

# Gestion et Traitement de QCM

Yann CHARBONNIER   Christian INGOUFF  
Pierre-Alexandre TYNDAL   Sonia SEDDIKI

EISTI, ING1

10 juin 2014

# Sommaire

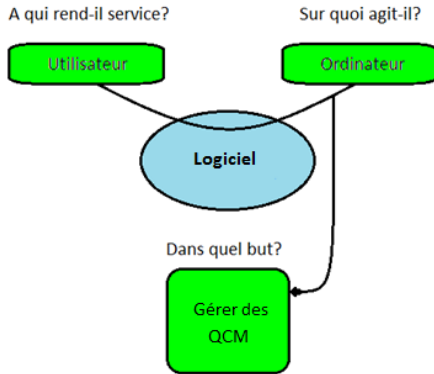
- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes

# Sommaire

- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes

# Sommaire

- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes



## Besoins fonctionnels

- Définir la hiérarchie des droits des utilisateurs
- Mémoriser les QCM entrés
- Mémoriser les réponses données aux différentes sessions

## Contraintes techniques

- Langage de développement : Java version 1.7
- Analyse et conception en UML
- Utilisation du pattern MVC
- Ergonomie

## Gestion de projet

- Un planning
- Autonome



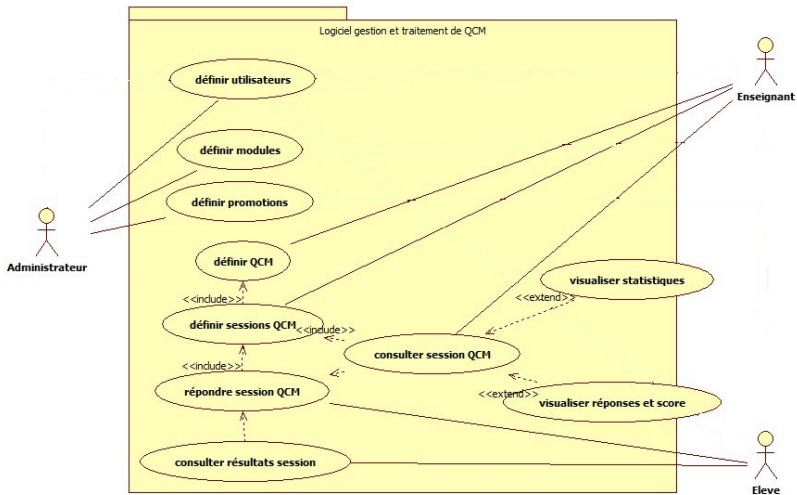
# Sommaire

- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes

# Intérêt

## Diagramme de cas d'utilisation

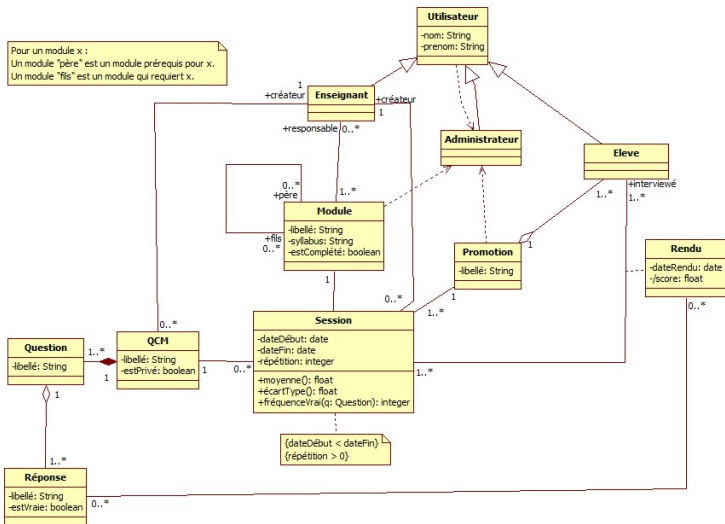
- Distinguer les utilisateurs
- Etablir des opérations
- Etablir des relations



# Intérêt

## Diagramme de classes

- Catégoriser des entités en classes
- Mettre en relation les classes créées



# Object Constraint Language : OCL

- Consultation de session

```
context Session
inv: self.interviewé->count(r : Rendu |
    r.session = self) <= self.répétition
```

- Date de rendu

```
context Rendu
inv: (self.dateRendu > self.session.dateDébut)
    AND (self.dateRendu < self.session.dateFin)
```

- Description plus précise du modèle
- Base pour les étapes suivantes
- À venir : analyse fonctionnelle