

# Gestion et Traitement de QCM

Yann CHARBONNIER   Christian INGOUFF  
Pierre-Alexandre TYNDAL   Sonia SEDDIKI

EISTI, ING1

6 juin 2014

# Sommaire

- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes
  - Diagramme d'activités
  - Diagramme de séquences
  - Diagramme d'états-transitions
- 3 Conception
- 4 Programmation

# Sommaire

- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes
  - Diagramme d'activités
  - Diagramme de séquences
  - Diagramme d'états-transitions
- 3 Conception
- 4 Programmation

# Sommaire

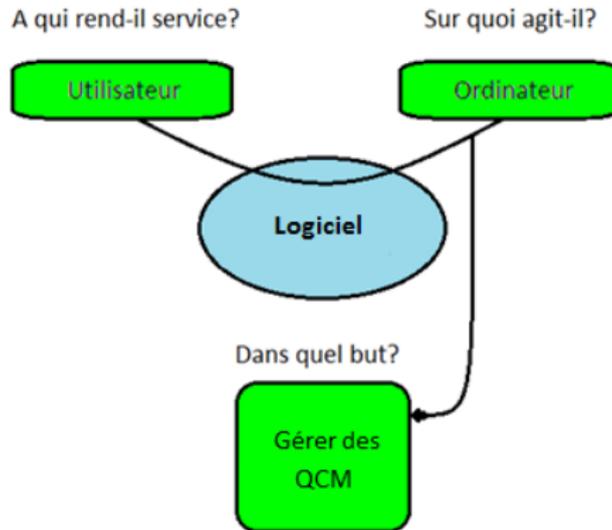
- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes
  - Diagramme d'activités
  - Diagramme de séquences
  - Diagramme d'états-transitions
- 3 Conception
- 4 Programmation

# Sommaire

- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes
  - Diagramme d'activités
  - Diagramme de séquences
  - Diagramme d'états-transitions
- 3 Conception
- 4 Programmation

# Sommaire

- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes
  - Diagramme d'activités
  - Diagramme de séquences
  - Diagramme d'états-transitions
- 3 Conception
- 4 Programmation



## Besoins fonctionnels

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

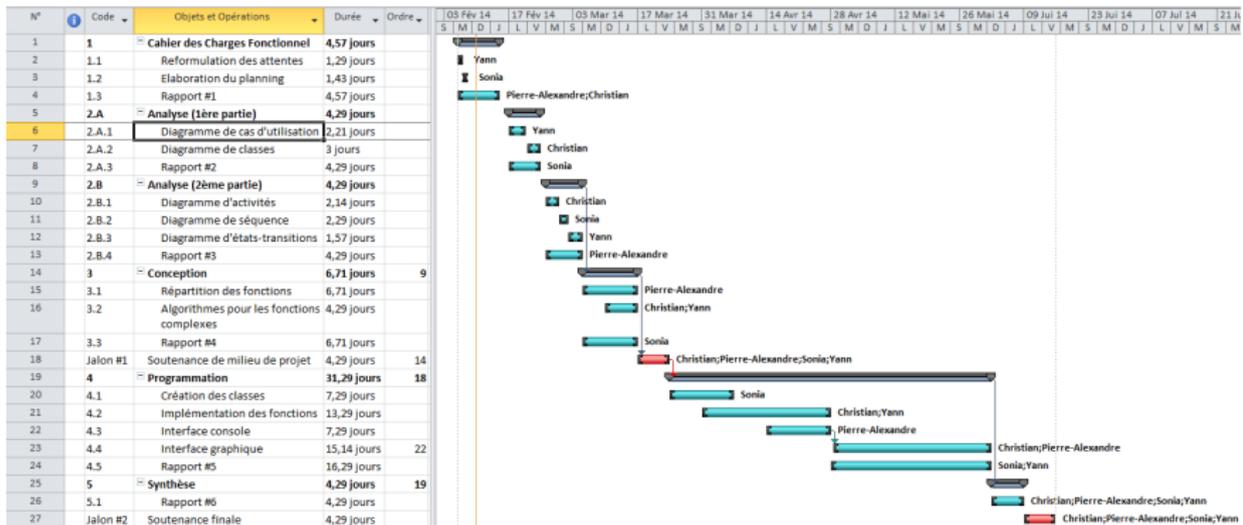
## Exigences techniques

- Langage de développement : Java version 1.7
- Utilisation du pattern MVC
- Analyse et conception en Unified Modeling Language (UML)

## Gestion de projet

- Un planning
- Autonome

# Diagramme de Gantt



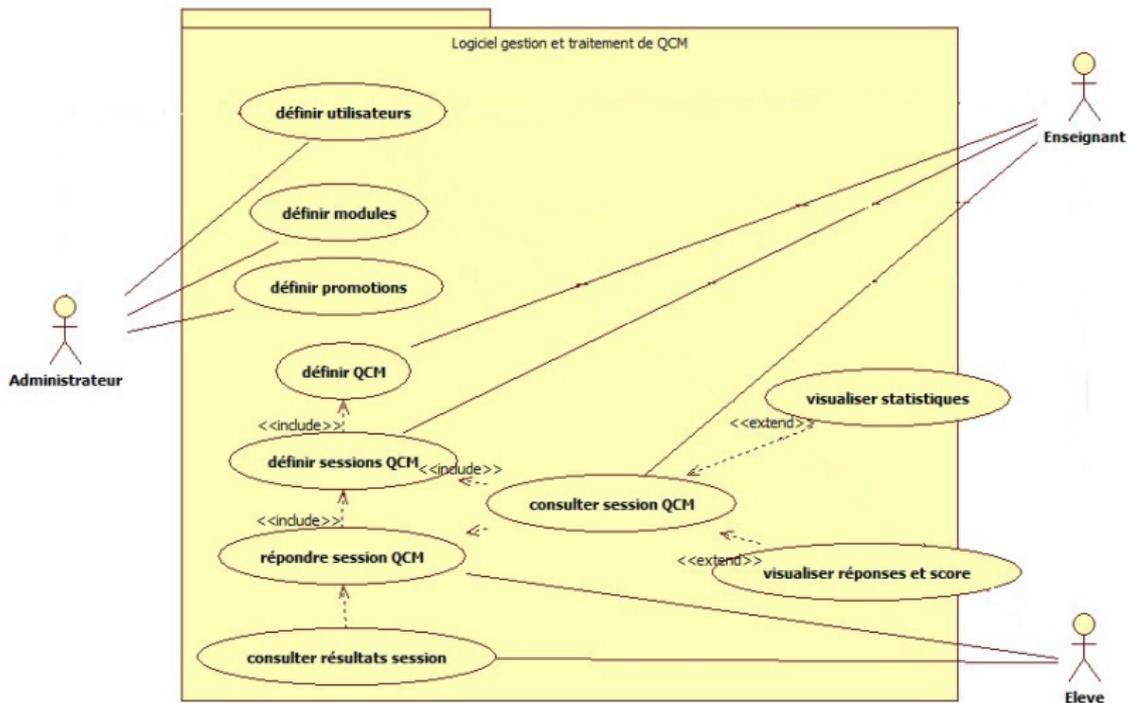
# Sommaire

- 1 Expression des besoins
- 2 **Analyse des besoins**
  - Diagramme de cas d'utilisation
  - Diagramme de classes
  - Diagramme d'activités
  - Diagramme de séquences
  - Diagramme d'états-transitions
- 3 Conception
- 4 Programmation

# Intérêt

## Diagramme de cas d'utilisation

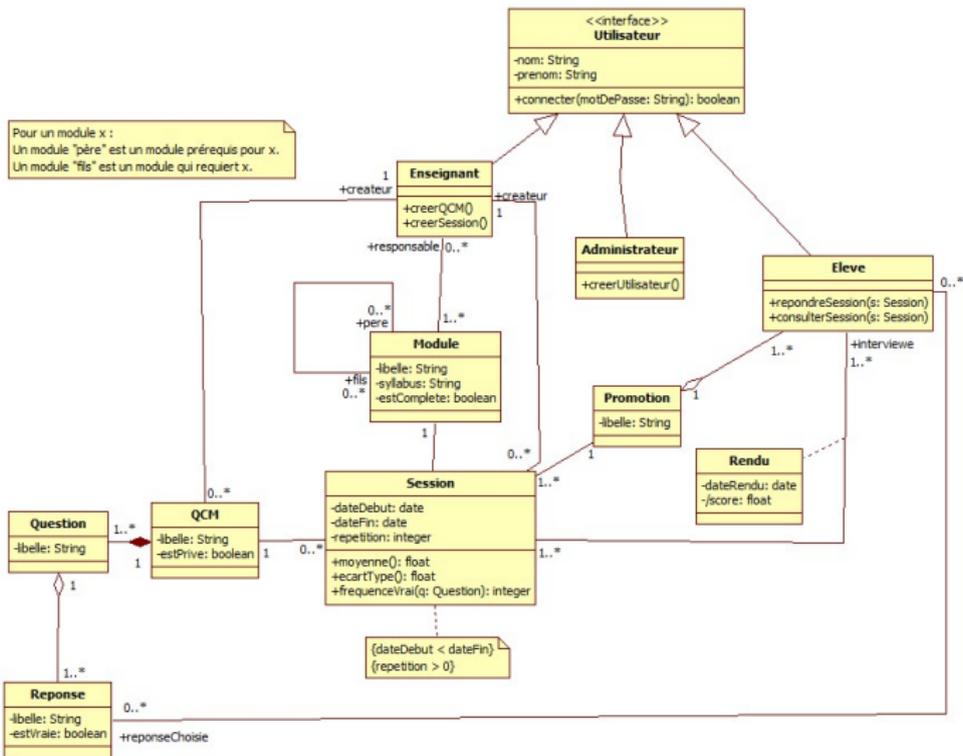
- Distinguer les utilisateurs
- Etablir des opérations
- Etablir des relations



# 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

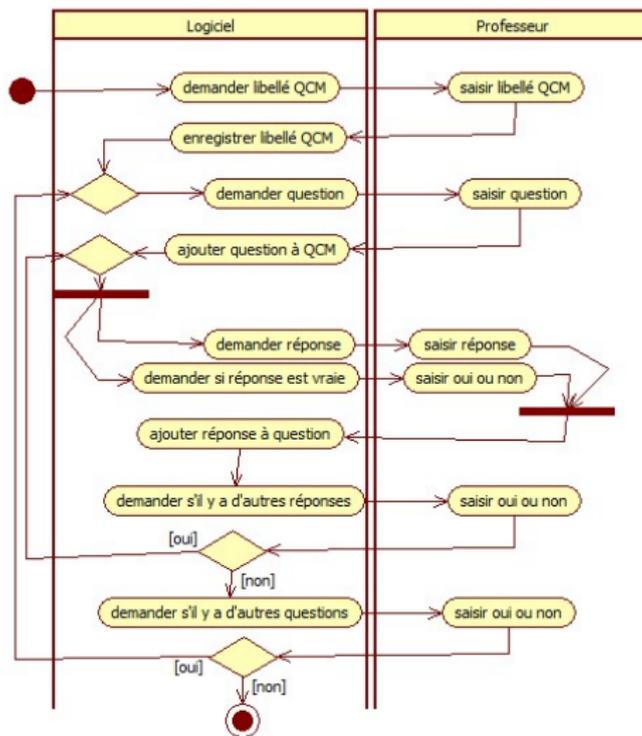
```
context Rendu
```

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

# Intérêt

## Diagramme d'activités

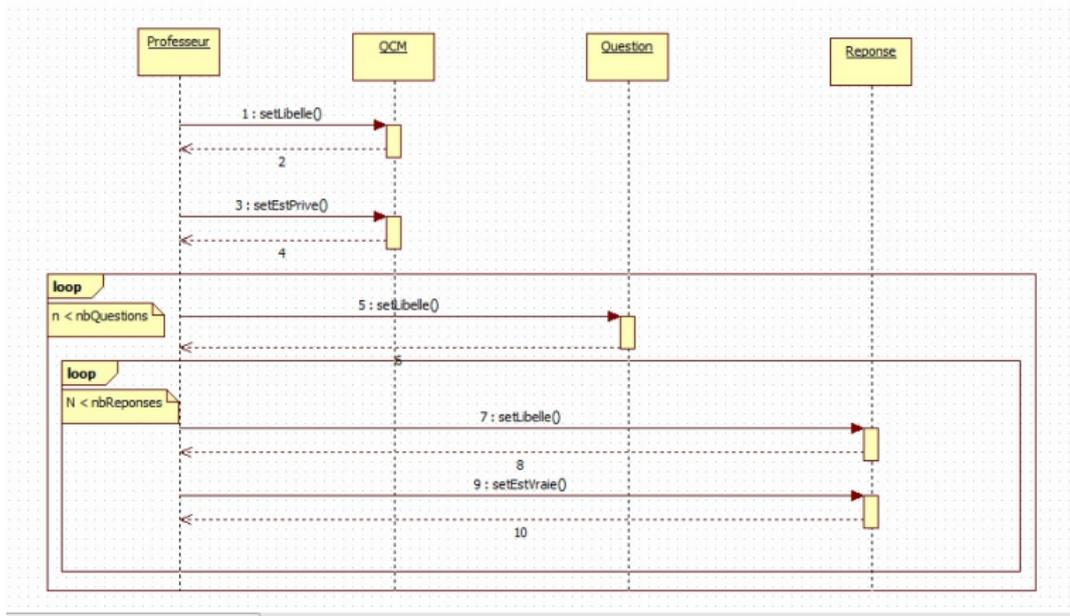
- Détailler le déroulement d'un cas d'utilisation
- Codifié, mais pas totalement : première approche



# Intérêt

## Diagramme de séquences

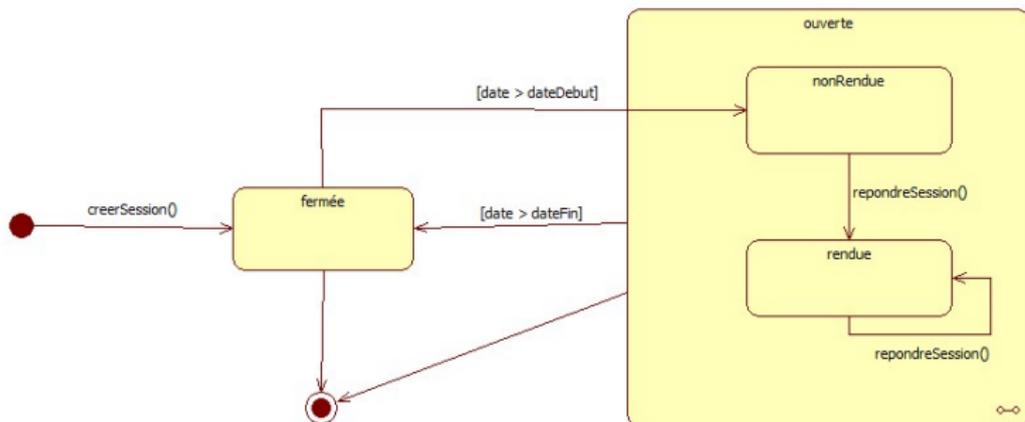
- Plus précis que le diagramme d'activités
- Décrit un cas d'utilisation en précisant :
  - Les classes concernées
  - Leurs interactions mutuelles (signaux)
  - La manière dont elles interagissent (méthodes)



# Intérêt

## Diagramme d'états-transitions

- Décrit les changements d'états d'un objet
- Décrit les transitions entre deux états



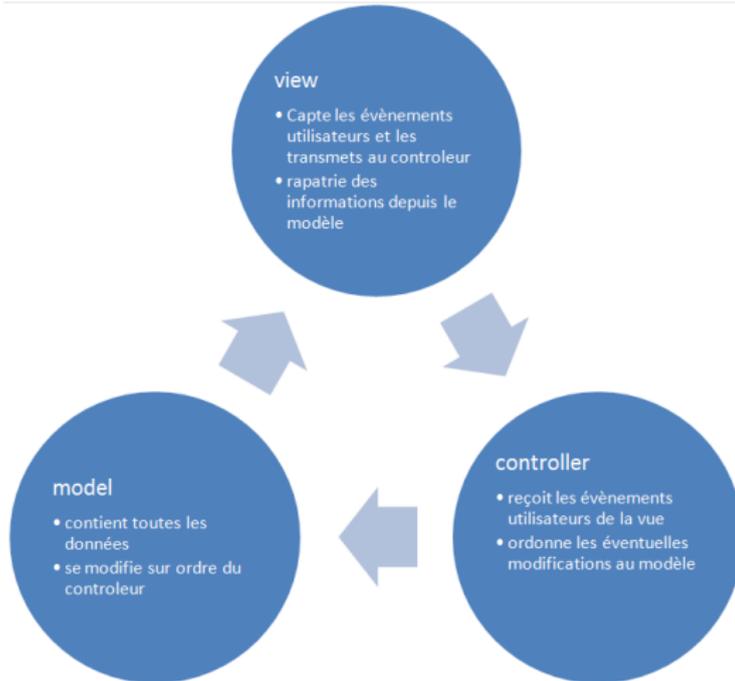
# Sommaire

- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes
  - Diagramme d'activités
  - Diagramme de séquences
  - Diagramme d'états-transitions
- 3 Conception**
- 4 Programmation

# Pattern MVC

## Principe

- MVC : Model, Controller, View
- Séparation du code en trois packages
- Mieux structuré, plus lisible



# Sommaire

- 1 Expression des besoins
- 2 Analyse des besoins
  - Diagramme de cas d'utilisation
  - Diagramme de classes
  - Diagramme d'activités
  - Diagramme de séquences
  - Diagramme d'états-transitions
- 3 Conception
- 4 Programmation

# Interface administrateur

```
Main [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (6 juin 2014 12:17:58)
```

```
azerty
```

```
Interface administrateur :
```

```
Choisissez votre tâche :
```

- 0. Quitter le programme
- 1. Gestion des utilisateurs
- 2. Gestion des modules
- 3. Gestion des promotions

```
1
```

```
Bienvenue à la gestion des utilisateurs !
```

```
Choisissez votre tâche :
```

- 0. Menu précédent
- 1. Création d'un utilisateur
- 2. Modification d'utilisateurs
- 3. Suppression d'utilisateurs

# Interface enseignant

Main [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (6 juin 2014 12:20:04)

azerty

Interface enseignant :

Choisissez votre tâche :

- 0. Quitter le programme
- 1. Gestion des QCM
- 2. Gestion des sessions
- 3. Consultation des sessions en cours

1

Bienvenue à la gestion des QCM !

Choisissez votre tâche :

- 0. Menu précédent
- 1. Création d'un QCM
- 2. Modification de QCM
- 3. Suppression de QCM

# Interface élève

```
Main [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (6 juin 2014 12:23:34)
Interface élève :
Choisissez votre tâche :
0. Quitter le programme
Sessions en cours :
1. QCM 'Le bal des Eistiens' ('Grec' pour CPI1) du 01-02-2013 au 01-01-2015

Sessions finies :
2. QCM 'Le bal des Eistiens' ('Grec' pour CPI1) du 01-01-2012 au 01-01-2014
2
Consultation de résultats :
Session du QCM 'Le bal des Eistiens' ('Grec' pour CPI1) du 01-01-2012 au 01-01-2014
Votre note est de 15.0 sur 20.
Retour au menu principal...
Choisissez votre tâche :
0. Quitter le programme
Sessions en cours :
<
```

# Travail en équipe

## Structure de l'équipe

- Rôles
  - Christian : Coordinateur, Organisateur, Concepteur
  - Pierre-Alexandre : Concepteur
  - Yann : Concepteur
  - Sonia : Propulseur, Perfectionneur
- Evolution de l'équipe
  - Formation cooptative
  - Stabilisation
  - Développement

# Conclusion

## Bilan du projet

- Rendu final
- Gestion du projet