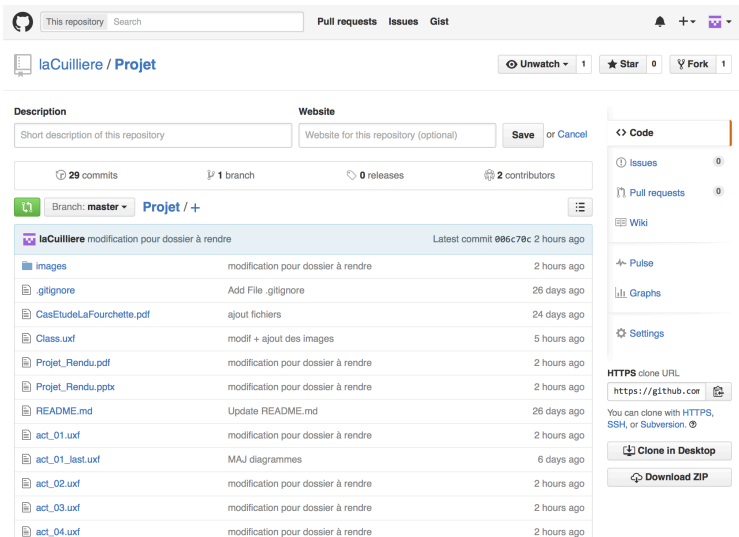


La Cuillière

Dans le cadre d'un projet UMLAgile, nous avons modélisé notre application de guide culinaire interactif (fortement inspiré de LaFourchette).

Afin de mener à bien ce projet, nous avons appliqué les méthodes agiles via différents outils : Trello + Github.



Backlog

Sprint 1 Todo

In progress

Tested

Done



Backlog

Use case _ API

Class

Activity_UC01 Rechercher un restaurant

Activity_UC02 Réserver un restaurant

Activity_UC03 Avis

Activity_UC04 Gérer restaurant

Activity_UC05 Gérer offres/cartes

StateMachine_Reservation

Activity_UC06 Gérer réservation

State machines

Ajouter une carte...

Sprint 1 Todo

Ajouter une carte...

In progress

Ajouter une carte...

Tested

Ajouter une carte...

Done

Ajouter une carte...

Backlog

Class

Activity_UC03 Avis

Activity_UC06 Gérer réservation

Activity_UC05 Gérer offres/cartes

Activity_UC04 Gérer restaurant

StateMachine_Reservation

State machines

Composant

Composants

StateMachine_Deal

StateMachine_Avis

Innovations

Ajouter une carte...

Sprint 1 Todo

Use case _ API

Activity_UC01 Rechercher un restaurant

Activity_UC02 Réserver un restaurant

Mettre en place GIT

Ajouter une carte...

In progress

Ajouter une carte...

Tested

Ajouter une carte...

Done

Ajouter une carte...

Backlog

Class

StateMachine_Reservation

State machines

Composant

Composants

StateMachine_Deal

StateMachine_Avis

Innovations

Activity_Team

Ajouter une carte...

Sprint 2 Todo

Activity_UC03 Avis

Activity_UC04 Gérer restaurant

Activity_UC05 Gérer offres/cartes

Activity_UC06 Gérer réservation

Ajouter une carte...

In progress

Use case _ API

Activity_UC01 Rechercher un restaurant

Activity_UC02 Réserver un restaurant

Ajouter une carte...

Tested

Ajouter une carte...

Done

Mettre en place GIT

Ajouter une carte...

Backlog

Composant

Composants

StateMachine_Deal

StateMachine_Avis

Innovations

Activity_Team

Ajouter une carte...

Sprint 3 Todo

Class

StateMachine_Reservation

State machines

Ajouter une carte...

In progress

Activity_UC03 Avis

Activity_UC04 Gérer restaurant

Activity_UC05 Gérer offres/cartes

Activity_UC06 Gérer réservation

Ajouter une carte...

Tested

Use case _ API

Activity_UC01 Rechercher un restaurant

Activity_UC02 Réserver un restaurant

Ajouter une carte...

Done

Mettre en place GIT

Ajouter une carte...

Backlog

Innovations

Activity_Team

Ajouter une carte...

Sprint 4 Todo

StateMachine_Deal

StateMachine_Avis

Composants

Composant

Ajouter une carte...

In progress

Class

StateMachine_Reservation

State machines

Ajouter une carte...

Tested

Activity_UC03 Avis

Activity_UC04 Gérer restaurant

Activity_UC05 Gérer offres/cartes

Activity_UC06 Gérer réservation

Ajouter une carte...

Done

Mettre en place GIT

Use case _ API

Activity_UC01 Rechercher un restaurant

Activity_UC02 Réserver un restaurant

Ajouter une carte...

Backlog

Ajouter une carte...

Sprint 5 Todo

Ajouter une carte...

In progress

Ajouter une carte...

Tested

Ajouter une carte...

Done

Activity_UC05 Gérer offres/cartes

Activity_UC06 Gérer réservation

Class

StateMachine_Reservation

State machines

StateMachine_Deal

StateMachine_Avis

Composants

Composant

Innovations

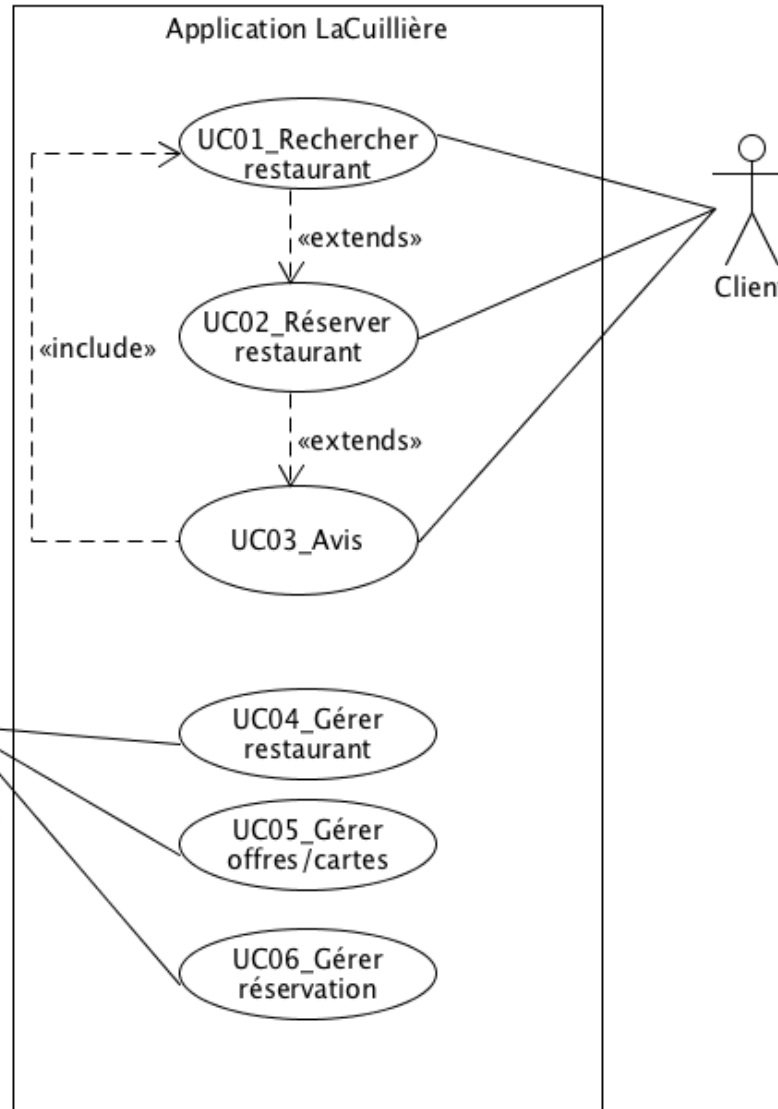
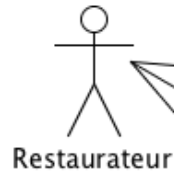
Activity_Team

Ajouter une carte...

UC04
 1. Le restaurateur peut créer une fiche restaurant
 2. Il peut supprimer un restaurant
 3. Il peut mettre à jour sa fiche restaurant
 4. Il peut voir les avis sur son restaurant

UC05
 1. Le restaurateur peut ajouter une offre
 2. Il peut modifier une offre (à supprimer)
 3. Il peut ajouter une carte
 4. Il peut modifier la carte

UC06
 1. Le restaurateur peut consulter la liste des réservations
 2. Il peut confirmer que le client est bien venu
 3. Il peut annuler une réservation



UC01
 1. Le client
 * choisit parmi nos sélections
 - les plus réservés
 - les meilleurs avis
 - les meilleurs promos
 (offres personnalisées en fonction des précédents repas du client)
 * et/ou entre ses critères :
 - géolocalisation (et/ou)
 - localisation
 - nb de personnes
 - date
 - heure
 - mot clé (nom, adresse, métro)
 - type cuisine

UC02
 1.1 Le client peut choisir/modifier :
 - date
 - heure
 - nb de personnes
 1.1.2 Le client valide la réservation
 1.2 Le client peut supprimer sa réservation
 1.2.1 Le client valide la réservation

UC03
 1. Le client peut consulter les avis d'un restaurant depuis sa page
 1.1 Le client peut commenter des avis
 1.2 Le client peut noter la pertinence d'un avis (+ ou -)
 2. Le client peut ajouter un avis
 3. Le client peut modifier son

Cas d'utilisation

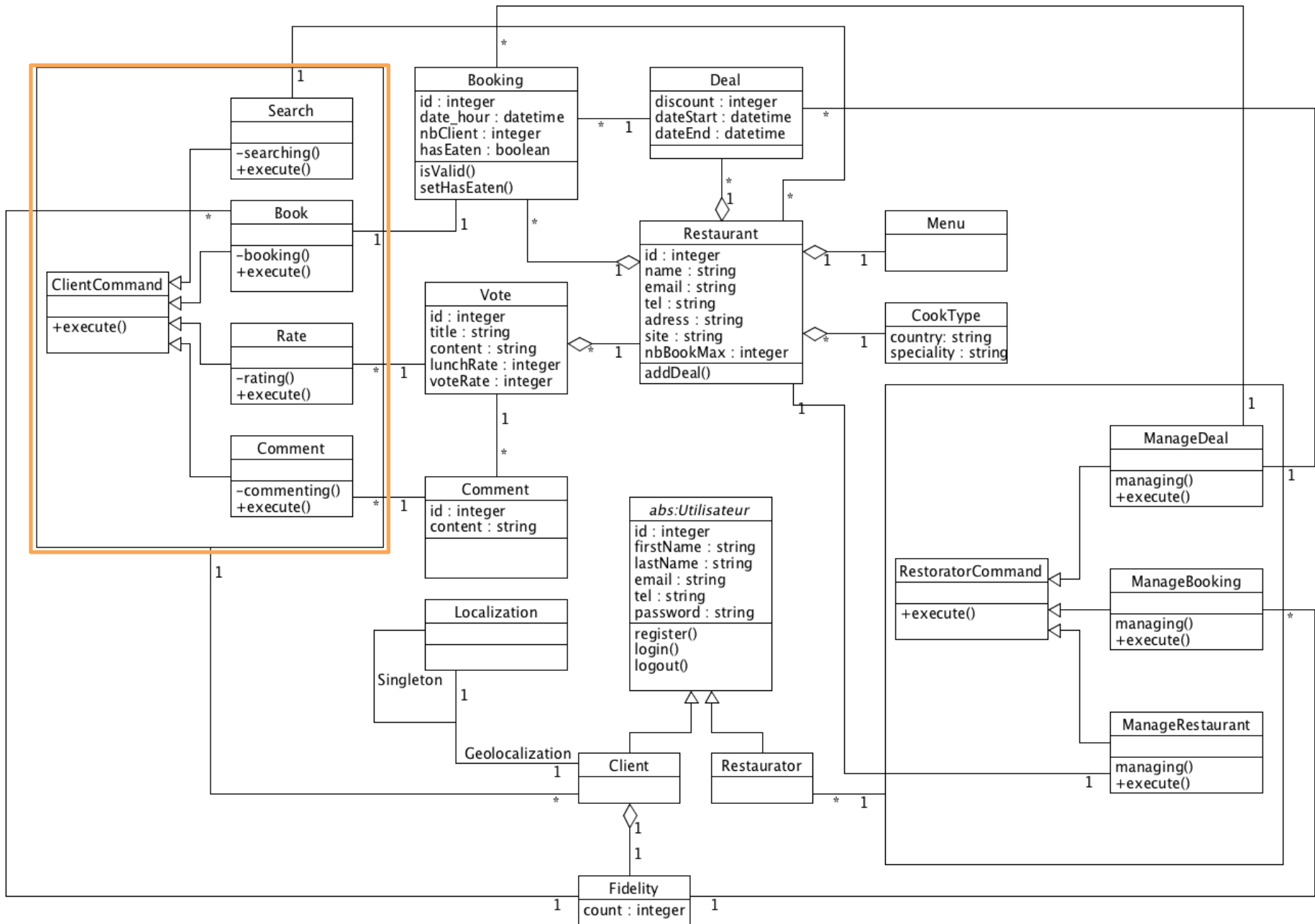
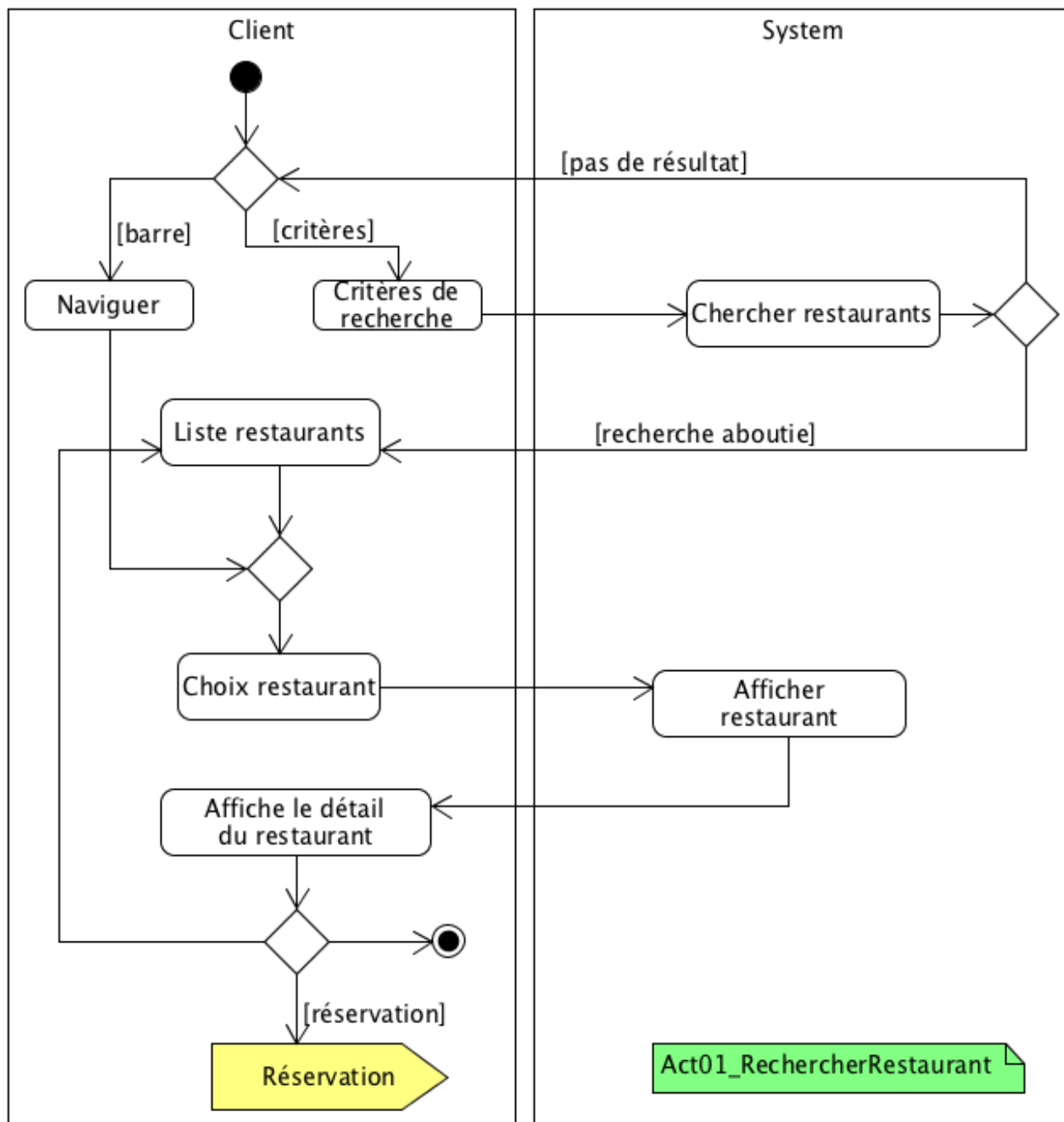
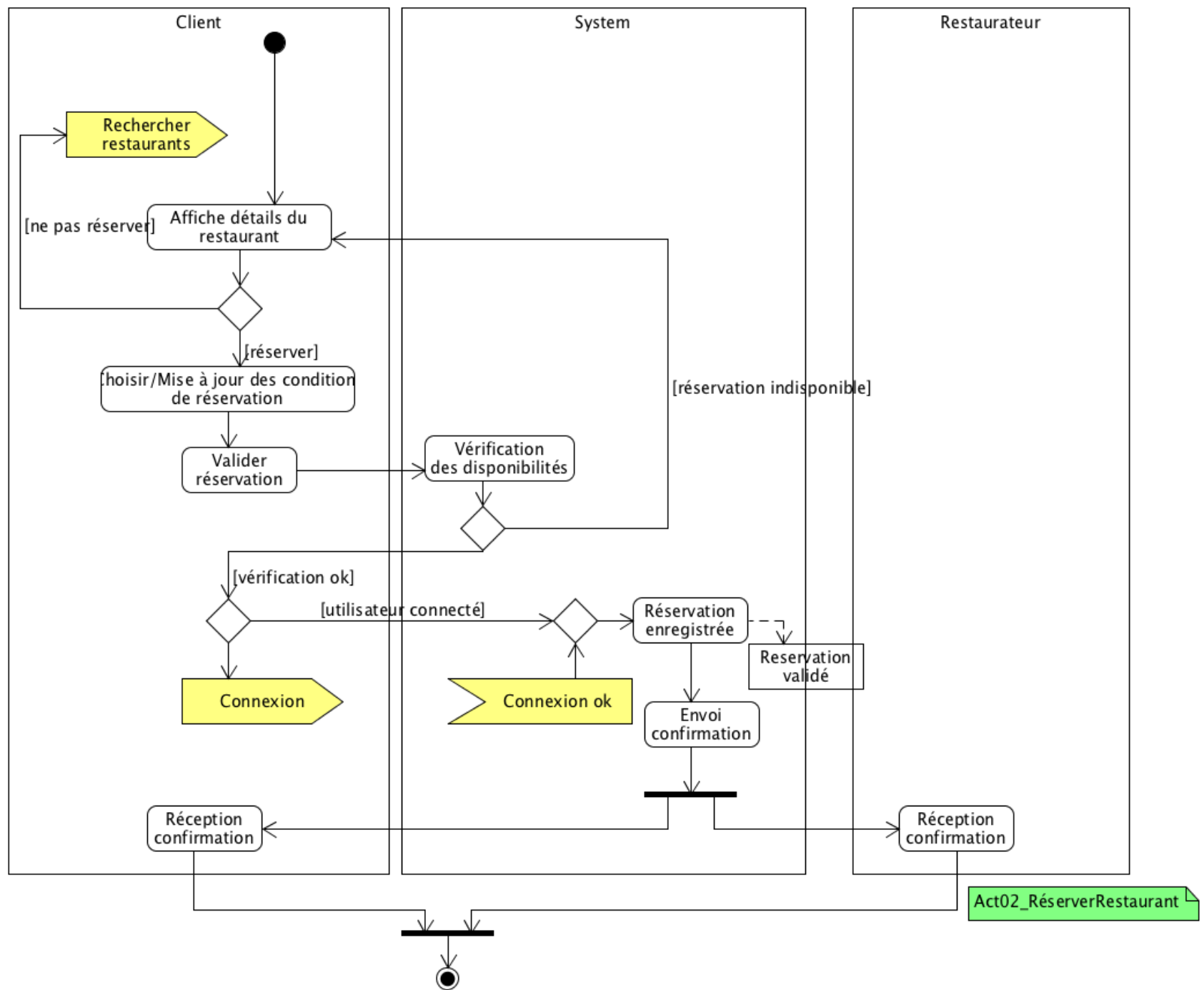
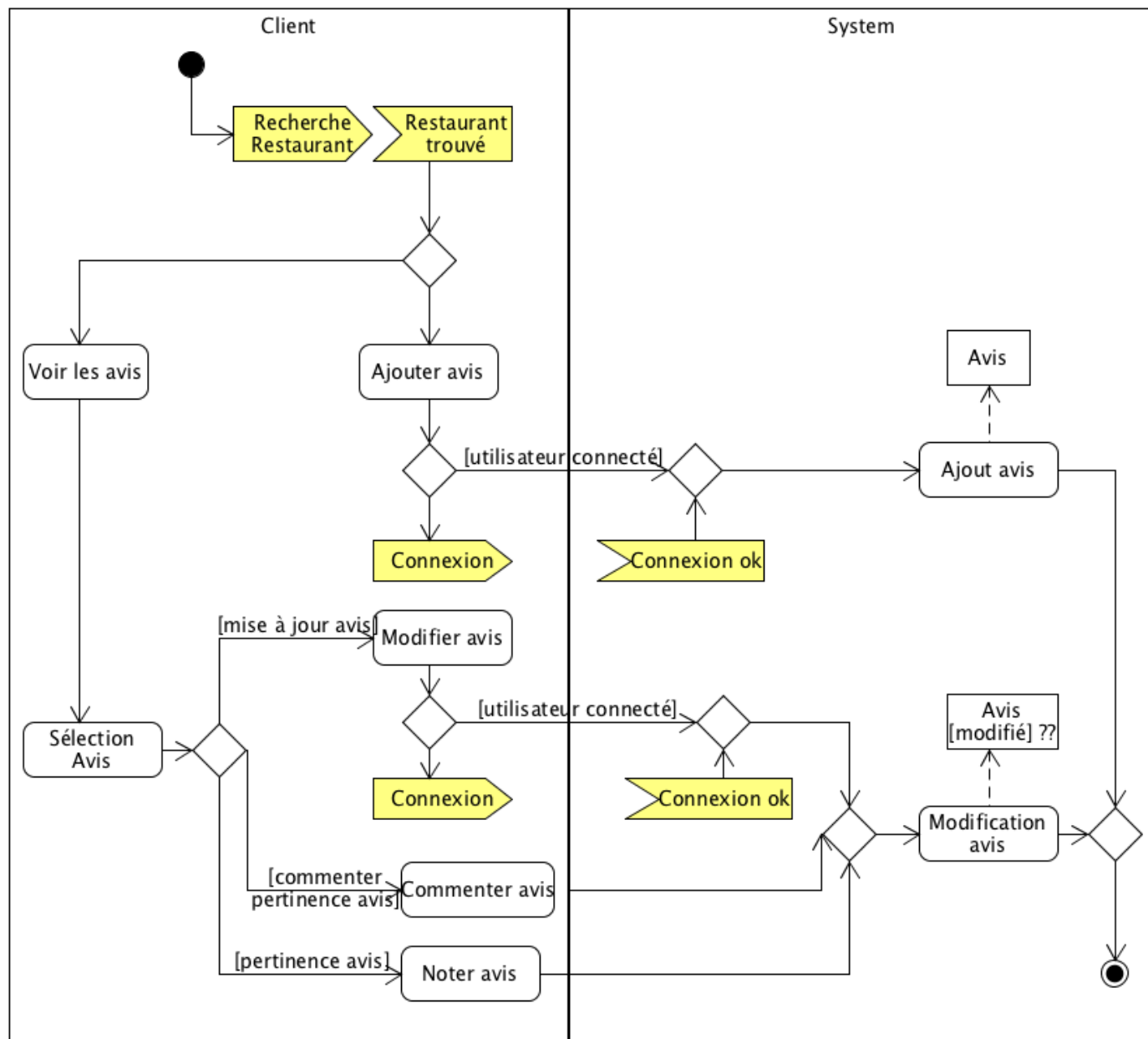


Diagramme de classe

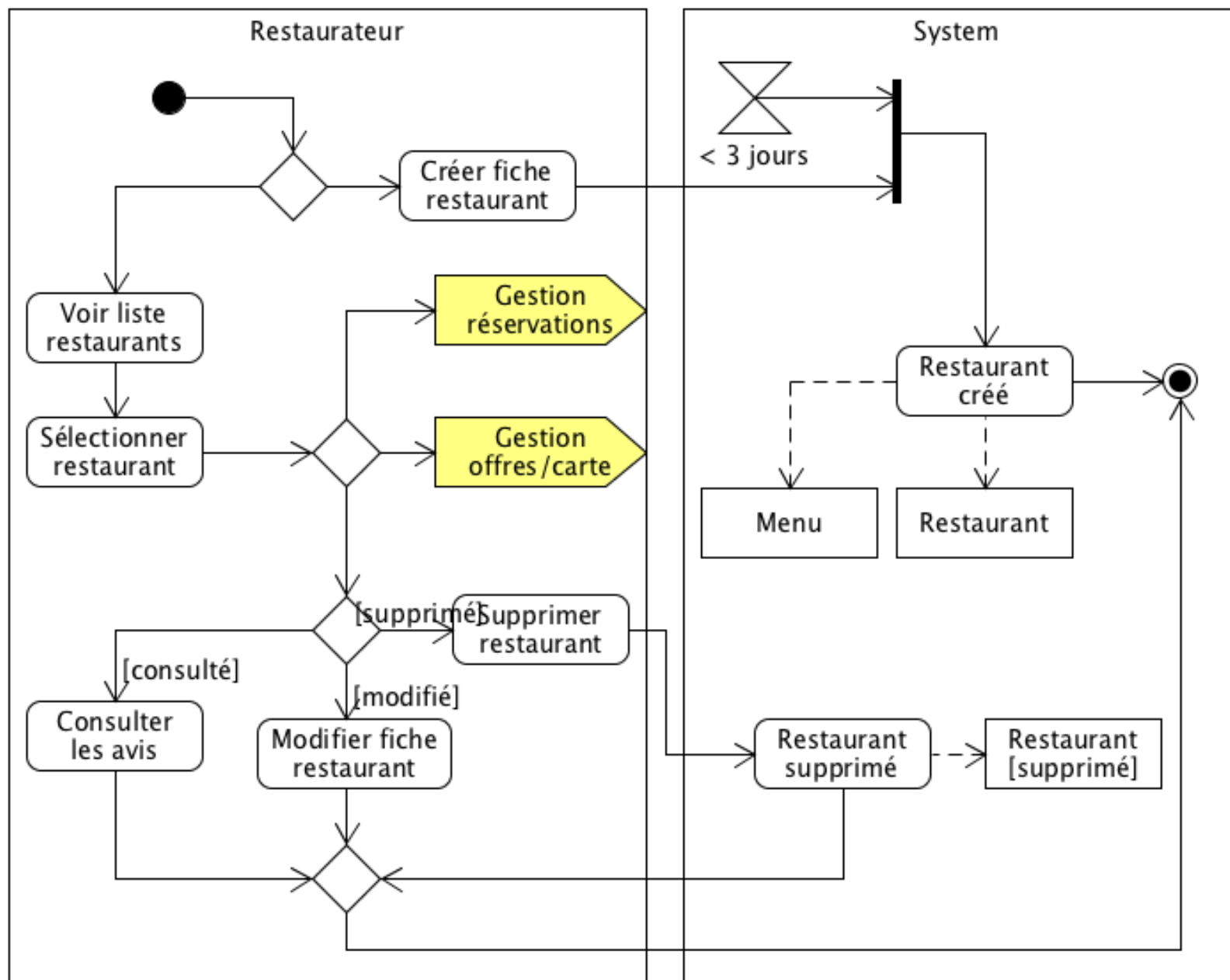




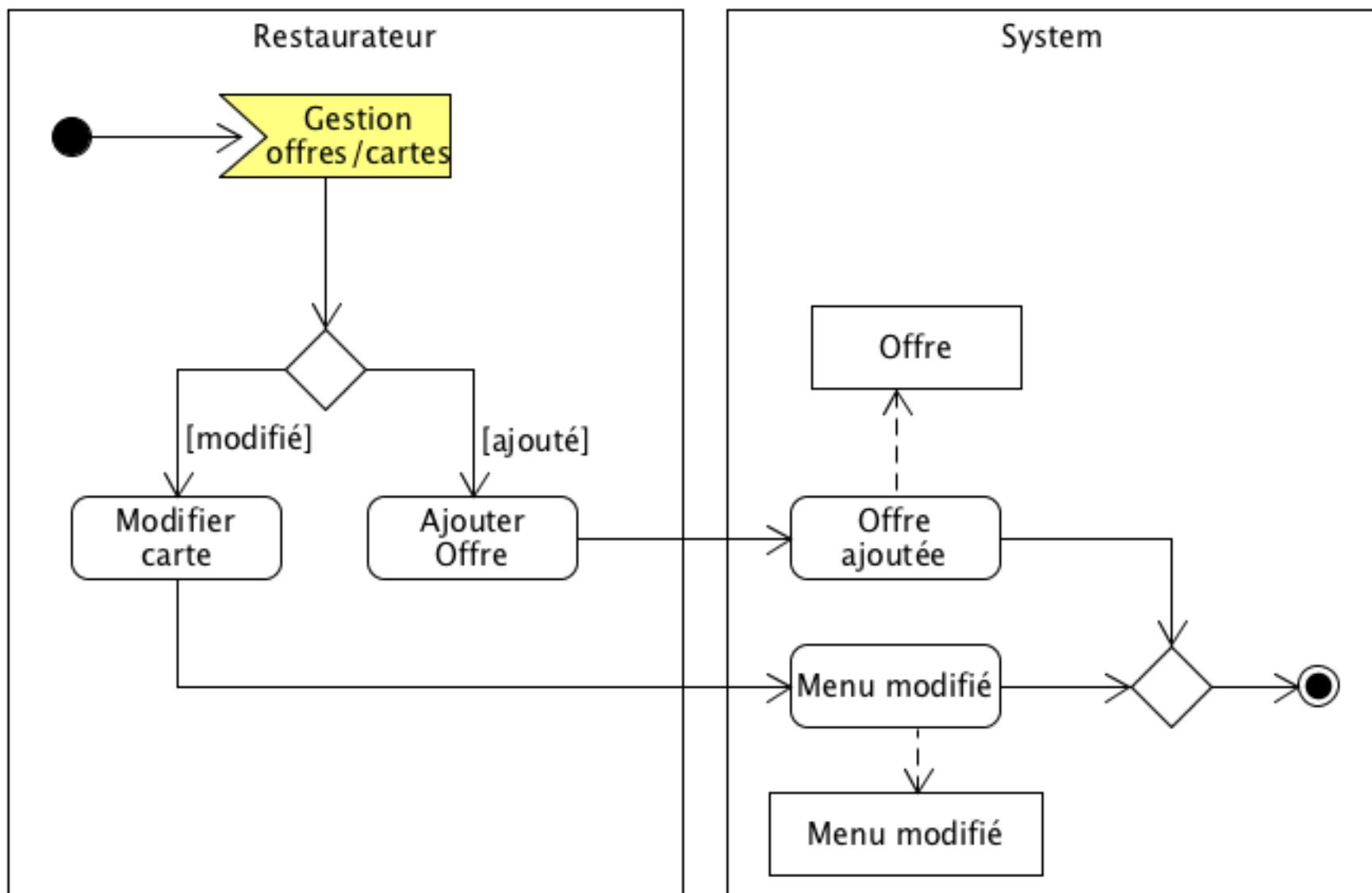


précision sur ajouter avis c'est :
 - mettre une note
 - mettre un commentaire

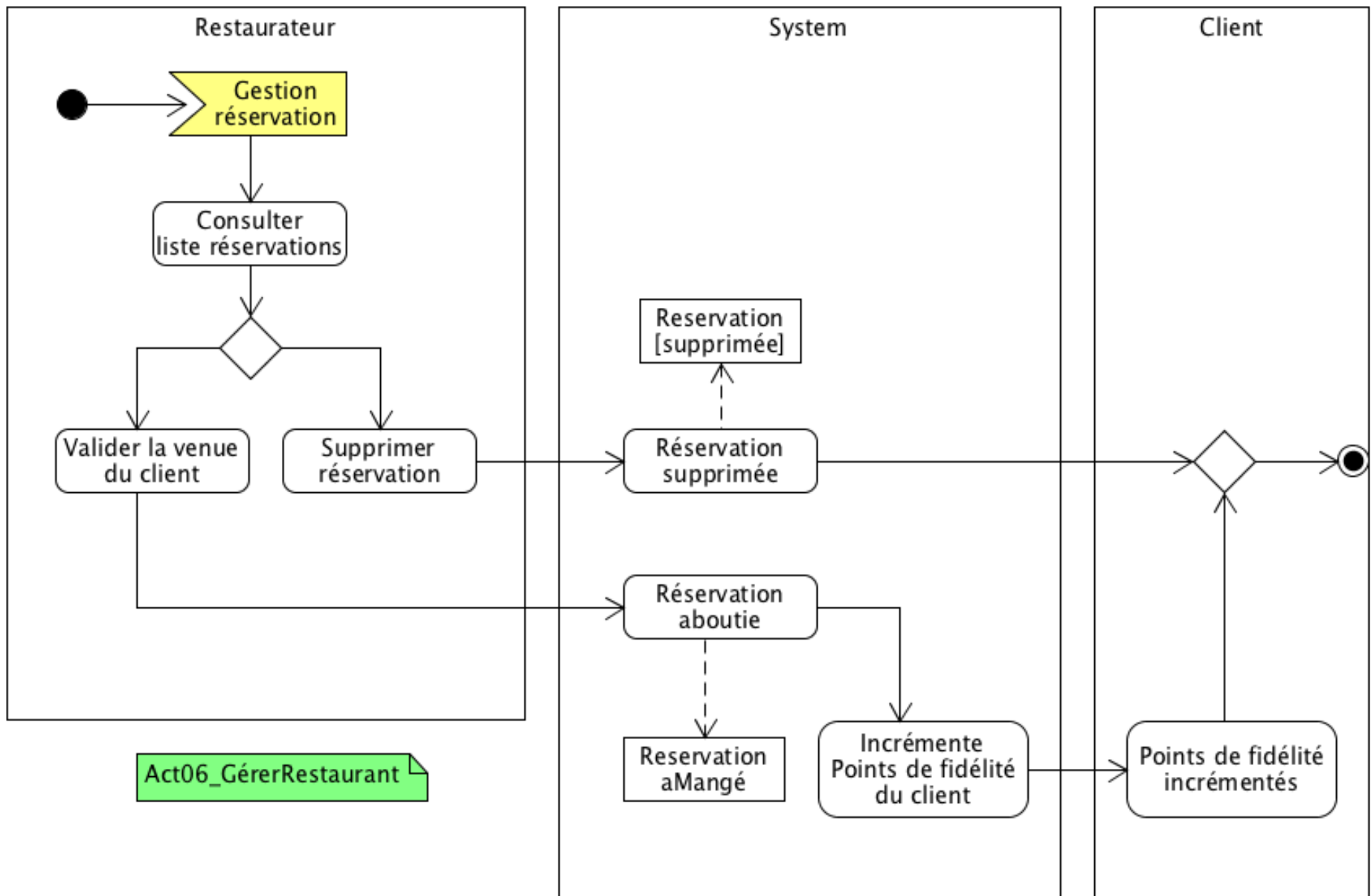
Act03_Avis



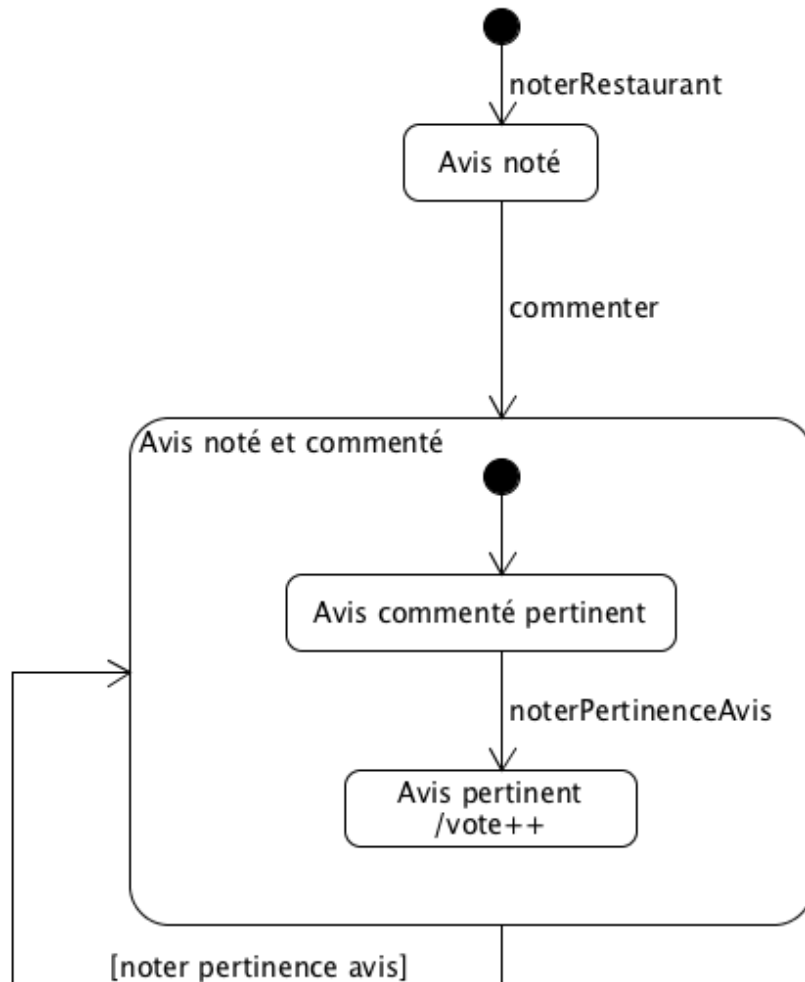
Act04_GérerRestaurant



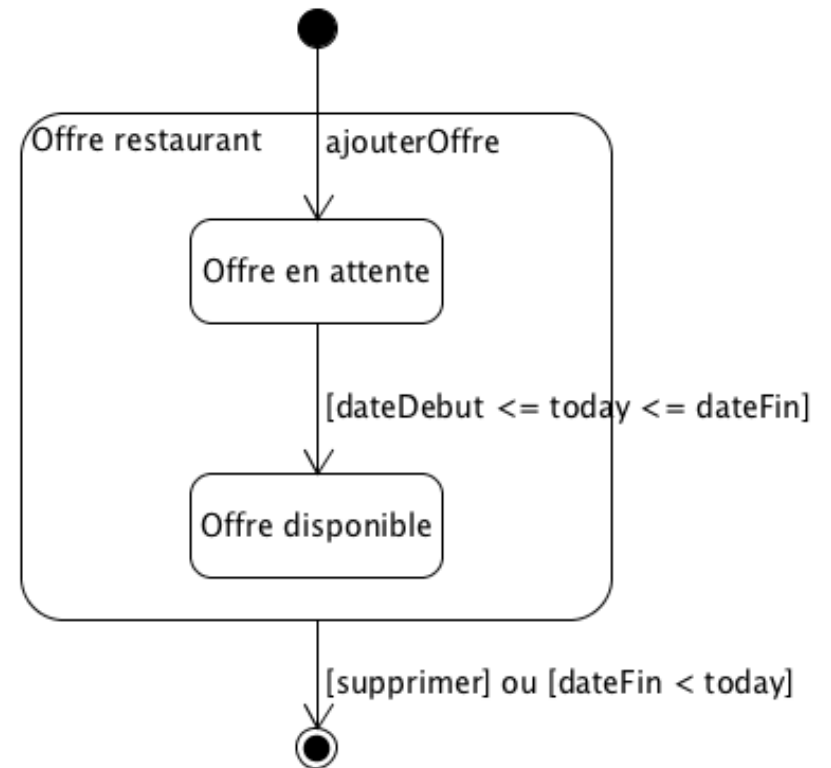
Act05_GérerOffres/Cartes



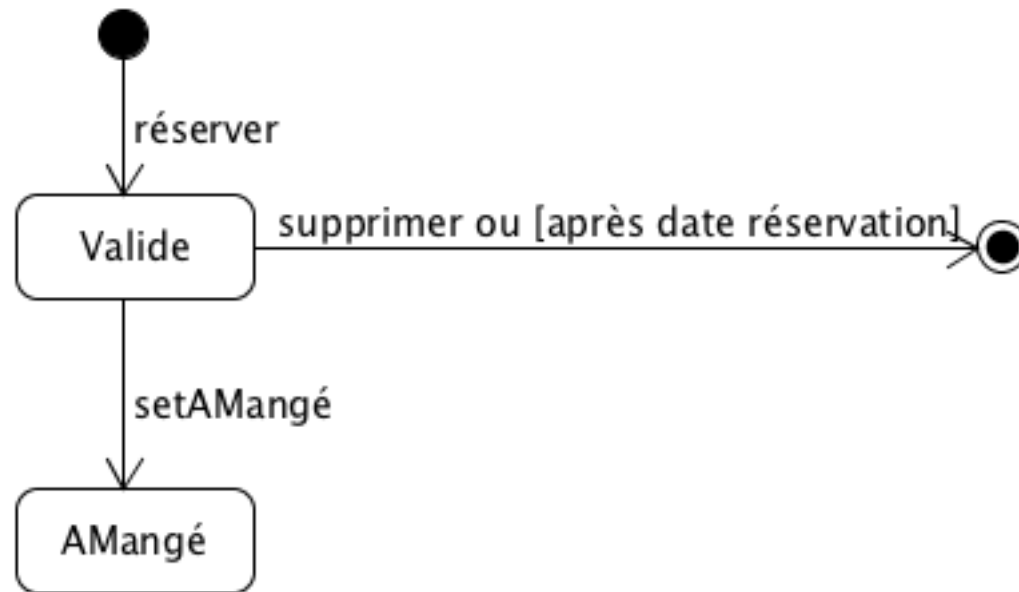
Etat/Transition : Avis

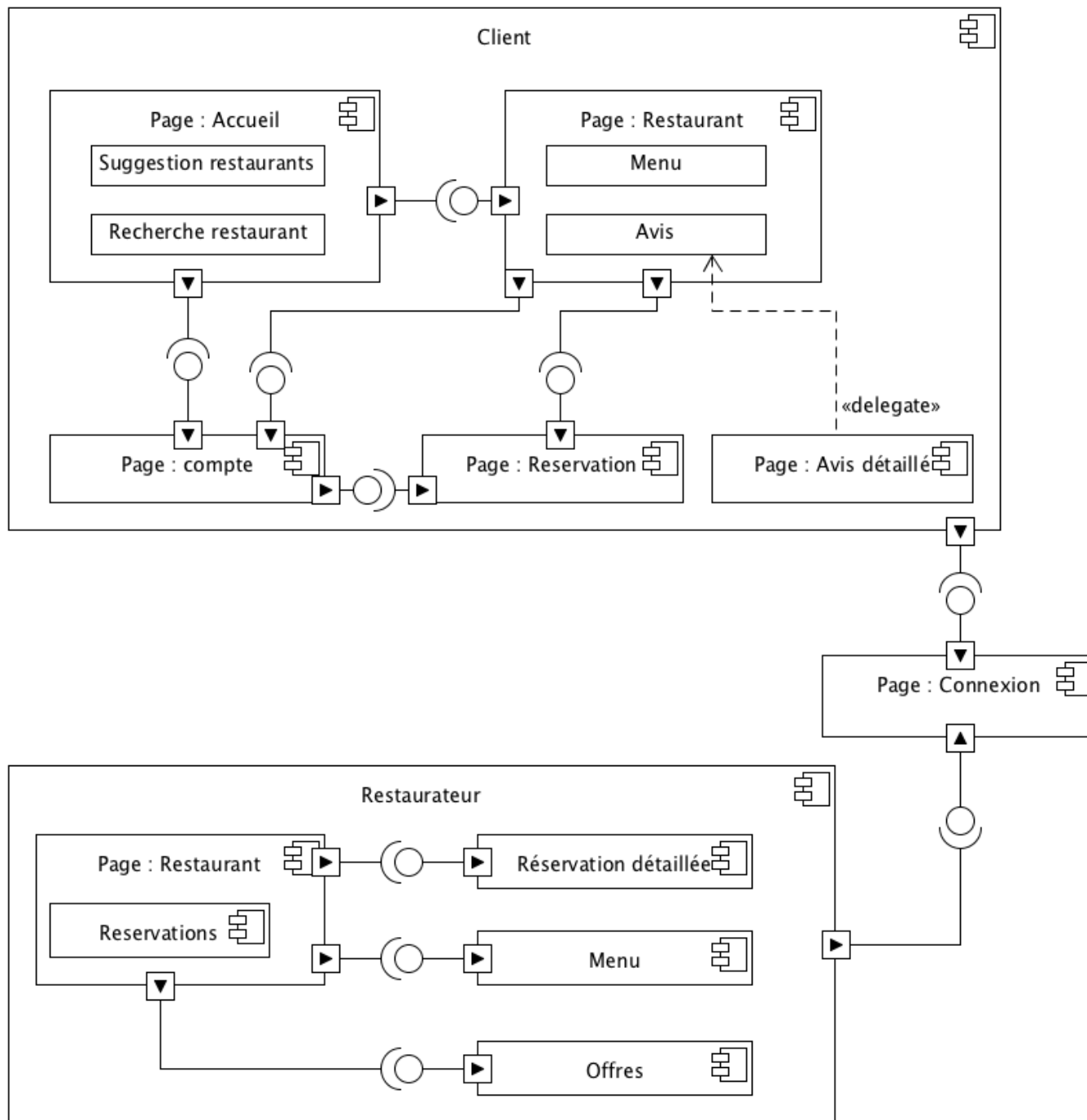


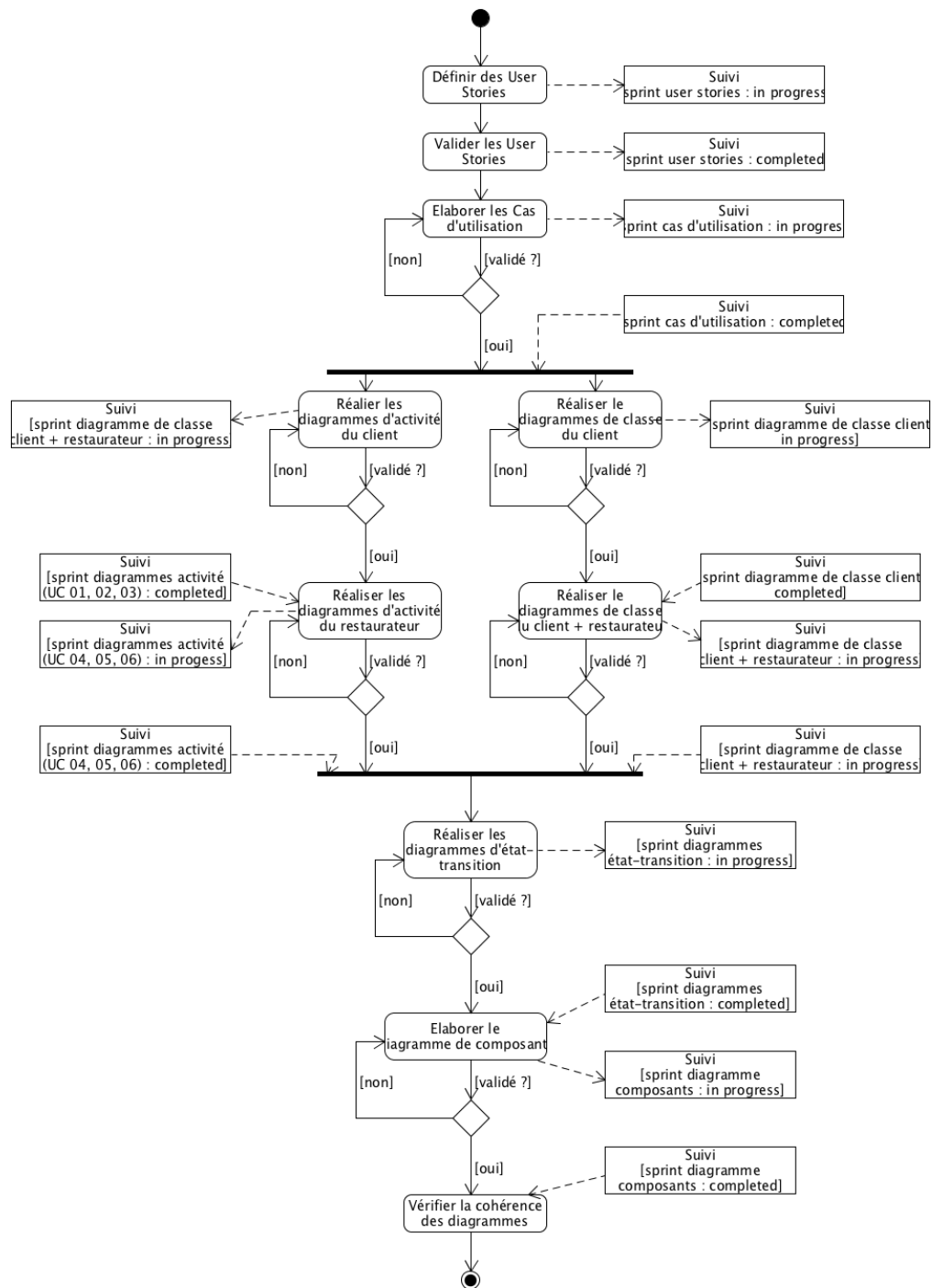
Etat/Transition : Offre



Etat/Transition : Réserveation







Innovations

Tout d'abord, la partie encadrée en orange sur le diagramme de classe (page3) fait partie des innovations proposées (proposées sous forme de pattern) :

- l'utilisateur peut commenter les avis d'un autre utilisateur
- l'utilisateur peut noter l'avis d'un autre utilisateur

D'autres innovations non présentes sur le diagramme de classe :

- le système de géolocalisation permet à l'utilisateur lorsqu'il passe à une certaine tranche horaire définie par le restaurateur à un lieu précis de recevoir une notification avec une réduction spécifique
- on pourrait aussi implémenter un système de follower comme sur twitter pour pouvoir suivre les aventures culinaires d'un autre utilisateur