Ministère de l’Enseignement Supérieur et de la Recherche Scientifique  
Université de Carthage  
Ecole Nationale d’Ingénieurs de Carthage



Rapport du projet

Spécialité : génie informatique

Elaboré par

Hadhraoui Marwa

**Chelly Oumaima**

**Application d’apprentissage de langue**

Enseignante

Nouira Sana

Table de matiere

Liste des tableaux

Liste des figures

**Introduction Générale**

Dans un monde de plus en plus globalisé, la maîtrise des langues est devenue un atout essentiel pour communiquer, apprendre et évoluer dans divers domaines. L'application que nous avons développée s'inscrit dans cette dynamique, en proposant une solution numérique interactive et pédagogique pour l'apprentissage des langues. Cette application, intitulée **LingoGo**, vise à offrir aux utilisateurs une expérience d'apprentissage personnalisée et engageante, grâce à des exercices variés, un suivi des progrès, et une interface conviviale.

L’objectif principal de ce projet est de fournir un outil simple et intuitif qui permet aux apprenants de s’améliorer dans différentes langues, à travers des leçons adaptées et des exercices interactifs qui renforcent l’acquisition des notions grammaticales, lexicales et phonétiques.

Le présent rapport décrit en détail le processus de conception, de développement et d’évaluation de cette application. Il est structuré en plusieurs chapitres, chacun couvrant une phase essentielle du projet :

* **Chapitre1: Etude préalable :**   
  Ce chapitre définit les objectifs fonctionnels et non fonctionnels de l’application, en mettant en avant les besoins de l’utilisateur et les contraintes techniques à respecter.
* **Chapitre 2 : Analyse et conception UML**  
  Ce chapitre détaille les diagrammes UML utilisés pour modéliser le fonctionnement de l’application, notamment les diagrammes de cas d’utilisation, de classes et de séquence. Ces diagrammes ont été conçus et corrigés pour répondre aux exigences du projet.
* **Chapitre 3 : Architecture de l’application**  
  Ce chapitre présente l’architecture technique de l’application, notamment la séparation entre la couche frontend, backend, et base de données. Une description des technologies utilisées, telles que JavaFX, MySQL et JavaFX Media, y est incluse.
* **Chapitre 4 : Guide d’utilisation de l’application**  
  Ce chapitre propose une documentation claire et détaillée pour l’utilisateur final. Il inclut des captures d’écran et des instructions pour naviguer dans les différentes fonctionnalités de l’application.

Enfin, une conclusion mettra en avant les réalisations obtenues, les éventuelles limitations rencontrées et les perspectives d’amélioration ou d’évolution pour l’avenir.

# Chapitre 1: Étude préalable

## 1.1. Présentation Générale du Projet

Dans le cadre du module de Programmation Avancée en Java, le projet intitulé **LingoGo** a pour objectif de développer une application interactive pour l’apprentissage des langues. Cette application a été conçue en respectant les consignes imposées pour un mini-projet destiné à évaluer les compétences des étudiants en programmation Java, notamment à travers l’utilisation des fonctionnalités avancées du langage, des interfaces graphiques en JavaFX et des concepts récents introduits avec Java 17.

## 1.2. Objectifs du Projet

L’objectif principal est de concevoir une application qui :

* **Met en œuvre des concepts avancés de Java** : Héritage restreint, classes scellées, records, interfaces fonctionnelles, expressions Lambda, gestion des exceptions via try-with-resources, et traitements basés sur des streams.
* **Répond aux besoins pédagogiques des utilisateurs** : Offrir un outil efficace et intuitif permettant aux apprenants d’améliorer leurs compétences linguistiques à travers des exercices interactifs, des leçons et des supports multimédias.
* **Respecte les principes du design pattern MVC** : Assurer une séparation claire entre les couches Modèle, Vue et Contrôleur.

## 1.3. Contraintes et Consignes Obligatoires

Le développement de l’application a été guidé par les consignes suivantes :

1. **Techniques** :
   * Utilisation de Java 17.
   * Implémentation d’une interface graphique en JavaFX.
   * Organisation en module avec plusieurs packages.
   * Gestion des exceptions avec des classes personnalisées et try-with-resources.
2. **Fonctionnelles** :
   * Utilisation exclusive d’interface fonctionnelle et d’expressions Lambda pour les traitements sur les collections.
   * Stockage et manipulation des données via les collections les plus adaptées (e.g., List, Map).
   * Exploitation des streams pour le filtrage, la transformation et l’agrégation des données.
3. **Ergonomiques** :
   * Une interface graphique attractive et intuitive développée en JavaFX avec une intégration possible de FXML.
   * Navigation fluide entre les différentes parties de l’application, notamment les leçons et les exercices.

## 1.4. Fonctionnalités Attendues

L’application **LingoGo** devra offrir les fonctionnalités suivantes :

1. **Gestion des leçons et exercices** :
   * Ajout, mise à jour et suppression des leçons et exercices.
   * Propositions d’exercices interactifs avec évaluation immédiate des réponses.
2. **Suivi des progrès** :
   * Suivi des performances et enregistrement des résultats de l’utilisateur.
3. **Support multimédia** :
   * Intégration de contenus audio et vidéo pour enrichir les leçons.
4. **Gestion des tests et certifications** :
   * Création de tests basés sur les leçons suivies par l’utilisateur.
   * Passage des tests par l’utilisateur pour évaluer son connaissances.
   * Délivrance d’une certification pour l’utilisateur ayant atteint un score minimal requis.
5. **Gestion des scores** :
   * Calcul et affichage des scores obtenus après chaque exercice ou test.
6. **Tableaux et statistiques** :
   * Affichage des données sous forme de TableView pour la gestion et le suivi des des leçons, et des tests.
   * Génération de statistiques détaillées (e.g., scores moyens, taux de réussite aux tests, progression globale).

## 1.5. Planification du Développement

Pour assurer un développement structuré, les étapes suivantes ont été suivies :

1. **Analyse des besoins et validation du sujet** :
   * Proposition et pré-validation du sujet auprès de l’enseignant.
   * Rédaction du cahier des charges.
2. **Conception UML** :
   * Réalisation des diagrammes de cas d’utilisation et de classes pour modéliser les fonctionnalités principales.
3. **Développement** :
   * Implémentation des classes, exceptions, et collections en respectant les contraintes.
   * Création de l’interface graphique en JavaFX et intégration des fonctionnalités interactives.
4. **Tests et validation** :
   * Vérification de la conformité aux exigences.
   * Débogage et amélioration des performances.
5. **Finalisation** :
   * Rédaction du rapport final.
   * Préparation pour la soutenance.

## 1.6. Résultats Attendus

## L’application livrée devra :

* Fournir une application fonctionnelle pour l’apprentissage des langues.
* Respecter les consignes imposées pour le projet, notamment en matière de structure et d’implémentation des concepts avancés de Java.
* Être intuitive, interactive et performante, tout en offrant une expérience utilisateur fluide.

# Chapitre 2: Analyse et conception UML

## ****2.1. Présentation du Cahier des Charges****

Cette section présente les objectifs et les fonctionnalités principales de l'application **LingoGo** en tenant compte qu'il s'agit d'une application **desktop monoposte**.

1. **Objectifs principaux** :
   * Fournir un outil interactif pour apprendre une langue via des leçons, des exercices, des tests, et des statistiques de progression.
   * Permettre à l'utilisateur de suivre ses scores et d'obtenir une certification après avoir passé des tests.
2. **Fonctionnalités clés** :
   * Gestion des leçons (création, modification, suppression).
   * Propositions d'exercices interactifs avec correction instantanée.
   * Passage de tests avec attribution de scores et de certificats.
   * Suivi des statistiques et visualisation des progrès.
   * Intégration de contenus multimédias (audio, vidéo).
3. **Contraintes techniques** :
   * Application développée en **Java 17** avec **JavaFX** pour l'interface graphique.
   * Utilisation de **streams**, **lambda expressions**, et **collections adaptées** pour manipuler les données efficacement.
   * Architecture organisée en modules et packages.

## ****2.2. Diagrammes UML****

1. **Diagramme de cas d'utilisation**
   * Acteur principal : **Utilisateur unique** (celui qui utilise l'application).

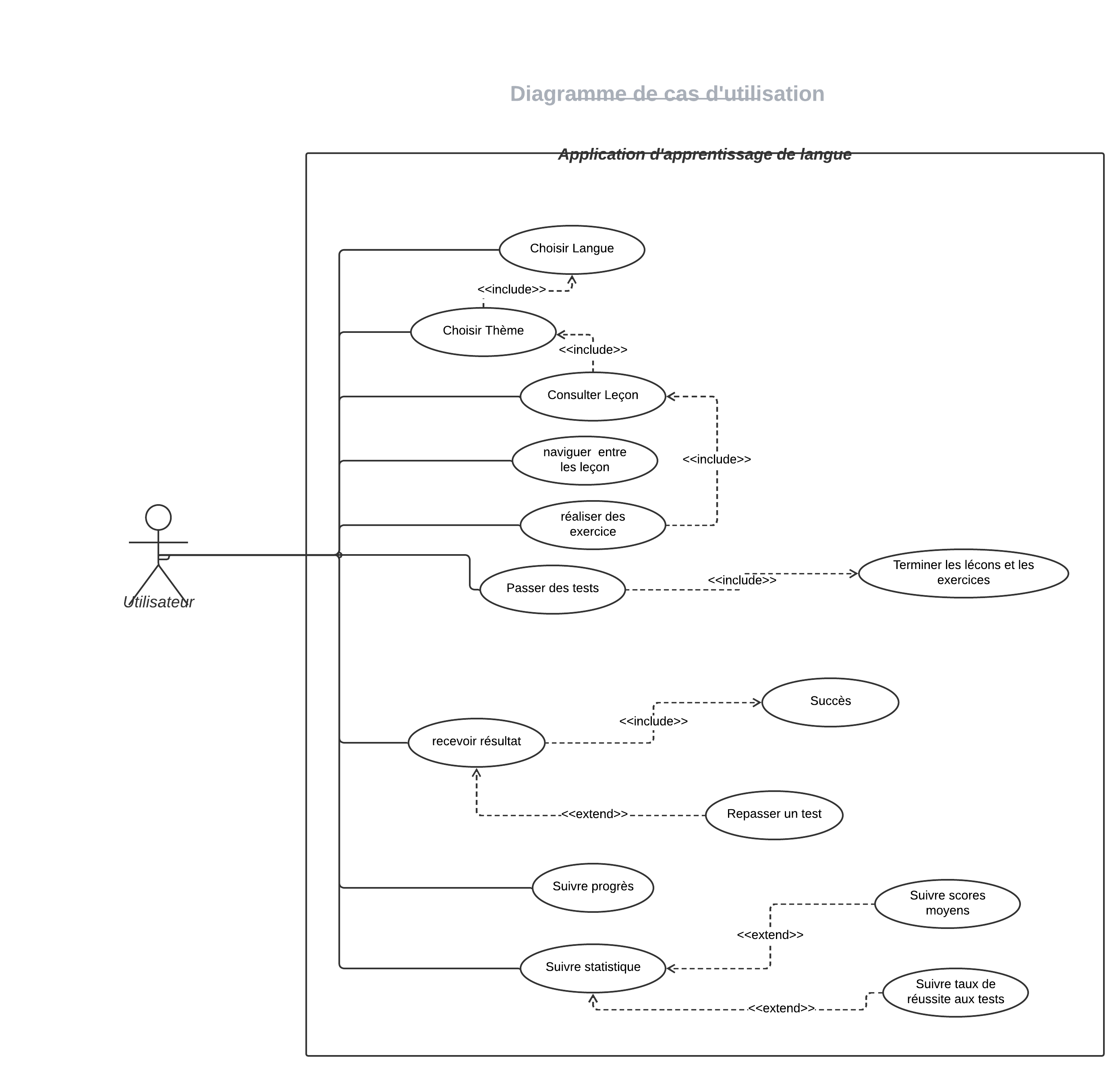


Figure 1 : Diagramme de cas d'utilisation

1. **Diagramme de classe**