



# Génie logiciel

## Analyse

Louis-Edouard LAFONTANT





# Analyse

---

*Le flux des exigences nous permet de comprendre le domaine et d'élucider les besoins du client.*

*⇒ Réponse à: Dans quoi on s'embarque?*

*⇒ Nouvelle question: De quoi avons-nous besoin? Quel est le plan?*

- Analyser les exigences sur un plan technique
- **Définir les contraintes de la conception et l'implémentation**

# Analyse

- Produire un **modèle d'analyse** du système
  - ✓ Correct
  - ✓ Complet
  - ✓ Cohérent
  - ✓ Vérifiable
- Obtenir une compréhension méticuleuse des exigences
  - Décrire les exigences pour produire une conception et une implémentation répondant aux contraintes.
- Structurer et formaliser les exigences
- Pas nécessairement compréhensible par le client

# Partie à informatiser

- Dépend de combien le client est prêt à dépenser
- Analyse du **coût/bénéfice** pour les différentes stratégies possibles
- Petit volume, ordinateur sur place  $\Rightarrow$  **Mode interactif**
  - Tâche requiert intervention **manuelle**
  - Peut nécessiter plus de ressources humaines
  - Utilisation **sur demande**
- Grand volume, contrôle détaillé  $\Rightarrow$  **Traitement en batch**
  - Exécution de **tâche** sans intervention manuelle
  - Moins de **ressources humaines** requises
  - **Taux d'utilisation** élevé

# Ressources nécessaires

## Base de données

- Tables (colonnes), données (rangées)
- Clé primaire, index, contraintes

## Disques durs (internes, externes)

## Puissance de calcul

- CPU, mémoire, réseau, graphique

## Définir l'emplacement de chaque donnée et de chaque fichier

# Exigences conceptuelles

- Les exigences conceptuelles déterminent comment l'information sera enregistrées et manipulées
- Modèle de données
  - Volume d'entrée (par jour, heure)
  - Taille, fréquence de chaque rapport à imprimer
- Format de données
- Format envisagé pour stocker les données
- Choix d'abstraction
  - Langage de programmation
  - Entités sauvegardées en base de données
- Droits d'accès

# Exigences physiques (matériels)

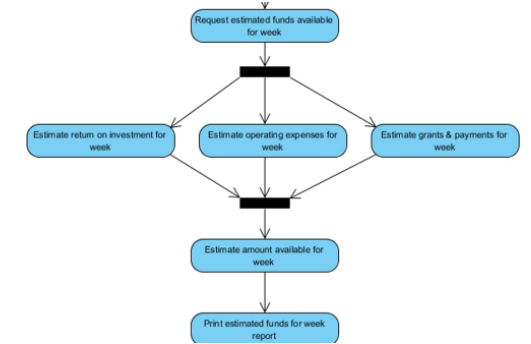
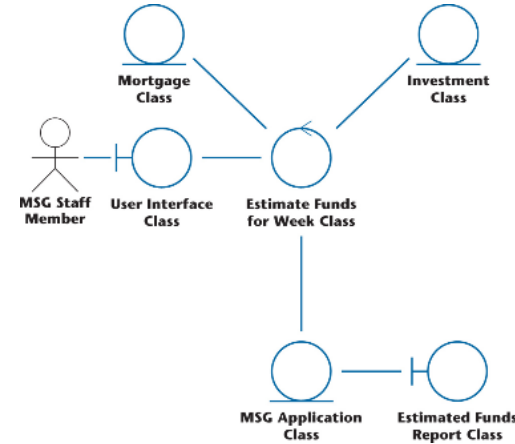
Les exigences physiques déterminent le support nécessaire pour l'exploitation du logiciel et les mécanismes d'interactions

- Mémoire de masse
- Stockage pour les sauvegardes
- Besoins pour les entrées
- Appareils de sortie
- Le matériel existant est-il adéquat ?
  - Recommander d'acheter ou de louer du matériel additionnel

# Modèle d'analyse

- **Modèle fonctionnel**

= Fonctionnalités du système  
→ Cas d'utilisations, scénarios

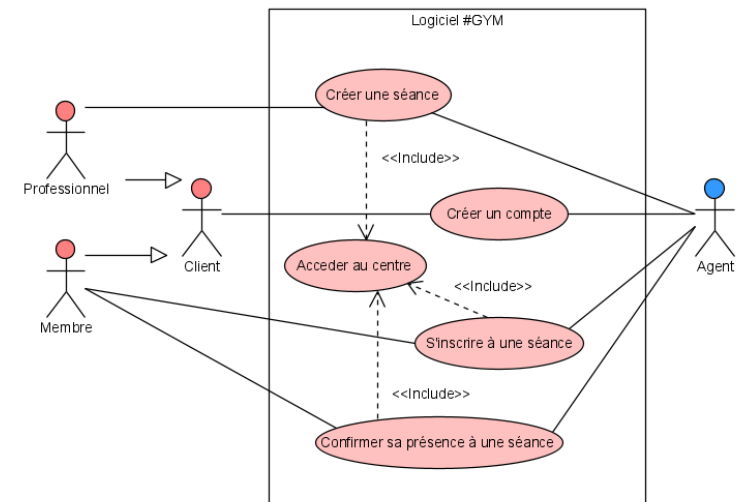


- **Modèle dynamique**

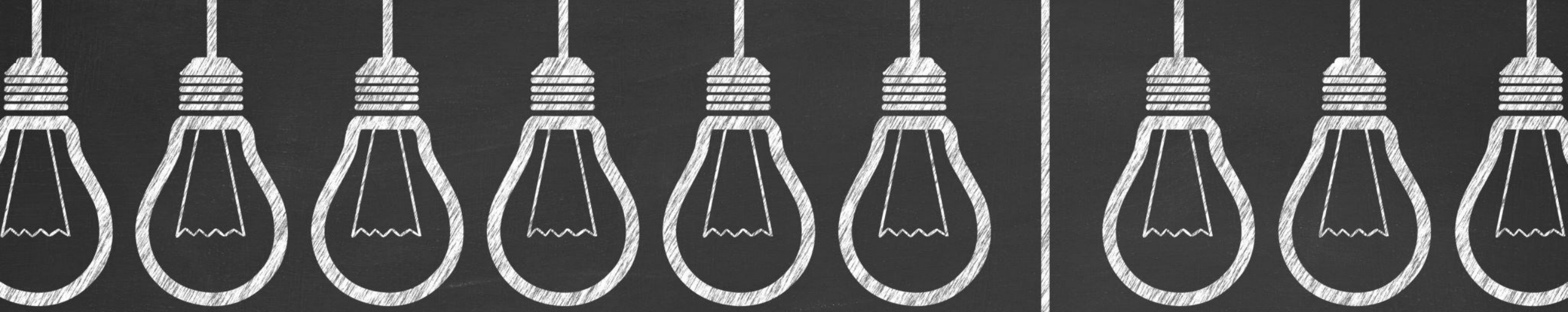
= Comportement du système  
→ Activités, flux de données

- **Modèle d'objet**

= Concepts individuels manipulés par le système et leurs propriétés  
→ Classes, composants





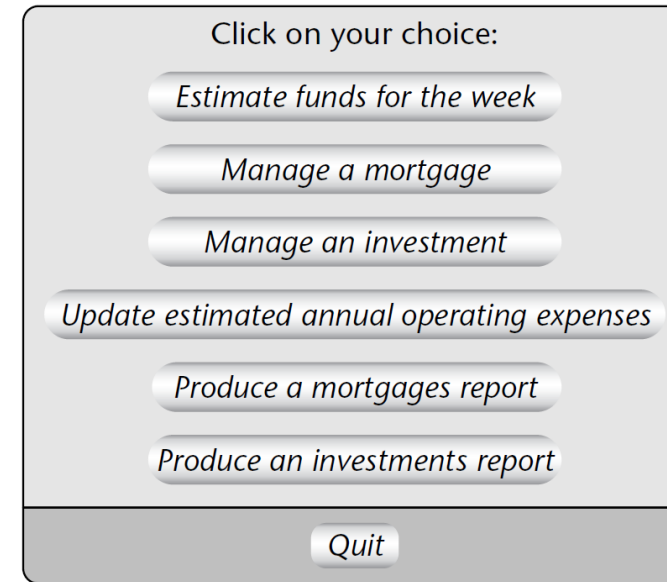


# Modèle dynamique

Comportement du système

# Maquettes pour le GUI

- Formulaires, écrans, impression
- Saisie d'information
- Composants (widgets)
- Disposition
- Présentation
- Interaction



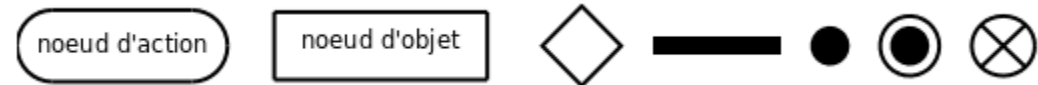
MAIN MENU  
MARTHA STOCKTON GREENGAGE FOUNDATION

1. Estimate funds available for week
2. Manage an asset
3. Update estimated annual operating expenses
4. Produce a report
5. Quit

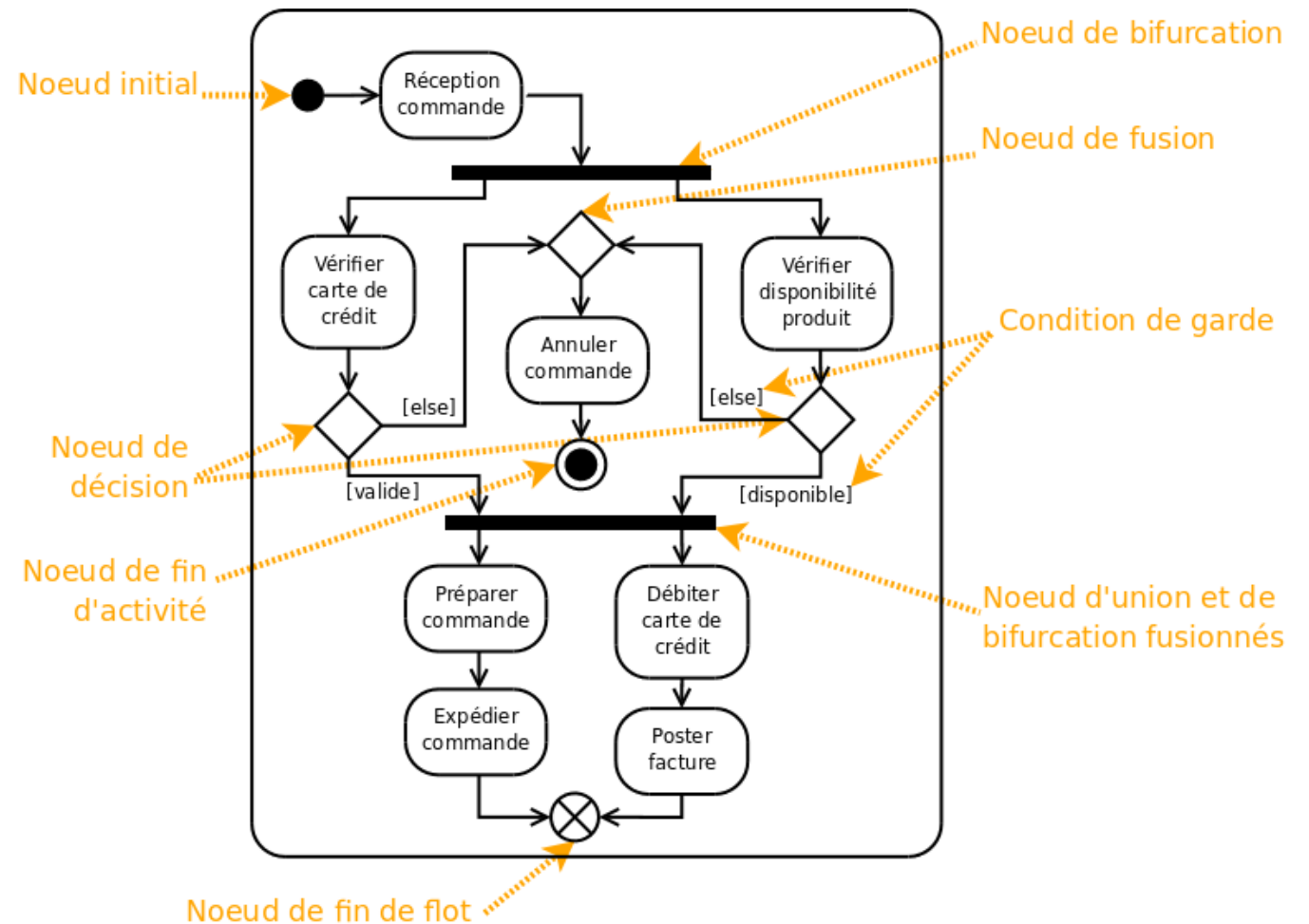
Type your choice and press <ENTER>:

# Modélisation du processus

- Identifier les activités nécessaires pour utiliser le logiciel
  - Définir les **étapes** du processus
  - Coordonner les différents **événements**
  - Identifier les responsabilités par **rôle**
- Pour cela on utilise un **diagramme d'activité UML**
  - Activités, actions
  - Transitions
  - Objets
  - Nœuds de contrôle
  - Partitions



# Diagramme d'activité UML



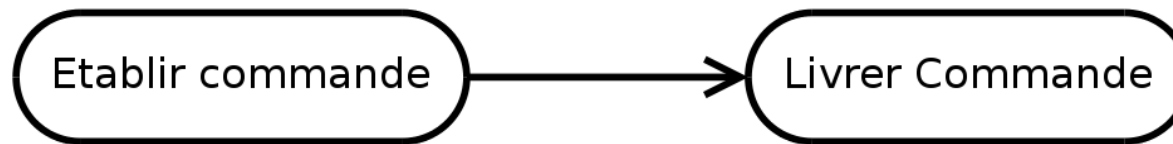
# Action

noeud d'action

- Plus petit **traitement** en UML qui a une incidence sur l'état du système
  - Affectation d'attributs
  - Accès à une propriété structurelle
  - Création d'objet
  - Calcul simple
  - Appel d'opération ou d'événement
  - Acceptation d'opération ou d'événement
  - Levée d'exception
- Activité est un comportement **complexe**
  - Terme **abstrait** représentant un séquençement d'actions

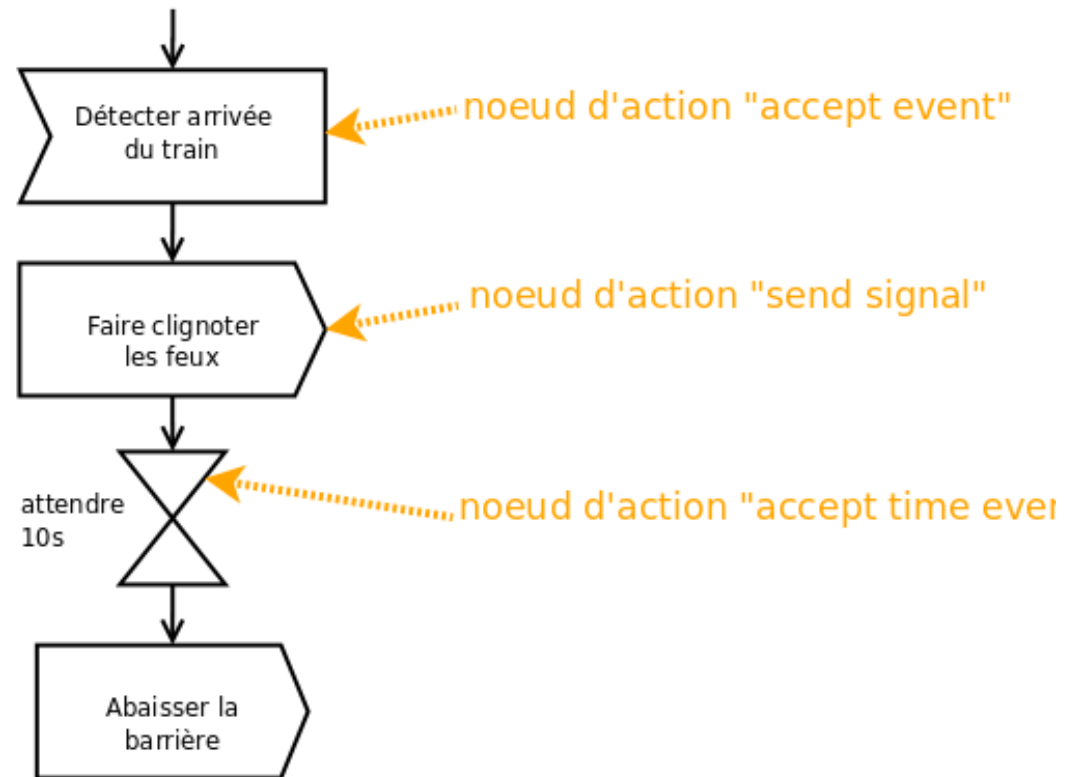
# Transition

- Passage d'une activité à une autre
  - Déclenchée quand l'activité source est terminée
  - Provoque le début de la prochaine activité cible
- Contrairement aux activités, les transitions sont **atomiques**
  - Les activités ont une durée donc peuvent être interrompues (pas les transitions !)

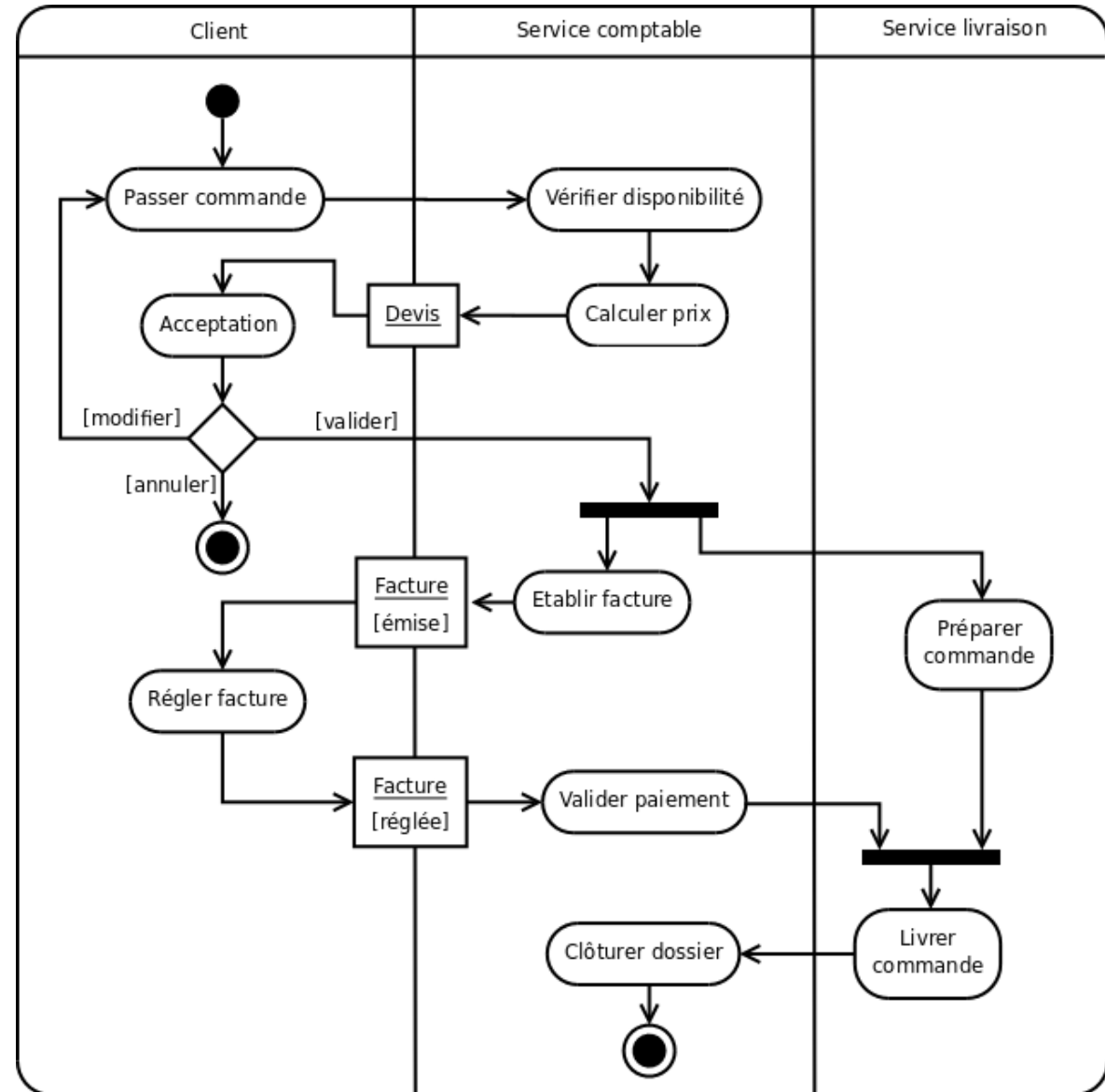


# Évènements

- Événements **externes**
  - Survient à l'extérieur du système (par un acteur)
    - Client passe une commande
- Événements **temporels**
  - Attente dans le temps
    - À chaque semaine
- Événements **d'état**
  - Survient à l'interne et déclenche un besoin de traitement
    - Rupture de stock




# Partition et roles





# Utilisation des diagrammes d'activités

- Met l'accent sur les traitements
  - Flots de **contrôle** et de **données**
- Illustre et consolide description textuelle des CU
  - Modélisation du **workflow** de chaque scenario
  - Concentre sur les activités **vues par les acteurs**



Centre de  
données #GYM

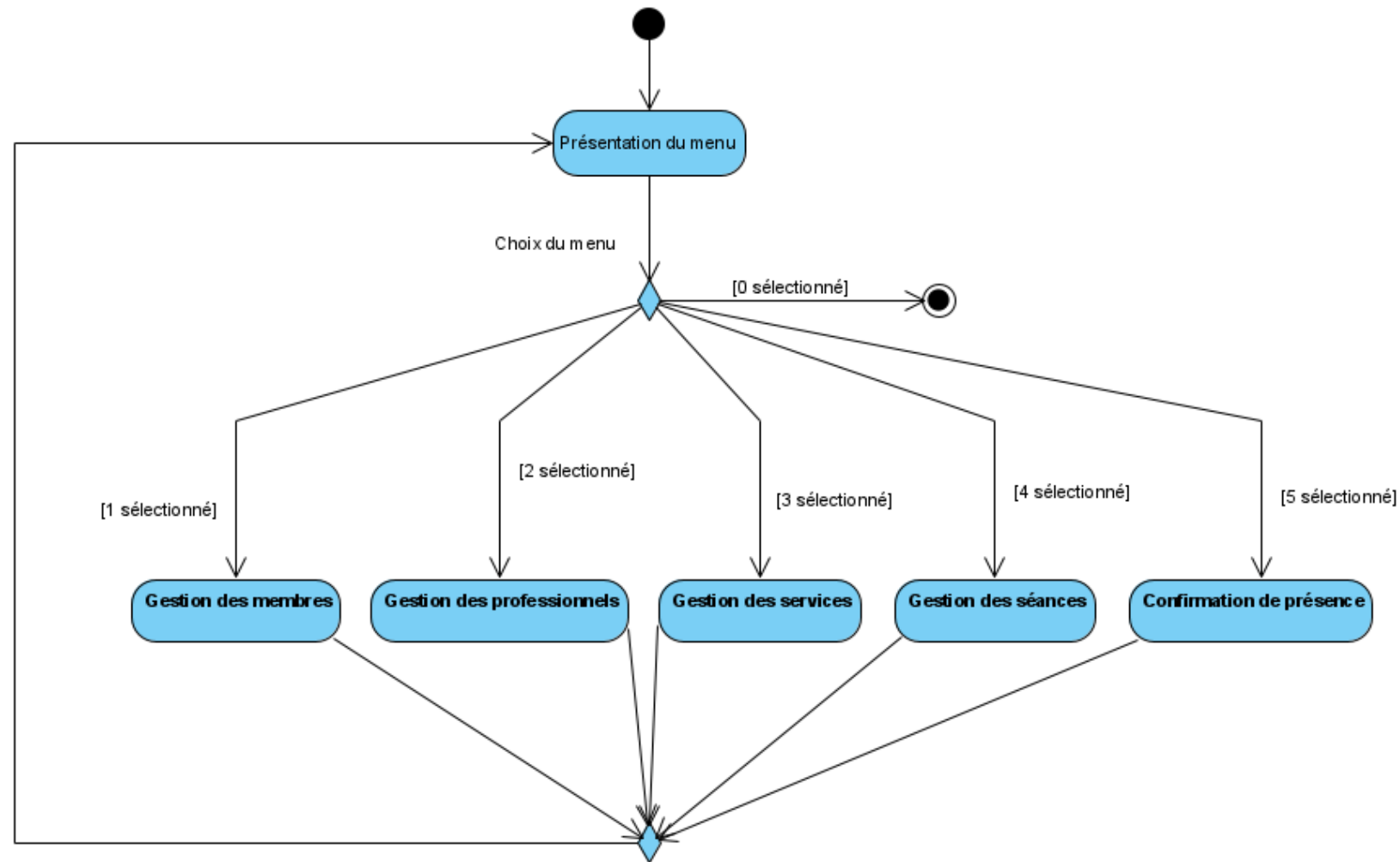
# Cas d'étude

# Fonctionnement

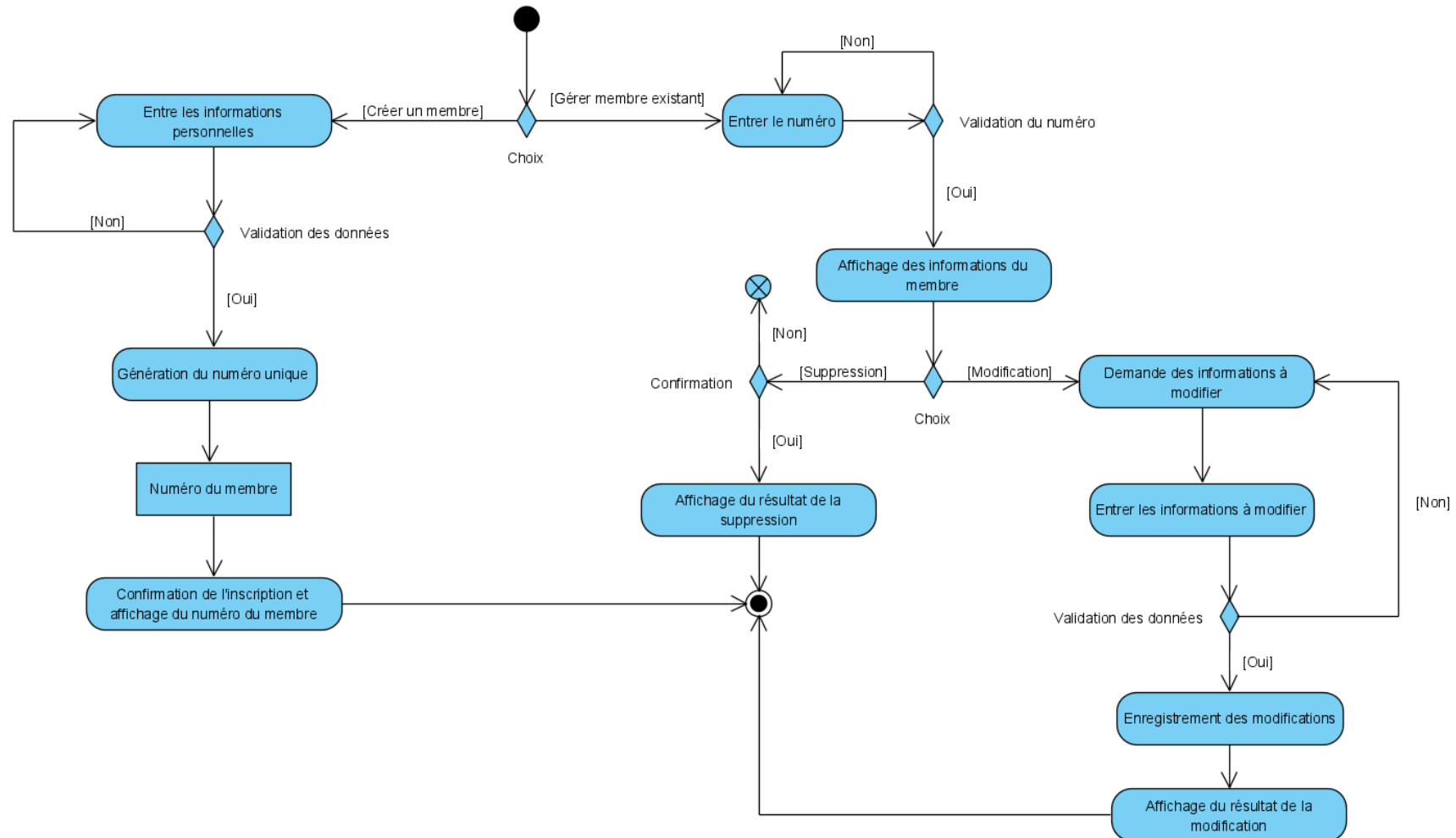
---

- #GYM est un centre sportif qui offre des services pour différentes activités physiques.
- Les clients peuvent y adhérer et devenir membres. Pour cela, le client doit se présenter à la réception du centre. L'agent lui demande ces informations personnelles, et l'enregistre dans le logiciel #GYM. Ceci crée le nouveau membre et lui assigne un numéro unique à neuf chiffres.
- Les professionnels peuvent fournir des séances de service. Pour fournir un service chez #GYM, le professionnel doit se présenter à la réception du centre. Si c'est un nouveau professionnel, l'agent lui crée un compte de façon similaire à un membre. Pour créer une séance de service, l'agent envoie l'information: numéro du professionnel, capacité maximale, date et heure (début et fin), récurrence, code de service.
- Pour accéder à #GYM, il faut présenter son numéro à l'agent. Si le numéro est valide, le mot *Validé* apparaît sur l'écran. Si le numéro est invalide, la raison est affichée, comme « Numéro invalide » ou « Membre suspendu » et l'accès est refusé.
- Pour participer à un cours ou obtenir tout autre service d'un professionnel, le membre doit s'inscrire à la séance. Grâce à l'agent, il sélectionne une séance, l'application lui demande de confirmer son inscription. Avant accéder à la séance, il devra confirmer sa présence à la réception.
- Pendant la journée, l'agent utilise le logiciel du Centre des Données #GYM pour faire la gestion des membres et des enregistrements de professionnels.

# Boucle principale



# Gestion des membres







# Modélisation des objets

# Identification des objets

- Identifier les objets (et classes) est la tâche la plus importante de la **conception orientée objet**
- Il n'y a pas de « formule magique »:
  - Besoin de bonnes **techniques**
  - Besoin de **connaissance** du domaine d'application
  - Besoin d'**expérience** à concevoir des systèmes
- Processus **itératif**

# Approches

## Basée sur les scénarios

- Identifier les objets, leurs attributs et méthodes par scénario

## Approche grammaticale

- Substantifs et verbes

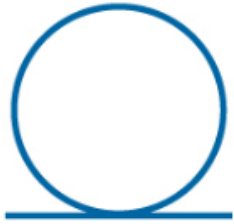
## Baser l'identification sur les choses tangibles du domaine

- Analyse du domaine
- Structures de données qui leur sont appropriées

## Approche comportementale

- Identifier les objets selon ce qui participe à chaque comportement du système





**Entity Class**



**Boundary Class**



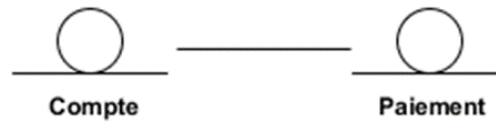
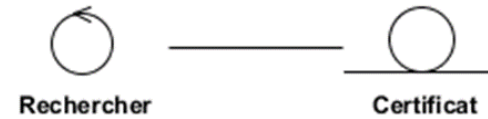
**Control Class**

# Extraction des classes

- Identifier quels concepts seront présents dans le design
- Classes d'**entité**
  - Concepts et information qui vit et persiste dans le logiciel
- Classes d'**interface**
  - Interactions entre le système et l'environnement/acteurs
- Classes de **contrôle**
  - Calculs et algorithmes complexes

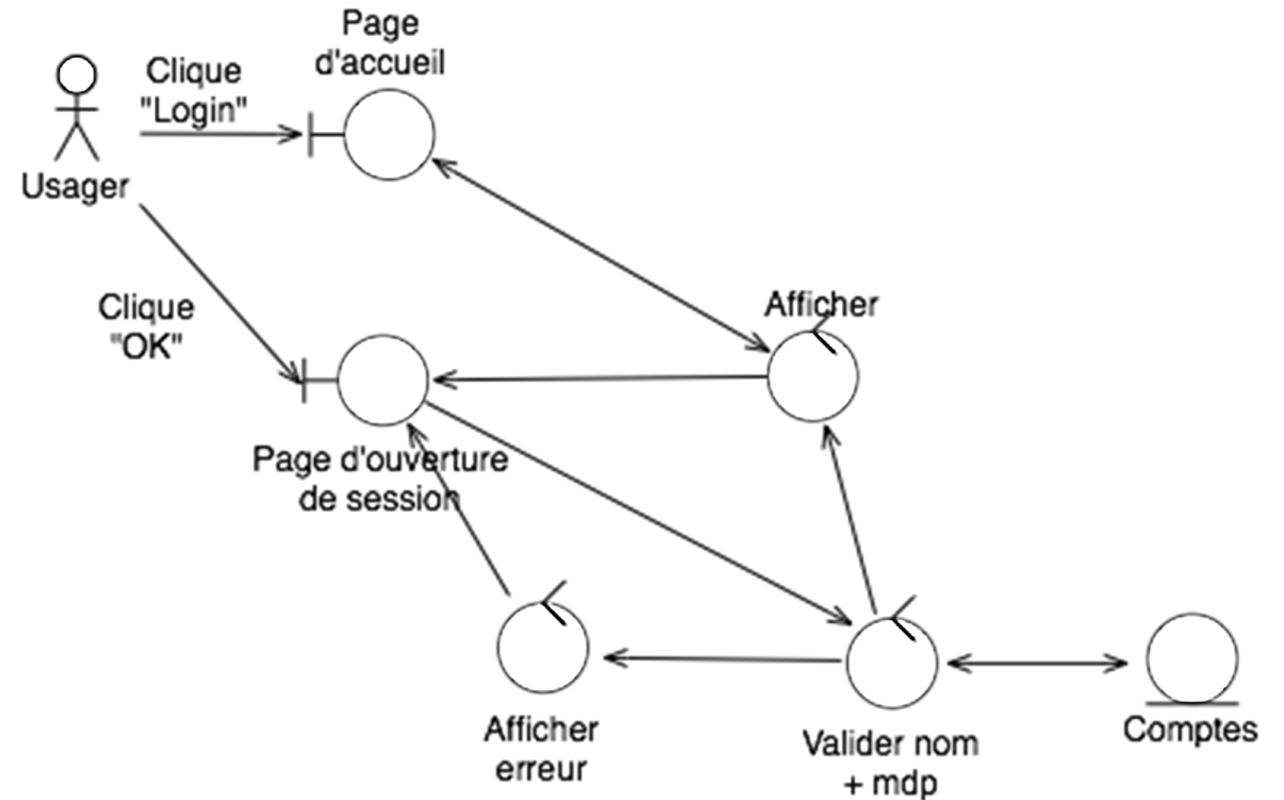
Utiliser un **diagramme de classes participantes UML** pour représenter un premier modèle conceptuel du système

# Association entre classes



# Exemple

- Classes d'**entité**
  - Comptes
- Classes d'**interface**
  - Page d'accueil
  - Page d'ouverture...
- Classes de **contrôle**
  - Afficher
  - Afficher erreur
  - Validation nom+mdp



# Modélisation des entités

Une classe entité est  
une classe métier qui  
provient du domaine

Persiste lors de  
l'exécution d'un CU et  
permet de stocker les  
données pertinentes

## Méthode d'extraction

- Étape 1: Décrire l'information du système en **un seul paragraphe**
- Étape 2: **Identifier les noms** dans ce paragraphe

# Quels noms représentent des objets?

## Pour justifier l'inclusion

- Le système doit-il **mémoriser** plus d'un élément de ce type ?
- S'agit-il d'un élément **unique** que le système doit connaître ?
- Entre-t-il dans le **cadre** de la portée du système ?
- Est-ce un élément qui constitue un **attribut** d'un autre élément ?

## Pour justifier l'exclusion

- Cet élément est-il le **synonyme** d'un autre déjà identifié ?
- Est-ce seulement une **sortie**
  - produite par le système à partir d'autres informations déjà identifiées ?
  - ayant pour effet d'enregistrer d'autres informations déjà identifiées ?

# Centre de données #GYM

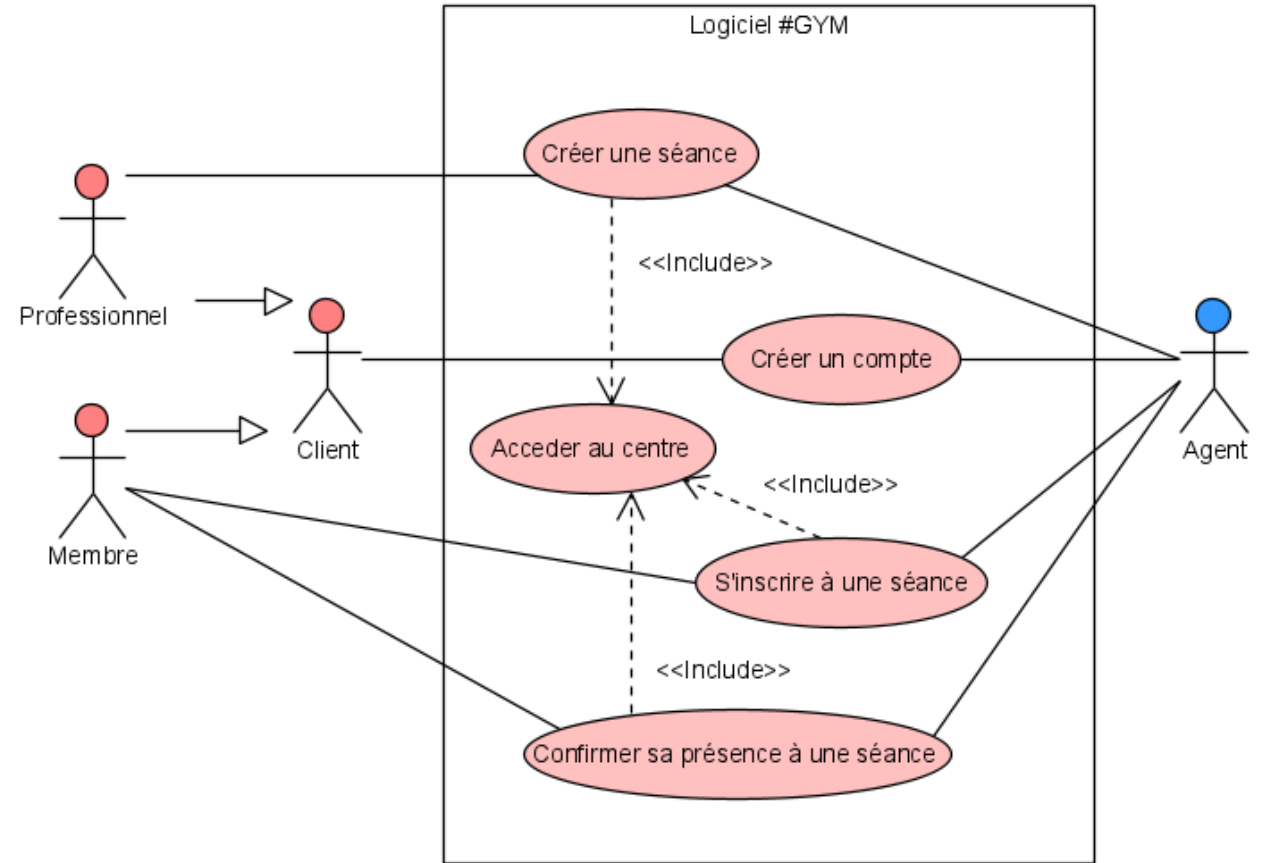
- #GYM est un centre sportif qui offre des services pour différentes activités physiques.
- Les clients peuvent y adhérer et devenir membres. Pour cela, le client doit se présenter à la réception du centre. L'agent lui demande ces informations personnelles, et l'enregistre dans le logiciel #GYM. Ceci crée le nouveau membre et lui assigne un numéro unique à neuf chiffres.
- Les professionnels peuvent fournir des séances de service. Pour fournir un service chez #GYM, le professionnel doit se présenter à la réception du centre. Si c'est un nouveau professionnel, l'agent lui crée un compte de façon similaire à un membre. Pour créer une séance de service, l'agent envoie l'information: numéro du professionnel, capacité maximale, date et heure (début et fin), récurrence, code de service.
- Pour accéder à #GYM, il faut présenter son numéro à l'agent. Si le numéro est valide, le mot *Validé* apparaît sur l'écran. Si le numéro est invalide, la raison est affichée, comme « Numéro invalide » ou « Membre suspendu » et l'accès est refusé.
- Pour participer à un cours ou obtenir tout autre service d'un professionnel, le membre doit s'inscrire à la séance. Grâce à l'agent, il sélectionne une séance, l'application lui demande de confirmer son inscription. Avant accéder à la séance, il devra confirmer sa présence à la réception.
- Pendant la journée, l'agent utilise le logiciel du Centre des Données #GYM pour faire la gestion des membres et des enregistrements de professionnels.

# Centre de données #GYM

Entités (jaune)  
Attributs (vert)

- #GYM est un centre sportif qui offre des services pour différentes activités physiques.
- Les clients peuvent y adhérer et devenir membres. Pour cela, le client doit se présenter à la réception du centre. L'agent lui demande ces informations personnelles, et l'enregistre dans le logiciel #GYM. Ceci crée le nouveau membre et lui assigne un numéro unique à neuf chiffres.
- Les professionnels peuvent fournir des séances de service. Pour fournir un service chez #GYM, le professionnel doit se présenter à la réception du centre. Si c'est un nouveau professionnel, l'agent lui crée un compte de façon similaire à un membre. Pour créer une séance de service, l'agent envoie l'information: numéro du professionnel, capacité maximale, date et heure (début et fin), récurrence, code de service.
- Pour accéder à #GYM, il faut présenter son numéro à l'agent. Si le numéro est valide, le mot Validé apparaît sur l'écran. Si le numéro est invalide, la raison est affichée, comme « Numéro invalide » ou « Membre suspendu » et l'accès est refusé.
- Pour participer à un cours ou obtenir tout autre service d'un professionnel, le membre doit s'inscrire à la séance. Grâce à l'agent, il sélectionne une séance, l'application lui demande de confirmer son inscription. Avant accéder à la séance, il devra confirmer sa présence à la réception.
- Pendant la journée, l'agent utilise le logiciel du Centre des Données #GYM pour faire la gestion des membres et des enregistrements de professionnels.

# Cas d'utilisation





# Classe d'interface

- Facile à identifier
- Issue directement de la maquette du GUI
  - Input/Output
  - Interfaces systèmes
  - Interaction entre interface (ex: GUI) et les utilisateurs
  - Écran de saisie, écran de sortie, rapports imprimés
- Au moins une interface pour chaque **association** entre un acteur et un CU
- En général, les interfaces vivent seulement le temps du déroulement du CU concerné

# Classe d'interface d'entrée

- Un seul écran devrait être suffisant pour les 5 CU de GYM
    - Créer une séance
    - Créer un compte
    - Accéder au centre
    - S'inscrire à une séance
    - Confirmer sa présence
- ⇒ Donc une seule classe d'interface pour la saisie d'entrées est suffisante

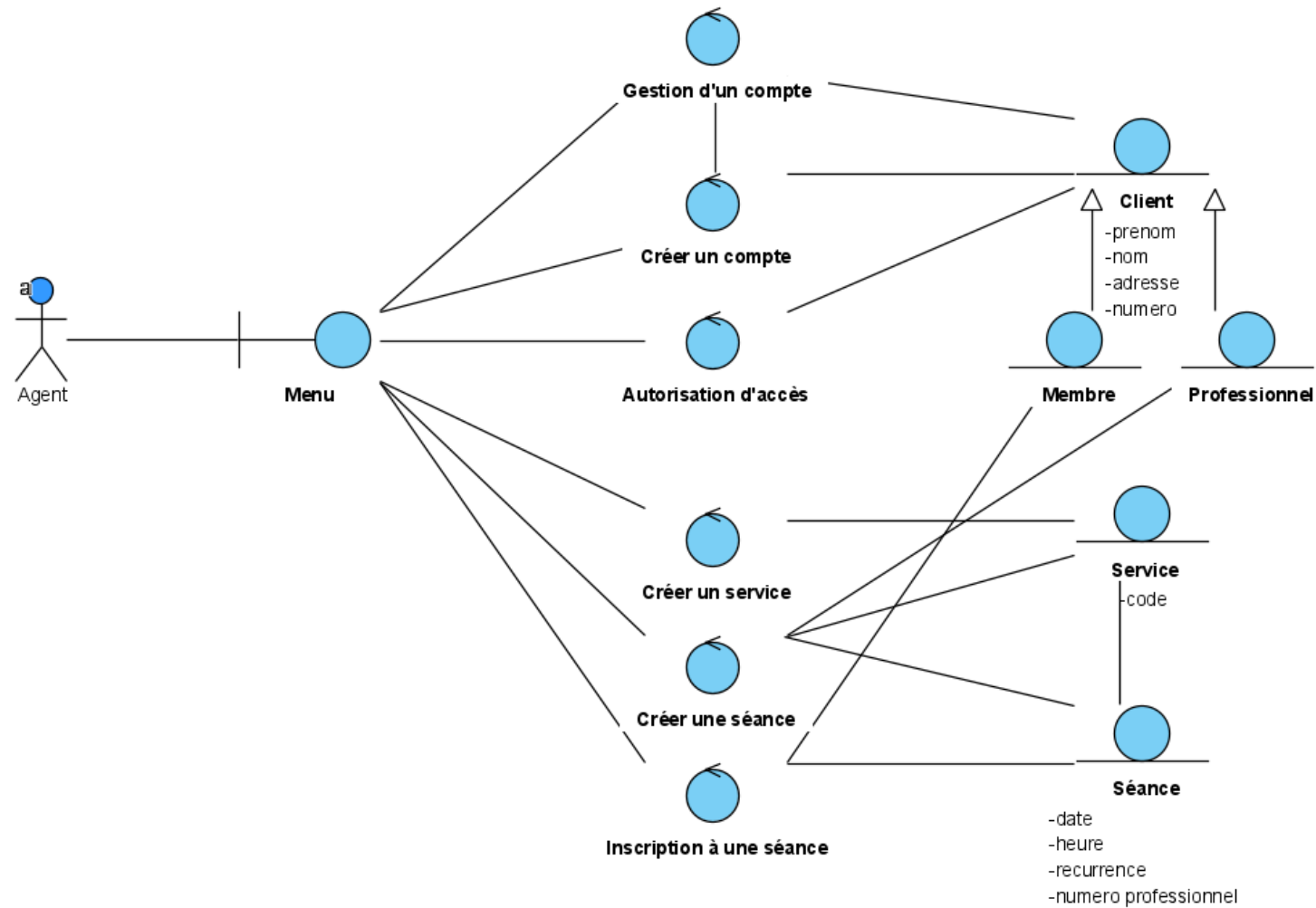
# Classe d'interface de sortie

- Aucun rapport ou exportation de données n'est requis à ce stade  
⇒ Donc aucune classe d'interface n'est nécessaire pour la sortie
- En cas de production de rapports, si leur contenu est différent  
⇒ Chacun est modélisé par une classe d'interface distincte

# Classe de contrôle

- Modélise le comportement de l'application
  - Réalise et accomplit un CU
- Fait la jonction entre interfaces et entités
- Contient les règles applicatives et les isolent des interfaces et des entités
- Pour GYM, nous en avons identifié 6 jusqu'à présent.

# Modèle d'objet (itération 1)

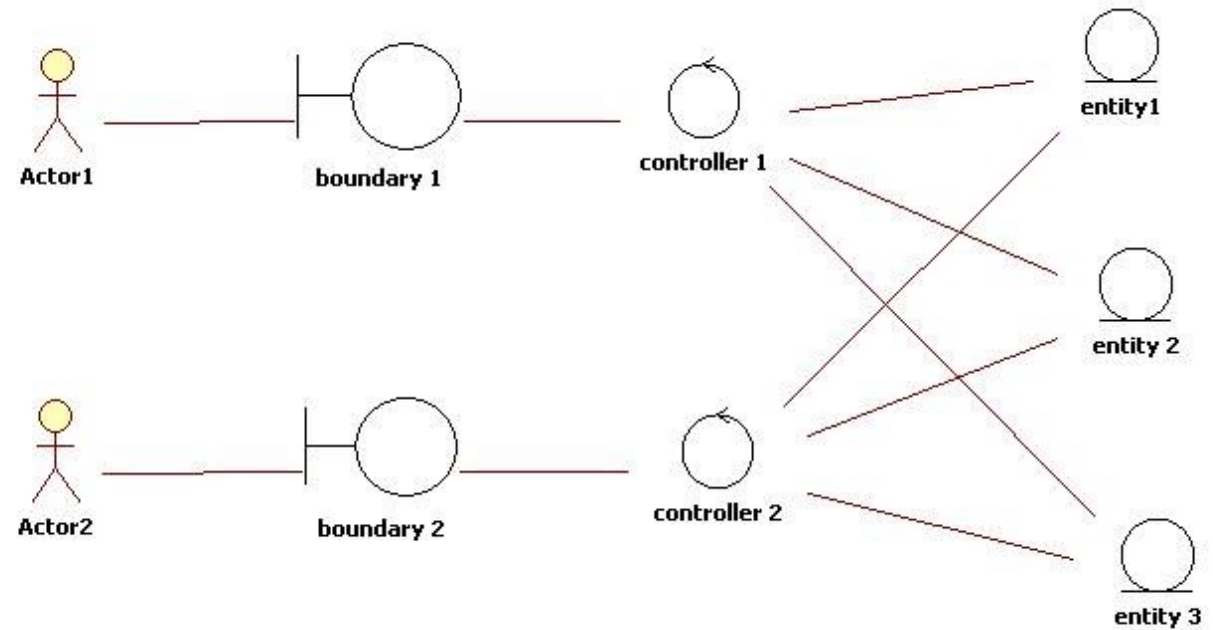


# Directives pour modèle d'objets

Les bonnes pratiques 😊

# Patron Entité-Contrôle-Interface

- Les acteurs n'interagissent qu'avec des classes d'interfaces
- Les entités représentent les données du système
- Les contrôles sont les médiateurs entre interfaces et entités



# Règles pratiques à suivre

- **Entités** issues du modèle du domaine ne comportent **que des attributs**
- **Contrôles ne comportent que des opérations**
  - Chaque contrôle est généralement associé à un CU et vice versa
  - Peut décomposer un CU complexe en plusieurs contrôles
- **Contrôles peuvent être associés à tous les types de classes**
  - Contrôle vers interface, vers entité ou vers autre contrôle (et l'inverse)
- **Interfaces peuvent contenir des attributs et opérations**
  - Attributs représentants des informations ou paramètres saisis par l'utilisateur ou des résultats d'actions
  - Opérations réalisent les actions que l'utilisateur demande, généralement par délégation aux contrôles