

# Document de Spécifications Logiciel

FALLET Laurent  
JOUANNO Guillaume  
MALLET Grégory  
MARTEAU Sylvie  
MORISSET Samuel\*

6 juin 2003

Pour voir toutes les figures présentes dans ce document en une qualité optimum, il est conseillé de visiter le site <http://asi.insa-rouen.fr/~lfallet/gl/>



FIG. 1 – Logo de TaskForce - Icone du projet

Ce document n'est pas un vrai DSL comme il peut être décrit par la norme IEEE. Il suit l'architecture du document pour la première partie ; la seconde partie consiste en une compilation des documents réalisés durant notre projet, et la dernière partie aborde les parties codées par le groupe prenant en charge notre projet.

---

\*Notre client et correcteur

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Objectifs du document . . . . .	3
1.2	Champ d'application (Description informelle) . . . . .	3
1.3	Définitions, acronymes et abréviations . . . . .	4
1.3.1	Définitions . . . . .	4
1.3.2	Acronymes . . . . .	4
1.4	Références et sources . . . . .	4
1.5	Organisation du document . . . . .	4
<b>2</b>	<b>Compilation des Artefacts</b>	<b>5</b>
2.1	Identification des acteurs . . . . .	5
2.2	Scénarios . . . . .	6
2.3	Grille de Levesque . . . . .	7
2.4	Cas d'utilisation rédigé . . . . .	8
2.5	Maquette . . . . .	9
2.6	Diagramme d'activité . . . . .	10
2.7	Modèle du domaine . . . . .	13
2.8	Diagramme d'états . . . . .	14
2.9	Spécifications détaillées . . . . .	15
<b>3</b>	<b>Cahier de conception pour le groupe codeur</b>	<b>19</b>
3.1	Evaluation des cas . . . . .	19
3.2	Travail à réaliser . . . . .	19
3.3	Diagramme de classe . . . . .	20
3.4	Diagramme de séquences . . . . .	21

## Table des figures

1	Logo de TaskForce - Icône du projet . . . . .	1
2	Identification des acteurs . . . . .	5
3	Scénarios . . . . .	6
4	Diagramme d'activité du Chef de Projet . . . . .	10
5	Diagramme d'activité du MétaDev . . . . .	11
6	Diagramme d'activité du Dév . . . . .	12
7	Diagramme de classes . . . . .	13

## Liste des tableaux

1	Table des acronymes . . . . .	4
2	Grille de Levesque . . . . .	7
3	Evaluation des cas . . . . .	19

# 1 Introduction

## 1.1 Objectifs du document

Ce document a pour but de décrire la conception et le développement du logiciel baptisé "SUIVI". Il comporte également la phase de spécifications.

Ce document s'adresse aux développeurs du logiciel, mais également au client. Les premiers y trouveront tous les détails nécessaires au codage du logiciel. Le client trouvera, dans les 2 premières parties, une description approfondie de ses besoins.

## 1.2 Champ d'application (Description informelle)

Un chef de projet désire manager dans le temps un projet, c'est-à-dire concevoir un diagramme de Gantt dynamique, consultable aisément par les développeurs.

Sa première tâche est de réunir tous les éléments nécessaires à la définition de son projet, et aux objectifs qu'il se fixe. La détermination de ces éléments est externe à notre logiciel, c'est une étape préliminaire.

Le chef de projet dispose de développeurs. Il va constituer des groupes de développeurs, ces groupes étant assignés à une tâche précise. Les groupes sont fixes dans le temps, mais un développeur peut appartenir à plusieurs groupes. Le CdP (Chef de Projet) désigne un méta développeur (MétaDev) qui sera chargé d'indiquer l'avancement de la tâche de son groupe. Les autres développeurs (Dev) du groupe ne peuvent que visualiser l'état d'avancement des tâches du ou des groupes auxquels ils appartiennent.

Les login s'effectueront par nom de personne, pas par nom de groupe. Le logiciel saura de lui-même s'il s'agit du CdP, d'un MétaDev, ou d'un Dev. L'interface sera soit une interface de modification du projet, modification des tâches, ou simple visualisation.

Les tâches peuvent revêtir une importance différente. L'évaluation d'une tâche s'évalue uniquement en temps (journées, semaines, mois, années) et en nombre de développeurs (un groupe). Le CdP peut définir l'importance par une valeur qualitative de 3 niveaux (urgent, normal, non urgent), qui se traduisent par les valeurs quantitatives suivantes : (0%, 5% ou 10% de retard). Ces valeurs sont les valeurs par défaut du logiciel, valeurs paramétrables pour l'ensemble du logiciel.

Voici donc les informations nécessaires à la définition du projet :

- découpage des étapes
- délais pour chaque étape
- importance de chaque étape
- nombre de développeurs et groupe par étape
- le méta développeur
- dates de livraison prévues

Le CdP peut changer ces éléments au cours du projet. Il peut également consulter l'outil de comparaison entre son planning et le déroulement du projet.

Un système de versioning permettra de réaliser une sauvegarde du projet comme l'a conçu le CdP. Une nouvelle version sera sauvegardée à chaque modification par le CdP. L'état d'avancement des tâches (rempli par les MétaDev) aura un système de versioning incrémental. Une mise à jour mineure est celle d'un MétaDev, et une mise à jour majeure englobe toutes les mises à jour mineures d'un intervalle déterminé (fixé à une semaine par défaut, mais paramétrable). On peut ainsi revenir à l'état d'avancement d'une tâche au jour près, en partant d'une MAJ majeure, et en lui ajoutant les sauvegardes incrémentielles. Une date sera associée à chaque version pour pouvoir revenir à un cliché d'un instant donné.

Besoin " surprise " : Lier le temps avec le coût. Ex : une personne a coûté un montant Y pour cette tâche. On peut ainsi évaluer l'efficacité des développeurs.

## 1.3 Définitions, acronymes et abréviations

### 1.3.1 Définitions

**Chef de Projet** : Manager d'un projet dans un domaine quelconque

**Méta-Développeur** : Développeur désigné par le Chef de Projet comme digne de confiance pour mettre à jour l'avancement des tâches

**Développeur** : Le plus souvent ce terme désigne le développeur en tant qu'acteur du logiciel, et non le programmeur de l'application SUIVI

**Groupe** : Ensemble de développeurs réunis sous la direction d'un méta-développeur et assignés à une tâche

**Projet** : Ensemble de tâches regroupées en parties dont le logiciel doit fournir le suivi (consultation et mise à jour)

**Partie** : Ensemble de tâches devant être conduites séquentiellement

**Tâche** : Élément atomique de développement. Ensemble d'actions ne pouvant être scindé sans perdre de vue leurs rôles dans le projet

### 1.3.2 Acronymes

CdP	Chef de Projet
MétaDév	Méta-Développeur
Dév	Développeur

TAB. 1 – Table des acronymes

## 1.4 Références et sources

Notre principale source est le cours de Frédéric BAUCHER (<http://frederic.baucher.free.fr/insa/gl/>)

Les cours sont disponibles sur le site et sur le forum de l'INSA de Rouen (<http://www.insa-rouen.fr/cgi-bin/forumsasi/YaBB.pl?>)

Le document "Rédaction du Document de Spécifications Logiciel" nous a également permis de rédiger ce DSL.

## 1.5 Organisation du document

Dans la seconde partie sont donnés les documents nécessaires à la compréhension du projet. La troisième partie contient des spécifications plus détaillées, et les instructions de codage.

## 2 Compilation des Artefacts

### 2.1 Identification des acteurs

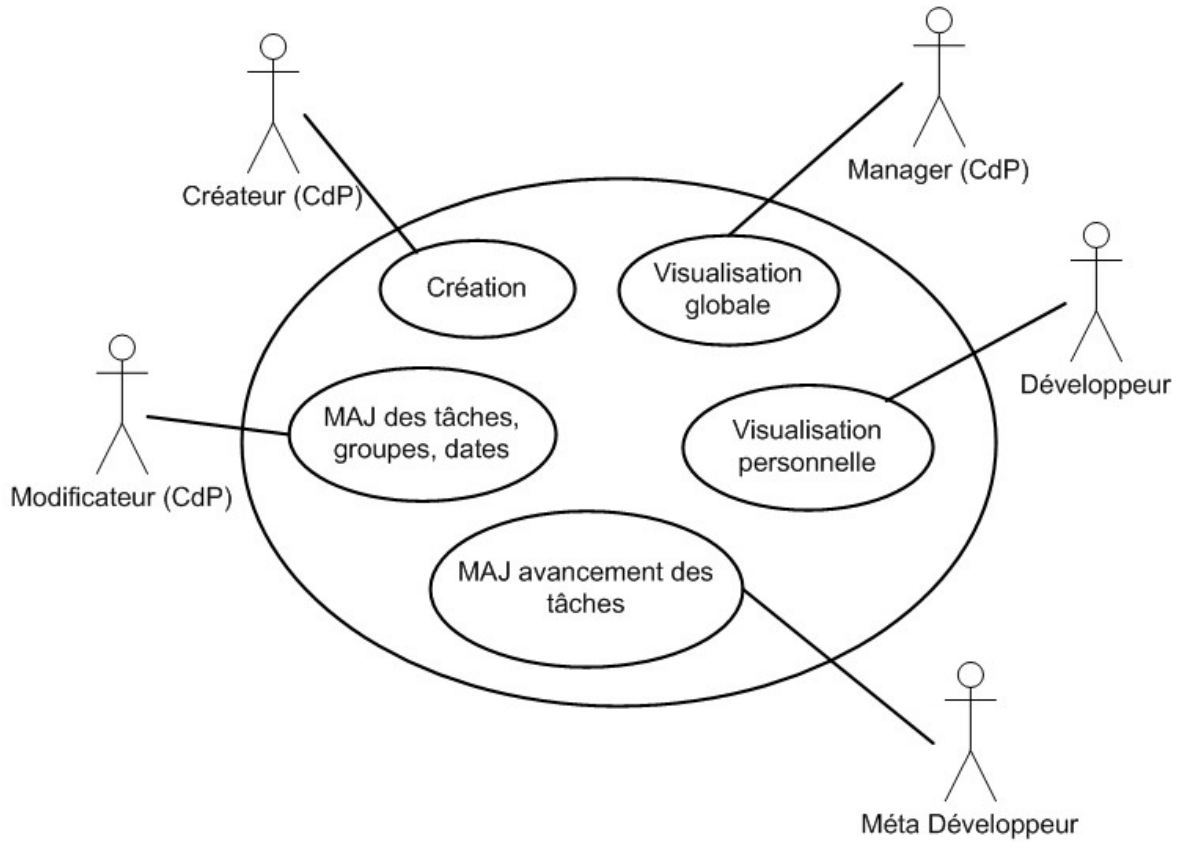


FIG. 2 – Identification des acteurs

## 2.2 Scénarios

<b>Scénarios Version 1.1 (29/03/03)</b>	
<b>Actions d'un développeur</b>	<b>Actions du système</b>
1 - Identification (Log avec pseudo & mot de passe)	2 - Vérification d'un développeur simple
4 - Navigation en visualisation uniquement	3 - Affichage des informations de visualisation
5 - Sortie du logiciel	6 - Fermeture de la session
<b>Actions d'un Méta développeur</b>	<b>Actions du système</b>
1 - Identification (Log avec pseudo & mot de passe)	2 - Vérification d'un méta développeur
4 - Navigation en maj	3 - Affichage des infos de visualisation et maj
6 - Modifications	5 - Affichage de l'interface de maj des tâches du groupe
8 - Sortie du logiciel	7 - Enregistrement par version incrémentale
	9 - Fermeture de la session
<b>Actions du CdP</b>	<b>Actions du système</b>
1 - Identification (Log avec pseudo & mot de passe)	2 - Vérification du CdP
	3 - Affichage des 3 choix:
	A - Créer un suivi de projet
	B - maj des infos du suivi de projet
	C - Visualisation de l'état d'avancement du projet
A - 1 - Création du suivi	A - 2 - Demande des informations relatives au projet
3 - Entrée manuelle des informations, sauvegarde	4 - Création et sauvegarde d'une version majeure
5 - Retour aux choix	
B - 1 - Mise à jour d'un suivi existant	B - 2 - Archivage de la version
3 - Entrée manuelle des informations, sauvegarde	4 - Sauvegarde d'une version majeure
5 - Retour aux choix	
C - 1 - Consultation	C - 2   Récupération de la version majeure la plus récente
	Application des sauvegardes incrémentales
	Affichage des informations
3 - Navigation dans les infos	
4 - Retour aux choix	
Fin - Sortie du logiciel	Fin - Fermeture de la session

FIG. 3 – Scénarios

## 2.3 Grille de Levesque

Hiérarchisation des besoins

1	Voir un planning global du projet
2	Découper le projet en tâches
3	Construire un diagramme de Gantt dynamique
4	Evaluer la durée d'une tâche
5	Définir l'importance des tâches
6	Constituer les groupes de développeurs responsables de tâches
7	Nommer un responsable de la saisie des tâches (dit méta développeur)
8	Permettre à tous les développeurs d'accéder au diagramme dynamique
9	Créer un login par utilisateur
10	Définir l'appartenance de l'utilisateur à un ou plusieurs groupes
11	Sauvegarder la saisie des tâches (sauvegarde mineure)
12	Modifier le diagramme
13	Gérer les versions de documents
14	Mise à jour hebdomadaire et datée du diagramme (sauvegarde majeure)
15	Calcul de la moyenne d'avancement des tâches
16	Calcul du taux d'écart entre le temps saisi et le temps effectif
17	Créer une interface claire et synthétique pour le logiciel
18	Saisie des dates de réunions de coordination par le chef de projet

TAB. 2 – Grille de Levesque

## 2.4 Cas d'utilisation rédigé

Cas d'utilisation rédigé Groupe "TaskForce" - TT3

**Titre :** Saisir les informations relatives au projet dans le logiciel

**Résumé :** Affectation par le chef de Projet des bases de temps aux groupes de développeurs pour l'accomplissement de leurs tâches

**Acteurs :** Chef de Projet

**Pré-conditions :** Objectifs et plan d'avancement du projet mûrements réfléchis  
L'utilisateur est un Chef de Projet qui a un login et mot de passe valide  
Les groupes et développeurs sont définis

**Action de départ :** Ouverture du logiciel

**Enchaînement :**

- entrer le login et mot de passe
- entrer dans le menu de création d'un nouveau projet
- renseigner les champs propres au projet (nom, dates...)
  - nom du projet
  - dates de début et de fin
  - fréquence des réunions
  - entrer un commentaire
- créer une partie
  - entrer le nom
  - entrer un commentaire
- créer les tâches dans l'ordre
  - entrer le nom
  - qualifier l'importance (urgent, normal, non-urgent)
  - dates de livraison prévues
  - sélection du groupe en charge de l'étape
  - sélection du méta développeur dans les membres du groupe

**Action de fin :** Fermeture du logiciel

**Post-conditions :** Un nom doit être indiqué pour le projet, la ou les parties et la ou les tâches

Un projet a au moins une partie

Une partie a au moins une tâche

Un groupe est assigné à chaque tâche

**Exceptions :** Pas de prévue pour le moment



## 2.5 Maquette

nom - groupe - développeur X

Taches du développeur x : terminées à 30 %

1er avril 12 mai

tache 1 tache 2 partie a

tache 3 tache 4 partie b

partie c

Liste des taches affectées à un développeur

nom - groupe - développeur X

Partie a - Tache 1 : terminée à 50 %

1er avril 12 mai

%Description par CdP%

%Commentaire%

Développeurs affectés :

- -

- -

Détail d'une tâche

nom - chef de projet X

Projet : toto > tache : tache 2 > affectations

<p>Choix d'un groupe déjà existant :</p> <div style="border: 1px solid black; padding: 2px; width: 100%;">liste déroulante</div> <p>Ou création d'un nouveau groupe :</p> <div style="border: 1px solid black; padding: 2px; width: 100%;">champ texte</div> <p>Ou modif. d'un groupe existant :</p> <div style="border: 1px solid black; padding: 2px; width: 100%;">liste déroulante</div>	<p>Affectation d'un équipier au gpe x :</p> <div style="border: 1px solid black; padding: 2px; width: 100%;">liste déroulante</div> <p>ou désaffectation d'un équipier :</p> <div style="border: 1px solid black; padding: 2px; width: 100%;">champ texte</div> <p>Ou création d'un nouvel équipier :</p> <div style="border: 1px solid black; padding: 2px; width: 100%;">champ texte</div>
--	--

OK

Affectation et désaffectation des développeurs à un ou plusieurs groupes

## 2.6 Diagramme d'activité

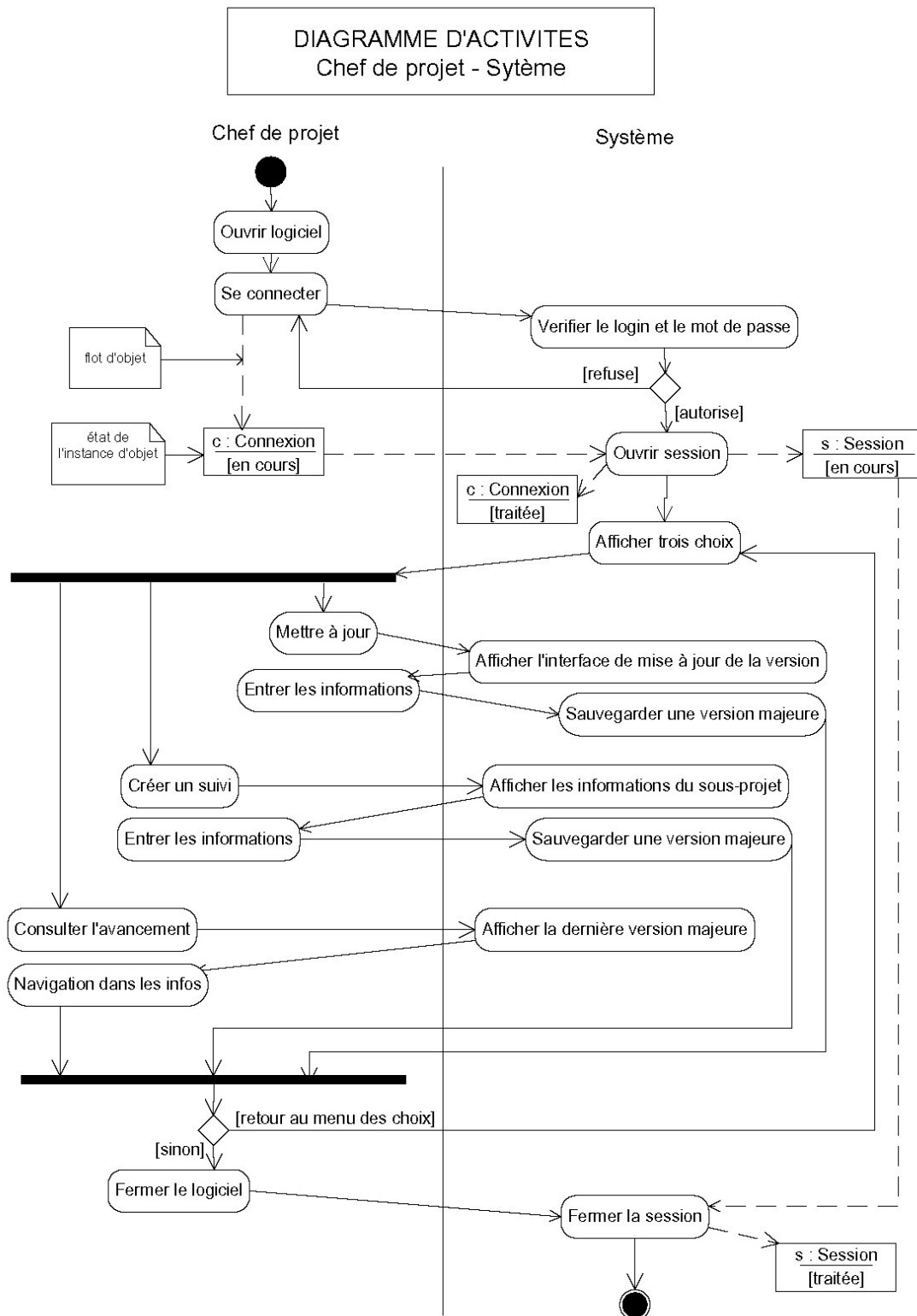


FIG. 4 – Diagramme d'activité du Chef de Projet

DIAGRAMME D'ACTIVITES  
Méta-développeur - Sytème

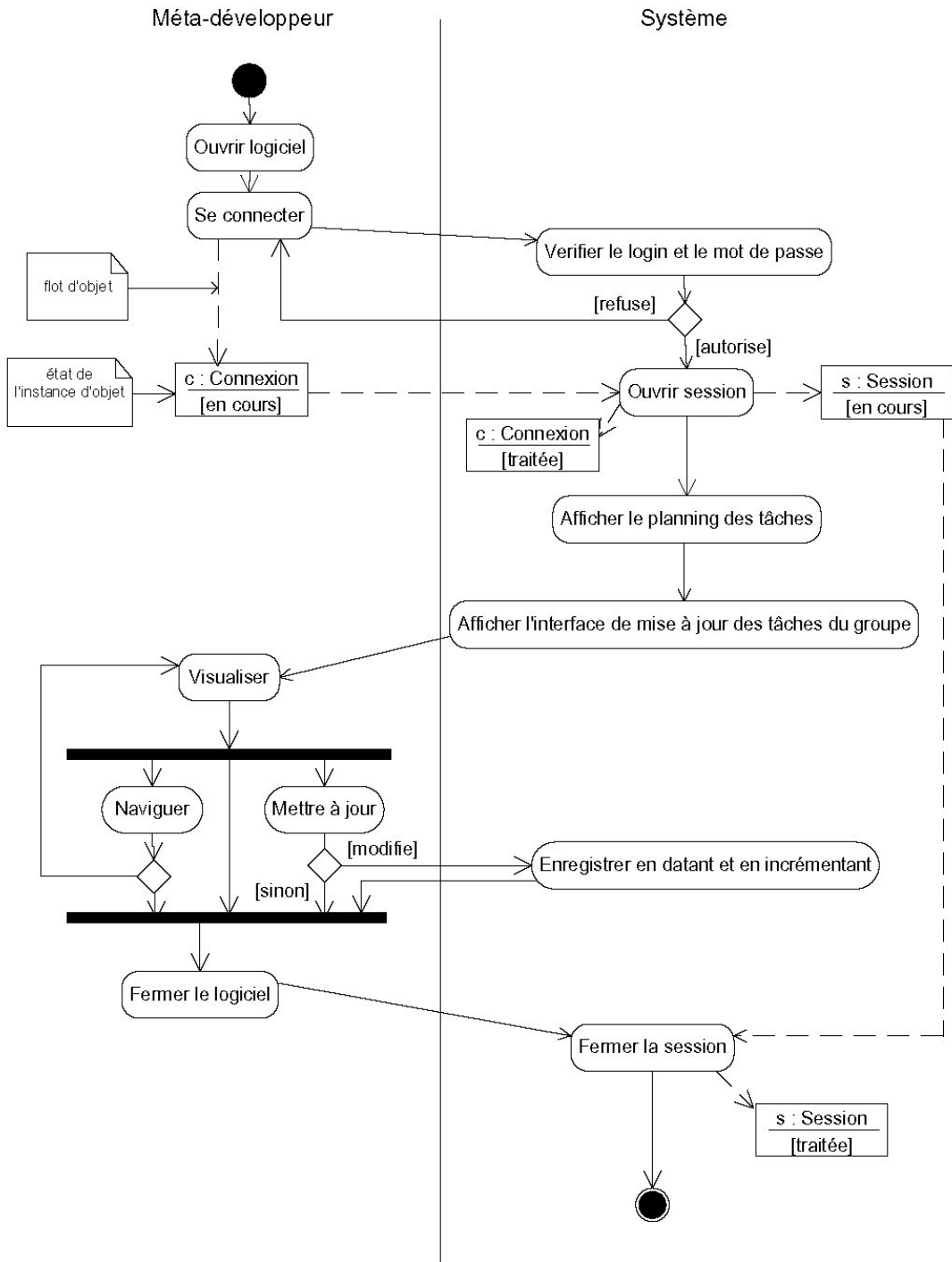


FIG. 5 – Diagramme d'activité du MétaDev

DIAGRAMME D'ACTIVITES  
Développeur - Sytème

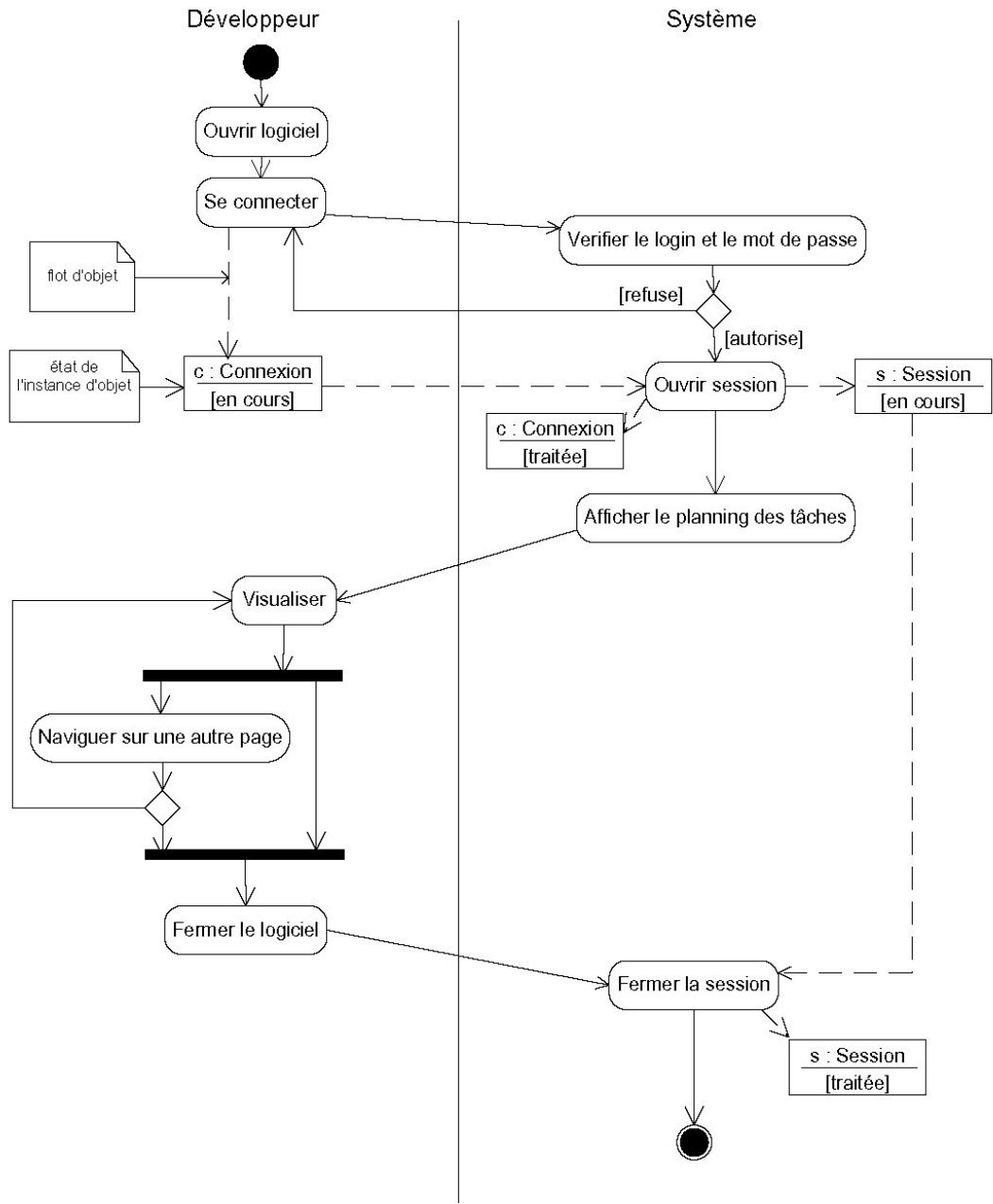


FIG. 6 – Diagramme d'activité du Dév

## 2.7 Modèle du domaine

Projet **SUIVI**  
Diagramme de Classe v1.3 - TT3

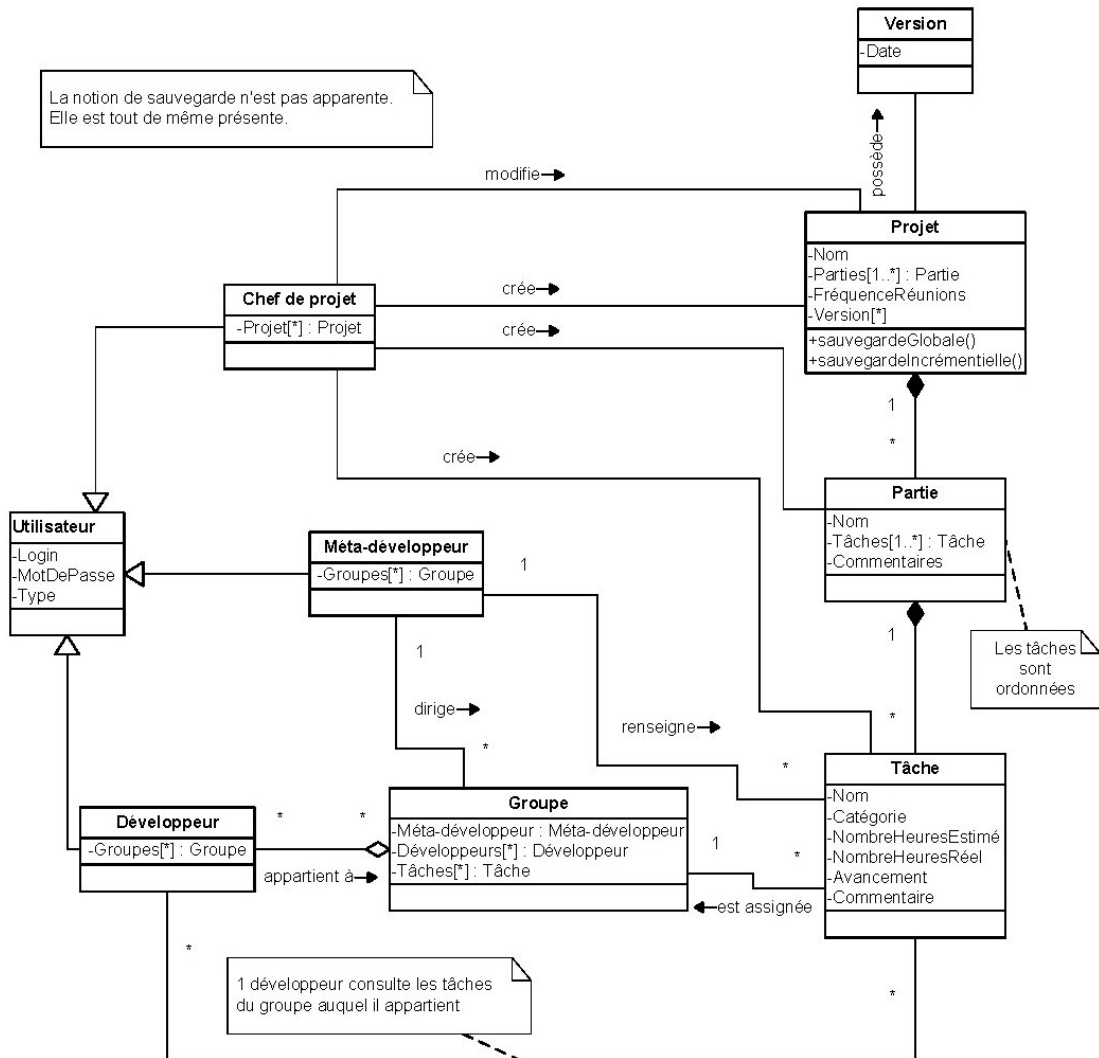
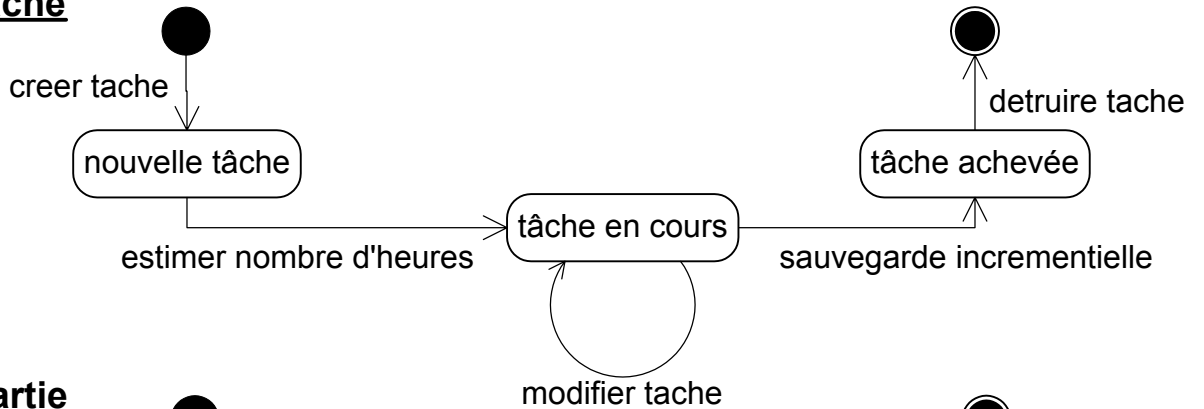


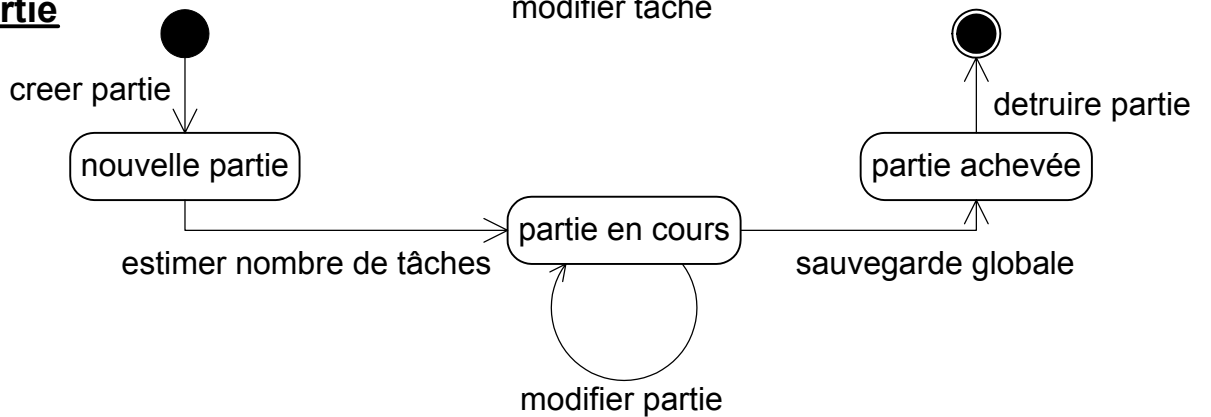
FIG. 7 – Diagramme de classes

## 2.8 Diagramme d'états

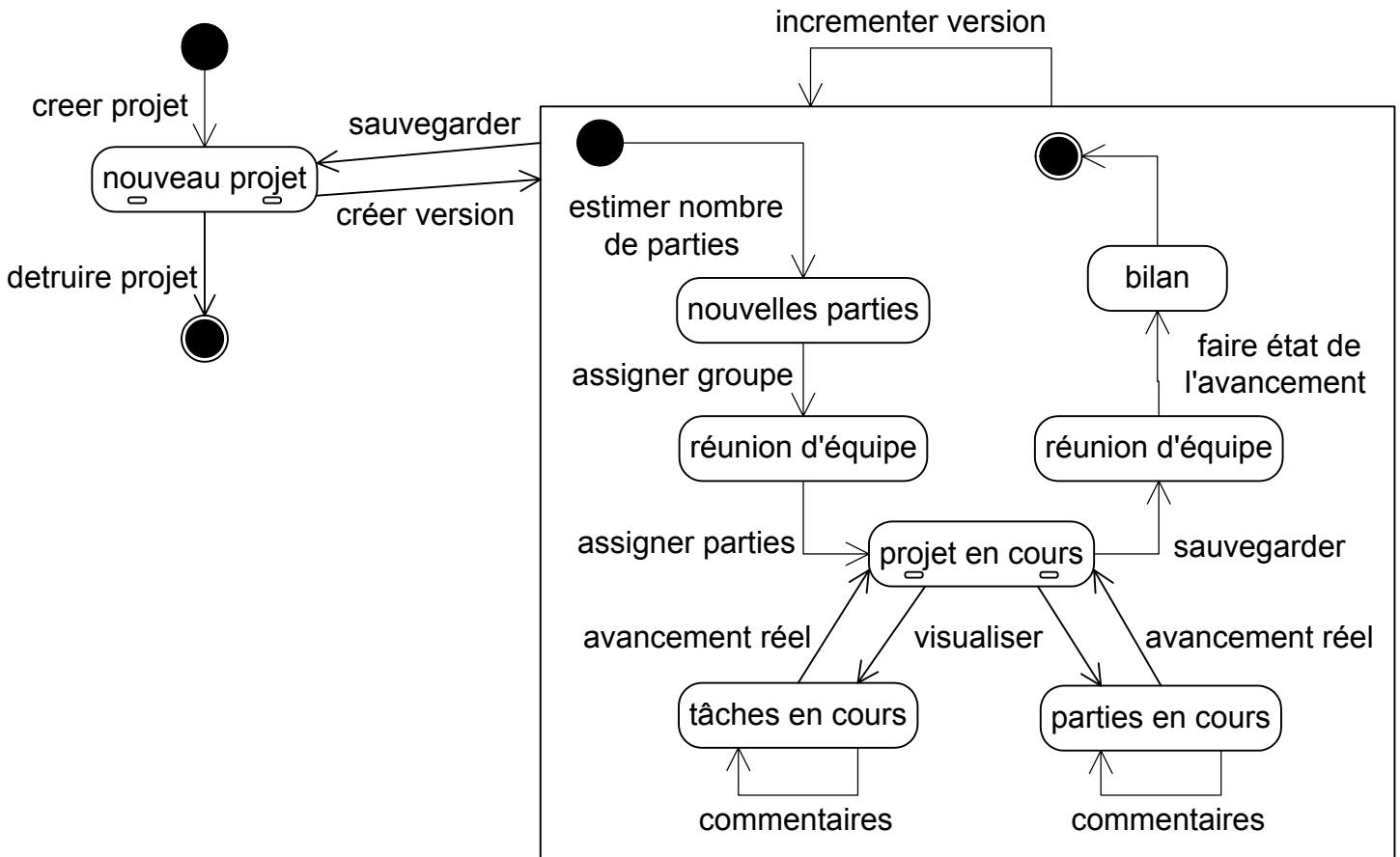
### Etat d'une tâche



### Etat d'une partie



### Etat d'un projet (créé et modifié par le chef de projet)



## 2.9 Spécifications détaillées

### Classe Utilisateur

#### Partie Privée

- fixerLogin (E :Login)
- obtenirLogin() : Login
- fixerMotDePasse (E : MotDePasse )
- obtenirMotDePasse() : MotDePasse
- fixerType (E :Type)
- obtenirType() :Type

#### Partie Publique

- créerUtilisateur (E :Login, MotDePasse, Type) : Utilisateur
- supprimerUtilisateur ()
- modifierUtilisateur (E : Login, MotDePasse, Type)

### Classe Chef De Projet

#### Partie Privée

- fixerProjet(E :Projet)
- obtenirProjets() : {Projet}

#### Partie Publique

- choisirProjet()  
T = obtenirProjet()  
Afficher T  
Récupérer Indice  
Retourner T[indice]
- naviguerProjet(E :Projet)  
Debut  
affichage.afficherProjet(Projet)  
  si "modifier"  
    Projet.modifier(...)  
  fsi  
  si "supprimer"  
    P=Projet.obtenirPartie  
    Pour chaque Partie de P  
      T=P[i].obtenirTache()  
      Pour chaque Tache de T  
        T[j].Supprimer()  
      finPour  
    P[i].supprimer()  
    finPour  
  Projet.supprimer(...)  
  fsi  
  si "afficherPartie(PartieN)"  
    affichage.afficherPartie(PartieN)  
    si "modifier"  
      PartieN.modifier(...)  
    Fsi  
    si "supprimer"  
      T=PartieN.obtenirTache()  
      Pour chaque Tache de T  
        T[j].Supprimer()  
      finPour  
    PartieN.supprimer()  
  fsi

```

        si "TacheN"
            affichage.afficherTache(TacheN)
            si "modifier"
                TacheN.modifier(...)
            Fsi
            si "supprimer"
                TacheN.supprimer()
            fsi
        fsi
    fsi
Fin

```

### Classe Meta-Developpeur

#### Partie Privée

- fixerGroupe(E :Groupe)
- obtenirGroupe() : {Groupe}

#### Partie Publique

- naviguerTaches()
    - Debut
    - G = obtenirGroupe()
    - Pour chaque groupe de G
      - T=G[i].obtenirTaches
      - Pour chaque taches de T
        - affichage.afficherTache(T[j])
        - si "action miseAJour"
          - T[j].miseAJour(...)
    - Fsi
  - FinPour
  - FinPour
  - Fin
- assignerGroupe(E : Groupe)
  - fixerGroupe(Groupe)
  - Groupe.fixerMeta

### Classe Developpeur

#### Partie Privée

- fixerGroupe(E: {Groupe})
- obtenirGroupe() : {Groupe}

#### Partie Publique

- naviguerTaches()
  - Debut
  - G = obtenirGroupe()
  - Pour chaque groupe de G
    - T=G[i].obtenirTaches
    - Pour chaque tache de T
      - affichage.afficherTache(T[j])
  - FinPour
  - FinPour
  - Fin
- assignerGroupe(E : Groupe)
  - fixerGroupe(Groupe)
  - Groupe.fixerDev(...)



## **Classe Groupe**

### Partie Privée

### Partie Publique

- fixerMeta (E : Meta-dev)
- fixerDev (E :Dev)
- fixerTache(E :Tache)

## **Classe Projet**

### Partie privée

- sauvegardeIncrémentale()
- sauvegardeGlobale()  
Debut  
    copier les fichiers ou exporter la base dans un fichier  
    numéroter et dater le fichier et le placer dans le répertoire « sauvegardes »  
Fin
- obtenirDateDerniereSauvegarde()  
Debut  
    récupérer le fichier ayant la numérotation la plus haute  
    le réinjecter dans la base ou dans le répertoire contenant tous les fichiers  
Fin

### Partie publique

- créerProjet(...) : Projet
- modifier(...) (regroupe les fonctions "fixer")
- associerPartie(Partie)
- obtenirParties() : {Partie}

## **Classe Partie**

### Partie privée

- fixerParties(E : {Partie})

### Partie publique

- créerPartie(E : Nom, Commentaires) : Partie  
Debut  
    Tant que le CdP réalise l'action "Ajouter une tâche"  
        P.associerTache(t1)  
    FinTantQue  
Fin
- modifier(...)
- associerTache(Tache)

## Classe Tache

### Partie Privée

- DépassementDelai(NbHeuresEstimees, NbHeuresRéelles, avancement) : Booléen  
(NbHeuresReelles/NbHeuresEstimees > avancement)

### Partie Publique

- créerTache(E : Nom, Catégorie, NombreHeuresEstime, Commentaire) : Tache
- modifier(E : Nom, Catégorie, NombreHeuresEstime, Commentaire) – CdP  
Debut  
    affichage.afficherTache(T)  
    On a toutes les infos de la tâches comme elle est à l'heure actuelle  
    Le CdP modifie les champs puis valide  
Fin
- miseAJour(E : NombreHeuresReel) - Meta

## Classe Interface

- afficherProjet(E : Projet)  
Debut  
    Pour i allant de 1 à NbreDeProjets  
        afficherPartie(projet, i)  
    FinPour  
Fin
- afficherPartie(E : Projet, Partie)
- afficherTache(E : Projet, Partie, Tâche)  
Debut  
  
Fin
- afficherGroupe(E : Groupe)
- afficherUtilisateur(E : Utilisateur)  
Debut  
    Ecrire(login, type)  
    Si type= « CdP » alors  
        afficher les projets à la charge du CdP  
    Sinon si type= « MetaDev » alors  
        afficher les groupes à la charge du MetaDev  
    Sinon si type= « Dev » alors  
        afficher les tâches à la charge du Dev  
Fin

### 3 Cahier de conception pour le groupe codeur

#### 3.1 Evaluation des cas

Cas d'utilisation	Couverture fonctionnelle	Risque technique
Création d'un projet complet	Haute	Moyen
Création d'un projet à 2 parties	Haute	Faible
Visualisation des tâches d'un développeur	Haute	Faible
Mise à jour d'une tâche par un MétaDév	Moyenne	Moyen
Gestion des sauvegardes complètes	Faible	Moyen
Gestion des sauvegardes incrémentielles	Moyenne	Haut
Gestion des groupes	Faible	Haut
Gestion des 3 types d'utilisateurs	Moyenne	Moyen
Versionning	Moyenne	Haut

TAB. 3 – Evaluation des cas

#### 3.2 Travail à réaliser

L'équipe chargée de l'implémentation du logiciel "SUIVI" n'aura pas à coder l'ensemble des cas d'utilisation, mais seulement celui-ci : "Création d'un projet constitué de 2 parties contenant 2 tâches chacune. Un seul utilisateur sera créé, celui du chef de projet."

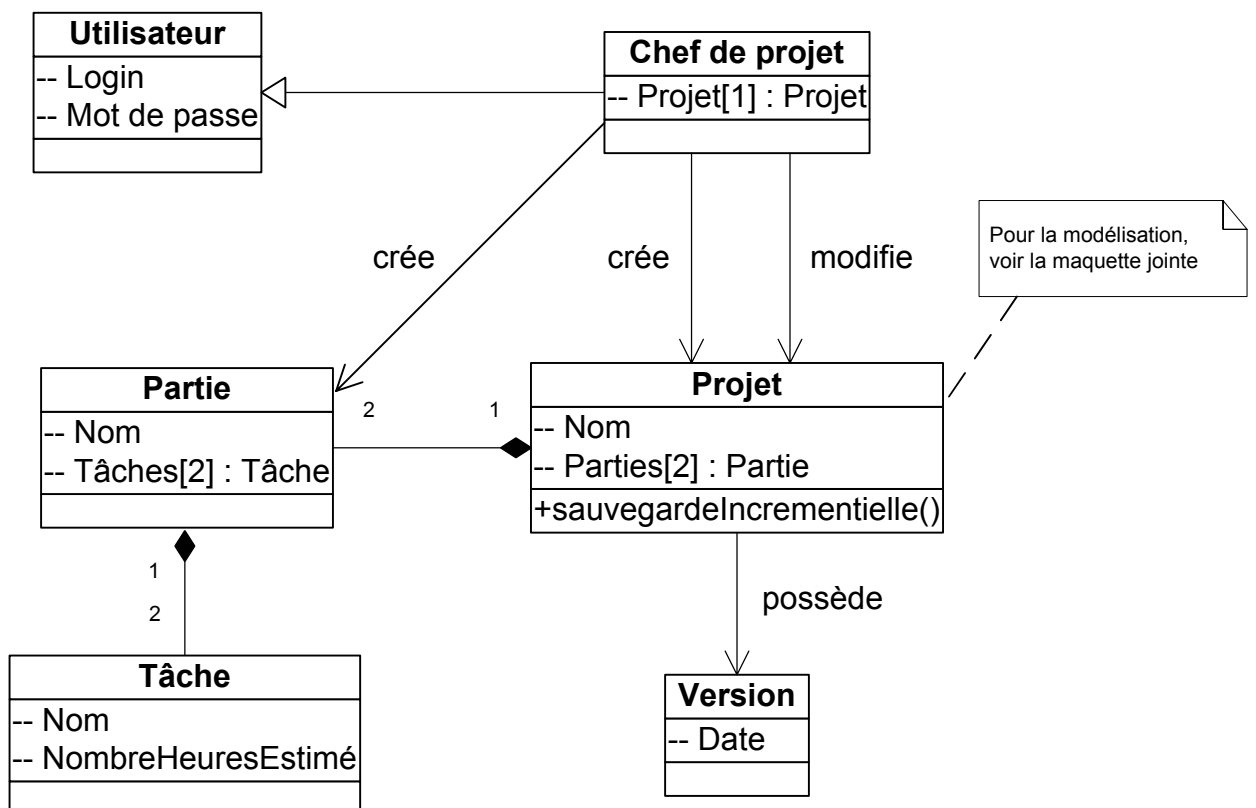
Le langage, la plate-forme utilisée pour l'implémentation du logiciel sont laissés au choix de l'équipe de codage. La seule contrainte est la possibilité du visionnage sur un réseau (fondement même du logiciel).

### 3.3 Diagramme de classe

Version spéciale pour le groupe de codage TT7

Explications pour le codage (comparaison du diagramme de classes complet avec celui destiné au codage) :

- modéliser l'utilisateur chef de projet (pas les autres utilisateurs)
- projet : créer son nom, sa version et deux parties seulement
- modéliser 2 parties (= deux tâches principales), avec deux sous-tâches par partie, une sauvegarde incrémentielle, pas de sauvegarde globale
- modélisation des tâches : nom et nombre d'heures estimé
- visualiser la maquette de modélisation du projet



### 3.4 Diagramme de séquences

#### Associé au cas d'utilisation : création de projet de suivi

