***UML***

En UML le client est appelé le Product Owner (PO). En analyse on répond à la question du ‘***Quoi ?***’ et dans la conception nous sommes dans le ‘***Comment ?***’.

**Analyse et conception**

**Analyse**: Ensemble de procédures logiques qui permettent d’aboutir à une solution/résultat logique ;

**Approche** : Ensemble de méthodes qui permettent de résoudre un problème.

MERISE : méthode la plus utilisée en approche de décomposition fonctionnelle.

**Approche Orientée Objet**

UML : langage de modélisation unifié

Les principes de UML :

* Principe itératif et incrémental : aller étape par étape
* Guidé par le besoin client et des utilisateurs : s’adapter aux besoins des utilisateurs
* Centré sur l’architecture logicielle : des normes et règles à respecter
* Qui décrit les actions et les informations dans une seule entité appelée classe

**Classe** : Ensemble d’objets ayant la même structure d’informations et effectuant les mêmes actions dans le système.

Quand on modélise sur UML, il faut réaliser le système sur 2 aspects :

|  |  |
| --- | --- |
| **Aspect fonctionnel** | **Aspect architecture** |
| Vue Processus  Vue logique(donnés) | Vue de déploiement  Vue des composants |

**Vue des besoins** :

* **Recueillir le besoin** : 1- identifier le problème 2- identifier le domaine de gestion 3- identifier le contexte de l’étude (domaine gestion et ses acteurs) 4- identifier les fonctionnalités(services) 5- proposer des formalismes de modélisation(diagrammes).

***Domaine de gestion(système)***: domaine dans lequel sera utilisée la solution digitale.

***Acteurs***: toute entité ou tout collaborateur pouvant utiliser la solution. Leur intérêt est une cartographie de l’ensemble du système

***Utilisateur***: tout acteur qui a accès au système et utilise les fonctionnalités du système. Ils sont les **acteurs principaux.**

***Collaborateur***: toute entité qui peut émettre/recevoir des informations aux systèmes mais n’y a pas accès. Ils sont les **acteurs secondaires**.

***Nb***: Un acteur n’est pas forcément un humain ohhhhh, il peut être un logiciel, du matériel…

La vue besoin sera représenté par 3 diagrammes :

**Diagramme de Contexte** ;

**Diagramme de Package** ;

**Diagramme de UseCase** ;

En UML on a au total 13/14 diagrammes :

Vue besoin : 3

Vue processus : 6

Vue logique : 2

Vue composant : 2

Vue déploiement : 1

**Diagramme DC**

**Diagram Acteurs**

Nom Dom Gestion

Nom Acteur

**Diagramme de Contexte**

Nom Dom Gestion

Nom Acteur

Acteurs Secondaires

Nom Acteur

Acteurs Principaux

**Nb**: Un acteur secondaire n’a pas de fonctionnalités dans le système car n’a pas accès.

**Usecase**: un service offert par le système à ses acteurs principaux.

Formalisme d’un use Case :

: Hériter

Si la relation entre le UseCase principal et le Usecase interne est obligatoire on parle de relation d’Inclusion, si facultatif donc relation d’extension.

***Extension*** : de principal à interne ; On dit qu'un cas d'utilisation A étend un cas d'utilisation B lorsque le cas d'utilisation A peut être appelé au cours de l'exécution du cas d'utilisation B. Exécuter B peut éventuellement entraîner l'exécution de A : contrairement à l'inclusion.

***Inclusion***: d’interne à principal ; Un cas A inclut un cas B si le comportement décrit par le cas A inclut le comportement du cas B : le cas A dépend de B. Lorsque A est sollicité, B l'est obligatoirement, comme une partie de A.

14/10/2022

Diagramme de contexte :

Domaine d’étude

Acteurs

Use Case = par acteur

En tant que Acteur je peux :

……………………

…………………..

………………….

Organiser Use Case 🡺 par lots = paquets

Diagramme de Package

Etude Use Case