

III.1 Introduction

Nous allons présenter dans ce chapitre, consacré à des étapes de réalisation de notre travail, la description et l'environnement de travail puis, la présentation des outils utilisés et finalement nous présentons quelques fenêtres de notre application.

III.2 l'environnement de travail

Pour réaliser aisément notre application, nous avons choisi de travailler sous **java**

III.2.1 Définition du java:

Développé par Sun Microsystems depuis la fin des années 1980, Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Il existe 2types de programmes en Java : les applets et les applications. Une application autonome (stand alone program) est une application qui s'exécute sous le contrôle direct du système d'exploitation. Une applet est une application qui est chargée par un navigateur et qui est exécutée sous le contrôle d'un plug in de ce dernier.

III.2.2 Les caractéristiques de java

Java possède un certain nombre de caractéristiques qui ont largement contribué à son énorme succès :

- ✓ **Java est interprété** : le source est compilé en pseudo code ou byte code puis exécuté par un interpréteur Java : la Java Virtual Machine (JVM). Ce concept est à la base du slogan de Sun pour Java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). En effet, le byte code, s'il ne contient pas de code spécifique à une plateforme particulière peut être exécuté et obtenir quasiment les mêmes résultats sur toutes les machines disposant d'une JVM.
- ✓ **Java est indépendant de toute plate-forme** : il n'y a pas de compilation spécifique pour chaque plate forme. Le code reste

indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine. Cette indépendance est assurée au niveau du code source grâce à Unicode et au niveau du byte code.

- ✓ **Java est orienté objet** : comme la plupart des langages récents, Java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les autres pour former une application. Java n'est pas complètement objet car il définit des types primitifs (entier, caractère, flottant, booléen,...).
- ✓ **Java est simple** : le choix de ses auteurs a été d'abandonner des éléments mal compris ou mal exploités des autres langages tels que la notion de pointeurs (pour éviter les incidents en manipulant directement la mémoire), l'héritage multiple et la surcharge des opérateurs, ...
- ✓ **Java est fortement type** : toutes les variables sont typées et il n'existe pas de conversion automatique qui risquerait une perte de données. Si une telle conversion doit être réalisée, le développeur doit obligatoirement utiliser un cast ou une méthode statique fournie en standard pour la réaliser.
- ✓ **Java assure la gestion de la mémoire** : l'allocation de la mémoire pour un objet est automatique à sa création et Java récupère automatiquement la mémoire inutilisée grâce au garbage collector qui restitue les zones de mémoire laissées libres suite à la destruction des objets.
- ✓ **Java est sûr** : la sécurité fait partie intégrante du système d'exécution et du compilateur. Un programme Java planté ne menace pas le système d'exploitation. Il ne peut pas y avoir d'accès direct à la mémoire. L'accès au disque dur est réglementé dans une applet. Les applets fonctionnant sur le Web sous soumises aux restrictions suivantes dans la version 1.0 de Java :
 - Aucun programme ne peut ouvrir, lire, écrire ou effacer un fichier sur le système de l'utilisateur ;

- Aucun programme ne peut lancer un autre programme sur le système de l'utilisateur ;
 - Toute fenêtre créée par le programme est clairement identifiée comme étant une fenêtre Java, ce qui interdit par exemple la création d'une fausse fenêtre demandant un mot de passe ;
 - Les programmes ne peuvent pas se connecter à d'autres sites Web que celui dont ils proviennent.
- ✓ **Java est économe** : le pseudo code a une taille relativement petite car les bibliothèques de classes requises ne sont liées qu'à l'exécution.
- ✓ **Java est multitâche** : il permet l'utilisation de threads qui sont des unités d'exécution isolées. La JVM, elle même, utilise plusieurs threads.

Ainsi a ce basant sur ces caractéristiques, nous avons porté notre choix sur ce langage pour le développement de notre application, dans le but de pouvoir déployé notre application largement dans n'importe quelle plate forme.

III.2.3 Les différentes éditions et versions de Java

- Sun/Oracle fournit gratuitement un ensemble d'outils pour permettre le développement de programmes avec Java. Ce kit, nommé JDK, est librement téléchargeable sur le site web de Sun <http://java.sun.com>
- Le JRE (Java Runtime Environment) contient uniquement l'environnement d'exécution de programmes Java. Le JDK contient lui-même le JRE. Le JRE seul doit être installé sur les machines où des applications Java doivent être exécutées.
- Sun définit trois plateformes d'exécution (ou éditions) pour Java pour des cibles distinctes selon les besoins des applications à développer :
 - ✓ Java Standard Edition (J2SE / Java SE) : environnement d'exécution et ensemble complet d'API pour des applications de type desktop. Cette plate-forme sert de base en tout ou partie aux autres plateformes.
 - ✓ Java Enterprise Edition (J2EE / Java EE) : environnement d'exécution reposant intégralement sur Java SE pour le développement d'applications d'entreprises.

- ✓ Java Micro Edition (J2ME / Java ME) : environnement d'exécution pour le développement d'applications sur appareils mobiles et embarqués dont les capacités ne permettent pas la mise en œuvre de Java SE.
- Sun fournit le JDK, à partir de la version 1.2, sous les plates-formes Windows, Solaris et Linux.
- La version 1.3 de Java est désignée sous le nom Java 2 version 1.3.
- La version 1.5 de Java est désignée officiellement sous le nom J2SE version 5.0.
- La version 6 de Java est désignée officiellement sous le nom Java SE version 6.

Chaque version de la plate-forme Java possède un numéro de version et un nom de projet. A partir de la version 5, la plate-forme possède deux numéros de version :

- ✓ Un numéro de version interne : exemple 1.5.0.
- ✓ Un numéro de version externe : exemple 5.0.A

III.3 Outils de développement

La communauté Java est très productive car elle regroupe :

III.3.1 JDOM (acronyme de l'anglais Java Document Object Model), est une bibliothèque open source pour manipulation des fichiers XML en Java. Elle intègre DOM et SAX, et supporte XPath et XSLT. Elle utilise des analyses syntaxiques externes pour construire les documents.

III.3.2 stanford-parser : Un analyseur de langage naturel est un programme qui fonctionne à la structure grammaticale des phrases, par exemple, où des groupes de mots vont ensemble (comme «phrases») et les mots qui font l'objet ou objet d'un verbe. Analyseurs probabilistes utilisent les connaissances de la langue acquise de phrases analysé à la main pour essayer de produire l'analyse la plus probable de nouvelles phrases. Ces analyseurs statistiques font encore quelques erreurs, mais souvent travaillent plutôt bien.

Leur développement a été l'une des plus grandes percées dans le traitement du langage naturel dans les années 1990.

III.3.3 Un dictionnaire : est un ouvrage de référence contenant l'ensemble des mots d'une langue ou d'un domaine d'activité généralement présentés par ordre alphabétique et fournissant pour chacun une définition, une explication ou une correspondance (synonyme, antonyme, cooccurrence, traduction, étymologie) don votre Project en utilisé un dictionnaire ce format XML téléchargé en Net.

III.4 Structure du système implémenté

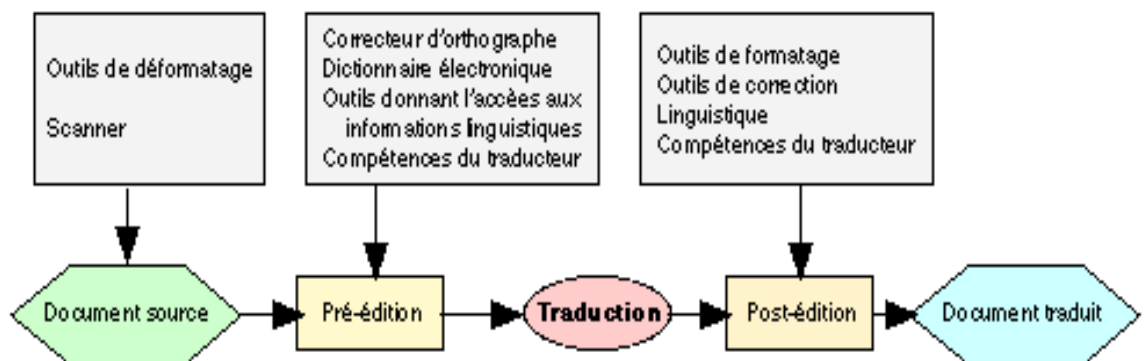


Figure. 3.1 : Structure du système implémenté

III.5 Présentation du système implémenté

Nous avons présenté les différentes phases de système implémenté comme suite :

III.5.1 l'analyse du texte source

Il décortique la structure de chaque phrase et étudie les fonctions des groupes de mots, disséquant sujet, verbe et compléments (analyse lexical).

III.5.2 l'application des règles

applique alors à tous les éléments analysés, les « règles » syntaxiques, sémantiques et morphologiques. Pour votre étude Enna appliqué 2 règle et les 2 règle fonctionné bien.

III.5.3 l'intervention du dictionnaire

le logiciel pioche dans son dictionnaire afin de passer en revue tous les cas possibles pour un mot.

III.6 fenêtres de notre application.

Notre application contient les fenêtres suivantes :

1. Fenêtre d'accueil

Contient 3 buttons:

- ♦ **Traduction** : pour avoir la fenêtre de travail.
- ♦ **Objectif** : pour avoir la fenêtre d'informations.
- ♦ **Analyse** : pour avoir les type de l'analyse.



Figure 3.2 : Fenêtre d'accueil

2. Fenêtre de travail

Contient 3 buttons:

- ♦ **Vérification d'orthographe** : pour vérifie l'orthographe de texte source
- ♦ **Traduire** : pour détecter l'angle d'inclinaison par la méthode de Trincklin sans faire les étapes manuellement.
- ♦ **Quitter**: pour retourner à la fenêtre d'accueil.

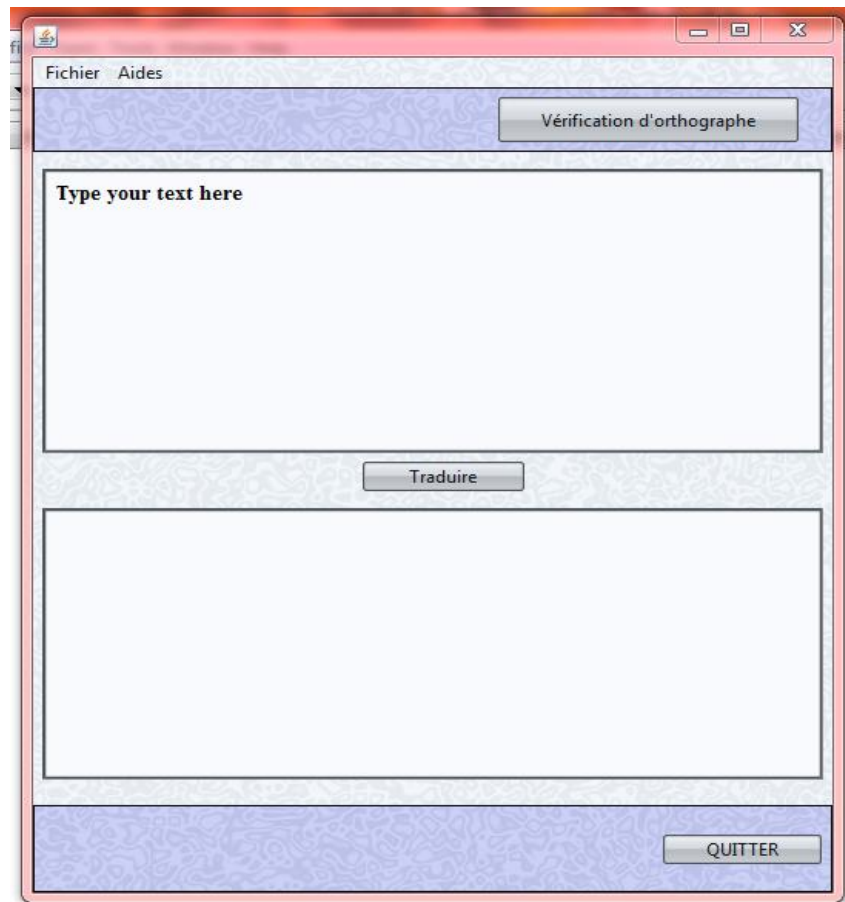


Figure.3. 3 : Fenêtre de travail

3. Fenêtre objectif

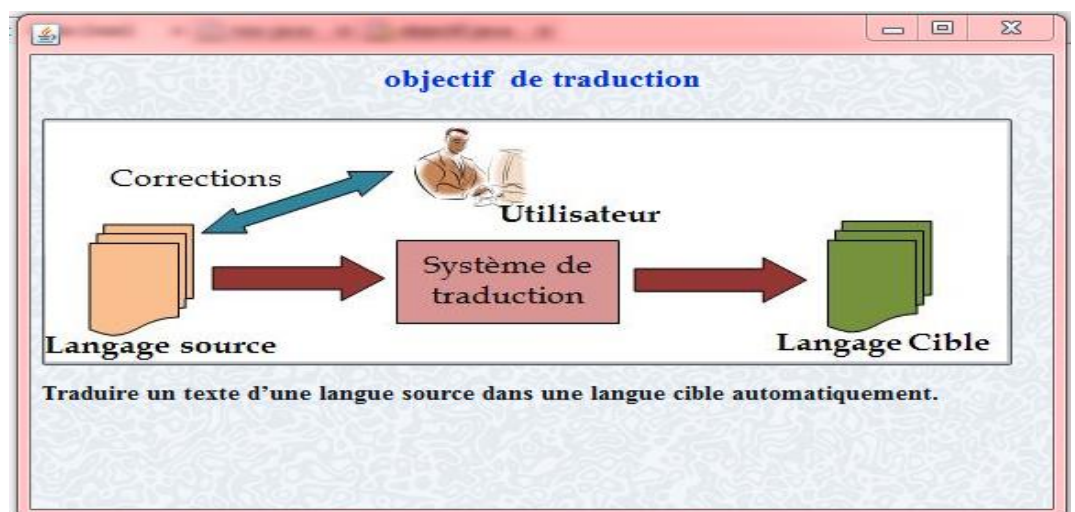


Figure.3.4 : Fenêtre objectif

4. Fenêtre d'Analyse

Contient 4 buttons:

- ♦ **Analyse lexicale** : pour décomposer une chaîne de caractères en unités ou entités lexicales
- ♦ **Analyse Syntaxique**: pour vérifie le syntaxe de phrase
- ♦ **Analyse de morphologique**: chaque mot reconnu, les diverses informations linguistiques
- ♦ **Quitter**: pour retourner à la fenêtre d'accueil.

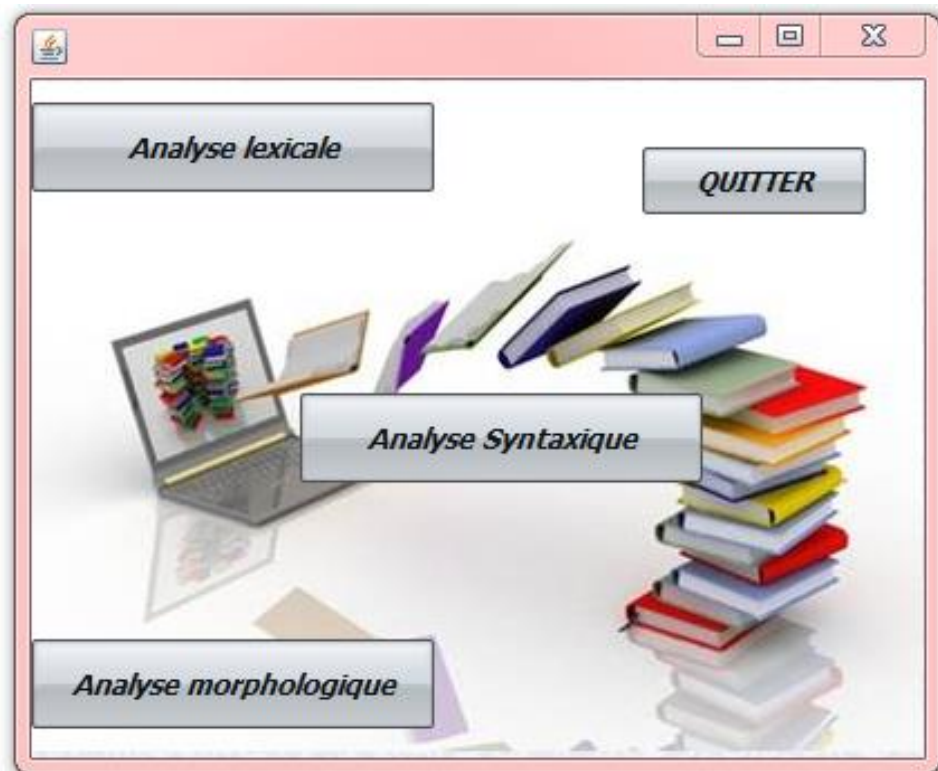


Figure. 3.5 : Fenêtre d'Analyse

A. Analyse lexicale

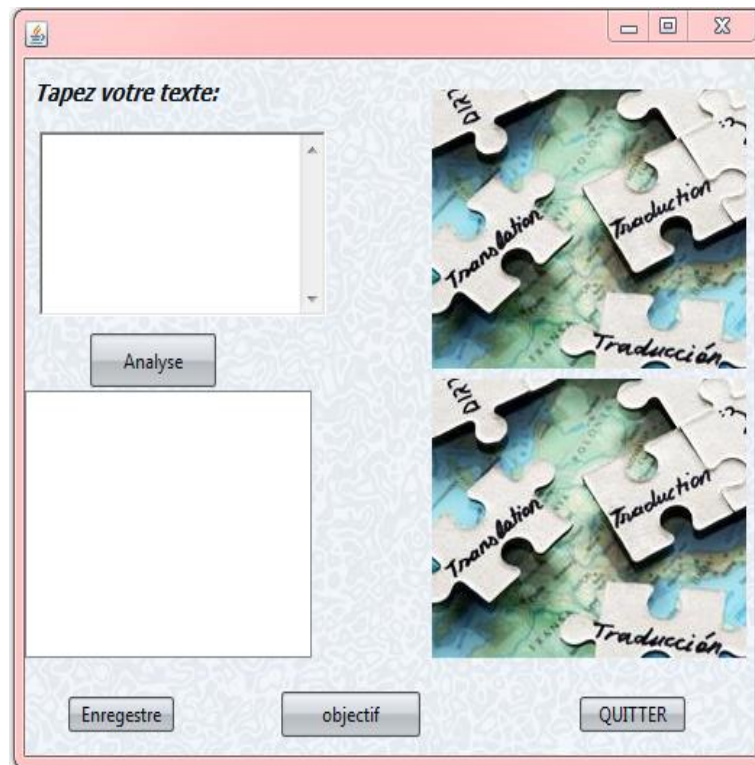


Figure. 3.6 : Fenêtre de l'Analyse lexicale

B. Analyse Syntaxique

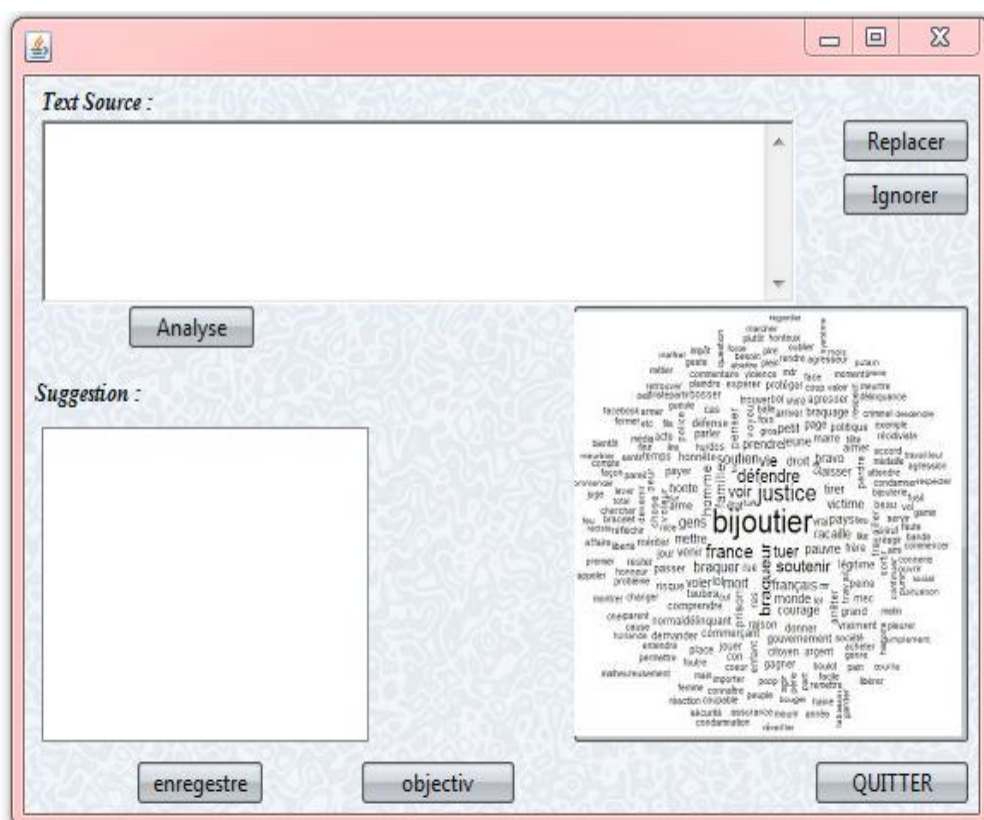


Figure. 3.7 : Fenêtre d' Analyse Syntaxique

C. Analyse de morphologique

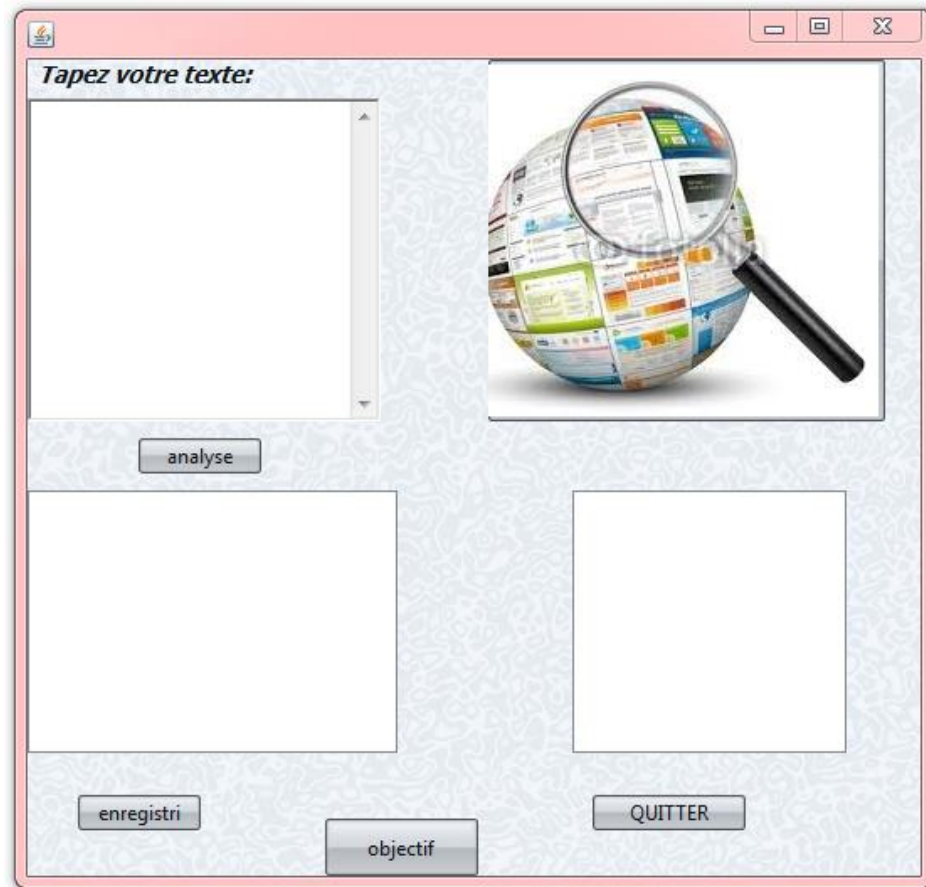
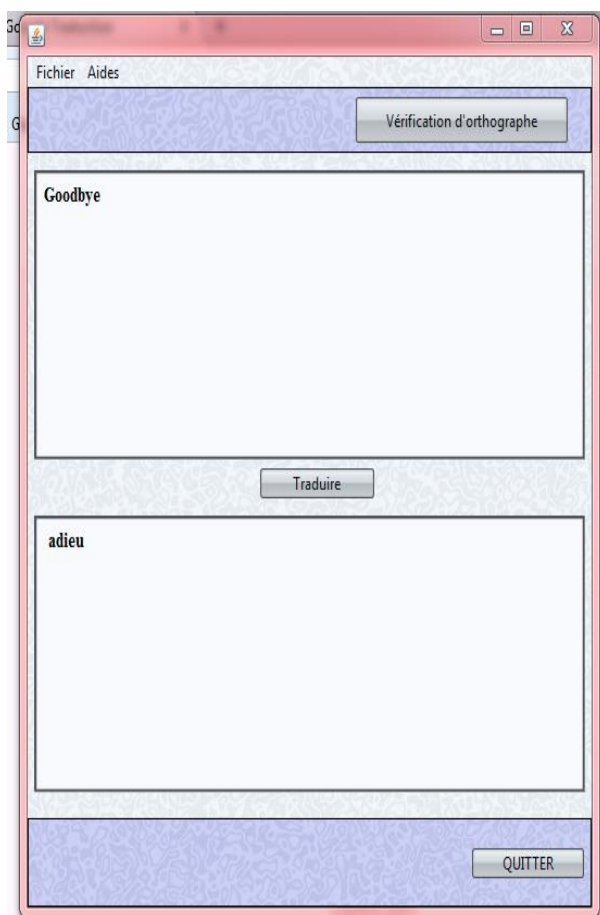
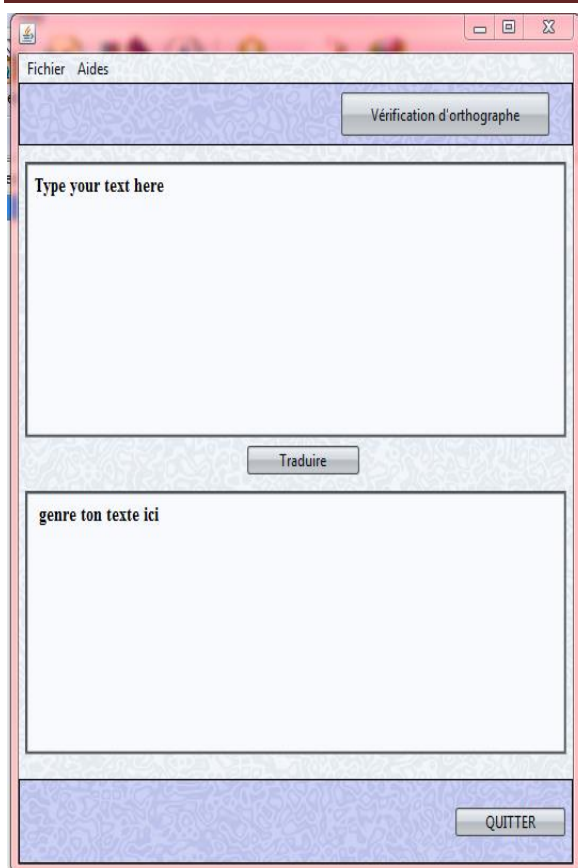


Figure. 3.8 : Fenêtre d' Analyse de morphologique

III.7 Résultats de traducteur

Nous avons testé notre système sur des phrase English simple et plus utilisé, et les résultats comme suit:



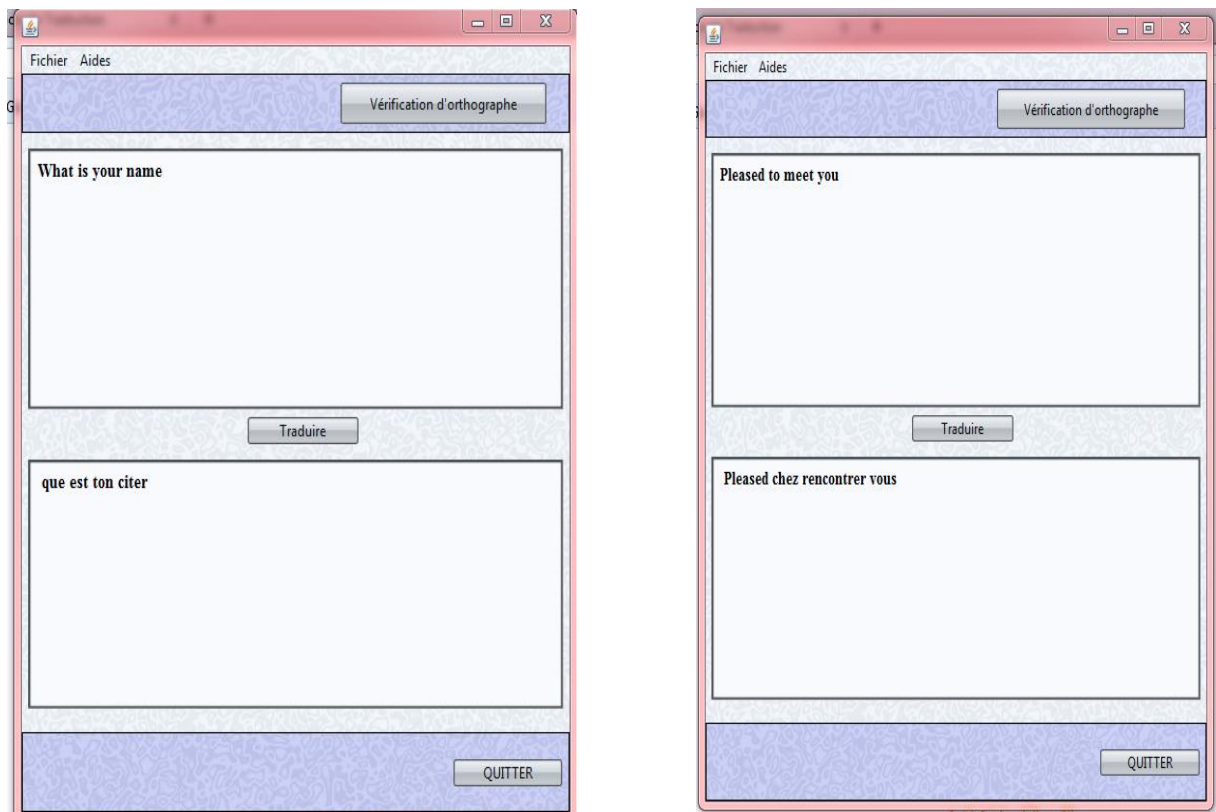


Figure. 3.9 : Résultats de traduction

Le resultat pas bon et pas mal mi A Chaque fois que nous ajoutons les règles de linguistique et grammatical, plus le niveau de la traduction.

III.8 Conclusion

Dans ce chapitre, on représente les étapes de réalisation de notre travail, la description et l'environnement de travail puis, la présentation des outils utilisés et finalement nous présentons quelques fenêtres de notre application.