# Chapitre 2

# Analyse et specification des besoins

## Introduction

Dans ce chapitre nous allons analyser le projet. Où nous allons résoudre la complexité du problème tout en citant une étude de cas et en précisant les besoins fonctionnels et non fonctionnels de l’application. Ces besoins seront mis en relief, par la suite, en se basant sur les diagrammes des cas d’utilisations qui servent, essentiellement, à définir d’une manière assez rigoureuse les fonctionnalités de l’application.

## Analyse de l’existant

Cette section présente un état des lieux : il s’agit d’une étude de l’existant suivie de critiques permettant au projet de présenter une amélioration résumant l’ensemble des solutions retenues.

### II.1.Étude de la procédure actuelle

### Pour que le chef d’une bureautique informatique puisse réussir les services techniques offerts aux clients, il doit passer par plusieurs étapes afin de satisfaire le besoin visé :

-chercher une méthode pour acquérir la confiance de son client

-organiser et diviser les tâches entre les différents membres de groupe dans la société

-Trouver une solution pour convaincre ses clients des prix demandés et des types de problème à chaque réparation

La réalisation de ses différentes étapes n'est pas assez facile puisque la majorité des directeurs des sociétés ne réussissent pas à les mettre en place .Ces étapes sont pénibles et prennent du temps.

### II.2.Critique de l’existant

L’étude de l’existant nous a permis de dégager un certain nombre de lacunes :

- la réparation d’un matériel pour un client prend largement du temps et en contrepartie le temps est une contrainte pour la réparation d’un matériel.

- la confiance du client est très importante pour la réputation de la société, il faut la soigner (réparer le matériel avant ou dans la période demandée)

- la bonne organisation et division des taches entre les différents membres de la société est une contrainte forte afin de produire un bon résultat.

### II.3. Solution proposée

Dans un souci de se limiter aux contraintes mentionnées ci-dessus on est appelé à concevoir une application qui rend la tâche de gestion des services techniques plus facile et non ennuyante.II

Le travail demandé se résume ainsi dans les fonctionnalités suivantes :

* Pour l’administrateur :
* S'inscrire
* S’authentifier
* Consulter l’historique de réparation des matériels
* Ajouter un commercial
* Ajouter un technicien
* Voir l’état de matériel
* Pour le technicien :
* s'authentifier
* générer une fiche de réparation de matériel
* changer l’état de matériel
* Voir l’état de matériel
* Pour le commercial :
* S'authentifier
* Générer un reçu lors de recevoir de matériel
* Voir l’état de matériel
* Pour le client :
* Demande la réparation des matériels
* Voir l’état de matériel

## Spécification des besoins

Le cycle de vie de tout projet informatique doit préalablement passer par la phase de spécification des besoins. Il s’avère indispensable d’appréhender l’étendue des besoins fonctionnels et non fonctionnels relatifs à notre projet.

Dans ce chapitre, nous présenterons les objectifs de notre application, ce qui nous amène à identifier les possibilités du système et les besoins des utilisateurs

## **III.1.Spécification informelle**

L’application envisagée doit satisfaire les besoins fonctionnels qui seront exécutés par le système

et les besoins non fonctionnels qui perfectionnent la qualité logicielle du système.

### III.1.1.Les besoins fonctionnels:

L’étape de spécification des besoins fonctionnels est nécessaire pour préciser les objectifs primordiaux de la future application. Ils conduisent à l’élaboration des modèles des cas d’utilisation.

Les fonctionnalités à prendre en compte sont :

* Offrir aux utilisateurs un espace pour remplir un formulaire par des informations personnel.
* Offrir la possibilité de consulter l’état de matériel par chaque utilisateur.
* Offrir la possibilité de Créer une fiche de matériel par le commercial.
* Offrir la possibilité de Changer l’état de matériel par le technicien.
* Offrir la possibilité de Créer des utilisateurs (technicien et commercial) par l’administrateur.
* Offrir la possibilité de créer un reçu de réparation de matériel par le technicien.

### III.1.2.Les besoins non fonctionnels

Le système doit répondre à certains besoins qui ne sont pas indispensables pour son fonctionnement mais qui sont importants pour améliorer la qualité de ses services tels que :

* une interface d’inscription intuitive et claire.
* Un affichage de message d’erreur claire et compréhensible pour toutes les anomalies détectées.
* Une sécurisation des informations de la base de données.
* La réalisation de ce projet doit mettre en œuvre des outils performants et efficaces : système de gestion de base de données performant et langage de programmation orienté objet.

**D’autres contraintes devraient être prise en compte telles que :**

* Le temps de réponse à une requête ne doit pas dépasser quelques secondes.
* L’utilisation et le parcours d'application doivent être faciles pour les différents utilisateurs.

### III.2.Spécification semi formelle

Pour la phase de spécification et de conception, nous avons choisi, comme langage de modélisation, le langage UML (langage de modélisation objet unifié).

En effet, UML permet de modéliser une application selon une vision objet. Il possède plusieurs facettes. C’est une norme, un langage de modélisation objet, un support de communication.

### Diagramme de cas d'utilisation :

Le diagramme de cas d’utilisation est une technique de description du système étudié privilégiant les points de vue de l’utilisateur. Il est composé d’un ensemble d’actions déclenchées par acteur externe produisant un résultat identifiable. Ils représentent la relation entre un acteur et une fonctionnalité du système. Il permette, notamment, de représenter les processus d’un domaine.

Les acteurs intervenant dans l’application sont l’administrateur, le commercial, technicien et le client.

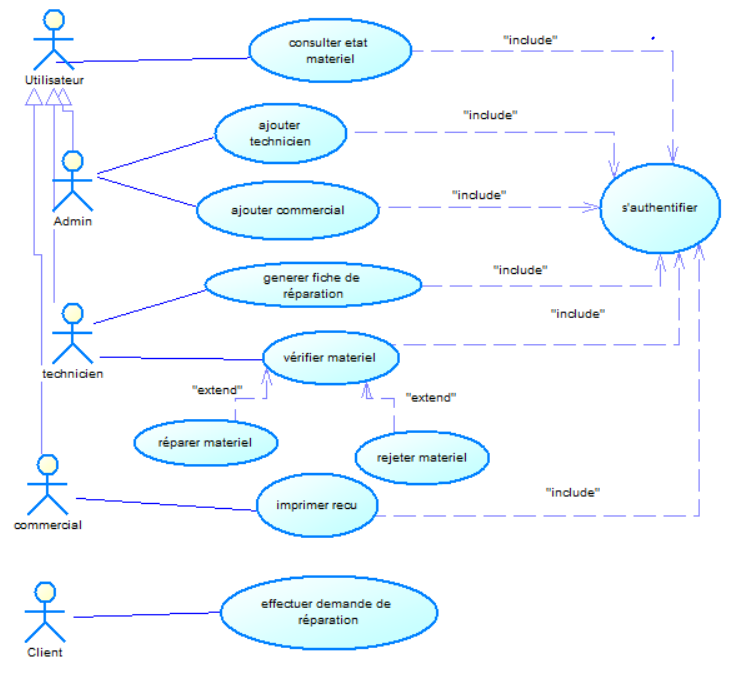


Figure 1:Diagramme de cas d'utilisation

**Acteurs :**

* Utilisateur : peut être un administrateur, technicien ou commercial, il peut consulter les matériels.
* Admin : un utilisateur qui va ajouter un technicien ou commercial.
* Technicien : un utilisateur qui générer une fiche de réparation et réparer le matériel.
* Commercial : un utilisateur qui va imprimer un reçu pour le client.
* Client : va demander la réparation de son materiel.

## Conclusion

Nous avons cité précédemment les besoins de notre application, qui seront dans le chapitre suivant représenter sous forme des diagrammes.

# Chapitre 3

# Conception

## Introduction

La conception est l’étape la plus importante d’un projet informatique, puisqu’elle permet de définir les composantes d'un système et leurs relations fonctionnelles.

Ce chapitre sert, donc, à présenter mon choix conceptuel, décrire les différents attributs et méthodes de l’application et expliquer brièvement le rôle de chaque composant à travers le diagramme de classe.

## Diagrammes de classe

Le diagramme de classes exprime la structure statique du système. Ils décriventl’ensemble des classeset leurs associations*.* Une classe décrit un ensemble d’objets (instances de la classe). Une association exprime une connexion sémantiquebidirectionnelle entre classes. Elle décrit un ensemble de liens(instancesde l’association). Le rôledécrit une extrémité d’une association. Les cardinalités (oumultiplicités) indiquent le nombre d’instances d’une classe pour chaque instance del’autre classe.

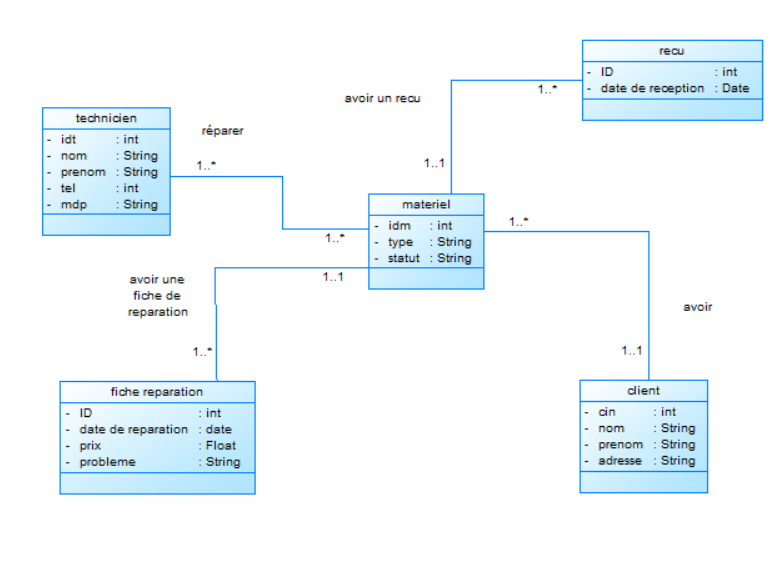


Figure 2:Diagramme de classe

**Descriptions de diagrammes des classes :**

* **Ce diagramme est constitué de sept classes :**
* Fiche de réparation
* Reçu
* Client
* Matériel
* Technicien
* **Les différents rôles de chaque classe :**

* Un matériel peut avoir un ou plusieurs reçus.
* Un ou plusieurs techniciens peuvent réparer un ou plusieurs matériels.
* Un matériel peut avoir un ou plusieurs fiches de réparation
* Un client peut avoir un ou plusieurs matériels.

## Conclusion

Dans ce chapitre, nous avons fixé l’architecture de l’application web à développer. Pour cela nous avons recourir au diagramme de cas d'utilisation et diagramme de classes.

Ceci sert à la réalisation de l'application web et à la conception de son interface qui sera détaillée dans la partie suivante du présent rapport.

# Chapitre 4

# Réalisation:

## Introduction

Après avoir achevé l’étape de conception de l’application web, nous entamons dans ce chapitre la partie réalisation qui constitue le dernier volet de ce rapport et qui a pour objectif d’exposer le travail réalisé.

Pour ce faire, je vais commencer, tout d’abord, par préciser l’environnement matériel et logiciel de mon travail. Ensuite je vais présenter le travail accompli tout au long de ce projet pour enchaîner avec une exposition des problèmes rencontrés.

## Environnement de développement

### II.1. Environnement materiel

Lors de la réalisation de cette application, J'ai utilisé un ordinateur ayant les caractéristiques ci-dessous :

|  |  |
| --- | --- |
|  | **Poste** |
| **Marque** | HP Pavilion g6 |
| **Processeur** | Intel Core i5 |
| **Disque dur** | 750Go |
| **Mémoire vive** | 8,00 GO |
| **Système d’exploitation** | Windows 8 |

### II.2. Environnement logiciel

### Pour la conception :

* **Power AMC**: C’est une puissante solution de modélisation des systèmes d’information. Cet ensemble d'outils supporte plusieurs techniques de modélisation standard : modélisation Merise (données et traitements), modélisation UML particulièrement adaptée à la logique des applications et modélisation des processus métiers dédiée aux non informaticiens pour leurs faciliter l'expression des besoins. L'originalité de la solution tient dans la forte intégration proposée entre ces différentes techniques.

### Pour le développement :

* Environnement de développement :
* **Microsoft Visual Studio** : est une suite de logiciels de développement pour Windows et MacOs conçue par Microsoft. La dernière version s'appelle **Visual Studio 2017**. Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles.
* Système de gestion de base de données :
* **Microsoft SQL Server** est un système de gestion de base de données (SGBD) en langage SQL incorporant entre autres un SGBDR (SGBD relationnel ») développé et commercialisé par la société Microsoft. Il fonctionne sous les OS Windows et Linux (depuis mars 2016), mais il est possible de le lancer sur Mac OS via Docker, car il en existe une version en téléchargement sur le site de Microsoft.

## III. Présentation de l'application

Pour présenter mon application, je vois les imprimes écrans et les traitements relatifs.

### propos

* Interface accueil de l’application **:**

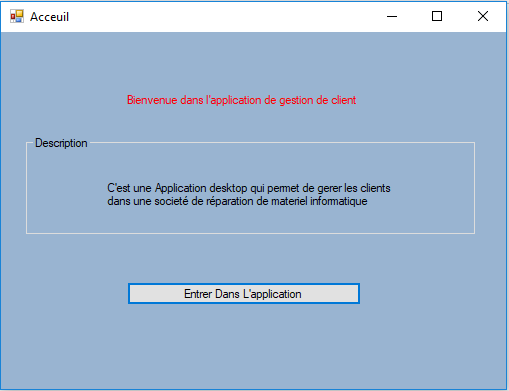


Figure 3:Accueil

* **Interface de navigation**: qui utilise le Gridview pour afficher une interface correspondante lorsque nous choisissons un menu.

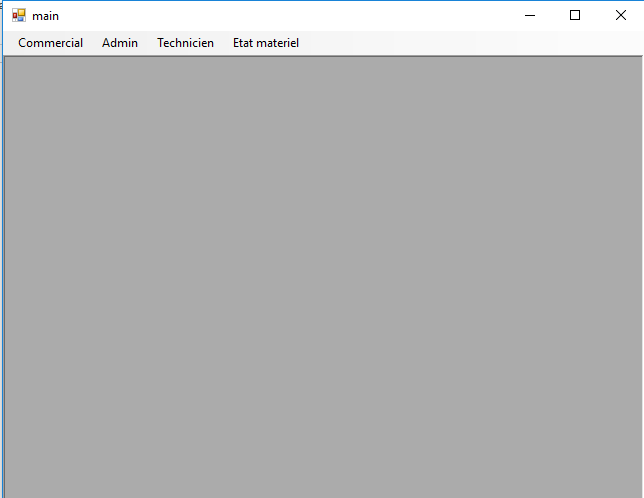
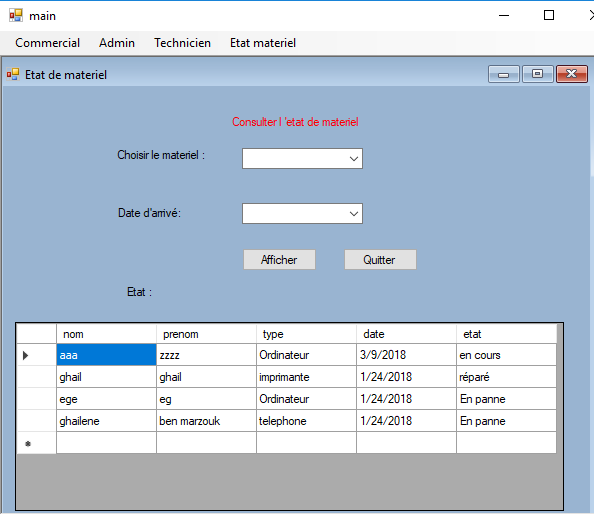


Figure 4:Fenetre d'affichage de forme

Cette interface permet de consulter l’état de matériel :



* **Interface ADMIN :**

Cette interface permet à l’administrateur de s’authentifier

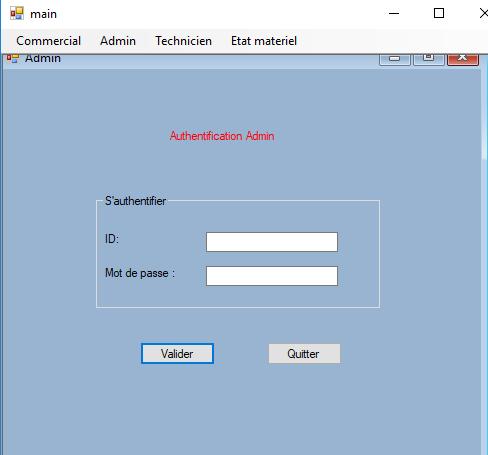


Figure 5:Interface authentification administrateur

Cette interface permet d’ajouter un commercial ou un technicien

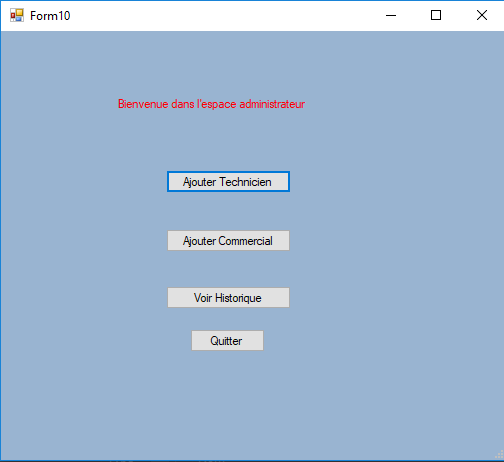


Figure 6:interface de gestion pour l' administrateur

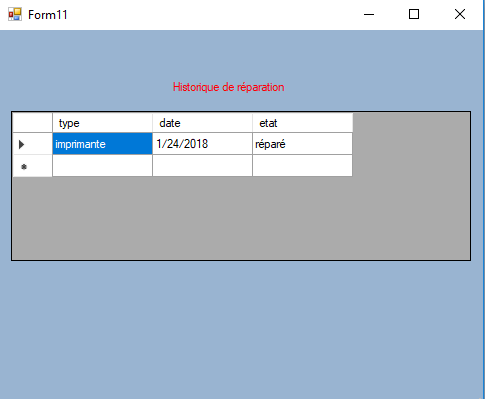


Figure 7:interface Historique de réparation

* **Interface Commercial :**

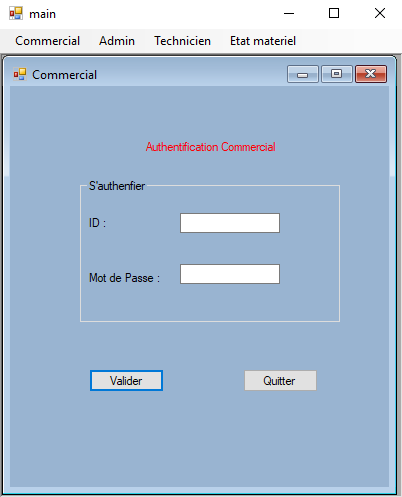


Figure 8:Interface authentification pour le commercial

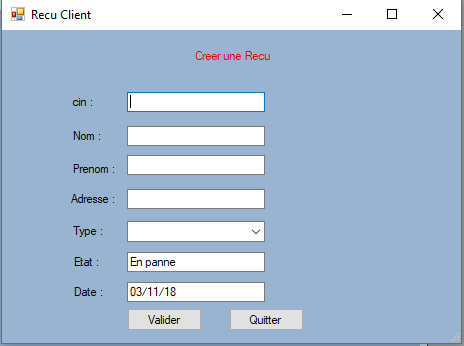


Figure 9:interface creation recu

Cette interface permet au technicien de changer l’état des matériels et de générer une fiche de reparation.

