

République algérienne démocratique et populaire
الجمهورية الجزائرية الديمقراطية الشعبية

Ministre de l'enseignement supérieur et de la recherche scientifique
وزارة التعليم العلي والبحث العلمي

Université MOHAMED EL BACHIR EL IBRAHIMI - Bordj Bou Arreridj
جامعة محمد البشير الإبراهيمي - برج بوعريريج



FACULTE DES MATHÉMATIQUES ET DE L'INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

MEMOIRE

Présenté pour l'obtention du diplôme de

MASTER EN INFORMATIQUE

Option : Réseaux et multimédia

Thème :

Application Android pour la traduction de l'arabe

Présenté Par :

- *Anis Derbel*

- *Seif Eddine AMAMRA*

Devant le jury composé de :

Président : Saïfi Linda

maitre de conférence Université de B.B.A

Examineur : Saidani khawter

maitre de conférence Université de B.B.A

Dirigé par :

- *Dr. Samir AKHROUF*

maitre de conférence Université de B.B.A

Date de soutenance : 26/09/2019

Promotion : 2018/2019

REMERCIEMENTS

Nous remercions Dieu de nous avoir accordé des connaissances de la science et de nous avoir aidés à réaliser ce travail.

Ce mémoire n'aurait pas pu être réalisé sans la contribution de nombreuses personnes que nous tenons à remercier par ces quelques lignes.

Nous remercions principalement nos parents pour tous les sacrifices qu'ils ont consentis pour notre éducation depuis l'école primaire jusqu'à ce jour, les sacrifices lesquels nous ne nous saurons énumérés

Nous tenons à remercier vivement notre chère encadreur Dr Samir AKHROUF Maître de conférence à l'Université de Bordj Bou Arreridj qui a fourni des efforts énormes, par ses informations ses conseils et ses encouragements.

Nos remerciements chaleureux ne doivent exclure plusieurs ne que cette petite famille ne peut en contenir tous notamment.

Nous remercions également les employés de la bibliothèque universitaire et de tous les techniciens de laboratoire.

Nous aussi remercier chaleureusement tous ceux qui nous enseigné durant nos années d'études.

Pour terminer nous souhaitons exprimer nos remerciements ainsi que nous respect à tout le personnel de l'université d'El Bachir El IBrahimi.

Dédicaces

Grace à dieu tout puissant et en signe de reconnaissance à tous les sacrifices consentis pour ma réussite et la volonté pour mener à bien

Ce modeste travail que je dédie :

Aux personnes les plus chères à mon cœur, pour tous leurs sacrifices, leurs amour, leurs tendresses, leurs soutiens et leurs prières tout au long de mes études, et qui ont attendu avec patience les fruits de leur bonne éducation, A mes chers parents.

A mon grand-mère, pour leurs encouragements permanents, et leur soutien moral.

A mon chers frères Chouaib et mes chères sœurs Soundes et Rahile, tous les mots que je puisse dire ne peuvent exprimer ma gratitude et mon amour vers vous.

A toute ma famille pour leur soutien tout au long de mon parcours universitaire.

A mon ami seifeddine.

A tous mes chers amis(O.Mouhamed,Z.Redha,L.Abdelramane,S.Ahmed)

et collègues avec lesquels j'ai partagé mes moments de joie et de bonheur

A mon encadreur Samir AKHROUF pour son soutien, sa disponibilité et son accompagnement bienveillant. Son aide m'a été très précieuse durant tout le parcours de ce travail.

A toutes les personnes qui ont aidé à la réalisation de ce travail.

A tous ceux dont l'oubli du nom n'est pas celui du cœur.

Anis Derbal

Dédicaces

On remercie dieu le tout puissant de nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Je dédie ce modeste travail :

A mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi.

A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur, maman que j'adore.

A mes chers sœurs Khaoula, Salsabil, Roufaïda et Sondos, Je te souhaite un avenir plein de joie, de bonheur, de réussite et de sérénité.

A ma chère femme, les mots ne suffisent guère pour exprimer l'attachement, l'amour et l'affection que je porte pour toi.

A tous les membres de ma famille sans aucune exception.

A mon ami Anis.

A mes chers amis et collègue Fouzi, Saleh, Youba, Fouad et zergou.

A mon encadreur Samir AKHROUF par ce que Sans son aide, ses conseils et ses encouragements ce travail n'aurait vu le jour.

À tous ceux et toutes celles qui m'ont accompagné et soutenu durant ma carrière.

Seif Eddine AMAMRA

SOMMAIRE

Remercîments	2
Dédicaces.....	3
Sommaire.....	5
Liste des abréviations	7
Liste des figures.....	8
Liste des tableaux.....	9
Introduction générale	10

CHAPITRE 01 : Le système androïde

1. Introduction.....	11
2. Que ce que c'est qu'Androïde ?.....	11
3. Historique	11
4. Versions d'Androïde	12
5. L'Architecture d'Androïde	14
5.1 Applications.....	15
5.2 Framework de développement.....	15
5.3 Bibliothèques.....	16
5.4 Androïde Runtime.....	18
5.5 Linux Kernel	18
6. Cycle de vie d'une Activity.....	19
7. Les outils de développement Androïde	21
7.1 Le SDK Androïde.....	21
7.2 Le débogueur « ADB ».....	21
7.3 Les IDE « Android Developer Tools » et « Android Studio »	21
7.4 La machine virtuelle Dalvik	22
7.5 La runtime Androïde (ART)	22
8. Conclusion	23

CHAPITRE 02 : Traduction automatique

1. Introduction	25
2. Historique	25
3. Les différents systèmes de la TA	25
4. Les connaissances nécessaires pour la TA.....	27
4.1 Les dictionnaires.....	27
4.2 Les connaissances syntaxiques	27
4.3 Les connaissances sémantiques	29
4.4 Les connaissances contextuelles	30
5. la traduction automatique et la langue arabe.....	31
6. Conclusion.....	32

CHAPITRE 03 : Conception

1.	Introduction	34
2.	Diagramme de cas d'utilisation	34
2.1	Définition de diagramme de cas d'utilisation	34
2.2	Diagramme de cas d'utilisation générale.....	34
3.	Diagramme de séquence.....	36
3.1	Définition de diagramme de séquence.....	36
3.2	Diagramme d'interaction authentification d'un utilisateur.....	37
3.3	Diagramme de séquence "Transalte"	38
3.4	Diagramme de séquence "History".....	39
3.5	Diagramme de séquence "Addword".....	40
4.	Diagramme de classe	41
4.1	Définition de diagramme de classe	41
4.2	Diagramme de classe de l'application	42
5.	Conclusion.....	42

CHAPITRE 04 : Réalisation

1.	Introduction	44
2.	Environnement du travail.....	44
2.1	Environnement matériel.....	44
2.2	Environnement logiciel	44
3.	Langages de programmation	44
3.1	Le langage JAVA	44
3.2	SQLite	45
3.3	XML	45
3.4	Firebase	45
4.	Description de l'interface de l'application.....	45
4.1	Interface« Loading».....	46
4.2	Interface « TranslatPage ».....	46
4.3	Code de traduction	48
4.4	Code de speaker	49
4.5	Interface« speaker ».....	50
4.6	Interface « login »	51
4.7	Interface « Favorite ».....	52
4.8	Interface « History ».....	53
4.9	Interface « Addword »	54
4.10	Code addword.....	55
4.11	Interface « Navigation drawer».....	56
5.	Conclusion	56
Conclusion générale et perspective.....		57
Bibliographie.....		58
Webographie.....		59
Résumé.		60

Liste des abréviations et acronymes

UP: unified processor ou processus unifié.

OHA : l'Open Handset Alliance

TA : Traduction automatique

PHP : Hypertexte Préprocesseur ou Personal Home Page.

CSS: Cascading Style Sheets.

XML: eXtensible Markup Language.

HTML: Hypertext Markup Language.

XHTML: eXtensible Hypertext Markup Language.

WWW: World Wide Web.

UML : Unified Modeling Language.

BDD : Base De Données.

SGBD : Serveur de Gestion de Base de Données.

SQL: Structured Query Language

MySQL: My Structured Query Language

JPEG: Joint Photographie Expert Group

PNG: Protocol Network Graphics

HTTP: Hyper Text Transfert Protocol

Liste des figures

Chapitre 01 :

Figure 1.1 : Les principaux composants du système d'exploitation android.....14

Figure 1.2 : L'architecture de ces médiathèques.17

Figure 1.3 : Fonctionnement de la pile d'activités.....20

Figure 1.4 : Le cycle de vie d'une application.....20

Chapitre 02 :

Figure 2.5 : Système traduction direct.25

Figure 2.6 : Système interlingual.....25

Figure 2.7 : Système par Transfer.....27

Chapitre 03 :

Figure 3.8 : Diagramme de cas d'utilisation.35

Figure 2.9 : Diagramme d'interaction authentification d'un utilisateur.....37

Figure 2.10 : Diagramme de séquence « translate ».38

Figure 2.11 : Diagramme de séquence « history ».39

Figure 2.12 : Diagramme de séquence « Addword».....40

Figure 2.13 : Diagramme des classes.....42

Chapitre 04 :

Figure 4.14 : Interface « Loading ».46

Figure 4.15 : Interface « PageTraduction».....47

Figure 4.16 : Code de traduction« PageTraduction ».48

Figure 4.17 : Code bouton « Speaker ».49

Figure 4.18 : Interface « speaker ».50

Figure 4.19 : Interface « login ».51

Figure 4.20 : Interface « favorite ».52

Figure 4.21 : Interface « History ».53

Figure 4.22: Interface « add word ».54

Figure 4.23 : Code « add word ».55

Figure 4.24 : Interface « page list ».56

Liste des tableaux

Chapitre 01 :

Tableau 1.1 : Les différentes versions d'Android.13

Tableau 1.2 : Tableau comparatif entre Eclipse et Android Studio 20

Chapitre 02 :

Tableau 2.3: Table des mots français anglais.....28

INTRODUCTION GENERAL

L'utilisation des Smartphones et des tablettes augmente jour après jour, les chiffres le montrent très bien. C'est ainsi que les statistiques montrent qu'il se vend environ 130 millions de Smartphones par mois dans le monde soit 1,56 milliard par an ou 50 par seconde. C'est en Chine qu'on trouve le plus grand nombre d'utilisateurs de Smartphones en 2017 (713,31 millions) devant l'Inde, les États-Unis, le Brésil, la Russie. En Algérie, l'utilisation des Smartphones est en pleine croissance surtout avec la baisse des prix qui les a rendus de plus en plus accessible à toutes les catégories de la population.

Par ailleurs, les applications Androïdes connaissent une croissance très rapide et l'innovation continue implique un nombre grandissant d'opportunités pour les développeurs d'applications qui vont de pair avec l'émergence de nouvelles tendances parmi les utilisateurs d'applications.

C'est dans ce contexte que se situe notre travail, c'est-à-dire le développement d'une application mobile sous androïde. Notre objectif est donc d'étudier le système androïde et de développer une application mobile. Pour ce faire nous avons choisi de développer une application permettant de faire la traduction de l'arabe. Ce choix est motivé par l'utilité d'une telle application pour les utilisateurs de Smartphones vu le nombre croissant d'utilisateurs et la diversité des sites accédés qui utilisent en majorité l'anglais.

Notre rapport, s'articule autour de quatre chapitres :

Le premier chapitre est consacré à une introduction d'androïde, citant son historique, ses versions, son architecture et ses outils de développements.

Dans le second chapitre nous présenterons la traduction automatique et ses différents systèmes.

Dans le troisième chapitre, nous allons faire une conception détaillée de notre application en présentant les cas d'utilisation sous différents diagrammes.

Le quatrième et dernier chapitre, sera la partie de l'implémentation et la réalisation de notre application, ou nous présenterons l'environnement de développement, la structure de notre application et quelques interfaces de celle-ci.

Enfin, nous clôturons ce mémoire par une conclusion générale et perspective.

CHAPITRE 01

Le Système Androïde

1. Introduction

Dans le monde actuel, les smartphones et tablettes font partis de notre vie quotidienne, en la rendant plus facile et commode, ils sont équipés d'applications qui sont conçues pour des plateformes mobiles et utilisés pour des services d'information, de medias sociaux, de jeux etc...

Un smartphone est doté d'un système d'exploitation et cela varie de sa marque, celui qui est le plus répandu sur le marché est « Android » (grâce à son open source).

Nous présentons dans ce chapitre une description du système d'exploitation Androïde, son historique, ses fonctionnalités et finalement son architecture.

2. Que ce que c'est qu'androïde ?

Androïde est un système d'exploitation open-source pour smartphones, PDA et autres terminaux mobiles, conçu par Androïde, une start-up rachetée par Google en juillet 2005. Il existe d'autres types d'appareils possédant ce système d'exploitation tels que les téléviseurs et les tablettes.

Afin de promouvoir ce nouveau système d'exploitation ouvert, Google a su fédérer autour de lui un consortium d'une trentaine d'entreprises : l'Open Handset Alliance (OHA) créée officiellement le 5 novembre 2007. Toutes ces entreprises interviennent, plus ou moins directement, dans le marché de la téléphonie mobile.

Le but de cette alliance est de mettre en place des normes ouvertes dans le domaine de la téléphonie mobile. Ce qui veut dire que les développeurs d'application Androïde pourront accéder aux fonctionnalités du cœur de téléphone via une API très fournie. Androïde aura comme principaux concurrents Apple avec l'iPhone, Microsoft et son Windows Mobile et Nokia avec Symbian mais également des solutions libres telles que LIMO ou OpenMoko.

3. Historique

En juillet 2005, Google a acquis Androïde, Inc., une petite startup qui développait des applications pour téléphones mobiles. C'est à ce moment-là que des rumeurs sur l'entrée de Google dans le secteur du mobile ont commencé. Mais personne n'avait des données sûres à propos des marchés dans lesquels ils allaient se positionner.

Après ce rachat fait par Google, une équipe dirigée par Andy Rubin, un ancien d'Androïde Inc, a commencé à travailler sur un système d'exploitation pour appareil mobile basé sur linux. Durant 2 ans, avant que l'OHA soit créé officiellement, un certain nombre de rumeurs ont circulé au sujet de Google. Il a été dit que Google développait des applications mobiles de son

moteur de recherche, qu'elle développait un nouveau téléphone mobile, etc. En 2007, le 5 novembre, l'OHA a été officiellement annoncée, ainsi que son but : développer des standards open sources pour appareil mobile.

Le premier standard annoncé a été Androïde, une plateforme pour appareils mobiles basée sur un kernel linux 2.6. En octobre 2008.

4. Versions d'Androïde

Androïde a connu plusieurs versions, et chaque nouvelle version sert à corriger des bugs et ajouter de nouvelles fonctionnalités, ayant chacune d'elle droit à un nom de code sur la thématique des desserts suivant une logique alphabétique.

Dans le tableau suivant, il y a le nom de code de chaque version ainsi que sa date de sortie avec les caractéristiques majeures ajoutées à chaque unes d'elles [1] :

Tableau 1 : Les différentes versions d'Androïde

Nom de code	Version	Date de sortie	Majeur caractéristique
Apple pie	1.0	11/11/2007	<ul style="list-style-type: none"> • Téléchargement et mises à jour des applications via Android Market. • Support du Wi-Fi et du Bluetooth.
Bananas Split	1.1	22/10/2008	<ul style="list-style-type: none"> • "Afficher" & "Cacher" le pavé numérique, inclus dans le menu d'appel • Support pour sauvegarder les fichiers attachés aux MMS
Cupcake	1.5	30/04/2009	<ul style="list-style-type: none"> • Support pour les Widgets, qui permettent d'accéder rapidement à certaines informations de l'application à laquelle ils sont rattachés. • Ajout d'une option de rotation Automatique
Donut	1.6	15/09/2009	<ul style="list-style-type: none"> • Possibilité pour les développeurs d'intégrer leurs contenus dans les résultats de recherche • Amélioration de la rapidité dans la recherche et les applications utilisant la caméra
Eclair	2.0	26/10/2009	<ul style="list-style-type: none"> • Performance matériel optimisée • Support de plus de taille d'écran et résolutions

Froyo	2.2	20/05/2010	<ul style="list-style-type: none"> • Amélioration de la vitesse d'exécution • Partage de connexion USB
Gingerbread	2.3	06/12/2010	<ul style="list-style-type: none"> • Support du NFC et de nouveaux capteurs (comme le gyroscope et le baromètre) • Amélioration de la gestion de l'alimentation et du contrôle des applications
Honeycomb	3.0	22/02/2011	<ul style="list-style-type: none"> • Meilleur support des tablettes • Bureau tridimensionnel avec widgets améliorés
Ice cream sandwich	4.0	19/10/2011	<ul style="list-style-type: none"> • Amélioration du texte d'entrée et du correcteur orthographique • Ajout du WI-FI direct
	4.1	09/07/2012	<ul style="list-style-type: none"> • Amélioration de la vitesse d'exécution et de la gestion d'appareil photo • Android Beam (Transfert de données rapide : NFC + Bluetooth)
Jelly Ban	4.2	13/11/2012	<ul style="list-style-type: none"> • Widgets en mode verrouillé • Multi-utilisateurs pour les tablettes
	4.3	24/07/2013	<ul style="list-style-type: none"> • Ajout du support pour la résolution 4K • Support de Bluetooth basse consommation (BLE) et du profil Bluetooth (AVRCP) 1.3
KitKat	4.4	31/10/2013	<ul style="list-style-type: none"> • Nouvelle interface translucide • Amélioration du système de notification et des performances
Lollipop	5.0	03/11/2014	<ul style="list-style-type: none"> • Nouvelle interface de design « Material design » • Amélioration de la rapidité et de la gestion de la batterie
	5.1	09/03/2015	<ul style="list-style-type: none"> • Protection par blocage en cas de perte ou vol • Amélioration de la stabilité et des Performances
Marshmallow	6.0	05/10/2015	<ul style="list-style-type: none"> • l'authentification par empreinte digitale. • Android pay • Amélioration de google now
Nougat	7.0	22/08/2016	<ul style="list-style-type: none"> • Unicode 9.0 emoji • Meilleur support du multitâche
	7.1	04/10/2016	<ul style="list-style-type: none"> • Mode réalité virtuelle « Daydream » • Eclairage mode nuit • MAJ système 'transparentes' • Alertes pour l'utilisation de la batterie
Oreo	8.0	21/08/2017	<ul style="list-style-type: none"> • Android instant apps • Amélioration du système de notifications

			<ul style="list-style-type: none"> • Redésigne de l'écran de verrouillage
	8.1	05/12/2017	<ul style="list-style-type: none"> • Changement d'interface pour 'étendre' et 'redémarrer' • Modification des messages de notification • MAJ des Emoji
Pie	9.0	09/08/2018	<ul style="list-style-type: none"> • Modification d'interface utilisateur • Ajouter d'un bouton de capture écran dans les options avancées • Notification des messages plus riches
Q	10.0	03/09/2019	<ul style="list-style-type: none"> • Meilleur confidentialité • Mode sombre • Navigation par geste

5. L'Architecture d'Androïde

Le diagramme suivant illustre les composants principaux du système d'exploitation Androïde. Chaque section sera décrite dans ce qui suit :

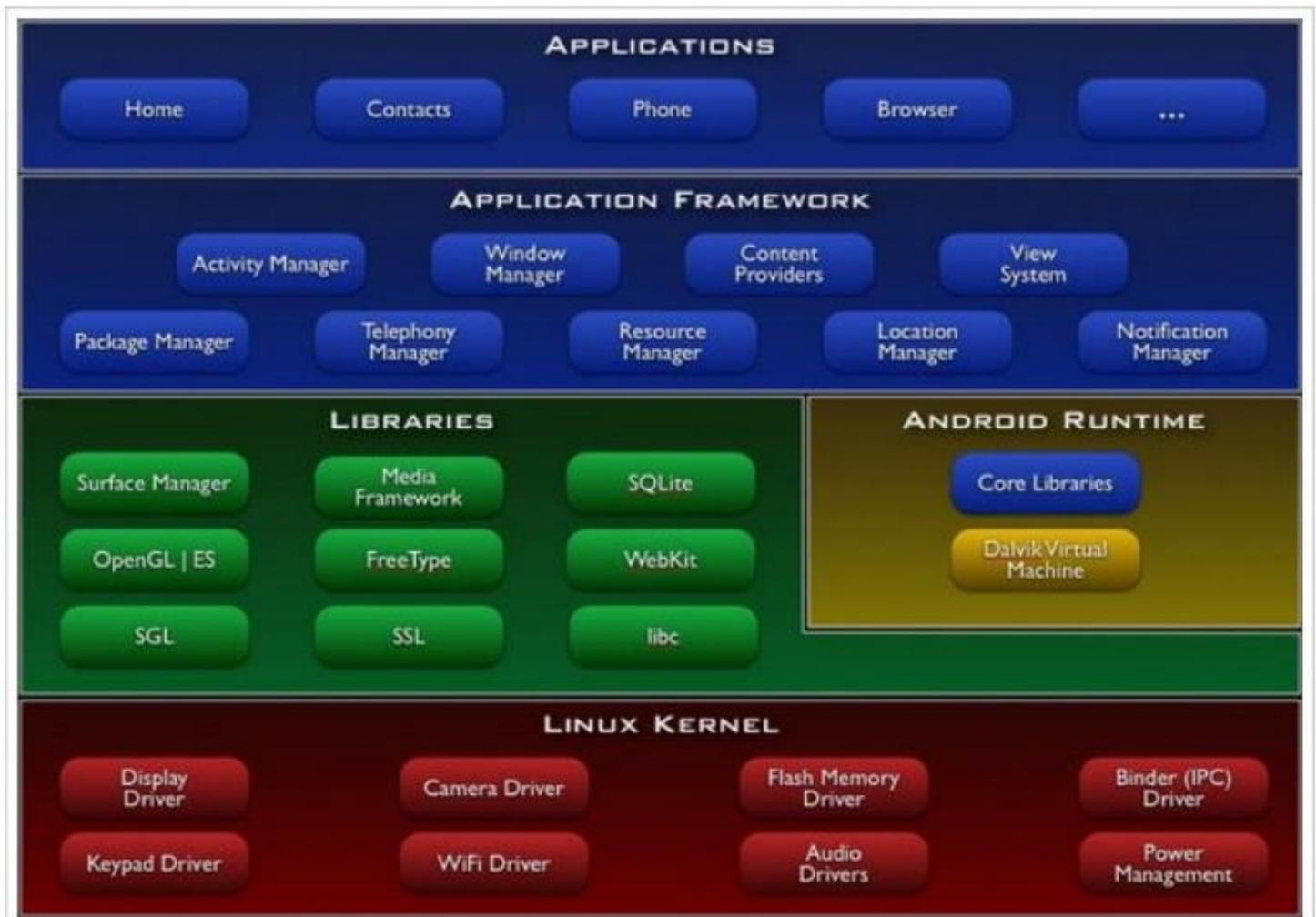


Figure 1.1 : Les principaux composants du système d'exploitation

Androïde est basé sur un kernel linux 2.6. xx.

Au-dessus de cette couche, on retrouve les bibliothèques C/C++ utilisées par un certain nombre de composants du système Android.

Au-dessus des bibliothèques, on retrouve l'Android Runtime. Cette couche contient les bibliothèques cœurs du Framework ainsi que la machine virtuelle exécutant les applications.

Au-dessus de la couche "Android Runtime" et des bibliothèques cœurs, on retrouve le Framework permettant au développeur de créer des applications. Enfin au-dessus du Framework, il y a les applications.

5.1. Applications

Androïde est fourni avec un ensemble d'applications dont un client email, une application SMS, un calendrier, un service de cartographie, un navigateur... toutes écrites en JAVA.

5.2. Framework de développement

En fournissant une plateforme de développement ouverte, Androïde offre aux développeurs la possibilité de créer des applications extrêmement riches et innovantes. Les développeurs sont libres de profiter du matériel périphérique et informations sur la localisation d'accès, exécuter des services d'arrière-plan, définir des alarmes, ajouter des notifications à la barre d'état, etc.

Les développeurs ont un accès complet au même framework API utilisé par les applications de base L'architecture d'application est conçue pour simplifier la réutilisation des composants; n'importe quelle application peut publier ses capacités et n'importe quelle autre application peut alors faire usage de ces capacités (soumis à des contraintes de sécurité appliquées par le framework). Ce même mécanisme permet aux composants d'être remplacés par l'utilisateur. Toutes les applications sous-jacentes forment un ensemble de services et de systèmes, y compris:

- Un jeu extensible de vues qui peuvent être utilisées pour construire une application.
- Des fournisseurs de contenu qui permettent aux applications d'accéder aux données d'autres applications (telles que les Contacts), ou de partager leurs propres données
- Un gestionnaire de ressources.
- Un gestionnaire de notification qui permet à toutes les demandes d'afficher des alertes personnalisées dans la barre d'état.

- Un gestionnaire d'activité qui gère le cycle de vie des applications et propose une navigation commune.

5.3. Bibliothèques

Androïde dispose d'un ensemble de bibliothèques C++ utilisées par les différents composants du système Android. Elles sont offertes aux développeurs à travers le Framework Android. En voici quelques-unes:

Système de bibliothèque C – une mise en œuvre dérivée de BSD de la bibliothèque C standard du système (libc), destinés aux systèmes embarqués basés sur Linux. Comme cela a été dit précédemment, Androïde ne supporte pas la glibc, donc les ingénieurs d'Androïde ont développé une bibliothèque C (libc) nommé Bionlibc Elle est optimisée pour les appareils mobiles et a été développée spécialement pour Android. Les ingénieurs d'Androïde ont décidé de développer une libc propre à la plateforme Android car ils avaient besoin d'une libc légère (la libc sera chargée dans chaque processus) et rapide (les appareils mobiles ne disposent pas de CPU puissant).

La Bionlibc a été écrit pour supporter les CPU ARM, bien que le support x86 soit présent. Il n'y pas de support pour les autres architectures CPU telles que PowerPC ou MIPS. Néanmoins, pour le marché des appareils mobiles, seulement l'architecture ARM est importante.

Cette libc est sous licence BSD. Elle reprend une grande partie du code des glibc issue d'OpenBSD, FreeBSD et NetBSD.

Ces caractéristiques importantes : Elle pèse environ 200Ko, soit la moitié de la glibc L'implémentation des pthreads (POSIX thread) a été complètement réécrite pour supporter les threads de la machine virtuelle Dalvik. De ce fait, la Bionlibc ne supporte pas les threads POSIX

Les exceptions C++ et les "wide char" ne sont pas supportés

Médiathèques – basée sur PacketVideo de OpenCore; les bibliothèques permettant la lecture et l'enregistrement audio et vidéo, ainsi que la gestion des fichiers image, y compris MPEG4, H.264, MP3, AAC, AMR, JPG et PNG.

Le schéma ci-dessous décrit tous les éléments de l'architecture de ces médiathèques:

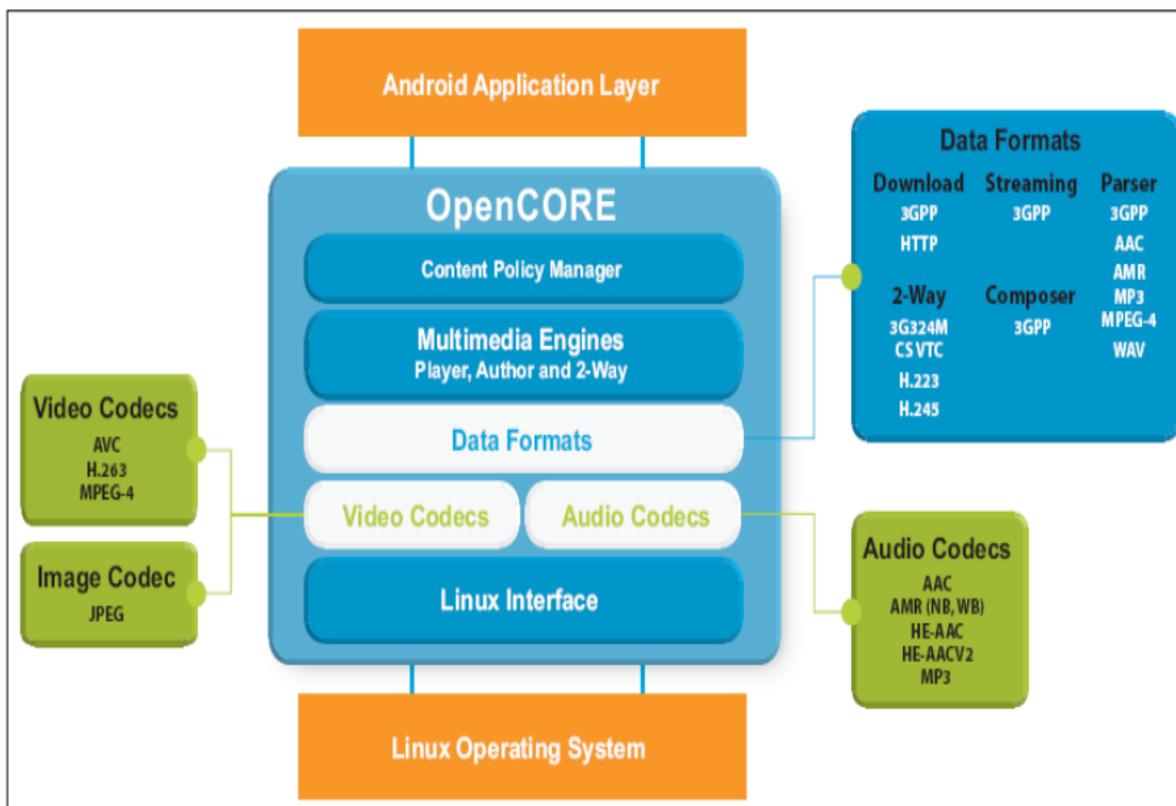


Figure 1.2 : L'architecture de ces médiathèques

- Surface Manager – gère l'accès au sous-système d'affichage et de façon transparente.
- LibWebCore – Le navigateur web présent dans Android est basé sur le moteur de rendu sous licence BSD WebKit.
- WebKit est moteur de rendu, qui fournit une "fondation" sur laquelle on peut développer un navigateur web. Il a été originellement dérivé par Apple du moteur de rendu KHTML pour être utilisé par le navigateur web Safari et maintenant il est développé par KDE project, Apple, Nokia, Google et d'autres. WebKit est composé de deux bibliothèques : WebCore et JavaScript Core qui sont disponibles sous licence GPL. WebKit supporte le CSS, JavaScript, DOM, AJAX. La dernière version a obtenu 100% au test Acid 3. La version de WebKit présent dans Androïde à été légèrement modifiée pour s'adapter aux appareils mobiles. Ainsi, le moteur de rendu basé sur WebKit présent dans Androïde supporte l'affichage sur une colonne.

- SGL – le moteur graphique 2D.
- Bibliothèques 3D – une implémentation basée sur OpenGL ES 1.0 API; les bibliothèques utilisent l'accélération 3D matérielle (si disponible).
- FreeType – bitmap et vectoriel de rendu de police.
- SQLite – un moteur de base de données relationnelles puissant et léger, disponible pour toutes les applications.

5.4. Androïde Runtime

Androïde inclut un ensemble de bibliothèques de base offrant la plupart des fonctionnalités disponibles dans les bibliothèques de base du langage de programmation Java. Chaque application Androïde s'exécute dans son propre processus, avec sa propre instance de la machine virtuelle Dalvik. Dalvik a été écrit pour que le dispositif puisse faire tourner plusieurs machines virtuelles de manière efficace. La machine virtuelle Dalvik exécute des fichiers dans l'exécutable Dalvik (. DEX), un format optimisé pour ne pas encombrer la mémoire. La machine virtuelle est la base de registres et fonctionne grâce aux classes compilées par un compilateur Java et transformées dans le format DEX.

La machine virtuelle Dalvik s'appuie sur le noyau Linux pour les fonctionnalités de base telles que le filetage et la gestion de la mémoire de bas niveau.

5.5. Linux Kernel

Androïde est basé sur un kernel linux 2.6 mais ce n'est pas linux. Il ne possède pas de système de fenêtrage natif (X window system). La glibc n'étant pas supportée, Androïde utilise une libc customisée appelée Bioniclibc.

Enfin, Androïde utilise un kernel avec différents patches pour la gestion de l'alimentation, le partage mémoire, etc. permettant une meilleure gestion de ces caractéristiques pour les appareils mobiles.

Androïde n'est pas linux mais il est basé sur un kernel linux. Pourquoi sur un kernel linux ?

Le kernel linux a un système de gestion mémoire et de processus reconnu pour sa stabilité et ses performances.

Le model de sécurité utilisé par linux, basé sur un système de permission, est connu pour être robuste et performant. Il n'a pas changé depuis les années 70

- Le kernel linux fournit un système de driver permettant une abstraction avec le matériel. Il permet également le partage de bibliothèques entre différents processus, le chargement et le déchargement de modules à chaud.
 - Le kernel linux est entièrement open source et il y a une communauté de développeurs qui l'améliorent et rajoutent des drivers.
- C'est pour les points cités ci-dessus que l'équipe en charge du noyau a décidé d'utiliser un kernel linux

6. Cycle de vie d'une Activity

Les applications Androïde ont un fonctionnement particulier elle réagissent à des changements d'états imposés par le système (démarrage, pause, reprise, arrêt..) elles n'ont aucun contrôle direct sur leur propre cycle de vie .

En cas de besoin, le système peut mettre en pause ou alors complètement arrêter une activité s'il juge nécessaire de le faire, par exemple si l'application consomme trop de ressources processeur ou mémoire. Ce comportement est dit non déterministe.

Chaque application fonctionne dans son propre processus. Le système androïde est responsable de la création et de la destruction des processus et gère ses ressources comme objectif la sécurité mais aussi la disponibilité et la réactivité de l'appareil.

Ainsi un processus consommant beaucoup de ressources processeur, ne pourra entraver d'autres fonctionnalités, comme le réponse à une appel entrant ceci implique qu'un processus peut être tué à n'importe quel moment sans son consentement pour libérer des ressources nécessaires à d'autres applications .

Pour comprendre comment se déroule le cycle d'activité il est important de connaître que une activité fournit un écran avec lequel les utilisateurs peuvent interagir et que une

Application androïde est constituée d'un ensemble d'activités liées entre elles.

Chaque fois qu'une activité est lancée, elle est placée dans une pile appelée back start

Chaque fois que vous lancez une nouvelle activité pour exécuter une nouvelle action, l'activité en premier plan est arrêtée, la nouvelle est démarrée et placée en haut de pile.

Quand vous revenez en arrière, l'activité en premier plan est détruite de la pile précédente est réactivée.

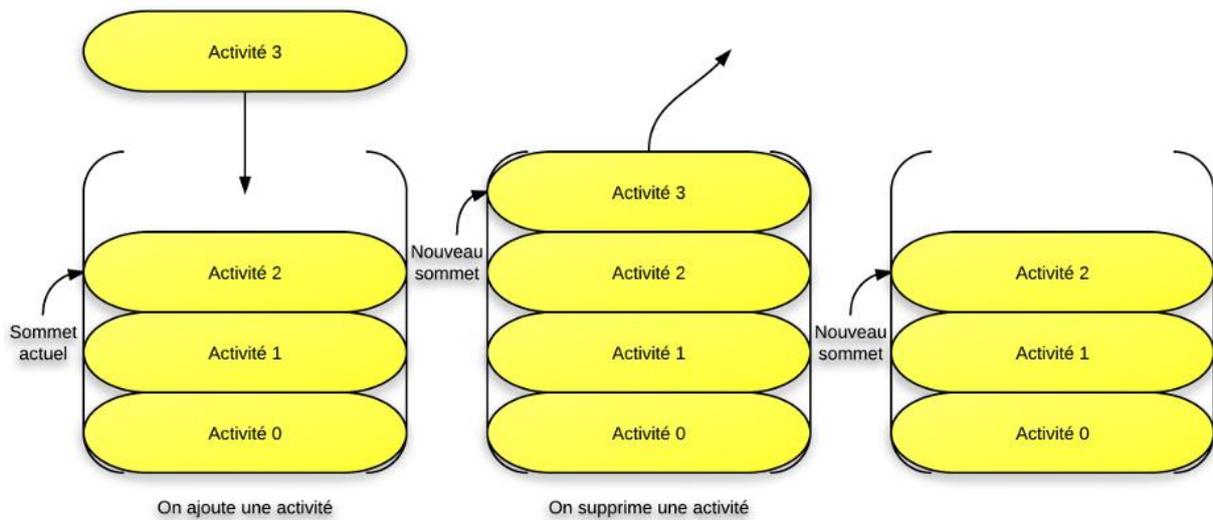


Figure 1.3 Fonctionnement de la pile d'activités.

Le schéma suivant montre le processus de vie d'une application androïde

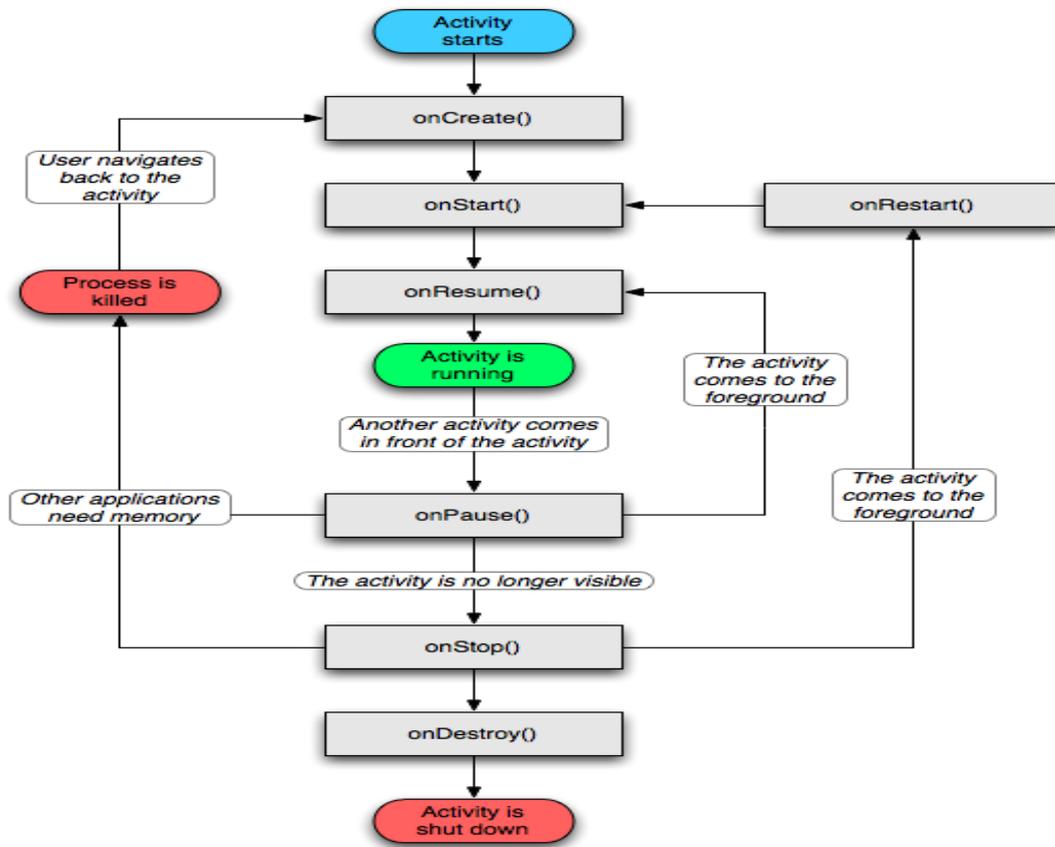


Figure 1.4 Le cycle de vie d'une application

- Lorsque l'activité est lancée, le système Androïde appelle les méthodes **onCreate**, **onStart** et **onResume**
- Lorsque l'activité s'arrête, le système Androïde appelle les méthodes **onPause**, **onStop** et **onDestroy**
- Lorsque l'activité n'est plus au premier plan, mais qu'elle est tout de même affichée **onPause** est appelée. Lorsque l'activité retourne au premier plan, **onResume** est appelée.
- Lorsque l'activité n'est plus affichée, **onResume** et **onStop** sont appelées. Lorsque l'activité est de nouveau affichée, **onRestart**, **onStart** et **onResume** sont appelées.
- Lorsque le système Android décide de tuer votre application, il appelle la méthode `onSaveInstanceState` qui vous permet de sauvegarder l'état de votre activité. Cet état sera passé en paramètre à la méthode **onCreate**, afin que vous puissiez restaurer l'état de l'activité.
- Le système Android s'occupe lui-même de sauvegarder et restaurer l'état des vues des layouts, à condition que ces vues aient chacune un ID unique dans le layout.
- Le changement de configuration comme le changement d'orientation de l'écran provoque un redémarrage de l'activité, vous devez donc sauvegarder et restaurer l'état dans ce cas.

7. Les outils de développement Android

Les différents outils de développement Androïde sont :

7.1. Le SDK Androïde

Le Kit de développement logiciel Android (Android SDK) contient les outils nécessaires pour créer, compiler et déployer les applications Android. La plupart de ces outils sont en ligne de commande.

7.2. Le débogueur « ADB »

Le SDK Androïde contient un débogueur appelé « Android debug bridge » ou aussi « adb », qui permet de connecter un appareil Android virtuel ou réel, dans le but de gérer le périphérique ou de déboguer votre application.

7.3. Les IDE « Android Developer Tools » et « Android Studio »

Google propose deux environnements de développement intégrés (IDE) pour développer de nouvelles applications.

a) Eclipse

Les outils de développement Android sont basés sur l'IDE Eclipse. ADT est un ensemble de composants (plug-ins), qui étendent l'IDE Eclipse avec des capacités de développement Android.

b) Android Studio

Google propose également cet IDE appelé pour la création d'applications Androïde quel que soit le terminal sous-jacent (smartphone, tablette, montre, TV...) et qui est basé sur l'IDE IntelliJ.

Le tableau qui suit, compare entre les deux IDE :

Tableau 1.2 Tableau comparatif entre Eclipse et Android Studio

	Eclipse	Android Studio
Facilité d'installation	Moyen	Simple
Langue	Nombreuse	Anglais
Performance	Peut être lourd	Rapide
Système de built	Ant	Gradle
Génération de variante et de multiple APK	Non	Oui
Android Code completion et refactoring	Base	Avancé
Editeur d'interface graphique	Oui	Oui
Signature d'APK et gestion de Keystore	Oui	Oui
Support NDK	Oui	A venir

7.4. La machine virtuelle Dalvik

Dalvik est une machine virtuelle incorporée dans le système d'exploitation Android. Destinée à permettre l'exécution simultanée de plusieurs applications sur un appareil de faible capacité (peu d'espace mémoire et peu de puissance de calcul).

7.5. La runtime Android (ART)

Les dernières versions d'Androïde introduisent une nouvelle machine, le runtime Android. Le moteur qui permet l'exécution des applications pour Android.

8. Conclusion

Nous avons analysé durant ce chapitre les différents composants principaux d'androïde ainsi que les diverses fonctionnalités offerts par cette plateforme.

Et au vu des points forts qui viennent d'être énumérés. Nous avons constaté qu'androïde représente un système d'exploitation mobile puissant, flexible et sécuritaire.

CHAPITRE02

La Traduction Automatique

1. Introduction

La traduction automatique (TA) qui fait partie d'un domaine plus large, le traitement automatique des langues (TAL) a pour objectif de traiter des données linguistiques (textes) exprimées dans une langue dite 'naturelle', pour transférer ces données dans une autre langue (naturelle). Ceci signifie que la T.A n'est autre que le transfert du 'fond' et de 'forme' d'un texte, pris comme représentant de la langue de départ, d'un système linguistique vers un autre avec tout ce que cela implique comme somme d'informations et de connaissances à projeter vers ce système linguistique à partir d'un système linguistique donné de départ.

L'objectif de la T.A c'est la conception de logiciels ou programmes, capable de traiter de façon automatique des données linguistiques, c'est-à-dire des données exprimées dans une langue « naturelle ». Ces données linguistiques peuvent être des textes écrits, ou bien des dialogues écrits, ou encore des unités linguistiques de taille inférieure à ce que l'on appelle habituellement des textes (des phrases, groupes de mots, ou simplement des mots isolés).

Un traitement automatique est une suite d'actions effectuée par la machine dans un certain ordre chronologique (traiter un objet linguistique de façon automatique).

2. Historique

On peut distinguer trois grandes étapes dans l'historique de la TA : la première (fin des années 40 jusqu'en 1965) a été marquée successivement par l'enthousiasme des débuts, puis un certain désenchantement, la seconde (65 à 75) a été une période de relative stagnation, due aux contrecoups du rapport ALPAC, et la troisième enfin de 1975 à nos jours témoigne d'une vitalité nouvelle, notamment en matière de recherche et développement.

3. Les différents systèmes de la TA

Le travail de nombreuses équipes aboutit à une diversification des méthodes. Si la traduction directe, d'abord conçue comme un mot-à-mot à peine amélioré, puis intégrant une part variable d'analyse syntaxique- domine largement, s'ajoutent deux autres approches. D'où la typologie suivante, qui perdurera.

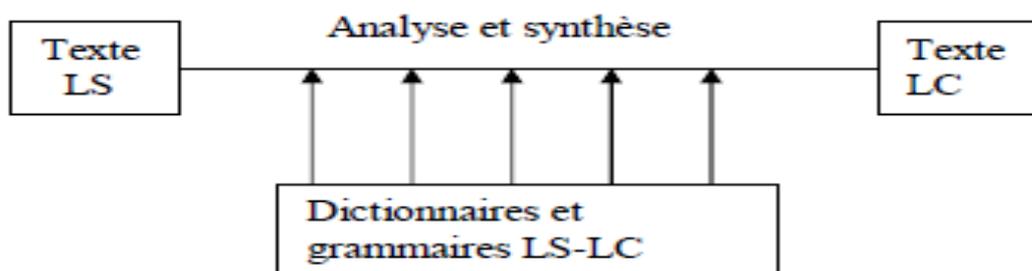


Figure 2.5 : Système traduction direct

Un système direct n'opère que les analyses indispensables pour la traduction d'une langue dans une autre : l'analyse de la syntaxe et du vocabulaire de la langue source n'est pas poussée au-delà de ce qui est indispensable pour choisir les expressions appropriées et l'ordre des mots de la langue (figure 2.5).

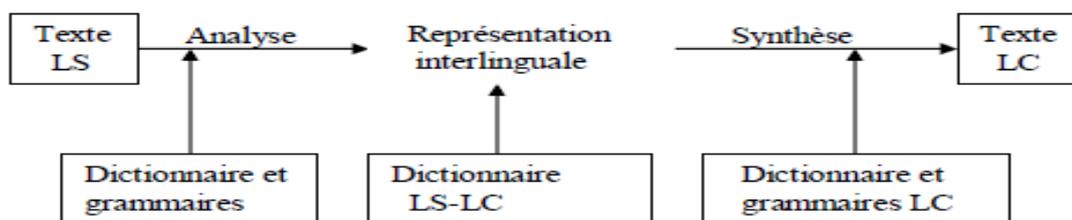


Figure 2.6 : Système interlingual

Un système interlingual procède en deux étapes nettement séparées : l'analyse complète du texte source en une représentation interlinguale et la synthèse du texte cible à partir de celle-ci (figure 2.6).

Dans un contexte de multi traduction, cette option est théoriquement beaucoup plus économique : si l'on doit traduire dans n langages, il faut $n(n-1)$ programmes de traduction directe et seulement $2n$ programmes interlinguaux. L'interlingual utilisée peut être une langue logique artificielle, une langue semi-naturelle comme l'espéranto, un ensemble de primitives sémantiques universelles ...

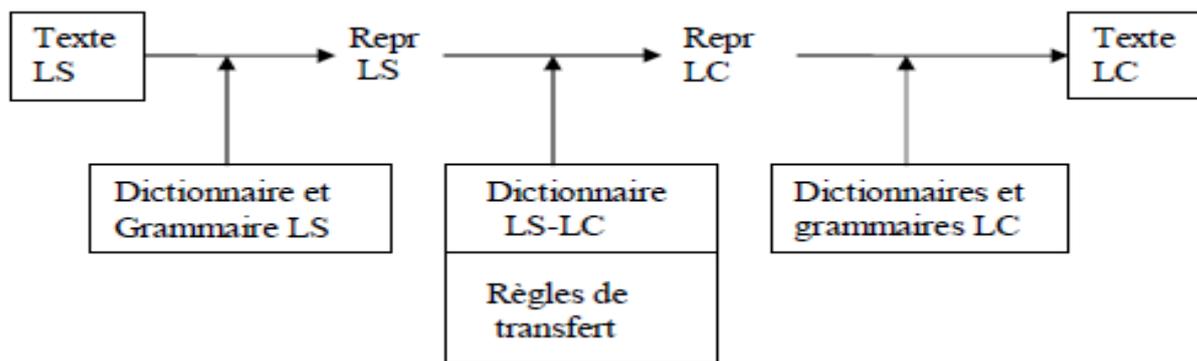


Figure 2.7 : Système par transfert

On appelle système par transfert un système qui comporte trois étapes, la première transforme le texte source en représentations de transfert de la langue source, la seconde transforme celles-ci en représentation de transfert de la langue cible, à partir des quelles est opérée la synthèse du texte en langue cible.

4. Les connaissances nécessaires pour la TA :

4.1. Les dictionnaires

Les traducteurs humains utilisent les dictionnaires comme source d'information pour les langues LS et LC.

Le type de dictionnaire souvent utilisé par l'homme est le dictionnaire bilingue, qui connecte les unités (mots) de la LS et la LC. Un point important à remarquer est que ces dictionnaires bilingues sont conçus pour l'usage humain.

La situation est peu différente quand le dictionnaire est utilisé par le programme d'un ordinateur.

On peut illustrer ces types de dictionnaire et les modules d'exploitation, comme suit : on suppose que le système reçoit en entrée un texte français : "le livre est sur la table"(1).

La traduction en anglais de (1)est : "The book is on the table." (2). Le dictionnaire nécessaire pour la performance de cette traduction est le suivant :

Table 2.3 : Table des mots français anglais

Français	Anglais
sur	on
livre	book
le	the
la	The
est	Is
table	table

Le programme de traduction a fourni une substitution des mots français en anglais un par un. L'analyseur à ce niveau est lexical ou orthographique, ie: fondé sur l'aspect des mots ou aspect morphologique.

Peut-on utiliser le même système pour passer de l'Anglais vers le Français ? Non, parce que dans ce cas plusieurs relations vont du mot "the" à "le" et "la".

Les connaissances inscrites dans ce dictionnaire sont insuffisantes pour résoudre cette ambiguïté et il faut ajouter d'autres connaissances pour faire le choix adéquat.

4.2. Les connaissances syntaxiques

L'un des problèmes de l'analyse automatique que résout la syntaxe est le processus de désambiguïsation relevant des différentes catégories d'un mot.

La syntaxe décrit la structure des mots et des connexions associées dans une phrase.

L'analyseur à ce niveau est grammatical, fondé sur les règles d'agencement des groupes de mots et des mots entre eux.

Comme par exemple : "The coach lost a set.", sans connaissance de la structure de la phrase anglaise, il est impossible de savoir si :

- 1- le mot "coach" est un nom ou un verbe,
- 2- le mot "lost" est un verbe ou un adjectif,
- 3- le mot "set" est un nom ou un verbe ou un adjectif.

Ainsi les connaissances syntaxiques de la langue anglaise, nous permettent, dans ce cas d'ambiguïtés et de faire le choix correct pour la compréhension.

Pour ce faire, la reconnaissance de la structure syntaxique de la phrase par des règles de réécriture permet d'identifier les différents mots, unités ou groupes de la phrase (syntagmes nominaux, syntagmes verbaux...).

Ce type de connaissance est enregistré dans la grammaire anglaise ou dans toute autre grammaire de la LS d'un système de TA.

Les règles de réécritures pour notre exemple :

S ^NP + VP

NP D + N

VP V + NP

Avec: S: (Sentence) phrase,

NP : (Noun Phrase) Syntagme

Nominal, VP : (Verbal Phrase)

Syntagme verbal, D : (Determiner)

Déterminant, N : (Noun) nom et

V : (Verb) verbe.

Le résultat de l'analyse est :

K S[NP[D(the),N(coach)],VP[V(lost),NP[D(a),N(set)]]].

4.3. Les connaissances sémantiques

La sémantique s'occupe du signifié des mots et des connexions associées dans une phrase. L'analyseur est fondé sur des liens de signification entre les mots, comme les anaphores, et traite les problèmes de polysémies (ie. les mots ayant plusieurs significations). L'une des premières tentatives dans le domaine de l'analyse sémantique est d'utiliser les représentations des connaissances d'un langage naturel pour un système de traduction. Ceci a été réalisé à l'université de YALE aux USA par les travaux de Shank en 1975: les Dépendances Conceptuelles (DC) dans un langage de représentation des connaissances.

Les DC sont utilisées pour représenter le sens de la phrase ; les dictionnaires bilingues utilisés dans ces systèmes prennent en considération des choix de marques ou de mailles, pour représenter les connaissances sémantiques.

Ces mailles sont utilisées lors de la phase de génération pour que le système connecte la signification au mot adéquat dans la LC.

On illustre la représentation en DC d'une phrase anglaise :

Phrase: "Mary hit John." DC

Phrase: (event EV001

(action PROPEL)

(agent Mary)

(object John)
 (instrument *?* unknown)
 (force * above-average *)
 (intentionality * positive *)).

4.4 Les connaissances contextuelles

La pragmatique s'occupe de la cohérence générale des éléments sémantiques entre eux.
 L'analyseur est fondé sur le bon sens.

Exemple : « Alors que Marie parlait, Jean buvait ses paroles. »

Morphologie : tous les mots sont corrects du point de vue orthographique.

Syntaxe : les règles de grammaires sont correctement appliquées.

Sémantique : l'anaphore « ses » se rapporte à Marie, ie: Alors que Marie parlait, Jean buvait les paroles de Marie.

Pragmatique : le verbe « boire » implique un complément d'objet direct « liquide » or « les paroles » n'est pas un « liquide » donc il y a une erreur pragmatique !

Par contre, si l'on sait que le contexte est « poétique », alors il s'agit d'une métaphore et la phrase est correcte. Les connaissances contextuelles permettent d'éliminer des ambiguïtés provenant de causes très diverses :

a) L'ambiguïté peut venir de la grammaire :

L'ambiguïté peut venir de la grammaire qui n'a pas pu choisir entre deux analyses. On illustre ce cas, par l'exemple suivant :

"Je viens d'écrire un article sur mon bureau."

Le syntagme prépositionnel "sur mon bureau", peut se rattacher à :

- 1- " article " ou
- 2- " écrire ".

b) L'ambiguïté peut venir du sens des mots :

Dans la phrase : "J'achète un terrain à mon voisin.", le verbe "acheter" exprime que le sujet reçoit la possession du complément direct qui était au paravent la possession du groupe prépositionnel.

c) La pragmatique ne peut pas tout résoudre :

Dans la phrase : " Pierre a acheté un bijou à ma voisine.", nous avons ici non seulement deux événements peu probables mais aussi des connaissances générales.

5. la traduction automatique et la langue arabe

Le traitement automatique de la langue arabe n'a pas encore atteint le même niveau de recherche et de résultats que les autres langues (notamment l'Anglais et le Français). Cela n'empêche pas que des tentatives, pas toujours fructueuses, furent menées qu'on va essayer d'en donner un aperçu pour mieux cerner les difficultés qui l'entourent. Ces difficultés découlent presque toutes de la nature de la langue arabe qui est caractérisée par sa forte morphologie.

Dans la même langue, le même terme peut avoir plusieurs sens différents, et ceci dépend du contexte dans lequel ce terme est utilisé. L'exemple le plus classique dans ce contexte pour le français est celui de l'avocat, qui peut être le fruit ou le métier. En arabe par exemple le mot عين (Eyn) peut être traduit par un œil ou par une source (d'eau) ou par un espion, etc. Un même mot peut être utilisé dans deux contextes différents avec deux sens différents.

Dans le domaine de la traduction automatique, ce problème est très connu et c'est un problème de recherche toujours d'actualité. La probabilité de traduire correctement un mot polysémique est très faible si le système de traduction utilisé n'est pas approprié au contexte du texte à traduire. Par exemple si un système de traduction est appris sur des données de type politique, la traduction d'un document sur l'industrie agroalimentaire avec ce système donnera de mauvaises performances (avocat le fruit sera traduit par l'avocat le métier). Nous nous intéressons dans ce manuscrit à étudier ce phénomène, et plus particulièrement à l'amélioration de nos systèmes de traduction automatique et à leur adaptation au type de données à traduire.

La langue arabe est une langue morphologiquement riche et complexe. C'est une langue sémitique, qui a la particularité d'avoir un vocabulaire à base de racines de mots trilitères consonantiques. À cette forme de base, peuvent s'ajouter des préfixes, suffixes, ainsi que des clitiques. Les clitiques et affixes sont agglutinés au mot de base pour former des mots de plus en plus complexes voire des phrases, comme par exemple le mot و سيكتبها (Et il va l'écrire),

qui est constitué de deux proclitiques و et س, un préfixe ي et un enclitique ها – sachant que le pronom l' réfère à un nom féminin.

L'absence de voyellation de l'arabe dans les textes courants provoque une inflation de solutions d'analyses des mots composés lorsque ceux-ci sont analysés indépendamment les uns des autres. La langue arabe est modélisée par un ensemble de règles linguistiques et grammaticales complexes.

6. Conclusion

Pour faire le TAL de l'arabe, il faut :

- comprendre les faits linguistiques ;
- les modéliser de façon adéquate ;
- maîtriser des techniques formelles et computationnelles diverses !

La linguistique est importante !

Dans ce chapitre, nous avons fait une étude sur la traduction automatique tout en présentant un bref historique, les différents systèmes de la TA, Ainsi que les connaissances nécessaire et on a fini le chapitre une aperçu sur la TA et la langue arabe.

CHAPITRE 03

Conception

1. Introduction

La conception est une étape préliminaire et primordiale qui doit précéder l'étape de développement de toute application informatique. Pour décrire la conception de l'application, on commencera avec les diagrammes de cas d'utilisation et les diagrammes de séquence. Par la suite on passera au diagramme de classes.

2. Diagramme de cas d'utilisation

2.1. Définition de diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est une représentation du comportement du système de point de vue de l'utilisateur, c'est une définition des besoins qu'attend un utilisateur du système, il contient tous les cas d'utilisation en liaison directe ou indirecte avec les acteurs.

- **Les acteurs** : représentés sous la forme de petits personnages nommés.

Un acteur représente le rôle joué par une personne ou une classe qui interagit avec les système.

- **Cas d'utilisation** : est un classificateur qui modélise une fonctionnalité d'un système ou d'une classe. L'instanciation d'un cas d'utilisation se traduit par l'échange de message entre le système et ses acteurs.

2.2. Diagramme de cas d'utilisation générale

La figure 2.6 représente le diagramme de cas d'utilisation générale qui fait preuve d'un acteur appelé client Androïde, qui admet les fonctionnalités suivantes :

- **Translate** : le client peut traduire un mot ou bien une phrase en 3 langues différentes.
- **Add in favoris** : il peut ajouter un mot ou un texte choisi aux favoris.
- **History** : le client peut revisiter les mots traduits précédemment.
- **Addword** : le client peut ajouter un nouveau mot dans la base de données (apprentissage de base de données).

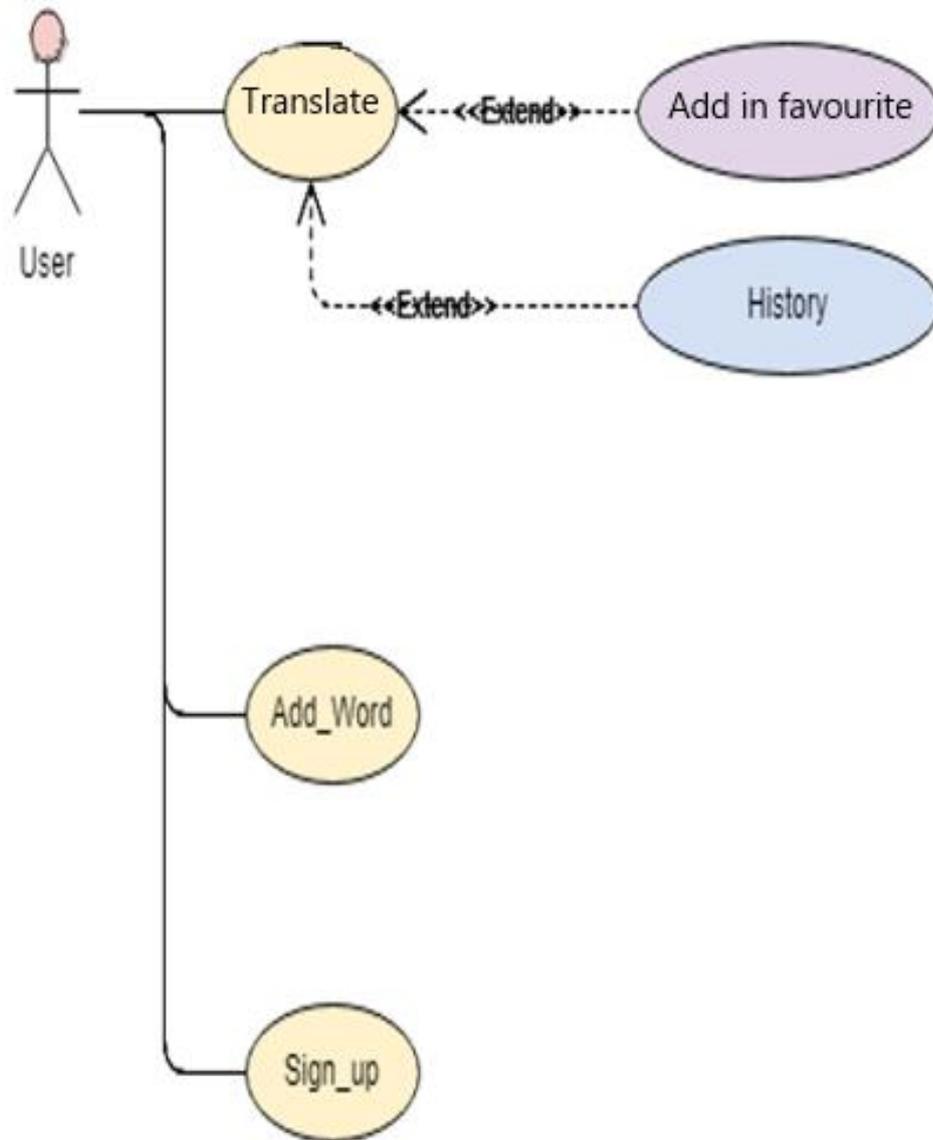


Figure 3.6 : Diagramme de cas d'utilisation

3. Diagramme de séquence

3.1. Définition de diagramme de séquence

Le diagramme de séquence représente la succession chronologique des opérations réalisées par les acteurs. Il montre les interactions entre les objets, en montrant les messages qu'ils échangent entre eux ordonnés dans le temps.

Les composants d'un diagramme de séquence sont les suivants :

- **Les objets** : sur un diagramme de séquence, les objets apparaissent toujours dans la partie supérieure, ce qui facilite l'identification des classes qui participent à l'interaction.
- **Le message** : élément de communication unidirectionnel entre objets qui déclenche une activité dans l'objet destinataire. La réception d'un message provoque un événement dans l'objet récepteur. La flèche pointillée représente un retour au sens UML. Cela signifie que le message en question est le résultat direct du message précédent.

3.2. Diagramme d'interaction authentification d'un utilisateur

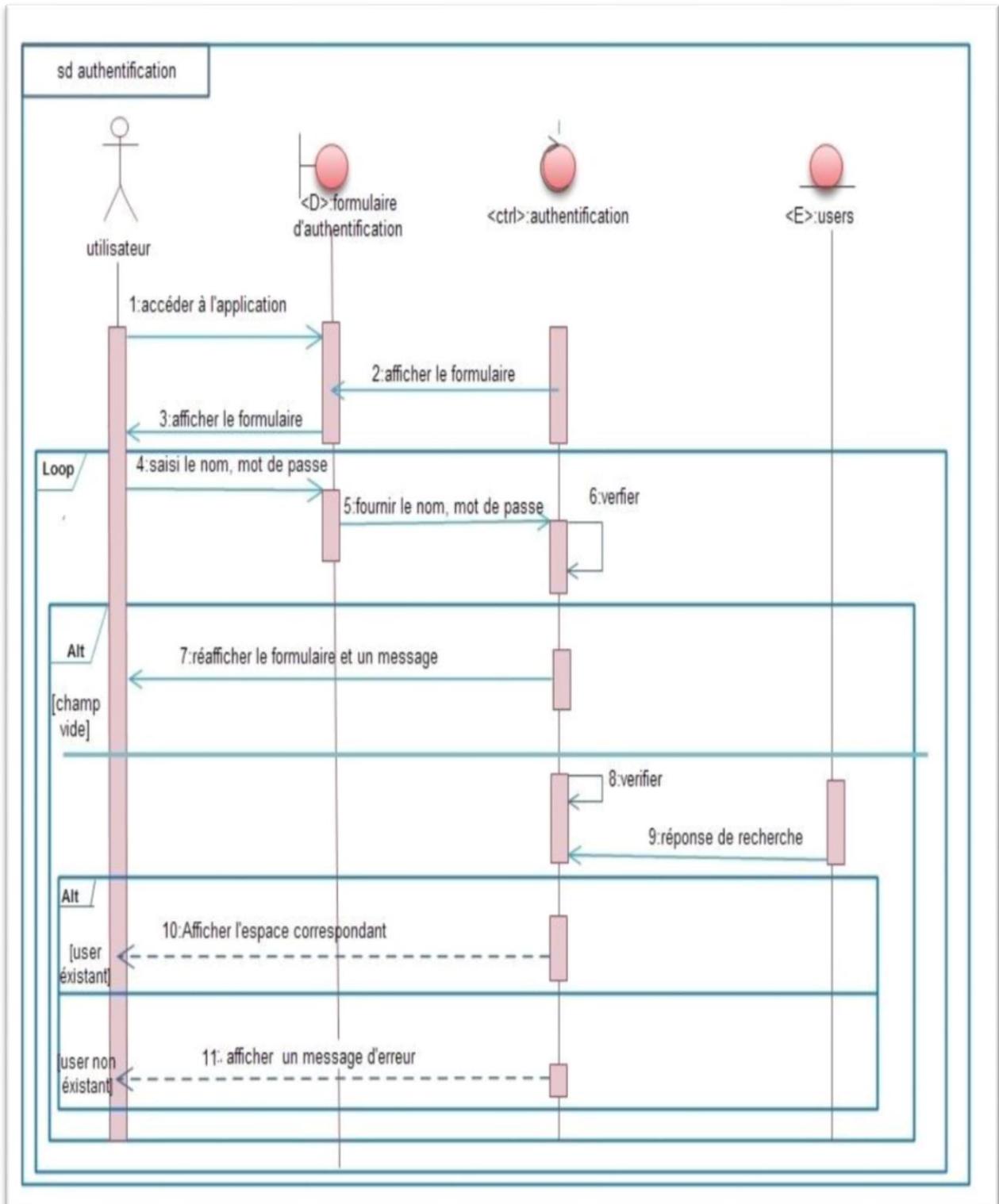


Figure 3.7 : Diagramme d'interaction authentification d'un utilisateur

3.3. Diagramme de séquence "Transalte"

Le diagramme suivant (figure 8) permet d'accéder à la page de traduction, d'où l'utilisateur peut tout d'abord saisir le texte qui veut le traduire en 3 langues, puis il sélectionne "Translator" qui retourne le texte traduit. L'utilisateur peut aussi ajouter le texte traduit au favorite par le cliquer sur l'icône « cœur ».

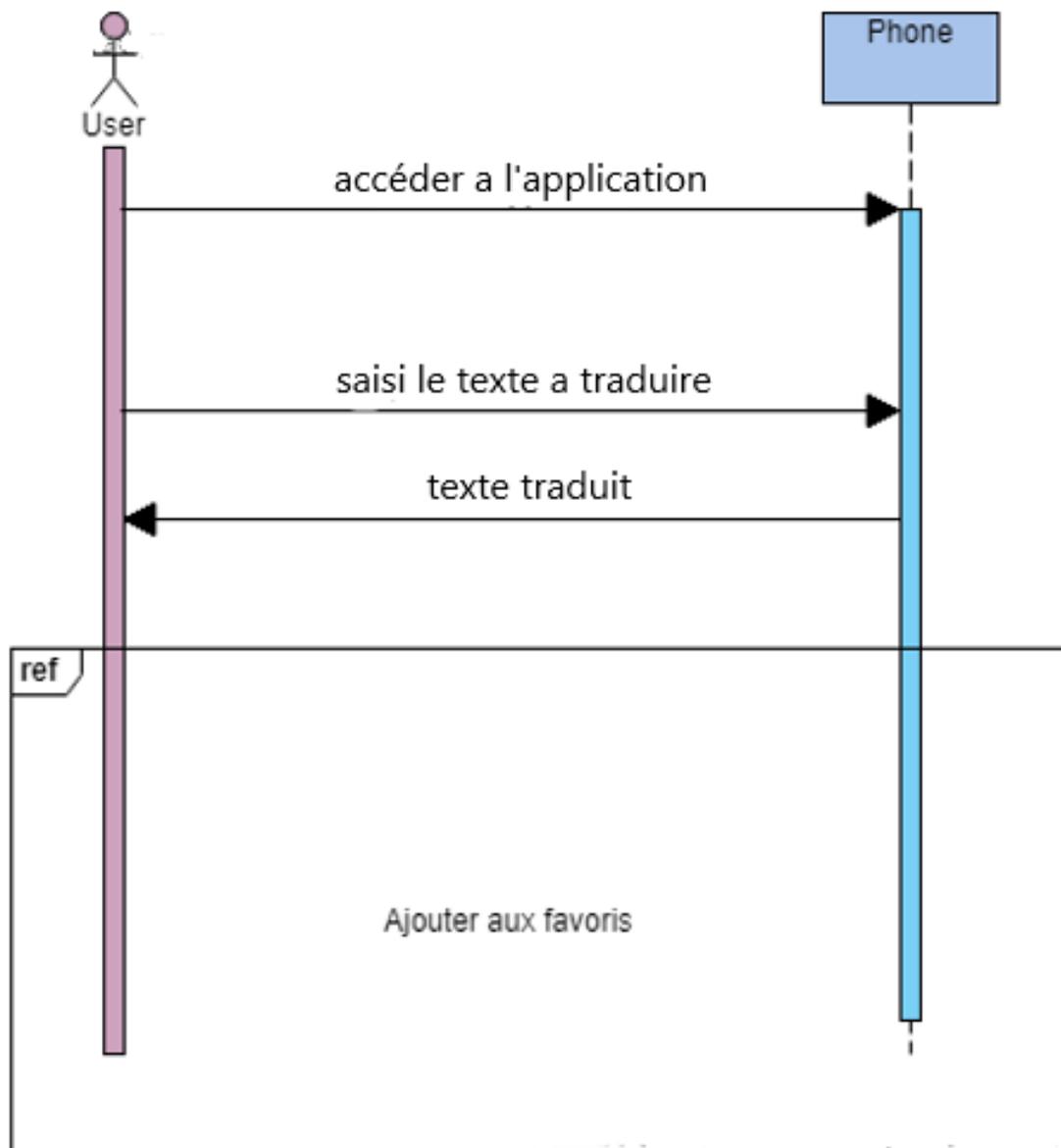


Figure 3.8 : Diagramme de séquence « Translate »

3.4. Diagramme de séquence "History"

Le diagramme suivant (figure 9) permet d'accéder à la page d'historique, d'où l'utilisateur peut consulter les textes traduits précédemment.

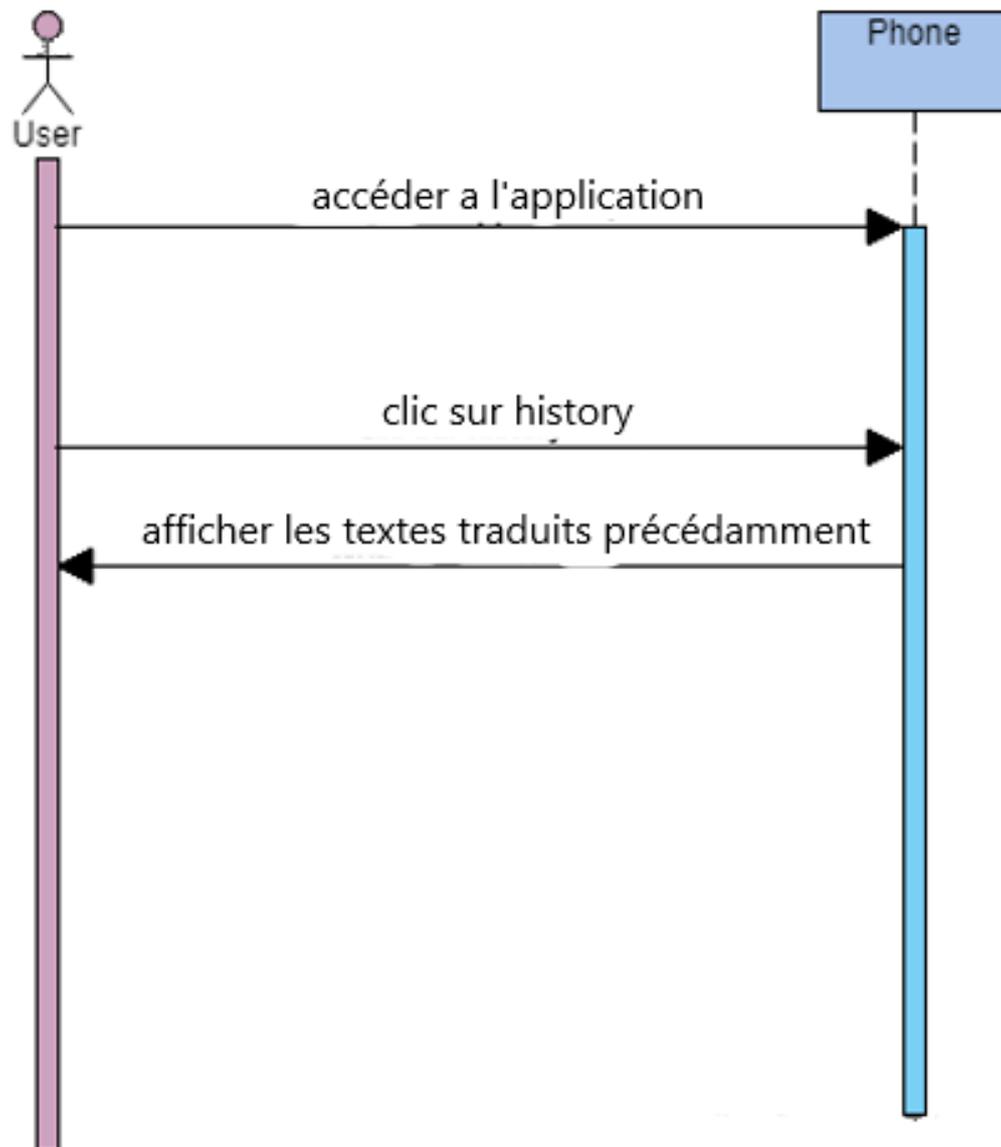


Figure 3.9 : Diagramme de séquence « History »

3.5. Diagramme de séquence "Addword"

Le diagramme suivant (figure 3.10) permet d'ajouter des nouveaux mots à la base de données.

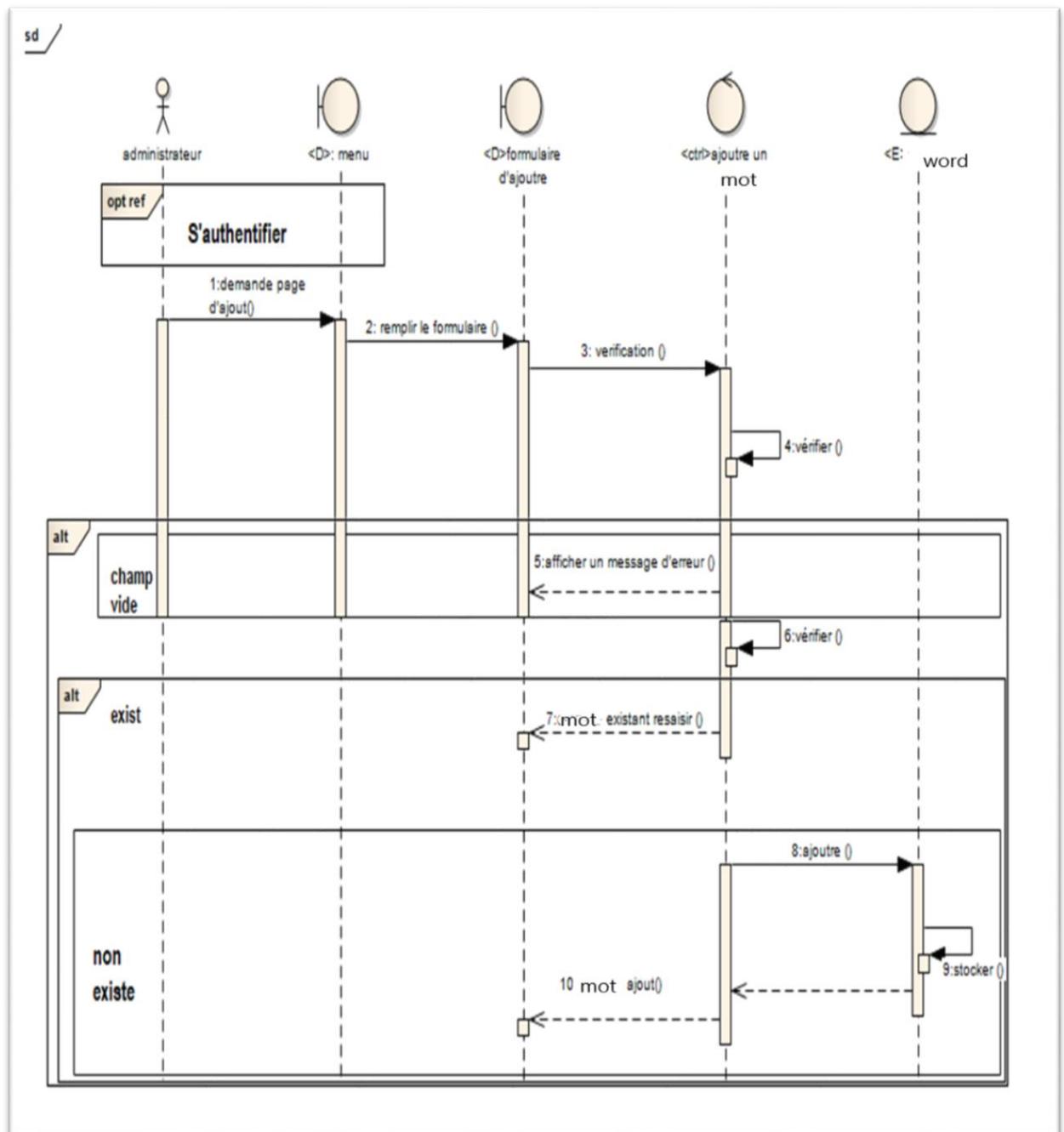


Figure 3.10 : Diagramme de séquence « Addword »

4. Diagramme de classe

4.1. Définition de diagramme de classe

Le diagramme de classes est le point central dans le développement orienté objet. Côté analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. Côté conception, le diagramme de classes représente la structure d'un code orienté objet ou, à un niveau de détail plus important, les modules du langage de développement.

Une classe est composée des attributs et opérations, et admet des associations et multiplicité.

4.2. Diagramme de classe de l'application

Comme le montre la figure 3.5 ci-dessous, le diagramme de classe de l'application admet six classes : User, Transalte, History, favoris, Add_word, word, Text.

User peut accéder aux différentes classes selon le choix.

User peut ajouter des mots à Add_word.

User peut ajouter Text aux favoris.

Chaque Texte contient une description.

Chaque Texte contient un word correspondant.

Chaque Texte peut être ajouté aux favoris.

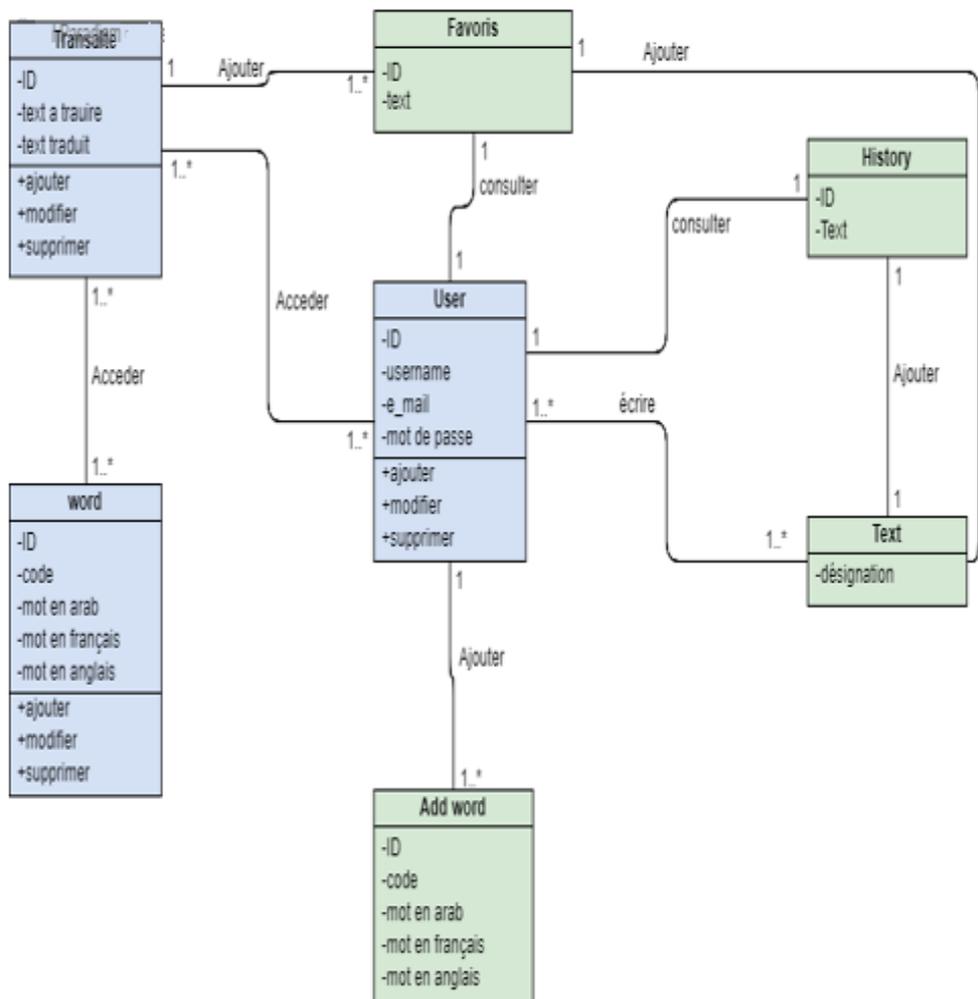


Figure 3.11 : Diagramme des classes

5. Conclusion

Tout au long de ce chapitre, nous avons détaillé la conception de notre application à travers le diagramme de classes ainsi que les diagrammes de séquences associées afin que la phase réalisation et la mise en place de l'application soit plus souple et plus aisée. Le chapitre suivant mettra en évidence, le fruit de ce passage et les différents résultats du développement de l'application demandée.

CHAPITRE 04

Réalisation

1. Introduction

Pour pouvoir mener à bien un projet informatique, il est nécessaire de choisir des technologies permettant de simplifier sa réalisation. Pour cela, après avoir complété l'étude conceptuelle dans le chapitre précédent, nous allons aborder la partie implémentation dans ce qui suit. Nous commençons par présenter l'environnement matériel et logiciel, et ensuite, l'état de réalisation.

2. Environnement du travail :

Le choix de système Androïde OS et les programmes de développement est requis pour réaliser notre application, Ce chapitre couvre les prés requis matériels et logiciels qui sont utilisés pour la réalisation de notre application.

2.1. Environnement matériel :

Pour la réalisation de l'application :

- Un micro portable pour le développement ayant les caractéristiques suivantes :
 - Intel core i3 2.53 GHz.
 - 4 Go de mémoire vive.
 - Windows 7 - 64 bits.
- Deux Smartphone Samsung S5 et Condor L2 pour réaliser les tests.

2.2. Environnement logiciel :

- Android studio.
- SDK de l'android.
- ADT.
- visual-paradigm online pour la conception des diagrammes et des figures.
- MySQL

3. Langages de programmation :

3.1. Le langage JAVA

Est un langage de programmation très utilisé, notamment par un grand nombre de développeurs professionnels, ce qui en fait un langage incontournable actuellement

3.2. SQLite

SQLite est un système de gestion de bases de données intégrée dans chaque appareil Androïde. L'utilisation d'une base de données SQLite sous Android ne nécessite pas de configuration ou d'administration de la base de données. On doit uniquement définir les instructions SQL pour créer et mettre à jour la base de données. Ensuite, celle-ci est gérée automatiquement par la plate-forme Androïde.

➤ **Avantage :**

Il ne nécessite pas de base MySQL sur le serveur, vous utilisez votre propre stockage, un fichier SQLite. On peut sauvegarder la base simplement en téléchargeant le fichier. On peut l'utiliser localement avec des programmes classiques écrit en C ou PHP.

➤ **Inconvénient**

L'accès aux bases de données SQLite peut ralentir si la taille devient importante. Plusieurs utilisateurs ne peuvent modifier simultanément une base. L'extension n'est pas activée par tous les hébergements mutualisés.

3.3. XML

Extensible Markup Language est un langage informatique de balisage générique. Il sert essentiellement à stocker/transférer des données de type texte Unicode structurées en champs arborescents.

3.4. Firebase

Firebase est un ensemble de services d'hébergement pour n'importe quel type d'application. Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale, et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel, ou nous avons utilisé pour **l'authentification Facebook et Google.**

4. Description de l'interface de l'application

Dans ce qui suit nous présentons les différentes interfaces de l'application en citant les détails de chaque imprime écran.

4.1. Interface « Loading »

Cette interface s'affiche lorsque lancé de l'application.



Figure 4.14 : Interface « Loading ».

4.2. Interface « TranslatPage »

C'est l'interface principale, elle s'affiche lors du lancement de notre application et elle est sensée à capter l'attention des clients par le choix des couleurs, l'objectif de cette page est de faire la traduction et ajouter les mots préférés aux favoris.

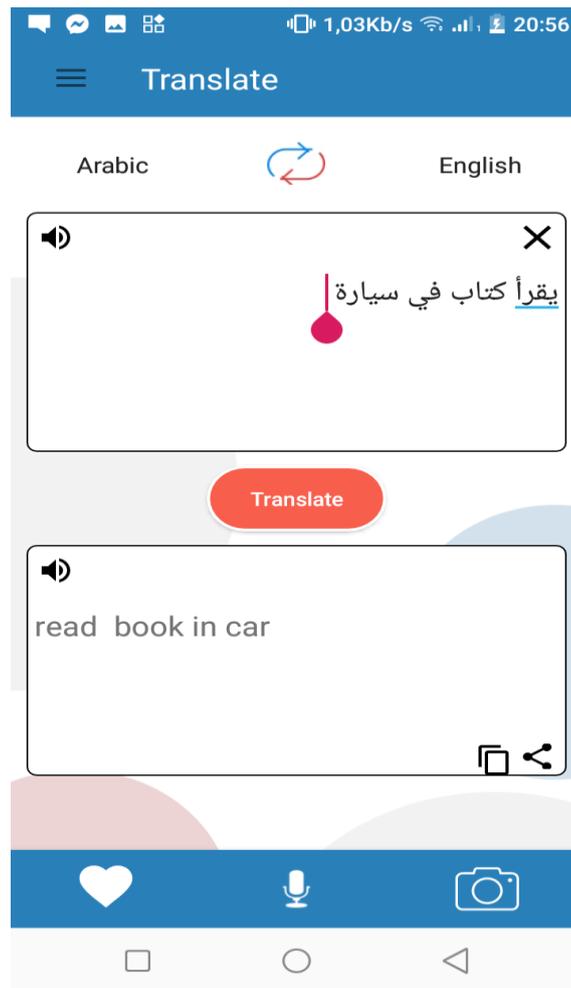
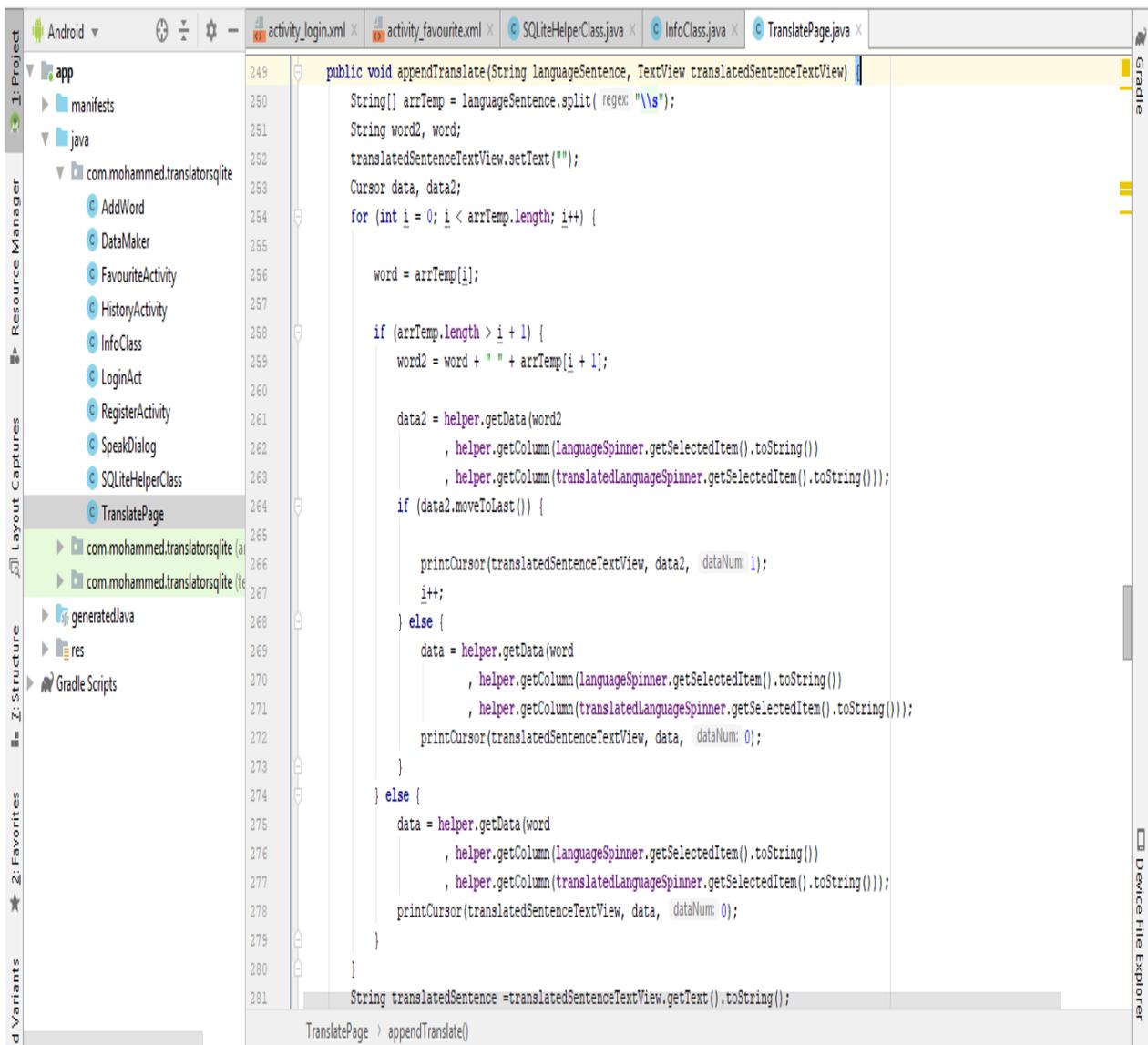


Figure 4.15 : Interface « TranslatPage».

Nous avons trouvé dans cette interface :

- deux champs le premier pour saisir le texte et la deuxième donne la résultat de traduction.
- Bouton speaker : pour écouter le texte saisi ou bien le texte traduit (en peut changer la vitesse de la lecture).
- Bouton cœur : pour ajouter le texte dans la page favorite.
- Bouton microphone : pour dicter le texte.

4.3. Code de traduction



```
249 public void appendTranslate(String languageSentence, TextView translatedSentenceTextView)
250     String[] arrTemp = languageSentence.split( regex "\\s" );
251     String word2, word;
252     translatedSentenceTextView.setText("");
253     Cursor data, data2;
254     for (int i = 0; i < arrTemp.length; i++) {
255
256         word = arrTemp[i];
257
258         if (arrTemp.length > i + 1) {
259             word2 = word + " " + arrTemp[i + 1];
260
261             data2 = helper.getData(word2
262                 , helper.getColumn(languageSpinner.getSelectedItem().toString())
263                 , helper.getColumn(translatedLanguageSpinner.getSelectedItem().toString()));
264             if (data2.moveToFirst()) {
265
266                 printCursor(translatedSentenceTextView, data2, dataNum: 1);
267                 i++;
268             } else {
269                 data = helper.getData(word
270                     , helper.getColumn(languageSpinner.getSelectedItem().toString())
271                     , helper.getColumn(translatedLanguageSpinner.getSelectedItem().toString()));
272                 printCursor(translatedSentenceTextView, data, dataNum: 0);
273             }
274         } else {
275             data = helper.getData(word
276                 , helper.getColumn(languageSpinner.getSelectedItem().toString())
277                 , helper.getColumn(translatedLanguageSpinner.getSelectedItem().toString()));
278             printCursor(translatedSentenceTextView, data, dataNum: 0);
279         }
280     }
281     String translatedSentence = translatedSentenceTextView.getText().toString();
```

Figure 4.16 : code de traduction « PageTraduction ».

4.4. Code de speaker

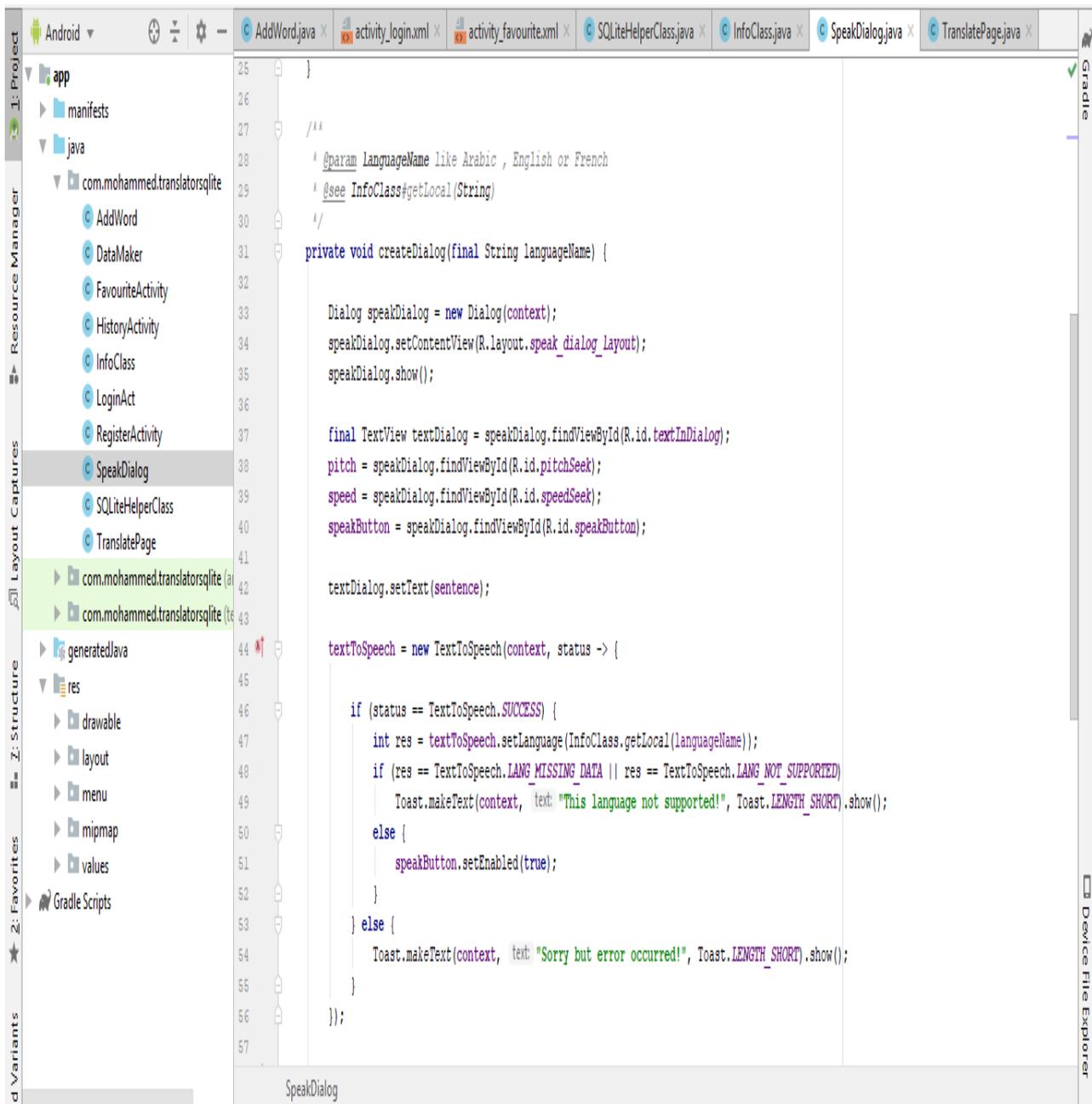


Figure 4.17: code bouton « Speaker ».

4.5. Interface « Speaker»

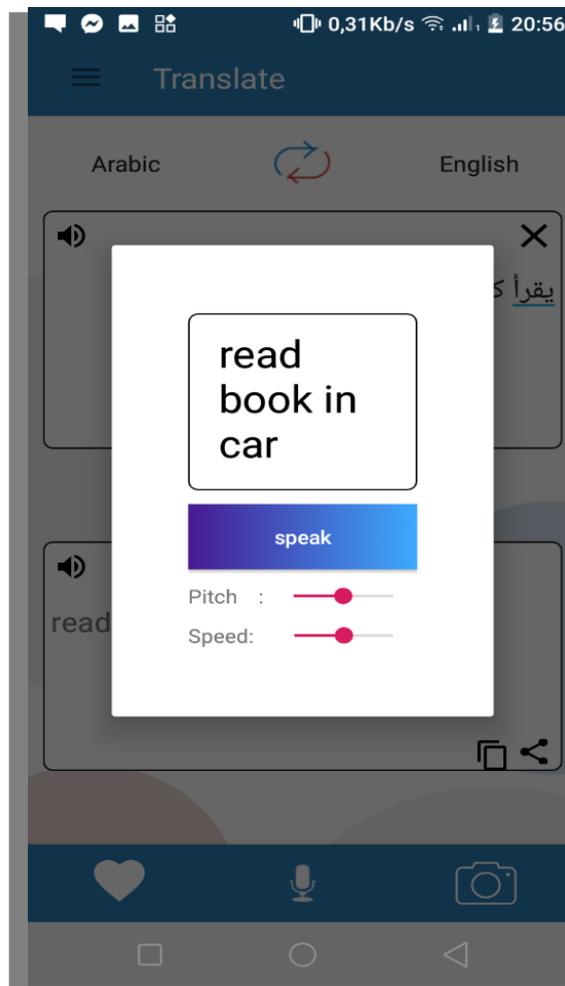


Figure 4.18: Interface « speaker ».

4.6. Interface « login »

Cette interface à l'objectif de faire crée une session à l'utilisateur.

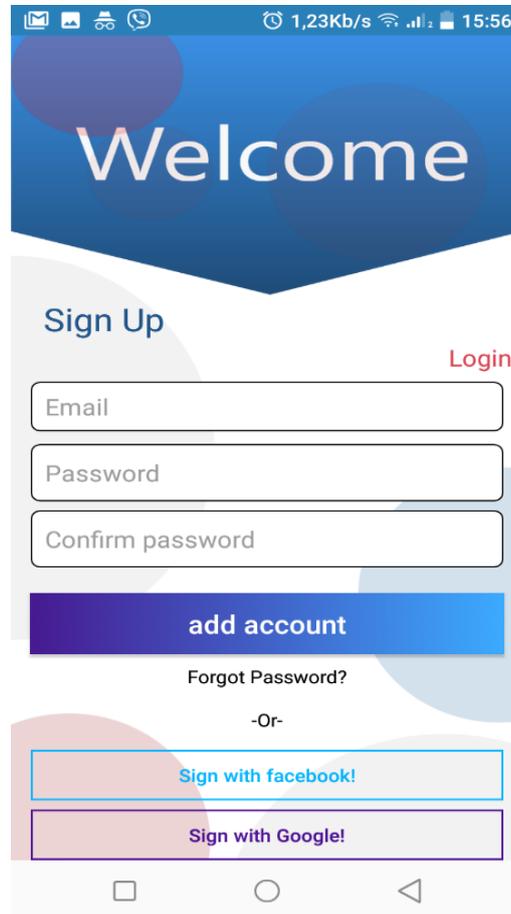


Figure 4.19: interface « login ».

4.7. Interface « Favorite » :

Cette interface contient les mots qu'on a ajoutés aux favoris.

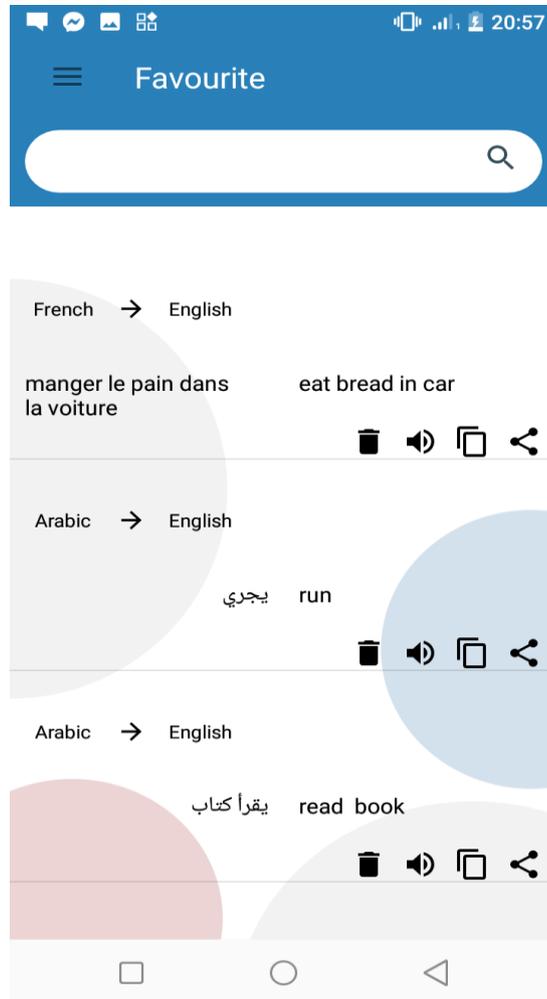


Figure 4.20: Interface « Favorite ».

4.8. Interface « History » :

Cette interface contient les mots qu'on a traduits précédemment.



Figure 3.21 : interface « History ».

4.9. Interface « Addword »

Cette interface permet d'ajouter des nouveaux mots à la BDD

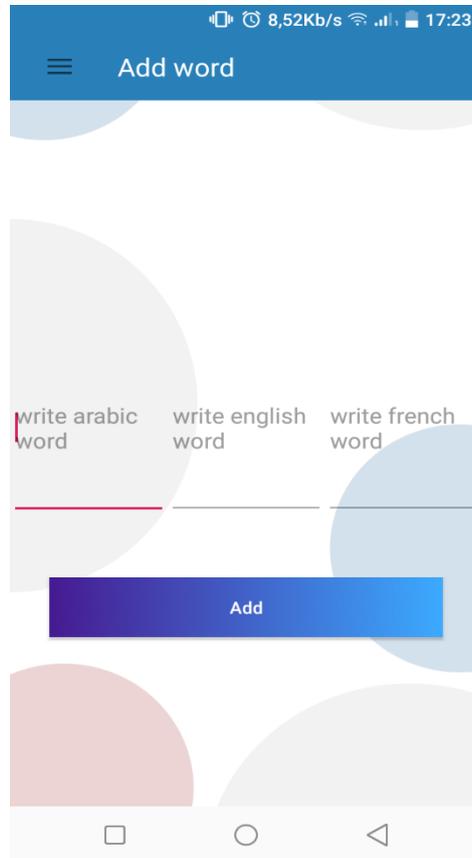


Figure 3.22 : Interface « Add word ».

4.10. Code addword

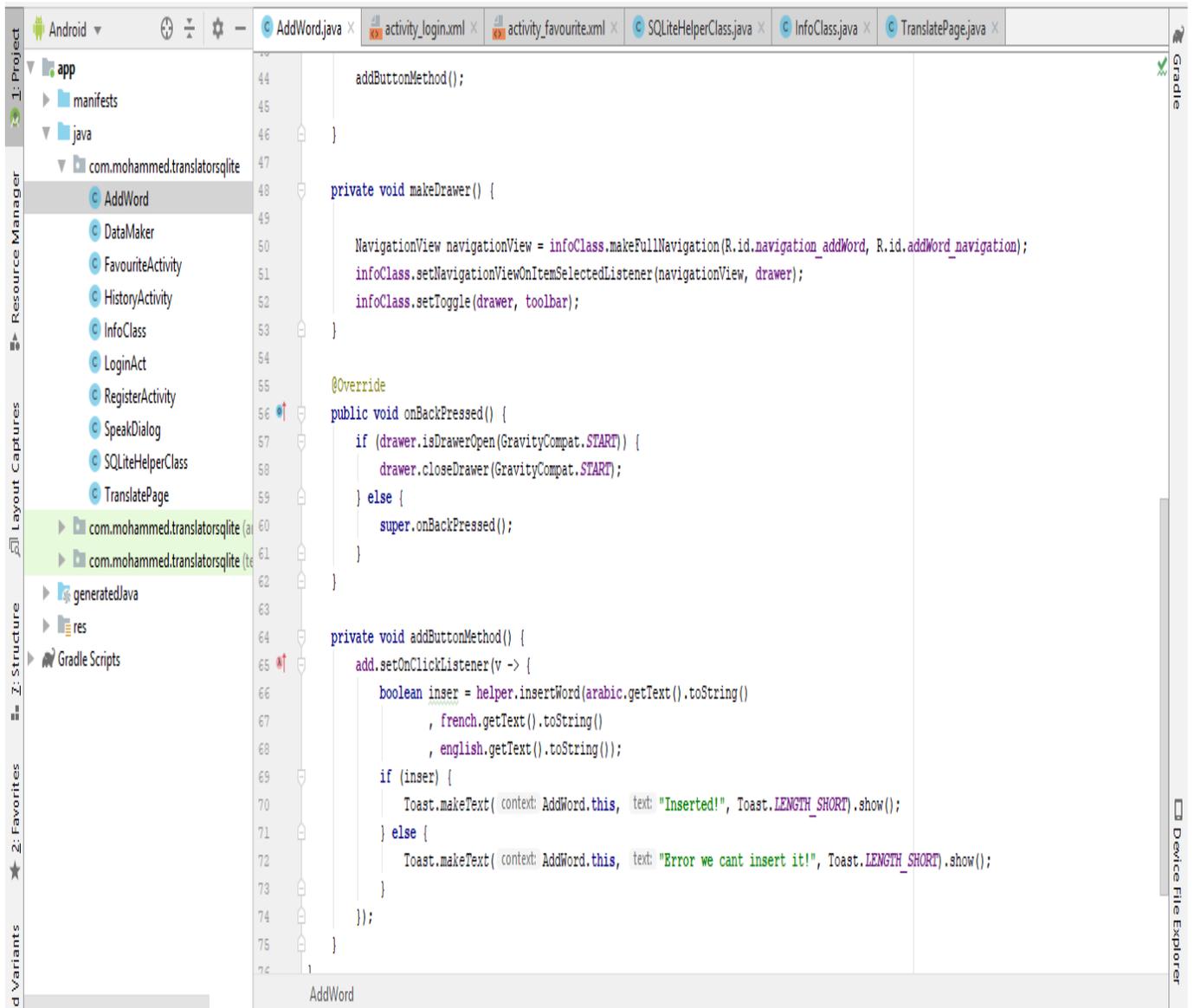


Figure 3.23 : Code addword « add word ».

4.11. Interface « Navigation Drawer » :

Cette interface permet de sélectionner les pages dans l'application.

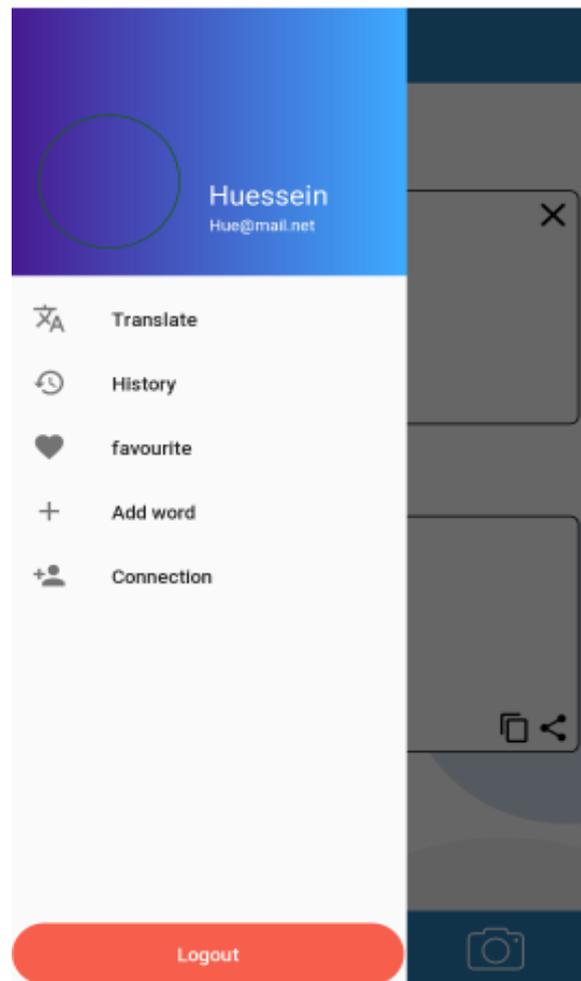


Figure 3.24 : Interface « Navigation Drawer ».

5. Conclusion

Dans ce chapitre nous avons parlé de la réalisation de l'application en voyant les outils de développement matériel et logiciel utilisés, ainsi que les interfaces graphiques de l'application.

Conclusion générale

Ce projet nous a permis d'avoir une idée plus claire et plus précise sur le développement sous Androïde, ce système d'exploitation qui a envahie le marché des terminaux mobiles.

Tout d'abord, nous avons traité d'une façon détaillé ce système en voyant son historique, ses versions, son architecture et ses outils de développements.

Puis, nous avons étudié l'utilisation des bases de données sous Android, les différentes façons de connecter une application à une BDD, et nous avons terminés ce chapitre avec la création d'une application utilisant les bases de données interne.

Et notre dernier chapitre concerne la conception et la mise en œuvre de notre application qui ai permet l'utilisateur de faire la traduction en 3 langue (arabe, français, anglais).

Nous croyons que cette application sera un complément efficace aux traductions automatiques, et une solution efficace au problème.

Avec des contraintes de temps, ne pouvons pas entrer tous les mots dans l'application nous choisirons le plus important dans cette version (mots très utilisé), mais dans l'avenir, nous travaillerons à l'améliorer, en plaçant tous les mots possibles et les rendre plus efficace.

Cette application (projet) a été très bénéfique pour nous, car il nous a permis de nos faires la traduction en trois langues sans l'obligation de se connecter à l'internet. Il nous a encore donné l'occasion de maîtriser l'Android studio.

En plus, ce projet était une bonne occasion pour réaliser un travail très concret, avec des objectifs clairs et bien définis et de se familiariser avec l'environnement du travail et la vie professionnelle.

L'objectif principal a été atteint, mais tout de même l'amélioration de notre application reste envisageable, comme l'intégration de deep learning dans l'algorithme de traduction.

BIBLIOGRAPHIE

[1] N.BENBOURAHLA,"Android 4 les fondamentaux de developpement d'application java",2012 .

[2] BAGHILI Z et BENMAZOUZ H, Conception et réalisation d'une application mobile sous Android de recherche de corps médicaux, Mémoire de fin d'études, Université Abou Bakr Belkaid– Tlemcen, 2014.

[3] BEN SAIDIA M, Application Android 'RiyadTowing', Mémoire de fin d'études, Université virtuel de Tunis, 2011.

[4] Sadik BESSOU, "Traduction Automatique à Base de Règles de l'Anglais vers l'Arabe", Université de sétif 1, 2016.

[5] Owen Rambow , " Introduction au traitement automatique de l'arabe",columbia university.

[6] KADI ALLAH F, LA TRADUCTION AUTOMATIQUE : état de l'art et les problèmes inhérents (Arabe-Français-Anglais), Mémoire de fin d'études, Université d'Oran, 2010.

WEBGRAPHIE

- [7] <https://online.visual-paradigm.com/fr/>
- [8] <http://socialcompare.com/fr/comparison/android-versions-comparison>.
- [9] <http://openclassrooms.com/courses/creez-des-applications-pour-android/1-architecture-dandroid>.
- [10] <http://www.univ-orleans.fr/lifo/Members/Jean-Francois.Lalande/enseignement/android/cours-android.pdf>.
- [11] <http://tutorielandroid.francoiscolin.fr/bdd.php>.
- [12] <http://fr.wikipedia.org/wiki/Smartphone>.
- [13] <https://www.01net.com/astuces/5-applications-de-traduction-gratuites-pour-iphone-et-android-1240297.html>.
- [14] <https://www.commentcamarche.net/forum/affich-5542562-logiciel-de-traduction-arabe-francais>.
- [15] <http://www.lefigaro.fr/secteur/high-tech/2016/12/07/32001-20161207ARTFIG00005-l-intelligence-artificielle-au-service-de-la-traduction-automatique.php>
- [16] <https://www.deepl.com/home>

Résumé

Durant ce modeste travail, nous sommes intéressés aux développements d'application mobile sous la plateforme « Android »

Notre travail a été réalisé en deux parties :

Une partie théorique dans la quel nous étudions cette plateforme en voyant son historique, ses différentes versions, son architecture et ses outils de développement ainsi que l'utilisation des bases de données sous Android.

Et une partie pratique dans laquelle nous avons réalisé une application Android, qui permet l'utilisateur de faire la traduction en trois langues (Arabe, Français et Anglais)

Mots-clés : Android, Traduction.

Abstract

During this modest work, we were interested in mobile application developments under the platform "Android".

Our work has been done in two parts:

A theoretical part in which we study this platform by seeing its history, its different versions, its architecture and its development tools and the use of Android databases.

And a practical part in which we realized an Android application, which allows the user to do the translation in three languages (Arabic, French and English)

Keywords: Android, Translation.

ملخص

خلال هذا العمل المتواضع، كنا مهتمين بتطوير تطبيقات الهاتف المحمول ضمن منصة "اندرويد"

تم عملنا على جزأين:

جزء نظري ندرس فيه هذه المنصة من خلال الاطلاع على تاريخها وإصداراتها المختلفة وهندستها وأدوات تطويرها واستخدام قواعد بيانات اندرويد.

وجزء عملي أدركنا فيه تطبيق اندرويد، والذي يسمح للمستخدم بإجراء الترجمة بثلاث لغات (العربية والفرنسية والإنجليزية)

كلمات مفتاحية: اندرويد، الترجمة.