

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Licence en Informatique

Thème

**Système
d'enquête sur les crimes de la cybercriminalité**

Réalisé par :

- **Moussa Haitem**
- **Ailas Fouad**

Présenté le 27 mai 2015 devant la commission d'examination composée de MM.

- *Merzoug M.* (Encadreur)
- *Smahi M. I.* (Examineur)
- *Benmouna Y.* (Examineur)

Année universitaire : 2014-2015

Remerciements

Mes remerciements vont à l'endroit de tous ceux qui ont contribué à la réussite de ce travail, en particulier à :

nous remercier le bon ALLAH, qui m'a donné la force, la volonté et le courage pour terminer ce modeste travail;

A tous mes parents, amis et collègues qui m'ont aidé dans l'anonymat

Mon épouse et mon enfant pour leur engagement, les divers sacrifices et leur soutien ;

· Mes frères et ma sœurs : pour son dévouement et sa collaboration ;

· Mes collaborateurs de service, ;

· tous les professeurs de l'université Abou Bekr Belkaid Tlemcen qui ont contribué à notre formation ;

Je tiens à exprimer toute ma gratitude à mon encadreur Mr MERZOUG Mohammed de l'université Abou Bekr Belkaid Tlemcen pour son encadrement, son écoute, ses élucidations, ses conseils, ses directives et encouragements qu'il m'a afflués.

Tout simplement à tous ceux et celles qui méritent mes remerciements

DEDICACE

Je dédie ce travail de fin d'études à :

Ma mère et mon père qui, grâce à sa collaborations, sa déterminations et m'on soutenues au prix de sacrifices inoubliables ;

· ma femme et mon enfant, qui me soutiennent au prix de mille et une privations dans mon engagement pour le développement ;

· A Dieu à qui j'adresse mes remerciements.

Table des matières

CHAPITRE I généralité Généralités surcybersécurité.....	04
I. Notion de base sur le client/serveur :.....	05
I.1 Cyber sécurité et cybercriminalité.....	05
I.2 Accès illégal (piratage, craquage).....	06
CHAPITRE II Etude technique.....	
I. Présentation de L'UML.....	08
II. Le langage de modélisation UML.....	08
III. Définition des diagrammes utilisés.....	08
III.1.Diagramme de cas d'utilisation.....	08
III.2.Diagramme de séquence.....	09
III.3.Diagramme de classes.....	09
IV. Présentation des diagrammes.....	09
IV.1Diagramme de cas d'utilisateur.....	10
IV.2.Diagramme de séquence	10
IV.3.Diagramme de classe.....	11
V. Présentation des tables de la base de données.....	12
V.1.1.Structure de la table plant.....	13
V.1.2.structure de la table plagnat.....	13
V.1.3.structure de la table poste police.....	13
V.1.4.Structure de la table témoins.....	13
V.1.5.Structure de la table suspects	13

V.1.6.structure de la table Arrête.....	14
V.1.7.Structure de la table Evidence	14
V.1.8.Structure de la table Propriétés.....	14
V.1.9.Structure de la table Enquêteur.....	14
VI. Conclusion.....	15
CHAPITRE III Réalisation	
I. Notion de base sur Microsoft Visual Studio Express.....	17
I.1.Introduction.....	17
I.2.Programmation C sharp.....	17
I.2.1.Caractéristiques partagées.....	17
I.2.1.1.Caractéristiques partagées avec le langage Java.....	18
I.2.1.2.Caractéristiques partagées avec le langage C++.....	18
I.2.1.3.Caractéristiques partagées avec d'autres langages.....	18
I.3.Exemple sur Le code source	19
II. Présentation de quelque pages du logiciel	23
III. conclusion.....	26
Conclusion générale	28
Bibliographie.....	29

Liste des figures

Chapitre I GENERALITE

Figure 1 : Réseaux affecté par la cybercriminalité.....[5]

Figure2 : Le logo montre que le site Internet a été piraté. L'auteur a modifié la page[6]

Chapitre II : Etude technique

Figure 3 : Diagramme de cas d'utilisateur.....[10]

Figure 4 : Diagramme de séquence coté policier.....[10].

Figure 5 : Digramme de classe[11]

Figure 6 : Diagramme de navigation.....[11] .

Chapitre III :Réalisation

Figure 7 : fiche admin[23]

Figure 8 : fiche global de logiciel[23]

Figure 9 : fiche des détail de plaignant.....[24]

Introduction

générale

Internet est devenu un outil précieux pour les gouvernements, les entreprises et les particuliers algériens. Tout comme le cyberspace est en constante évolution, il en est de même des cybermenaces à notre sécurité, prospérité et qualité de vie. Notre gouvernement est déterminé à protéger les algériens contre la menace de la cybercriminalité. La cybercriminalité est généralement définie comme une infraction criminelle ayant l'ordinateur pour objet (piratage informatique, hameçonnage, pollupostage) ou pour instrument de perpétration principal (pornographie juvénile, crime haineux, fraude informatique). Des criminels peuvent également utiliser des ordinateurs à des fins de communication et de stockage de documents ou de données. Il s'agit donc d'une nouvelle forme de criminalité et de délinquance qui se distingue des formes traditionnelles en ce qu'elle se situe dans un espace virtuel, le « cyberspace ». Depuis quelques années la démocratisation de l'accès à l'informatique et la globalisation des réseaux ont été des facteurs de développement du cybercrime.

Chapitre I

Généralités

Sur cybersécurité

I. Notion de base sur le client/serveur :

I.1. Cybersécurité et cybercriminalité :

La cybersécurité joue un rôle essentiel dans le développement des technologies de l'information et des services en ligne³³. Pour garantir leur sécurité et leur bien-être économique, tous les pays doivent absolument renforcer la cybersécurité (et la protection des internautes) et protéger les infrastructures essentielles de l'information, objectif qui préside aujourd'hui au développement des nouveaux services, mais aussi à l'élaboration des politiques gouvernementales³⁴. La prévention de la cybercriminalité fait partie intégrante de toute stratégie nationale de cybersécurité et de protection des infrastructures essentielles de l'information

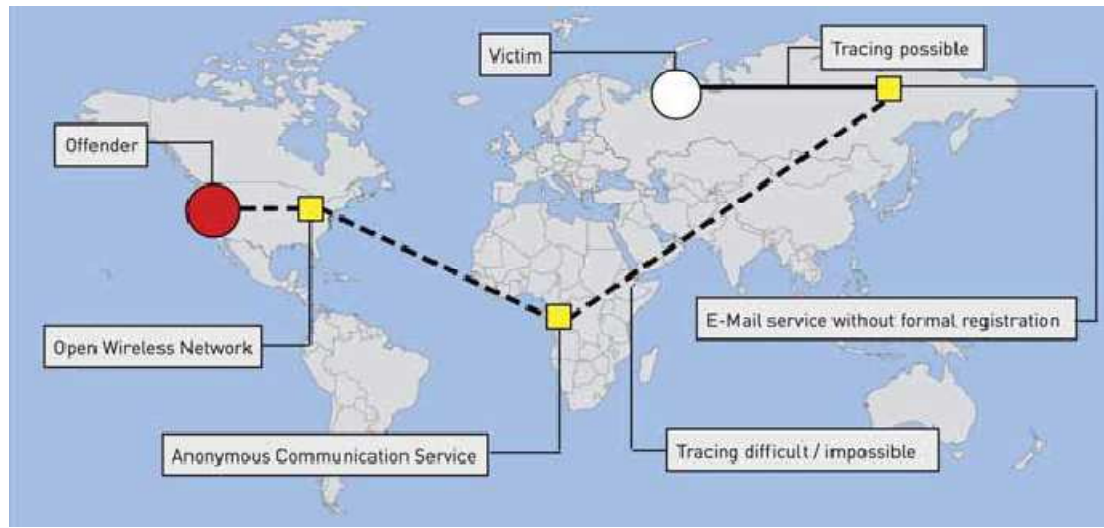


Figure 1 : Réseaux affecté par la cybercriminalité.

I.2. Accès illégal (piratage, craquage)

Le "piratage" (*hacking*) désigne l'accès illégal à un ordinateur. C'est l'une des infractions informatiques les plus anciennes1 ver le développement des réseaux informatiques (notamment d'Internet), cette infraction est devenue un phénomène de masse.

Certaines organisations bien connues ont été victimes de piratage. Ainsi la NASA (*United States National Aeronautics and Space Administration*), l'armée de l'air des Etats-Unis, le Pentagone, Yahoo!, Google, Ebay et l'administration allemande. Les exemples suivant illustrent quelques infractions entrant la catégorie du "piratage":

- craquage d'un mot de passe ou de sites Internet protégés par mot de passe;
- contournement d'une protection par mot de passe sur un ordinateur.

On peut citer quelques exemples d'actes préparatoires:

- exploitation d'une faille logicielle ou matérielle pour obtenir illégalement un mot de passe permettant d'entrer dans un système informatique;
- création de sites Internet d'"espionnage"(*spoofing*) conçus pour amener les utilisateurs à révéler leur mot de passe;
- installation de matériels ou de logiciels d'enregistrement de frappe (par exemple, "enregistreurs de frappe" ou *keyloggers*), qui enregistrent toutes les frappes au clavier et, par conséquent, tous les mots de passe saisis sur l'ordinateur et/ou le dispositif.

Tous les auteurs d'infraction n'ont pas les mêmes motivations. Certains contournent des mesures de sécurité dans l'unique but de montrer ce dont ils sont capables



Figure2 : Le logo montre que le site Internet a été piraté. L'auteur a modifié la page

Chapitre II

Etude Technique

I. Présentation de l'UML

Face à la diversité des formalismes utilisés par les méthodes d'analyse et de conception objet, UML (*Unified Modified Language* « langage de modélisation objet unifié ») représente un réel facteur de progrès par l'effort de normalisation.

En effet, UML est issu de la fusion de trois méthodes qui ont le plus influencé la modélisation objet au milieu des années 90 : Booch Grady Booch, OMT (*Object Modelling Technique*) de James Rumbaugh et OOSE (*Object Oriented Software Engineering*) d'Ivar Jacobson. UML est à présent un standard défini par l'OMG (*Object Management Group*).

II. Le langage de modélisation UML

L'UML est un langage graphique de modélisation des données et des traitements, fondé sur des concepts orientés objets. UML propose de décrire un système à l'aide de 9 diagrammes:

- Diagrammes de cas d'utilisation
- Diagrammes d'objets
- Diagrammes de classes
- Diagrammes de composants
- Diagrammes de déploiement
- Diagrammes de collaboration
- Diagrammes de séquence
- Diagrammes d'états-transitions
- Diagrammes d'activités

Ces diagrammes, d'une utilité variable selon les cas, ils ne sont pas nécessairement tous produits à chaque modélisation. Pour le cas de mon application, les diagrammes utilisés sont les diagrammes de cas d'utilisation, de séquence et de classes.

III. Définition des diagrammes utilisés

III.1. Diagramme de cas d'utilisation

Un cas d'utilisation définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles. D'après Bittner et Spence, « Un cas d'utilisation, défini simplement, permet de décrire une séquence d'événements qui, pris tous ensemble, définissent un système

faisant quelque chose d'utile ». Chaque cas d'utilisation contient un ou plusieurs scénarios qui définissent comment le système devrait interagir avec les utilisateurs (appelés acteurs) pour atteindre un but ou une fonction spécifique d'un travail. Un acteur d'un cas d'utilisation peut être un humain ou un autre système externe à celui que l'on tente de définir.

III.2.Diagramme de séquence

Le diagramme de séquences permet de cacher les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent par le biais de messages

III.3.Diagramme de classes

Le diagramme de classes est un schéma utilisé en génie logiciel pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Ce diagramme fait partie de la partie statique d'UML car il fait abstraction des aspects temporels et dynamiques.

Une classe décrit les responsabilités, le comportement et le type d'un ensemble d'objets. Les éléments de cet ensemble sont les instances de la classe.

Une classe est un ensemble de fonctions et de données (attributs) qui sont liées ensemble par un champ sémantique. Les classes sont utilisées dans la programmation orientée objet. Elles permettent de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.

Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence des relations de parenté. D'autres relations sont possibles entre des classes, chacune de ces relations est représentée par un arc spécifique dans le diagramme de classes.

Elles sont finalement instanciées pour créer des objets (une classe est un *moule à objet* : elle décrit les caractéristiques des objets, les objets contiennent leurs valeurs propres pour chacune de ces caractéristiques lorsqu'ils sont instanciés).

IV. Présentation des diagrammes

IV.1. Diagramme de cas d'utilisateur :

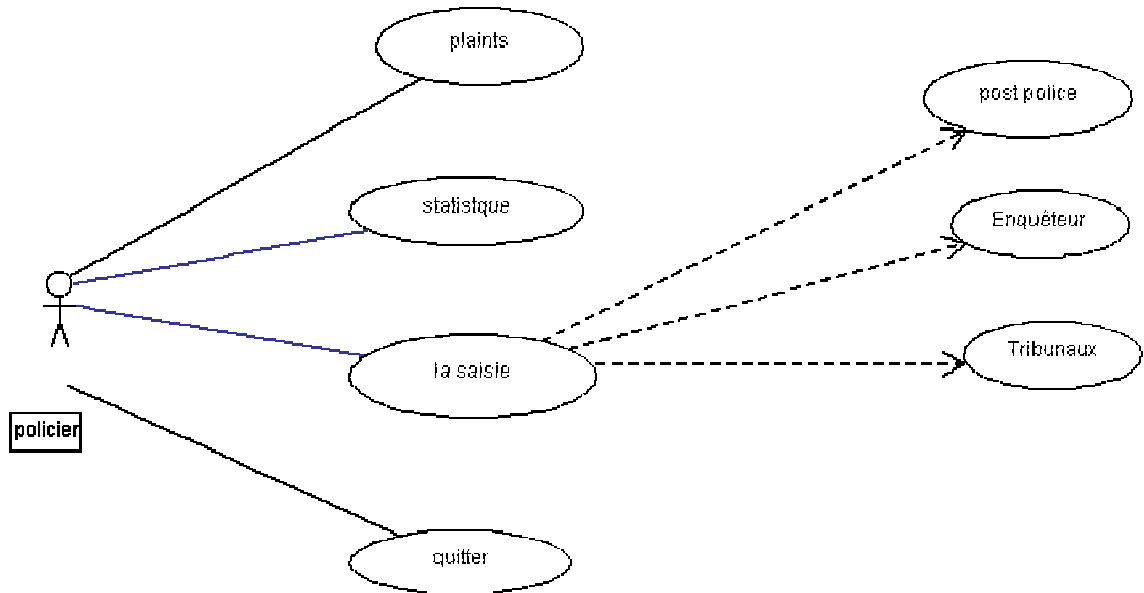


Figure 3 : Diagramme de cas d'utilisateur

IV.2. Diagramme de séquence :

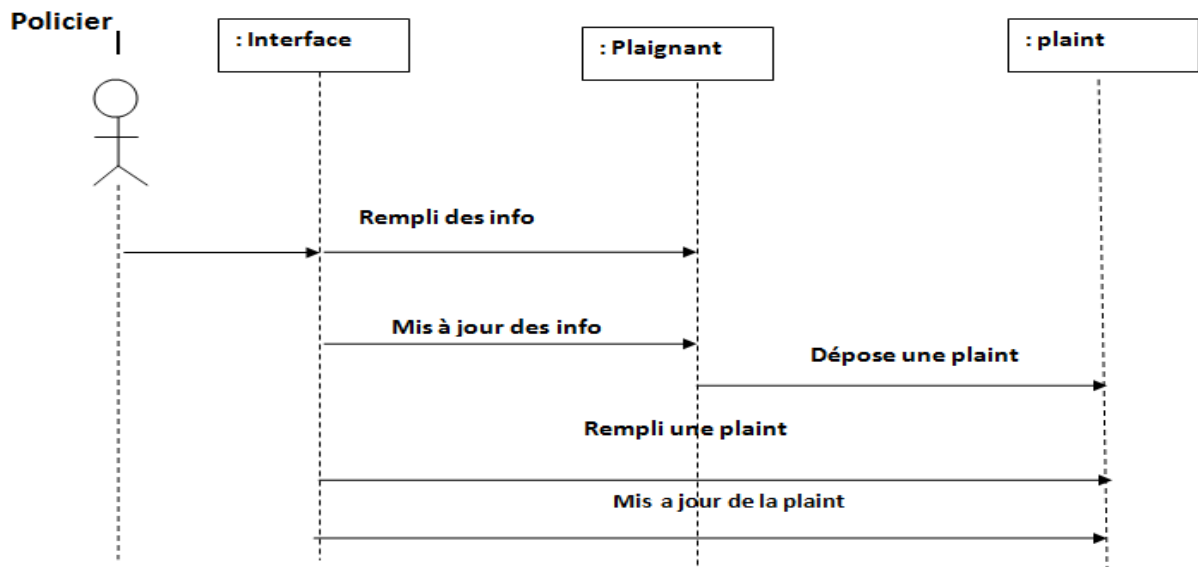


Figure 4: Diagramme de séquence coté policier

IV.3. Diagramme de classe

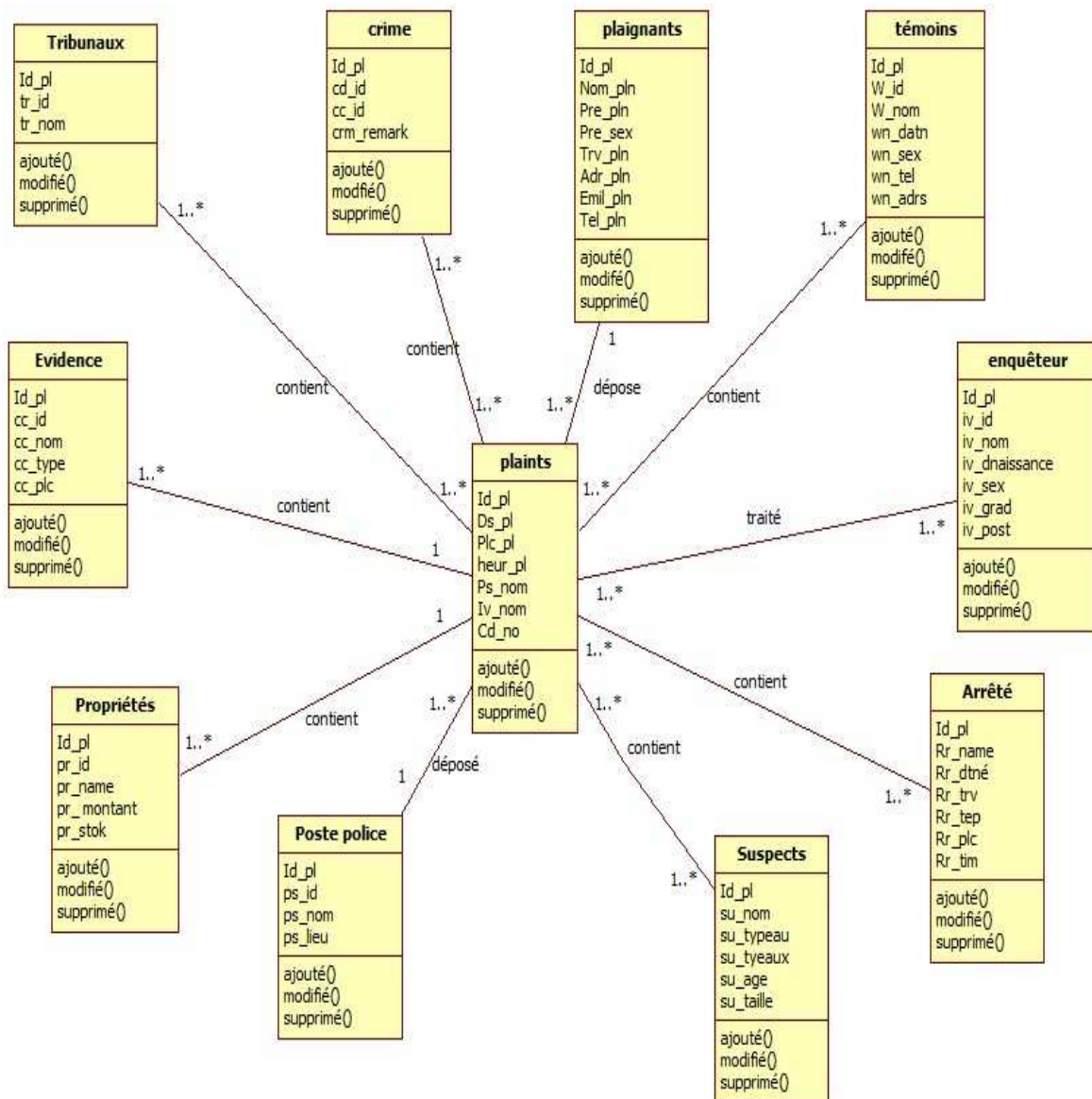


Figure 5 : Diagramme de classe

V. Présentation des tables de la base de données

Structure de la table `plaint`

```
` Id_pl `reel(10)
`Name `string(50)
`Dat-n `date(10)
`Time `time(10)
`Ps_nom `string(50)
`Iv_nom `string(50)
`Cd_no `string(50)
```

Structure de la table ``plaignant``

```
` Id_pl `reel(10)
`Nom_pln string(50)
`Pre_pln string (50)
sex_plnstring (10)
`Trv_pln string(50)
`Adr_pln verchar(50)
`Emil_pln verchar (20)
Tel_pln reel (20)
```

Structure de la table `Poste police``

```
` Id_pl reel(10)
`ps_id reel(10)
`ps_nom string(20)
`ps_lieu verchar(50)
```

Structure de la table ``témoins`

```
`W_id string(10)
`W_nom string(20)
`wn_datn date(10)
`wn_sex string(10)
`wn_tel reel(10)
`wn_adrs verchar(50)
```

Structure de la table ``Suspects``

```
` Id_pl reel(10)
`su_nomstring(20)
`su_typeausting(20)
`su_tyeauxstring(20)
```

```
`su_age reel(10)
`s_u_taille intger(10)
```

Structure de la table `Arrêté

```
`Id_pl reel(10)
`Rr_name string(10)
`Rr_dtné date(20)
`Rr_trv string(20)
`Rr_tepreel(10)
`Rr_plc string(50)
`Rr_tim time(10)
```

Structure de la table ` Evidence

```
`Id_pl date(10)
`cc_id reel(10)
`cc_nom string(20)
`cc_type string(20)
`cc_plc string(20)
```

Structure de la table ` Propriétés

```
`Id_pl reel(10)
`pr_id reel(10)
`pr_name string(20)
`pr_montant reel(20)
`pr_stok reel(20)
```

Structure de la table ` enquêteur

```
`Id_pl reel(10)
`iv_id reel(10)
`iv_nom string(20)
`iv_dnaissance date(10)
`iv_sex string(10)
`iv_grad string(20)
`iv_post string(20)
```

VI. Conclusion

Dans ce chapitre, on a concentré sur les aspects analytique et conceptuel de notre application ainsi que les bases de données qui interagissent avec elle. La phase conceptuelle est une étape fondamentale pour la réalisation de n'importe quel projet.

Pour cela, on a défini le cas d'utilisations puis, sa traduction au travers de la construction des diagrammes de séquence, de classe et de navigation.

Enfin, on a défini toutes les tables des bases de données utilisées.

Ainsi à cet instant précis, les choses sont plus claires et l'implémentation de l'application s'en trouve facilitée.

Le chapitre suivant sera consacré à l'implémentation de notre application.

Chapitre III

Réalisation

I. Notion de base sur Microsoft Visual Studio Express

I.1. Introduction:

Microsoft Visual Studio Express est un ensemble d'environnements de développement intégrés gratuits développé par Microsoft. Il s'agit d'une version allégée de Microsoft Visual Studio. L'idée de ces éditions "express" est, selon Microsoft, de fournir un environnement de développement facile à utiliser et à apprendre pour des jeunes ou des passionnés.

La première version de Visual Studio Express (2005) est sortie le 7 novembre 2005 et était supposée n'être gratuite que pour un an (mis à part SQL Server 2005 Express Edition). Toutefois, Microsoft a annoncé le 19 avril 2006 que ces éditions resteraient gratuites sans limite de date¹.

Visual Studio Express est composé de ces différents produits, chacun supportant un seul langage : Visual Basic, C#, C++, J#, Web Developer et SQL Server. Visual J# n'est plus disponible dans Visual Studio Express 2008. L'installation passe par le téléchargement d'un logiciel de moins de 3 Mo qui téléchargera et installera la version de Visual studio choisie (356 Mo pour Visual Basic). Le choix d'installation du répertoire par défaut est défini lors de la première installation d'une des versions choisie (version, langage, internationalisation).

I.2. Programmation C sharp :

Le langage de programmation C# (C dièse en français, ou prononcé C-sharp en anglais) a été développé par la société Microsoft, et notamment un de ses employés, Anders Hejlsberg, pour la plateforme .NET (point NET / dot NET).

Ce langage est orienté objet, avec un typage fort. Il est très proche du langage Java.

Il est précompilé en MSIL (Microsoft Intermediate Language), puis exécuté sur une machine virtuelle, ou compilé en code natif à l'exécution. Il dispose d'un ramasse-miettes (garbage collector). Il utilise l'API .NET en remplacement des MFC (Microsoft foundation class). Il semble être le nouveau langage pour développer des applications Windows, avec Visual Basic et C++.

I.2.1.Caractéristiques partagées :

Le langage C# possède un certain nombre de caractéristiques communes avec d'autres langages de programmation. Il sera donc plus facile de l'apprendre si ces autres langages sont connus.

I.2.1.1.Caractéristiques partagées avec le langage Java

- Syntaxe : les mots clés communs avec Java s'utilisent dans les mêmes circonstances et de la même manière : public, private, protected, abstract, class, interface, try, catch, finally, throw, new, return, this, if, while, do, for, foreach, enum... et le mot clé lock est l'équivalent C# du mot clé Java synchronized;
- Garbage collector : les objets qui ne sont plus référencés sont traités par le ramasse-miettes afin de libérer la mémoire qu'ils occupent ;
- Références : les objets sont en fait des références ;
- Documentation automatique : cette caractéristique commune utilise cependant une syntaxe différente dans les 2 langages : le langage Java utilise les commentaires spéciaux `/** */` au format HTML, avec des tags commençant par le caractère arobase `@` ; tandis que le langage C# utilise les commentaires `///` au format XML ;

Méthodes courantes : une majeure partie des méthodes de l'API de C# ressemblent à celles de Java, excepté que leur nom commence par une majuscule : Main, Equals, ToString, Length, IndexOf,

I.2.1.2.Caractéristiques partagées avec le langage C++ :

- surcharge des opérateurs ;
- structures (mot clé `struct`) ;
- énumérations (mot clé `enum`) ;
- pointeurs : il est possible, en mode *unsafe*, d'utiliser des pointeurs au lieu de références

I.2.1.3. Caractéristiques partagées avec d'autres langages :

- propriétés (Delphi) : une propriété est un couple de méthodes (*get* et *set*) appelées lorsque celle-ci est lue ou modifiée ;
- attributs : un attribut est lié à une déclaration et contient des méta-données sur celle-ci (méthode obsolète, importée d'une DLL, ...)
- *delegate* : un *delegate* est un modèle de méthode, qui lorsqu'il est appelé, appelle toutes les méthodes qui lui ont été associées. Pour faire un parallèle avec le C++, les *delegates* peuvent être comparés à des pointeurs de fonction. Leur sémantique est toutefois nettement plus riche qu'en C++.

I.3. Exemple sur Le code source :

Voici des exemples sur code sources C#

exemple d'un programme Hello world typique, écrit en :

```
using System;

class HelloWorld
{
    public static void Main()
    {
        Console.WriteLine("Hello World!");
    }
}
```

C# possède les instructions try et catch permettant de gérer les exceptions (comportement non attendu des instructions du programme), similaires dans la syntaxe à celles du C++.

Exemple de code tentant de créer un fichier « document.txt » sur le serveur « Toto » :

```
try
{
    // Tentative de création du fichier 'document.txt' sur le serveur
    'Toto'
    File.Create(@"\\toto\document.txt");
}
catch
{
    // Impossible de contacter le server Toto
    MessageBox.Show("L'application n'arrive pas à créer le fichier
'document.txt' sur le serveur 'Toto' !", "Alerte");
}
```

Notez l'utilisation d'une chaîne de caractères verbatim : le caractère arobase précède le guillemet donc l'anti-slash n'est pas doublé. Ce genre de chaîne de caractères est pratique pour les chemins sous Windows.

Dans cet exemple, la fonction `File.Create` retourne un flux (`FileStream`), ou elle peut lancer une exception si une erreur s'est produite (problème de connexion par exemple).

Dans cet exemple, aucune information sur l'exception n'est obtenue : on cherche juste à savoir si le programme ne s'est pas comporté normalement, auquel cas on arrive dans le bloc `catch`.

```
try
{
    // Tentative de création du fichier 'document.txt' sur le serveur
    'Toto'
    File.Create(@"\\toto\document.txt");
}
catch(Exception err)
{
    // Impossible de contacter le server Toto
    MessageBox.Show("L'application n'arrive pas à créer le fichier
'document.txt' sur le serveur 'Toto' ! Erreur:" + err.Message, "Alerte");
}
```

À l'instar de [C++](#) qui a un type d'exception de base (`class exception` dans l'en-tête `<exception>`) et dont les autres exceptions héritent, toute exception C# est héritée (ou une instance) du type `System.Exception`. Ainsi, si on cherche à savoir ce qui s'est passé, une solution simple reste d'obtenir une référence vers l'exception de la manière suivante :


```
try
{
    // Tentative de création du fichier 'document.txt' sur le serveur
    'Toto'
    File.Create(@"\\toto\document.txt");
}
catch(Exception err)
{
    // Impossible de contacter le server Toto
    MessageBox.Show("L'application n'arrive pas à créer le fichier
'document.txt' sur le serveur 'Toto' ! Erreur:" + err.Message, "Alerte");
}
```

Ainsi, une information complète sera retournée, décrivant la nature de l'exception qui s'est produite.

En fonction des fonctions appelées, le framework .NET fournit la liste des exceptions que l'appel est susceptible de retourner en cas d'erreur. Dans le cas de la fonction 'Create', voici la liste des exceptions possibles :

- System.UnauthorizedAccessException
- System.ArgumentException
- System.ArgumentNullException
- System.IO.PathTooLongException
- System.IO.DirectoryNotFoundException
- System.IO.IOException
- System.NotSupportedException

```
try
{
    // Tentative de création du fichier 'document.txt' sur le serveur
    'Toto'
    File.Create(@"\\toto\document.txt");
}
catch(System.ArgumentException ArgumentErr)
{
    // L'argument n'est pas valable. Le nom de fichier 'document.txt' n'est
    pas valable
    MessageBox.Show("L'argument n'est pas valable. Le nom de fichier
'document.txt' n'est pas valable ! Erreur:" + ArgumentErr.Message,
    "Alerte");
}
catch(Exception err)
{
    // Impossible de contacter le server Toto
    MessageBox
```

De la même manière qu'en [C++](#), l'envoi d'une exception se fait avec le mot-clef `throw` :

```
public uint Divide(uint num, uint div) // Fonction de division dans N
{
    if(div == 0)
        throw new Exception ("Division par 0 !");
    if(num < div)
        throw new Exception ("num est strictement inférieur à div : le
résultat de la division ne sera pas un entier naturel !");
    return num / div;
}
//Code ailleurs :
try
{
    Divide(12, 6); // Retourne 2, aucune exception
    Divide(10, 20); // Exception, cette division ne donne aucun résultat
dans N
    Divide(10, 0); // Exception, division par zéro impossible
}
catch (Exception e)
{
    MessageBox.Show("L'erreur suivante a été retournée :\n" + e.Message,
"Alerte");
}
```

L'envoi d'une exception (via `throw`) ou la levée d'une exception (dans un bloc `try/catch`) met immédiatement fin au bloc en cours. Ainsi, si `div` est nul, le code de la fonction `Divide` s'arrêtera à la ligne 2. De même, dans l'exemple précédent, la troisième fonction `Divide` ne sera jamais exécutée à cause de la levée d'une exception lors du deuxième appel à `Divide`. Une exception non attrapée (*catchée*) - c'est-à-dire que l'application n'a pas encadré le code lançant l'exception par un bloc `catch` approprié - met fin à l'application immédiatement sous la forme d'une exception de type `UnhandledException` (exception non gérée en français).

Comme en [Java](#), il est possible d'ajouter un bloc `finally` pour exécuter une série d'instructions, quoi qu'il se passe (exception lancée ou non). Cela est utile pour libérer des ressources quel que soit ce qui peut se passer entre la prise de ressource et la libération. Exemple typique : lecture de fichiers

```
FileStream fs = new FileStream(@"C:\Fichier.txt", FileMode.OpenOrCreate,
FileAccess.ReadWrite);
// Arrivé ici, le fichier est ouvert
try
{
    // ... opérations de lecture / écriture pouvant lancer des exceptions
...
}
finally
{
    // Fermer le fichier quoi qu'il se passe :
    fs.Close();
}
```

II. Présentation de quelque pages du logiciel :



Figure 7 : fiche Admin



Figure 8 : fiche global de logiciel

Plaint

Description :

Place :

Date/Time :

Poste police :

Enquêteur :

Court :

Témoins Evidence Arêté
Propriétés Suspect Crime

Nouveau Supprimer
Mettre à jour Rechercher

<< Premier Dernier >>
< Precedent Suivant >

ID	Description	Place	Date/Time	Police St
13	fiche rassist	tlemcen	26/05/2015 23:2...	post polic
12	agression	tlemcen	26/05/2015 23:2...	post polic

Figure : fiche plaint.

The screenshot shows a web application window titled "Complainant Details" with a green background. The main heading is "Les Détails des plaignants". The form contains several input fields: "Complaint : Name" (with a dropdown arrow), "Name" (filled with "mohamed"), "Date of Birth" (filled with "24/05/1985"), "Gender" (radio buttons for "Male" and "Female", with "Male" selected), "Occupation" (filled with "azedine"), "Address" (filled with "rue 09 maghnia"), "Email" (filled with "ddd"), and "Tel" (filled with "07777"). To the right, there are fields for "Contact Person" and "Contact Person Tel". A set of buttons includes "Nouveau", "Supprimer", "Mettre à jour", "Rechercher", "<< Premier", "Dernier >>", "< Precedent", and "Suivant >". Below the form is a table with the following data:

	ID	Name	Complaint ID	Complaint Name	Date Of B
▶	1	mohamed			24/05/19

Figure 9 : fiche des détail de plaignant

Conclusion

Dans ce chapitre, on a en premier lieu, présenté les différents outils et langages utilisés dans l'implémentation de notre application. Par la suite, on a présenté quelques interfaces de l'application ainsi qu'on l'a conçu. L'application fonctionne correctement et offre toutes les fonctions utiles et nécessaires au bon fonctionnement d'application de la gestion, mais on espère encore l'améliorer.

Conclusion

générale

.

L'objectif de notre projet était de développer un logiciel permettant la gestion des plaintes des différents plaignants avec objectif analyse des différentes crimes de cybercriminalités

Cette automatisation concerne la gestion de d'une plainte en utilisant Le langage de modélisation UML .

Après le passage par les différentes étapes de développement, l'application a abouti à un logiciel fonctionnel qui répond globalement aux critères imposés dans ce domaine.

Le présent travail nous a permis d'acquérir des connaissances dans le domaine de la programmation , et de conforter nos connaissances en conception logicielle.

Bibliographie

Liste des ouvrages

- [1]· NARVARTH : Edition 2008 Apprenez à programmer en C# sur .NET
- [2]Hungry Minds. : Edition 2001 C# pour les Nuls 2001 nb de page 565 page
- [3] Dunod : Edition 2002 **C# pour les nuls**, nb page 264

Liste DVD et vidéo de Formation

- [4]· VIDEO 2 BRAIN

Sites web

- [5] <http://uploads.siteduzero.com/pdf/344102-apprenez-a-programmer-en-c-sur-net.pdf>
- [6] <http://uploads.siteduzero.com/pdf/344102-apprenez-a-programmer-en-c-sur-net.pdf>
- [7] <http://openclassrooms.com/courses/apprenez-a-programmer-en-c-sur-net>
- [8] http://fr.wikibooks.org/wiki/Programmation_C_sharp
- [9] <http://codes-sources.commentcamarche.net/source/list/c-net-13/last>