

Introduction Générale

La digitalisation du secteur bancaire a permis le cumul d'un gros volume de données. Certes que la capacité de stockage de données et le temps de réponse et de calcul sont importants, mais l'analyse des données qui permet de transformer ces derniers en informations utiles est primordial.

Avec l'accumulation continue de données qui sont dispersées, non structurées, et hétérogènes, les banques se sont rendu compte que l'informatique traditionnelle se trouve inefficace vis-à-vis du gigantesque volume de données à traiter, alors ils doivent exploiter cette richesse de manière efficace afin de pouvoir dégager de l'information pertinente ce qui permettra d'améliorer leurs activités.

Pour répondre à une telle sollicitation, nous allons adopter l'informatique décisionnelle qui s'avère être la solution la plus adéquate. Il s'agit d'une sélection des informations opérationnelles et pertinentes qui seront par la suite normalisées pour l'entreposage, puis l'analyse et finalement la diffusion. De ce concept, est née alors la notion de modélisation multidimensionnelle, essentielle pour répondre aux exigences d'analyse. En outre, l'informatique décisionnelle permet de produire des indicateurs et des rapports à l'attention des analystes en proposant des outils de visualisation, de navigation et d'interrogation de l'entrepôt.

C'est dans cette perspective que s'insère notre projet qui a pour finalité concevoir et réaliser une application bancaire qui utilise l'informatique décisionnelle pour le traitement de données, avec des tableaux de bord qui offre une vision globale et détaillée sur l'activité de l'entreprise.

Afin d'aboutir à notre objectif nous allons commencer par présenter le contexte général du projet et la méthodologie adoptée pour la réalisation du projet, ensuite la présentation des concepts de base nécessaires à l'élaboration de ce projet, après nous allons aborder la phase d'analyse du projet en présentant les exigences, la partie suivante fait l'objet de la conception globale et détaillée, la dernière section est consacré pour la présentation des principaux aperçus graphiques de l'application. Nous clôturerons par une conclusion qui établit le bilan du travail et ouvre des nouvelles perspectives pour améliorer l'application.

Chapitre 1

Cadre Général du Projet

Dans ce chapitre, nous nous intéresserons tout d'abord au cadre général de notre projet. La première partie sera destinée au contexte général du projet, par la suite la méthodologie adoptée pour la réalisation de cette solution.

1.1 Contexte général du projet

1.1.1 Présentation de l'organisme d'accueil

Fondée en mars 2014 à Monastir, Tunisie, OdasLab est une entreprise spécialisée en développement informatique, conseil métier et opérationnel dans les secteurs de la finance, banques, l'assurance et la transformation digitale



FIGURE 1.1 – Logo OdasLab
[URL1]

1.1.2 Objectif du projet

Ce projet a pour objectif de concevoir et développer une application bancaire qui permet les activités quotidiennes de transactions bancaire tel que le transfert et le dépôt d'argent,

muni d'un tableau de bord et des rapport basé sur l'informatique décisionnelle qui vont aider à la prise de décision et offriront une vision globale et détaillée, ainsi la détection des fraudes grâce aux fouilles de données « data mining ».

1.2 Méthodologie Adoptée

Afin de réduire la complexité des problèmes et de respecter les besoins du client dans les délais prévus aussi de garantir coût-efficacité, nous devons suivre une bonne méthode de gestion des projets informatiques qui garantie la flexibilité tout au long du cycle de développement.

C'est pour cette raison, ce projet adoptera la méthodologie agile, spécifiquement la méthodologie Kanban.

1.2.1 Méthodologie agile

« Une méthode agile est une approche itérative et incrémentale, qui est menée dans un esprit collaboratif avec juste ce qu'il faut de formalisme. » [Veronique, 2013]

Cette approche forme un ensemble de méthodes permettant en effet de gérer essentiellement les projets informatiques. La particularité de la méthode agile réside dans le fait qu'elle applique un principe de développement itératif. Ce principe consiste à découper le projet en « itérations ». Ces itérations présentent des mini-projets définis au préalable avec le client en déterminant les différentes fonctionnalités à développer ainsi que leurs priorités. Cette méthode permet d'éviter les imprévus rencontrés en cours du projet grâce à sa flexibilité aux changements et son adaptabilité aux modifications.

1.2.2 Kanban

La méthode Kanban inspirée du constructeur automobile Toyota au Japon, et de son organisation dite "lean". C'est aussi un système visuel très utilisé en développement agile. La méthode Kanban visait à améliorer la production en se basant sur le juste-à-temps et sur des processus évolutifs. Aujourd'hui, elle est à la fois une méthode de gestion de projet et un outil, guidée par deux principes : le visuel et le temps-réel. Pour cela, les équipes travaillent avec des tableaux et post-it, appelés étiquettes.

kanban permet de :

- Visualiser le flux des travaux (workflow).
- Limiter le nombre de tâches en cours.
- Gérer le déroulement du travail.
- Établir des règles d'organisation.
- Proposer des actions d'amélioration.

Fonctionnement :

Présentez chaque user story sous forme d'affichette, et positionnez-la sur un tableau en fonction de son statut d'avancement[URL2] . En développement agile, vous pouvez déplacer les affichettes selon 3 statuts :

- Aucune tâche de la user story n'a encore été exécutée (To do/À faire).
- La user story est en cours de réalisation (In progress/Doing/En cours).
- Toutes les tâches de la user story sont réalisées (Done/Terminé).

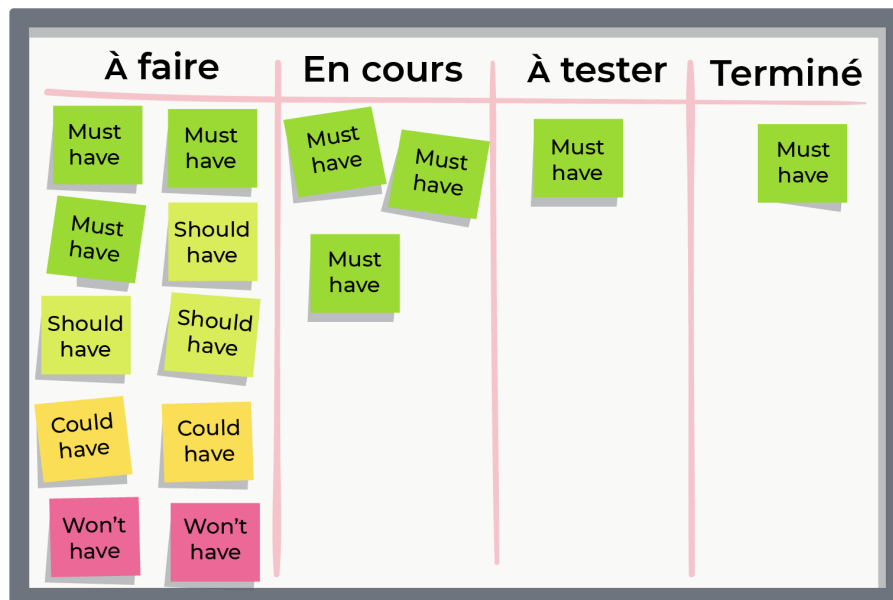


FIGURE 1.2 – Tableau Kanban
[URL3]

Conclusion

Dans ce premier chapitre, nous avons présenté le contexte général du projet suivi par la méthodologie adopter. la prochaine partie sera consacré à une étude préliminaire nécessaire pour l'élaboration du projet.

Chapitre 2

Étude Préalable

Dans dans cette partie, nous nous intéresserons aux concepts de base nécessaires à l'élaboration de projet. Par la suite, nous ferons une étude de l'existant en dégagant les points faibles . Enfin, nous allons nous intéresser à la présentation de la solution proposée.

2.1 Concept général

2.1.1 Business Intelligence

L'informatique décisionnelle (En anglais business intelligence (BI)) est l'informatique à l'usage des décideurs et des dirigeants d'entreprises. «Elle désigne les moyens, les outils et les méthodes qui permettent de collecter, consolider, modéliser et restituer les données, matérielles ou immatérielles, d'une entreprise»[URL4] en vue d'offrir une aide à la décision et de permettre à un décideur d'avoir une vue d'ensemble de l'activité traitée.

« La Business Intelligence permet aux entreprises de mieux comprendre et d'analyser voire même d'anticiper ce qui se passe en leur sein et dans l'environnement économique en général»[michel franco, 2010]. La BI permet de transformer des données en informations utiles et pertinentes, pour faciliter la prise de décisions opportunes en toute connaissance de cause. Elle combine les données issues de sources diverses pour offrir une vue d'ensemble unifiée et actualisée.

la figure suivante montre les avantages de la business intelligence.



FIGURE 2.1 – Avantages de la Business Intelligence

À la suite de ces avantages, nous introduisons les concepts clés qui forment la chaîne appelée Modèle DIKW. Il met en œuvre les notions de bases expliquées ci-dessous et leur hiérarchie.

- **Données** : ce sont les faits bruts. C'est le résultat direct d'une mesure qui peut être obtenu par un outil de mesure.
- **Information** : c'est une donnée avec une signification, liée à un contexte.
- **Connaissance** : c'est le résultat des analyses et des réflexions sur les informations obtenues. Elle se base sur le suivi et l'interprétation des informations et des expériences.
- **Sagesse** : c'est savoir utiliser les connaissances, c'est donc l'art de la prise de décisions.

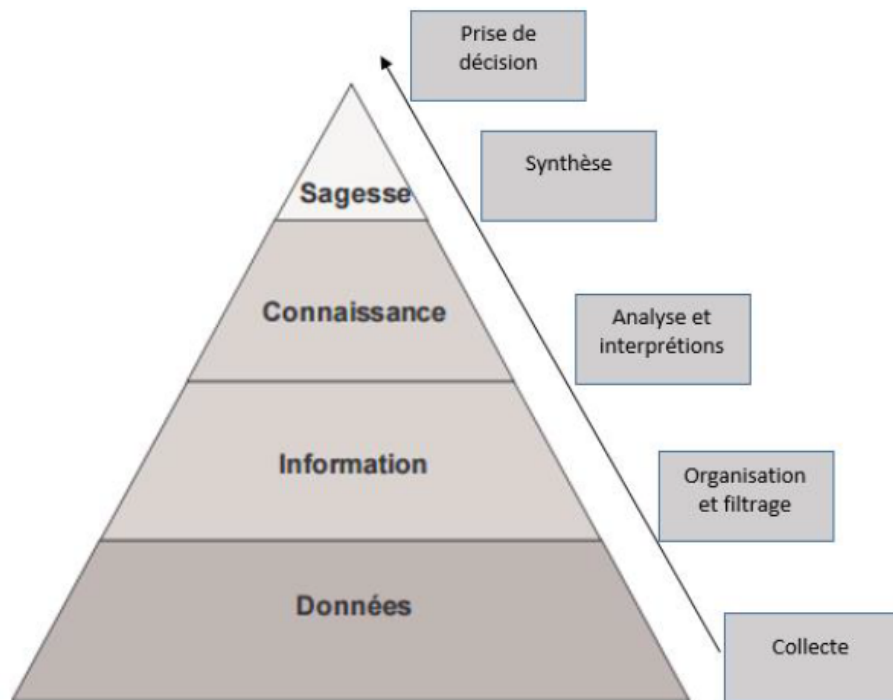


FIGURE 2.2 – Modèle DIKW
[URL5]

Les phases de l'analyse décisionnelle

L'informatique décisionnelle désigne les moyens, les outils et les méthodes qui permettent de collecter, intégrer, organiser et restituer les données immatérielles d'une entreprise en vue d'offrir une aide à la décision et de permettre aux responsables de la stratégie d'entreprise d'avoir une vue d'ensemble de l'activité traitée. Pour la réussite d'un projet BI il est nécessaire de mettre en place différentes étapes dans une chaîne décisionnelle afin de profiter pleinement et efficacement d'une plateforme de business intelligence.

les phases de l'analyse décisionnelle sont

- **Collecter les données** : extraire les données des systèmes de production et les adapter à un usage décisionnel.
- **Intégrer les données** : stocker les données structurées et traitées.
- **Organiser** : distribuer et faciliter l'accessibilité des informations.
- **Restituer** : exploiter du mieux possible l'utilisateur afin d'extraire l'information

des données a cet usage.
le schéma suivant résume les 4 phases de l'analyse décisionnelle

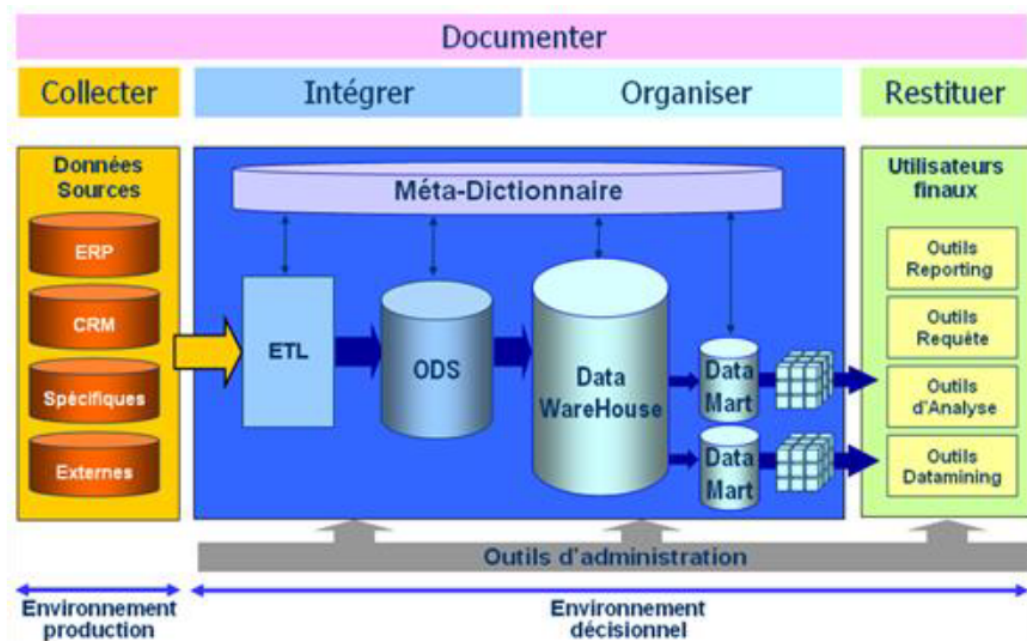


FIGURE 2.3 – Phases de l'analyse décisionnelle
URL6

ETL (Extract, Transform, Load)

Signifie Extract-Transform-Load. C'est un processus d'entreposage de données chargé de retirer des données des systèmes source et de les placer dans un entrepôt de données.

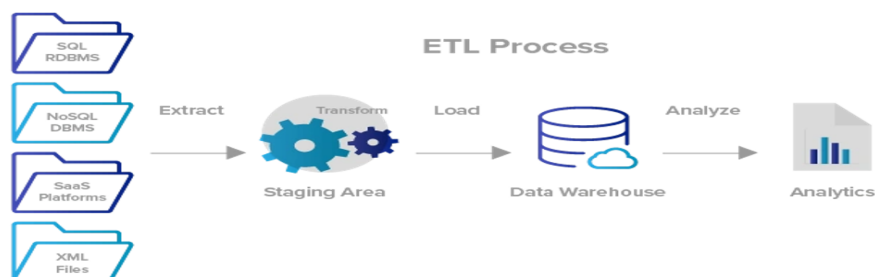


FIGURE 2.4 – Processus Extract-Transform-Load
[URL6]

Extract

L'étape d'extraction comprend l'extraction de données à partir du système source et le rend accessible à un traitement ultérieur. L'étape de nettoyage est l'une des plus importantes car elle garantit la qualité des données dans l'entrepôt de données.

Transform

L'étape de transformation applique un ensemble de règles pour transformer les données de la source vers la cible. Cela comprend la conversion de toutes les données mesurées à la même dimension en utilisant les mêmes unités pour qu'elles puissent ensuite être rejointes. L'étape de transformation exige également de joindre les données provenant de plusieurs sources, générant des agrégats, des clés de substitution, le tri, dériver de nouvelles valeurs calculées et l'application des règles de validation avancées.

Load

Lors de l'étape de chargement, il est nécessaire de veiller à ce que le chargement soit exécuté correctement et avec le moins de ressources que possible. La cible du processus Load est souvent un chargement vers la base de données.

Data Warehouse

Le Data Warehouse est une base de données recueillant et gérant toutes les données collectées au sein de l'organisme, dans le cadre de la prise de décision. Son but principal est de soutenir l'analyse et faciliter le processus de prise de décision.

Data Warehouse présentent quatre caractéristiques spécifiques :

- **Orienté sujet** : les données sont organisées par thème .
- **Intégré** : les données provenant de sources hétérogènes, elles utilisent chacune un type de format. Elles doivent donc être intégrées avant d'être proposées à utilisation.
- **Non volatile** : les données ne disparaissent pas et ne changent pas au fil des traitements, au fil du temps.
- **Historisé** : les données sont horodatées, afin de visualiser l'évolution dans le temps d'une valeur donnée.

OLAP

OLAP est un acronyme Online Analytical Processing. OLAP effectue une analyse multidimensionnelle des données et offre la possibilité de calculs complexes, d'analyse des tendances et de modélisation sophistiquée des données. OLAP permet aux utilisateurs finaux d'effectuer une analyse des données en plusieurs dimensions, fournissant ainsi la compréhension dont ils ont besoin pour une meilleure prise de décision.



FIGURE 2.5 – Cube OLAP
[URL7]

Le tableau de bord

Le tableau de bord appartient à l'ensemble des outils de pilotage qui permettent aux gestionnaires de prendre connaissance de l'état de l'évolution des systèmes qu'ils pilotent et d'identifier les tendances. C'est un outil qui nous renvoi une synthèse du passé et des prévisions sur le futur en intégrant des alertes aux décideurs pour raffiner au meilleur la prise de décision. Ce fut un outil d'évaluation de la performance par le biais des indicateurs sur des périodes précises.

Les tableaux de bord jouent un rôle essentiel dans le suivi des performances de l'organisation. Ils remplissent plusieurs fonctions.

- **Responsabilisation** : Implication des équipes dans l'identification, la réalisation et le suivi des performances.
- **Mesure du progrès** : Information sur le degré de réalisation des objectifs, mise

en évidence des écarts entre prévisions et réalisations grâce aux indicateurs de performance.

- **Anticipation** : Alertes dès lors qu'une tendance met en évidence un risque ou un problème.
- **Aide à la décision** : Diagnostic de la situation et analyse de l'information pour faire des choix pertinents.

Le reporting

C'est la préparation des rapports en se basant sur des activités et des objectifs souhaités. C'est une présentation périodique sous une structure prédéterminée telles que des présentations graphiques et des grandeurs de calculs afin d'avoir des bilans pertinents. L'objectif du reporting est d'informer les chargés de contrôle et de supervision en interne comme en externe, ou bien tout simplement ceux qui sont concernés par ces activités ou ces résultats, on parle alors de dirigeants et décideurs.

2.1.2 Data minning

Data Mining, également appelée découverte de données ou de connaissances, consiste à analyser des données et à les transformer en un aperçu qui informe les décisions d'affaires. C'est ainsi le procédé qui permet d'extraire des corrélations, dégager et détecter des tendances, découvrir des patterns entre les données permettant d'expliquer le passé ou prédire le futur ainsi que de dégager de la valeur à partir de données auparavant inexploitable. Ce que l'on peut traduire par la transformation des données en connaissances.

Parmi les objectifs du data mining :

- Prédire les tendances futures.
- Déterminer les habitudes de la clientèle.
- Aide à la prise de décision.
- Analyse basée sur le marché.
- Détection rapide de la fraude.

Processus d'Extraction de Connaissances à partir des Données

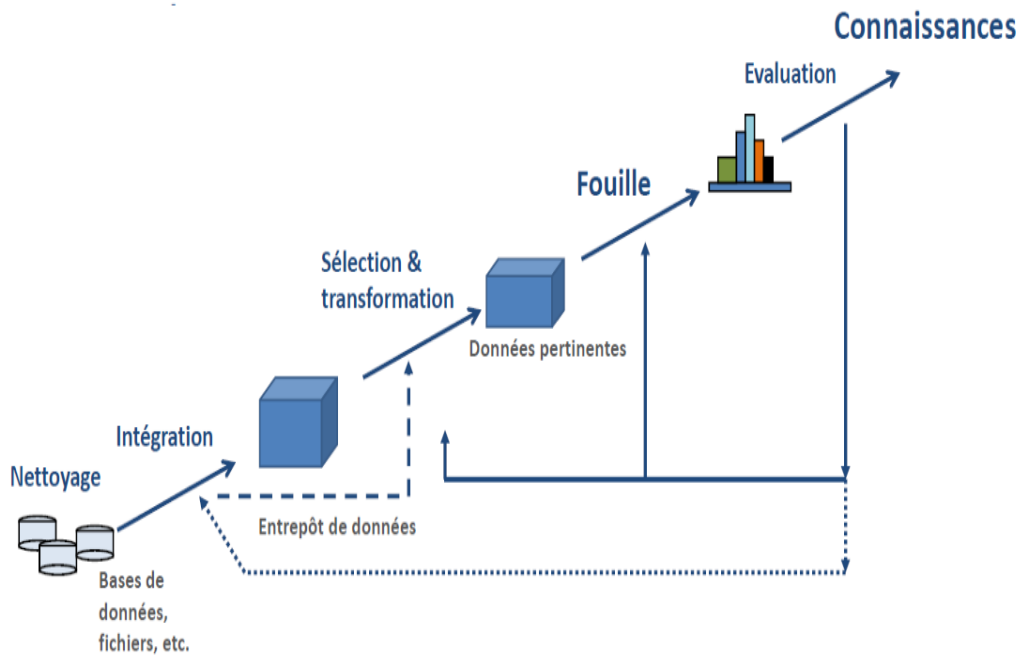


FIGURE 2.6 – Processus d'Extraction des Connaissances
[URL8]

L'exploration de données est un processus itératif qui implique généralement les phases suivantes :

1. Développer et comprendre le domaine de l'application

C'est le premier pas dans ce processus. Il prépare la scène pour comprendre et développer les buts de l'application.

2. Sélection des données

La sélection et la création d'un ensemble de données sur lequel va être appliqué le processus d'exploration.

3. Le prétraitement et le nettoyage des données

Cette étape inclut des opérations comme l'enlèvement du bruit et des valeurs aberrantes si nécessaire, des décisions sur les stratégies qui vont être utilisées pour traiter les valeurs manquantes.

4. La transformation des données

Cette étape est très importante pour la réussite du projet et doit être adaptée en fonction de chaque base de données et des objectifs du projet. Dans cette étape nous cherchons les méthodes correctes pour représenter les données. Ces méthodes incluent la réduction des dimensions et la transformation des attributs.

5. Choisir la meilleure tâche pour Data mining

Nous devons choisir le type de Datamining à utiliser et en décidant le but du modèle. Par exemple : classification, régression, regroupement...

6. Choisir l'algorithme de Data mining

Dans cette étape nous sommes amenés à choisir la méthode spécifique pour faire la recherche des motifs, en décidant quel modèle et des paramètres sont appropriés.

7. Implémenter l'algorithme de Data mining

Dans cette étape nous procéderons à l'implémentation des algorithmes de Data mining choisis dans l'étape antérieure. Il se peut qu'on applique l'algorithme plusieurs fois pour avoir le résultat attendu.

8. Evaluation

Evaluation et interprétation des patterns découverts. Cette étape donne la possibilité de :

- Retourner à une des étapes précédentes.
- Avoir une représentation visuelle des motifs.
- Enlever les motifs redondants ou non-représentatifs.
- Les transformer dans des termes compréhensibles pour l'utilisateur. .

9. Utiliser les connaissances découvertes

Incorporation de ces connaissances dans d'autres systèmes pour d'autres actions. Nous devons aussi mesurer l'effet de ces connaissances sur le système, vérifier et résoudre les conflits possibles avec les connaissances antérieures.

2.2 Étude et critique de l'existant

Actuellement, OdaLab dispose d'une application qui permet de gérer les activités quotidiennes de transférer, retirer et déposer l'argent. Certes les outils de l'application bancaire sont satisfaisants pour les transactions mais d'un autre côté ces outils présentent plusieurs

déficiences, comme :

- **Absence de Business Intelligence** : A l'heure actuelle, l'application bancaire ne possède pas un outil BI qui permet la génération des rapports et tableau de bord et qui faciliteront la prise de décision et offriront une vision globale et détaillée. Les rapports et les tableaux de bord sont générés à l'aide de EXCEL.
- **Difficulté dans l'élaboration des rapports d'activité** : L'élaboration des rapports d'activité fait intervenir plusieurs intermédiaires. C'est à dire qu'à chaque fois qu'il est nécessaire d'établir les rapports d'activité, il faudra procéder à l'extraction des données à partir de plusieurs sources de données qui sont réparties sur plusieurs serveurs pour les diriger vers une structure centralisée.
- **Absence de moyen pour découvrir les transactions frauduleuses** : lors qu'il y a des transactions frauduleuses c'est-à-dire en cas d'opérations pour lesquelles vous n'avez pas donné votre consentement l'application bancaire ne permet pas de les détectées et les découvrir.
- **Absence de moyen d'aide à la décision** : manque de moyen de diagnostic de la situation et d'analyse de l'information pour faire des choix pertinents.
- **Lenteur de la procédure de Reporting** : Le processus de Reporting actuel confronte certaines difficultés. Les décideurs ont besoin des rapports dans les brefs délais. Vu la charge du travail et la lourdeur de la procédure, ce processus prend plus du temps qu'il le faut.

2.3 Solution proposée

Rappelons l'objectif de ce projet est de concevoir et développer une application bancaire qui permet de gérer les activités quotidiennes de transactions, avec un tableau de bord et des rapports qui sont basé sur l'informatique décisionnelle pour aider à la prise de décision et qui offriront une vision globale et détaillée, ainsi la détection des fraudes grâce aux fouilles de données « data mining ».

En effet, la solution adéquate consiste à concevoir un « système centralisé », un entrepôt de données rassemblant toutes les données relatives. L'idée est d'arriver, à la fin du processus, de générer des rapports de qualité ainsi que la construction d'un tableau de bord destiné aux décideurs pour avoir une vision profonde sur l'activité de l'entreprise et qui mène à des décisions innovantes pour bien améliorer le domaine d'activité de la banque.

La solution décrite ci-dessus doit permettre :

- La conception d'une application permettant les transactions bancaires.
- La conception d'un Data Warehouse de données centralisé, intégré, non volatile et historisé.
- L'extraction des données depuis le Data Warehouse pour construire des cubes qui sont des éléments clés de l'analyse multidimensionnelle.
- L'utilisation des cubes et du Data Warehouse dont la mise à jour est automatique pour créer des rapports et des tableaux de Bord.
- Déploiement des tableaux de bord et des rapports et leur envoie automatique par courrier électronique « email ».
- Une analyse fine des données en utilisant la technique de data mining afin de détecter les activités frauduleuses.

Conclusion

Durant ce chapitre, nous avons présenté les concepts nécessaires pour élaborer ce projet. Ensuite, nous avons étudié l'existant en soulignant ses limites, ensuite, nous avons proposé une solution qui répond aux besoins de l'utilisateur. la prochaine partie sera consacré à l'analyse et la spécification des besoins.

Chapitre 3

Analyse et Spécification des Besoins

Afin de réussir notre système décisionnel, notre projet doit être bien détaillé et claire. Dans cette section, nous commencerons par présenter les attentes à réaliser dans notre projet à travers les besoins fonctionnels et les besoins non fonctionnels, par la suite la spécification semi-formelle de ces besoins à travers des diagrammes.

3.1 Analyse des besoins

Une bonne spécification des besoins est primordiale pour bien représenter le travail d'un projet. Donc, cette phase consiste à détailler les différents besoin fonctionnels et non fonctionnels qui doivent être pris en considération par notre application.

3.1.1 Identification des acteurs

Tout système est conçu pour interagir, directement ou indirectement avec des entités externes (des acteurs). « Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. » [URL9]. Un acteur représente une personne ou un autre système informatique qui attend un ou plusieurs services offerts par une interface d'accès.

Notre application est destinée à un seul acteur qui est l'utilisateur de l'application bancaire.

3.1.2 Besoins fonctionnels

Les besoins fonctionnels consistent à décrire les différentes fonctionnalités offertes par l'application pour répondre aux attentes des utilisateurs.

Les besoins fonctionnels sont les suivants :

- Transférer, retirer et déposer l'argent des comptes clients.
- Utiliser le Tableau de bord basé sur l'informatique décisionnelle qui permet de prendre connaissance de l'état de l'évolution des systèmes et d'identifier les tendances.
- Générer des rapports dynamiques qui peuvent être exportés ou imprimés sous formats PDF et leur envoi automatique par courrier électronique « email ».
- Analyser les données en utilisant la technique de datamining afin de détecter les activités frauduleuses.

3.1.3 Besoins non fonctionnels

En plus des besoins fonctionnels cités, le projet doit répondre à certaines exigences dites besoins non fonctionnels. Ce sont des besoins en matière de performance, de type de conception ou le type de matériel. Dans ce qui suit, nous citons les plus importants parmi eux :

- **La sécurité** : Assurer la sécurité et la confidentialité des données des clients de la banque.
- **La performance** : L'application doit répondre aux besoins sans ambiguïté.
- **L'ergonomie** : L'utilisation de l'application doit être assez simple et le design de l'interface doit être compréhensible et faciles à utiliser.
- **L'évolutivité** : L'application doit avoir la capacité d'évoluer pour répondre aux futures exigences de la banque.

3.2 Spécification des besoins

Cette section offre une meilleure compréhension des exigences mentionnées en les déclarant de manière semi-formelle grâce aux diagrammes.

3.2.1 Diagramme de cas d'utilisation

Nous présentons les besoins de notre application en utilisant le diagramme des cas d'utilisation dont l'objectif est d'obtenir une vision globale du comportement fonctionnel de notre système.

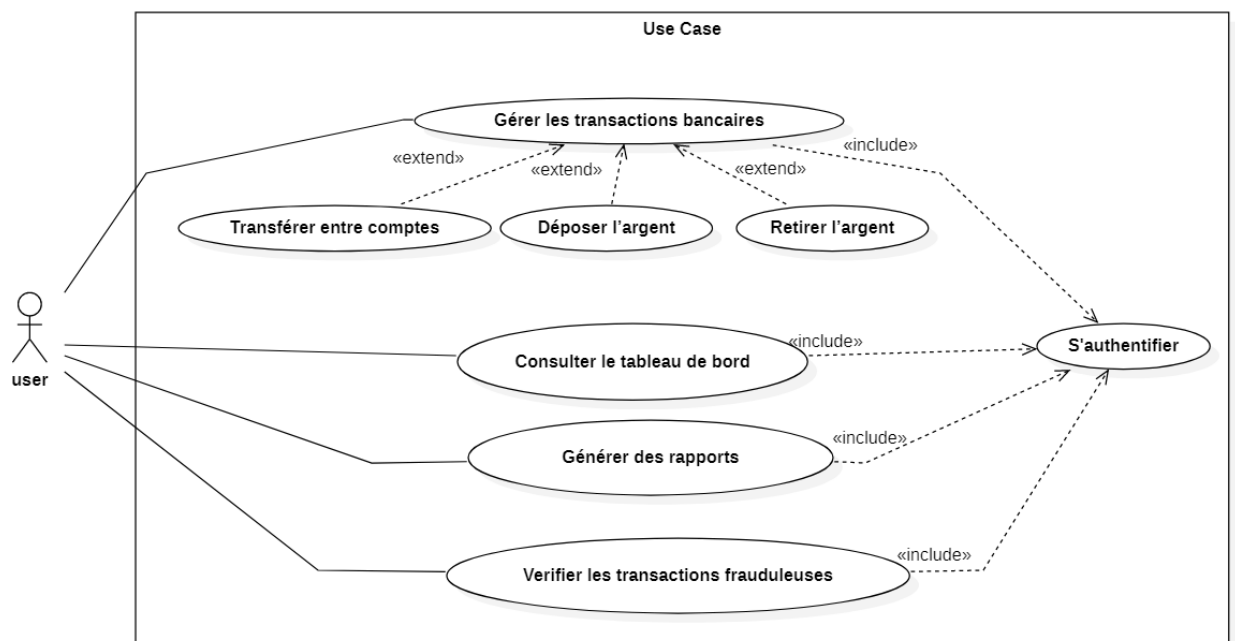


FIGURE 3.1 – Diagramme de cas d'utilisation

3.2.2 Diagramme de séquence système

Dans cette partie, nous présentons les besoins de notre application en utilisant les diagrammes de séquence dont l'objectif est d'obtenir une vision globale du comportement dynamique de notre système.

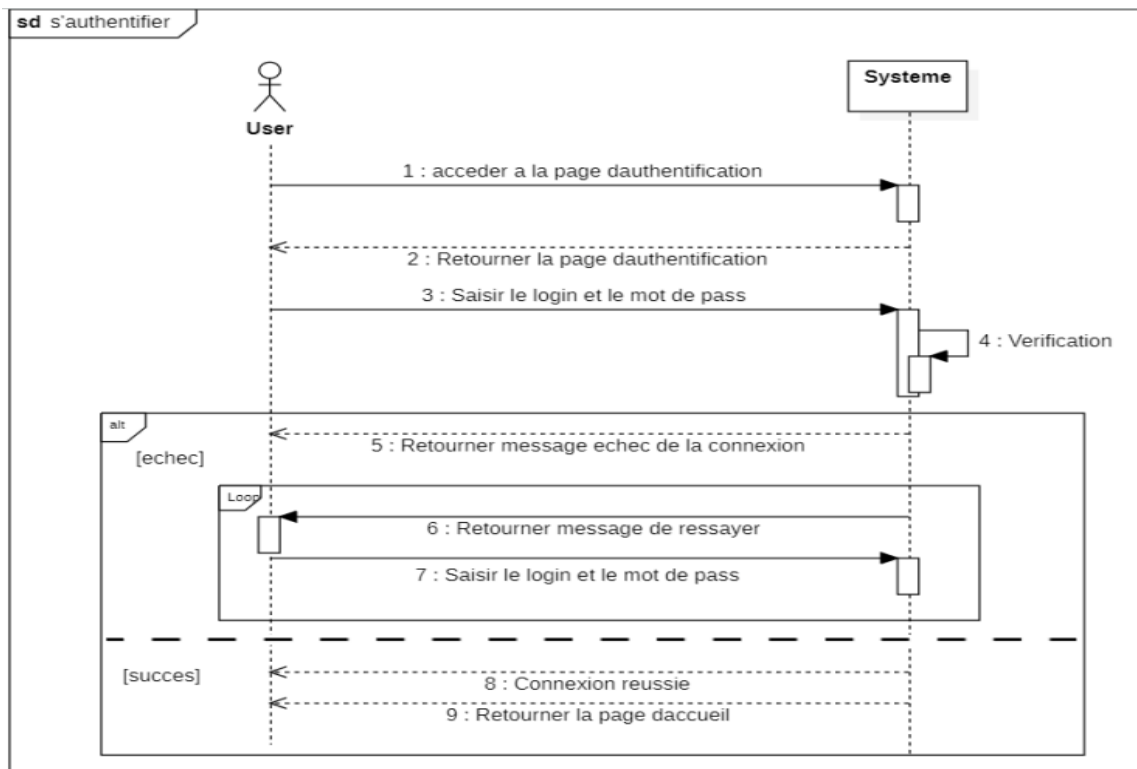


FIGURE 3.2 – Diagramme de séquence authentification

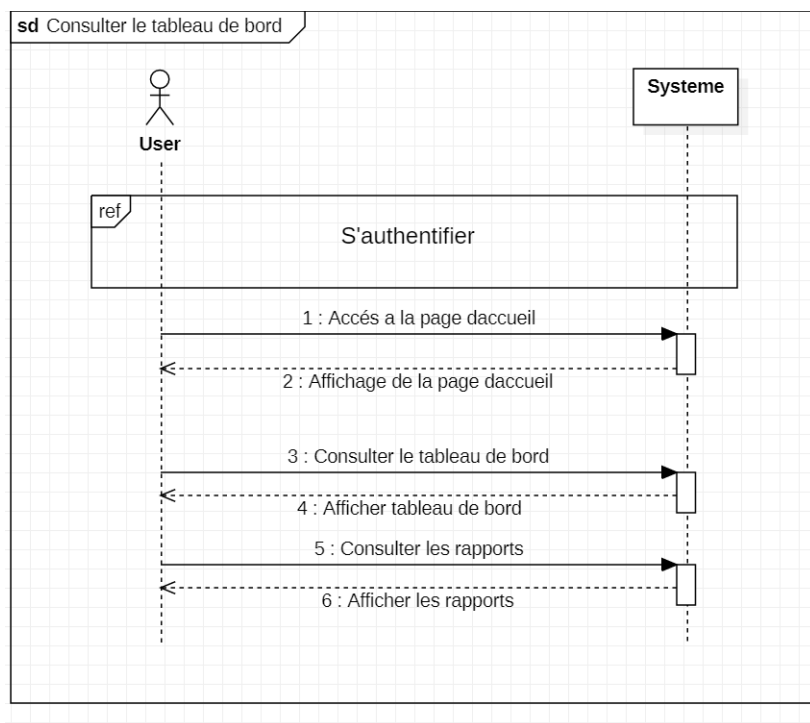


FIGURE 3.3 – Diagramme de séquence consulter le tableau de bord

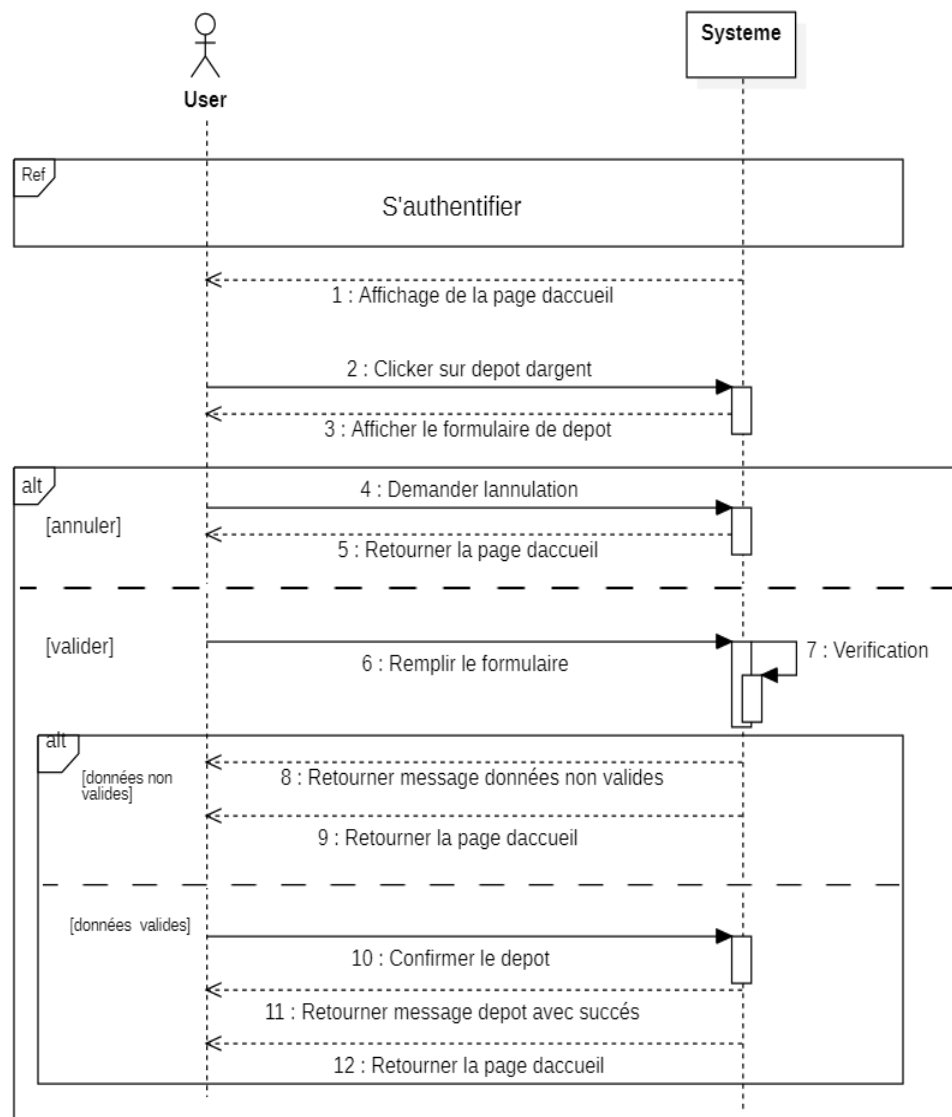


FIGURE 3.4 – Diagramme de séquence dépôt d'argent

Conclusion

Dans cette section nous avons commencé par préciser les différents besoins fonctionnels et non fonctionnels du projet puis la spécification semi-formelle de ces besoins à travers des diagrammes. Dans le chapitre qui suit nous allons présenter la conception de notre projet.

Chapitre 4

Conception

Après avoir spécifié et analysé les différents besoins à satisfaire, ce chapitre s'intéressera à présenter l'aspect conceptuel, la conception est un processus créatif permettant la description du fonctionnement du système afin d'en faciliter la réalisation. Elle constitue une étape préliminaire préparatoire de la phase d'implémentation de la solution adoptée.

4.1 Conception globale

La planification de l'architecture des applications est une étape importante dans le processus de conception. Elle dépend de divers facteurs tels que les exigences de performance, l'extensibilité, la modularité et les perspectives. Dans cette partie, nous allons donc décrire l'architecture physique et logique sur laquelle nous allons élaborer notre projet.

4.1.1 Architecture physique

La figure ci-dessous montre une représentation de l'architecture physique de notre application, composée principalement d'une couche de présentation, d'un serveur d'application et d'un serveur de base de données.

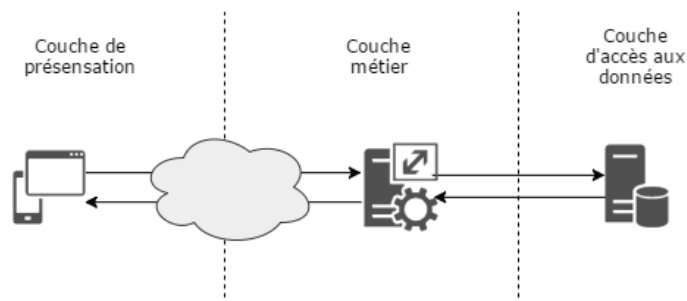


FIGURE 4.1 – Architecture physique 3 tiers
[URL10]

- **Couche de présentation** : ou affichage si l'on souhaite, correspondant à l'affichage qui interagit avec l'utilisateur à travers le navigateur.
- **Le serveur d'application** : c'est le serveur qui contient l'ensemble des règles de gestion et de la logique applicative.
- **Le serveur de base de données** : c'est un serveur de base de données utilisé pour stocker la partie persistance des données de notre application.

4.1.2 Architecture logique

La structuration de notre application se traduit par une décomposition logique de chaque application en 5 couches :

- **Présentation** : la couche présentation gère et assure l'affichage de l'interface graphique utilisateur ou les Interfaces Homme-Machine IHM.
- **Contrôleur** : le contrôle de la cinématique des écrans et l'invocation des appels de services.
- **Services** : correspond aux traitements qu'effectue l'application elle représente l'implémentation de la logique des cas d'utilisation use-case fonctionnels.
- **Domaine** : responsable de la gestion des objets métiers, elle recense les objets métiers manipulées par l'application.
- **Persistance** : fournit des services de stockage des données et de mapping entre les différentes formes des objets métiers du domaine.

La Figure suivante illustre une représentation de l'architecture logique de notre application qui est l'architecture multicouche, elle nous donne un aperçu sur la relation entre les différentes parties.

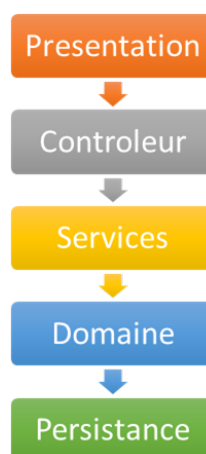


FIGURE 4.2 – Architecture logique

4.2 Conception détaillée

Cette partie a pour objectif de transformer le modèle d'analyse en un modèle concret qui aide le développeur à réaliser son application, nous présentant dans ce qui suit une vue statique et dynamique de notre application moyennant les diagrammes appropriés.

4.2.1 Diagramme de classes

La figure suivante présente le diagramme de classes relatif à notre application.

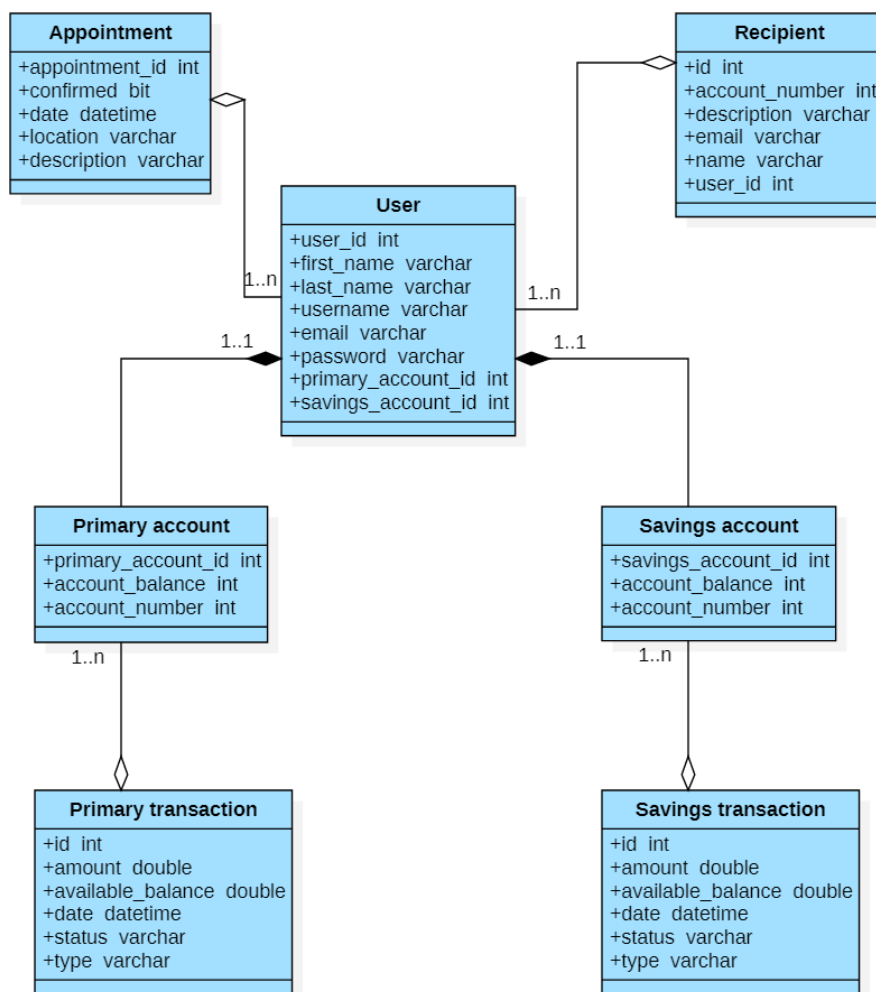


FIGURE 4.3 – Diagramme de classes

4.2.2 Diagramme de la base de données

La figure suivante est un modèle de données ou diagramme pour des descriptions de haut niveau de modèles conceptuels de données

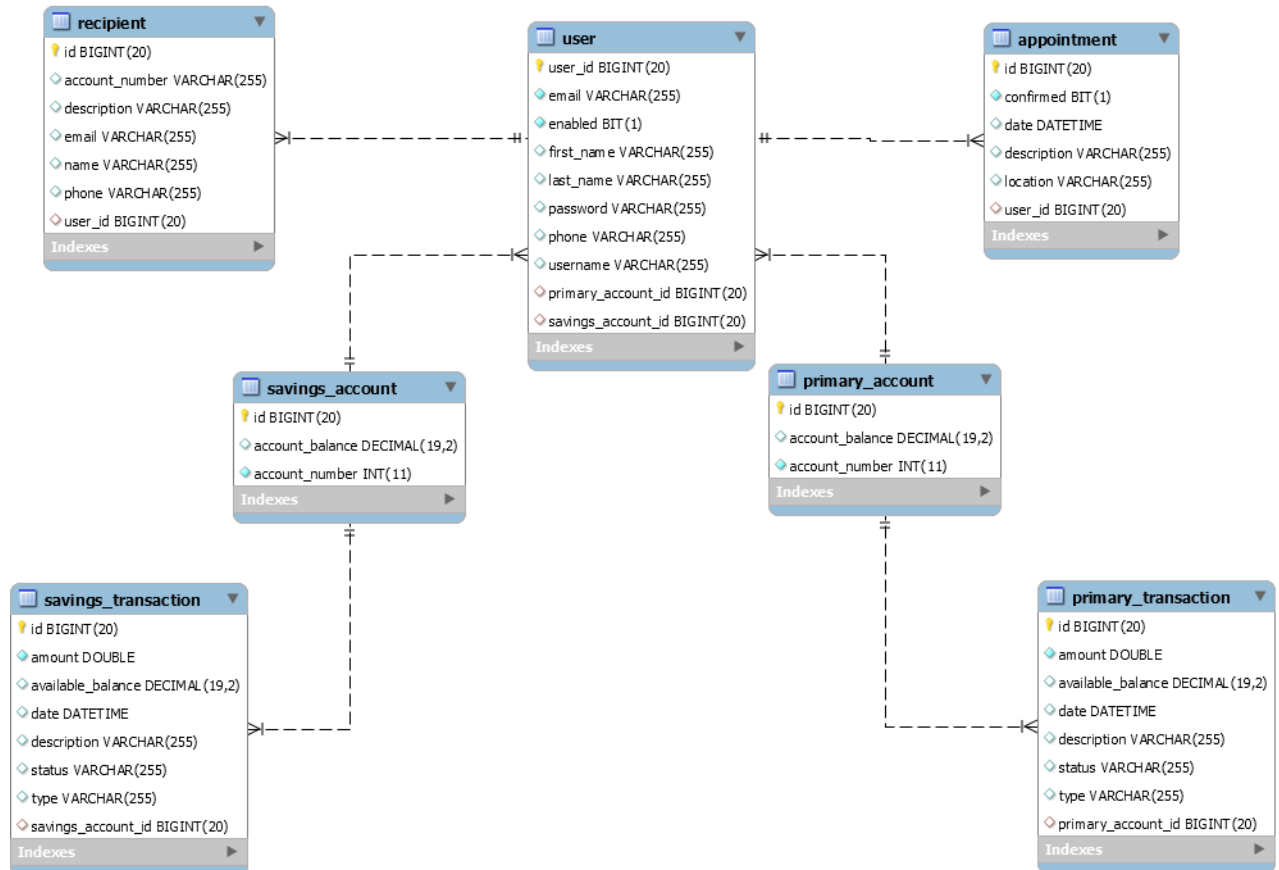


FIGURE 4.4 – Diagramme entité association

4.2.3 Conception de l'entrepôt de données

Dans un contexte d'un projet décisionnel, notre objectif c'est d'arriver à restaurer un ensemble de tableaux de bords qui se basent sur des indicateurs de performance (KPI) générés à partir de données instaurées dans l'entrepôt de données. Pour se faire, il faut que nous identifions le modèle conceptuel de l'entrepôt et les dimensions et les mesures retenus.

Etude du modèle conceptuel

Il existe 3 modèles conceptuel d'entrepôt de données qui sont :

- **Modèle en étoile**
- **Modèle en flocon de neige**
- **Modèle en constellation**

Pour notre solution, nous avons opté pour le modèle en flocon de neige qui permet de formaliser une hiérarchie de dimensions en fonction de la granularité de l'information qui sont reliées entre elles par un ensemble de relations jusqu'à arriver à la granularité la plus fine, ce model permet de s'approfondir dans les détails, ce qui facilite l'analyse et réduit le volume des données.

La figure suivante représente un exemple de modèle en flocon de neige.

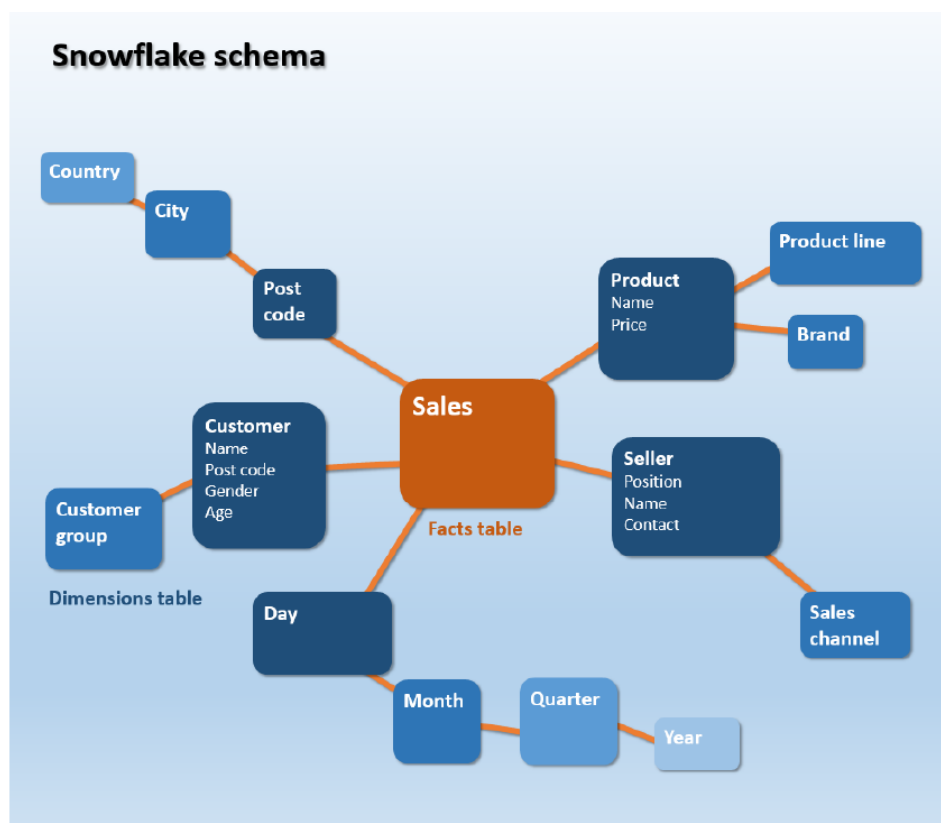


FIGURE 4.5 – Modèle en flocon de neige
[URL11]

Choix des dimensions

On entend par dimensions, tous les axes ou thèmes suivant lesquels les données seront analysées. On attribue une table à chaque dimension ce qui implique qu'il existe autant de tables de dimension que de dimensions.

Une table de dimension comporte des attributs et une clé primaire indépendante des attributs. Notre modèle comportera les dimensions suivantes :

- Dimension date.
- Dimension user.
- Dimension transactions.
- Dimension primary account.

Choix des mesures

Pour préparer la table de faits, il faudra bien se servir des éléments recueillis et collectés lors de la phase de préparation et étude des données sources. Il faudra alors préciser les éléments que nous souhaitons mesurer.

Dans notre table de faits, nous avons retenu les mesures suivantes :

- Chiffre d'affaires.
- Capitaux propres.

Table de faits

La table de fait contient les données observables "les faits" que l'on possède sur un sujet et que l'on veut étudier, selon divers axes d'analyse "les dimensions", notre table de fait, elle est constituée d'une table où sont spécifiées les clés des dimensions sous forme de clés étrangères et aussi des mesures.

Modélisation du Data Warehouse

Le schéma conceptuel final de l'entrepôt de données est le suivant :

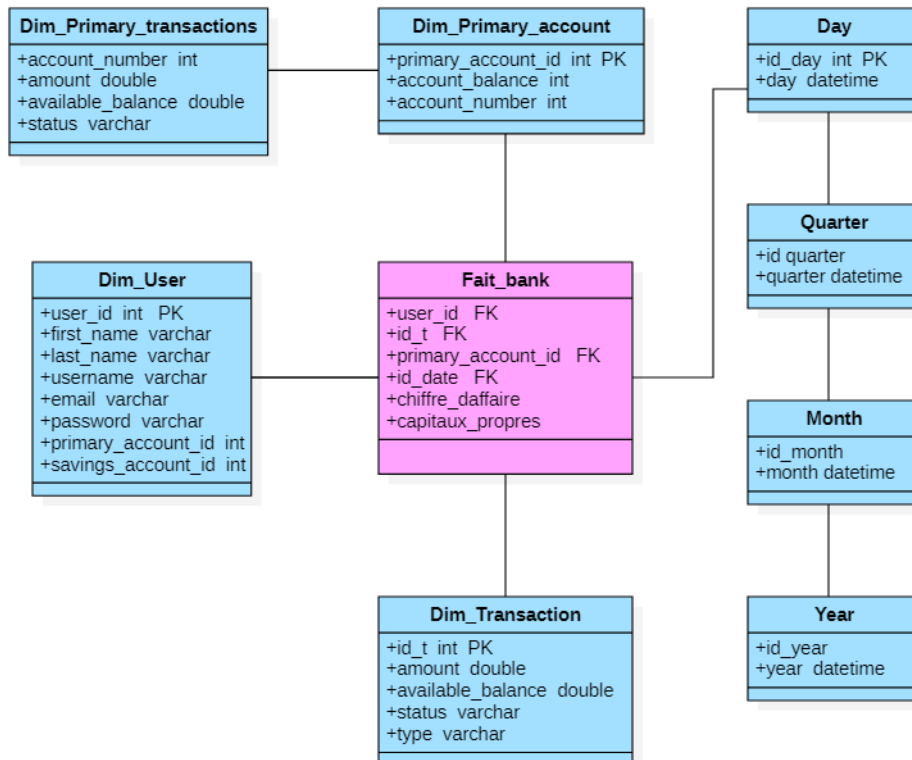


FIGURE 4.6 – Modèle conceptuel en flocon de neige du Data Warehouse

4.2.4 Diagrammes de séquences

Dans cette partie, nous présentons les interactions entre les différentes parties de notre application en utilisant les diagrammes de séquence dont l'objectif est d'obtenir une vision globale du comportement dynamique de notre système.

La figure suivante présente les interactions entre les différentes parties lors de l'authentification de l'utilisateur :

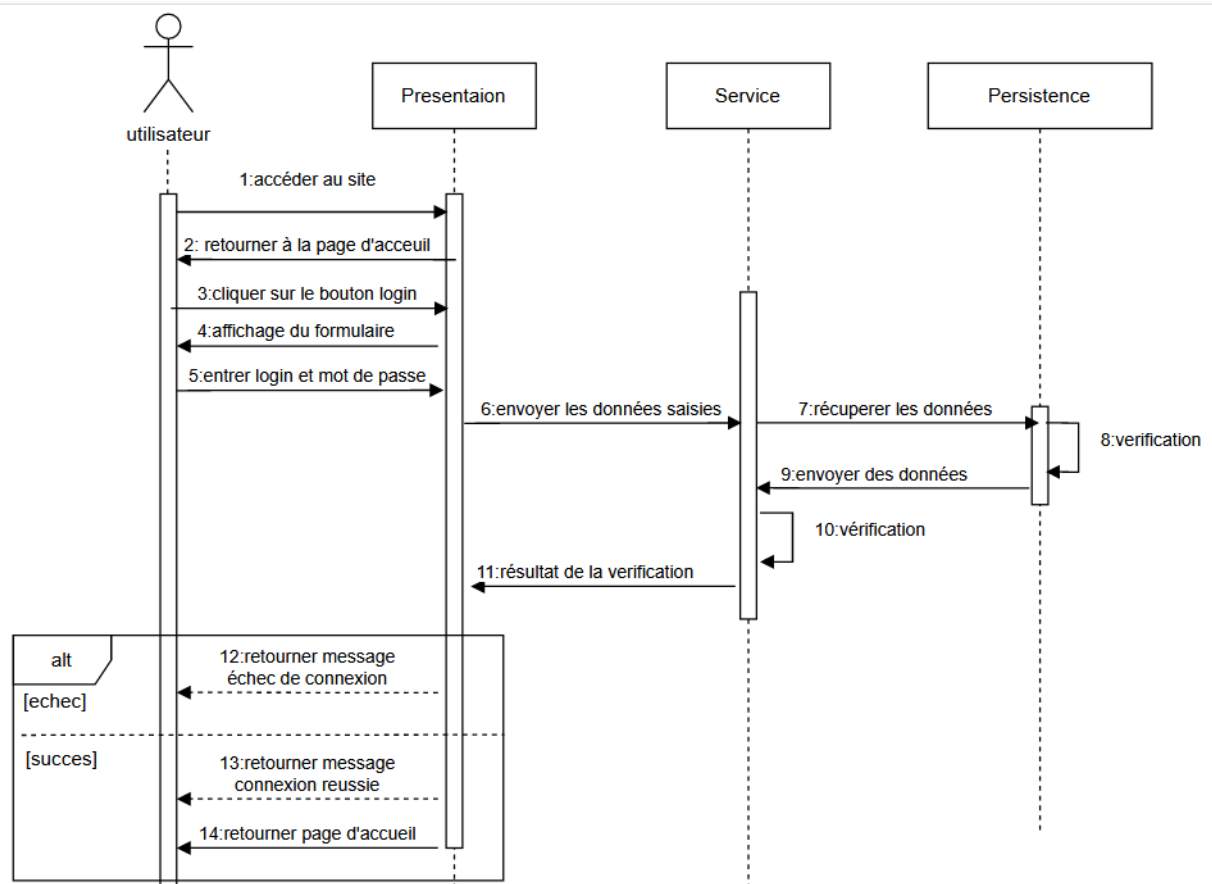


FIGURE 4.7 – Diagramme de séquence authentification

La figure suivante présente les interactions entre les différentes parties lors du dépôt d'argent de l'utilisateur :

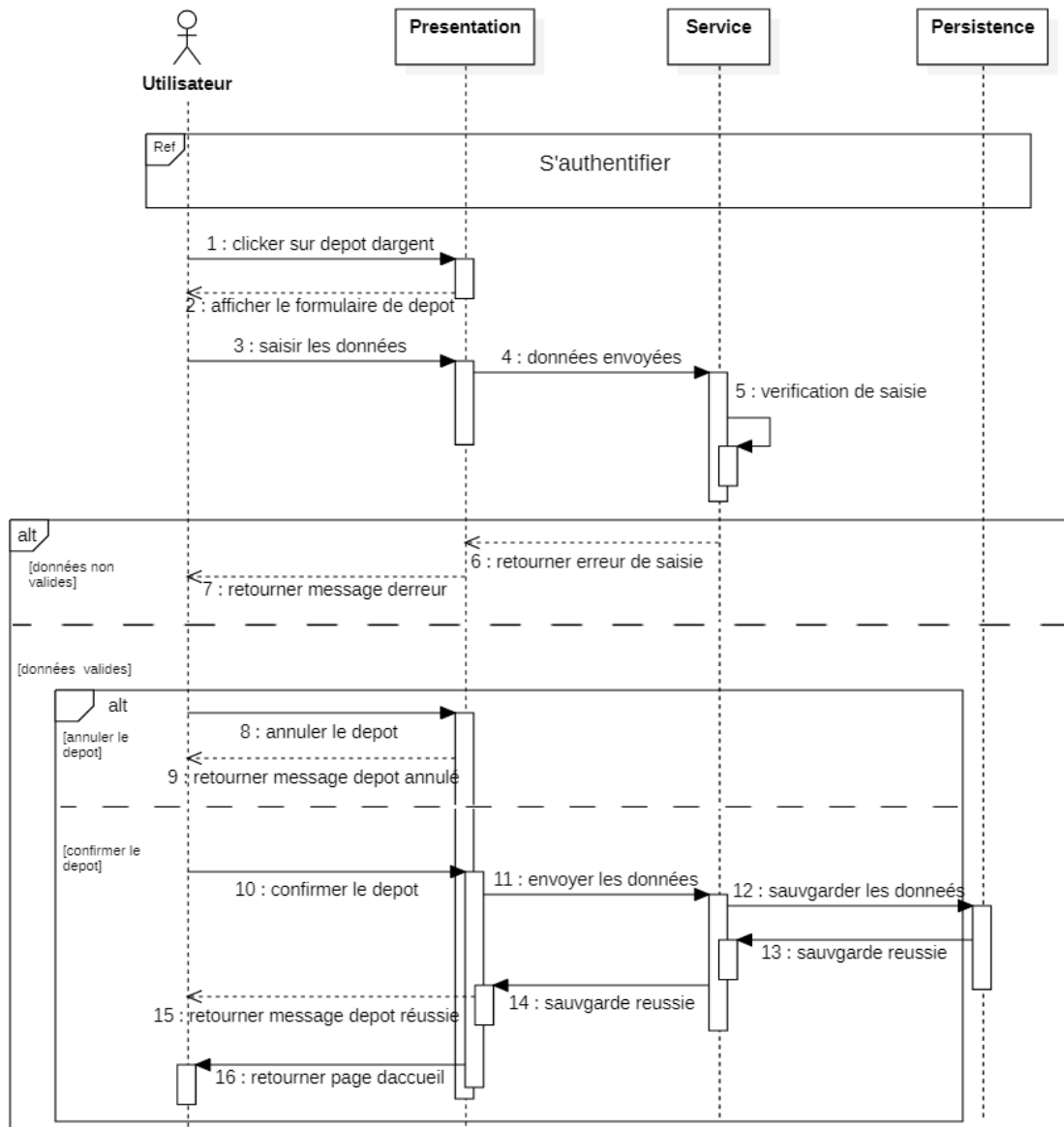


FIGURE 4.8 – Diagramme de séquence dépôt d'argent

Conclusion

Durant ce chapitre nous avons commencé par la conception globale en précisant l'architecture physique et logique, puis la conception détaillée en utilisant les différents diagrammes pour modéliser l'aspect statique et dynamique. la prochaine partie sera dédiée à la réalisation.

Chapitre 5

Réalisation

Ce dernier chapitre mettra fin à cet ouvrage, il représente l'implémentation et la réalisation du projet. Tout d'abord, nous présenterons les outils et technologies utilisées, puis nous allons présenter les principales interfaces graphiques de notre application.

5.1 Environnement de travail

Dans cette partie, nous détaillerons, les différentes caractéristiques qui constituent l'environnement spécifique de travail à savoir l'environnement logiciel et matériel.

5.1.1 Environnement matériel

De point de vue matériel, nous avons utilisé un ordinateur portable dont la configuration est présentée dans le tableau suivant

Marque	Asus
Processeur	Intel Core i7 2.6GHz
Mémoire	16 Go
Disque Dur	1 To
Système d'exploitation	Windows 10 Professionnel

FIGURE 5.1 – Environnement matériel

5.1.2 Environnement logiciel

Dans cette section, nous énumérons les différents outils et technologies que nous allons utiliser au cours de la réalisation du système.

outils logiciels

- **Microsoft Visual Studio 2018 Professional** : c'est un éditeur de code extensible développé par Microsoft.
- **Microsoft SQL Server** : qui est un système de gestion de base de données en langage SQL.
- **StarUml** : c'est un logiciel de modélisation UML.
- **Spring Tool suit (STS)** : est un IDE étendu pour Eclipse, il se spécialise dans le développement des applications JEE.
- **PowerBI** : est une solution d'analyse de données de Microsoft. Il permet de créer des visualisations de données avec une interface suffisamment simple.
- **Talend** : est un éditeur de logiciel spécialisé dans l'intégration de données.
- **JasperSoft studio** : est un générateur de rapports de développement open source.
- **Pentaho workbench** : c'est une interface graphique permettant de produire des schémas Mondrian pour les analyses de données OLAP (Online Analytical Processing).

Framework et Technologies

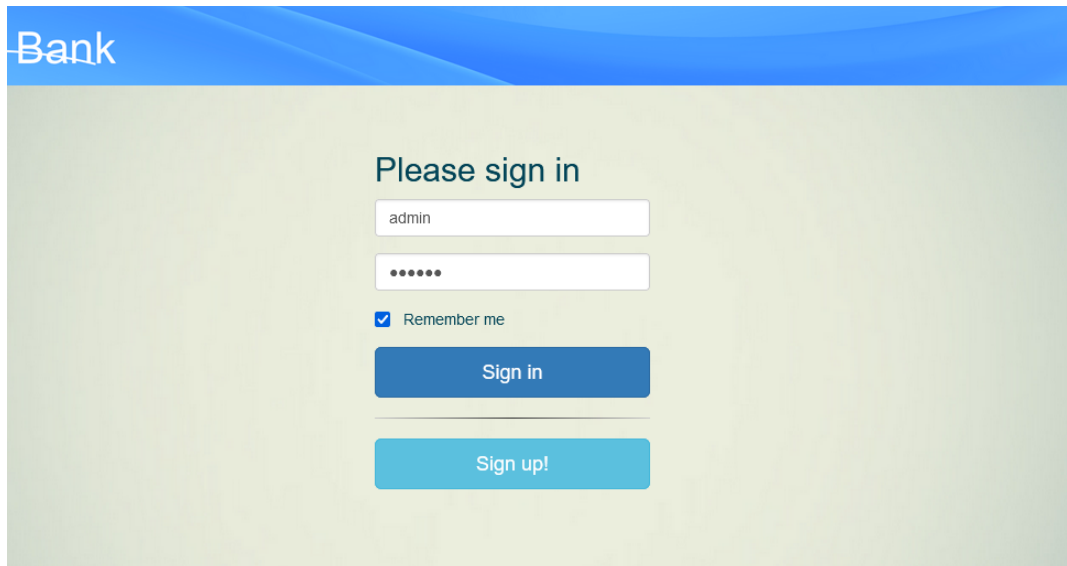
- **Springboot** : est un micro-Framework open source, il fournit aux développeurs Java une plate-forme pour démarrer avec une application Spring configurable automatiquement.
- **Angular** : est un Framework Javascript côté client qui permet de réaliser des applications de type "Single Page Application". Il est basé sur le concept de l'architecture MVC.

5.2 Travail réalisé

Dans cette section nous présentons le travail que nous avons réalisé en l'illustrant à l'aide de quelques captures d'écrans.

5.2.1 Transactions bancaires

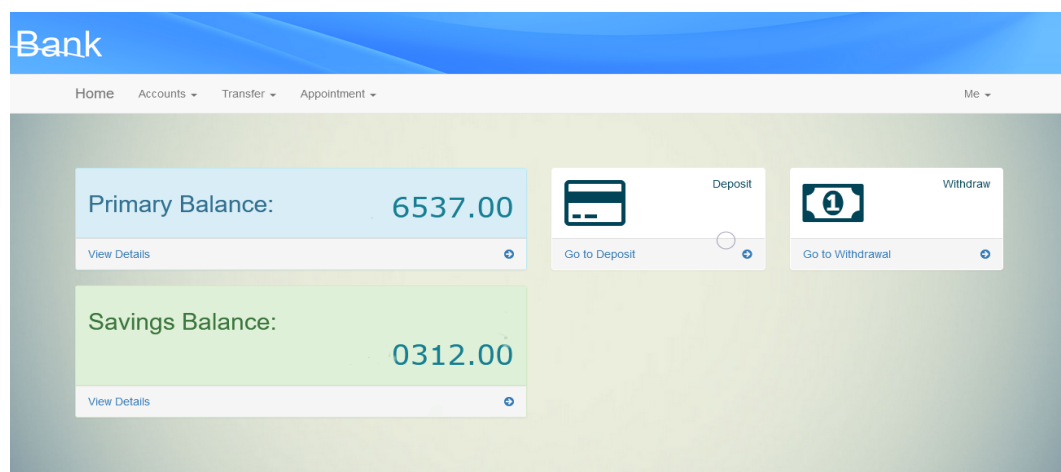
La figure suivante illustre la page de d'authentification. Ici, l'utilisateur saisit son login et mot de passe.



The image shows a web page for a bank's authentication system. At the top, there is a blue header with the word "Bank" in white. Below the header, the page has a light green background. In the center, there is a section titled "Please sign in" in bold. Below this title, there are two input fields: the first contains the text "admin", and the second contains a series of dots representing a password. Below the password field, there is a checkbox labeled "Remember me" which is checked. Below the checkbox, there are two buttons: a dark blue button labeled "Sign in" and a light blue button labeled "Sign up!".

FIGURE 5.2 – Authentification

La figure suivante montre la page d'accueil de notre application, qui permet le dépôt, le transfert d'argent aussi montre la somme d'argent disponible.



The image shows a web page for a bank's home interface. At the top, there is a blue header with the word "Bank" in white. Below the header, there is a navigation bar with links: "Home", "Accounts", "Transfer", "Appointment", and "Me". Below the navigation bar, the page has a light green background. On the left side, there are two balance cards: the first is titled "Primary Balance:" and shows the amount "6537.00" in large blue text, with a "View Details" link below it; the second is titled "Savings Balance:" and shows the amount "0312.00" in large blue text, with a "View Details" link below it. On the right side, there are two action cards: the first is titled "Deposit" and shows a credit card icon, with a "Go to Deposit" link below it; the second is titled "Withdraw" and shows a withdrawal icon, with a "Go to Withdrawal" link below it.

FIGURE 5.3 – Page d'accueil

Cette interface indique le compte et le montant déposé

The screenshot shows a web interface for making a deposit. At the top, there is a navigation bar with links: Home, Accounts (with a dropdown arrow), Transfer (with a dropdown arrow), and Appointment (with a dropdown arrow). Below the navigation bar, the main content area has a light green background. It contains two steps: 1. 'Please select the account you would like to deposit:' with a dropdown menu showing 'Primary'. 2. 'Please specify the amount you would like to deposit:' with a text input field containing '100'. Below these fields is a large blue button labeled 'Deposit'.

FIGURE 5.4 – Dépôt d'argent

La figure suivante montre l'historique des transactions avec leur détails.

The screenshot shows a web interface for viewing transaction history. At the top right, there is a summary box with 'Primary Balance: 6637.00'. Below this, there is a search bar and a 'Show 10 entries' dropdown. The main part of the interface is a table with the following columns: Post Date, Description, Type, Status, Amount, and Available Balance. The table contains 13 entries. At the bottom, there is a pagination bar showing 'Showing 1 to 10 of 13 entries' and navigation links for 'Previous', '1', '2', and 'Next'.

Post Date	Description	Type	Status	Amount	Available Balance
2021-05-28 12:13:28.0	Deposit to Primary Account	Account	Finished	133.0	1000.00
2021-05-28 12:13:52.0	Withdraw from Primary Account	Account	Finished	10.0	990.00
2021-05-28 12:15:44.0	Deposit to Primary Account	Account	Finished	170.0	6545.00
2021-05-28 12:15:59.0	Withdraw from Primary Account	Account	Finished	10.0	6535.00
2021-05-28 12:16:35.0	Between account transfer from Primary to Savings	Account	Finished	100.0	6435.00
2021-05-28 12:18:45.0	Transfer to recipient amir	Transfer	Finished	55.0	6446.00
2021-05-31 18:25:46.0	Deposit to Primary Account	Account	Finished	10.0	6456.00
2021-06-02 11:52:48.0	Deposit to Primary Account	Account	Finished	10.0	6466.00
2021-06-02 11:53:05.0	Withdraw from Primary Account	Account	Finished	67.0	6399.00
2021-06-02 11:53:22.0	Between account transfer from Primary to Savings	Account	Finished	8.0	6391.00

FIGURE 5.5 – Historique des transactions

5.2.2 Phases de l'analyse décisionnelle

dans cette partie, nous allons montrer les phases de l'implémentation de la partie décisionnelle.

L'implémentation ETL consiste à mettre en place ce qu'on appelle en nomenclature (instance de classification) de Talend des « jobs » permettant d'extraire les données à partir de la base sources, d'effectuer les transformations nécessaires et de les charger dans le Data Warehouse.

Extraction et transformation des données

Au cours de cette étape on a eu recours à l'extraction et à la transformation, l'extraction des données se fait par connexion avec la base de données avec Talend après, nous procédant à la régulation et la correction des données pour éviter les éventuels problèmes, parmi ces derniers on peut citer :

Le séparateur virgule qui engendrera beaucoup de problèmes au cours des prochaines étapes et pour cela nous avons remplacé les virgules par des points, aussi nous avons remarqué une erreur au niveau de la date, qui se présente sous différent format comme dd/mm/yyyy ou mm/dd/yyyy, donc nous avons standardisé ce format.

Après nous avons créé les dimensions d'analyse du dataware house, les figures ci-dessous représentent le travail réalisé d'extraction et de transformation de données avec le logiciel Talend.

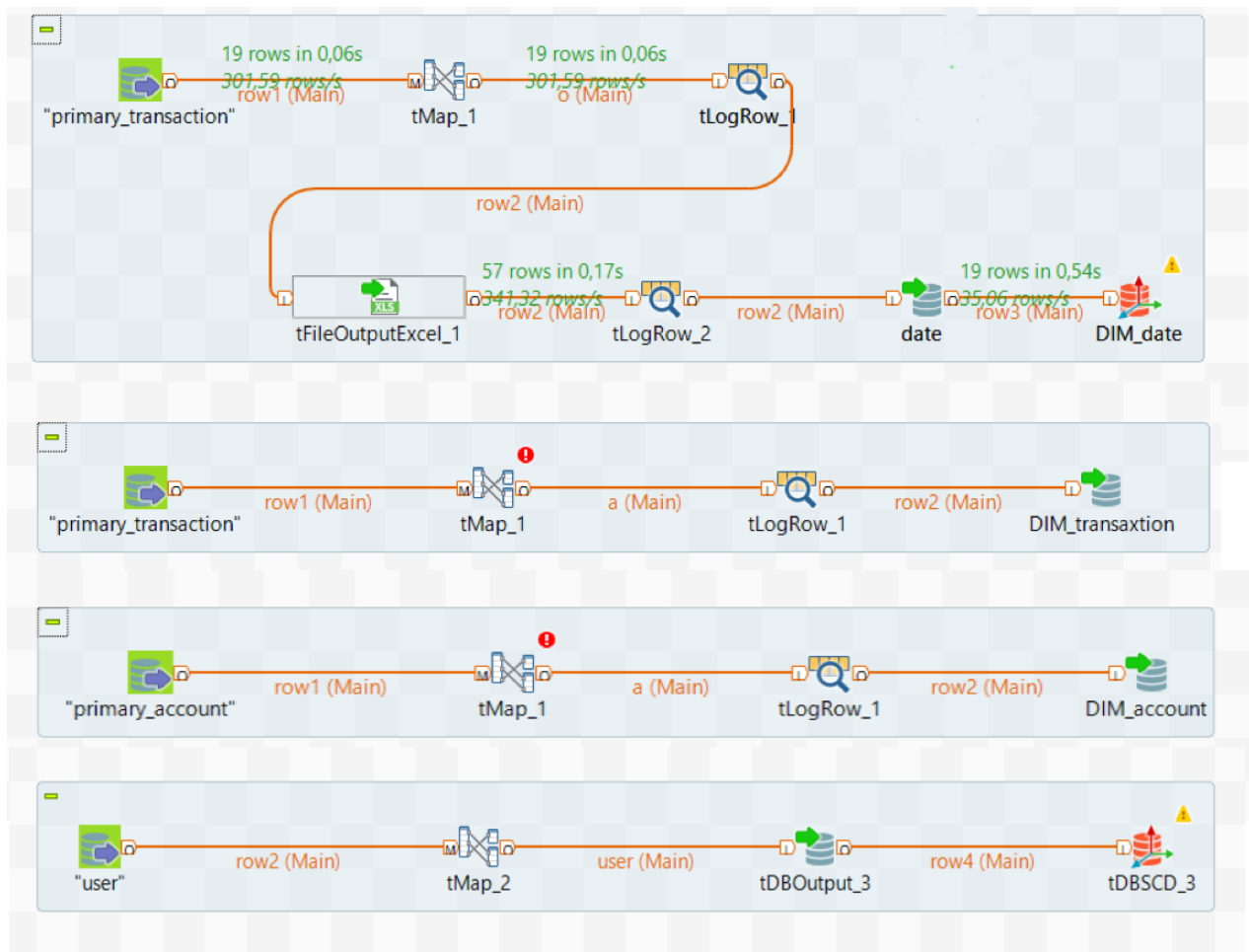


FIGURE 5.6 – Extraction et transformation de données

Chargement des données

Après toutes les extractions et transformations effectuées dans les étapes précédentes de l'ETL, les données seront chargées dans notre dataware house qui est installé sur un serveur de base de données, la figure suivante est le schéma relatif au job d'alimentation de la table de fait de notre entrepôt de données.

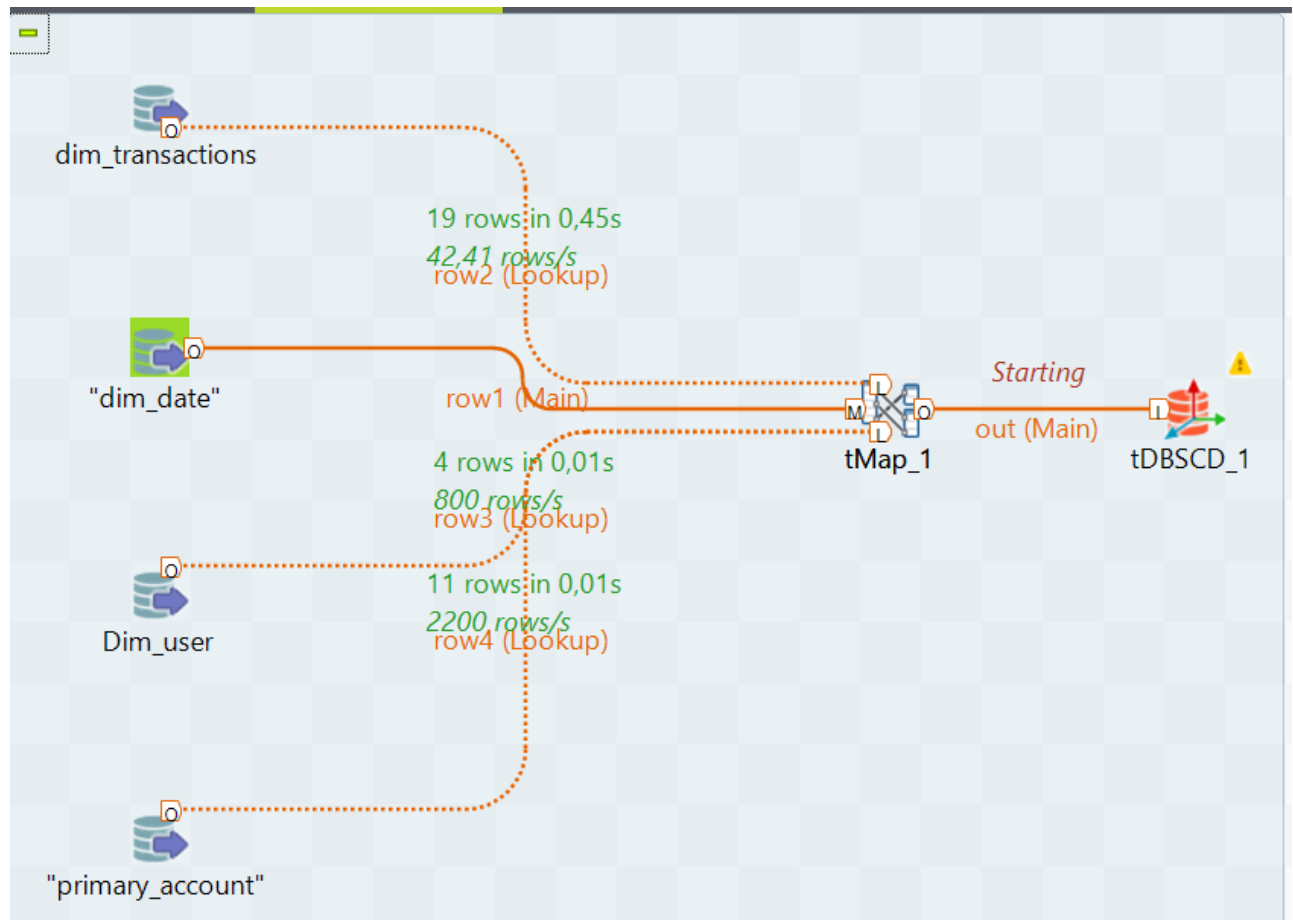


FIGURE 5.7 – Chargement des données

Construction du Cube OLAP

Le Cube OLAP est un tableau multidimensionnel de données qui offre la possibilité de calculs complexes d'analyse des tendances et de modélisation sophistiquée des données. Nous avons utilisé Pentaho workbench qui est une interface graphique permettant de produire des schémas Mondrian pour les analyses de données en multidimension, la figure suivante montre le cube réalisé avec ses différents dimensions et mesures.

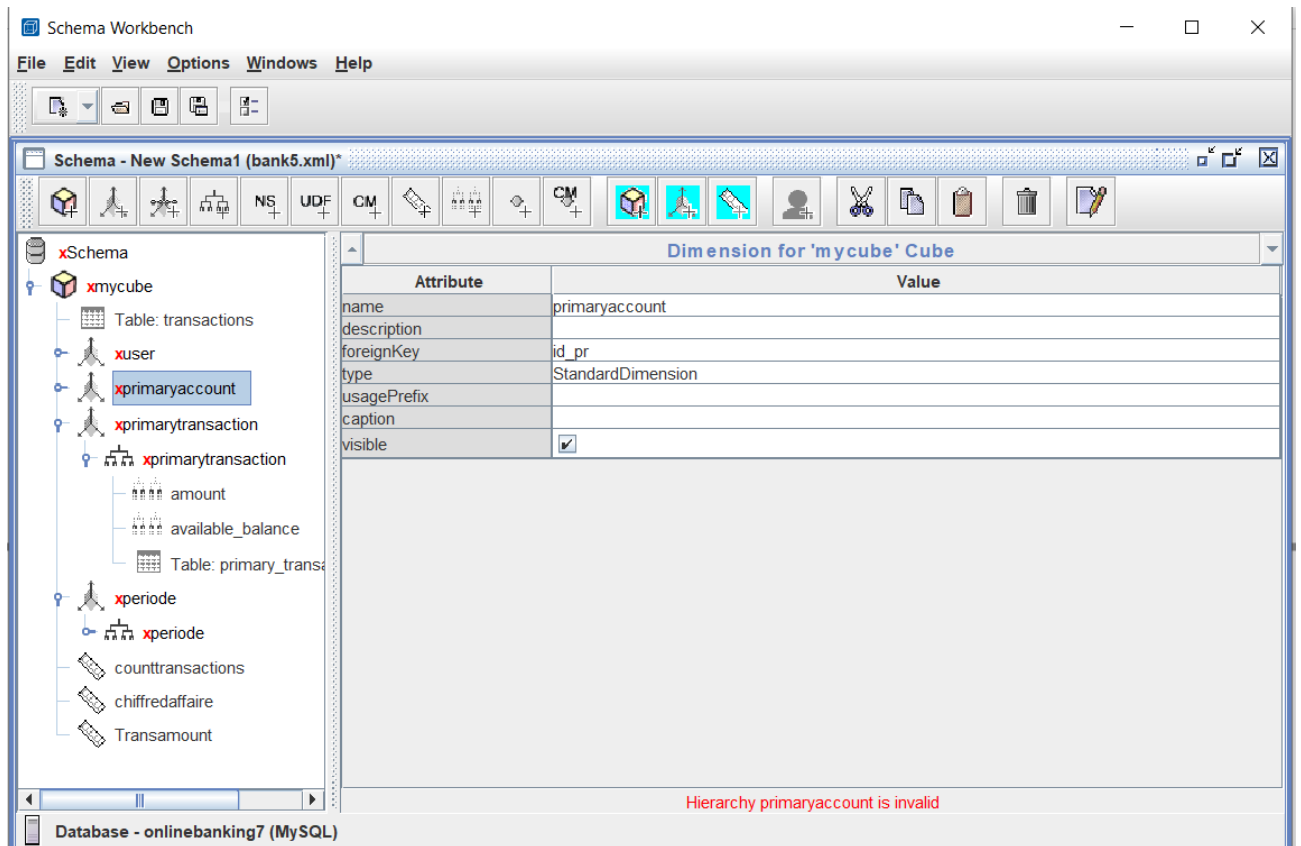


FIGURE 5.8 – Construction du Cube OLAP

Restitution des données

C'est une étape très importante dans notre projet car nous allons visualiser le résultat de notre travail effectué en illustrant des analyses et des visualisations. Ci-dessous un exemples de rapport et tableaux de bord que nous avons réalisés dans notre projet en utilisant Power BI et Jasper Soft studio.

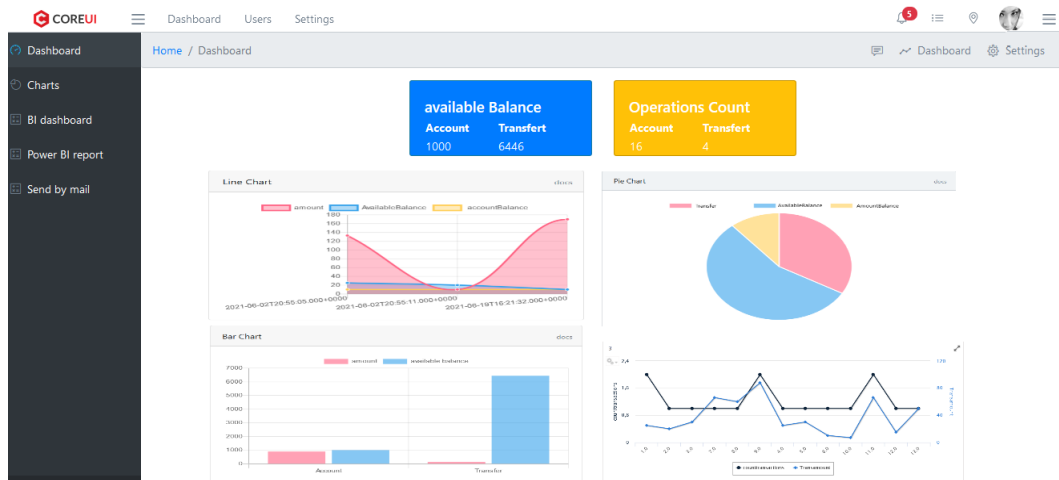


FIGURE 5.9 – Tableau de bord évolution des transactions en cours du temps

Nous présentons dans la figure suivante un aperçu du rapport d'analyse des transferts bancaires exporter sous extension PDF et en PNG et qui sont envoyées automatiquement par e-mail aux adresses des exploitants.

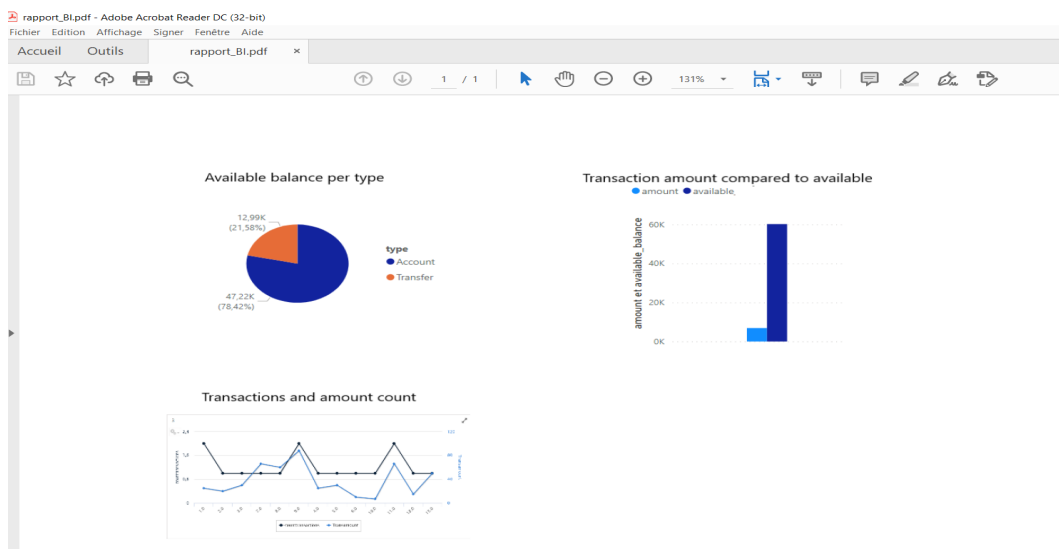


FIGURE 5.10 – Rapport d'analyse des transferts bancaires

5.2.3 Détection de fraude

Le problème à résoudre dans cette partie est de détecter les transactions frauduleuses à l'aide des modèles d'apprentissage automatique. Les transactions frauduleuses constituent une menace importante pour les banques et les clients.

Dans ce qui suit nous allons suivre des étapes ordonnées allant de l'exploration des données jusqu'à l'évaluation du modèle passant par des étapes primordiales.

1. Compréhension et exploration des données

Ici, nous devons charger les données et comprendre les fonctionnalités qui y sont présentes, les données sont chargées à partir de notre base de données de la banque grâce à la librairie Pandas de Python, la figure suivante nous informe sur les données traitées.

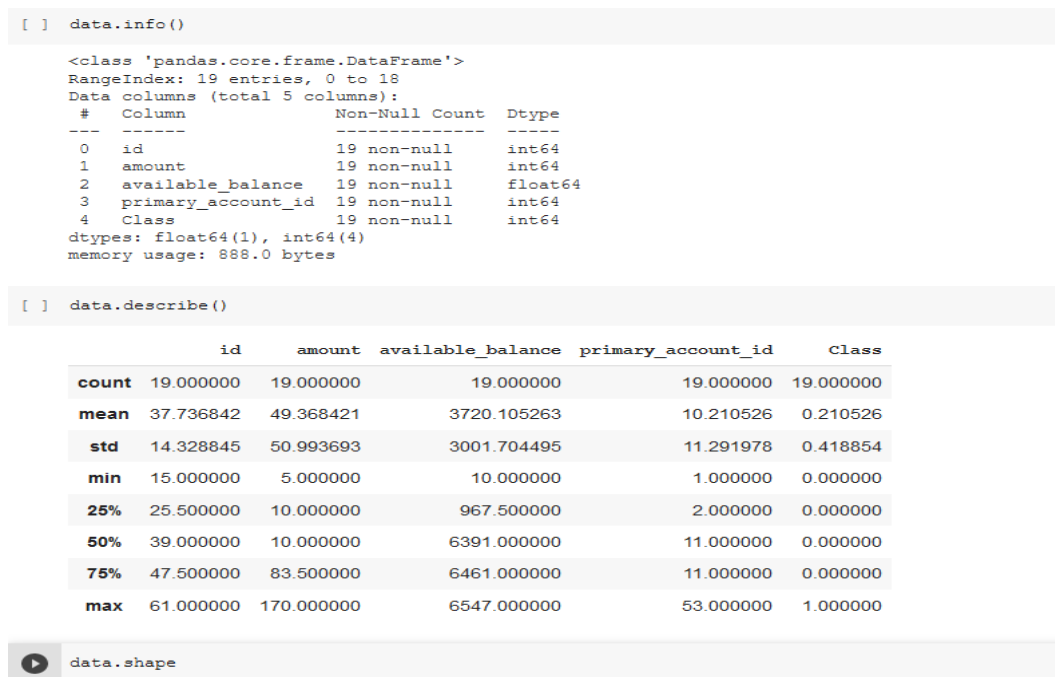


FIGURE 5.11 – Exploration des données

2. Nettoyage des Données

En science de données, les jeux de données comportent souvent des irrégularités et des erreurs. Ces dernières peuvent être des données manquantes ou aberrantes. Savoir traiter ces données permettra de produire un modèle précis et efficace. Les données que nous avons choisies d'utiliser sont peu susceptibles d'être parfaitement propres (sans erreurs), dans la figure suivante nous avons vérifié s'il existe des valeurs manquantes.

```

▶ credit_card_data.isnull().sum()

id          0
amount      0
available_balance  0
primary_account_id  0
Class       0
dtype: int64

▶ data.isnull().values.any()

False

```

FIGURE 5.12 – Vérification des valeurs manquantes

3. Analyse des Données

Dans cette partie on observe la distribution de nos classes et leur symétrie grâce aux barres graphique qui montre le nombre et le pourcentage des transactions frauduleuses vs non frauduleuses. D'après la figure suivante on constate que les classes sont très déséquilibré.



FIGURE 5.13 – Analyse des Données

4. Préparation les données pour la modélisation

Afin de faire face au déséquilibre des classes utilisées nous allons procéder à la séparation des transactions normal et frauduleuse puis l'échantillonnage et enfin la concaténation des données comme le montre les figures suivantes.

0 -> Normal Transaction

1 -> fraudulent transaction

```
[ ] # separating the data for analysis Splitting up the data
legit = credit_card_data[credit_card_data.Class == 0]
fraud = credit_card_data[credit_card_data.Class == 1]
```

```
[ ] print(legit.shape)
    print(fraud.shape)
```

```
[ ] # statistical measures of the data
legit.amount.describe()
```

```
count    15.000000
mean      56.400000
std       53.908388
min        5.000000
25%       10.000000
50%       55.000000
75%      100.000000
max      170.000000
Name: amount, dtype: float64
```

```
# compare the values for both transactions
credit_card_data.groupby('Class').mean()
```

	id	amount	available_balance	primary_account_id
Class				
0	30.60	56.4	3782.533333	10.666667
1	30.75	23.0	3486.000000	8.500000

```
data1= data.sample(frac = 0.1,random_state=1)
```

```
data1.shape
```

FIGURE 5.14 – Echantillonnage de données

Concatenating two DataFrames

```
[ ] new_dataset = pd.concat([legit, fraud], axis=0)
```

```
[ ] new_dataset.head()
```

```
id amount available_balance primary_account_id Class
0 15 133 1000.0 11 0
2 19 170 6545.0 11 0
4 23 100 6435.0 11 0
5 28 55 6446.0 11 0
6 33 10 6456.0 11 0
```

```
[ ] new_dataset.tail()
```

```
id amount available_balance primary_account_id Class
18 61 10 10.0 1 0
1 17 10 990.0 11 1
3 21 10 6535.0 11 1
8 37 67 6399.0 11 1
14 48 5 20.0 1 1
```

```
[ ] new_dataset['Class'].value_counts()
```

```
0 15
1 4
Name: Class, dtype: int64
```

```
[ ] new_dataset.groupby('Class').mean()
```

FIGURE 5.15 – Concaténation des données

5. Diviser les données

Diviser les données en train set et test set afin de vérifier les performances de nos modèles avec des données qui ne subissent pas l'entraînement, Ici, pour validation nous pouvons utiliser la méthode de validation croisée k-fold. Nous devons choisir une valeur k appropriée pour que la classe minoritaire soit correctement représentée dans les données de test.

Dans notre cas, Pour mesurer les performances du modèle, nous avons divisé les données en deux parties.

- 80 pourcent pour l'entraînement du model.
- 20 pourcent pour tester le model.

Split the data into Training data & Testing Data

```
[ ] X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=2)
```

FIGURE 5.16 – Division des données

6. Construction du model

C'est la dernière étape dans laquelle nous allons essayer différents algorithmes. Les types d'algorithmes que nous allons utiliser pour essayer de détecter des anomalies sur cet ensemble de données sont les suivants

Isolation Forest Algorithm

L'une des techniques pour détecter les anomalies s'appelle « Isolation Forest » forêts d'isolement qui identifie les anomalies en utilisant l'isolement. L'algorithme est basé sur le fait que les anomalies sont des points de données peu nombreux et différents. En raison de ces propriétés, les anomalies sont sensibles à un mécanisme appelé isolement.

L'utilisation de cette L'algorithme d'isolement est considérée comme l'un des moyen les plus efficace et efficient de détecter les anomalies, en plus il a une faible complexité temporelle aussi une faible exigence en mémoire.

Le Fonctionnement de L'algorithme de forêt d'isolement consiste à isoler les observations en sélectionnant au hasard une entité puis sélectionner aléatoirement une valeur de fractionnement entre les valeurs maximale et minimale de l'entité déjà sélectionnée.

Un score d'anomalie peut être calculé comme le nombre de conditions requises pour séparer une observation donnée.

La façon dont l'algorithme construit la séparation consiste à créer d'abord des arbres d'isolement ou arbres de décision aléatoires, ensuite le score est calculé en fonction de la longueur du chemin pour isoler l'observation.

La figure suivante nous informe sur l'isolement de l'anomalie en utilisant « Isolation Forest »

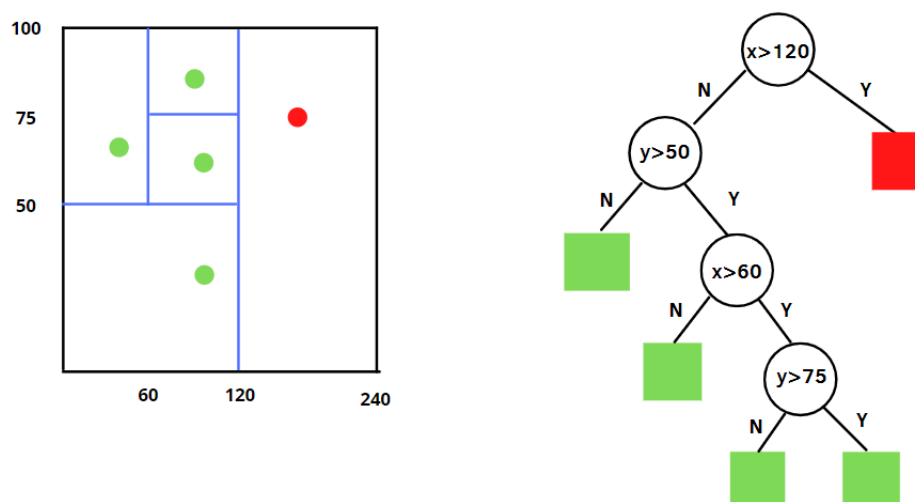


FIGURE 5.17 – Exemple de l'isolement d'anomalie avec algorithme Isolation Forest

Local Outlier Factor(LOF) Algorithm

Cet algorithme identifie les points aberrants spatiaux dans les entités ponctuelles en calculant le facteur LOF de chaque entité. Les points aberrants spatiaux sont des entités dans des emplacements anormalement isolés et le facteur LOF est une mesure qui décrit l'isolement d'un emplacement par rapport à ses voisins locaux. Une valeur LOF plus élevée indique un isolement plus important.

L'outil peut également servir à produire une surface de prévision raster permettant d'estimer si de nouvelles entités seront classées comme points aberrants en tant compte de la distribution spatiale des données.

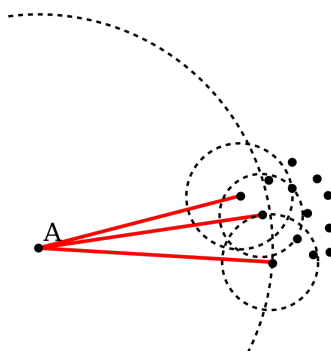


FIGURE 5.18 – Exemple d'isolement d'un emplacement par rapport à ses voisins avec LOF

L'idée de base du LOF : est de comparer la densité locale d'un point avec les densités de ses voisins. Le point A a une densité beaucoup plus faible que ses voisins.

```
classifiers = {  
    "Isolation Forest": IsolationForest(n_estimators=100, max_samples=len(X),  
                                       contamination=outlier_fraction, random_state=state, verbose=0),  
    "Local Outlier Factor": LocalOutlierFactor(n_neighbors=20, algorithm='auto',  
                                              leaf_size=30, metric='minkowski',  
                                              p=2, metric_params=None, contamination=outlier_fraction)  
}
```

FIGURE 5.19 – Les algorithmes choisis

La figure suivante montre la formation du modèle avec les entrées données grâce au `model.fit()` et la génère des prédictions de sortie pour les échantillons en entrée par l'utilisation du `model.predict()` de la bibliothèque libre Python Sklearn.

```
[ ] model = IsolationForest()  
  
[ ] model.fit(X_train, Y_train)  
  
IsolationForest(behaviour='deprecated', bootstrap=False, contamination='auto',  
                max_features=1.0, max_samples='auto', n_estimators=100,  
                n_jobs=None, random_state=None, verbose=0, warm_start=False)  
  
[ ] # accuracy on training data  
X_train_prediction = model.predict(X_train)  
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

FIGURE 5.20 – Formation du modèle

7. Evaluation de Model

La figure suivante montre l'évaluation des modèles entraînés et des modèles prédit du training data et du test data, l'évaluation est sous forme de score d'exactitude.

Accuracy Score

```
[ ] # accuracy on training data
X_train_prediction = model.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)

[ ] print('Accuracy on Training data : ', training_data_accuracy)

Accuracy on Training data : 0.9415501905972046

[ ] # accuracy on test data
X_test_prediction = model.predict(X_test)
test_data_accuracy = accuracy_score(X_test_prediction, Y_test)

[ ] print('Accuracy score on Test Data : ', test_data_accuracy)

Accuracy score on Test Data : 0.9390862944162437
```

FIGURE 5.21 – Évaluation des modèles utilisées

La figure suivante montre l'évaluation de l'algorithme LOF et Isolation Forest utilisées et les métriques pour mesurer les performances du modèle

```
Isolation Forest: 73
Accuracy Score :
0.9974368877497279
Classification Report :
              precision    recall  f1-score   support

         0           1.00      1.00      1.00      28432
         1           0.26      0.27      0.26         49

    micro avg           1.00      1.00      1.00      28481
    macro avg           0.63      0.63      0.63      28481
   weighted avg           1.00      1.00      1.00      28481

Local Outlier Factor: 97
Accuracy Score :
0.9965942207085425
Classification Report :
              precision    recall  f1-score   support

         0           1.00      1.00      1.00      28432
         1           0.02      0.02      0.02         49

    micro avg           1.00      1.00      1.00      28481
    macro avg           0.51      0.51      0.51      28481
   weighted avg           1.00      1.00      1.00      28481
```

FIGURE 5.22 – Resultat de l'évaluation des modèles utilisées

les termes de performances de modèle sont les suivant

- FN et TP sont des transactions frauduleuses.
- FP et TN sont des transactions légitimes.
- TP et FP sont des transactions identifiées frauduleuses par le model.

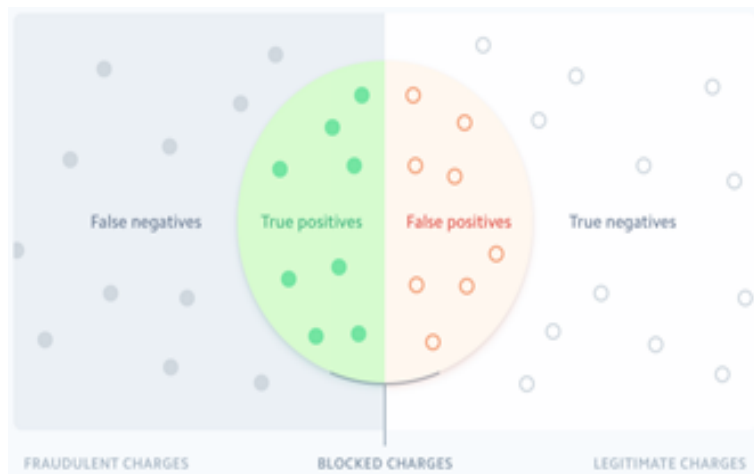


FIGURE 5.23 – Les termes de performances du modèle
[URL12]

Les métriques pour mesurer les performances sont les suivants :

- **Précision** : fraction des transactions que le modèle classe comme fraude qui sont réellement frauduleuses.
Précision = $TP / (TP + FP)$.
- **Recall** : Fraction de toutes les fraudes identifiées par notre modèle.
Recall = $TP / (TP + FN)$
- **F-Measure** : Une bonne mesure à utiliser lorsque nous recherchons un équilibre entre précision et rappel et qu'il existe une répartition inégale des classes.
F-Measure = $2 * Précision * Recall / (Précision + Recall)$

Observations

Nous devons évaluer les modèles à l'aide de mesures d'évaluation appropriées

- Isolation Forest a détecté 73 erreurs par rapport au facteur local aberrant LOF en détectant 97.

- Isolation Forest a 99.74 pourcent de précision cependant LOF of 99.65 pourcent.
- Lors de la comparaison entre la précision et le recall pour les 2 modèles Isolation Forest est mieux que LOF parce que comme on peut le constater qu'il a détecté plus de cas de fraude.
- Nous pouvons également améliorer cette précision en augmentant la taille de l'échantillon ou en utilisant des algorithmes d'apprentissage profond "deep learning".

Conclusion

Dans ce dernier chapitre nous avons montré les différentes technologies et outils avec lesquels ce projet a été réalisé, puis nous avons présenté les interfaces de notre application ainsi que l'implémentation de nos algorithmes de data mining. La partie suivante présentera la conclusion générale et les perspectives.

Conclusion et perspectives

Pour faire face au gros volume de données du secteur bancaire qui est dispersé, non structuré, et hétérogène et afin de transformer ces données en informations utiles et pertinentes, nous avons adopté l'informatique décisionnelle.

Durant ce travail, nous avons commencé par concevoir et mettre en place une application bancaire capable de gérer les activités quotidiennes, puis nous avons collecté les données à partir des systèmes de production source et nous l'avons adaptée à l'usage décisionnel, après nous avons emmagasiné les données structurées et traitées, ensuite nous avons organisé les données sous forme de schémas de données pour faciliter les analyses multidimensionnelles et pour finir nous avons restitué les données pour visualiser le résultat de notre travail sous forme de tableaux de bord et de rapports qui offrent une vision globale et détaillée sur l'activité de la banque et aide à la prise de décision. Nous avons aussi suivi le processus de découverte des connaissances qui utilise la fouille de données pour découvrir les transactions frauduleuses qui constituent une véritable menace pour la banque.

Ce projet est un noyau sur lequel peuvent s'étendre plusieurs perspectives pour l'enrichir et l'améliorer. En effet, le projet que nous avons réalisé est évolutif et modulaire. Il est donc possible de l'enrichir. Nous avons remarqué que nous ne pouvons pas déterminer les habitudes des clients. Alors nous envisageons également, comme perspective, d'utiliser l'intelligence artificielle pour déterminer les habitudes de la clientèle afin de garantir une analyse basée sur le marché et pour mesurer les risques financiers.