**République Tunisienne**

**Ministère de l’Enseignement Supérieur**

**Et de la Recherche Scientifique**

**Université de Carthage**

**Ecole Nationale d’Ingénieurs de Carthage**

****

**Projet réalisé par :**

**Feres NEFZI**

**Malek BARNAT**

**Encadré par :**

**Mme. BESSOUDA**

**Mr. BEN SLIMEN**

Groupe A

Année Universitaire : 2020/2021

**Sommaire**

# **Introduction…………………………………...………………………………...……………………**

## Histoire de la banque en ligne.…………………………………………………………..**3**

## Présentation du problème………………….………….…………….…..…………….…**3**

## Résultats attendus.…………………………….………………………….………………….**4**

# **Présentation des logiciels utilisés………………………….…………..…………………4**

# **Etude des besoins fonctionnels et non fonctionnels……………...………………6**

# **Présentation de la partie analyse & conception.......…………….…………….……..**

# Présentation de la démarche suivie…………………………………………..…….……….. **11**

# Un petit aperçu des diagrammes………………………………………………….………………..

# Diagramme des cas d’utilisations principal……………………………...…………**11**

1. Diagramme des classes…………….……………………………………………….............**12**

# **Présentation du modèle EA de la base de données………………………..........13**

# **Réalisation et tests de l’application………………………………….………………....13**

# **Conclusion……………………………………………………………………………...………….. 17**

1. **Introduction**

**A. Histoire de la banque en ligne**

Dans les années 80 les banques britanniques sont les premières à diffuser des informations en ligne via le réseaux téléphonique ou Prestel : Le premier système de Post-Banking : C’est le début de l’ère du Banking Online

Partout ailleurs, il faudra attendre tardivement l’arrivée d’internet pour convaincre la clientèle. Les clients sont assez réticents aux changements et **la bulle internet des années 2000 ce qui ne permet même pas aux établissements bancaires de surfer sur la vague du numérique**. Au contraire le modèle de banque traditionnel en ressort renforcé et la prise de risque est limitée.

La relation bancaire évolue, le lien entre le client et le conseiller est devenu de nos jours moins personnel, l’agence est devenue aussi moins fréquentée. Il faut avouer que la clientèle a progressivement été éduquée pour être autonome avec les nouvelles technologies : distributeurs automatiques, cartes bancaires, téléphones, internet & évidemment à quoi on va s’intéresser dans ce sujet : **les applications**.

**B. Position du problème**

Le contact humain n’est plus nécessaire et **le libre service bancaire** est désormais le mode de relation bancaire par défaut dans de nombreux réseaux. Banque en ligne, e-Banking, **banque mobile,** home banking… quelque soit le terme utilisé [la banque par internet & application devient la relation privilégiée](https://www.culturebanque.com/banques/banques-en-ligne/) par la clientèle et par la banque !

**L’intérêt de la banque en ligne renait** par l’automatisation des tâches à faibles valeurs ajoutées dans la banque. Les virements, commandes de chéquiers, les relevés de comptes sont maintenant exécutés par l’informatique. La clientèle peut réaliser elle-même les opérations simples par téléphone, au distributeur **ou encore en ligne à travers les applications …**

La clientèle s’abonne par millions aux outils en ligne des banques traditionnelles et les établissements accompagnent cette tendance en proposant systématiquement la dématérialisation des relevés de compte ou des ordres de bourse. La clientèle apprécie **la facilité et la rapidité de ces nouveaux services par applications mobiles & desktop via internet.**

**C. Résultats attendus**

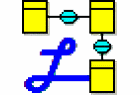
L'étude de notre application desktop **Digi-Bank** a fait l'objet de réduire ces tâches à faibles valeurs ajoutées dans la banque et de les instaurer et automatiser grâce à des manipulations & fonctions basiques et complètes au moyen d'un interfaçage simple, facile à manipuler et captivant, tout en instaurant un système de gestion d'utilisateurs qui gère et protège leurs données.

1. **Présentation des logiciels utilisés**

Le projet est répartit essentiellement selon trois grandes étapes :

1. Analyse & conception
2. Implémentation de la base de données
3. Développement et codage

Pour la première partie, c’est-à-dire pour la création des modèles conceptuels, nous avons utilisé :

* Star UML : est un logiciel de modélisation UML (Unified Modeling Language), qui a été cédé comme open source, qui remplace plusieurs logiciels coûteux. Star UML gère la plupart des diagrammes spécifiés dans la norme UML 2.0, il est donc adéquat pour la modélisation des diagrammes.
* Looping : est un logiciel de modélisation conceptuelle de données qui permet d'organiser et de structurer vos données en diagrammes et représentations graphiques ordonnées. Vu sa flexibilité et sa multifonction, Looping  permet notamment de modéliser des entités et des associations, de générer des diagrammes de classes UML, de créer des associations réflexives avec définitions des rôles, de transformer ces associations en entités avec identifiants relatifs, d'exporter des scripts SQL pour module VB-Access, d'afficher en temps réel les modules logiques de données textuels, ou encore d'exécuter des requêtes SQL de création de tables de la base de données instantanément. C’est pour ces raisons que nous avons tiré profit de ce logiciel au cours de la phase conception.

Quant à la deuxième partie, l’implémentation a exigé :

* Oracle Database : est un système de gestion de base de données relationnelle qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet, par le biais de ce système, nous avons pu implémenter et traiter une quantité énorme de données.
* SQL Developer : est un environnement de développement intégré multiplateforme, fourni gratuitement par Oracle Corporation et utilisant la technologie Java. C'est un outil graphique permettant d'interroger des bases de données Oracle à l'aide du langage SQL.

Pour la dernière phase, les logiciels utilisés sont :

* SceneBuilder : un outil interactif de conception d'interfaces graphiques pour JavaFX. Créé par Oracle et développé par Gluon, il permet de construire rapidement des interfaces utilisateurs et fournit comme résultat un fichier FXML qui peut ensuite être combiné avec un projet Java en liant l'interface utilisateur à la logique de l'application.
* NetBeans : est un [environnement de développement intégré](https://fr.wikipedia.org/wiki/Environnement_de_d%C3%A9veloppement_int%C3%A9gr%C3%A9) (EDI), placé en [open source](https://fr.wikipedia.org/wiki/Open_source) qui nous a permis la prise en charge native du langage Java et d’autres langages utilisés dans ce projet tels le CSS, SQL et le format de données textuelles FXML 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), projets [multi-langage](https://fr.wikipedia.org/wiki/Multi-langage), [refactoring](https://fr.wikipedia.org/wiki/Refactoring), éditeur graphique d'interfaces et de pages Web).

**III. Etude des besoins fonctionnels & non-fonctionnels**

Le slogan de notre application Digi-Bank est *« Transactions Otherwise »* qui veut dire *« Les transactions autrement »* : L’objectif de ce projet est donc de faciliter et satisfaire les besoins du client. C’est devant ce fait que nous présentons cette étude des fonctionnalités et des besoins afin de bien préciser les nécessités de l’application et ses fonctions optionnelles. En effet, l’application sert en premier lieu à effectuer et faciliter les opérations bancaires après avoir créé un compte Digi-Bank.

Ceci exige donc les fonctions :

|  |  |
| --- | --- |
| Fonction : Etablissement de la connexion | |
| Objectif | Accéder à son compte Digi-Bank | |
| Description | Accès à la base de données et récupération du compte suite à l’introduction d’un nom d’utilisateur & d’un mot de passe. L’utilisateur pourra demander un nouveau mot de passe s’il l’avait oublié. | |
| Contraintes / règles de gestion | -Nom d’utilisateur existant dans la base de données  -Mot de passe valide | |
| Niveau de priorité | Priorité moyenne | |

|  |  |
| --- | --- |
| Fonction : Création d’un compte | |
| Objectif | Créer un compte Digi-Bank |
| Description | Créer un nouveau compte Digi-Bank en fournissant les informations personnelles nécessaires aux différents champs |
| Contraintes / règles de gestion | -Format correct aux différents champs  -Nom d’utilisateur disponible  -Contrainte au format du mot de passe |
| Niveau de priorité | Priorité moyenne |

|  |  |
| --- | --- |
| Fonction : Gestion des données personnelles | |
| Objectif | Modifier & mettre à jour les données relatives à l’utilisateur |
| Description | Suite à l’introduction de nouvelles informations et confirmation du mot de passe, accès à la base de données et mise à jour des données personnelles modifiées |
| Contraintes / règles de gestion | -Connexion établie  -Format valide aux différents champs  -Confirmation par le mot de passe correct |
| Niveau de priorité | Priorité haute |

|  |  |
| --- | --- |
| Fonction : Créditer/Débiter son compte | |
| Objectif | Créditer/débiter son compte |
| Description | Mettre à jour le solde du compte après avoir fourni le montant à créditer/débiter et cliquer sur le bouton Deposit/Withdraw |
| Contraintes / règles de gestion | -Connexion établie  -Format valide au champ Amount  -Solde suffisant |
| Niveau de priorité | Priorité haute |

|  |  |
| --- | --- |
| Fonction : Effectuer/Recevoir un virement | |
| Objectif | Effectuer et recevoir des virements bancaires vers un autre compte bancaire |
| Description | Mise à jour des données des comptes affectés par le virement suite à l’introduction du RIB, du montant et du PIN valide de la carte avec enregistrement dans le tableau d’historiques |
| Contraintes / règles de gestion | -Connexion établie  -Solde suffisant  -Format des champs montant, RIB & PIN valides |
| Niveau de priorité | Priorité haute |

|  |  |
| --- | --- |
| Fonction : Gestion de l’historique des transactions & virements | |
| Objectif | Ajout, lecture, tri ou suppression de l’historique des transactions & virements |
| Description | -Accès à l’historique des transactions & virements effectuée avec les informations relative et lecture en temps réel des tableaux grâce à un table-view avec possibilité de tri selon l’identifiant, le type ou la date  -Suppression de l’historique des transactions & virements |
| Contraintes / règles de gestion | -Connexion établie  -Chaque virement/transaction effectuée entraine l’enregistrement de cette action dans le tableau d’historiques adéquat |
| Niveau de priorité | Priorité haute |

|  |  |
| --- | --- |
| Fonction : Générer des documents | |
| Objectif | Générer des documents |
| Description | Générer des documents tels que la fiche des données personnelles de l’utilisateur ou l’historique des transactions effectuées |
| Contraintes / règles de gestion | -Connexion établie |
| Niveau de priorité | Priorité moyenne |

Quant aux besoins non-fonctionnels, Digi-Bank vérifie :

**Sécurité**• Besoins d’établissement de la connexion – niveaux d’accès « CRUD levels » (Create, Read, Update & Delete)

**Audit**• Éléments de données vérifiés   
• Champs de données vérifiés

**Performance**• Temps de réponse – le chargement de l’application, ouverture d’écran et des délais de rafraîchissement rapides.  
• En temps de traitement – fonctions, calculs, importations/exportations de données performants  
• L’interrogation de données et accès – temps de chargement initial et des chargements suivants rapides

**Capacité**• Bande passante – grand nombre de transactions/heure que le système est capable de traiter   
• Mémoire(Stockage) – système capable de stocker un grand nombre de données

**Disponibilité**• Les heures d’opération – 24h/24 sauf maintenances occasionnelles du serveur  
• Les emplacements d’opération – Ordinateur + Accès internet

**Fiabilité**• Moyenne des temps de bon fonctionnement – environ 24 heures par an  
• Le temps Moyen de Rétablissement – 1 heure

**Intégrité**• La capture des erreurs d’entrée-sortie   
• Le traitement des mauvaises données  
• Intégrité des données – intégrité référentielle dans tables de base de données et interfaces

**Rétablissement**  
• Validation des étapes importantes par demande de mot de passe  
• Fréquences des sauvegardes

**Compatibilité**  
• La compatibilité avec des applications tierces   
• La compatibilité sur des systèmes d’exploitation différents

**Aptitude à la maintenance**  
• La conformité au standards de développement – Manifeste Agile

**Ergonomie**• Les standards d’ergonomie –Densité faible & acceptable d’éléments sur écran, disposition maniable, flux & couleurs vives, Interfaçage & Dashboards dynamiques

**Documentation & traçabilité**• Eléments de documentation imprimables et historiques

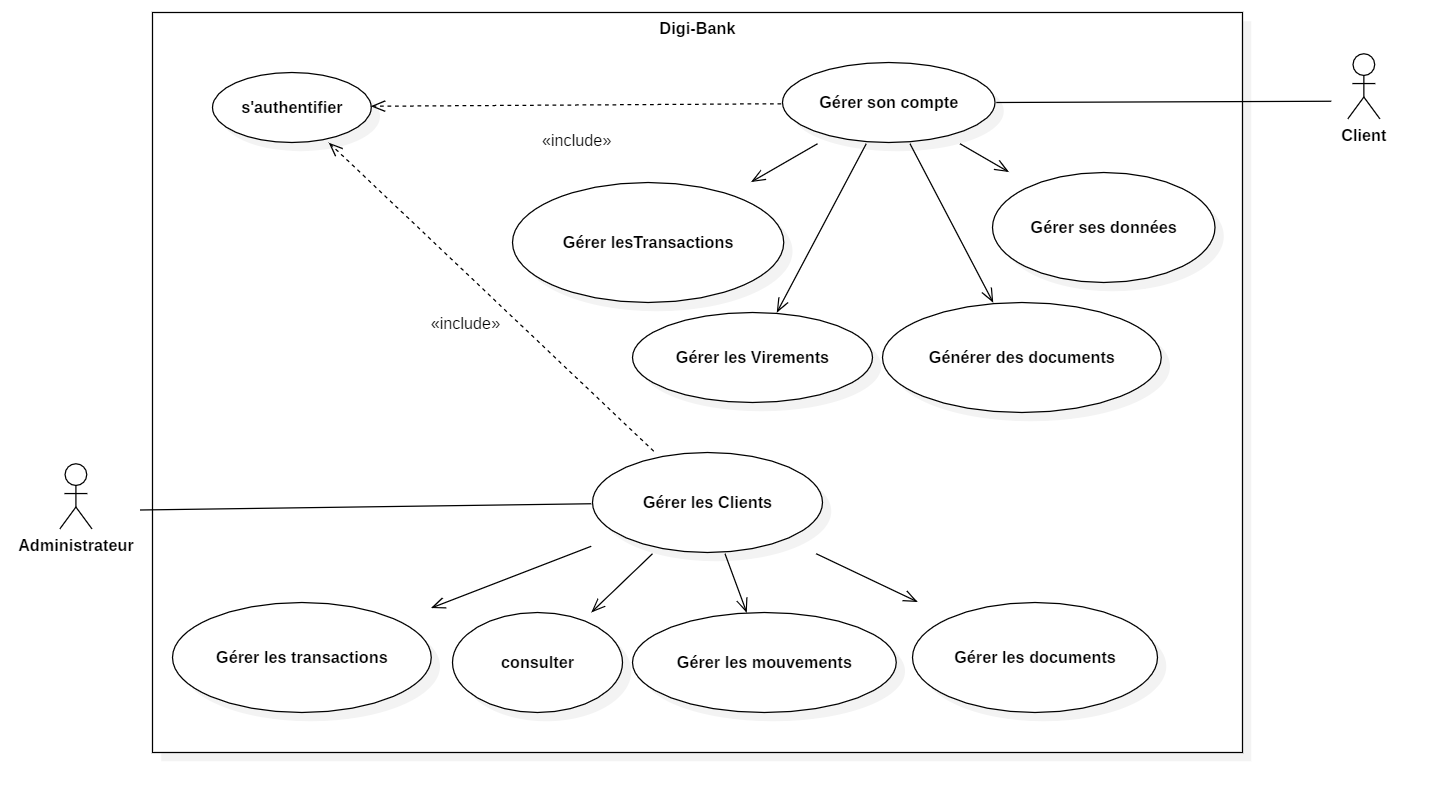
# **IV. Présentation de la partie analyse & conception**

# **A. Présentation de la démarche suivie**

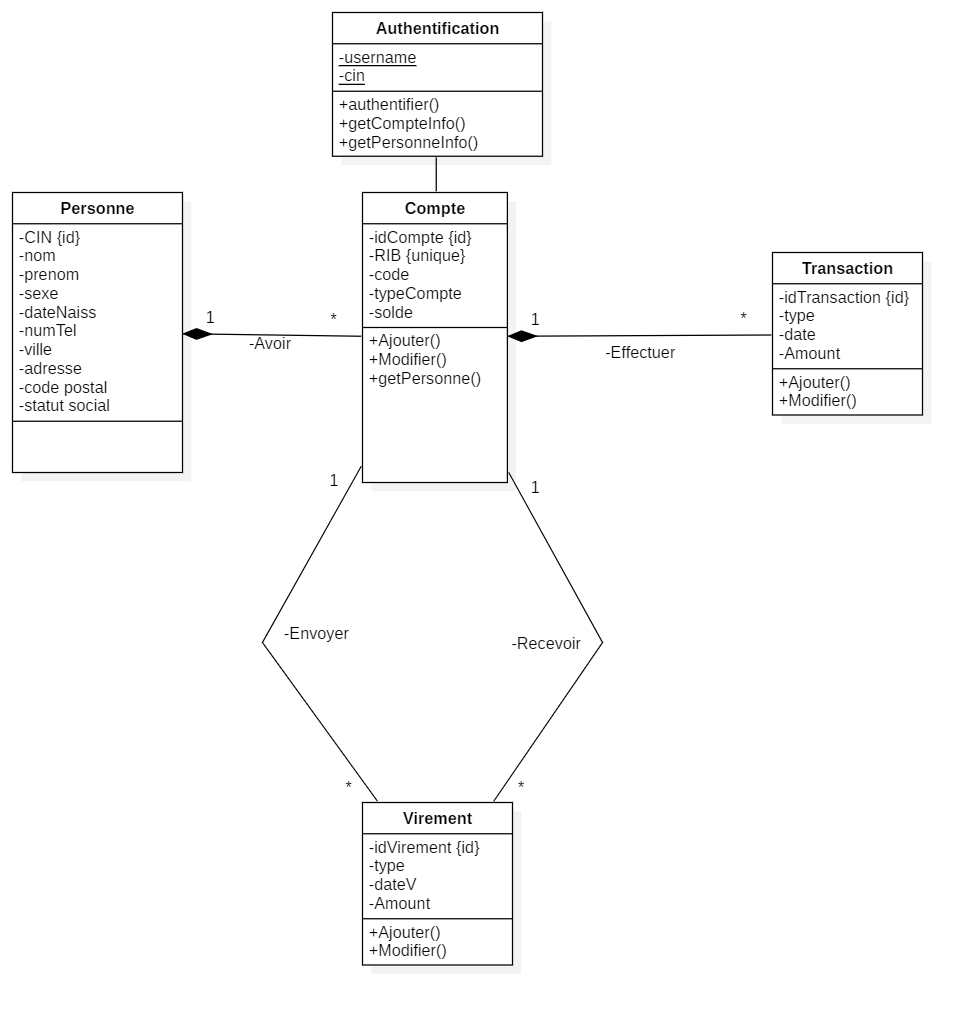
La priorité est la satisfaction du client. Pour cela, nous avons adopté le manifeste Agile dont les livraisons ont été rapides et régulières de façon à ce qu’il puisse d’une part vérifier l’avancement du projet, et d’autre part contrôler et tester qualitativement l’application au fur et à mesure du processus de création. Et ce, à travers les livraisons exécutables livrées lors des séances d’avancement de projet avec notre encadrante Mme BESSOUDA. Les demandes de changements ont été accueillies positivement, même tard dans le projet. Un processus d’amélioration continue est également mis en place afin que nous devenions de plus en plus efficaces au fil du temps.

# **B. Un petit aperçu des diagrammes**

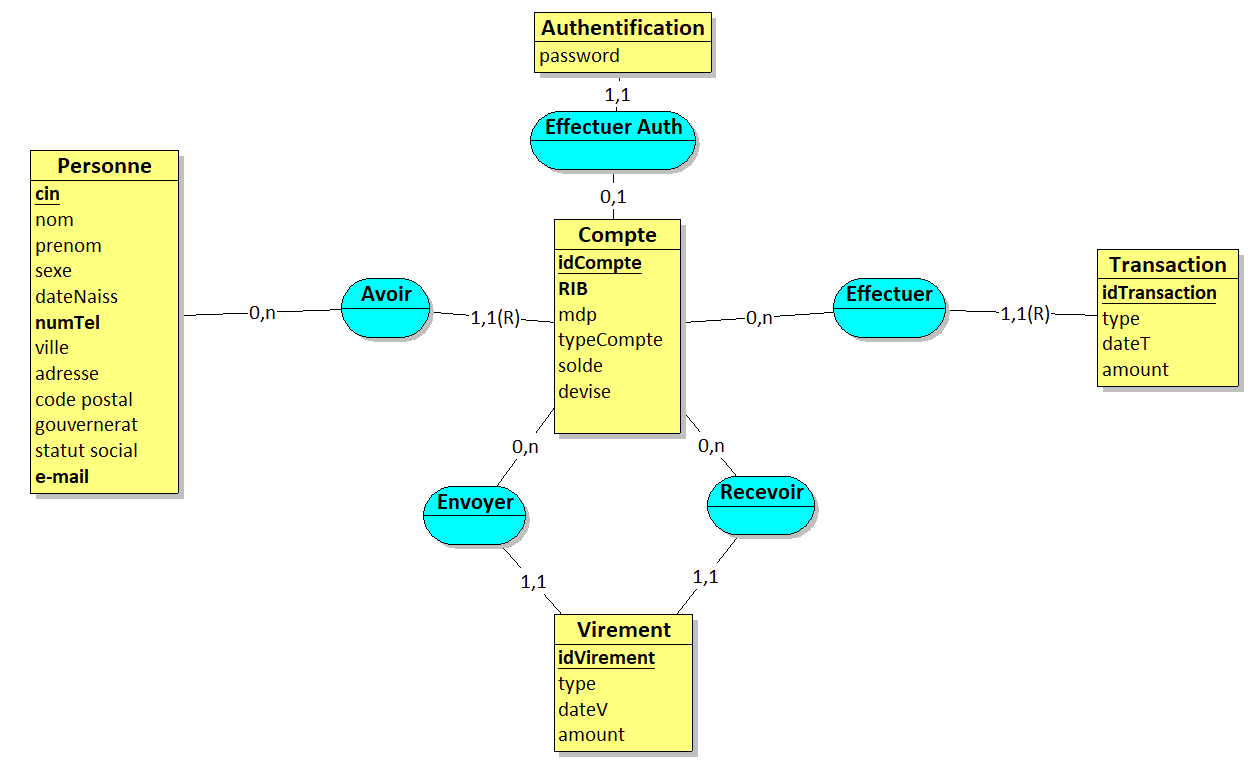
# **Diagramme des cas d’utilisations principal**



1. **Diagramme des classes**



**V. Présentation du modèle EA de la base de données**

****

**VI. Réalisation et tests de l’application**

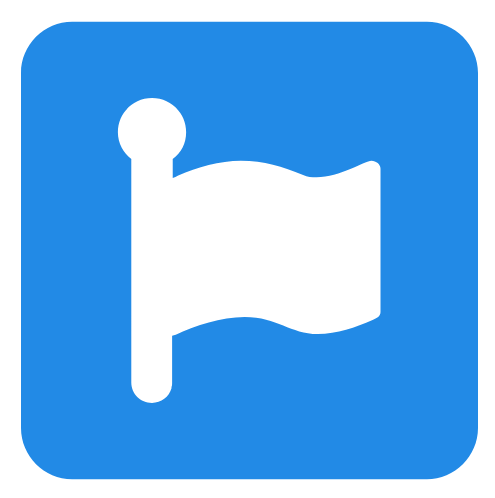
Préconditions :

* Il faut d’abord qu’il y ait un utilisateur dans la base de données sous le nom de « tpjava » avec le mot de passe « 123456 » vu que c’est une application « Local-Host » non encore déployée sur un serveur
* La modification de cet utilisateur dans la base de données s’effectue dans la classe « DBConn »
* Il faut avoir les bibliothèques :
  + **JFoenix-Open Collective** : Pour les boutons

animés et les « text fields » animés, etc.



* **Font Awesome Icons :** Pour les différentes icones



* **JavaMail :** Suite à l’implémentation des fonctionnalités de génération de mails automatique (Envoi du code de confirmation au cas de demande d’un nouveau mot de passe)

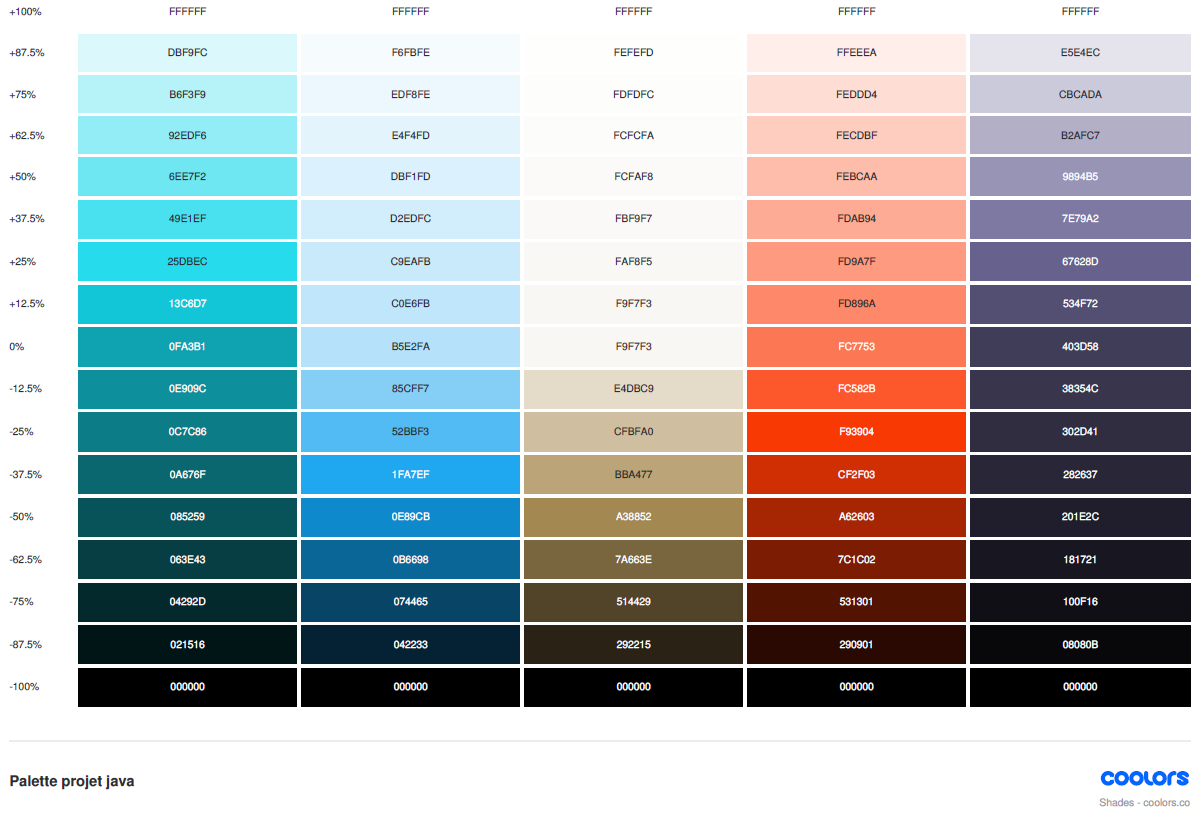


* **Oracle JDBC :** Le Greffon NetBeans/Interface de programmation créée par Sun Microsystems qui permet aux applications Java d'accéder par le biais d'une interface commune à des sources de données pour lesquelles il existe des pilotes JDBC.



Choix des couleurs propres à l’application & palettes utilisées





***TESTS & DEMONSTRATIONS : Soutenance Technique***

**VII. Conclusion**

Ce travail nous a permis d’acquérir une expérience personnelle et professionnelle unique dans son genre:

Un fruit de collaboration et de Teamwork : Il nous a été très bénéfique d'avoir eu la chance d’améliorer nos connaissances dans le domaine de la conception sur le plan théorique mais aussi de découvrir et d’acquérir de nouvelles connaissances en matière de programmation orientée objet en Java, en JavaFX et du développement des bases de données et la manipulation des produits Oracle en ce qui concerne l’aspect pratique & données.

Vu les délais & Deadlines rigides qui ont été préétablies, ce projet nous a tout à fait été un **challenge** : On a dû laisser quelques fonctionnalités comme la génération des documents et l’aspect administratif pour les développer après la soutenance ainsi que toute amélioration future et optimisation de cette application.

Mais finalement on a pu implémenter **toutes les fonctionnalités qu’on a vu en cours.**

Ceci dit, nous soussignés : Malek BARNAT & Feres NEFZI sommes ravis & fiers de ce projet semestriel de l’année universitaire 2020/2021 et sommes ouverts à vos remarques, votre Feed-Back et de futures collaborations !