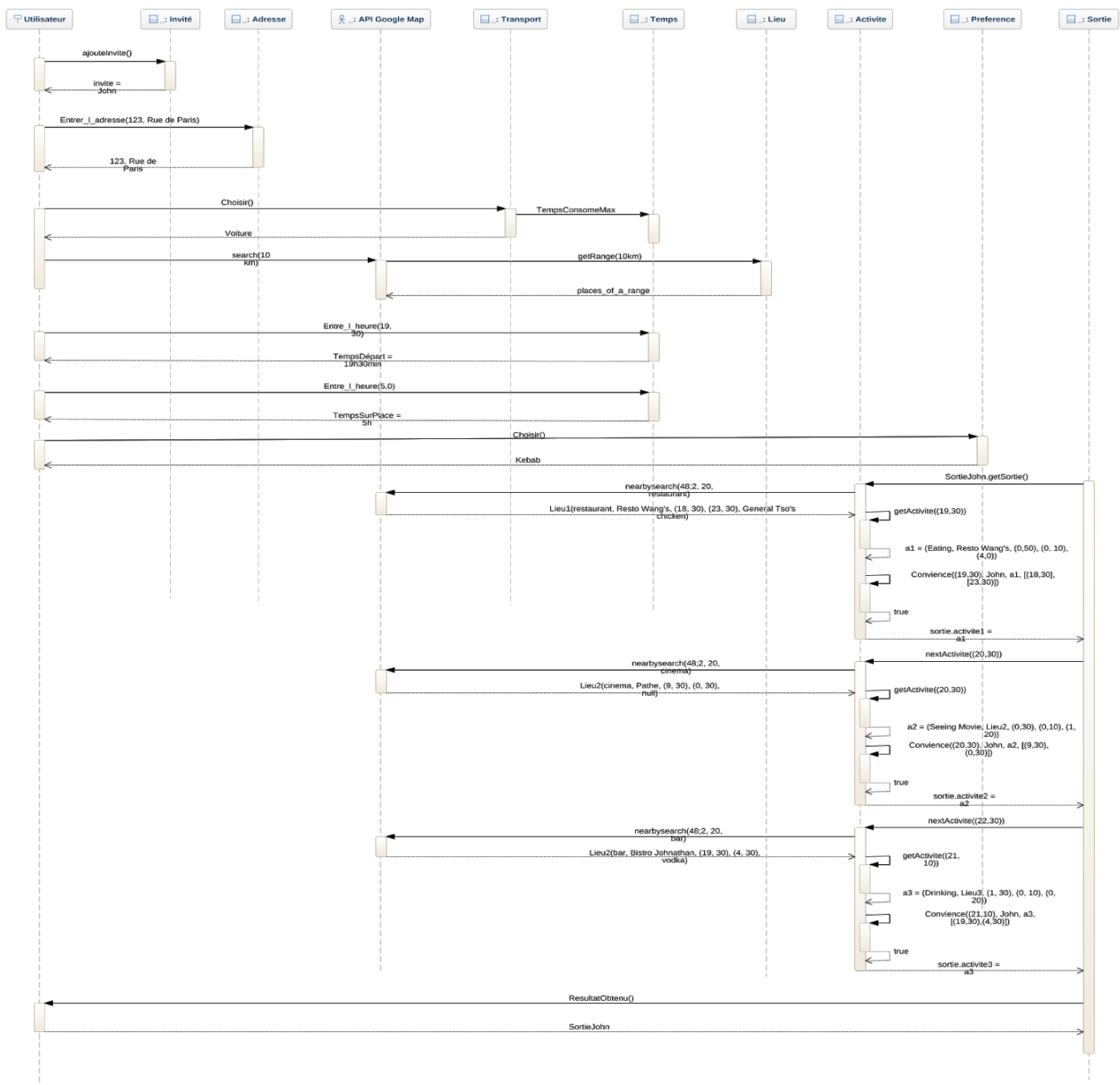
Les fonctions :

- Ajouter un invité : Ajouter un invité à la liste des invités en respectant le type des paramètres.
- Choisir des sorties dans la liste des sorties : Permet de choisir les sorties parmi la liste qui a été proposé lors de l'ajout de toutes les informations des invités.
- Consulter les sorties : Consulter les sorties après le choix des sorties par tous les invités.
- Ajouter une sortie non présente avec des informations cohérents : Après affichage de la liste des sorties, ajout d'une sortie non présente dans cette liste.
- Saisir son nom : Un invité doit avoir son nom saisi par l'utilisateur pour participer à la sortie.
- Saisir son adresse : Ajout de l'adresse d'un invité.
- Choisir ou saisir sa préférence alimentaire : Choisir parmi une liste la préférence alimentaire d'un invité ou propose un ajout à cette liste si non présente.
- Choisir le mode de transport : Ajout du mode de transport.
- Entrer la date et l'heure de départ : Ajout de la date et l'heure de départ.
- Entrer la durée estimée d'événement : Ajout de la durée de l'événement.

2. Diagrammes de séquence et fonctions utilisées

1. Ajouter un invité

a. Diagramme



b. Scénario

Identification :

Objectif : Permet de rajouter un invité

Acteur concerné : l'utilisateur

Fonctions:

ajouteInvite : Entrée : Ø

Sortie : nom de l'invité(String)

Entrer_I_adresse : Entrée : numero(int), rue(String)

Sortie : adresse(Adresse)

Choisir : Entrée : transport(String)

Sortie : transport(Transport)

TempsConsommeMax : Entrée : heures(int), minutes(int)

Sortie : Ø

getRange : Entrée : rayon(int)

Sortie : liste de lieux(Lieu)

Entre_I_heure : Entrée : heures(int), minutes(int)

Sortie : heure(Temps)

getSortie : Entrée : Ø

Sortie : activite(Activite)

getActivite : Entrée : heure(Temps)

Sortie : activite(Activite)

Convience : Entrée : activite(Activite)

Sortie : valide(booléen)

nextActivite : Entrée : heure(Temps)

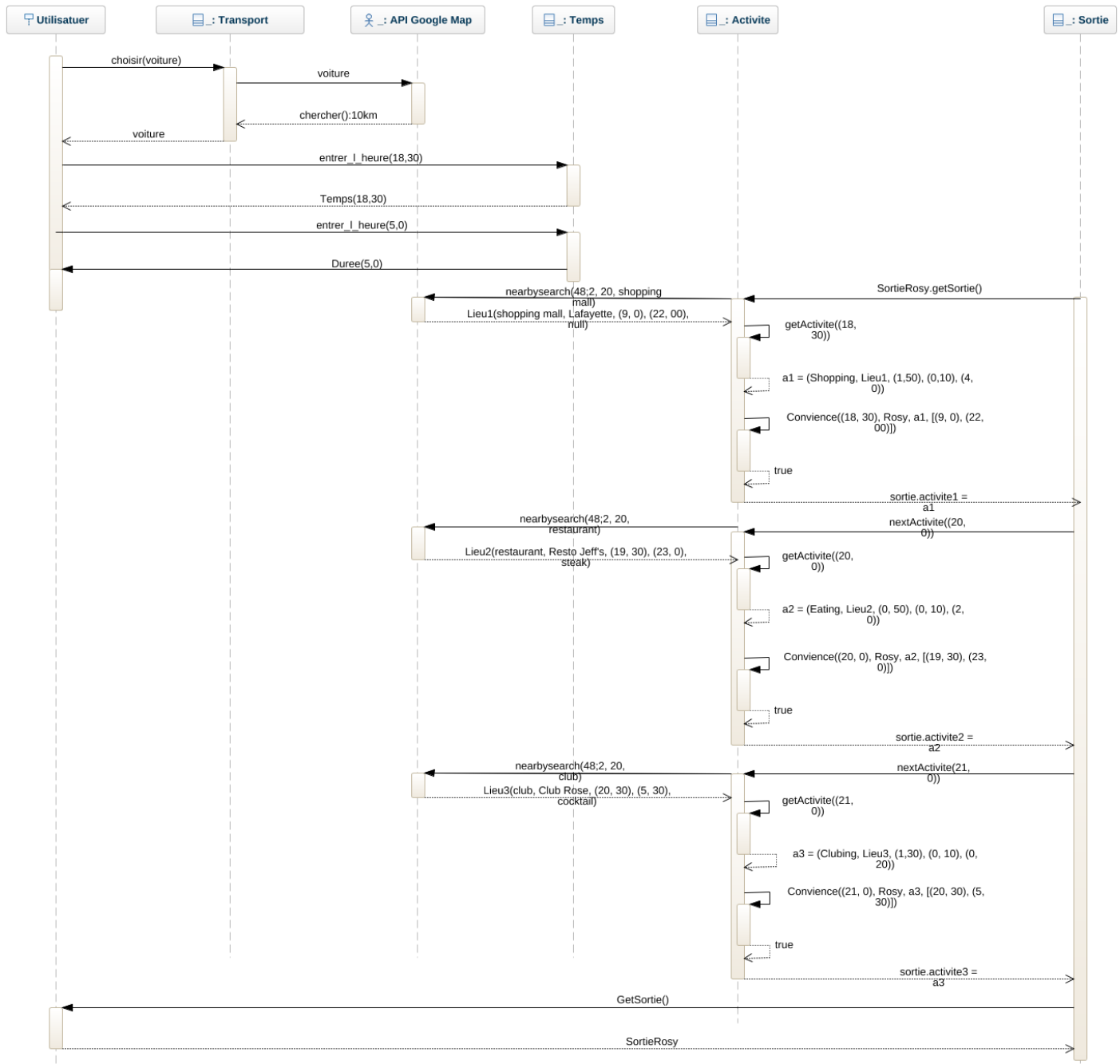
Sortie : activite(Activite)

ResultatObtenu : Entrée : Ø

Sortie : Sortie(sortie)

2. Choisir des sorties dans la liste des sorties

a. Diagramme



b.Scénario

Identification :

Objectif: détaille les étapes permettant à un l'utilisateur de choisir des sorties dans la liste des sorties

Acteur principal: l'utilisateur

Fonctions :

Choisir : Entrée : transport(String)

Sortie : transport(Transport)

Entrer_l_heure : Entrée : heures(int), minutes(int)

Sortie : heure(Temps)

getSortie : Entrée : Ø

Sortie : activite(Activite)

getActivite : Entrée : heure(Temps)

Sortie : activite(Activite)

Convience : Entrée : activite(Activite)

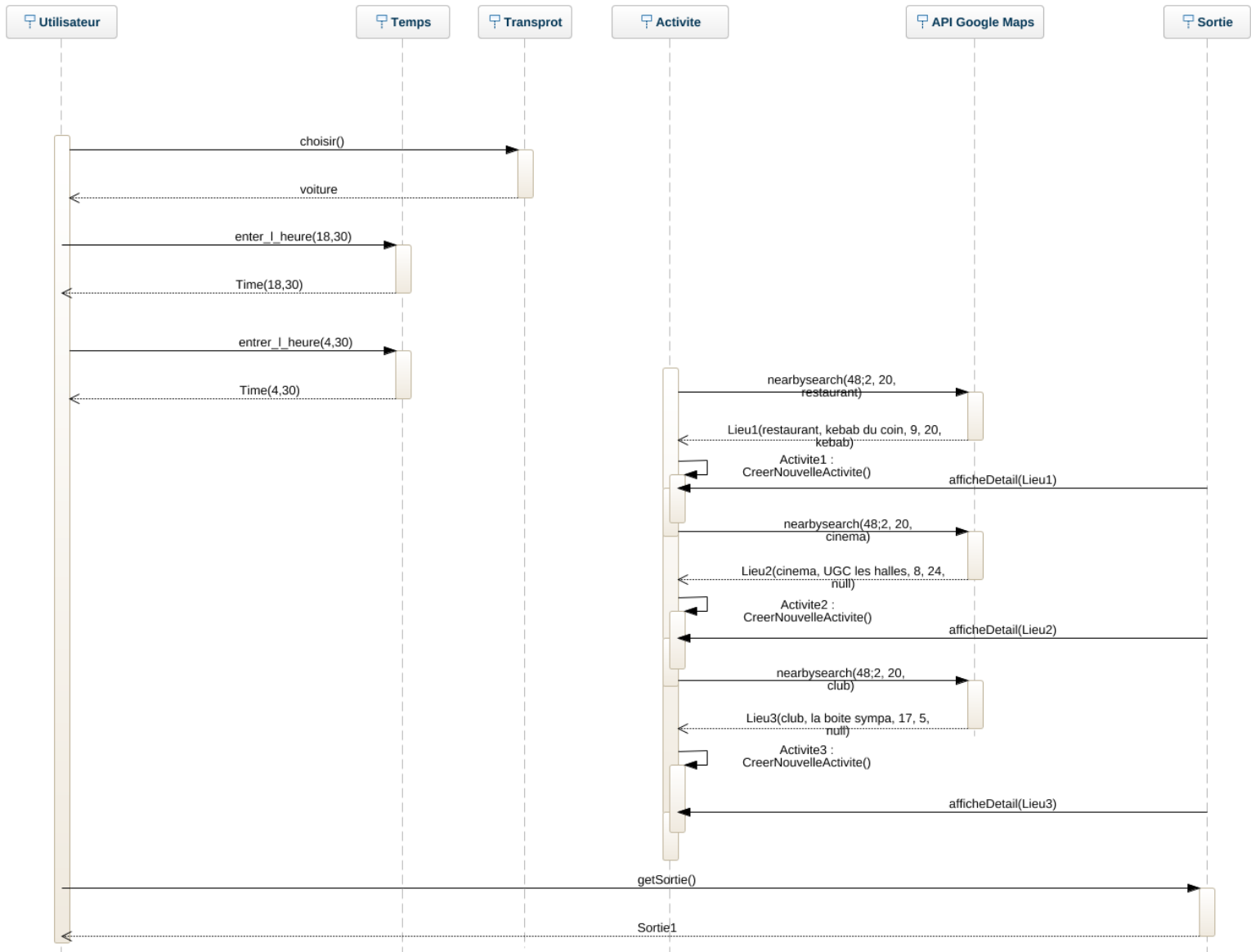
Sortie : valide(booléen)

nextActivite : Entrée : heure(Temps)

Sortie : activite(Activite)

3. Ajouter une sortie non présente avec informations cohérentes

a. Diagramme



b. Scénario

Identification :

Objectif : Permet de rajouter une sortie personnalisée parmi la liste des sorties proposées.

Acteur concerné : l'utilisateur

Fonctions :

Choisir : Entrée : transport(String)

Sortie : transport(Transport)

Entrer_l_heure : Entrée : heures(int), minutes(int)

Sortie : heure(Temps)

creerNouvelleActivite : Entrée : Ø

Sortie : activite(Activite)

afficheDetail : Entrée : lieu(Lieu)

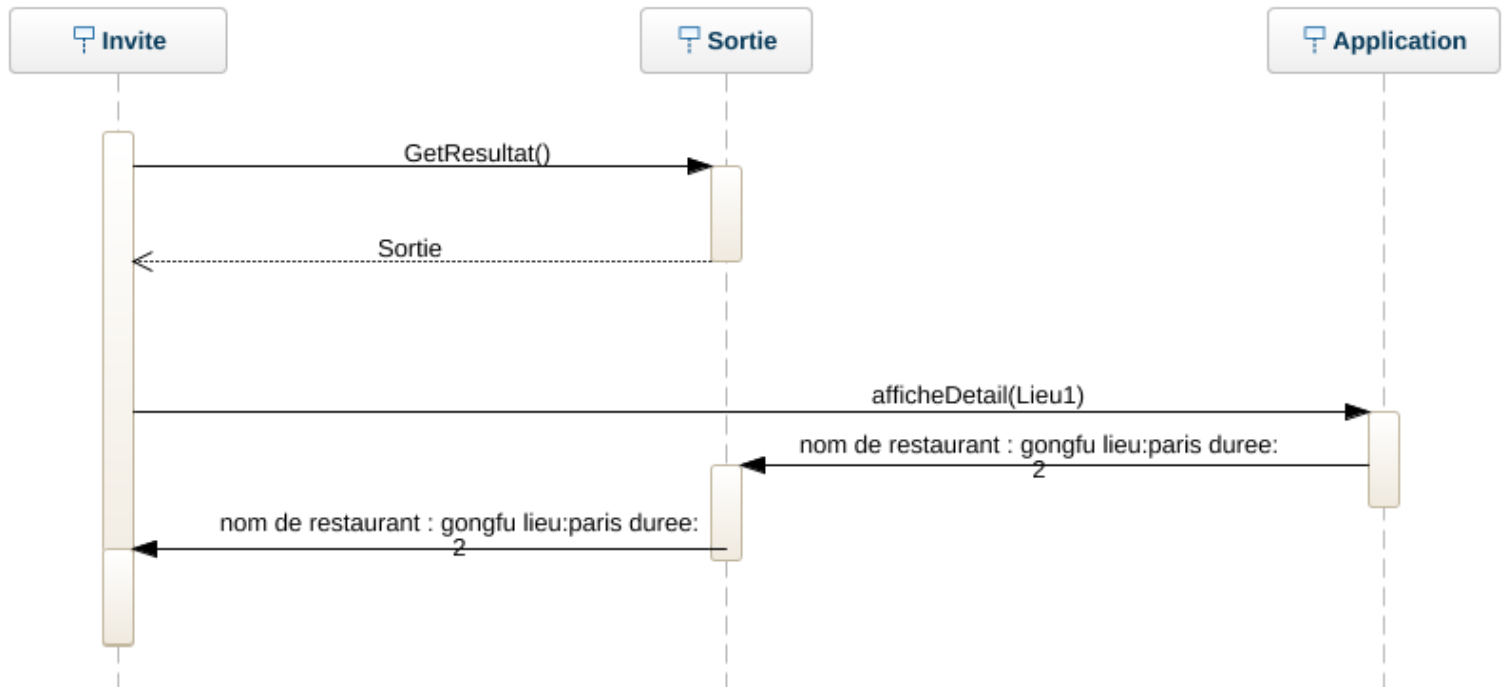
Sortie : Ø

getSortie : Entrée : Ø

Sortie : activite(Activite)

4. Consulter les sorties

a. Diagramme



b. Scénario

Identification :

Objectif: détaille les étapes permettant à un l'utilisateur consulter une sortie ou plusieurs sorties dans la liste

Acteur principal: l'invité ou l'utilisateur(qui est aussi un invité)

Fonctions :

GetResultat : Entrée : \emptyset

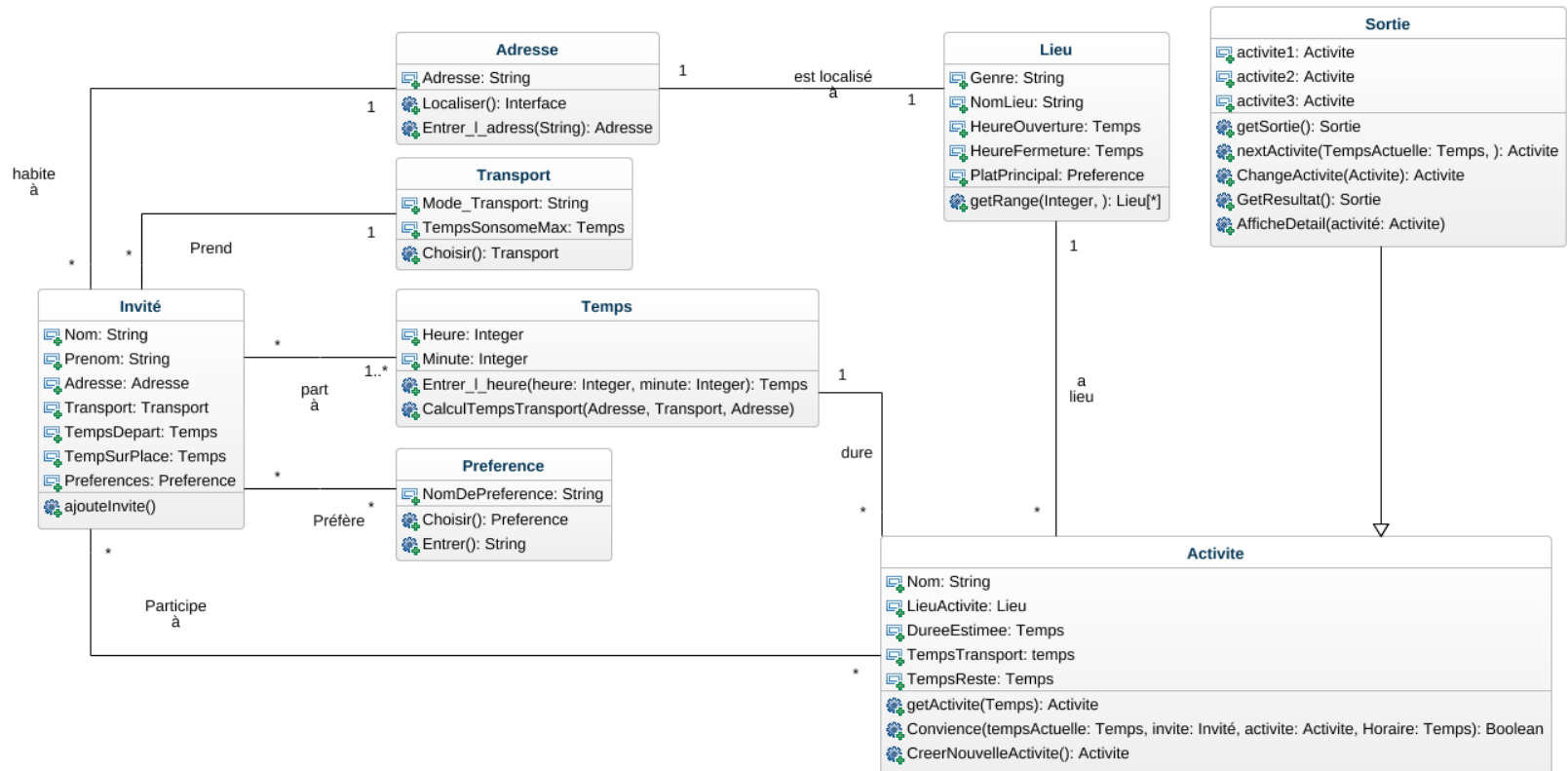
Sortie : sortie(Sortie)

afficheDetail : Entrée : \emptyset

Sortie : activite(Activite)

3. Diagramme de classe

1. Diagramme



2. Explications

Descriptions des classes:

(1), Invite

- (i), **Nom**: String.
- (ii), **Prenom**: String.
- (iii), **Adresse**: Adresse
- (iv), **Transport**: Transport

- (v), **TempsDepart**: Temps
- (vi), **TempsSurPlace**: Temps
- (vii), **Preferences**: Preference

(i), **ajouteInvite()**

(2). Adresse

- (i), **Adresse**: String
- (i), **Localiser()**: Interface
- (ii), **Entrer_I_adresse(String)**: Adresse

(3). Transport

- (i), **Mode_Transport**: String
- (ii), **TempsConsommeMax**: Temps
- (i), **Choisir()**: Transport

(4). Temps

- (i), **Heure**: Integer
- (ii), **Minute**: Integer
- (i), **Entrer_I_heure(Integer, Integer)**: Temps
- (ii), **CalculTempsTransport(Adresse, Transport, Adresse)**

(5). Preference

- (i), **NomDePreference**: String
- (i), **Choisir()**: Preference
- (ii), **Entrer()**: String

(6). Lieu

- (i), **Genre**: String
- (ii), **NomLieu**: String
- (iii), **HeureOuverture**: Temps
- (iv), **HeureFermeture**: Temps
- (v), **PlatPreference**: Preference

- (i), **GetRange(Integer)**: [Lieu]

(7). Activite

- (i), **Nom**: String
 - (ii), **LieuActivite**: Lieu
 - (iii), **DureeEstimée**: Temps
 - (iv), **TempsTransport**: Temps
 - (v), **TempsReste**: Temps
-
- (i), **getActivite(Temps)**: Activite
 - (ii), **Convience(Temps, Invité, Activité, Temps)**: Boolean
 - (iii), **CreerUneNouvelle()**: Activite

(8). Sortie

- (i), **activite1**: Activite
 - (ii), **activite2**: Activite
 - (iii), **activite3**: Activite
-
- (i), **getSortie()**: Sortie
 - (ii), **nextActivite(Temps)**: Activite
 - (iii), **changerActivite(Activite)**: Activite
 - (iv), **getResultat()**: Sortie
 - (v), **afficheDetail(Activite)**

Relations entre les objets engendrés par les classes:

Classe	Objet engendré
Invite	invité
Adresse	adresse
Transport	transport
Temps	L'heure de départ, L'heure sur place, L'heure de transport, L'heure d'ouverture, L'heure de fermeture.
Preference	préférence
Lieu	lieu
Activite	activité