* 
* **Rapport de projet**
* **Java**

**Réalisé par** : **Elabed Sabrine**

**Jraidi Houssem**

**Élèves ingénieur : 2 ème année Génie informatique groupe C**

**Sujet : Jeu d’Echecs**

**A.U: 2018/2019**

Table des matières

[CHAPITRE I : Contexte du travail et cahier de charge 4](#_Toc8430)

[Présentation du projet  5](#_Toc3556)

[CHAPITRE II : Etude conceptuelle 9](#_Toc32628)

[I. Méthodologie de conception adoptée 10](#_Toc27147)

[1. Présentation d’UML : 10](#_Toc16054)

[2. Les avantages d’UML 11](#_Toc18109)

[3. Logiciel utilisé 11](#_Toc7487)

[III. Le diagramme élaboré 11](#_Toc3045)

[Diagramme des classes 11](#_Toc25302)

[Chapitre III : Réalisation 13](#_Toc26024)

[I. Environnement du travail 14](#_Toc24781)

[1. Environnement matériel 14](#_Toc22972)

[2. Environnement de développement 14](#_Toc4168)

[3. Environnement logiciel 16](#_Toc10061)

[II. Les interfaces de l’application 17](#_Toc5685)

[1. Interface d’Accueil « Sign Up » 17](#_Toc16839)

[2. Interface « Log In » 18](#_Toc22946)

[3. Interface « Jeu » 18](#_Toc30732)

Liste des figures

[Figure 3 : Logo UML 11](#_Toc28862)

[Figure 4 : StarUML 11](#_Toc491)

[Figure 8 : Diagramme des classes 12](#_Toc2893)

[Figure 9 : Logo du langage JAVA 14](#_Toc17712)

[Figure 10 : Logo de JavaFX 15](#_Toc21895)

[Figure 11 : Logo d'Oracle 15](#_Toc21865)

[Figure 12 : Logo de NetBeans 16](#_Toc9794)

[Figure 13 : Logo de SQL Developer 16](#_Toc20915)

[Figure 14 : Interface d'Accueil 17](#_Toc20784)

[Figure 15 : Interface Log In 18](#_Toc17379)

[Figure 20 : Interfaçe Jeu 18](#_Toc3167)

**Introduction générale**

C

’est avec enthousiasme que j’ai choisir de developper un jeu d’echec au sein de projet java

Il s'agissait au cours du projet d'étudier et de développer un jeu d’echecs sous Java. Avec la passion du jeu dans notre jours, les développements du jeu sont de plus en plus demandés sur le marché. Avoir un PC est devenu incontournable pour les jeunes de nos jours. D'où nait l'idée de développer une application utile, permettant aux joueurs d’amuser par jouer ce jeu.

Ce rapport présente l’ensemble des étapes suivies pour développer le jeu. Il contient trois chapitres organisés comme suit :

* Le premier chapitre intitulé «Contexte du travail et cahier de charge » est consacré à la présentation du contexte du travail.
* Le troisième chapitre intitulé « Etude conceptuelle du projet » sera consacré pour l’étude conceptuelle des différentes fonctionnalités de notre application. Cette étude décrira d’une manière précise toutes les étapes de la conception en se basant sur la méthodologie UML.
* Le dernier chapitre intitulé « Réalisation » présente l’environnement de travail, ainsi que les outils logiciels que nous avons utilisés pour la réalisation de notre projet. Il illustre aussi le travail réalisé avec un ensemble d’interfaces graphiques conçues pour l’application.

Pour conclure, nous terminons le rapport par une conclusion générale.

CHAPITRE I : Contexte du travail et cahier de charge

## **Présentation du projet**

Je commence par introduire le principe du jeu d’echecs, definir chaque piece, et expliquer les lois du jeu.

**1.Principe du jeu d’echecs**

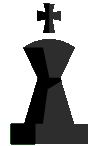
Le jeu d’échecs se joue entre deux adversaires qui déplacent alternativement des pièces sur un plateau carré appelé "échiquier". Le joueur ayant les pièces blanches commence la partie. On dit qu’un joueur "a le trait" lorsque le coup de son adversaire a été "joué".

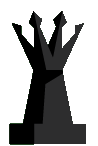
L’objectif de chaque joueur est de placer le roi adverse "sous une attaque" de telle manière que l’adversaire n’ait aucun coup légal. On dit que le joueur qui atteint ce but a "maté" le roi adverse et gagné la partie.

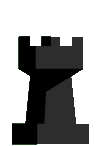
L’échiquier se compose d’une grille 8x8 de 64 cases identiques alternativement claires (les cases "blanches") et foncées (les cases "noires"). L’échiquier est placé entre les joueurs de telle manière que la case d’angle à la droite de chaque joueur soit blanche. Au début d’une partie, un joueur dispose de 16 pièces claires (les pièces "blanches") et l’autre de 16 pièces foncées (les pièces "noires").

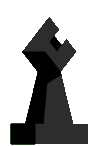
1. **Les types de pièces**

Les pièces sont les suivantes :

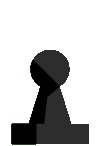
Un roi noir ayant pour symbole usuel 

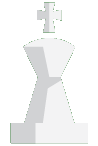
Une dame noire ayant pour symbole usuel 

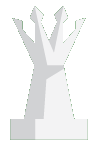
Deux tours noires ayant pour symbole usuel 

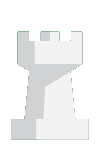
Deux fous noirs ayant pour symbole usuel 

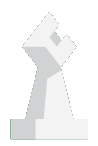
Deux cavaliers noirs ayant pour symbole usuel 

Huit pions noirs ayant pour symbole usuel 

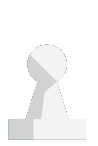
Un roi blanc ayant pour symbole usuel 

Une dame blanche ayant pour symbole usuel 

Deux tours blanches ayant pour symbole usuel 

Deux fous blancs ayant pour symbole usuel 

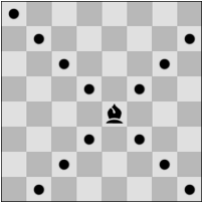
Deux cavaliers blancs ayant pour symbole usuel 

Huit pions blancs ayant pour symbole usuel 

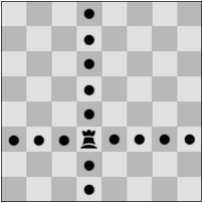
**3.Les lois du jeu**

Il n’est pas permis de déplacer une pièce sur une case occupée par une pièce de même couleur. Si une pièce se déplace sur une case occupée par une pièce adverse, cette dernière est prise et retirée de l’échiquier comme partie intégrante du coup. On dit qu’une pièce attaque une pièce adverse si elle peut éventuellement effectuer une prise sur cette case

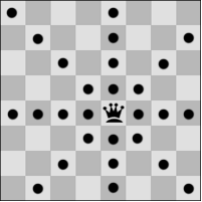
Le fou se déplace sur toute case d’une diagonale sur laquelle il se trouve.



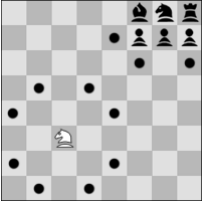
La tour se déplace sur toute case de la colonne ou de la rangée sur laquelle elle se trouve.



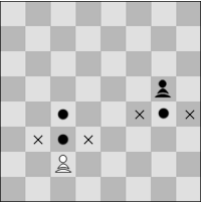
La dame se déplace sur toute case de la colonne, de la rangée ou d’une diagonale sur laquelle elle se trouve.



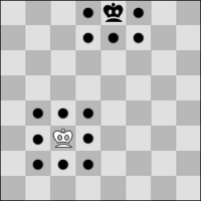
Le cavalier se déplace sur l’une des cases les plus proches de celle sur laquelle il se trouve, mais pas sur la même colonne, rangée ou diagonale.



Le pion se déplace sur la case inoccupée immédiatement devant lui sur la même colonne, ou à son premier coup, il peut avancer de deux cases sur la même colonne à condition qu’elles soient toutes deux inoccupées, ou il se déplace sur une case occupée par une pièce adverse, située devant lui en diagonale sur une colonne adjacente



Le roi se deplace sur l’une des cases adjacentes qui n’est pas attaquée par une ou plusieurs pièces adverses



**Conclusion**

Ce chapitre introductif nous a permis de détailler le cadre général du jeu. Et ce en présentant le principe du jeu, l’ensemble des pièces, et les principaux lois du jeu. Dans ce qui suit je vais entamer la première phase de la conception de mon projet «Etude conceptuelle» pour identifier les différentes fonctionnalités de l’application.

CHAPITRE II : Etude conceptuelle

**Introduction**

L

a phase d’analyse et de spécification des besoins présente une étape primordiale dans le cycle de développement d’un projet. Le but de cette phase est de comprendre le contexte du système. Il s’agit d’éclaircir au mieux les besoins fonctionnels et non fonctionnels, apparaitre les acteurs et identifier les cas d’utilisation.

## 

## Méthodologie de conception adoptée

Avant de programmer l’application et se lancer dans l’écriture du code : il faut tout d’abord organiser les idées, les documenter, puis organiser la réalisation en définissant les modules et les étapes de la réalisation. Cette démarche antérieure à l’écriture que l’on appelle modélisation ; son produit et un module.

La modélisation consiste à créer une représentation virtuelle d’une réalité de telle façon à faire ressortir les points auxquels on s’intéresse. Dans le cadre de mon projet j’ai utilisé la méthodologie UML pour la modélisation des différents diagrammes.

### Présentation d’UML :

En regardant les objectifs fixés pour la réalisation du projet, nous remarquons que nous sommes face à une application modulaire et qui devra rester ouverte pour les améliorations futures. De ce fait, il est très important d’utiliser un langage universel pour la modélisation afin de clarifier la conception et de faciliter les échanges.

Notre choix est porté sur le langage UML puisqu’il convient pour toutes les méthodes objet et se prête bien à la représentation de l’architecture du système.

* **UML (Unified Modeling Language)** est un langage de modélisation unifié permet de modéliser une application logicielle d’une façon standard dans le cadre de conception orienté objet. UML permet de couvrir le cycle de vie d’un logiciel depuis la spécification des besoins jusqu’au codage en offrant plusieurs moyens de description et de modélisation des acteurs et de utilisation système, du comportement des objets, du flot de contrôle internes aux opérations, des composants d’implémentation et leurs relations, de la structure matérielle et de la distribution des objets et des composants indépendamment des techniques d’implémentation et peut être mis à jour selon les besoins.

****

Figure 3: Logo UML

### Les avantages d’UML

* Universel
* Adopté par les grandes entreprises
* Notation unifié
* Facile à comprendre
* Adopté par plusieurs processus de développement
* Limite les risques d’erreur
* N’est pas limité au domaine informatique

### Logiciel utilisé

staruml logo**StarUML** est un logiciel de modélisation [UML](https://fr.wikipedia.org/wiki/UML_(informatique)" \o "UML (informatique)), cédé comme [open source](https://fr.wikipedia.org/wiki/Open_source" \o "Open source) par son éditeur, à la fin de son exploitation commerciale, sous une licence modifiée de GNU GPL. L'objectif de la reprise de ce projet était de se substituer à des solutions commerciales comme *[IBM Rational Rose](https://fr.wikipedia.org/wiki/Rational_Rose" \o "Rational Rose)* ou *Borland Together*.



Figure 4: StarUML

## Le diagramme élaboré

### Diagramme des classes

Les diagrammes de classes sont destinés à décrire les propriétés structurelles des objets du monde réel, les concepts spécifiques du domaine considéré ou encore les notions abstraites que le système doit appréhender. Ce sont les diagrammes les plus fréquemment utilisés en modulation orientée objets. En phase de conception du logiciel, ils sont exploités pour décrire l’architecture statique du système et les interdépendances entre ses constituants

Ci-dessous, je présente le diagramme des classes.

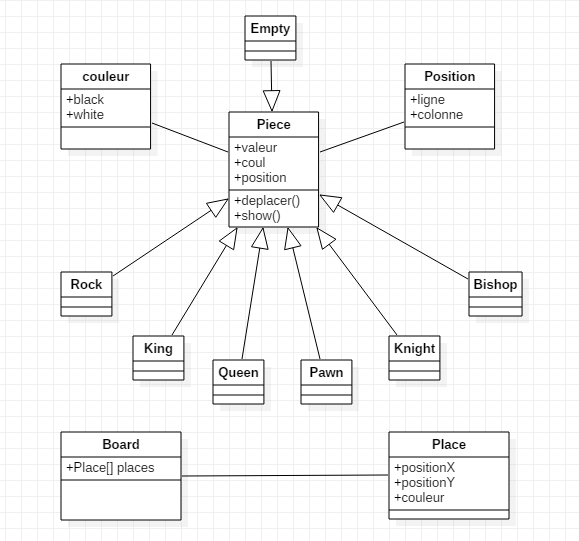


Figure 8: Diagramme des classes

**Conclusion**

Dans cette phase, j’ai effectué la conception de l’application. J’ai effectué le diagramme de classe. Tout en spécifiant la méthodologie de conception adoptée pour ce diagramme.

Chapitre III : Réalisation

**Introduction**

D

ans les chapitres précédents, j’ai décrit les différentes notions nécessaires à la bonne compréhension du sujet, par le biais d’une spécification et analyse des besoins, ainsi qu’une conception du système. Dans cette partie je présente en premier lieu l'environnement de travail dans lesquels mon application a été développée en indiquant les technologies utilisées, ensuite je décris les langages de programmation. Je finis ce chapitre par quelques captures d'écran traduisant le déroulement de l’application.

## Environnement du travail

### Environnement matériel

Au cours de la phase de réalisation, j’ai utilisé l’ordinateur ayant les caractéristiques suivantes :

|  |  |
| --- | --- |
| **Modèle** | Packard bell |
| **Système d’exploitation** | Windows 10 |
| **Processeur** | Intel® Core™ i3-2310M |
| **Mémoire installée (RAM)** | 4 Go |

Tableau 2 : Environnement matériel

### Environnement de développement

* **Java**

Android est un système d’exploitation conçu pour téléphone mobile développé par Google, qui a mis à disposition un kit de développement logiciel (SDK) basé sur le langage Java. Nous rappelons que Java, étant un langage de programmation orienté objet utilisable sur divers systèmes d’exploitation, est un langage assez robuste, portable et à hautes performances.



Figure 9: Logo du langage JAVA

* **JavaFX**

**JavaFX** est une technologie créée par [Sun Microsystems](https://fr.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) qui appartient désormais à [Oracle](https://fr.wikipedia.org/wiki/Oracle_(entreprise)" \o "Oracle (entreprise)), à la suite du rachat de [Sun Microsystems](https://fr.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) par [Oracle](https://fr.wikipedia.org/wiki/Oracle_(entreprise)" \o "Oracle (entreprise)) le 20 avril 2009.

Avec l'apparition de Java 8 en mars 2014, JavaFX devient la bibliothèque de création d'interface graphique officielle du langage Java, pour toutes les sortes d'application (applications mobiles, applications sur poste de travail, applications Web), le développement de son prédécesseur [Swing](https://fr.wikipedia.org/wiki/Swing_(Java)" \o "Swing (Java)) étant abandonné (sauf pour les corrections de bogues).

JavaFX est désormais une pure API Java (le langage de script spécifique qui a été un temps associé à JavaFX est maintenant abandonné).

JavaFX contient des outils très divers, notamment pour les médias audio et vidéo, le graphisme 2D et 3D, la programmation Web, la programmation multi-fils etc.



Figure 10 : Logo de JavaFX

* **Oracle Database**

***Oracle Database*** est un [système de gestion de base de données](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es" \o "Système de gestion de base de données) [relationnelle](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9es_relationnelle" \o "Base de données relationnelle) (SGBDR) qui depuis l'introduction du support du [modèle objet](https://fr.wikipedia.org/wiki/Programmation_orient%C3%A9e_objet" \o "Programmation orientée objet) dans sa version 8 peut être aussi qualifié de [système de gestion de base de données relationnel-objet](https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_base_de_donn%C3%A9es_relationnel-objet" \o "Système de gestion de base de données relationnel-objet) (SGBDRO). Fourni par [Oracle Corporation](https://fr.wikipedia.org/wiki/Oracle_Corporation" \o "Oracle Corporation), il a été développé par [Larry Ellison](https://fr.wikipedia.org/wiki/Larry_Ellison" \o "Larry Ellison), accompagné entre autres, de [Bob Miner](https://fr.wikipedia.org/wiki/Bob_Miner" \o "Bob Miner) et [Ed Oates](https://fr.wikipedia.org/wiki/Ed_Oates" \o "Ed Oates).



Figure 11 : Logo d'Oracle

### Environnement logiciel

* **NetBeans**

**NetBeans** est un [environnement de développement intégré](https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9" \o "Environnement de développement intégré) (EDI), placé en *[open source](https://fr.wikipedia.org/wiki/Open_source" \o "Open source)* par [Sun](https://fr.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) en [juin 2000](https://fr.wikipedia.org/wiki/Juin_2000" \o "Juin 2000) sous licence CDDL ([Common Development and Distribution License](https://fr.wikipedia.org/wiki/Common_Development_and_Distribution_License" \o "Common Development and Distribution License)) et [GPLv2](https://fr.wikipedia.org/wiki/Licence_publique_g%C3%A9n%C3%A9rale_GNU" \o "Licence publique générale GNU). En plus de Java, NetBeans permet la prise en charge native de divers langages tels le [C](https://fr.wikipedia.org/wiki/C_(langage)" \o "C (langage)), le [C++](https://fr.wikipedia.org/wiki/C_plus_plus" \o "C plus plus), [le JavaScript](https://fr.wikipedia.org/wiki/JavaScript" \o "JavaScript), [le XML](https://fr.wikipedia.org/wiki/Extensible_Markup_Language" \o "Extensible Markup Language), [le Groovy](https://fr.wikipedia.org/wiki/Groovy_(langage)" \o "Groovy (langage)), [le PHP](https://fr.wikipedia.org/wiki/PHP" \o "PHP) et [le HTML](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language" \o "Hypertext Markup Language), ou d'autres (dont [Python](https://fr.wikipedia.org/wiki/Python_(langage)" \o "Python (langage)) et [Ruby](https://fr.wikipedia.org/wiki/Ruby" \o "Ruby)) par l'ajout de *greffons*. Il offre toutes les facilités d'un IDE moderne ([éditeur avec coloration syntaxique](https://fr.wikipedia.org/wiki/Coloration_syntaxique" \o "Coloration syntaxique), projets [multi-langage](https://fr.wikipedia.org/wiki/Multi-langage" \o "Multi-langage), [refactoring](https://fr.wikipedia.org/wiki/Refactoring" \o "Refactoring), éditeur graphique d'interfaces et de pages Web).

Compilé en Java, NetBeans est disponible sous [Windows](https://fr.wikipedia.org/wiki/Microsoft_Windows" \o "Microsoft Windows), [Linux](https://fr.wikipedia.org/wiki/Linux" \o "Linux), [Solaris](https://fr.wikipedia.org/wiki/Solaris_(informatique)" \o "Solaris (informatique)) (sur [x86](https://fr.wikipedia.org/wiki/X86" \o "X86) et [SPARC](https://fr.wikipedia.org/wiki/Architecture_SPARC" \o "Architecture SPARC)), [Mac OS X](https://fr.wikipedia.org/wiki/Mac_OS_X" \o "Mac OS X) ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit [JDK](https://fr.wikipedia.org/wiki/JDK" \o "JDK) est requis pour les développements en Java.



Figure 12: Logo de NetBeans

* **Oracle SQL Developer**

**Oracle SQL Developer** est un [environnement de développement intégré](https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9" \o "Environnement de développement intégré) (EDI) multi-plateforme, fourni gratuitement par [Oracle Corporation](https://fr.wikipedia.org/wiki/Oracle_Corporation" \o "Oracle Corporation) et utilisant la technologie Java ([Java Development Kit](https://fr.wikipedia.org/wiki/Java_Development_Kit" \o "Java Development Kit)). C'est un outil graphique permettant d'interroger des [bases de données](https://fr.wikipedia.org/wiki/Base_de_donn%C3%A9e" \o "Base de donnée) [Oracle](https://fr.wikipedia.org/wiki/Oracle_Database" \o "Oracle Database) à l'aide du langage [SQL](https://fr.wikipedia.org/wiki/Structured_Query_Language" \o "Structured Query Language).



Figure 13 : Logo de SQL Developer

## Les interfaces de l’application

### Interface d’Accueil « Sign Up »

Cette interface permet à l’utilisateur de créer un compte ou bien passer à l’interfaçe Log In s’il a déjà un compte

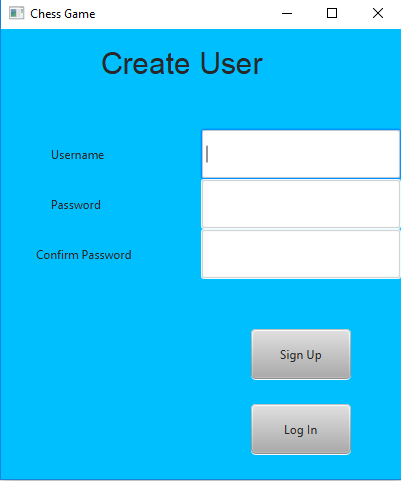


Figure 14 : Interface d'Accueil

Si la bouton Sign Up est cliquée, les données de l’utilisateur seront enregistrées dans la base de données et on passe à l’interfaçe Log In.. Si la bouton Log In est cliquée, on passe directement à l’interfaçe Log In .

### Interface « Log In »

Les deux joueurs doient entrer leurs noms et mots de passe puis cliquer sur le bouton jouer pour commencer à jouer, la bouton return permet de retourner à la page précédente

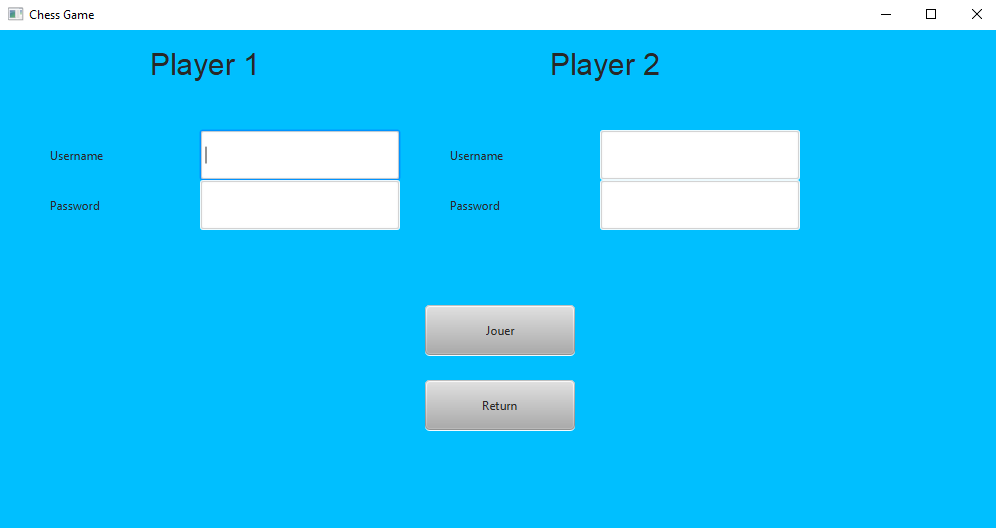


Figure 15 : Interface Log In

### Interface « Jeu »

Quand le jeu commence la fenetre va etre comme suit:



Figure 20 : Interfaçe Jeu

**Conclusion**

Dans ce chapitre de réalisation, j’ai présenté l‘environnement matérielles et logicielles de mon projet, ainsi que les technologies de développement utilisées. J’ai par la suite, présenté les principales interfaces de mon application.

Conclusion générale

Mon projet a été pour moi une expérience enrichissante qui m’a a permis de :

* Découvrir de nouveaux outils et environnements
* Mettre en pratique mes connaissances théoriques

J’espère qu’avec la réalisation de mon Jeu d’Echecs, je pourrais atteindre mon objectif pour améliorer le service rendu aux clients. En passant par la conception, nous avons procédé à une spécification des besoins afin de faciliter la conception générale, détaillée et technique. Cette réalisation m’a permis d’acquérir les compétences suivantes :

* permet de créer un base de données par la méthode data base adapter (Oracle) de façon approfondie.
* Cette occasion j’ai permis d’expérimenter le travail binôme, ce qui m’a initié au milieu de travail et Comment dépasser les obstacles, et comment travailler en équipe.

Ce projet a été une bonne expérience pour expliquer les différentes connaissances vues dans la formation reçu à EniCarthage tout en tenant en compte les différentes étapes de réalisation d’un projet et en intégrant mes propres connaissances.

Du point de vue technique, ce projet m’a permis de m’adapter avec l’environnement du développement informatique, de même il m’a permis de maîtriser la méthode de développement en Java.