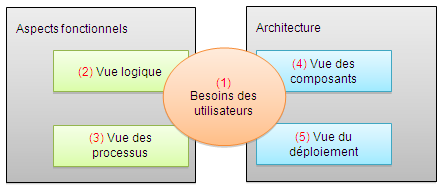
http://openclassrooms.com/courses/debutez-l-analyse-logicielle-avec-uml/les-differents-types-de-diagrammes

Quant à UML modélise, la conception d’un logiciel est organisée autour des aspects fonctionnels et d’architecture. Ces deux aspects sont représentés par le schéma de 4 vues, à partir des besoins des utilisateurs (parfois intitulé des cas d’utilisation), appelé 4+1 vues.

* Les aspects fonctionnels :il décrit l’utilisateur et la fonction d’application. Le déroulement du logiciel. Les informations utilisées pour cela.
* Les aspects liés à l’architecture : les différents composants logiciels à utiliser (base de données, librairies, interfaces, etc.)

# Les 4+1 vues d’un système

Une première décomposition d’une problématique ou système peut être faite à l’aide de 4+1 vues. Le schéma ci-dessous montre les différentes vues permettant de répondre aux besoins des utilisateurs, organisées selon les deux aspects (fonctionnels et architecture). Chacune des vues est constituée de diagrammes.

## Les besoins des utilisateurs

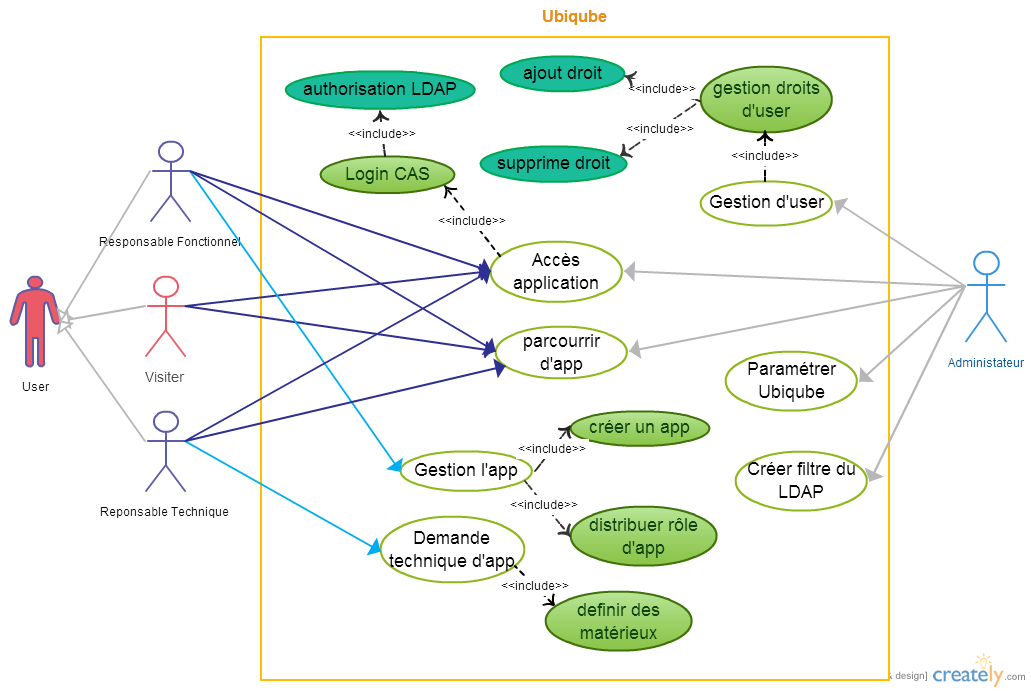
Cette partie représente le cœur de l’analyse-les spécifications système. On y décrit le contexte, les acteurs ou utilisateurs de l’application, les fonctionnalités du logiciel mais aussi les interactions entre ces acteurs et ces fonctionnalités.

### Le diagramme de packages

On décompose le système en catégories/parties plus facilement observables, appelés « packages ». Cela permet d’indiquer les acteurs qui interviennent dans chacun des packages.

### Le diagramme de cas d’utilisation

Un cas d’utilisation est défini comme un ensemble de scénarios d’utilisation, chaque scénario représentant une séquence d’interaction des utilisateurs (acteurs) avec le système.



## Vue logique

La vue logique constitue la principale description architecturale d’un système informatique. Elle permet d’identifier les différents éléments et mécanismes du système à réaliser. Cette vue de façon statique et dynamique, le système en termes d’objets et de classes.

### Le diagramme de classes

Utiliser xx à générer Class/structure de code source RoR

### Le diagramme d’objets

La structure de base de données d’application

C:\Users\jchen\Downloads\Database ubiqube.tiff

## Vue des processus

### Le diagramme de séquence

#### P:\Jing\Authentication&CAS.pngCAS Authentication

#### Gestion des droits d’utilisateur

### Le diagramme d’activités

## Vue des composants

Cette vue permet de visualiser l’organisation des composants (bibliothèque dynamique et statique, code source..) dans l’environnement de développement.

### Le diagramme de composants

### Sitemap

## Vue du déploiement

Elle représente le système dans son environnement d’exécution. Elle traite des contraintes géographiques (distribution des processeurs dans l’espace),

### Le diagramme de déploiement