DEFINITIONS POUR LA PRESENTATION DU PROJET PERSONNEL

**Cas d’utilisation ou Use Case**

La présentation des cas d’utilisations est la représentation graphique des fonctionnalités du site ou d’un logiciel. Cela montre tout ce que l’on peut faire sur le dit site ou logiciel et permet au développeur de structurer et de construire le code en fonction de ses fonctionnalités.

Il y a autant de cas d’utilisation que de use case.

Le use case se compose de :

Acteur : personne ayant une activité sur le logiciel ou le site et possédant des droits d’utilisations définis en fonction de son rang.

Include : L'include est une relation entre 2 cas d'utilisations, elle permet d'étendre un cas d'utilisation avec un autre cas d'utilisation de façon obligatoire.

Extend : L'extend est une relation entre 2 cas d'utilisations, elle permet d'étendre un cas d'utilisation avec un ou plusieurs autres cas d'utilisation de façon facultative.

**Diagramme d’activité ou Activity diagram**

Le diagramme d'activité sert à décomposer les différentes actions d'un cas d'utilisation sous une forme algorithmique.

Etat initial : c’est l’état dans lequel on se trouve en débutant le processus

Etat final : c’est l’état attendu à la fin du processus

**Diagramme de séquence ou Sequence diagram**

Le diagramme de séquence est un diagramme qui représente les différentes actions sous une forme chronologique, on ne représente dessus qu'un seul scénario le cas nominal.

Message : Le message est synchrome, cela signifie qu'il va attendre une réponse du système

Async message : Le message asynchrome est un message qui n'attend pas une réponse du système.

Reply message : Le reply message est le retour du système d'un message.

Lifeline : la lifeline permet de séparer les différents acteurs sur un diagramme de séquence.

scénario alternatif : scénario envisagé en cas de divergence du scénario nominal

scénario erreur : scénario terminant le processus en le stoppant sans arriver au résultat attendu

Self message : Le système travaille en interne (sur lui-même)

**Modèle logique de données ou MCD**

Entité c’est un objet qui aura les meme caractéristique

Objet

Caract

Clé primaire + etrangère la clé primaire permet d’identifier l’enregistrement unique

enregistrement ou tuple

Association de Type 1 et type n

l’associaion devient une relation

Cardinalités

attributs

**Modèle logique de données ou MLD**

Clé étrangère

Table

Attribut

Type 1 absorbe clé étrangère

Table d’association deux attributs qui sont clé primaire et qui sont également clé étrangère car elle ont été absorbée

Clé primaire absorbée des tables voisine

Table

**SQL**

Alter table

Table