MINISTERE DE L’ENSEIGNEMENT SUPERIEUR

DE LA RECHERCHE SCIENTIFIQUE

---\*\*\*\*---

Brevet de Technicien Supérieur



**Rapport Projet de Fin d’Etudes**



Réalisation d’une Application web dynamique

de gestion de matériel

**Réalisé par : Encadré par :**

* Zineb dekini Mr. Hamid Mechhour
* Ait Taleb Ouiam Mr.Hamid Alhaine

**Année Universitaire 2016-2017**

**Résume :**

Le présent projet, Réalisation d’une application web dynamique de gestion de matériel. Des promotions, est réalisé dans le cadre de la préparation du projet de fin d’étude présenté en vue de l’obtention du diplôme de technicien supérieur pour l’année universitaire 2016/2017

Ll s’agit d’une application web appelé << gestion de matériel >> qui est développée en java , servlet , jsp sous la plateforme JEE .cette application consiste a gérer le matériel de BTS en assurant les fonctionnalité suivantes :

* Gestion des professeurs
* Gestion du matériel ( nombre, caractéristiques, affectation, ...)
* Gestion des emprunts de matériels

**Mots clés :**

Jsp,servelet ,JEE,

****

**Dédicace :**

Avant toutes choses,

On aime bien dédier ce rapport

A nos chers parents qui nous ont beaucoup

Soutenus et encouragé pendant toute la période des études,

Et en particulier à nos mères qui pour elle seule nous avons pu confronter

L’impossible pour atteindre ce qui est meilleur

A tous nos camarades de classe, sans oublier

Nos chères fidèles amis, pour eux tous, on dédie ce rapport

Finalement, on tient aussi à dédier ce rapport à nos chers formateurs pour

Tous leurs efforts durant notre formation à BTS de Kenitra.

**Remerciements :**

A toutes nos familles, Aucun terme et aucune langue ne pourra exprimer

Nos amours et nos sentiments envers vous.

On tient à remercier aussi tout le corps enseignant qui a déployé tous ses

Efforts pour assurer le bon déroulement de notre formation.

Nos remerciements s’adressent également à nos encadrant : Mr. Lajouad ,

Mr. Bouabid ainsi que Mr. Machhour et Mr. Alhiaine qui

Ont eu L’amabilité de veiller à ce que notre rapport de stage soit enrichissant

C’est avec un réel plaisir que nous exprimons ici

Nos reconnaissances à tous ceux qui ont contribué de près ou de loin à

L’élaboration de ce travail.

Finalement, on espère que nous avons été à la hauteur des attentes de nos

Encadrant bien au niveau technique

Et de productivité qu’au niveau du comportement.

**Table des matières :**

Introduction générale 10

**Chapitre1 : Etude préalable et spécification des besoins**

* 1. Introduction……………………………………………………………………………13
  2. Etude de l’existant……………………………………………………………..………13
  3. Analyse de l’application web Stock Net…………………………………………..… 13
     1. Présentation de l’application web Stock Net …………………………..14
     2. Analyse Fonctionnelle……………………………………………………………. 14
     3. Analyse Technique…………………………………………………………..……..15

1.4 Choix de la méthodologie de conception adoptée: ……………………12

[1.4.1 Choix de la méthode de conception : 13](#_Toc230317458)

[1.4.2 Choix du cycle de vie logiciel : 13](#_Toc230317459)

* 1. [Planning des tâches : 15](#_Toc230317460)

1.6 Spécification des besoins fonctionnels………………………………………..………..17

* + 1. Identification des acteurs…………………………………………………………..17
    2. Description des exigences fonctionnelles………………………………...……… 18

**Chapitre2 : Conception technique**

* 1. Introduction…………………………………………………………………………....20
  2. Conception graphique………………………………………………………………......20
     1. Charte graphique…………………………………………………………………...20
     2. Choix des formes et des lignes…………………………………………………..…20
     3. Choix de la typographie……………………………………………………………21
  3. Conception détaillée…………………………………………………………………...21
  4. Description de la vue statique……………………………...…………………………...23
     1. Diagramme de classe de l’application………………………...……………………23
     2. Cas d’utilisation global………………………………………….…………………23
  5. Cas d’utilisation global relatif à un employé………………………………………….24
  6. Le diagramme de cas d’utilisation relatif au secrétariat…………………………..…...25
  7. Cas d’utilisation relatif à un directeur ………………………………………………...26
  8. Cas d’utilisation relatif à un directeur d’étude………………………………………...27
  9. Description de la vue dynamique…………………………………………………...…28
  10. Les diagrammes de séquence …....................................................................................28
  11. Diagramme de séquence d’authentification…………………………………………...29
  12. Diagramme de séquence de gestion des demandes administratives…………......……30
  13. Diagramme de séquence de gestion d’absences des employés………………………..31
  14. Diagramme de séquence de gestion des fiches personnelles des employés………...…32
  15. Diagramme de séquence de gestion des rapports journaliers………………………….33

1. Chapitre 3 : Réalisation et Test

Chapitre 3 : Réalisation et Test

Table des matières

* 1. Diagramme de séquence de gestion des listes des employés………...………………..33
  2. Conclusion…………………………………………………………………………….34

**Introduction générale :**

D

ans le cadre de notre projet de fin d’étude, nous nous intéressons à

Développer une application qui sert à la gestion de matériel, Cette application peut être développée sur la plates-formes JEE .

La création d'applications web avec Java EE semble compliquée à beaucoup de débutants. Une énorme nébuleuse de sigles en tout genre gravite autour de la plate-forme, un nombre conséquent de technologies et d'approches différentes existent : servlet, JSP, Java Bean, MVC, JDBC, JNDI, EJB, JPA, JMS, JSF, Struts, Spring, Tomcat, Glass Fish, JBoss, WebSphere, Weblog... La liste n'en finit pas, et pour un novice ne pas étouffer sous une telle avalanche est bien souvent mission impossible !

..

Non, ce cours a pour objectif de guider vos premiers pas dans l'univers Java EE : après quelques explications sur les concepts généraux et les bonnes pratiques en vigueur, vous allez entrer dans le vif du sujet et découvrir comment créer un projet web, en y ajoutant de la complexité au fur et à mesure que le cours avancera. À la fin du cours, vous serez capables de créer une application web qui respecte les standards reconnus dans le domaine et vous disposerez des bases nécessaires pour utiliser la plupart des technologies se basant sur Java EE.

Donc on a décidé de développer une application web dynamique de gestion de matériel pour gérer les matériel de l’établissement technique Ibn Sina.

L'application sera déployer dans le serveur de l'établissement, ces utilisateur pourront donc s'y connecter via un client Http.

**Cadre du projet**

Ce travail s’inscrit dans le cadre de notre projet de fin d’études pour

L’obtention du diplôme de Technicien en Informatique. Au cours de ce projet,

Nos encadrant nous ont confié la conception et la réalisation d’une application

Dans un milieu pur professionnel.

**Organisation du Rapport**

Au niveau de ce manuscrit, nous décrivons la mise en place d’une application

Permettant la gestion des ressources humaines. Le rapport présente

Aussi les impacts opérationnels, internes et externes.

Dans un premier chapitre nous présentons le cadre général, nous analyserons

Quelques applications spécialisés dans la gestion des ressources humaines afin

De dégager les besoins de notre application. Dans le deuxième, nous

Procéderons à la description de la conception de la solution. Ensuite, nous

Détaillerons les spécifications et les besoins. Dans le troisième chapitre, nous

Procéderons à la Réalisation et les tests. Les annexes se situeront au niveau du

Quatrième chapitre De notre rapport.

Nous terminons le rapport par une conclusion générale et nous

Présentons quelques perspectives et futur améliorations concernant notre

Application.

**Etude préalable**

**Et**

**Spécification des besoins**

**Introduction :**

Dans le cadre de ce chapitre, nous allons présenter la spécification de besoins

Qui consiste à la qualification des besoins fonctionnels et non fonctionnels

Attendus du système afin de mieux comprendre le projet.

**Etude de l’existant :**

L’étude de l’existant permet de déterminer les points faibles et les points

forts d’un produit actuel pour pouvoir déterminer les besoins du client, en

vue d’en prendre en considération lors de la conception et la réalisation

de la boutique en ligne. Dans cette section, nous présentons une analyse de

quelques exemples d’applications marchands. Ensuite, nous formulerons une solution de la problématique.

**Analyse de l’application web Stock Net :**

Dans cette section, nous présentons l’application Stock Net. Ensuite

nous procédons à une analyse fonctionnelle, technique et graphique de l’application étudié.

**Présentation de l’application web Stock Net :**

Stocknet est une application de gestion de stock en ligne, hébergée sur nos serveurs et accessible depuis n'importe quel poste connecté à Internet (on parle de mode ASP).

Stocknet est un logiciel qui se loue (on parle de mode SAAS ou Software As A Service).

Pionnier depuis 2003 sur ce marché innovant de l'informatique externalisée (ou Cloud computing), Stocknet a évolué dans un esprit de réactivité, en accompagnant les besoins de ses clients : fonctions de gestion d'entrepôts, des achats et des ventes,... Stocknet est désormais devenu un véritable PGI (Progiciel de Gestion Intégré)

**Analyse fonctionnelle :**

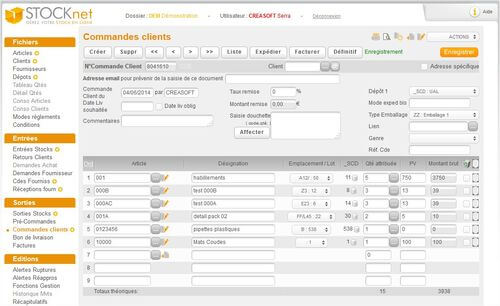
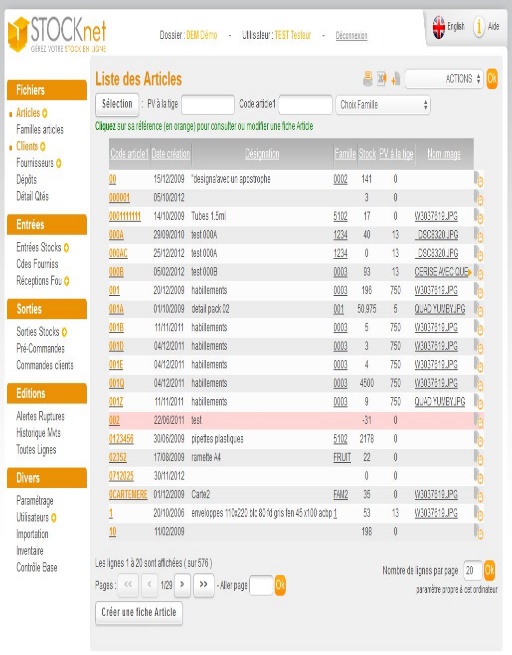
Tel que le montre la figure 1.1, l’application Stock Net offre plusieurs fonctionnalités au visiteur. Il lui permet de :

1. Effectuer immédiatement des achats parmi la sélection des matériels de l’application Stock Net et ce ci via les serveurs sécurisés du site.
2. Consulter des images de matériel et lire des commentaires de client où que vous soyez et au moment que vous choisirez.
3. Suivre le statut d’une commande récente depuis votre téléphone.

4. Obtenir immédiatement des informations sur le matériel, dont son prix et

sa disponibilité (pour les téléphones avec un appareil photo autofocus)

**1.3 Analyse de l’application web Stock Net :**

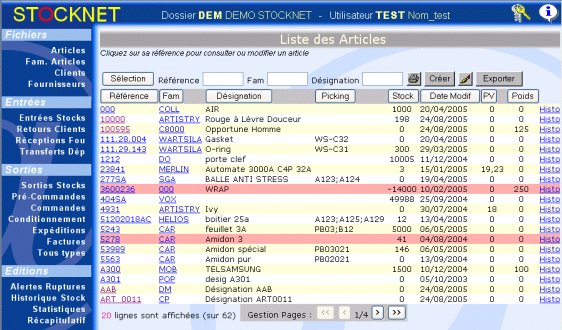
 

Figure 1.1 – application Stock Net

**1.6 Spécification des besoins fonctionnels :**

– La figure 1.5 (a) illustre la page de la description détaillé d’un matriel

– La figure 1.5 (b) illustre la page de la comparaison entre deux produit

– La figure 1.5 (c) illustre la liste des services offertes par l’application

– La figure 1.5 (d) illustre le résultat d’une recherche

**1.5.3 Analyse technique :**

L’application Stock Net est développée avec le langage java

Les images contenues dans l’application sont des images dont le

format est JPEG et GIF.

|  |  |
| --- | --- |
| Dénotation | Connotation |
| La page d’accueil est disposée en  Deux parties horizontales. | Cette disposition donne un sens  précis de lecture qui rend la page plus  large. |
| La gamme des couleurs utilisées sont  les tons du bleu, le vert et noir. | L’utilisation du contraste entre le  bleu, le noir et le vert ravive  l’interface.  L’utilisation du noir et du vert pour réveiller le bleu et empêcher sa continuité. |
| Les formes utilisées sont des formes  rectangulaires avec des coins arrondis. | Les formes rectangulaires donnent de la stabilité à l’interface et les coins arrondis ajoutent du dynamisme. |

Table 1.3 – Connotation et dénotation du l’application Stock Net.

* 1. **Choix de la méthodologie de conception adoptée:**

« La qualité du processus de développement d’un logiciel est garante de la qualité du produit.» [4]. La conception est une étape fondamentale dans le cycle de vie d’une application informatique. En effet, c’est d’elle que dépendent la qualité et la cohérence du produit réalisé au développement. Des méthodes de génie logiciel ont alors été développées afin de guider le concepteur dans sa tâche.

Pour mener à bien ce travail, il est nécessaire de définir une méthodologie de travail. Dans cette partie, nous présentons les choix conceptuels en termes de méthode de conception, d’outils techniques et de cycle de vie logiciel.

* 1. Choix de la méthode de conception :

Le plus grand avantage d’une méthode orientée objet est qu’elle permet de structurer un système sans centrer l’analyse uniquement sur les données ou uniquement sur les traitements mais sur les deux à la fois. Une telle approche a pour but de modéliser les propriétés statiques et dynamiques de l’environnement du système. Elle met en correspondance le problème et la solution, en préservant la structure et le comportement du système analysé.

Ceci, nous a conduit à adopter l’approche orientée objet pour modéliser notre système en se basant sur les diagrammes UML [3]. (cf. figure 2).

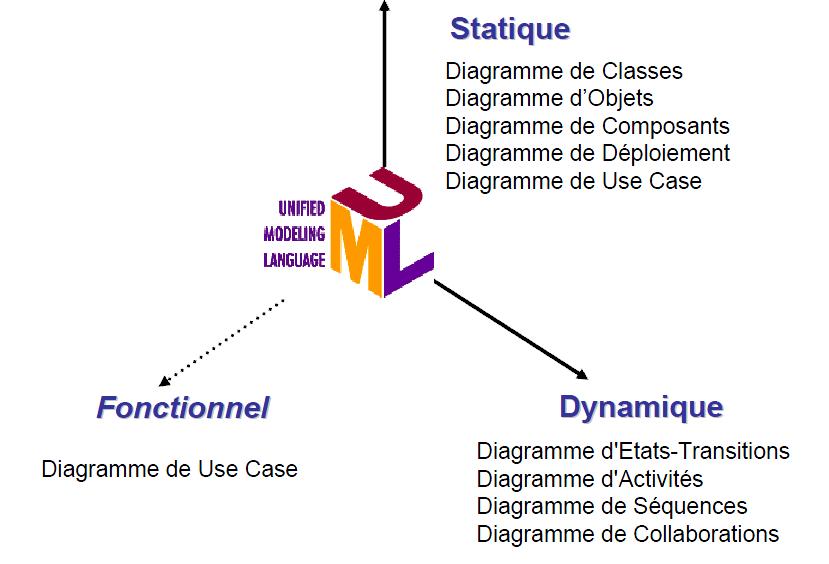


Figure 2: Les différentes vues du langage UML

* 1. Choix du cycle de vie logiciel :

Le choix de la méthodologie qui sera utilisée doit répondre aux critères suivants :

* Exprimer au mieux les besoins des futurs clients
* Permettre de développer une application robuste et évolutive
* Elaborer une application répondant aux besoins des clients dans des délais respectables.

C’est pour cette raison qu’on a choisi d’adopter le processus unifié comme étant la méthode de l’étude conceptuelle. En effet, cette méthode englobe l’ensemble des activités exigées par un projet logiciel à travers un ensemble de principes génériques, adaptables en fonction des spécificités des projets.

Donc c’est un Framework de processus générique pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d’application, à différents types d’entreprises, à différents niveaux de compétences et à différentes tailles de projets.

Ce processus présente plusieurs avantages, que nous citons :

* Limiter les coûts, en termes de risques, aux strictes dépenses liées à une seule itération.
* Limiter les risques de retard de mise en place du produit développé.
* Accélérer le rythme de l’ensemble du développement.
* Prendre en compte le fait que les besoins des utilisateurs et les exigences correspondantes ne puissent être intégralement définies à l’avance.

A cet effet, il est intéressant de voir de quoi est composée la méthodologie, choisie dans le cadre de ce projet, « Processus unifié (PU) » :

En fait, le Processus Unifié (UP) est un processus de développement logiciel « **itératif** et **incrémental**, **centré sur l’architecture**, **guidé par les cas d’utilisation** et **piloté par les risques**» :

* Itératif et incrémentale : le projet est découpé en itérations ou étapes de courte durée qui permettent de mieux suivre l’avancement global. A la fin de chaque itération une partie exécutable du système finale est produite de façon incrémentale (par ajout).
* Centré sur l’architecture : tout système complexe doit être décomposé en partie modulaire afin d’en faciliter la maintenance et l’évolution. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML, et pas seulement documentée en texte.
* Piloté par les risques : les risques majeurs du projet doivent être identifiés au plutôt mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l’ordre d’itération.
* Guidé par les cas d’utilisation : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d’utilisation du futur système sont identifiés, décrits avec précision et classés par priorité.

La gestion d’un tel processus est organisée selon les quatre phases suivantes : initialisation, élaboration, construction et transition. (cf. figure 10)

La phase d’initialisation conduit à définir la « vision » du projet, sa portée, sa faisabilité, son « business case », pour décider au mieux de sa poursuite ou de son arrêt.

La phase d’élaboration poursuit trois objectifs principaux en parallèle :

* Identifier et décrire la majeure partie des besoins utilisateurs ;
* Construire (et pas seulement décrire dans un document) l’architecture de base du système ;
* Lever les risques majeurs du projet.

La phase de construction consiste surtout à concevoir et à implémenter l’ensemble des éléments opérationnels (autres que ceux de l’architecture de base). C’est la phase la plus consommatrice en ressources et en efforts.

Enfin, la phase de transition permet de faire passer le système informatique des mains des développeurs à celles des utilisateurs finaux. Les mots-clés en sont : conversion des données, formation des utilisateurs, déploiement, bêta-tests.

Les activités de développement sont définies par cinq disciplines fondamentales qui décrivent la capture des exigences, l’analyse et la conception, l’implémentation, le test et le déploiement. La modélisation métiers est une discipline en amont, optionnelle et transverse aux projets. PU doit donc être compris comme une trame commune des meilleures pratiques de développement et non comme l’ultime tentative d’élaborer un processus universel.

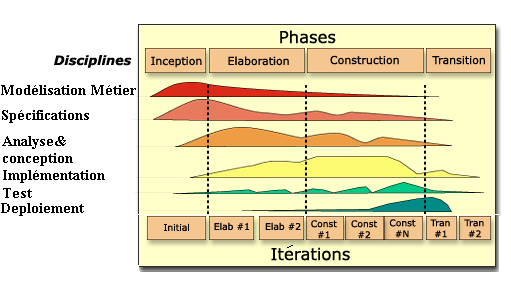


Figure 3: Le schéma synthétique du PU (Processus Unifié)

* 1. Planning des tâches :

La figure ci-dessous (cf. figure 4) illustre la planification des tâches, qu’on a prévues, durant la période du stage :

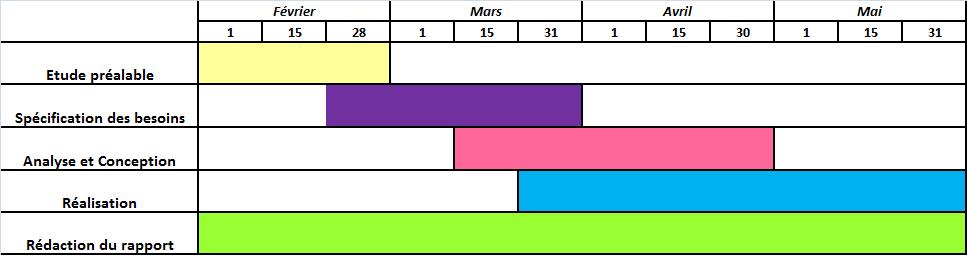


Figure 4: Planning de réalisation

Etude de domaine :

BTS brevet de technicien supérieur est une école a plusieurs branches.

Les étudiants ont besoin des matériels pour appliquer et pratiquer ce qu’ont appris.

Ces matériels sont stockés dans un magasin : est un espace de conservation fermé, gérer par le magasinier.

Les professeurs peuvent emprunter ces matériels avec des différentes mesures.

Problématique :

Les problématiques rencontrées lors de la gestion du matériel par le magasinier sont :

* Les données n’ont pas bien organisées.
* Cette méthode prend plus de temps.
* Perte de données reliées aux professeurs et aux matériels.
* L’archivage papiers est une grande espace de stockage
* L’absence des statistiques.
* La Redondance des informations.
* Le magasinier rencontre des difficultés pour chercher ou modifier

Les informations d’un tel matériel.

Solution :

Réaliser une application Dynamique permettre de gérer le matériel de BTS en assurant la fonctionnalité suivantes :

• Gestion des professeurs

• Gestion du matériel (nombre, caractéristiques, affectation, ...)

• Gestion des emprunts de matériels

L'application sera déployer dans le serveur de l'établissement, ces utilisateur pourront donc s'y connecter via un client Http.

* 1. 1.4 Spécification des besoins fonctionnels

Dans cette partie nous identifions les acteurs de notre application afin de

Pouvoir dégager les besoins fonctionnels.

1. Identification des besoins fonctionnels :

Nous présentons dans ce qui suit les différentes fonctionnalités du système. Ces fonctionnalités doivent exprimer les attentes des différents utilisateurs envers notre application. Les besoins spécifiés doivent être persistants, spécifiques et réalisables.

Les besoins fonctionnels doivent répondre aux questions suivantes :

* A quoi sert le système ?
* Ce qui doit faire le système et ces fonctions outils ?
* La description des données manipulées ?

1. Identification des acteurs

L’application doit fournir un ensemble de fonctionnalités aux utilisateurs.

En effet, l’application permettra aux personnels de l’établissement technique Ibn

Sina d’effectuer des demandes des documents administratifs, de consulter ses

Informations personnelles, ses absences ainsi que l’état de sa demande.

Cependant, nous parvenons à dégager quatre acteurs principaux qui sont :

- magasiner qui utilise l’application et gérer le stock de matériel

-professeur qui emprunte et consulter les matériel

-technicien qui doit réparer les matériel

1. Identification des cas d’utilisation du système :
2. Définition d’un cas d’utilisation :

Un cas d’utilisation représente une fonctionnalité du système qui a une plus value attendue et mesurable à chacun des utilisateurs potentiels du système [3]. Ainsi, un cas d’utilisation représente un ensemble de séquences d’actions qui sont réalisées par le système et qui produisent un résultat observable et intéressant pour un acteur particulier.

Il modélise un service rendu par le système. Il exprime les interactions acteurs/système et apporte une valeur ajoutée « notable » à l’acteur concerné.

De ce fait, les cas d’utilisation sont, principalement, utilisés pour :

* définir le contour du système à modéliser (ils précisent le but à atteindre),
* et aussi, pour permettre d'identifier les fonctionnalités principales (critiques) du système.

1. Description des exigences fonctionnelles

Cette application devra permettre de gérer le matériel de BTS en assurant les fonctionnalité suivantes :

Gestion des professeurs

Gestion du matériel ( nombre, caractéristiques, affectation, ...)

Gestion des emprunts de matériels

L'application sera déployer dans le serveur de l'établissement, ces utilisateur pourront donc s'y connecter via un client Http.

Les professeurs peuvent consulter la disponibilité ou les caractéristiques d'un matériel. Le magasinier est la personne en charge de la gestion du magasin de stockage du matériel, l'application devra lui permettre de ( gérer les professeur, gérer le matériel, gérer les emprunt).

**1.4.2 Description des exigences fonctionnelles**

L’application doit permettre au magasiner de :

1. Gérer le stock du matériel

L’application doit permettre au technicien de :

1. Gérer les matériels en panne

L’application doit permettre au professeur de :

1. Consulter la disponibilité de matériel
2. Emprunte un matériel
3. Consulter les caractéristique du matériel
   * 1. Classification des cas d’utilisation par acteur :

Dans cette partie, nous recensons les principales fonctionnalités offertes par le système, tout en les associant aux acteurs qui devront en bénéficier.

|  |  |
| --- | --- |
| **Les acteurs du système** | **Les cas d’utilisation à travers lesquels ils réagissent** |
|  | Gérer les matériels en panne |
|  | Gérer le stock du matériel |
|  | Consulter la disponibilité de matériel  Emprunte un matériel  Consulter les caractéristique du matériel |

**Chapitre2 :**

**Conception technique**

**Introduction :**

Aujourd’hui, l’informatique a atteint une prodigieuse évolution technologique tans différents domaines (réseaux informatiques, bases de données, le web, etc.) cette évolution est nécessaire pour remédier aux problèmes rencontres dans la vie actuelle.

Le dynamisme est l’une des caractéristiques les plus essentielles de l’informatique. Ce c’est qui nus a pousses à créer une application dynamique, qui sera aussi accessible par des utilisateurs dans un réseau informatique local.

Chaque création nécessite une modélisation avec un langage universel bien spécifie tel qu’UML, la réalisation quant a elle nécessite des outils de développement bien adaptes au contexte de l’application. Pour les bases de données, l’utilisation d’un SGBD tel que MYSQL

Notre travail a abouti a la conception d’une application en utilisant une base des données, pour la gestion des matériels du BTS .l’application a été développée en général avec ECLIPSE, le langage de programmation utilise est le java EE.

Dans cette partie nous présentons une modélisation de notre futur

Application. Nous construisons une vue statique de la solution sous forme de

Diagrammes de classes et aussi une vue dynamique sous forme de diagramme de séquence.

***2.2 Conception graphique***

Lors de cette étape, nous essayerons de mettre en harmonie les couleurs, les

Formes, et la typographie. Nous présentons dans cette partie la charte graphique

De l’établissement.

2.2.1 Charte graphique

La charte graphique est un document de travail comprenant les

Recommandations d’utilisation et les caractéristiques des différents éléments

Graphiques qui peuvent être utilisés sur les différents supports de

Communication de l’enseignement. La charte graphique permet de garantir

L’homogénéité de la communication visuelle au sein de l’enseignement.

2.2.2 Choix des formes et des lignes

Pour concevoir notre interface, nous avons opté pour les formes

Rectangulaires qui symbolisent la stabilité, la vérité et le dynamisme.

Ces formes sont utilisées pour contenir des informations, représentées

2.4 Description de la vue statique

L’animation et encadrer certains objets. Les rectangles sont présentés à

L’horizontale pour produire un effet panoramique.

2.2.3Choix de la typographie

Nous avons choisi une simple typographie pour les attestations demandés par

Les employés tel que : la fiche personnelle

***2.3 Conception détaillée***

La conception détaillée est composée de deux vues à savoir la vue statique et

La vue dynamique. Dans une première partie, nous introduisons une

Modélisation Des diagrammes statiques (i.e., diagramme de classe). Par la suite,

Nous décrivons le comportement des classes et des entités dans les diagrammes

Dynamiques

***2.4 Description de la vue statique***

Le diagramme de classe décrit l’architecture du système. Il représente les

Classes et les relations entre elles. Les figures ci-dessous montrent le diagramme

De classes de notre application.

2.4.1 Diagramme de classe de l’application13

2.4.1 Diagramme de classe de l’application

La Figure 2.3 illustre le diagramme de classe de l’application :

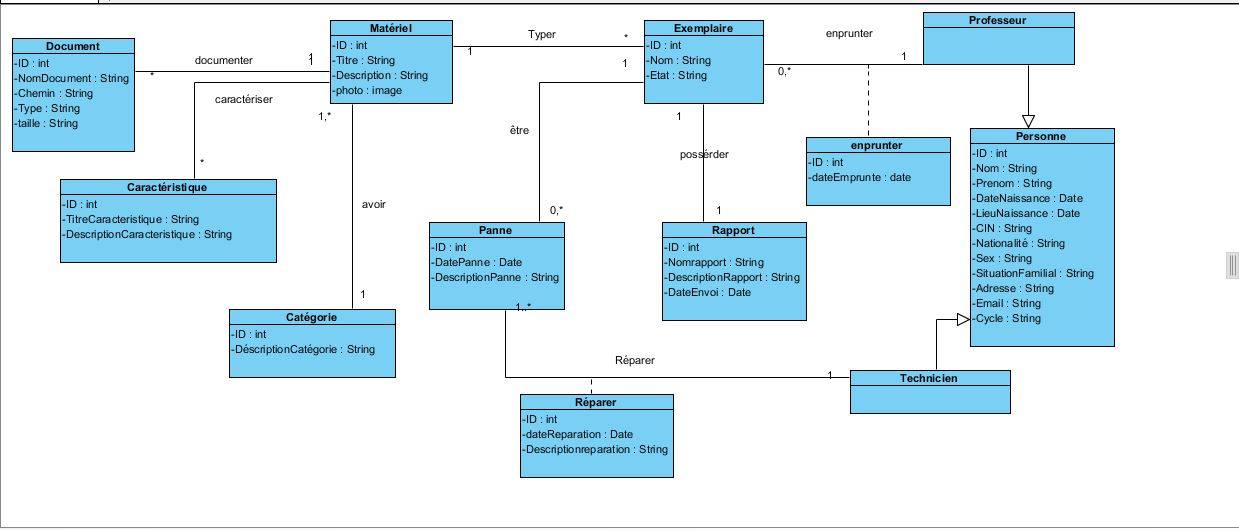


Figure 2.3 Diagramme de classe de l’application

2.4.2 Cas d’utilisation global de l’application

2.4.2 Cas d’utilisation global

La figure 2.4 illustre le cas d’utilisation global de l’application de gestion des

matériel

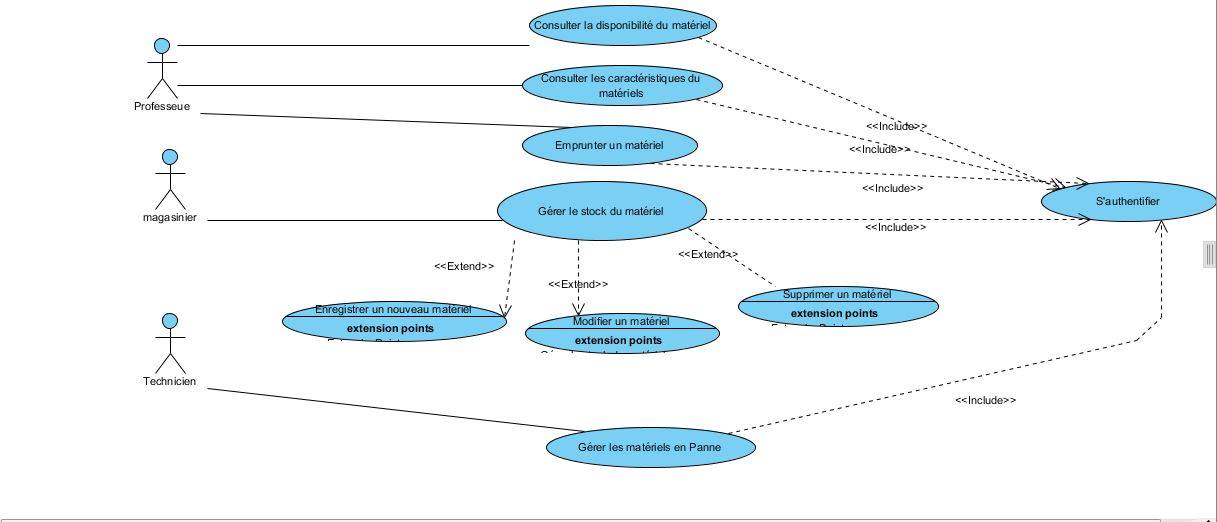


Figure 2.4 Diagramme de cas d’utilisation global de l’application

2.5 Cas d’utilisation relatif à un magasinier

2.5 Cas d’utilisation relatif à un magasinier

La figure 2.5 illustre le cas d’utilisation relatif à un magasinier

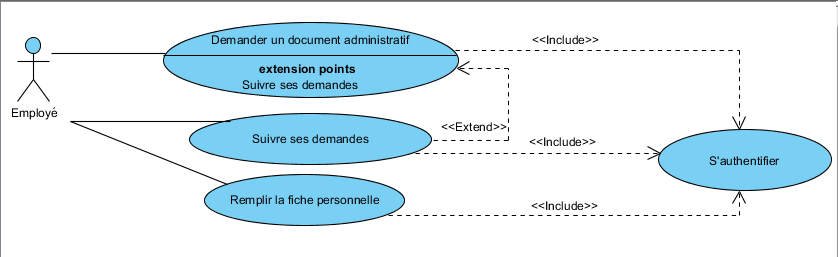


Figure 2.5 Cas d’utilisation relative à un magasinier

Le tableau 2.1 décrit le cas d’utilisation relatif à un magasinier

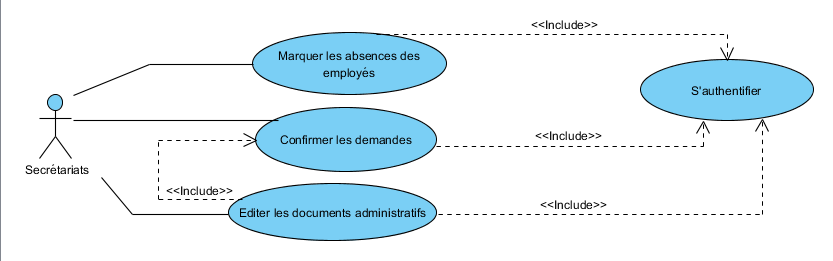
|  |  |
| --- | --- |
| Acteur | magasinier |
| Pré condition | magasinier lance l’application |
| Post-Condition | Magasinier utilise, manipule l’application |
| Principal Scénario | Ajouter un matériel  Supprimer un matériel  Modifier un matériel  Changer l’état de matériel |
| Alternatif | Néant |

Table 2.1 Description du cas d’utilisation global relatif à un magasinier

2.6 Cas d’utilisation relatif au secrétariat

2.6 Cas d’utilisation relatif au technicien

La figure 2.6 illustre le diagramme de cas d’utilisation relatif au technicien

Figure 2.6 Cas d’utilisation relative au technicien

Le tableau 2.2 décrit le cas d’utilisation relatif au technicien

|  |  |
| --- | --- |
| Acteur | Technicien |
| Pré condition | Technicien lance l’application |
| Post-Condition | Technicien utilise l’application |
| Principal Scénario | Gérer les matériels en panne |
| Alternatif | Néant |

Table 2.2 Description du cas d’utilisation global relatif au technicien

2.7 Cas d’utilisation relatif au professeur

2.7 Cas d’utilisation relatif au professeur

La figure 2.7 illustre le cas d’utilisation relatif au professeur

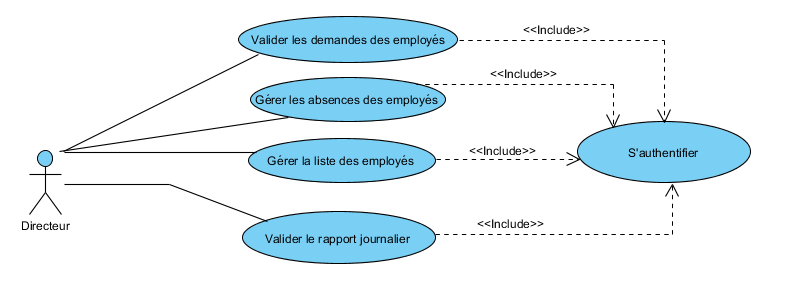


Figure 2.7 Cas d’utilisation relative au professeur

Le tableau 2.3 décrit le cas d’utilisation relatif au professeur

|  |  |
| --- | --- |
| Acteur | Professeur |
| Pré condition | Professeur lance l’application |
| Postcondition | Professeur utilise l’application |
| Principal Scénario | Consulter la disponibilité de matériel  Emprunte un matériel  Consulter les caractéristiques du matériel |
| Alternatif | Néant |

Table 2.3 Description du cas d’utilisation global relatif au professeur

**D’après la réalisation du diagramme de cas d’utilisation qui est pour donner une vision globale du comportement fonctionnel d'un système logiciel.**

**Nous devons montrer les sous actions de quelques cas d’utilisations qui sont vagues pour l’utilisateur.**

**Les cas d’utilisations :**

1. **Consulter la disponibilité du matériel.**
2. **Consulter les caractéristiques du matériel.**
3. **Faire une demande d’emprunte.**
4. **Gérer le Stock des matériels.**
5. **Gérer la disponibilité du matériel.**
6. **Gérer les matériels disparais et dégradés.**
7. **Gérer les matériels en panne.**

2.9 Description de la vue dynamique

Dans ce paragraphe nous présentons une modélisation conceptuelle des

Traitements précisément les diagrammes de séquence qui mettent en œuvre les

Différents objets.

2.9 Description de la vue dynamique

2.10 Les diagrammes de séquence

Un diagramme de séquence présente une collaboration avec une interaction Superposée. En général une séquence porte sur un type spécifique d’action dont La description devrait être renforcée.

**Consulter la disponibilité du matériel :**

**Le professeur a le droit de consulter la disponibilité du matériel, il peut savoir si le matériel est existe a emprunté, ou il est déjà emprunté par un autre professeur .Ce droit facilitera au professeur L’opération de l’emprunt. Car Après cette Consultation et dans le cas Où le matériel est déjà emprunté le professeur ne fera pas les autres mesures.**

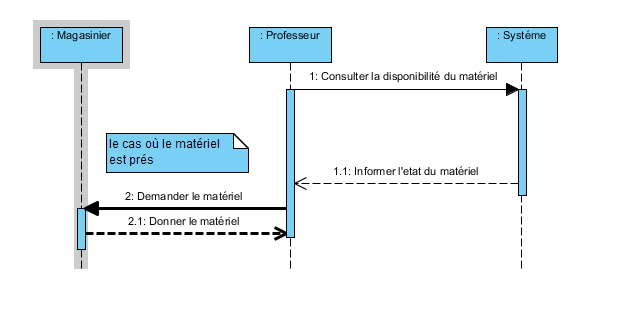
**Consulter les caractéristiques du matériels**

**Chaque professeur peut consulter les caractéristiques d’un ou plusieurs matériels, Ce cas d’utilisation lui permettra de savoir tout les caractéristiques du matériel qu’il veut emprunter.**

**Ces caractéristiques du matériel sont représentées par son numéro d’inventaire, sa spécification, sa quantité, et ces Observations.**

**Faire une demande d’emprunte.**

**L’objectif du diagramme de séquence est de montrer explicitement les interactions pouvant intervenir entre des objets.**

****

**« Figure2 : Diagramme de séquence d’une demande d’emprunte. »**

**Pour emprunter un matériel, le professeur consultera son disponibilité dans le système,**

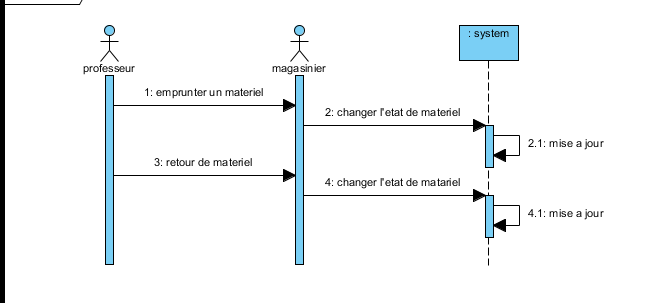
**Après il le demandera dans le cas où il existe.**

**Le magasinier doit le donner.**

**Gérer le stock du matériel :**

**Après la réception des nouveaux matériels par le magasinier, ce dernier s’occupera à insérer toutes les informations « caractéristiques» sur chaque matériel dans la base de données.**

**Gérer la disponibilité du matériel.**

**Puisque ce cas d’utilisation est un peu compliqué, il sera expliqué par un diagramme de séquence**

**« Figure 3 : Diagramme de séquence da la gestion de la disponibilité du matériel ».**

**Après chaque emprunte ou bien retour d’un matériel le magasinier s’occupe à changer**

**Son état dans le système.**

**Gérer les matériels disparais et dégradés .**

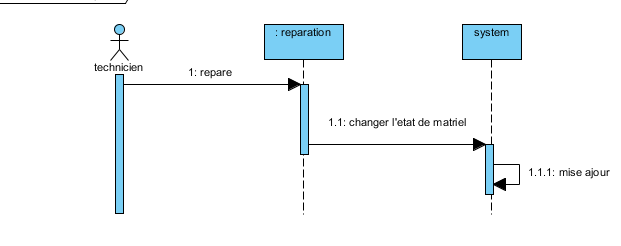
**Après l’emprunt du matériel, le professeur doit le retourner au magasinier.**

**Dans le cas où le matériel est dégradés ; le magasinier l’ajoute aux archives des matériels dégradés.**

**Dans le cas où le matériel est disparais. le magasinier écrit un rapport dans lequel il déclare qu’un matériel est disparait.**

**Gérer les matériels en panne :**

**Après l’emprunt du matériel c\*e dernier peut être en panne .**

****

**« Figure 4 : diagramme de séquence de la gestion des matériels en panne.»**

**Lorsque un matériel tombe en panne, le technicien le réparera après il changera son état dans le système.**

2.17 Conclusion

2.17 Conclusion

Dans ce chapitre, nous avons détaillé les phases de conception : conception

Graphique et conception technique qui nous a permis de dégager la structure de

La base de données afin qu’on puisse réaliser notre solution dans le chapitre

Suivant.

**Chapitre 3 : Réalisation**

**et**

**Test**

***3.1Introduction***

Ce chapitre est consacré à la présentation de l’environnement matériel et

Logiciel utilisés pour le développement de la solution proposé, nous

Expliquerons éventuellement nos choix techniques relatif aux langages de

Programmation et des outils utilisés. Nous donnons ensuite une description

Des résultats aboutis approuvés par quelques imprimes écrans.

* 1. ***Environnement du travail***

Dans cette section, nous décrirons l’environnement du travail.

* + 1. ***Environnement matériel***

L’application a été développée sur la machine possédant les caractéristiques

Suivantes :

|  |  |
| --- | --- |
| Processeur Intel | Centrino duo |
| Mémoire | 1 Go |
| Fréquence d’horloge | 1.73 GHZ |
| Ecran | 15,4 ‘’ |
| Disque dur | 230 Go |

|  |  |
| --- | --- |
| Processeur Intel | Centrino duo |
| Mémoire | 4 Go |
| Fréquence d’horloge | 3 GHZ |
| Ecran | 17,4 ‘’ |
| Disque dur | 640 Go |

3.2 Environnement du travail

* + 1. ***Environnement logiciel***

Pour réaliser notre application de gestion des ressources humaines, nous

avons eu recours aux logiciels suivants :

* Visual Paradigm ( logiciel de création de diagrammes ).
* Eclipse ( Outil de développement d’application ).
* My SQL  (Système de gestion de base de données).

***Scénario d’exécution***

A ce stade, nous présentons notre application à travers les divers imprimes

Écrans réalisés. Nous présentons au début l’interface de la page de menu. En

Effet, l’utilisateur doit choisir s’il est un employé ou bien un administrateur

**Chapitre 4 :**

**Annexes**

4.1 Le formalisme d’UML

UML est un langage graphique de modélisation des données et des

Traitements. C’est une formalisation très aboutie et non propriétaire de la

Modélisation objet utilisée en génie logiciel. L’OMG diffuse depuis Novembre

2007 la version UML 2.1.2, et travaille à présent sur la version 2.2.

* 1. ***Le formalisme d’UML***

Le modèle UML est composé de 13 types de diagrammes (9 en UML 1.3).

UML n’étant pas une méthode, leur utilisation est laissée à l’appréciation de

chacun, même si le diagramme de classes est généralement considéré comme

l’élément central d’UML, des méthodologies, telles que l’Unified-Process, axent

elles l’analyse en tout premier lieu sur les diagrammes de cas d’utilisation (Use

Case). De même, on peut se contenter de modéliser partiellement un système,

par exemple certaines parties critiques. UML se décompose en plusieurs sous-

ensembles Les vues : Les vues sont les observables du système. Elles décrivent

le système d’un point de vue donné, qui peut être organisationnel, dynamique,

temporel, architectural, géographique, logique, etc.

En combinant toutes ces vues il est possible de définir le système complet.

* 1. ***Les diagrammes***

Les diagrammes sont des éléments graphiques. Ceux-ci décrivent le contenu

des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de

plusieurs vues.

4.2 Les diagrammes

* 1. ***Les modèles d’élément***

Les modèles d’élément sont les briques des diagrammes UML, ces modèles

sont utilisés dans plusieurs types de diagramme. Exemple d’élément : cas

d’utilisation, classe, association, etc.

Mise en œuvre d’une démarche à l’aide d’UML : les vues une façon de mettre

en œuvre UML est de considérer différentes vues qui peuvent se superposer

pour collaborer à la définition du système.

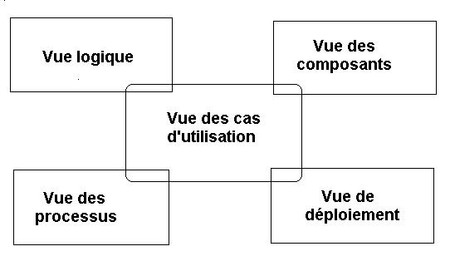
****

Figure 4.1 Les vues

-Vue des cas d’utilisation : c’est la description du modèle « vue » par les acteurs

Du système. Elle correspond aux besoins attendus par chaque acteur.

-Vue logique : c’est la définition du système vu de l’intérieur. Elle explique

Comment peuvent être satisfaits les besoins des acteurs.

-Vue d’implémentation : cette vue définit les dépendances entre les modules.

-Vue des processus : c’est la vue temporelle et technique, qui met en œuvre les

Notions de taches concurrentes, contrôle, synchronisation

4.3 Les modèles d’élément

-Vue de déploiement : cette vue décrit la position géographique et l’architecture

Physique de chaque élément du système

***4.4 Les diagrammes***

Les 13 diagrammes UML sont dépendants hiérarchiquement et se complètent

la hiérarchie des diagrammes UML 2.0 sous forme d’un diagramme de classes.

4.4.1 ***Diagrammes Structurels ou Diagrammes Statiques***

***(cf. Structure Diagram)***

Diagramme de classes : (cf. Class Diagram) : il représente les classes

intervenant dans le système.

Diagramme d’objets : (cf. Object Diagram) : il sert à représenter les instances de

classes (objets) utilisées dans le système.

Diagramme de composants : (cf. Component Diagram) : il permet de montrer les

composants du système d’un point de vue physique, tels qu’ils sont mis en

œuvre (Fichiers, bibliothèques, bases de données…)

Diagramme de déploiement : (cf. Deployment Diagram) : il sert à représenter les

éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage..)

et la manière dont les composants du système sont répartis sur ces éléments

matériels et interagissent avec eux.

Diagramme de paquetages : (cf. Package Diagram) : un paquetage étant un

conteneur logique permettant de regrouper et d’organiser les éléments dans le

modèle UML, le diagramme de paquetage sert à représenter les dépendances

4.4 Les diagrammes

entre paquetages, c’est-à-dire les dépendances entre ensembles de définitions.

Diagramme de structure composite : (cf. Composite Structure Diagram) :

permet de décrire sous forme de boite blanche les relations entre composants

d’une classe.

***4.4.2 Diagrammes Comportementaux***

Diagrammes des cas d’utilisation : (Use Cases) (cf. Use Case Diagram) : il

permet d’identifier les possibilités d’interaction entre le système et les acteurs

(intervenants extérieurs au système), c’est-à-dire toutes les fonctionnalités que

doit fournir le système.

Diagramme états-transitions : (cf. State Machine Diagram) : permet de décrire

sous forme de machine à états finis le comportement du système ou de ses

composants.

Diagramme d’activité : (cf. Activity Diagram) : permet de décrire sous forme de

flux ou d’enchainement d’activités le comportement du système ou de ses

composants.

Diagramme d’interactions ou Diagrammes dynamiques

Diagramme de séquence : (cf. Sequence Diagram) : représentation séquentielle

du déroulement des traitements et des interactions entre les éléments du système

et/ou de ses acteurs.

Diagramme de communication : (cf. Communication Diagram) : représentation

simplifiée d’un diagramme de séquence se concentrant sur les échanges de

messages entre les objets.

4.5 Standardisation et Certification UML

Diagramme global d’interaction : (cf. Interaction Overview Diagram) : permet

de décrire les enchainements possibles entre les scénarios préalablement

identifiés sous forme de diagrammes de séquences (variante du diagramme

d’activité)

Diagramme de temps : (cf. Timing Diagram) : permet de décrire les variations

d’une donnée au cours du temps. Les modèles d’éléments.

***4.5 Standardisation et Certification UML***

UML n’est pas un standard de fait, mais un standard « industriel » de l’OMG

(novembre 1997). Ceci étant, vu le succès initial de ce langage, il aurait pu tout

aussi bien être simplement « standard de fait ». Depuis juillet 2005, la première

Version 2.\* de UML est validée par l’OMG. Par ailleurs, depuis 2003, l’OMG a

mis en place un programme de certification à la pratique et la connaissance

d’UML : OCUP ( OMG Certified UML Professional)

* 1. ***Logiciels de modélisation UML***

Il existe de nombreux outils logiciels de modélisation UML.

Malheureusement aucun d’entre eux ne respectent strictement aucune des

versions d’UML, particulièrement UML2 : beaucoup de ces outils introduisent

des notations.

4.6 Logiciels de modélisation UML 57

|  |  |
| --- | --- |
| Diagramme de cas d’utilisation | Spécification, cahier des charges |
| Diagramme de séquence |  |
| Diagramme d’objet | Conception Architecturale |
| Diagramme d’activité |  |
| Diagramme de classe |  |
| Diagramme de communication |  |
| Diagramme de déploiement |  |
| Diagramme de composant |  |

Table 4.1 Structure des diagrammes

Particulières non conformes, très peu supportent les différents types de

diagrammes définis par le standard. Beaucoup en revanche incluent des outils de

génération de squelette de code, particulièrement à partir du diagramme de

classes, qui sont ceux qui se prêtent le mieux à une telle automatisation. Les outils

plus réputés, à défaut d’être conformes, sont les outils d’openModelSphere.

**Bibliographie**