

RAPPORT DE STAGE

En vue de l'obtention du diplôme de Licence en Business Computing
Parcours Business Intelligence

Soutenu le : ... **2024**

Réalisé par : **Malek Fendri**



« Mise en place d'un système décisionnel pour la Gestion optimisée
des Flux Financiers avec Interaction via Chatbot et Déploiement sur
Application Web »

06/02/2024 – 07/06/2024

Maître de stage : Mr Adnen Gaied

Encadrante académique : Mme Ibtihel Rebhi

Année universitaire 2023/2024

	<i>Nom & Prénom</i>	<i>Date et Signature</i>
Maître de stage	<i>Mr Adnen Gaied</i>	
Encadrante académique	Mme Ibtihel Rebhi	



Résumé :

Dans un contexte où la transformation numérique est devenue essentielle pour la compétitivité des entreprises, ce projet vise à développer une solution intégrée de Business Intelligence (BI) pour la gestion des flux financiers au sein des Assurances Biat. En mettant en place un entrepôt de données, des tableaux de bord interactifs avec Power BI, un chatbot intelligent nommé FlowBot et une application web interactive, ce projet offre une plateforme unifiée pour l'analyse des données et l'assistance utilisateur. L'entrepôt de données est conçu selon un modèle en étoile, facilitant l'organisation et l'accès aux données. Les tableaux de bord fournissent des insights précieux sur les opérations financières, tandis que FlowBot améliore l'expérience utilisateur en répondant aux questions en temps réel. L'application web, nommée Arima Insight, centralise toutes ces fonctionnalités, assurant une gestion sécurisée et flexible des données. Ce projet démontre l'importance de l'informatique décisionnelle pour la prise de décision éclairée et l'optimisation des opérations commerciales.

Mots-clés :

Informatique décisionnelle, Système décisionnel, OLAP, OLTP, ETL, Tableaux de bord, Gestion des données, flux , encaissements , décaissements , analyse , aide à la décision , traitement du langage naturel , chatbot , intelligence artificielle , application web , FlowBot , Arima Insight , gestion profils , agile , sprint , scrum.

Abstract :

In an era where digital transformation is crucial for business competitiveness, this project aims to develop an integrated Business Intelligence (BI) solution for managing cash flows at Assurances Biat. By implementing a data warehouse, interactive dashboards with Power BI, an intelligent chatbot named FlowBot, and an interactive web application, this project provides a unified platform for data analysis and user assistance. The data warehouse is designed with a star schema, facilitating the organization and access to data. The dashboards offer valuable insights into financial operations, while FlowBot enhances user experience by responding to queries in real time. The web application, named Arima Insight, centralizes all these features, ensuring secure and flexible data management. This project highlights the importance of BI in making informed decisions and optimizing business operations.

Key words :

Business Intelligence, Decision Support System, OLAP, OLTP, ETL, Dashboards, Data Management, flows, inflows, outflows, analysis, decision support, Natural Language Processing, chatbot, Artificial Intelligence, web application , FlowBot, Arima Insight, profile management , sprint , scrum .

Dédicaces

À mes chers parents,

À **Mon cher père défunt**, Ton absence est une douleur constante, mais chaque jour je m'efforce de vivre de manière à te rendre fier. Ton souvenir m'inspire à donner le meilleur de moi-même dans tout ce que j'entreprends. J'espère que, de là où tu es, tu vois mes efforts et mon dévouement.

À **Ma chère mère**, Ta force et ton amour inconditionnel ont été ma boussole. Tu as été mon père, ma mère, et ma meilleure amie, toujours présente dans les moments difficiles comme dans les moments de joie. Ton soutien indéfectible m'a permis de surmonter chaque obstacle, et je te suis infiniment reconnaissante.

À mes chers frères Mehdi et Melek,

À **Mehdi**, tu as pris sur toi une grande responsabilité et tu as été un pilier de stabilité pour notre famille. Ta présence et ton soutien constants ont été essentiels pour moi. Merci d'être le frère exemplaire sur qui je peux toujours compter.

À **Melek**, même si la distance nous sépare, ton amour et ton soutien traversent les kilomètres. Ta présence, même lointaine, m'a toujours apporté réconfort et courage. Merci d'être là pour moi malgré la distance.

À mes chers amis,

À **Aziz, à Souha, à Miss, et à mes très proches amis**, je tiens à exprimer toute ma gratitude envers vous pour votre soutien indéfectible et votre encouragement constant. Merci pour votre amitié précieuse et votre soutien inestimable. Cette réalisation est autant la vôtre que la mienne.

À **Youssef, à Wissem**, merci d'avoir été toujours là à mes côtés et d'avoir rendu mes années d'université les meilleures. Merci pour tout, pour chaque rire, chaque encouragement, et chaque instant partagé. Vous avez fait de cette période une expérience inoubliable.

À ma famille,

À **Azza**, ma chère belle-sœur, ta présence et ton énergie ont été une source constante de réconfort et ton amour inconditionnel ont enrichi notre famille. Merci d'être la sœur que je n'ai pas eue.

À **Ma famille**, merci pour votre amour, votre soutien et votre présence lors des moments les plus importants. Vous êtes ma force et ma motivation.

Malek Fendri

Remerciements

Je tiens tout d'abord à exprimer ma gratitude envers **Dieu**, pour m'avoir donné la force et la détermination nécessaires pour mener à bien ce projet.

Je souhaite remercier tout particulièrement **Madame Ibtihel Rebhi**, mon encadrante académique, pour ses conseils avisés, son soutien inestimable, et sa patience tout au long de cette aventure académique.

Mes sincères remerciements vont également à **Monsieur Adnen Gaied**, mon maître de stage, pour son encadrement et ses précieux conseils, qui ont été d'une aide inestimable dans la réalisation de ce projet.

Un grand merci à Monsieur **Aladin Nefzi**, pour ses conseils et son suivi constant, qui ont grandement contribué à la réussite de cette expérience.

Je tiens à exprimer ma reconnaissance envers tous **mes enseignants de l'ESPRIT School of Business et de l'Université Claude Bernard** pour leur enseignement de qualité et leur soutien académique.

Enfin, je remercie chaleureusement toute **l'équipe des assurances BIAT** pour m'avoir accueilli au sein de leur organisation, pour leur disponibilité et leur aide précieuse.

Merci à vous tous

Table des matières

1	Contexte général du projet	3
1.1	Organisme d'accueil	3
1.1.1	Historique et réalisations	3
1.1.2	Organigramme de l'organisme d'accueil	4
1.1.3	Secteur d'activités	5
1.1.4	Services	5
1.2	Cadre général	6
1.2.1	Étude de l'existant	6
1.2.2	Solution proposée	7
1.3	Méthodologie de gestion du projet	7
1.3.1	Méthodologies et étude comparative	8
1.3.2	Choix de la méthodologie	9
1.3.3	Méthodes de travail	9
1.3.4	Méthode choisie	12
2	Planification du projet et outils techniques	14
2.1	Spécifications et analyse des besoins	14
2.1.1	Besoins fonctionnels	15
2.1.2	Besoins non fonctionnels	15
2.2	Planification	16
2.2.1	Backlog produit	16
2.2.2	Diagramme de GANTT	18
2.3	Méthodes techniques	18
2.3.1	Les systèmes décisionnels	19
2.3.2	L'intelligence artificielle et ses fondamentaux	21
2.3.3	Le développement web	23
2.4	Environnement technique	25
2.4.1	Outils de stockage des données	25
2.4.2	Outils pour ETL	26
2.4.3	Outils de restitution	26
2.4.4	Outils de développement de chatbot	27
2.4.5	Outils de développement de l'application web	28

3	Préparation des données et modélisation du Data Warehouse	30
3.1	Sprint Backlog	31
3.2	Compréhension et exploration des données	31
3.2.1	Compréhension des données	32
3.3	Génération des données	33
3.4	Chargement des données sources	34
3.5	La modélisation du Data Warehouse	37
3.5.1	Les différents modèles conceptuels	38
3.5.2	Choix du sujet d'analyse	39
3.5.3	Choix des dimensions	40
3.5.4	Choix des mesures	40
3.5.5	Notre modèle conceptuel	41
3.6	Intégration de l'entrepôt de données	41
3.6.1	Chargement des dimensions	41
3.6.2	Chargement de la table de fait	47
3.6.3	Automatisation du processus ETL	48
4	Restitution des données	50
4.1	Sprint Backlog	50
4.2	Intégration des données	51
4.3	Création des mesures	51
4.3.1	Mesure du montant net	51
4.3.2	Variation des montants nets	52
4.3.3	Regroupement des tranches d'âges	52
4.4	Interprétation des tableaux de bord	53
4.4.1	Tableau de bord global	53
4.4.2	Tableau de bord des encaissements	55
4.4.3	Tableau de bord des décaissements	57
4.4.4	Tableau de bord des clients	59
5	Développement d'un Chatbot et Déploiement d'une Application Web	61
5.1	Sprint Backlog	61
5.2	Le processus de la création du Chatbot	62
5.2.1	Présentation de FlowBot	62
5.2.2	Fonctionnalités de FlowBot	63
5.3	Le processus du développement de l'application Web	67
5.3.1	Présentation de l'application Web Arima Insight	67
5.3.2	Fonctionnalités d'Arima Insight	68
	Conclusion	84
	Bibliographie	86

Table des figures

1.1	Logo des Assurances BIAT	4
1.2	Organigramme des Assurances BIAT	4
1.3	Méthode SCRUM	9
2.1	Diagramme de GANTT	18
2.2	Architecture du système décisionnel	19
2.3	Architecture d'un chatbot basé sur RASA	22
2.4	Logo de l'outil pgadmin	25
2.5	Logo de l'outil MySQL	25
2.6	Logo de l'outil Talend Open Source	26
2.7	Logo de l'outil Microsoft SQL Server	26
2.8	Logo de l'outil Power BI	27
2.9	Logo de l'outil Tableau	27
2.10	Logo de l'environnement Visual Studio	28
2.11	Logo de l'outil RASA	28
2.12	Logo de l'outil Star UML	28
2.13	Logo du framework Flask	29
3.1	Données sources	32
3.2	Données sources (Suite)	32
3.3	Données sources (Suite)	32
3.4	Données sources (Suite)	32
3.5	Page de connexion PostgresSQL	34
3.6	Création de notre base de données sur postgres	34
3.7	Importation des fichiers sources	35
3.8	Fenêtre d'établissement de la connexion	35
3.9	Exemple JOB "STG_2021"	36
3.10	L'icône Tmap	36
3.11	L'icône tUniqRow	36
3.12	L'icône tLogRow	37
3.13	Base de données Postgres	37
3.14	Structure de la table de fait	37
3.15	Structure de la table de dimension	38
3.16	Modèle en étoile	38

3.17	Modèle en flocon de neige	39
3.18	Modèle en constellation	39
3.19	Modèle conceptuel de notre entrepôt de données	41
3.20	Job "DIM_PRODUIT"	42
3.21	Fenêtre de tUniqRow "DIM_PRODUIT"	42
3.22	Élaboration d'une séquence	42
3.23	Expression remplacement des données manquantes pour la "DIM_PRODUIT"	42
3.24	Job "DIM_Client"	43
3.25	Expression âge client de la dimension "DIM_Client"	43
3.26	Job "DIM_NATURE"	43
3.27	Expression de normalisation "DIM_NATURE"	44
3.28	Job "DIM_BRANCHE"	44
3.29	Job "DIM_DATE"	45
3.30	Expression année "DIM_DATE"	45
3.31	Expression mois "DIM_DATE"	45
3.32	Expression jour "DIM_DATE"	45
3.33	Job "DIM_SENS"	46
3.34	Job "DIM_RESEAUX"	46
3.35	job "FAIT_FLUX"	47
3.36	tMap "FAIT_FLUX"	47
3.37	Expression de conversion montant "FAIT_FLUX"	47
3.38	Job d'exécution processus ETL	48
3.39	Jour et heure de l'automatisation	49
4.1	Établissement de la connexion entre POWER BI et l'entrepôt de données	51
4.2	Expression de la mesure Montant Net	51
4.3	Expression de la mesure Montant net précédent	52
4.4	Expression de la mesure variation des montants net	52
4.5	Tableau de bord de vue global	53
4.6	Tableau de bord des encaissements	55
4.7	Tableau de bord des encaissements	56
4.8	Tableau de bord des décaissements	57
4.9	Tableau de bord des décaissements	58
4.10	Tableau de bord des clients	59
5.1	Logo FlowBot	62
5.2	Diagramme de séquence "Assistance technique"	63
5.3	RASA shell "Assistance technique"	64
5.4	Composante NLU "Assistance technique"	64
5.5	Composante Stories "Assistance technique"	65
5.6	Diagramme de séquence "Répondre aux questions"	65
5.7	RASA shell "Répondre aux questions"	66
5.8	Composante NLU "Répondre aux questions"	66

5.9	Composante Stories "Répondre aux questions"	67
5.10	Logo "ARIMA INSIGHT"	67
5.11	Diagramme de cas d'utilisation "ARIMA INSIGHT"	68
5.12	Base de données "database_pfe".	69
5.13	Diagramme de séquence "ARIMA INSIGHT : Authentification"	69
5.14	Page d'authentification "ARIMA INSIGHT"	70
5.15	Diagramme de séquence "ARIMA INSIGHT : Réinitialisation"	71
5.16	Processus de récupération et de réinitialisation "ARIMA INSIGHT"	73
5.17	Diagramme de séquence "ARIMA INSIGHT : Accès aux dashboards"	74
5.18	Page dashboard sur "ARIMA INSIGHT"	75
5.19	Diagramme de séquence "ARIMA INSIGHT : Chatbot"	75
5.20	Fenêtre de pop up de "FlowBot" sur "ARIMA INSIGHT"	77
5.21	Diagramme de séquence "ARIMA INSIGHT : Modifier profil"	77
5.22	Page profil "ARIMA INSIGHT"	79
5.23	Diagramme de séquence "ARIMA INSIGHT : Ajout utilisateur"	79
5.24	Page ajout utilisateur "ARIMA INSIGHT "	81
5.25	Diagramme de séquence "ARIMA INSIGHT : Suppression profil"	81
5.26	Page suppression utilisateur "ARIMA INSIGHT "	83

Liste des tableaux

1.1	Comparaison entre l'approche traditionnelle et l'approche agile	8
1.2	Rôles dans la méthode Scrum	10
1.3	Comparaison entre la méthode Scrum et la méthode Kanban	12
1.4	Équipe de SCRUM	13
2.1	Backlog produit	16
2.2	Tableau comparatif entre les systèmes décisionnels et systèmes transactionnels	20
3.1	Sprint Backlog : Compréhension, génération et chargement des données	31
3.2	Description des colonnes de la table source	33
4.1	Sprint Backlog : Restitution des données	50
5.1	Sprint Backlog : Développement du Chatbot et Application Web	61
5.2	Description Cas d'Utilisation Assistance Technique	64
5.3	Description Cas d'Utilisation "Poser une question"	66
5.4	Description diagramme de séquence "Arima Insight : Authentification"	70
5.5	Description diagramme de séquence : "Arima Insight : Réinitialisation"	72
5.6	Description diagramme de séquence "ARIMA INSIGHT : Accès aux dashboards"	74
5.7	Description diagramme de séquence "ARIMA INSIGHT : Utilisation de FlowBot"	76
5.8	Description diagramme de séquence "ARIMA INSIGHT : Modification de profil"	78
5.9	Description diagramme de séquence "ARIMA INSIGHT : Ajouter un utilisateur"	80
5.10	Description diagramme de séquence "ARIMA INSIGHT : Suppression de profil"	82

Liste des abréviations

- **BI** : Business Intelligence.
- **Dim** : Dimension.
- **DWH** : Data Warehouse ou entrepôt de données.
- **ETL** : Extract-Transform-Load.
- **NLP** : Natural Language Processing.
- **NLU** : Natural Language Understanding.
- **OLAP** : Online Analytical Processing.
- **OLTP** : Online Transactional Processing.
- **ODS** : Operational Data Storage.
- **SI** : Système d'information.

Introduction générale

Dans l'ère actuelle de la transformation numérique, la capacité à collecter, analyser et exploiter les données est devenue un atout stratégique incontournable pour les entreprises. L'informatique décisionnelle joue un rôle central en fournissant des outils et des méthodes pour transformer des données brutes en informations précieuses, permettant ainsi aux organisations de prendre des décisions éclairées et de maintenir un avantage concurrentiel. La quantité de données générées quotidiennement ne cesse de croître, et la capacité à les analyser de manière efficace est essentielle pour répondre aux défis complexes du monde des affaires.

Le projet que nous avons réalisé au sein de l'organisme d'accueil "Assurances Biat" vise à répondre à ces besoins critiques en matière de BI. Notre objectif était de développer une solution intégrée et innovante qui permettrait une gestion efficace des flux financiers, tout en offrant des outils avancés d'analyse et d'assistance utilisateur. Ce projet se distingue par la création d'un système décisionnel complet, le développement d'un chatbot intelligent et la mise en place d'une application web interactive, constituant ainsi une plateforme unifiée pour les utilisateurs.

Dans ce rapport, nous allons explorer les différentes phases et composants de notre projet de manière détaillée.

Nous commencerons par la maîtrise des données, où nous analyserons la structure des données et la préparation nécessaire pour établir un entrepôt de données efficace. Nous décrirons le processus de compréhension et d'exploration des données, y compris la génération de données synthétiques pour alimenter notre entrepôt de données.

Ensuite, nous passerons à la conception et la modélisation du data warehouse, en présentant les principaux modèles conceptuels et les étapes d'élaboration des processus ETL (Extraction, Transformation et Chargement). Nous expliquerons notre choix de modèle et détaillerons la construction de notre entrepôt de données.

Dans le chapitre suivant, nous nous concentrerons sur la restitution des données. Nous décrirons l'intégration de Power BI avec notre entrepôt de données et la création de tableaux de bord interactifs. Ces tableaux de bord offrent une visualisation claire et efficace des données, permettant aux décideurs de prendre des décisions informées.

Nous aborderons ensuite le développement d'un chatbot. Nous présenterons FlowBot, notre assistant intelligent conçu pour répondre aux questions des utilisateurs et fournir une assistance technique en temps réel. Nous détaillerons les fonctionnalités de FlowBot et son intégration avec notre système.

Enfin, nous discuterons du déploiement de l'application web Arima Insight. Nous montrerons comment toutes les composantes de notre solution sont centralisées sur cette plateforme unique, offrant aux utili-

sateurs un accès intégré aux tableaux de bord, au chatbot et aux fonctionnalités de gestion des profils.

À travers ces chapitres, nous mettrons en évidence notre méthodologie de travail, les défis rencontrés et les solutions mises en place pour garantir le succès de notre projet. Notre approche rigoureuse et l'utilisation de technologies avancées assurent que notre solution est prête à répondre aux exigences actuelles et futures de l'informatique décisionnelle.

En conclusion, ce projet démontre l'importance et l'impact de la BI dans le monde moderne. En permettant aux entreprises d'exploiter pleinement leurs données, notre solution contribue à améliorer la prise de décision et à optimiser les opérations commerciales. Notre travail au sein des Assurances Biat a mis en évidence le potentiel des outils BI pour transformer les données en informations exploitables, et nous sommes fiers de la solution que nous avons développée.

Chapitre 1: Contexte général du projet

Contents

1.1	Organisme d'accueil	3
1.1.1	Historique et réalisations	3
1.1.2	Organigramme de l'organisme d'accueil	4
1.1.3	Secteur d'activités	5
1.1.4	Services	5
1.2	Cadre général	6
1.2.1	Étude de l'existant	6
1.2.2	Solution proposée	7
1.3	Méthodologie de gestion du projet	7
1.3.1	Méthodologies et étude comparative	8
1.3.2	Choix de la méthodologie	9
1.3.3	Méthodes de travail	9
1.3.4	Méthode choisie	12

Introduction

Dans ce chapitre, nous présenterons notre organisme d'accueil. Nous aborderons le cadre général du projet visant à optimiser la gestion des flux financiers de l'entreprise. Nous analyserons la problématique et les solutions proposées.

1.1 Organisme d'accueil

Dans cette section, nous présenterons BIAT Assurances, son historique, ses réalisations, ainsi que les services qu'elle offre à ses clients.

1.1.1 Historique et réalisations

Fondée en 2002, **BIAT Assurances** est une filiale de la Banque Internationale Arabe de Tunisie (**BIAT**), la première banque de Tunisie. Depuis sa création, BIAT Assurances a su s'imposer comme un acteur clé sur le marché tunisien de l'assurance grâce à une combinaison d'innovation constante et d'un engagement ferme envers la satisfaction client.

En moins de deux décennies, l'entreprise a développé une gamme complète de produits d'assurance adaptés aux besoins spécifiques du marché tunisien, qu'il s'agisse de particuliers, de professionnels ou d'entreprises. L'innovation a toujours été au cœur de la stratégie de BIAT Assurances.

De plus, BIAT Assurances a établi un réseau de distribution étendu à travers tout le pays, garantissant une accessibilité optimale à ses services. Cette proximité avec les clients, combinée à des solutions d'assurance personnalisées et à une gestion efficace des réclamations, a permis à BIAT Assurances de se démarquer de la concurrence.



Figure 1.1 – Logo des Assurances BIAT

1.1.2 Organigramme de l'organisme d'accueil

L'organigramme des Assurances BIAT illustre la structure organisationnelle de l'entreprise, montrant les différentes divisions et leur interaction. Vous trouverez ci-dessous l'organigramme de l'entreprise.

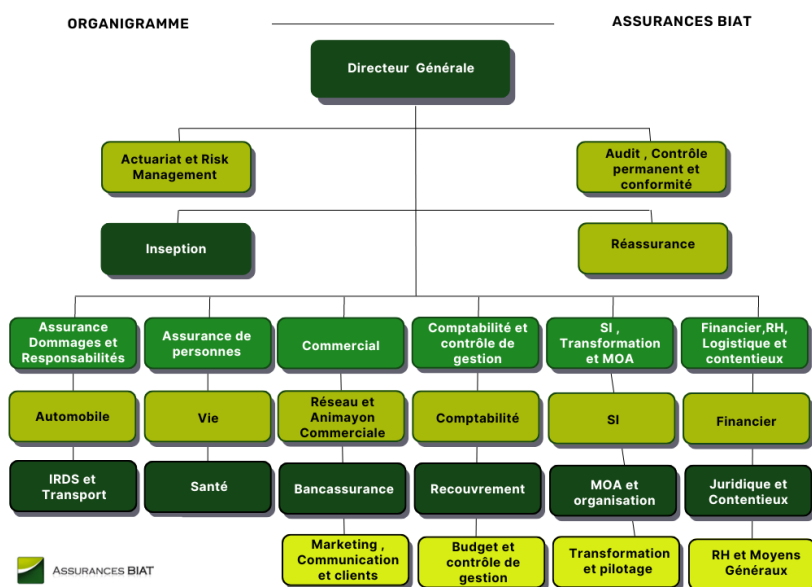


Figure 1.2 – Organigramme des Assurances BIAT

1.1.3 Secteur d'activités

Assurances BIAT opère dans le secteur des assurances, offrant une gamme complète de produits couvrant toutes les branches : assurance vie, assurance prévoyance, assurance santé et assurance de biens. Elle distribue ses produits via le réseau BIAT, ses 22 agences, et une vingtaine de courtiers partenaires.

1.1.4 Services

Les principaux produits et services proposés par Assurances BIAT incluent :

Particuliers

- **Assurance Auto** : Protection complète pour les véhicules individuels, couvrant les dommages, le vol, et la responsabilité civile.
- **Multirisque Habitation** : Protection des biens immobiliers et mobiliers contre les incendies, les catastrophes naturelles, et le vol.
- **Assurance Accidents** : Couverture en cas d'accidents corporels, offrant une protection financière et des services de soins.
- **Assurance en Couverture de Prêts** : Sécurisation des prêts en garantissant le remboursement en cas de décès, d'invalidité ou de perte d'emploi.
- **Prévoyance Individuelle** : Solutions de protection pour l'avenir, incluant des garanties en cas d'invalidité ou de décès.
- **Épargne de Demain** : Plans d'épargne dédiés aux études, projets futurs et retraite, assurant une sécurité financière à long terme.

Professionnels

- **Accidents Corporels** : Protection des employés contre les accidents sur le lieu de travail, avec des indemnités en cas de blessure ou d'invalidité.
- **Assurance Auto** : Couverture des véhicules professionnels contre les dommages, le vol, et la responsabilité civile.
- **Multirisque Professionnels** : Protection globale des activités professionnelles, incluant les locaux, les équipements, et les stocks.

Entreprises

- **Groupe Santé** : Couverture médicale collective pour les employés, offrant des soins de santé complets et un accès à des services médicaux de qualité.
- **Groupe Retraite** : Plans de retraite collectifs pour les employés, assurant une sécurité financière après la fin de carrière.
- **Flotte Auto** : Couverture des flottes de véhicules d'entreprise, incluant les dommages, le vol, et la responsabilité civile.
- **Multirisque Entreprises** : Protection globale des activités et des biens de l'entreprise contre les risques divers, incluant les incendies, les catastrophes naturelles, et les vols.

1.2 Cadre général

Nous avons entrepris une étude approfondie de l'existant afin d'identifier ses lacunes et ses points forts. Cette analyse critique nous a permis de dégager les besoins fonctionnels et non fonctionnels essentiels pour l'élaboration de la solution proposée.

1.2.1 Étude de l'existant

Systèmes existants

Chez Assurances BIAT, le système existant pour la gestion des encaissements et des décaissements repose sur une application où chaque transaction est enregistrée en détail. Ce système fonctionne comme un système d'informations, consignait les transactions financières sans fournir les outils nécessaires pour une prise de décision éclairée.

Critique de l'existant

Le système actuel chez Assurances BIAT présente plusieurs limitations majeures qui réduisent son efficacité en tant qu'outil décisionnel. Il manque de réactivité, empêchant une prise de décision rapide et éclairée en raison de l'absence de contextes analytiques et de visualisations pertinentes.

La recherche d'informations spécifiques est chronophage en raison de la volumétrie des données et du manque de fonctionnalités analytiques avancées.

De plus, le système ne permet pas une analyse approfondie des données par branche ou produit, limitant ainsi l'accès à des informations détaillées et pertinentes pour une prise de décision éclairée. En conséquence, beaucoup de temps est perdu à rechercher des informations spécifiques, retardant ainsi les processus critiques et la prise de décision.

1.2.2 Solution proposée

Pour remédier aux limitations du système actuel chez Assurances BIAT, notre solution proposée est un système décisionnel incluant un Data Warehouse (DWH), un chatbot intelligent et une application web. Nous avons déployé le chatbot et les tableaux de bord interactifs et dynamiques dans l'application web, pour faciliter l'analyse approfondie des données.

Cette solution vise à réduire le temps consacré à la recherche d'informations spécifiques, à offrir des contextes analytiques pertinents et à faciliter une prise de décision rapide et éclairée.

1.3 Méthodologie de gestion du projet

Une méthodologie de gestion de projet est un ensemble de principes, de pratiques et de procédures utilisées pour gérer et exécuter les projets de manière structurée et efficace. Elle fournit un cadre de référence permettant d'organiser le travail, de définir les étapes à suivre, et d'assurer une gestion cohérente et prévisible des tâches. Le choix de la méthodologie de gestion de projet est crucial pour le bon déroulement de notre travail.

Il existe plusieurs méthodologies de gestion de projet, mais nous avons opté pour étudier les deux plus populaires et largement utilisées : la méthodologie traditionnelle et la méthodologie Agile. Dans cette section, nous examinerons et comparerons ces méthodologies afin de sélectionner celle qui convient le mieux à notre projet, en tenant compte de ses spécificités et de ses exigences.

1.3.1 Méthodologies et étude comparative

Nous avons réalisé une étude comparative entre les méthodologies traditionnelle et Agile, comme le présente le tableau ci-dessous.

Table 1.1 – Comparaison entre l'approche traditionnelle et l'approche agile

Thème	Approche traditionnelle	Approche agile
Cycle de vie	En cascade ou en V, sans rétroaction possible, phases séquentielles.	Itératif et incrémental.
Planification	Prédictive, caractérisée par des plans plus ou moins détaillés et d'exigences définis au début du projet.	Adaptative avec plusieurs niveaux de planification avec ajustements si nécessaire en fonction des changements survenus.
Documentation	Produite en quantité importante comme support de communication, de validation et contractualisation.	Réduite au profit d'incréments fonctionnels opérationnels pour obtenir le livrable attendu par le client.
Équipe	Une équipe avec des ressources spécialisées, dirigées par un chef de projet.	Une équipe où l'initiative et la décision sont prises de manière collective, soutenue par le chef de projet.
Qualité	Contrôle qualité à la fin du cycle de développement. Le client découvre le produit fini.	Un contrôle qualité précoce et permanent, au fur et à mesure du processus. Le client est impliqué tout au long du cycle.
Changement	Résistance voire opposition au changement. Processus lourds de gestion des changements acceptés.	Accueil favorable au changement avec simplification des démarches de changements au cours de la réalisation.
Suivi de l'avancement	Mesure de la conformité aux plans initiaux. Analyse des écarts.	Un seul indicateur d'avancement : le nombre de fonctionnalités implémentées et le travail restant à faire.
Gestion des risques	Processus distinct et rigoureux de gestion des risques.	Intégration de la gestion des risques dans les démarches courantes et auto-assurance des solutions face aux risques.
Mesure du succès	Respect des engagements initiaux en termes de coûts, de budget et de niveau de qualité.	Satisfaction client par la livraison de valeur ajoutée.

1.3.2 Choix de la méthodologie

Après avoir étudié nos besoins et les exigences spécifiques de notre projet, nous avons décidé d'opter pour **l'approche Agile**. Cette décision est motivée par la nécessité d'une grande flexibilité, d'une adaptation rapide aux changements, et d'une collaboration étroite avec les parties prenantes tout au long du cycle de vie du projet. L'approche Agile nous permet de livrer des fonctionnalités de manière itérative et incrémentale, garantissant ainsi une meilleure réactivité et une amélioration continue du produit final.

1.3.3 Méthodes de travail

Une méthode est un ensemble structuré de pratiques, de techniques et de procédures spécifiques utilisées pour accomplir une tâche ou un projet de manière systématique et efficace. En gestion de projet, une méthode détaille les étapes et les outils nécessaires pour appliquer une approche de gestion de projet. Parmi les méthodes Agile les plus utilisées, on trouve Scrum et Kanban. Ces méthodes sont largement adoptées pour leur capacité à améliorer la flexibilité, la réactivité et la collaboration au sein des équipes de projet.

Méthode scrum

Scrum est une méthode Agile de gestion de projet utilisée principalement dans le développement de logiciels, bien qu'elle soit également applicable à d'autres domaines. Elle se concentre sur la livraison rapide et continue de petites incréments de travail, permettant une grande flexibilité et réactivité face aux changements. La figure ci-joint représente le déroulement de la méthode SCRUM :

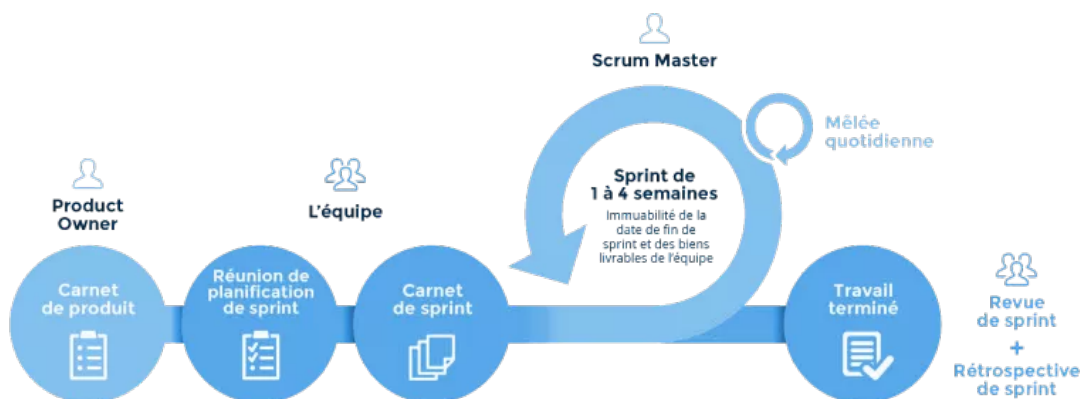


Figure 1.3 – Méthode SCRUM

1. Rôles

Table 1.2 – Rôles dans la méthode Scrum

Rôle	Description
Product Owner	Il gère le backlog du produit, en s'assurant que les éléments sont clairement définis, priorisés et compréhensibles pour l'équipe. Le Product Owner est le lien principal entre les parties prenantes et l'équipe de développement.
Scrum Master	Facilitateur pour l'équipe Scrum. Il aide à comprendre et à adopter Scrum, assure que l'équipe respecte les pratiques et valeurs de Scrum, et élimine les obstacles qui pourraient entraver la progression de l'équipe.
Équipe de Développement	Composée de professionnels qui réalisent le travail de livraison d'un incrément de produit potentiellement livrable à la fin de chaque sprint.

2. Évènements

- **Sprint** : Période fixe de 2 à 4 semaines durant laquelle un incrément de produit fini et utilisable est créé.
- **Planification du Sprint (Sprint Planning)** : Réunion pour définir l'objectif du sprint et sélectionner les éléments du Product Backlog à accomplir.
- **Scrum Quotidien (Daily Scrum)** : Réunion quotidienne de 15 minutes pour synchroniser les activités et créer un plan pour les prochaines 24 heures.
- **Revue de Sprint (Sprint Review)** : Réunion pour inspecter l'incrément et adapter le Product Backlog si nécessaire.
- **Rétrospective de Sprint (Sprint Retrospective)** : Réunion pour réfléchir sur le sprint écoulé et identifier les améliorations potentielles pour les prochains sprints.

3. Artefacts

- **Product Backlog** Liste ordonnée de tout ce qui pourrait être nécessaire dans le produit.
- **Sprint Backlog** : Ensemble des éléments du Product Backlog sélectionnés pour le sprint, ainsi qu'un plan pour délivrer l'incrément de produit et réaliser l'objectif du sprint.
- **Increment** : Somme de tous les éléments du Product Backlog complétés durant un sprint et ceux des sprints précédents, en état de fonctionner et répondant aux critères de définition de fini de l'équipe.

Méthode Kanban

Kanban est une méthode Agile de gestion de projet et de travail en flux tiré, qui se concentre sur l'optimisation du flux de travail et l'amélioration continue. Elle utilise un système visuel pour gérer les tâches et les projets .

1. Éléments du Tableau Kanban

- **Colonnes** : Représentent les différentes étapes du processus de travail (par exemple, À faire, En cours, En révision, Terminé).
- **Cartes** : Représentent les tâches ou les éléments de travail individuels. Chaque carte contient des informations sur la tâche, telles que la description, le responsable, et les dates d'échéance.
- **Limites de WIP** : Indiquent le nombre maximum de tâches pouvant être en cours à chaque étape. Elles aident à prévenir la surcharge de travail et à identifier les goulots d'étranglement.
- **Swimlanes** : Lignes horizontales sur le tableau qui permettent de catégoriser ou de classer les tâches par type, priorité, ou autre critère pertinent.

Étude comparative entre les méthodes

Nous avons réalisé une étude comparative entre les méthodes Scrum et Kanban afin de mettre en évidence leurs principales différences et similitudes.

Le tableau ci-dessous présente cette comparaison de manière détaillée :

Table 1.3 – Comparaison entre la méthode Scrum et la méthode Kanban

Aspect	Scrum	Kanban
Cadre de travail	Cadre de travail structuré avec des rôles, des événements et des artefacts bien définis.	Système de gestion du travail flexible, sans rôles, événements ou artefacts prédéfinis.
Rôles	Product Owner, Scrum Master, Équipe de développement.	Pas de rôles spécifiques, mais peut inclure des responsables de processus.
Cycles de travail	Sprints itératifs de durée fixe (2 à 4 semaines).	Flux continu, pas de cycles de travail fixes.
Planification	Planification du sprint au début de chaque cycle.	Planification continue au fur et à mesure que de nouvelles tâches sont ajoutées au tableau Kanban.
Visualisation	Tableau des tâches du sprint avec colonnes pour les différentes étapes du processus de développement.	Tableau Kanban avec colonnes pour chaque étape du processus de travail et limites de WIP.
Responsabilité	L'équipe est responsable de la livraison de l'incrément à la fin de chaque sprint.	L'équipe est responsable de la gestion du flux de travail et de l'optimisation continue.
Boucles de rétroaction	Réunions régulières : Daily Scrum, Sprint Review, Sprint Retrospective.	Boucles de rétroaction continues : réunions de service, rétrospectives, analyse des flux de travail.
Livraison	Livraison à la fin de chaque sprint (2 à 4 semaines).	Livraison continue au fur et à mesure que les tâches sont complétées.
Amélioration continue	Principalement lors des rétrospectives de sprint.	Amélioration continue intégrée dans le flux de travail quotidien.

1.3.4 Méthode choisie

Pour répondre aux besoins de notre projet, nous avons choisi la méthode Scrum. Cette approche Agile nous permet de gérer efficacement les tâches et les priorités, tout en assurant une livraison continue et incrémentale des fonctionnalités.

Nous démontrerons également comment nous dirigeons un projet en utilisant cette méthode.

Commençons par présenter l'équipe dirigeante du projet, comme illustré dans le tableau suivant :

Table 1.4 – Équipe de SCRUM

Rôle	Personnel
Product owner	Mr Adnen Gaied
Scrum master	Mme Ibtihel Rebhi
Équipe de développement	Malek Fendri

- Mr **Adnen Gaied** , chef département Informatique au sein des Assurances BIAT.
- Mme **Ibtihel Rebhi** , cheffe département Business Analytics (BA) et Docteure-Ingénieur en Statistiques ches Esprit School of Business .
- **Malek Fendri** , stagiaire chez Assurances BIAT et étudiante en 3 ème année SIAD à Esprit School of Business .

Conclusion

En conclusion de ce premier chapitre, nous avons établi le cadre général de notre projet en présentant l'organisme d'accueil, en effectuant une analyse détaillée de l'existant, et en exposant les méthodologies de travail choisies.

Cette mise en contexte nous a permis de comprendre les enjeux spécifiques du projet, les ressources disponibles, et les défis à relever. L'étude approfondie de l'environnement actuel et le choix de méthodologies adaptées constituent des bases solides pour la suite de notre travail, assurant une approche structurée et efficace pour atteindre les objectifs fixés.

Chapitre 2: Planification du projet et outils techniques

Contents

2.1	Spécifications et analyse des besoins	14
2.1.1	Besoins fonctionnels	15
2.1.2	Besoins non fonctionnels	15
2.2	Planification	16
2.2.1	Backlog produit	16
2.2.2	Diagramme de GANTT	18
2.3	Méthodes techniques	18
2.3.1	Les systèmes décisionnels	19
2.3.2	L'intelligence artificielle et ses fondamentaux	21
2.3.3	Le développement web	23
2.4	Environnement technique	25
2.4.1	Outils de stockage des données	25
2.4.2	Outils pour ETL	26
2.4.3	Outils de restitution	26
2.4.4	Outils de développement de chatbot	27
2.4.5	Outils de développement de l'application web	28

Introduction

Ce chapitre présente l'environnement technique dans lequel le projet est réalisé, le diagramme de Gantt pour le planning des tâches, ainsi que le backlog produit détaillant les fonctionnalités à implémenter.

2.1 Spécifications et analyse des besoins

Nous présentons les besoins fonctionnels et non fonctionnels de notre projet.

2.1.1 Besoins fonctionnels

Les besoins fonctionnels, décrits ci-dessous, constituent les éléments essentiels que le système doit intégrer pour répondre efficacement aux exigences des utilisateurs et aux objectifs du projet.

1. Implémentation du Processus ETL

- Implémenter un processus ETL pour alimenter un DWH centralisé.

2. Visualisations Interactives

- Fournir des visualisations interactives et dynamiques pour faciliter la prise de décision.

3. Rapports Personnalisés

- Générer des rapports personnalisés basés sur les critères définis par l'utilisateur.

4. Réponses aux questions de l'utilisateur

- Répondre aux questions posées par l'utilisateur sur les dimensions (branche, produit, réseau de distribution, etc.) et les dates (par année, mois, jour).

5. Assistance technique

- Guider l'utilisateur à signaler des problèmes.

6. Authentification Sécurisée

- Authentification sécurisée pour les utilisateurs.

7. Gestion des Utilisateurs

- Gestion des utilisateurs pour les administrateurs.

8. Interactivité pour les Utilisateurs Standards

- Les utilisateurs standards peuvent interagir avec le chatbot, consulter les visualisations, et générer des rapports personnalisés.

2.1.2 Besoins non fonctionnels

Les besoins non fonctionnels, présentés ci-après, englobent les critères de qualité et les contraintes opérationnelles que le système doit respecter pour garantir une performance optimale.

1. Système d'Authentification Sécurisé

- Assurer que l'application web dispose d'un mécanisme d'authentification robuste pour protéger l'accès aux données et aux fonctionnalités. Seuls les utilisateurs autorisés peuvent accéder aux différentes sections de l'application.

2. Disponibilité

- Garantir que le système soit disponible 24/7 avec un temps d'arrêt minimal pour maintenance. Les utilisateurs doivent pouvoir accéder au système à tout moment sans interruption.

3. Ergonomie

- L'interface utilisateur doit être intuitive, facile à utiliser et accessible sur différents appareils, y compris les navigateurs web et les appareils mobiles. Les utilisateurs doivent pouvoir naviguer dans l'application sans difficulté.

4. Présentation

- Les visualisations et les rapports doivent être clairs et informatifs, facilitant l'interprétation des données. L'interface doit être esthétiquement plaisante et professionnelle, renforçant l'expérience utilisateur globale.

2.2 Planification

Cette section présente la planification du projet, incluant le backlog et le diagramme de Gantt. Le backlog priorise les tâches à réaliser, tandis que le diagramme de Gantt offre une vue d'ensemble des délais et de l'avancement des activités, facilitant ainsi le suivi et la gestion du projet.

2.2.1 Backlog produit

Voici le backlog produit, qui priorise les tâches à réaliser pour notre projet :

Table 2.1 – Backlog produit

ID	Équipe	User Story	Responsable	Tâches
1	Compréhension et génération des données	En tant qu'utilisateur, je veux comprendre et générer des données à partir des sources disponibles afin de les utiliser pour des analyses futures.	Analyste de données	<ul style="list-style-type: none"> — Comprendre les différentes sources de données. — Générer des données basées sur la source donnée.
2	ETL	En tant que développeur, je veux extraire, transformer et charger (ETL) les données de différentes sources dans le DWH afin de les rendre disponibles pour analyse.	Équipe de développement	<ul style="list-style-type: none"> — Extraire des données. — Appliquer les règles de transformation définies. — Charger les données dans le DWH. — Automatisation.

Table 2.1 – suite

ID	Équipe	User Story	Responsable	Tâches
3	Tableau de bord	En tant que décideur, je veux accéder à un tableau de bord interactif afin de visualiser les indicateurs clés de performance (KPI).	Product Owner	<ul style="list-style-type: none"> — L'intégration du DWH. — Créer un tableau de bord interactif avec des filtres et des visualisations.
4	Chatbot	En tant qu'utilisateur, je veux poser des questions au chatbot pour obtenir des informations spécifiques sur les données.	Équipe de développement	<ul style="list-style-type: none"> — Développer un chatbot capable de répondre aux questions en anglais et en français. — Développer une interface utilisateur pour le chatbot dans l'application web.
5	Application web	En tant qu'administrateur, je veux déployer l'application web afin que les utilisateurs puissent accéder au tableau de bord et au chatbot.	Équipe de développement	<ul style="list-style-type: none"> — Développer un système d'authentification pour les utilisateurs. — Déployer le tableau de bord sur l'application web. — Intégrer le chatbot dans l'application web.

Découpage des sprints

- **Release 1 : Compréhension et génération des données :**
 - **Sprint 1 :** Collecte et compréhension des données.
 - **Sprint 2 :** Génération des données.
- **Release 2 : Conception et modélisation du DWH :**
 - **Sprint 1 :** Conception multidimensionnelle du DWH .
 - **Sprint 2 :** Extraction des données.
 - **Sprint 3 :** Transformation des données .
 - **Sprint 4 :** Chargement de notre entrepôt de données .
 - **Sprint 5 :** Automatisation du processus ETL.
- **Release 3 : Restitution des données :**
 - **Sprint 1 :** Liaison avec notre entrepôt de données .

- **Sprint 2** : Mise en place des dashboards .
- **Release 4 : Développement du chatbot** :
 - **Sprint 1** : Développement du modèle de reconnaissance d'intentions et d'extraction d'entités.
 - **Sprint 2** : Création des histoires et des dialogues de formation.
 - **Sprint 3** : Implémentation des actions personnalisées .
 - **Sprint 4** : Tests et validation du chatbot.
- **Release 5 : Développement de l'application web** :
 - **Sprint 1** : Conception de l'architecture de l'application.
 - **Sprint 2** : Développement du backend (API, base de données et intégration du chatbot Rasa).
 - **Sprint 3** : Développement du frontend et son intégration avec le frontend.

2.2.2 Diagramme de GANTT

Voici le diagramme de Gantt, qui illustre visuellement le calendrier des tâches du projet, permettant de suivre les délais, les interdépendances et l'avancement des différentes activités.

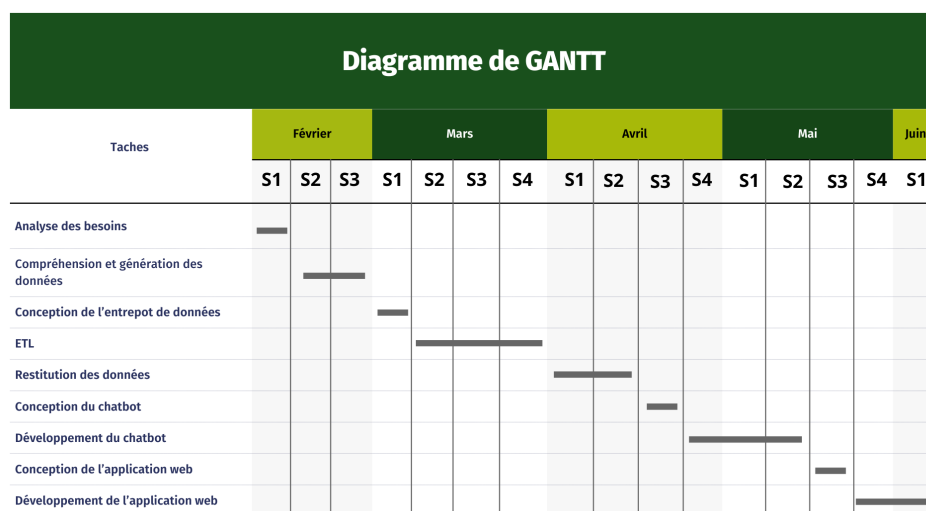


Figure 2.1 – Diagramme de GANTT

2.3 Méthodes techniques

Dans cette section, nous explorerons les différentes méthodes techniques pertinentes à notre projet, en abordant les systèmes décisionnels, l'intelligence artificielle et le développement web des applications.

2.3.1 Les systèmes décisionnels

Les systèmes décisionnels, aussi appelés systèmes d'aide à la décision (DSS), sont des outils informatiques créés pour assister les entreprises et les organisations dans la prise de décisions éclairées. Ils utilisent des techniques de collecte, de traitement et d'analyse des données pour convertir les données brutes en informations exploitables. Ces systèmes fournissent aux décideurs un accès rapide à des rapports, des analyses et des visualisations de données, simplifiant ainsi les décisions stratégiques fondées sur des faits et des analyses quantitatives.

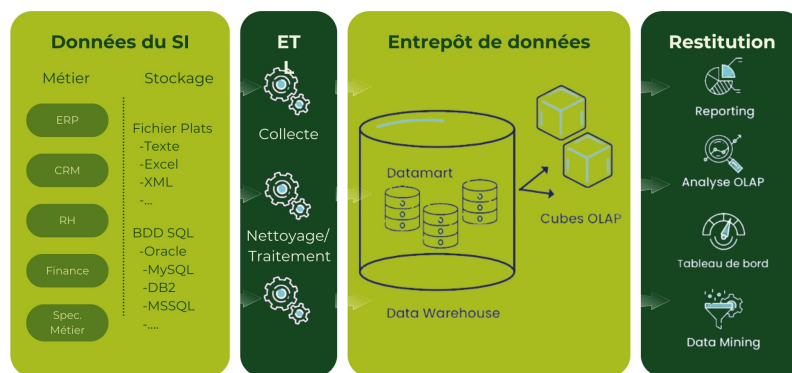


Figure 2.2 – Architecture du système décisionnel

Les systèmes décisionnels vs les systèmes transactionnels

OLAP (Online Analytical Processing) et OLTP (Online Transaction Processing) sont deux systèmes d'information qui sont conçus pour regrouper et organiser les données de deux manières distinctes.

Les systèmes décisionnels OLAP

Les systèmes OLAP collectent, traitent et analysent les données brutes pour les transformer en informations exploitables, permettant des analyses complexes, des rapports détaillés et la planification stratégique.

Les systèmes transactionnels OLTP

Les systèmes transactionnels (OLTP) gèrent et enregistrent efficacement les transactions quotidiennes d'une organisation, assurant rapidité, précision et intégrité des données pour des opérations telles que les ventes, achats et gestion de stock.

Table 2.2 – Tableau comparatif entre les systèmes décisionnels et systèmes transactionnels

Caractéristique	Système Transactionnel (OLTP)	Système Décisionnel (OLAP)
Objectif principal	Gestion des transactions courantes	Analyse des données pour la prise de décision
Type de données	Données détaillées, actuelles	Données agrégées, historiques
Opérations	Insertions, mises à jour, suppressions fréquentes	Lecture et analyse de données (requêtes complexes)
Modèle de données	Modèle normalisé pour éviter la redondance	Modèle dénormalisé pour améliorer les performances
Performance	Optimisé pour les transactions rapides	Optimisé pour les requêtes complexes
Utilisateurs	Opérationnels (caissiers, utilisateurs quotidiens)	Cadres, analystes, décideurs
Multidimensionnalité	Monodimensionnel (enregistrement de ligne)	Multidimensionnel (cubes de données)
Traitement	Transactionnel	Analytique

Les étapes d'un système décisionnel

La mise en place d'un système décisionnel suit plusieurs étapes cruciales pour garantir une collecte, une gestion et une analyse efficaces des données. Voici les principales étapes :

1. Définition des objectifs et des besoins : Identifier les objectifs stratégiques de l'entreprise et comprendre les besoins des utilisateurs finaux.

2. Collecte et Intégration des Données : Rassembler les données provenant de diverses sources, les intégrer dans un format cohérent et les stocker dans un entrepôt de données.

3. Nettoyage et Préparation des Données : Nettoyer les données pour éliminer les incohérences et les transformer pour les rendre utilisables pour l'analyse.

4. Modélisation des Données : Créer des modèles de données multidimensionnels, définir les dimensions et les mesures.

5. Développement des Applications Analytique : Concevoir et développer des outils et des applications pour analyser les données, tels que des tableaux de bord, des rapports, et des visualisations.

6. Mise en Œuvre et Test : Tester la performance, la fiabilité, et la précision des analyses fournies par le système décisionnel.

7. Formation et Adoption des Utilisateurs : Former les utilisateurs finaux à l'utilisation des outils et des applications analytiques.

8. Maintenance et Évolution : Assurer la maintenance continue du système pour corriger les problèmes et mettre à jour les données.

2.3.2 L'intelligence artificielle et ses fondamentaux

En élaborant notre projet nous avons fait recours à l'intelligence artificielle afin de développer un chatbot. L'IA est une branche de l'informatique qui crée des systèmes capables de réaliser des tâches nécessitant l'intelligence humaine. Ses principaux domaines sont :

- **Apprentissage Automatique** : Permet aux machines d'apprendre à partir de données et de s'améliorer sans être explicitement programmées.
- **Réseaux de Neurones Artificiels** : Modèles inspirés du cerveau humain, efficaces pour traiter des informations complexes comme les images, les sons et les textes.
- **Traitement du Langage Naturel** : Permet aux machines de comprendre et répondre au langage humain de manière naturelle, facilitant les interactions homme-machine.

Le traitement du Langage Naturel

Le langage naturel se réfère à la manière dont les humains communiquent naturellement, par la parole ou par écrit. Il comprend les nuances, les contextes et les significations implicites qui peuvent varier en fonction de nombreux facteurs tels que la culture, l'éducation et l'expérience personnelle. Le TLN englobe plusieurs sous-domaines, dont :

- **Analyse syntaxique** : Étude de la structure grammaticale des phrases.
- **Analyse sémantique** : Compréhension du sens des mots et des phrases.
- **Reconnaissance des entités nommées** : Identification de noms propres (personnes, lieux, organisations, etc.) dans un texte.
- **Analyse du sentiment** : Détection des émotions ou des attitudes exprimées dans le texte.

Applications des Chatbots

Les chatbots, étant un aspect de l'intelligence artificielle, sont de plus en plus présents dans divers secteurs, offrant des solutions innovantes pour :

- **Service à la Clientèle** : Fournir une assistance immédiate et automatisée, réduisant les temps d'attente et améliorant la satisfaction client.
- **Commerce Électronique** : Aider les clients à trouver des produits, passer des commandes et obtenir des informations, augmentant ainsi les taux de conversion et de satisfaction.
- **Assistance Technique** : Résoudre les problèmes techniques courants, offrant un support 24/7 sans intervention humaine.

Architecture d'un Chatbot Basé sur RASA

Rasa est une plateforme open-source utilisée pour le développement de chatbots avancés. L'architecture d'un chatbot basé sur Rasa comporte plusieurs composants essentiels qui collaborent pour comprendre et répondre aux requêtes des utilisateurs. Voici une description détaillée des différentes parties de cette architecture, illustrée par l'image ci-dessous :

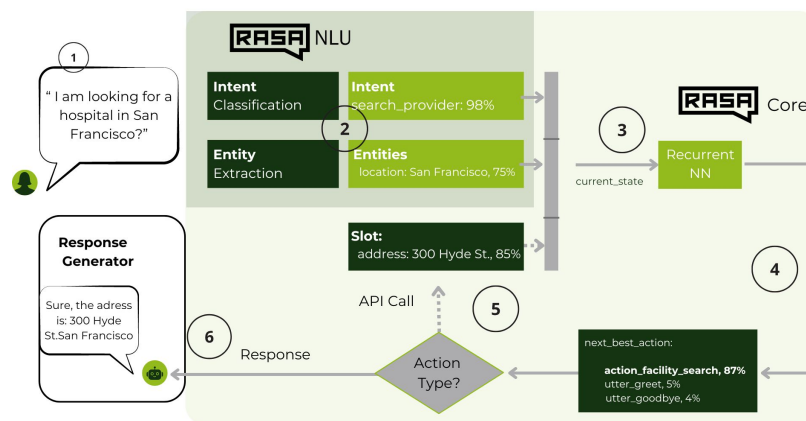


Figure 2.3 – Architecture d'un chatbot basé sur RASA

— Interaction de l'utilisateur

- L'utilisateur interagit avec le chatbot via une interface utilisateur (UI), en posant une question ou en faisant une demande, par exemple : "Je cherche un hôpital à San Francisco ?"

— Rasa NLU (Natural Language Understanding)

- **Classification de l'intention** : Le module NLU identifie l'intention de l'utilisateur, ici "search_provider" avec une probabilité de 98 %.
- **Extraction des entités** : Le module NLU extrait les entités pertinentes de la question, ici "location" avec la valeur "San Francisco" et une probabilité de 75 %.

— État actuel du dialogue

- Les informations extraites sont utilisées pour mettre à jour l'état actuel du dialogue, qui est ensuite transmis au module Rasa Core.

— Rasa Core (Gestion du Dialogue)

- **Réseau de Neurones Récurrents (RNN)** : Utilisé pour prédire la prochaine meilleure action à prendre en se basant sur l'état actuel du dialogue.
- **Prochaine meilleure action** : Par exemple, "action_facility_search" avec une probabilité de 87 %.

- **Actions Personnalisées**

- Les actions personnalisées sont définies dans le fichier `actions.py` et peuvent inclure des interactions avec des API externes, des bases de données, ou d'autres systèmes pour récupérer ou envoyer des données spécifiques. Par exemple, l'action "action_facility_search" peut faire un appel API pour récupérer la liste des hôpitaux à San Francisco.

- **Appel API pour obtenir des informations supplémentaires :**

- Le chatbot peut effectuer un appel API externe pour récupérer des données nécessaires à la réponse, par exemple, une recherche d'adresse.

- **Génération de la réponse**

- La réponse est générée en fonction des informations recueillies. Par exemple : "L'adresse est : 300 Hyde St., San Francisco."

- **Réponse à l'utilisateur**

- La réponse est envoyée à l'utilisateur via l'interface utilisateur.

- **Stories et Training Data**

- **Stories** : Les stories dans RASA sont des exemples de dialogues entre l'utilisateur et le chatbot, utilisés pour entraîner le modèle à comprendre le flux de la conversation.

2.3.3 Le développement web

Le développement web consiste à créer et maintenir des sites web et des applications web. Il comprend plusieurs aspects, dont la conception (frontend), la programmation côté serveur (backend), la gestion de bases de données, et l'intégration des fonctionnalités pour offrir une expérience utilisateur optimale.

Développement avec des Frameworks

Un framework est un ensemble de bibliothèques et d'outils qui aident les développeurs à construire des applications plus rapidement et de manière plus cohérente. Les frameworks web sont essentiels car ils fournissent des solutions prêtes à l'emploi pour les problèmes communs et facilitent le respect des bonnes pratiques de développement.

Les frameworks frontend, comme React, Angular, et Vue.js, aident à construire des interfaces utilisateur interactives. Les frameworks backend, comme Django, Flask, Ruby on Rails, et Express.js, facilitent la gestion des serveurs, des bases de données, et de la logique applicative.

Voici quelques avantages clés des frameworks :

1. **Productivité Améliorée** : Les frameworks réduisent le temps de développement en fournissant des modèles et des bibliothèques prêtes à l'emploi pour des tâches courantes telles que la gestion des URL, la validation des formulaires et l'interaction avec les bases de données.
2. **Sécurité Renforcée** : En suivant les bonnes pratiques de sécurité par défaut, les frameworks réduisent les risques de failles de sécurité telles que les injections SQL et les attaques XSS (Cross-Site Scripting).

3. **Structuration du Code** : Les frameworks encouragent une structure de projet cohérente et bien organisée, facilitant la maintenance à long terme et la collaboration au sein des équipes de développement.
4. **Écosystème et Communauté** : Les frameworks populaires comme Django, Rails, et Express bénéficient de vastes communautés de développeurs et d'une multitude de modules et extensions, offrant ainsi un support et des fonctionnalités supplémentaires.

Développement d'une Application Web avec Flask

Flask est un micro-framework pour Python, ce qui signifie qu'il est léger et extensible. Il est souvent préféré pour les projets où la flexibilité et la simplicité sont essentielles.

Flask est basé sur Werkzeug, une bibliothèque de WSGI, et Jinja2, un moteur de template. Il utilise un système de routage pour diriger les requêtes HTTP vers les fonctions de vue correspondantes.

Architecture Flask

L'architecture de flask peut être divisée en trois couches principales : la couche de template, la couche backend, et la couche base de données.

1. **Couche de Template (Layer Template)** La couche de template est responsable de la présentation de l'application web. Elle utilise des technologies comme HTML (HyperText Markup Language) pour structurer le contenu des pages web, CSS (Cascading Style Sheets) pour styliser et mettre en forme ce contenu, et JavaScript pour ajouter des interactions dynamiques et des comportements interactifs aux pages web. Des frameworks comme Bootstrap peuvent être utilisés pour faciliter la création de designs responsives et mobiles. Jinja2, le moteur de templates utilisé par Flask, permet de générer des pages HTML dynamiques en insérant des données dans des templates HTML.
2. **Couche Backend (Layer Backend)** La couche backend est le cœur logique de l'application. Elle est principalement développée en utilisant Flask, un framework web léger pour Python. Flask gère les requêtes et réponses HTTP, c'est-à-dire qu'il reçoit les requêtes des utilisateurs, les traite et renvoie les réponses appropriées. Le backend contient toute la logique métier de l'application, comme la gestion des utilisateurs, l'authentification, la validation des données et les interactions avec la base de données.
3. **Couche Base de Données (Layer Database)** La couche base de données est responsable du stockage persistant des données de l'application. Dans l'architecture illustrée, PostgreSQL est utilisé comme système de gestion de base de données relationnelle. Le backend, via Flask, utilise des bibliothèques de connexion à la base de données pour exécuter des requêtes SQL, récupérer des données ou effectuer des opérations de mise à jour et de suppression.

2.4 Environnement technique

Dans cette section, nous détaillons l'environnement technique mis en place pour le projet, en abordant les différents outils et technologies utilisés pour le stockage des données, les processus ETL, la visualisation des données, ainsi que le développement de chatbots et de solutions web.

2.4.1 Outils de stockage des données

Nous allons commencer par examiner et comparer les différents outils de stockage des données.

Pgadmin

pgAdmin est une interface graphique open-source utilisée pour gérer et administrer les bases de données PostgreSQL. Elle offre des fonctionnalités avancées telles que la gestion des objets de base de données, l'exécution de requêtes SQL, et la surveillance des performances, facilitant ainsi l'administration des bases de données PostgreSQL.



Figure 2.4 – Logo de l'outil pgadmin

MySQL

MySQL est un système de gestion de bases de données relationnelles open-source très populaire, connu pour sa rapidité, sa fiabilité et sa facilité d'utilisation. Utilisé par de nombreuses applications web et entreprises, MySQL permet de stocker, manipuler et gérer efficacement les données grâce à une vaste gamme de fonctionnalités et d'outils intégrés.



Figure 2.5 – Logo de l'outil MySQL

L'outil retenu

Après avoir présenté les outils de stockage des données les plus reconnus, nous avons opté pour **PostgreSQL** en raison de sa fiabilité, de ses performances optimales et de son extensibilité.

2.4.2 Outils pour ETL

Dans cette section, nous allons comparer les différents outils d'ETL afin de guider le choix de l'outil le plus adapté aux besoins spécifiques de notre projet.

Talend Open Studio

Talend est une suite d'outils ETL open-source et commerciaux, offrant une intégration facile des données de diverses sources. Ses principaux avantages incluent une interface conviviale, une grande flexibilité, et des fonctionnalités avancées pour la qualité des données et le master data management (MDM).



Figure 2.6 – Logo de l'outil Talend Open Source

Microsoft SQL Server

Microsoft SQL Server Integration Services (SSIS) est un outil ETL puissant et polyvalent inclus dans SQL Server. Il permet de déplacer, transformer et charger des données depuis diverses sources. Les principaux avantages de SSIS incluent une intégration étroite avec l'écosystème Microsoft, des performances élevées, et des capacités avancées de gestion des flux de travail et des erreurs.



Figure 2.7 – Logo de l'outil Microsoft SQL Server

L'outil retenu

Après avoir comparé les différents outils ETL, nous avons opté pour **Talend** en raison de ses fonctionnalités supérieures. Talend offre une interface conviviale, une grande flexibilité et une intégration facile avec diverses sources de données. Ses capacités avancées en matière de qualité des données et de master data management (MDM) en font le choix idéal pour notre projet.

2.4.3 Outils de restitution

Lors de la restitution des données, le choix de l'outil adéquat est primordial pour garantir une analyse précise et pertinente. Parmi les outils les plus couramment utilisés, on retrouve Power BI et Tableau, chacun

offrant des fonctionnalités spécifiques pour répondre aux besoins variés des utilisateurs.

Power BI

Power BI, développé par Microsoft, est un outil de business intelligence qui transforme des données brutes en visualisations interactives et exploitables. Il se distingue par sa capacité à intégrer diverses sources de données, ses visualisations avancées, son interface conviviale et son intégration avec d'autres produits Microsoft, facilitant la création de tableaux de bord et de rapports pour une prise de décision éclairée.



Figure 2.8 – Logo de l'outil Power BI

Tableau

Tableau est un outil de business intelligence qui permet de créer des visualisations interactives, des tableaux de bord et des rapports à partir de diverses sources de données. Il se distingue par sa facilité d'utilisation, sa rapidité de traitement et ses puissantes fonctionnalités de visualisation, permettant même aux utilisateurs non techniques de produire des analyses complexes et de partager des insights clairs.



Figure 2.9 – Logo de l'outil Tableau

L'outil retenu

Pour répondre efficacement aux besoins de notre projet, nous avons choisi d'utiliser **Power BI**. Cet outil de business intelligence de Microsoft offre des capacités avancées de visualisation et d'analyse des données, ainsi qu'une intégration fluide avec diverses sources de données. Power BI nous permettra de créer des tableaux de bord interactifs et des rapports détaillés.

2.4.4 Outils de développement de chatbot

Dans cette section, nous aborderons les outils que nous avons utilisés pour développer notre chatbot.

Visual Studio

Visual Studio est un environnement de développement intégré (IDE) complet et polyvalent développé par Microsoft. Il prend en charge de nombreux langages de programmation et offre des fonctionnalités avancées pour le codage, le débogage et les tests. Idéal pour créer des applications de bureau, web, mobiles et cloud.



Figure 2.10 – Logo de l'environnement Visual Studio

RASA

Rasa est une plateforme open-source pour créer des chatbots et des assistants virtuels. Elle est gratuite, hautement personnalisable et utilise le machine learning pour gérer les conversations intelligemment. Flexible et compatible avec diverses plateformes.



Figure 2.11 – Logo de l'outil RASA

2.4.5 Outils de développement de l'application web

Dans cette section, nous aborderons les outils que nous avons utilisés pour développer notre application web afin de déployer notre dashboard et notre chatbot.

Star UML

StarUML est un outil de modélisation logiciel pour créer des diagrammes UML. Il se distingue par son interface intuitive, sa prise en charge de nombreux types de diagrammes, et sa capacité d'intégration avec d'autres outils de développement grâce à des extensions et des scripts personnalisés.



Figure 2.12 – Logo de l'outil Star UML

Flask

Flask est un micro-framework web en Python, apprécié pour sa simplicité et sa flexibilité. Il est léger, facile à utiliser, et hautement extensible, permettant aux développeurs d'ajouter des fonctionnalités selon leurs besoins. Flask offre un contrôle total sur le développement de l'application et bénéficie d'une large communauté de support.



Figure 2.13 – Logo du framework Flask

Conclusion

Dans ce chapitre, nous avons procédé à une analyse détaillée des besoins, établi une planification rigoureuse du projet en utilisant le backlog produit et le diagramme de Gantt, et mis en place l'environnement technique nécessaire. Ces étapes cruciales nous ont permis de définir clairement les objectifs, de prioriser les fonctionnalités, et de planifier efficacement les tâches et les ressources, assurant ainsi une base solide pour le succès du projet.

Chapitre 3: Préparation des données et modélisation du Data Warehouse

Contents

3.1	Sprint Backlog	31
3.2	Compréhension et exploration des données	31
3.2.1	Compréhension des données	32
3.3	Génération des données	33
3.4	Chargement des données sources	34
3.5	La modélisation du Data Warehouse	37
3.5.1	Les différents modèles conceptuels	38
3.5.2	Choix du sujet d'analyse	39
3.5.3	Choix des dimensions	40
3.5.4	Choix des mesures	40
3.5.5	Notre modèle conceptuel	41
3.6	Intégration de l'entrepôt de données	41
3.6.1	Chargement des dimensions	41
3.6.2	Chargement de la table de fait	47
3.6.3	Automatisation du processus ETL	48

Introduction :

Dans ce chapitre, nous nous concentrerons sur la préparation des données et la modélisation du Data Warehouse. Nous débuterons par une analyse approfondie des données, en examinant leur structure, leur qualité et leur intégrité. Ensuite, nous explorerons les techniques de génération de données afin de construire un entrepôt de données robuste et adapté aux besoins décisionnels de l'entreprise.

3.1 Sprint Backlog

Ceci est le sprint backlog pour la compréhension, génération et chargement des données dans un ODS :

Table 3.1 – Sprint Backlog : Compréhension, génération et chargement des données

Sprint	User Story	Scrum Task	Priorité
Sprint 1	En tant qu'analyste de données, je souhaite comprendre les exigences en matière de données afin de garantir que toutes les données nécessaires sont identifiées.	<ul style="list-style-type: none"> — Compréhension des données sources — Développement d'un code Python pour la génération des données 	Haute
	En tant qu'administrateur de base de données, je souhaite importer les fichiers de données.	<ul style="list-style-type: none"> — Création de la base sur Postgres — Importation des fichiers générés — Chargement des données dans l'ODS 	Haute
Sprint 2	En tant qu'analyste de données, je veux concevoir et modéliser l'entrepôt de données.	<ul style="list-style-type: none"> — Conception de l'entrepôt de données. 	Haute
	En tant qu'analyste de données, je souhaite charger mon entrepôt de données et automatiser son exécution .	<ul style="list-style-type: none"> — Extraction des données. — Transformation des données — Chargement des données dans le DWH . — Automatisation du processus ETL . 	Haute

3.2 Compréhension et exploration des données

Dans cette section, nous aborderons les étapes de la compréhension et de la génération des données. Nous analyserons d'abord les données disponibles pour identifier leur structure ,leur qualité et leur perti-

nence. Ensuite, nous explorerons les techniques de génération de données, y compris la création de données de test et la simulation de données manquantes.

3.2.1 Compréhension des données

On nous ont fourni un fichier Excel afin d’analyser la structure des données disponibles. Notre objectif est d’identifier les champs importants et de comprendre leur organisation. Cette analyse nous permettra ensuite de créer une base de données adaptée et efficace. La structure du fichier est la suivante :

A	B	C	D	E	F
BQ	MPTE_BENEFICIAIRE	NATURE_FLUX	CODE_BUDGETAIRE	LIBELLÉ	SENS
BIAT	GB27AJXF98352i	Chèques	R-CA_VIE_PREV	Décaissement p D	
BIAT	GB74YUTH70992	Virement Recu	R-CA_VIE_PREV	Encaissement p C	
BIAT	GB39VRJD73484	Chèques	R-CA_VIE_EPAR	Encaissement p C	
BIAT	GB28PCNF38797	Virement Recu	R-CA_VIE_PREV	Encaissement p C	
BIAT	GB90CDOW3319	Chèques	R-CA_VIE_PREV	Décaissement p D	
BIAT	GB18QUGO1593	Virement Recu	R-CA_NVIE	Décaissement p D	
BIAT	GB98PZAG22447	R-PRLV PA	R-CA_NVIE	Encaissement p C	
BIAT	GB34FTQX74520	Chèques	R-CA_VIE_EPAR	Encaissement p C	
BIAT	GB94RMR93385	R-PRLV PA	R-CA_VIE_PREV	Décaissement p D	
BIAT	GB26YFKH31183	Virement Recu	R-CA_VIE_EPAR	Décaissement p D	
BIAT	GB12TTNU32916	Chèques	R-CA_NVIE	Décaissement p D	
BIAT	GB31FXWL97341	Virement Recu	R-CA_VIE_PREV	Décaissement p D	
BIAT	GB49JLOA60230	Virement Recu	R-CA_VIE_PREV	Encaissement p C	
BIAT	GB91GOIX58639	R-PRLV PA	R-CA_NVIE	Décaissement p D	
BIAT	GB79OPFH04644	Virement Recu	R-CA_VIE_EPAR	Décaissement p D	
BIAT	GB93CEQD43392	Virement Recu	R-CA_NVIE	Décaissement p D	
BIAT	GB95YEA13474	Espèce	R-CA_NVIE	Décaissement p D	
BIAT	GB88VGLJ70490	Virement Recu	R-CA_VIE_EPAR	Encaissement p C	
BIAT	GB42VQLW0915	Virement Recu	R-CA_VIE_PREV	Décaissement p D	
BIAT	GB18PPT23100	Virement Recu	R-CA_NVIE	Décaissement p D	
BIAT	GB28DCF736422	Virement Recu	R-CA_VIE_EPAR	Encaissement p C	

Figure 3.1 – Données sources

G	H	I	J	K	L
MONTANT_TR	DEV_TR	DATE_OPERATION	DATE_VALEUR	REFERENCE	TYPE_MVT
9461,678 TND		2022-06-09	2022-06-09	CEOP531439071	Décaissement
3992,149 TND		2023-04-22	2023-04-22	UAOS065406216	Encaissement
55582,633 TND		2023-02-25	2023-02-25	JLIN7394856969	Encaissement
3517,979 TND		2022-05-03	2022-05-03	EPOJ7229748266	Encaissement
5251,883 TND		2024-03-09	2024-03-09	PQEN026933812	Décaissement
4004,421 TND		2023-01-12	2023-01-12	SCOH781391548	Décaissement
4181,266 TND		2024-02-15	2024-02-15	CODW96484518	Encaissement
74861,515 TND		2022-09-10	2022-09-10	EOKI2731185874	Encaissement
6372,621 TND		2022-12-03	2022-12-03	ODHK315312583	Décaissement
4869,153 TND		2021-09-08	2021-09-08	JJLR8788432365	Décaissement
2473,682 TND		2023-05-27	2023-05-27	JZRV8227343281	Décaissement
1235,428 TND		2022-12-26	2022-12-26	MJMS893548714	Décaissement
6712,79 TND		2023-01-31	2023-01-31	SLGF0173131497	Encaissement
2292,921 TND		2023-02-16	2023-02-16	RJAY6565786709	Décaissement
5211,91 TND		2023-12-13	2023-12-13	XCJB3475062918	Décaissement
1625,461 TND		2022-09-06	2022-09-06	JXAI2707142926	Décaissement
6060,223 TND		2021-08-21	2021-08-21	GWIC520786049	Décaissement
6046,686 TND		2023-03-08	2023-03-08	LISY2327181711	Encaissement
980,465 TND		2023-03-24	2023-03-24	SWXY449252996	Décaissement
7679,718 TND		2021-07-03	2021-07-03	BCNZ501859306	Décaissement
14799,366 TND		2022-12-17	2022-12-17	IMTE2504733106	Encaissement

Figure 3.2 – Données sources (Suite)

M	N	O	P	Q	R
NB_BDX	ID_FLUX	BRANCHE	AUX_DISTRIBUT	PRODUIT	NOM
7645	30743	Prévoyance	bureaux directs	produit prévoy	Gharbi
61410	42842	Prévoyance	banque	produit prévoy	Ben Ali
7384	176267	Epargne	bureaux directs	produit épargne	Haddad
18706	2958	Prévoyance	bureaux directs	produit prévoy	Trabelsi
187335	88215	Prévoyance	agents generale	produit prévoy	Haddad
196897	95084	Transports	siège	MULTIRISQUE B/	Bouzid
118034	183175	Santé	bureaux directs	MANUAL - Santé	Trabelsi
142488	123489	Epargne	agents generale	produit épargne	Essid
158861	128457	Prévoyance	courtiers	produit prévoy	Bouzid
18160	4460	Epargne	banque	produit épargne	Triki
146037	198951	RC générale	agents generale	MANUAL - RC gé	Ben Ali
103110	24417	Prévoyance	courtiers	produit prévoy	Gharbi
88066	36729	Prévoyance	agents generale	produit prévoy	Chouchane
68999	75366	Risques divers,	courtiers	MULTIRISQUE H/	Bouzid
23221	134530	Epargne	bureaux directs	produit épargne	Mekki
102743	144939	Incendie	bureaux directs	BRIS DE MACHIN	Trabelsi
169917	14072	Construction	siège	MANUAL - Cons	Bouazizi
83518	174216	Epargne	courtiers	produit épargne	Ben Ali
44991	101021	Prévoyance	bureaux directs	produit prévoy	Fendri
69903	112803	RC générale	bureaux directs	MANUAL - RC gé	Bouzid
35861	116151	Epargne	bureaux directs	produit épargne	Bouazizi

Figure 3.3 – Données sources (Suite)

S	T	U	V	W	X	Y
PRÉNOM	GENRE	DATE_DE_NAISSAN	PAYS	VILLE	ACTIVITÉ	
Anis	M	2005-12-03	Tunisia	Gabes	Ouvrier(e)	
Mehdri	M	1950-12-18	Tunisia	Sfax	Retraité(e)	
Meriem	F	1986-03-18	Tunisia	Kebili	Fermier(e)	
Nada	F	2002-09-14	Tunisia	Gabes	Fermier(e)	
Fatma	F	1968-11-29	Tunisia	Silliana	Ouvrier(e)	
Melek	M	2000-10-22	Tunisia	Bizerte	Professeur	
Youssef	M	1940-07-03	Tunisia	Ariana	Retraité(e)	
Amel	F	1976-04-23	Tunisia	Manouba	Technicien	
Youssef	M	1981-02-26	Tunisia	Bizerte	Médecin	
Housse	M	1958-11-26	Tunisia	Kebili	Ouvrier(e)	
Mohamed	M	1949-01-09	Bahamas	Lake Kelly	Retraité(e)	
Meriem	F	1996-11-30	Tunisia	Tataouine	Ouvrier(e)	
Amel	F	1971-09-02	Tunisia	Mahdia	Chauffeur	
Mehdri	M	2004-05-21	Tunisia	Ben Arous	Artisan	
Sami	M	2003-09-15	Iran	Daleport	Medical technical officer	
Youssef	M	1967-12-26	Tunisia	Tozeur	Ouvrier(e)	
Ali	M	1995-05-27	Bhutan	Jocelynmouth	Océanographe	
Housse	M	1974-04-18	Tunisia	Sousse	Technicien	
Housse	M	1938-09-27	Tunisia	Sidi Bouzid	Retraité(e)	
Amel	F	1953-05-21	Tunisia	Sfax	Retraité(e)	
Ali	M	2004-04-03	Tunisia	Gafsa	Indépendant	

Figure 3.4 – Données sources (Suite)

La table suivante présente chaque colonne avec sa propre description pour expliquer le contenu de la table des données sources .

Table 3.2 – Description des colonnes de la table source

Colonne	Description
NB_BDX	Numéro de bordereau.
ID_FLUX	Identifiant du flux.
BRANCHE	Branche d'assurance (par exemple, prévoyance, santé, épargne).
RÉSEAUX_DISTRIBUTIONS	Canal de distribution (par exemple, bureaux directs, agents généraux, courtiers).
PRODUIT	Type de produit d'assurance.
NOM	Nom de l'assuré.
PRÉNOM	Prénom de l'assuré.
GENRE	Sexe de l'assuré (M/F).
DATE_DE_NAISSANCE	Date de naissance de l'assuré.
PAYS	Pays de résidence de l'assuré.
VILLE	Ville de résidence de l'assuré.
ACTIVITÉ	Profession ou activité de l'assuré.
BQ	Code de la banque.
COMPTE_BENEFICIAIRE	Numéro de compte du bénéficiaire.
NATURE_FLUX	Type de flux financier (par exemple, chèque, virement reçu).
CODE_BUDGETAIRE	Code budgétaire.
LIBELLÉ	Description ou intitulé de l'opération.
SENS	Sens de l'opération (Débit ou Crédit).
MONTANT_TR	Montant de la transaction.
DEV_TR	Devise de la transaction.
DATE OPERATION	Date de l'opération.
DATE VALEUR	Date de valeur de l'opération.
REFERENCE	Référence de l'opération.
TYPE MVT	Type de mouvement (par exemple, encaissement, décaissement).

3.3 Génération des données

Dans cette section, nous nous pencherons sur la génération des données, un aspect crucial pour assurer la robustesse d'un système décisionnel. En raison de la confidentialité des informations, il est nécessaire de générer notre propre base de données. Ces étapes sont indispensables pour garantir un entrepôt de données complet et prêt pour des analyses approfondies.

Nous avons défini les valeurs possibles pour chaque colonne des transactions financières et des informations d'assurance. Cela inclut les types de flux financiers, les branches d'assurance, les valeurs de sens des transactions, et les codes budgétaires, les produits d'assurance disponibles, leurs sous-catégories, ainsi que les réseaux de distribution et les régions en Tunisie.

Nous avons appliqué des conditions spéciales pour ajuster les montants et les dates des transactions en fonction de l'année et du sens (crédit ou débit), et ajouté ces transactions aux listes de données. Nous avons également vérifié si le compte bénéficiaire était unique, ajoutant les détails du client à une liste séparée pour garantir l'unicité des comptes.

3.4 Chargement des données sources

Cette section a pour objectif de transformer tous les fichiers Excel en tables au sein d'une base de données. Pour cela, la première étape consiste à se connecter à PostgreSQL, à créer une nouvelle base de données, puis à importer les données.

— Connexion à PostgreSQL

Nous sommes connectés en premier lieu à PostgreSQL comme illustré ci-dessous :

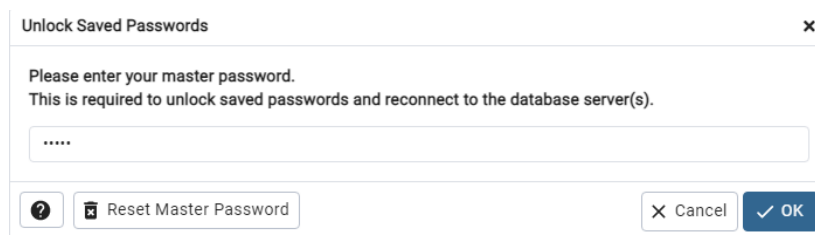


Figure 3.5 – Page de connexion PostgreSQL

— Création de la base de données DWH_PFE

Dans cette étape, nous avons créé une base de données nommé DWH_PFE comme illustré ci-dessous :

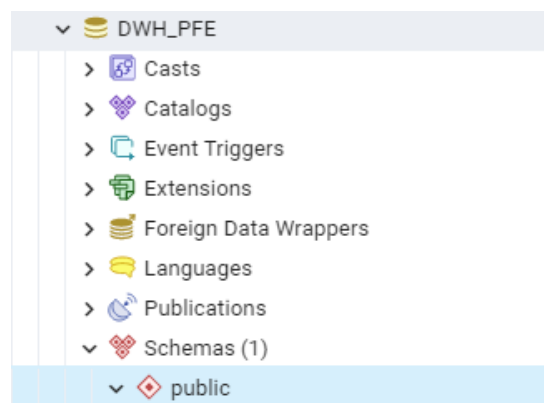


Figure 3.6 – Création de notre base de données sur postgres

— Importation des fichiers Excel

Vous trouverez ci-dessous la figure qui illustre l'importation des fichiers Excel dans Talend :

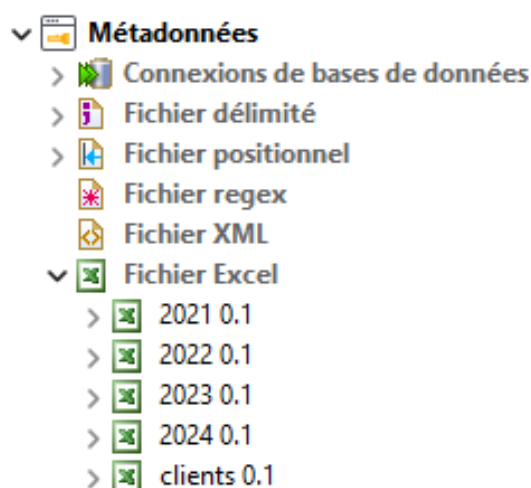


Figure 3.7 – Importation des fichiers sources

— Etablissement de la connexion entre Talend et PostgreSQL

La figure ci-dessous illustre la connexion établie entre Talend et la base de données :

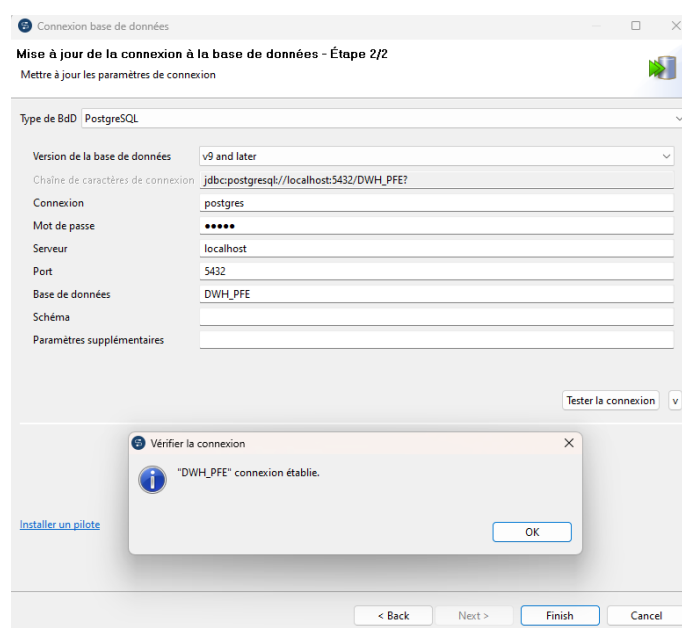


Figure 3.8 – Fenêtre d'établissement de la connexion

Nous abordons maintenant l'étape de la création des jobs pour importer les données sources des fichiers Excel dans un staging area. L'exemple ci-dessous illustre un job visant à importer les données du fichier "2021" dans le staging area "STG_2021".

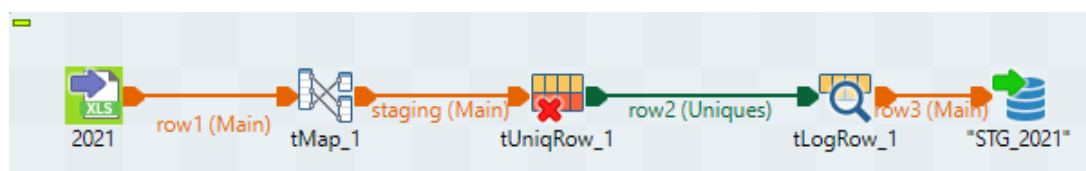


Figure 3.9 – Exemple JOB "STG_2021"

Pour effectuer le staging de cette table, nous avons utilisé les composants suivants :

- tMap : Cette composante permet de transformer et mapper les colonnes d'entrée aux colonnes de sortie.



Figure 3.10 – L'icône Tmap

- tUniqRow : Cette composante est utilisée pour filtrer les lignes uniques dans les données.



Figure 3.11 – L'icône tUniqRow

- tLogRow : Cette composante affiche ou enregistre les lignes de données à des fins de log ou de débogage.



Figure 3.12 – L'icône tLogRow

Après l'exécution de tous les jobs, toutes les tables ont été chargées dans la base de données, comme suit :

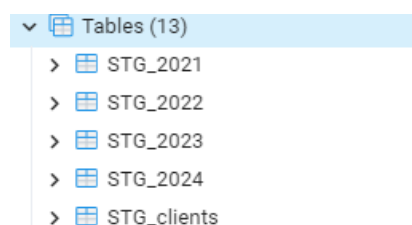


Figure 3.13 – Base de données Postgres

3.5 La modélisation du Data Warehouse

La modélisation du Data Warehouse consiste à structurer les données de manière à faciliter leur stockage, leur accès et leur analyse. Cette phase inclut généralement les trois principales approches : le modèle en étoile et le modèle en flocon de neige. Chaque modèle intègre le concept de table de faits et de table de dimensions.

- **Table de Faits** : La table de faits regroupe les mesures ou indicateurs clés de performance utilisés lors de l'analyse.

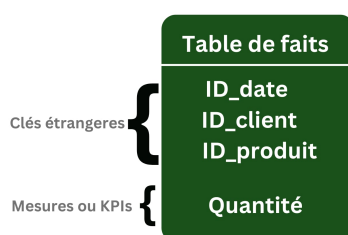


Figure 3.14 – Structure de la table de fait

- **Table de Dimensions** : La table de dimensions est une table dénormalisée qui inclut une clé primaire ainsi que les attributs descriptifs spécifiques à cette dimension.

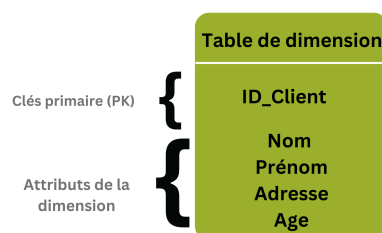


Figure 3.15 – Structure de la table de dimension

3.5.1 Les différents modèles conceptuels

Pour structurer efficacement les données et leurs relations dans un entrepôt de données, il est essentiel d'adopter un modèle de données précis. Trois modèles multidimensionnels distincts sont disponibles à cet effet :

- **Le modèle en étoile** : utilisé dans les entrepôts de données, comporte une table de faits centrale liée à plusieurs tables de dimensions. Il facilite les analyses rapides et simples.

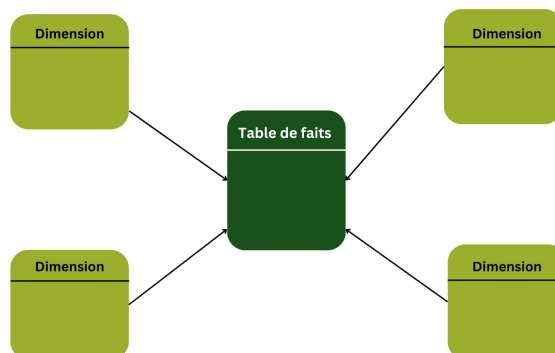


Figure 3.16 – Modèle en étoile

- **Le modèle en flocon de neige** : est une structure de données utilisée dans les entrepôts de données, où les tables de dimensions sont normalisées en plusieurs tables liées. Cela réduit la redondance des données .

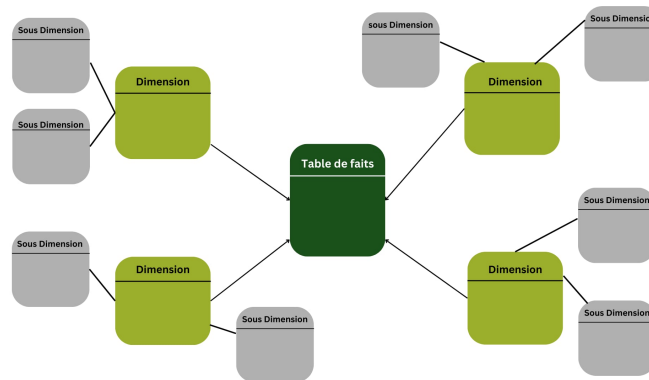


Figure 3.17 – Modèle en flocon de neige

- **Le modèle en constellation** : regroupe plusieurs modèles en étoile en partageant des tables de dimensions entre différentes tables de faits, permettant ainsi une analyse plus complexe et interconnectée des données.

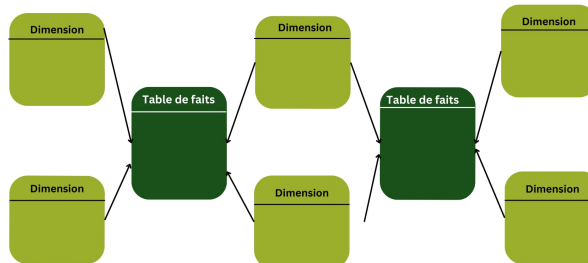


Figure 3.18 – Modèle en constellation

Sélection du modèle conceptuel

Nous avons opté pour le modèle en étoile comme modèle de données, car il s'est avéré le plus adapté à notre projet pour la création d'un entrepôt de données.

3.5.2 Choix du sujet d'analyse

Dans le cadre de notre projet, nous souhaitons mettre en place un entrepôt de données pour la gestion des flux, incluant les opérations d'encaissement et de décaissement .

3.5.3 Choix des dimensions

En ce qui concerne le choix des dimensions, nous avons décidé d'analyser le sujet selon les sept dimensions suivantes :

- **Dimension_Branche** : une table de dimension qui représente les différentes branches ou unités .
- **Dimension_Client** : Une table de dimension qui se concentre sur les caractéristiques des clients. elle fournit une compréhension approfondie des interactions financières avec chaque client.
- **Dimension_Nature** : Une table de dimension qui représente les modes de paiement utilisés, tels que les chèques ou les espèces.
- **Dimension_Produit** : Une table de dimension qui affiche les produits . Elle permet de suivre les flux financiers associés à chaque produit
- **Dimension_Réseaux** : Une table de dimension qui se réfère aux canaux par lesquels les transactions sont effectuées.
- **Dimension_Date** : Une table de dimension qui affiche les dates des opérations. Elle permet de segmenter les données selon les périodes temporelles
- **Dimension_Sens** : Une table de dimension qui distingue les mouvements de trésorerie entrants des mouvements sortants.

3.5.4 Choix des mesures

- **Montant** : Agrégé et analysé en fonction des différentes dimensions. Cela permet de quantifier précisément les flux financiers et d'identifier les tendances et les anomalies dans les encaissements et décaissements de l'organisation.
- **Montant net** : Agrégé et analysé en fonction des différentes dimensions. Cela permet de quantifier précisément les flux financiers et d'identifier les tendances et les anomalies dans les encaissements et décaissements de l'organisation.
- **Variation du montant net** : Agrégé et analysé en fonction des différentes dimensions. Cela permet de quantifier précisément les flux financiers et d'identifier les tendances et les anomalies dans les encaissements et décaissements de l'organisation.

3.5.5 Notre modèle conceptuel

La figure suivante illustre le modèle en étoile de notre entrepôt de données :

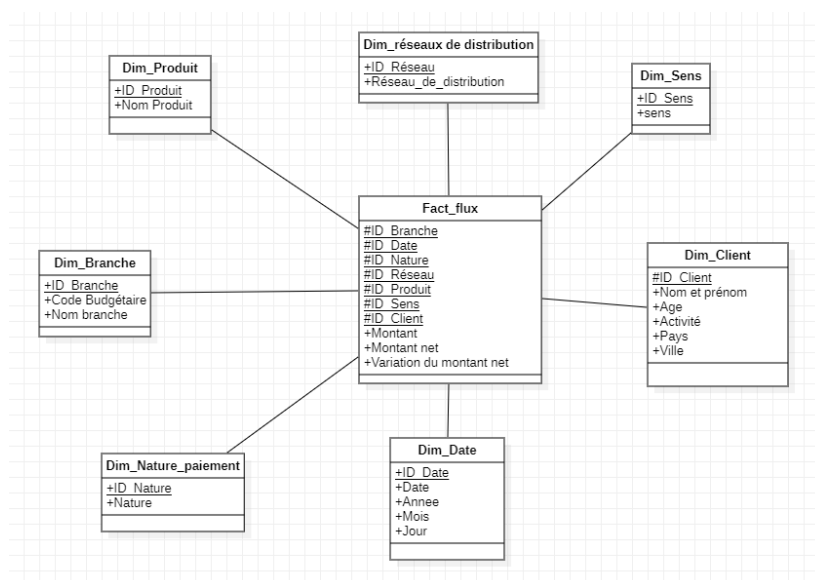


Figure 3.19 – Modèle conceptuel de notre entrepôt de données

3.6 Intégration de l'entrepôt de données

Dans cette section, nous aborderons le processus ETL (Extraction, Transformation et Chargement) ainsi que les techniques de nettoyage des données. Nous nous pencherons également sur l'automatisation de ces processus pour améliorer l'efficacité et la précision de la gestion des données.

3.6.1 Chargement des dimensions

La transformation des données est une étape clé du processus ETL, essentielle pour préparer les données brutes à une analyse efficace.

Chargement de la dimension "DIM_PRODUIT"

Le job "DIM_PRODUIT" ci-dessous représente le processus ETL utilisé pour charger la dimension produit dans l'entrepôt de données.

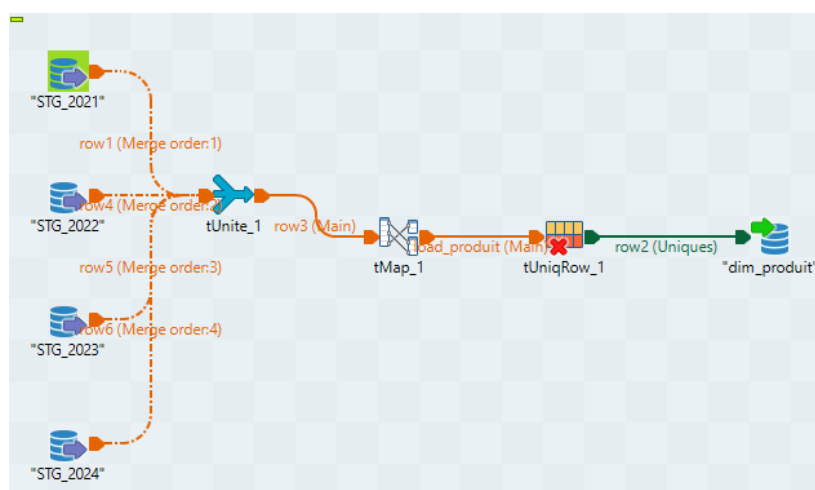


Figure 3.20 – Job "DIM_PRODUIT"

Cette image illustre la configuration du composant UniqRow, qui est essentiel pour garantir l'unicité des enregistrements dans la table "dim_produit"

Clé unique		
Column	<input type="checkbox"/> Attribut de clé	<input type="checkbox"/> Sensible à la casse
id_produit	<input type="checkbox"/>	<input type="checkbox"/>
nom_produit	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figure 3.21 – Fenêtre de tUniqRow "DIM_PRODUIT"

Cette figure montre l'utilisation de la fonction "Numeric.sequence" pour créer une séquence numérique comme étant une clé primaire autoincrément .

```
Numeric.sequence("s1",1,1)
```

Figure 3.22 – Élaboration d'une séquence

La figure ci-dessous illustre une expression utilisée pour gérer les données manquantes dans la dimension produit

```
row3.PRODUIT == null ? ("produit" + row3.BRANCHE) :  
row3.PRODUIT
```

Figure 3.23 – Expression remplacement des données manquantes pour la "DIM_PRODUIT"

Chargement de la dimension "DIM_Client"

Le job "DIM_CLIENT" ci-dessous représente le processus ETL utilisé pour charger la dimension client dans l'entrepôt de données.

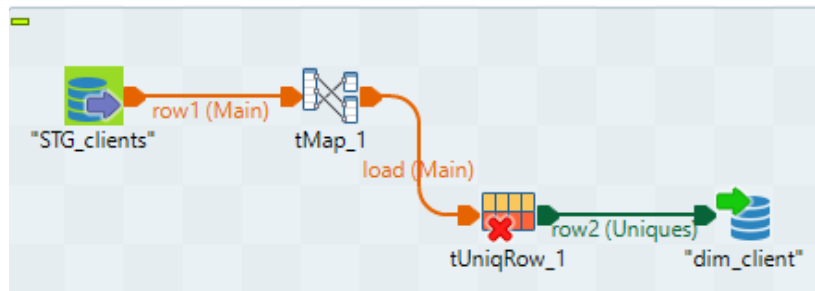


Figure 3.24 – Job "DIM_Client"

La figure ci-jointe montre l'expression utilisée pour calculer l'âge du client en années en utilisant la différence entre la date actuelle et la date de naissance du client.

```
TalendDate.diffDate (TalendDate.getCurrentDate (),
row1.DATE_DE_NAISSANCE, "yyyy")
```

Figure 3.25 – Expression âge client de la dimension "DIM_Client"

Chargement de la dimension "DIM_NATURE"

Cette image montre le job ETL "DIM_NATURE", qui charge la dimension nature. Il extrait les données de la source, les transforme via tMap, et les charge dans la table cible "dim_nature" après avoir vérifié l'unicité avec tUniqRow.

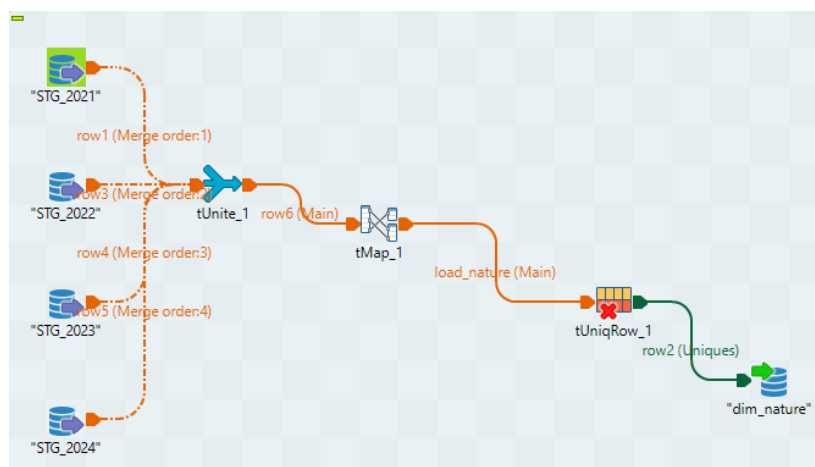


Figure 3.26 – Job "DIM_NATURE"

La figure ci-jointe montre l'expression utilisée pour assurer la normalisation des nature de paiement dans la dimension nature .

```
row6.NATURE_FLUX.equals("Virement Recu") ? "Virement" :  
(row6.NATURE_FLUX.equals("R-PRLV PA") ? "Prélèvement  
automatique" : row6.NATURE_FLUX)
```

Figure 3.27 – Expression de normalisation "DIM_NATURE"

Chargement de la dimension "DIM_BRANCHE" :

Cette image montre le job ETL "DIM_BRANCHE" pour le chargement de la dimension branche. Il commence par extraire les données de différentes sources , les transforme via tMap, puis vérifie l'unicité avec tUniqRow avant de charger les données transformées dans la table cible dim_branche.

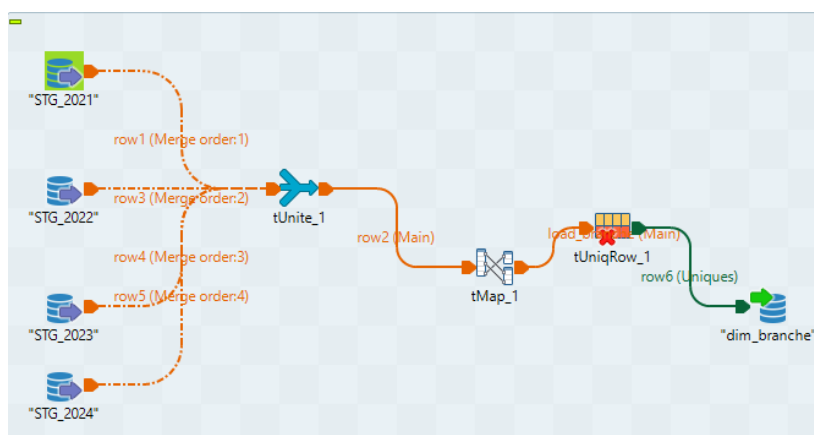


Figure 3.28 – Job "DIM_BRANCHE"

Chargement de la dimension "DIM_DATE"

Cette image montre le job ETL "DIM_DATE" pour le chargement de la dimension date. Il extrait les données des sources, les transforme via tMap, puis vérifie l'unicité avec tUniqRow avant de charger les données transformées et ordonnées dans la table cible dim_date.

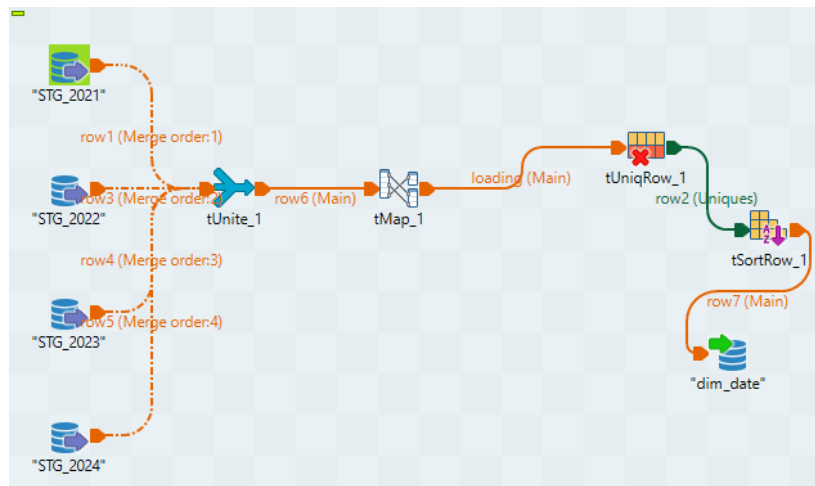


Figure 3.29 – Job "DIM_DATE"

La figure ci-jointe montre l'expression utilisée pour extraire l'année de la date d'opération .

```
String.valueOf(TalendDate.getPartOfDate("yyyy",
row1.DATE_OPERATION)) |
```

Figure 3.30 – Expression année "DIM_DATE"

La figure ci-dessous illustre l'expression utilisée pour extraire le mois de la date d'opération .

```
String.valueOf(TalendDate.getPartOfDate("MM",
row1.DATE_OPERATION)) |
```

Figure 3.31 – Expression mois "DIM_DATE"

Cette figure montre l'expression utilisée pour extraire le jour de la date d'opération .

```
String.valueOf(TalendDate.getPartOfDate("dd",
row1.DATE_OPERATION)) |
```

Figure 3.32 – Expression jour "DIM_DATE"

Chargement de la dimension "DIM_SENS"

Cette image montre le job ETL "DIM_SENS" pour le chargement de la dimension sens. Il extrait les données des sources et les transforme via tMap, puis vérifie l'unicité avec tUniqRow avant de charger les données transformées dans la table cible "DIM_SENS".

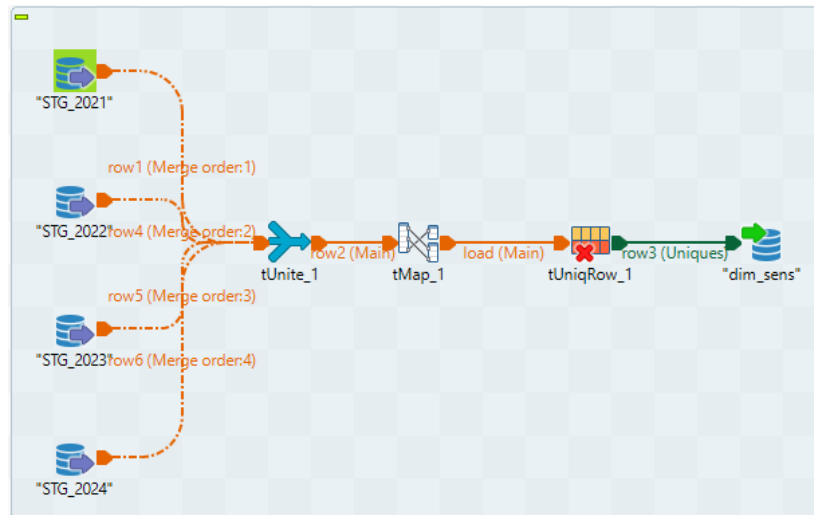


Figure 3.33 – Job "DIM_SENS"

Chargement de la dimension "DIM_RESEAUX"

Cette image montre le job ETL "DIM_RESEAUX" utilisé pour charger la dimension des réseaux. Le processus commence par l'extraction des données à partir des sources, suivie de leur transformation via tMap. L'unicité des enregistrements est vérifiée avec tUniqRow avant de charger les données dans la table cible.

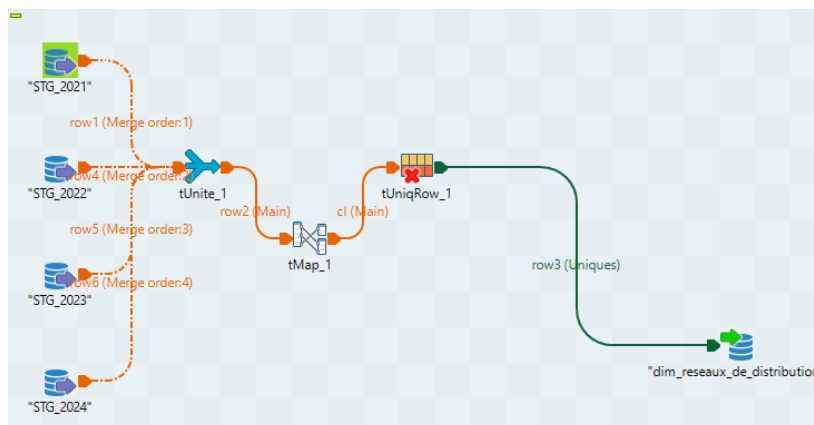


Figure 3.34 – Job "DIM_RESEAUX"

3.6.2 Chargement de la table de fait

Cette image montre le job ETL "FAIT_FLUX" pour le chargement de la table de faits des flux. Le processus commence par l'extraction des données à partir de plusieurs sources, suivie de leur transformation via divers composants comme tMap. Les données transformées sont ensuite chargées dans la table de faits "FAIT_FLUX", assurant ainsi la consolidation des informations nécessaires pour l'analyse.

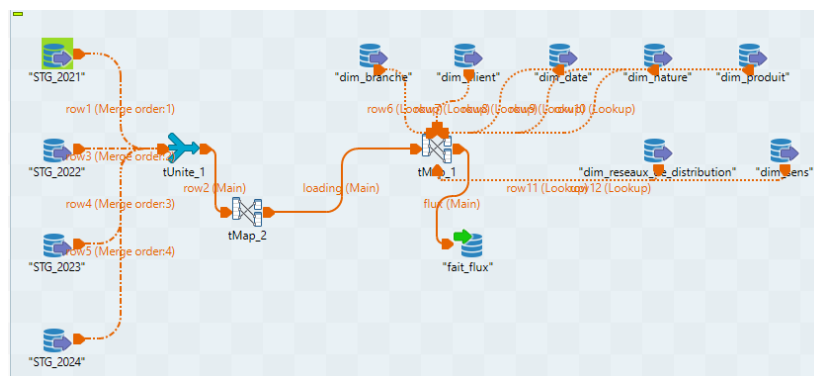


Figure 3.35 – job "FAIT_FLUX"

La figure ci-dessous illustre le composant Tmap utilisé pour le mappage des colonnes des données sources vers celles de la table cible fait_flux

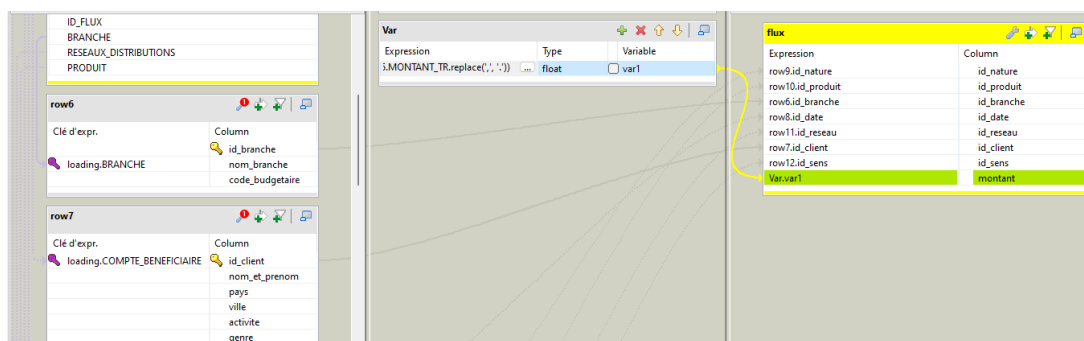


Figure 3.36 – tMap "FAIT_FLUX"

La figure ci-jointe montre l'expression utilisée pour convertir les montants en format flottant pour les calculs financiers.

```
Float.parseFloat(row5.MONTANT TR.replace(',', '.'))
```

Figure 3.37 – Expression de conversion montant "FAIT_FLUX"

3.6.3 Automatisation du processus ETL

Nous avons programmé et automatisé notre processus ETL pour assurer un fonctionnement quotidien automatique. Les figures illustrent la construction du job, la configuration du planificateur des tâches et la spécification des déclencheurs pour garantir une exécution régulière et autonome, assurant ainsi l'intégrité et la mise à jour continue des données.

Cette image montre le job d'exécution pour automatiser le processus ETL. Elle illustre comment les différentes dimensions (DIM_Branche, DIM_Client, DIM_Nature, etc.) et la table de faits "FAIT_FLUX" sont intégrées dans un flux automatisé, garantissant l'ordonnancement et l'exécution séquentielle des différents composants ETL.

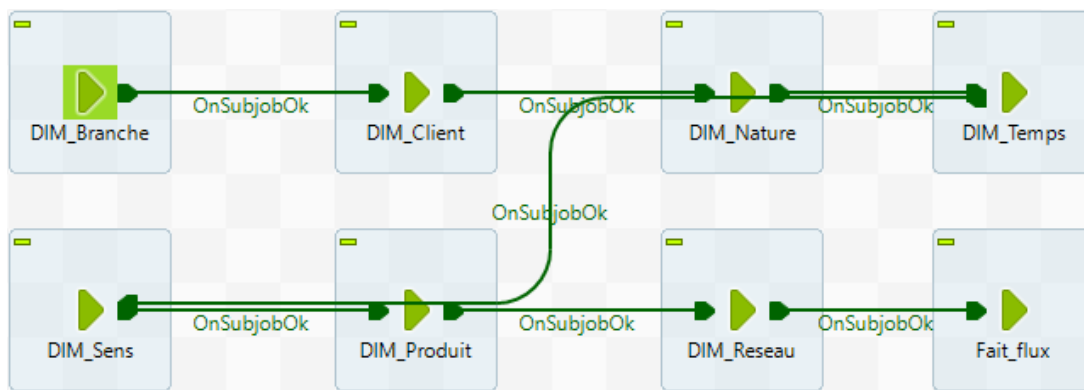


Figure 3.38 – Job d'exécution processus ETL

Et puis on passe à la configuration pour la construction du job d'automatisation du processus ETL. Les options incluent la sélection des paramètres de construction, tels que le dossier de destination et l'application des valeurs des paramètres, pour garantir que le job soit configuré correctement avant l'exécution.

Le planificateur des tâches est utilisé pour automatiser l'exécution du processus ETL qui permet de configurer les options de sécurité, les déclencheurs et les conditions d'exécution, assurant que le job ETL s'exécute de manière autonome et sécurisée.

Nous avons programmé et automatisé notre processus ETL pour assurer un fonctionnement quotidien automatique.

Dans notre cas, nous avons opté pour une exécution quotidienne à 20h, garantissant ainsi l'intégration des nouvelles données et permettant également l'exécution manuelle si nécessaire, assurant ainsi la continuité et l'intégrité des données. comme montre la figure suivante :

Lorsque vous créez une tâche, vous pouvez spécifier les conditions qui la déclenchent. Pour modifier ces déclencheurs, ouvrez les pages de propriétés de la tâche à l'aide de la commande Propriétés.

Déclenchement	Détails	Statut	
Tous les jours	À 20:00 tous les jours	Activé	

Figure 3.39 – Jour et heure de l'automatisation

Conclusion :

En résumé, la préparation minutieuse des données et la modélisation efficace du Data Warehouse sont essentielles pour établir un système décisionnel robuste. En analysant la structure et la qualité des données, nous avons jeté les bases d'un entrepôt fiable. Ces pratiques améliorent la capacité des entreprises à prendre des décisions stratégiques basées sur des données précises et pertinentes.

Chapitre 4: Restitution des données

Contents

4.1	Sprint Backlog	50
4.2	Intégration des données	51
4.3	Création des mesures	51
4.3.1	Mesure du montant net	51
4.3.2	Variation des montants nets	52
4.3.3	Regroupement des tranches d'âges	52
4.4	Interprétation des tableaux de bord	53
4.4.1	Tableau de bord global	53
4.4.2	Tableau de bord des encaissements	55
4.4.3	Tableau de bord des décaissements	57
4.4.4	Tableau de bord des clients	59

Introduction

La restitution des données est une étape cruciale dans le processus de gestion et d'analyse des informations. Elle consiste à transformer des données brutes en informations compréhensibles et exploitables pour les décideurs et les parties prenantes. Ce chapitre se concentre sur l'intégration des données et le développement des tableaux de bord, outils essentiels pour une visualisation claire et efficace des informations.

4.1 Sprint Backlog

Le sprint backlog de ces releases est le suivant :

Table 4.1 – Sprint Backlog : Restitution des données

Sprint	User Story	Scrum Task	Priorité
Restitution des données	En tant que développeur, je souhaite mettre en place des tableaux de bord dynamiques afin d'aider les décideurs dans leurs prises de décision.	— Intégration des données — Création des tableaux de bord	— Haute

4.2 Intégration des données

Nous allons lier Power BI avec notre entrepôt de données en établissant une connexion sécurisée et en chargeant les tables et vues nécessaires. Après l'importation, nous modéliserons les données et créerons des relations entre les tables. Enfin, nous concevrons des visualisations interactives.

Nous commençons par établir la connexion entre Power BI et notre base de données PostgreSQL, comme illustré à la Figure suivante :



Figure 4.1 – Établissement de la connexion entre POWER BI et l'entrepôt de données

4.3 Création des mesures

Power BI offre DAX, un langage qui nous permet de créer des mesures pour effectuer des calculs avancés et obtenir des insights significatifs à partir des données. Dans cette section, nous détaillerons les mesures que nous avons créées.

4.3.1 Mesure du montant net

Nous avons commencé par la première mesure, qui est le montant net. Cette mesure représente la différence entre le montant des encaissements et le montant des décaissements.

```
Montant Net =
SUMX(
    'public fait_flux',
    SWITCH(
        TRUE(),
        RELATED('public dim_sens'[sens]) = "C", 'public fait_flux'[montant], // Encaissement
        RELATED('public dim_sens'[sens]) = "D", -'public fait_flux'[montant], // Décaissement
        0
    )
)
```

Figure 4.2 – Expression de la mesure Montant Net

4.3.2 Variation des montants nets

Pour la mesure des variations des montants nets, nous procédons en trois étapes. Tout d'abord, nous calculons le montant net actuel, comme abordé dans la partie "Montant Net". Ensuite, nous calculons le montant net précédent.

- Calcul du montant net précédent

Le calcul se fait comme illustré dans la figure suivante.

```
Flux Net Précédent =  
CALCULATE(  
    [Montant Net],  
    DATEADD('public dim_date'[Date], -1, MONTH) // Ajustez en fonction de votre période d'analyse, ici mensuelle  
)
```

Figure 4.3 – Expression de la mesure Montant net précédent

- Calcul des variations des montants nets

Enfin, nous calculons les variations des montants nets. comme suit :

```
Variation des Flux de Trésorerie =  
IF(  
    [Flux Net Précédent] <> 0,  
    ([Montant Net] - [Flux Net Précédent]) / [Flux Net Précédent] * 100,  
    BLANK()  
)
```

Figure 4.4 – Expression de la mesure variation des montants net

4.3.3 Regroupement des tranches d'âges

nous abordons la création de regroupements de tranches d'âges dans Power BI. Le regroupement permet de segmenter les données en catégories définies, facilitant l'analyse entre différentes cohortes d'âge. Nous avons créé un regroupement des tranches d'âge avec un intervalle de 10 ans pour chacun.

4.4 Interprétation des tableaux de bord

Nous allons maintenant commencer la partie de l'interprétation des tableaux de bord. Dans cette section, nous présenterons chaque tableau de bord, en expliquant chaque visuel et filtre utilisé.

4.4.1 Tableau de bord global

Nous débutons avec le tableau de bord offrant une vision globale. Ce tableau de bord est conçu pour fournir un aperçu général sur les encaissements et les décaissements au cours de ses années.

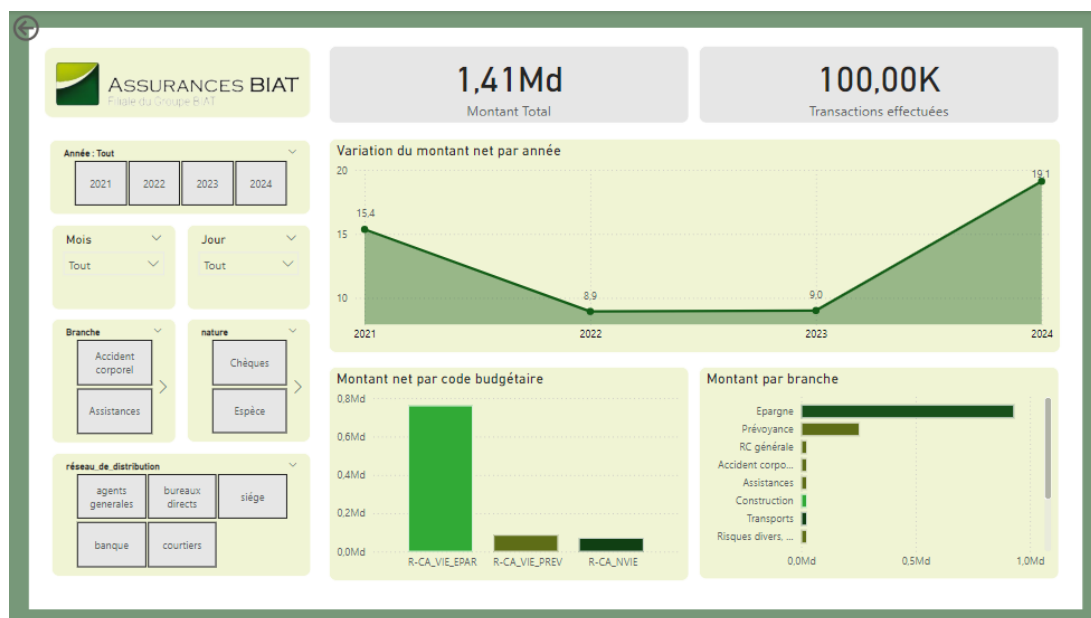


Figure 4.5 – Tableau de bord de vue global

Ce tableau de bord présente une vision globale des performances de l'entreprise "Assurances BIAT". Il est composé de plusieurs visuels et filtres interactifs, permettant une analyse approfondie des données.

Pour en faire nous avons utilisé les visuels suivants :

- **Carte de synthèse**

- **Analyse** : Ce visuel permet de saisir en un coup d'œil le volume financier traité par l'entreprise et le montant total des transactions.

- **Graphique en aires**

- **Analyse** : Permet de montrer l'évolution du montant net par année afin d'identifier les tendances sur plusieurs années et de détecter les variations significatives dans les performances financières.

- **Histogramme empilé**

- **Analyse** : Permet de représenter le montant net réparti par différents codes budgétaires afin de comparer les performances financières entre différentes catégories budgétaires, aidant à identifier les domaines les plus rentables.

- **Diagramme en barres verticales**

- **Analyse souhaitée** : Aide à Afficher le montant net par différentes branches d'activité pour identifier quelles branches contribuent le plus ou le moins aux résultats globaux, permettant de cibler les domaines nécessitant des améliorations ou des investissements.

Pour en faire nous avons établis les filtres suivants :

- **Année**

- **Type de filtre** : Une vignette.
- **Utilisation** : Permet de filtrer les données par année.

- **Mois**

- **Type de filtre** : Liste déroulante.
- **Utilisation** : Permet de filtrer les données par mois.

- **Jour**

- **Type de filtre** : Liste déroulante.
- **Utilisation** : Permet de filtrer les données par jour.

- **Branche**

- **Type de filtre** : Une vignette
- **Utilisation** : Permet de filtrer les données par branche d'activité.

- **Nature**

- **Type de filtre** : Une vignette
- **Utilisation** : Permet de filtrer les données par nature de transaction.

- **Réseau de distribution**

- **Type de filtre** : Liste déroulante
- **Utilisation** : Permet de filtrer les données par réseau de distribution.

4.4.2 Tableau de bord des encaissements

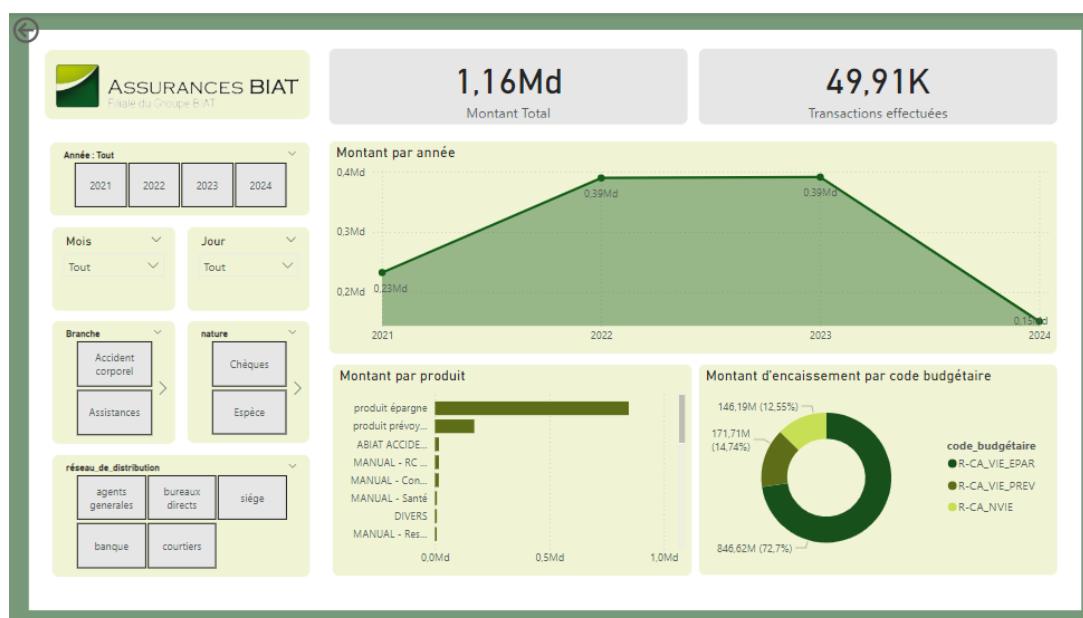


Figure 4.6 – Tableau de bord des encaissements

— Carte de synthèse

- **Analyse :** Ce visuel permet de saisir en un coup d'œil le montant total des encaissements (1,16 milliard) et le nombre total de transactions effectuées (49,91K).

— Graphique en aires

- **Analyse :** Permet de montrer l'évolution du montant des encaissements par année afin d'identifier les tendances sur plusieurs années et de détecter les variations significatives dans les performances financières.

— Diagramme en barres

- **Analyse :** Permet de représenter le montant des encaissements répartis par différents produits afin de comparer les performances financières entre différents produits, aidant à identifier les domaines les plus rentables.

— Diagramme en anneau

- **Analyse :** Permet de visualiser la répartition du montant des encaissements par code budgétaire, aidant à identifier les codes budgétaires les plus significatifs.

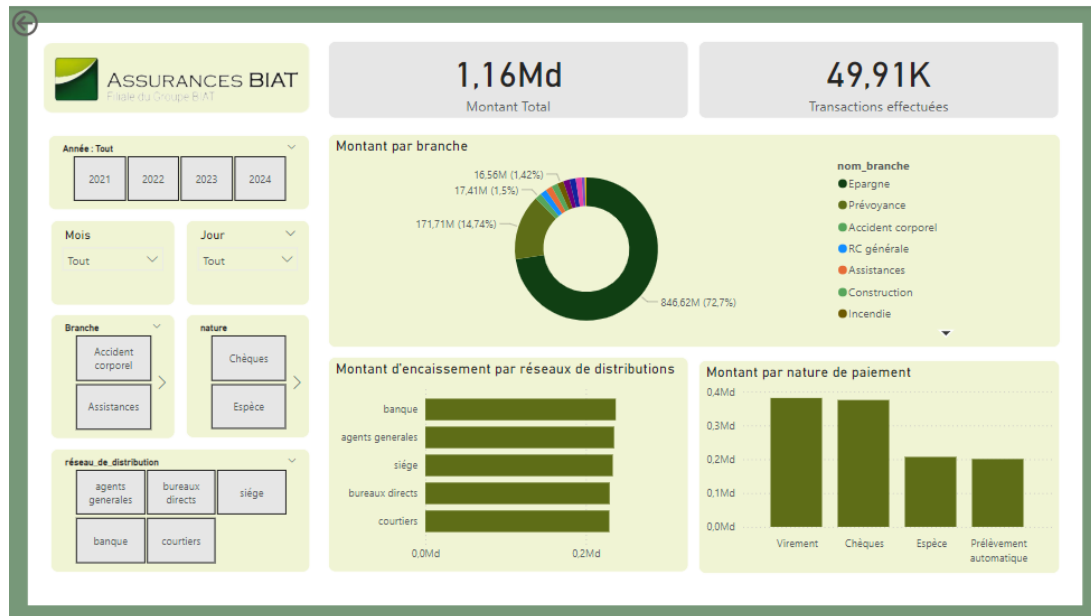


Figure 4.7 – Tableau de bord des encaissements

— Carte de synthèse

- **Analyse :** Ce visuel permet de saisir en un coup d'œil le montant total des encaissements (1,16 milliard) et le nombre total de transactions effectuées (49,91K).

— Diagramme en anneau

- **Analyse :** Permet de visualiser la répartition du montant des encaissements par branche, aidant à identifier les branches les plus significatives.

— Diagramme en barres horizontales

- **Analyse :** Permet de représenter le montant des encaissements répartis par réseaux de distribution afin de comparer les performances financières entre différents réseaux.

— Histogramme empilé

- **Analyse :** Permet de visualiser la répartition du montant des encaissements par nature de paiement, aidant à identifier les modes de paiement les plus utilisés.

4.4.3 Tableau de bord des décaissements

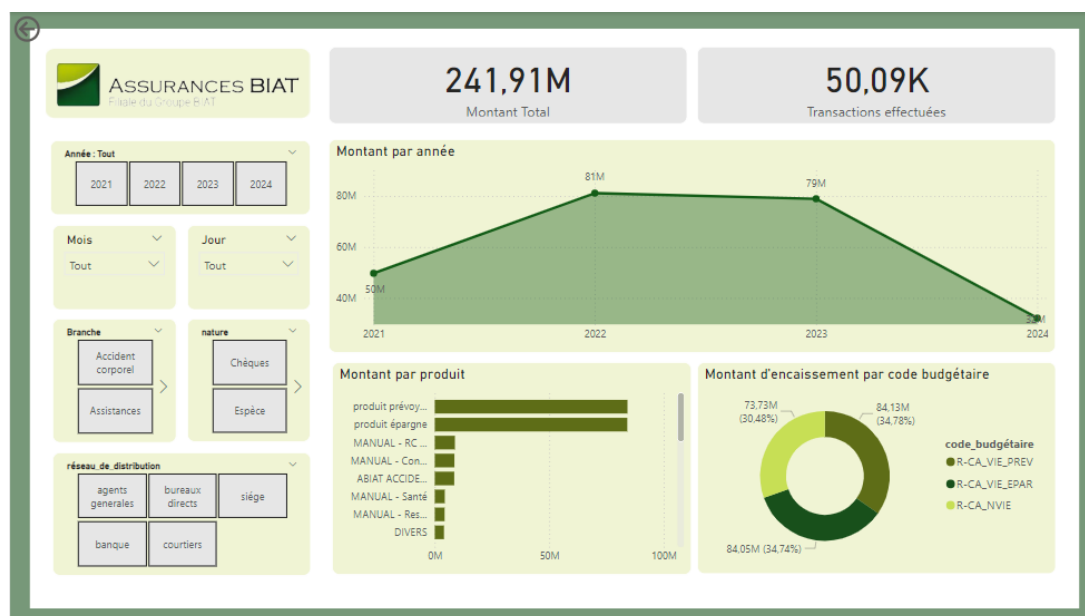


Figure 4.8 – Tableau de bord des décaissements

— Carte de synthèse

- **Analyse :** Ce visuel permet de saisir en un coup d'œil le montant total des décaissements (241,91M) et le nombre total de transactions effectuées (50,09K).

— Graphique en aires

- **Analyse :** Permet de montrer l'évolution du montant des décaissements par année afin d'identifier les tendances sur plusieurs années et de détecter les variations significatives dans les performances financières.

— Diagramme en barres

- **Analyse :** Permet de représenter le montant des décaissements répartis par différents produits afin de comparer les performances financières entre différents produits, aidant à identifier les domaines les plus rentables.

— Diagramme en anneau

- **Analyse :** Permet de visualiser la répartition du montant des décaissements par code budgétaire, aidant à identifier les codes budgétaires les plus significatifs.

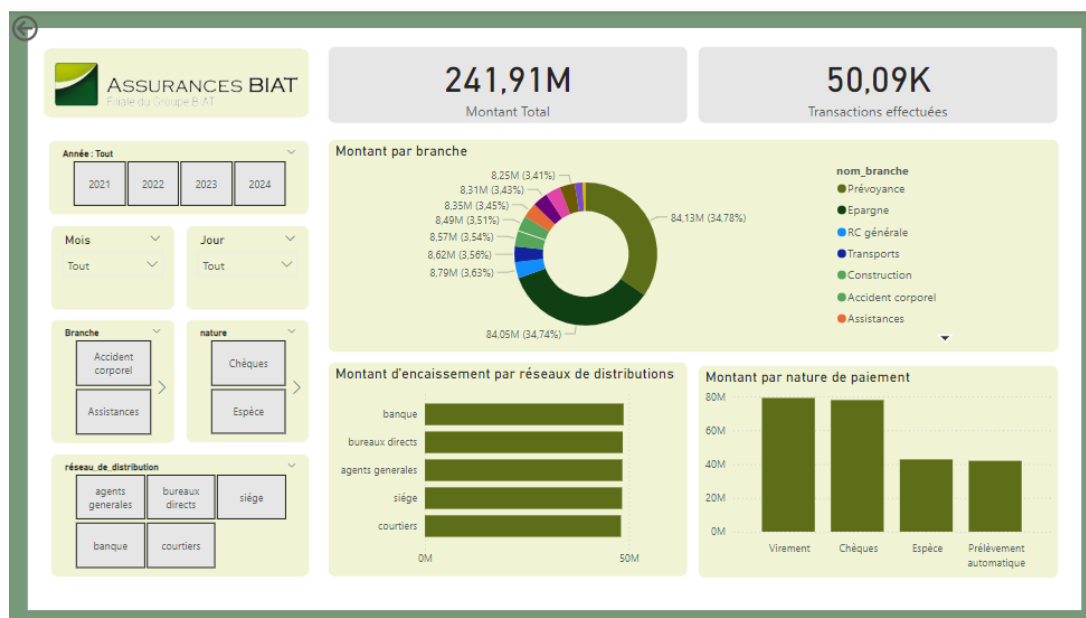


Figure 4.9 – Tableau de bord des décaissements

— Carte de synthèse

- **Analyse :** Ce visuel permet de saisir en un coup d'œil le montant total des décaissements (241,91M) et le nombre total de transactions effectuées (50,09K).

— Diagramme en anneau

- **Analyse :** Permet de visualiser la répartition du montant des décaissements par branche, aidant à identifier les branches les plus significatives.

— Diagramme en barres horizontales

- **Analyse :** Permet de représenter le montant des décaissements répartis par réseaux de distribution afin de comparer les performances financières entre différents réseaux.

— Histogramme empilé

- **Analyse :** Permet de visualiser la répartition du montant des décaissements par nature de paiement, aidant à identifier les modes de paiement les plus utilisés.

4.4.4 Tableau de bord des clients

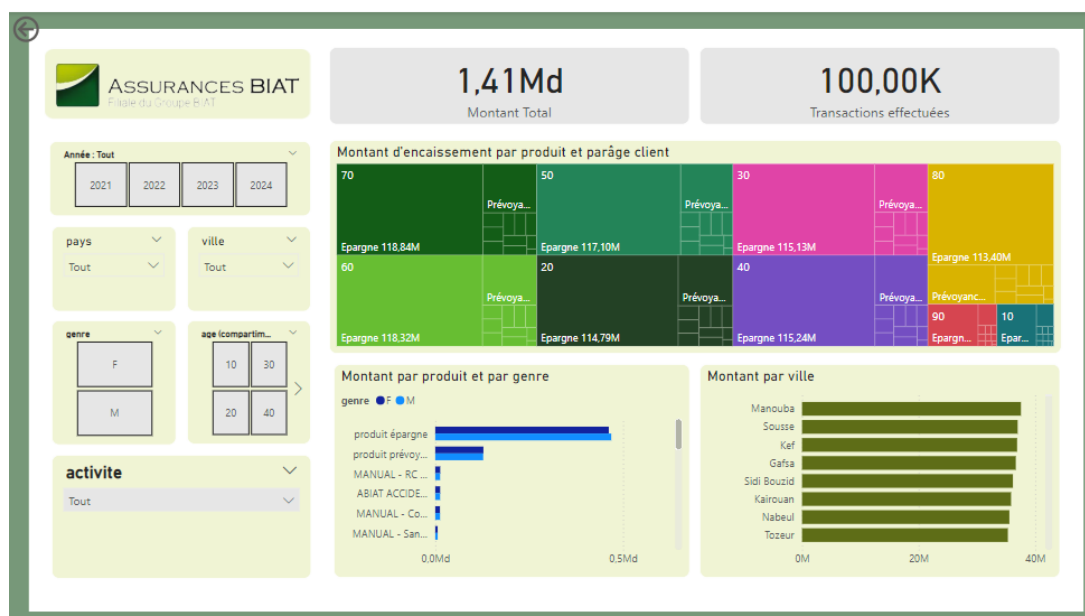


Figure 4.10 – Tableau de bord des clients

— Carte de synthèse

- **Analyse** : Ce visuel permet de saisir en un coup d'œil le montant total (1,41 milliard) et le nombre total de transactions effectuées (100,00K).

— Treemap

- **Analyse** : Permet de visualiser la répartition du montant des encaissements par produit et par âge client, aidant à identifier les produits les plus significatifs et les segments de clients les plus importants.

— Diagramme en barres

- **Analyse** : Permet de représenter le montant des encaissements répartis par produit et par genre afin de comparer les performances financières entre différents produits et genres.

— Diagramme en barres horizontales

- **Analyse** : Permet de visualiser la répartition du montant des encaissements par ville, aidant à identifier les villes les plus significatives en termes d'encaissements.

Filtres :

— Année

- **Type de filtre** : Une vignette.

- **Utilisation** : Permet de filtrer les données par année.
- **Pays**
 - **Type de filtre** : Liste déroulante.
 - **Utilisation** : Permet de filtrer les données par pays.
- **Ville**
 - **Type de filtre** : Liste déroulante.
 - **Utilisation** : Permet de filtrer les données par ville.
- **Genre**
 - **Type de filtre** : Une vignette.
 - **Utilisation** : Permet de filtrer les données par genre (Féminin ou Masculin).
- **Âge (compartiment)**
 - **Type de filtre** : Une vignette.
 - **Utilisation** : Permet de filtrer les données par tranches d'âge (10, 30, 20, 40).
- **Activité**
 - **Type de filtre** : Liste déroulante.
 - **Utilisation** : Permet de filtrer les données par activité.

Conclusion

Ce chapitre a traité de l'intégration des données, de la création de mesures et de l'interprétation des tableaux de bord pour calculer les indicateurs clés de performance (KPIs). En conclusion, nous avons interprété ces KPIs à l'aide de tableaux de bord, ce qui facilite l'analyse et la prise de décisions stratégiques.

Chapitre 5: Développement d'un Chatbot et Déploiement d'une Application Web

Contents

5.1	Sprint Backlog	61
5.2	Le processus de la création du Chatbot	62
5.2.1	Présentation de FlowBot	62
5.2.2	Fonctionnalités de FlowBot	63
5.3	Le processus du développement de l'application Web	67
5.3.1	Présentation de l'application Web Arima Insight	67
5.3.2	Fonctionnalités d'Arima Insight	68

Introduction

Dans ce chapitre, nous allons en premier lieu explorer le processus de développement d'un chatbot, en détaillant les différentes étapes et composants nécessaires à sa création. Par la suite, nous aborderons le déploiement de l'application web, en mettant en lumière les techniques et les outils utilisés pour rendre le chatbot accessible aux utilisateurs finaux.

5.1 Sprint Backlog

La table suivante représente le sprint backlog pour notre chapitre développement du chatbot et déploiement d'une application web.

Table 5.1 – Sprint Backlog : Développement du Chatbot et Application Web

Sprint	User Story	Scrum Task	Priorité
Sprint 1	En tant qu'utilisateur, je veux que le chatbot puisse répondre à mes questions sur la base de données	Concevoir le FlowBot Développer les fonctionnalités Intégrer le chatbot avec MySQL Tester les réponses	Haute
	En tant qu'utilisateur, je veux signaler un problème et obtenir une assistance technique via le chatbot	Concevoir des diagrammes Développer la NLU Développer les actions et règles Tester l'assistance technique	Haute

Table 5.1 – suite

Sprint	User Story	Scrum Task	Priorité
Sprint 2	En tant qu'utilisateur, je veux pouvoir m'authentifier, réinitialiser mon mot de passe, me déconnecter et modifier mon profil sur l'application web	Concevoir et développer l'authentification Concevoir et développer la réinitialisation du mot de passe Concevoir et développer la déconnexion Concevoir et développer la modification de profil Tester	Moyenne
	En tant qu'administrateur, je veux pouvoir ajouter et supprimer des utilisateurs	Concevoir et développer l'ajout d'utilisateur Concevoir et développer la suppression de profil Tester	Haute

5.2 Le processus de la création du Chatbot

Dans cette section, nous allons parler de FlowBot, détailler ses fonctionnalités principales et fournir des diagrammes de séquence accompagnés d'explications pour illustrer son fonctionnement.

5.2.1 Présentation de FlowBot

FlowBot est un assistant intelligent capable de répondre à diverses questions spécifiques liées à notre projet. Il est conçu pour fournir des réponses précises et rapides à des questions telles que le montant total des encaissements et la répartition du montant total par branche. De plus, FlowBot offre une assistance technique en temps réel, en collectant et en enregistrant les problèmes signalés par les utilisateurs dans une base de données MySQL dédiée.



Figure 5.1 – Logo FlowBot

5.2.2 Fonctionnalités de FlowBot

Nous allons présenter les deux fonctionnalités principales de FlowBot, en abordant leur conception, le coding et leur réalisation. Les deux fonctionnalités principales sont :

- Poser des questions : L'utilisateur peut interagir avec FlowBot pour poser des questions.
- Signaler un problème : L'utilisateur peut utiliser FlowBot pour signaler un problème.

Nous allons commencer par la première fonctionnalité qui est l'assistance technique du chatbot, où il guide les utilisateurs à signaler un problème.

Assistance technique

Cette figure montre un diagramme de séquence pour l'assistance technique :

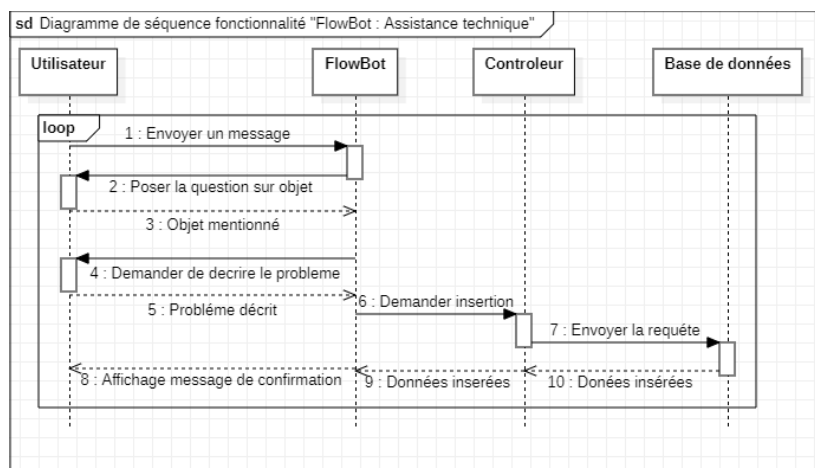


Figure 5.2 – Diagramme de séquence "Assistance technique"

Le tableau ci-dessous résume les rôles, les actions, et les conditions nécessaires pour l'assistance technique, mettant en lumière le processus de signalement de problème via FlowBot :

Table 5.2 – Description Cas d'Utilisation Assistance Technique

Acteur	Scénario	Précondition
Utilisateur	L'utilisateur initie une requête pour signaler un problème. FlowBot demande des détails supplémentaires et enregistre le problème.	L'utilisateur doit avoir un problème à signaler.
FlowBot	Le chatbot traite la requête de l'utilisateur, demande des informations supplémentaires, et enregistre le problème dans la base de données.	FlowBot doit être opérationnel et accessible.
Base de données	Les informations du problème sont enregistrées pour un suivi ultérieur.	La base de données doit être connectée et fonctionnelle.

Cette figure illustre un exemple d'interaction en ligne de commande avec RASA shell pour l'assistance technique. Le chatbot guide l'utilisateur pour signaler un problème, demandant des détails spécifiques et confirmant l'enregistrement du problème.

```

Your input -> Bonjour
Hello! How can I help you?
Your input -> je veux signaler un problème
Ça concerne quel objet ? L'application web, le dashboard ou la modification du profil ?
Your input -> Le dashboard
Quel est votre problème ?
Your input -> Les données ne sont pas actualisées
Merci, votre problème a été enregistré.

```

Figure 5.3 – RASA shell "Assistance technique"

Nous allons maintenant passer au développement du chatbot en montrant des extraits de chaque composante RASA nécessaires pour assurer le fonctionnement de l'exemple que nous avons fourni. La figure suivante montre un extrait de la configuration NLU pour l'assistance technique. Ce composant permet au chatbot de comprendre les intentions de l'utilisateur, telles que signaler un problème, et d'identifier les entités pertinentes :

```

- intent: assistance_technique
  examples: |
    - j'ai besoin d'aide technique
    - j'ai un problème avec l'application
    - assistance technique

- intent: choisir_objet
  examples: |
    - dashboard
    - application web
    - modification du profil

- intent: informer_probleme
  examples: |
    - n'est pas actualisé
    - ne permet pas de se connecter
    - je ne peux pas modifier mes coordonnées

```

Figure 5.4 – Composante NLU "Assistance technique"

Cette figure illustre un extrait des stories de RASA. Les stories sont des exemples de dialogues que le chatbot utilise pour apprendre comment répondre aux différentes intentions de l'utilisateur de manière contextuelle.

```

- story: assistance technique - dashboard not updating
  steps:
  - intent: assistance_technique
  - action: utter_ask_objet
  - intent: choisir_objet
  - entities:
    - objet: "dashboard"
  - action: utter_ask_description
  - intent: informer_probleme
  - entities:
    - description: "n'est pas actualisé"
  - action: action_insert_proposal
  - action: utter_confirm

- story: assistance technique - web app not logging in
  steps:
  - intent: assistance_technique
  - action: utter_ask_objet
  - intent: choisir_objet
  - entities:
    - objet: "application web"
  - action: utter_ask_description
  - intent: informer_probleme
  - entities:
    - description: "ne permet pas de se connecter"
  - action: action_insert_proposal
  - action: utter_confirm
  
```

Figure 5.5 – Composante Stories "Assistance technique"

Nous allons maintenant passer à la deuxième fonctionnalité du chatbot, qui est de répondre aux questions posées par les utilisateurs.

Poser une question

La figure ci-jointe montre le diagramme de séquence pour la fonctionnalité de réponse aux questions.

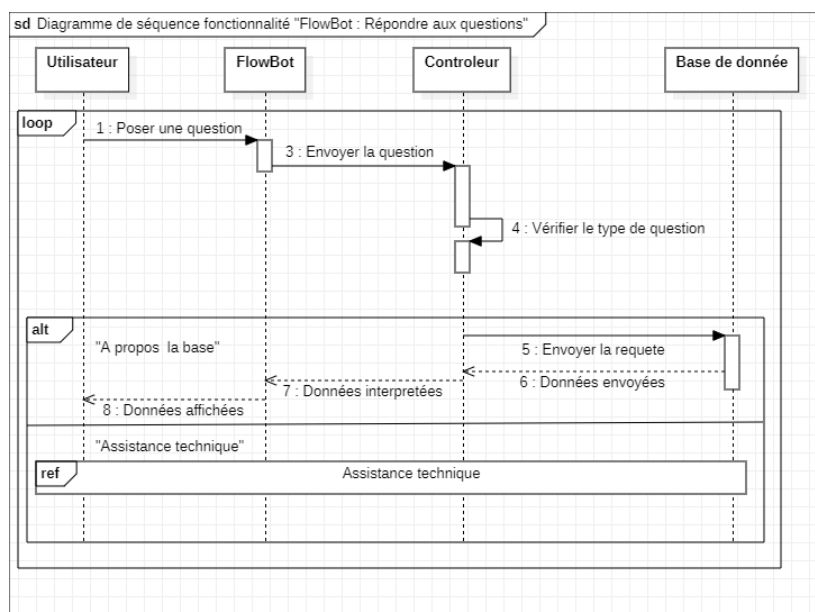


Figure 5.6 – Diagramme de séquence "Répondre aux questions"

Le tableau ci-dessous résume les rôles, les actions, et les conditions nécessaires pour poser des questions :

Table 5.3 – Description Cas d'Utilisation "Poser une question"

Acteur	Scénario	Précondition
Utilisateur	L'utilisateur pose une question à FlowBot. FlowBot traite la question, vérifie les informations dans la base de données, et fournit une réponse.	L'utilisateur doit avoir une question à poser.
FlowBot	Le chatbot traite la question de l'utilisateur, interagit avec la base de données, et fournit une réponse appropriée.	FlowBot doit être opérationnel et accessible.
Base de données	Les informations nécessaires pour répondre aux questions sont stockées et accessibles.	La base de données doit être connectée et fonctionnelle.

Cette figure illustre un exemple d'interaction en ligne de commande avec RASA shell pour la fonctionnalité de réponse aux questions. Le chatbot guide l'utilisateur en répondant aux questions posées.

```

Your input -> Bonjour
Hello! How can I help you?
Your input -> je veux signaler un problème
Ça concerne quel objet ? L'application web, le dashboard ou la modification du profil ?
Your input -> Le dashboard
Quel est votre problème ?
Your input -> Les données ne sont pas actualisées
Merci, votre problème a été enregistré.

```

Figure 5.7 – RASA shell "Répondre aux questions"

Cette figure présente un extrait de la configuration NLU pour la fonctionnalité de réponse aux questions. Ce composant permet au chatbot de comprendre les questions posées par l'utilisateur et d'identifier les entités pertinentes.

```

- intent: top_branches_by_credit_amount
  examples: |
    - montre-moi les branches avec le montant d'encaissement le plus élevé
    - quelles sont les branches avec le montant d'encaissement le plus élevé

```

Figure 5.8 – Composante NLU "Répondre aux questions"

Cette figure montre un extrait des stories de RASA pour la fonctionnalité de réponse aux questions. Les stories sont des exemples de dialogues que le chatbot utilise pour apprendre comment répondre aux différentes intentions de l'utilisateur de manière contextuelle :

```
- story: top branches by credit amount
steps:
- intent: top_branches_by_credit_amount
- slot_was_set:
  - top_n: 5
- action: action_top_branches_by_credit_amount
```

Figure 5.9 – Composante Stories "Répondre aux questions"

5.3 Le processus du développement de l'application Web

Pour l'application web Arima Insight, nous allons présenter la conception et le développement de chaque fonctionnalité. Dans cette section, nous parlerons de FlowBot, détaillerons ses principales fonctionnalités et fournirons des diagrammes de séquence accompagnés d'explications pour illustrer son fonctionnement.

5.3.1 Présentation de l'application Web Arima Insight

Dans cette section, nous présentons l'application web Arima Insight. Cette application se compose de deux volets principaux : un côté administrateur et un côté utilisateur.

Le volet administrateur permet de gérer les profils, d'ajouter et de supprimer des utilisateurs. Le volet utilisateur offre l'accès à des tableaux de bord interactifs, un support technique via un chatbot, et la possibilité de modifier son profil.

Pour accéder à l'application, les administrateurs et les utilisateurs doivent s'authentifier. À la fin de leur session, ils peuvent se déconnecter. En cas d'oubli de mot de passe, une fonctionnalité de récupération de mot de passe est disponible pour faciliter leur accès. Nous explorerons les fonctionnalités et les étapes de mise en œuvre de cette application.



Figure 5.10 – Logo "ARIMA INSIGHT"

5.3.2 Fonctionnalités d'Arima Insight

La figure ci-jointe illustre un diagramme de cas d'utilisation de l'application "ARIMA INSIGHT". Ce diagramme présente les différentes interactions possibles entre les utilisateurs (administrateurs et utilisateurs réguliers) et le système :

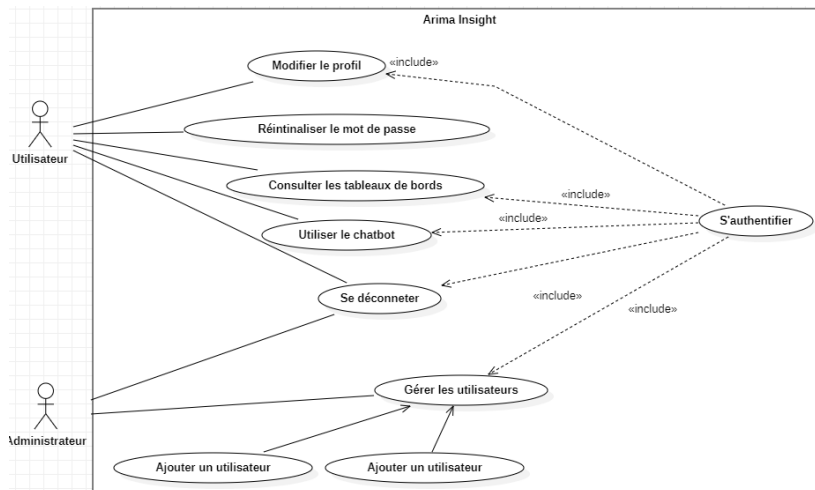


Figure 5.11 – Diagramme de cas d'utilisation "ARIMA INSIGHT"

Deux types d'acteurs sont représentés :

Utilisateur

- S'authentifier
- Réinitialiser le mot de passe si oublié
- Accéder aux tableaux de bord
- Utiliser le chatbot
- Modifier le profil
- Se déconnecter

Administrateur

- S'authentifier
- Réinitialiser le mot de passe si oublié
- Ajouter un utilisateur
- Supprimer un utilisateur
- Se déconnecter

Avant de commencer, nous avons créé une base de données MySQL intitulé "database_pfe" dont elle possède 3 tables .

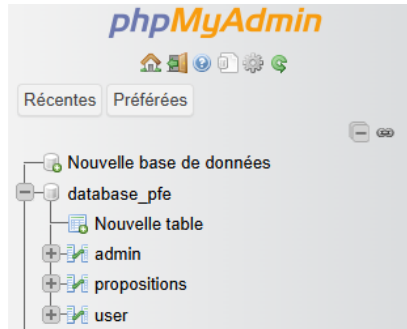


Figure 5.12 – Base de données "database_pfe".

1. admin : qui possède les paramètres de connexion de l'administrateur .
2. user : qui présente la liste des utilisateurs enregistrés dans l'application "ARIMA Insight" .
3. propositions : qui présente les problèmes que les utilisateurs ont signalé à l'aide du chatbot .

Nous allons commencer par la première fonctionnalité qui est l'authentification , une fonctionnalité accessible aux deux acteurs .

Authentification

Le diagramme de séquence montre les interactions entre les différents composants du système durant le processus d'authentification.

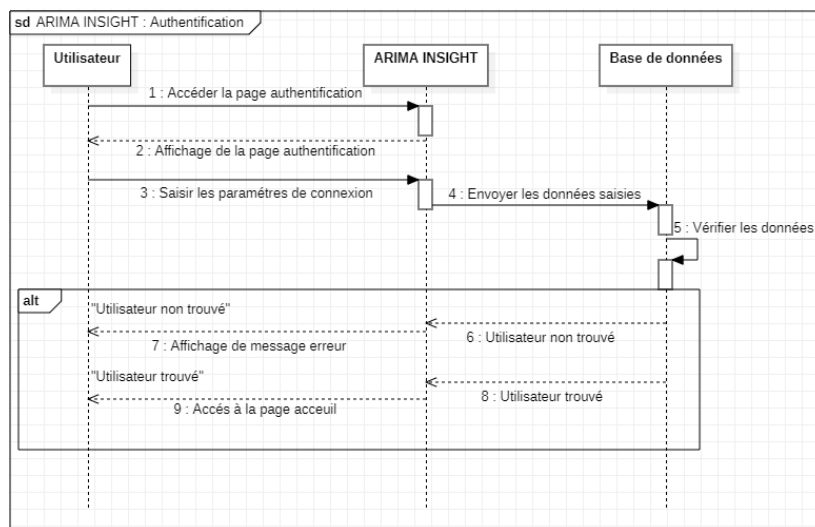


Figure 5.13 – Diagramme de séquence "ARIMA INSIGHT : Authentification"

Ce tableau résume les actions spécifiques de chaque acteur impliqué dans le processus d'authentification de "ARIMA INSIGHT", ainsi que les préconditions nécessaires pour que chaque étape soit réalisée avec succès :

Table 5.4 – Description diagramme de séquence "Arima Insight : Authentification"

Acteur	Scénario	Précondition
Utilisateur	<ol style="list-style-type: none"> 1. Demande la page d'authentification. 2. Saisit le nom d'utilisateur et le mot de passe. 3. Soumet les informations d'identification. 	L'utilisateur doit avoir un compte valide.
Système "ARIMA INSIGHT"	<ol style="list-style-type: none"> 1. Fournit la page d'authentification. 2. Envoie les informations d'identification à la base de données. 3. Affiche la page d'accueil ou un message d'erreur. 	Le système doit être en état de fonctionnement.
Base de données	<ol style="list-style-type: none"> 1. Reçoit les informations d'identification. 2. Vérifie les informations. 3. Retourne le résultat de la validation. 	La base de données doit contenir les comptes utilisateurs.

— Développement :

La figure ci-jointe montre la page d'authentification de "ARIMA INSIGHT". Cette page comprend un champ pour le nom d'utilisateur, un champ pour le mot de passe et un bouton de connexion, tous reliés à une base de données.

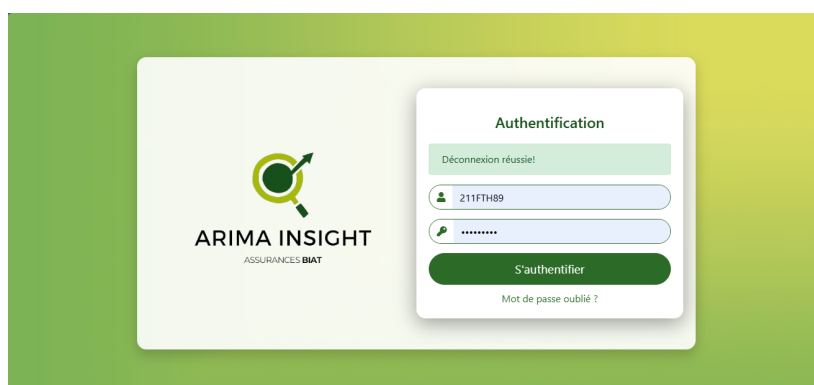


Figure 5.14 – Page d'authentification "ARIMA INSIGHT"

Réinitialisation du mot de passe

Le diagramme de séquence illustré dans la figure 5.15 présente le processus de réinitialisation du mot de passe pour l'application Web "ARIMA INSIGHT". Ce diagramme met en évidence les interactions entre différents acteurs et le système à travers les étapes de réinitialisation.

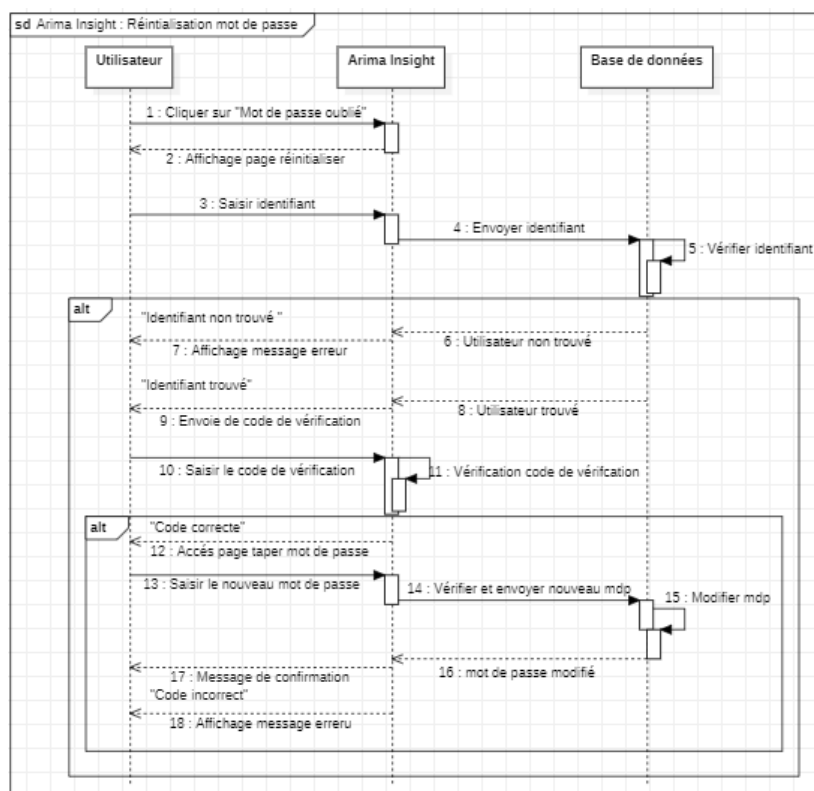


Figure 5.15 – Diagramme de séquence "ARIMA INSIGHT : Réinitialisation"

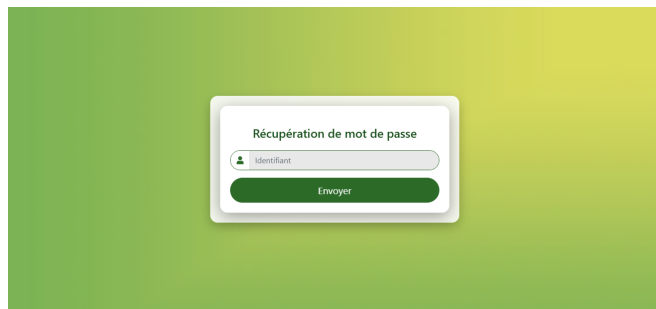
Ce tableau résume les actions spécifiques de chaque acteur impliqué dans le processus de réinitialisation de mot de passe sur "ARIMA INSIGHT", ainsi que les préconditions nécessaires pour que chaque étape soit réalisée avec succès. :

Table 5.5 – Description diagramme de séquence : "Arima Insight : Réinitialisation"


Acteur	Scénario	Pré-condition
Utilisateur	<ol style="list-style-type: none"> 1. Demande la réinitialisation du mot de passe. 2. Fournit l'adresse email associée à son compte. 3. Reçoit un email avec un lien de réinitialisation. 4. Clique sur le lien de réinitialisation. 5. Entre et confirme le nouveau mot de passe. 	L'utilisateur doit être inscrit dans le système.
Système "ARIMA INSIGHT"	<ol style="list-style-type: none"> 1. Envoie un email avec le lien de réinitialisation à l'adresse fournie par l'utilisateur. 2. Affiche la page de réinitialisation du mot de passe lorsque l'utilisateur clique sur le lien. 3. Vérifie et met à jour le mot de passe dans la base de données. 	Le système doit être en état de fonctionnement.
Base de données	<ol style="list-style-type: none"> 1. Stocke les informations de réinitialisation du mot de passe. 2. Met à jour le mot de passe de l'utilisateur. 	La base de données doit contenir les comptes utilisateurs.

Développement

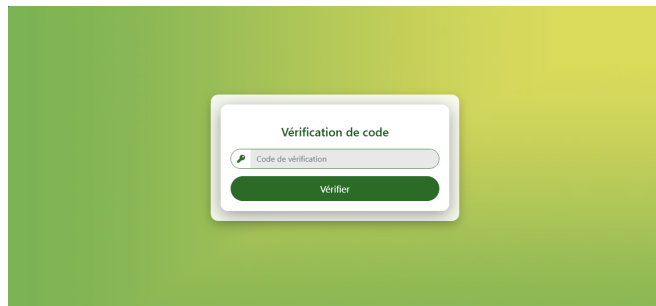
Les images illustrées dans la figure ci-jointe présentent les étapes du processus de récupération et de réinitialisation du mot de passe pour "ARIMA INSIGHT"



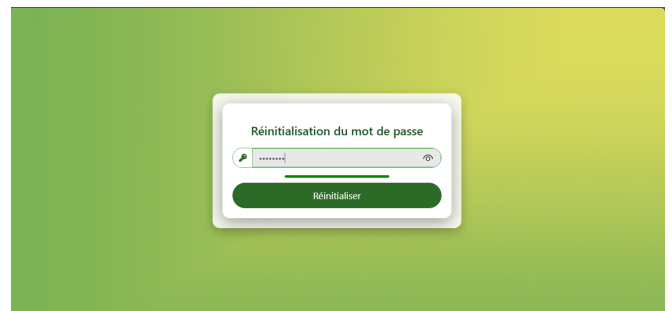
(a) Page de saisie Identifiant "ARIMA INSIGHT"

 **malekfendri2002@gmail.com**
À moi ▾
Votre code de vérification est : 430836

(b) Le mail de code de vérification



(c) Page de saisie de code de vérification "ARIMA INSIGHT"



(d) Page de réinitialisation "ARIMA INSIGHT"

Figure 5.16 – Processus de récupération et de réinitialisation "ARIMA INSIGHT"

L'image (a) montre la page de saisie de l'identifiant, où l'utilisateur entre son nom d'utilisateur. Ensuite, un code de vérification est envoyé à l'utilisateur via SMTP, comme illustré dans l'image (b), qui montre l'email contenant le code. L'image (c) représente la page où l'utilisateur saisit le code de vérification reçu. Enfin, l'image (d) montre la page de réinitialisation du mot de passe, où l'utilisateur peut entrer et confirmer son nouveau mot de passe. Ces étapes garantissent un processus sécurisé de réinitialisation de mot de passe.

Volet Utilisateur

Dans cette section, nous allons commencer par le volet utilisateur, en présentant les fonctionnalités offertes pour l'utilisateur.

Les tableaux de bords

Le diagramme de séquence suivant illustre le scénario d'accès aux dashboards .

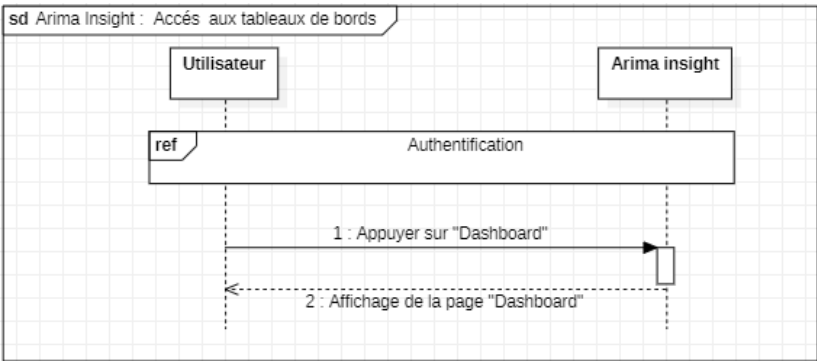


Figure 5.17 – Diagramme de séquence "ARIMA INSIGHT : Accès aux dashboards"

Ce tableau résume les actions spécifiques de chaque acteur impliqué dans le processus d'accès aux tableaux de bords au sein de l'application web "ARIMA INSIGHT" :

Table 5.6 – Description diagramme de séquence "ARIMA INSIGHT : Accès aux dashboards"

Acteur	Scénario	Pré-condition
Utilisateur	<div>1. L'utilisateur appuie sur le bouton "Dashboard".</div> <div>2. L'utilisateur attend que la page s'affiche.</div>	L'utilisateur est connecté à l'application ARIMA INSIGHT.
ARIMA INSIGHT	<div>1. ARIMA INSIGHT reçoit la requête de l'utilisateur.</div> <div>2. ARIMA INSIGHT affiche la page "Dashboard".</div>	Le système est opérationnel et les données sont disponibles.

- Développement :
La figure ci-jointe illustre la page du tableau de bord sur "ARIMA INSIGHT", mettant en évidence les différentes métriques et graphiques utilisés pour analyser et visualiser les données.

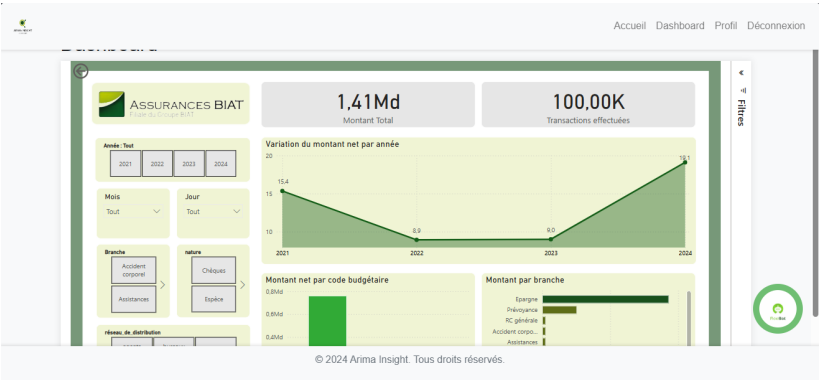


Figure 5.18 – Page dashboard sur "ARIMA INSIGHT"

Déploiement du chatbot

Le diagramme de séquence suivant illustre le scénario d'utilisation de "FlowBot" au sein de l'application web "ARIMA INSIGHT".

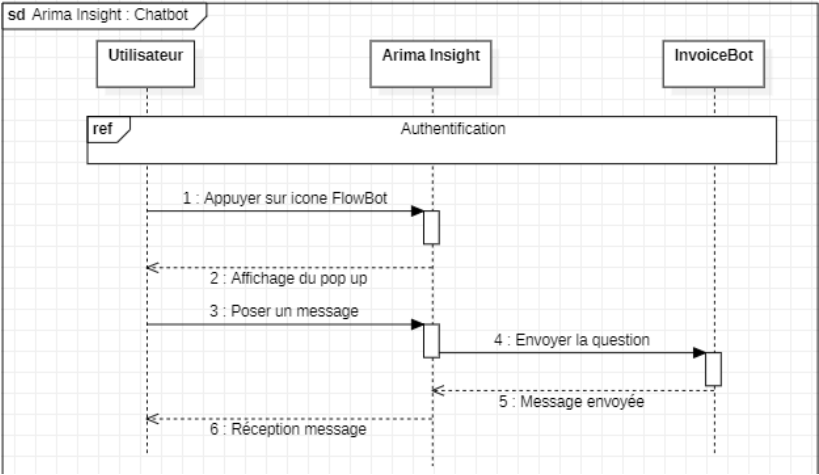


Figure 5.19 – Diagramme de séquence "ARIMA INSIGHT : Chatbot"

Ce tableau résume les actions spécifiques de chaque acteur impliqué dans le processus de l'utilisation du chatbot "FlowBot" sur "ARIMA INSIGHT", ainsi que les pré-conditions nécessaires pour que chaque étape soit réalisée avec succès :

Table 5.7 – Description diagramme de séquence "ARIMA INSIGHT : Utilisation de FlowBot"

Acteur	Scénario	Pré-condition
Utilisateur	<ol style="list-style-type: none"> 1. L'utilisateur appuie sur l'icône "FlowBot". 2. L'utilisateur voit s'afficher un pop-up. 3. L'utilisateur pose un message. 	L'utilisateur est connecté à l'application ARIMA INSIGHT.
ARIMA INSIGHT	<ol style="list-style-type: none"> 1. ARIMA INSIGHT reçoit la requête pour afficher le pop-up. 2. ARIMA INSIGHT affiche le pop-up. 	Le système est opérationnel.
InvoiceBot	<ol style="list-style-type: none"> 1. InvoiceBot reçoit la question posée par l'utilisateur via ARIMA INSIGHT. 2. InvoiceBot envoie le message de réponse à l'utilisateur via ARIMA INSIGHT. 	InvoiceBot est intégré et configuré correctement.

— Développement :

La figure ci-jointe illustre la fenêtre de pop-up de "FlowBot" sur "ARIMA INSIGHT" :

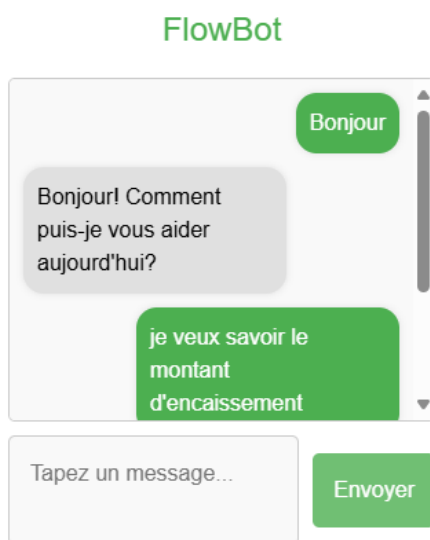


Figure 5.20 – Fenêtre de pop up de "FlowBot" sur "ARIMA INSIGHT"

Modification profil

Le diagramme de séquence suivant illustre le scénario de modification profil sur "ARIMA INSIGHT".

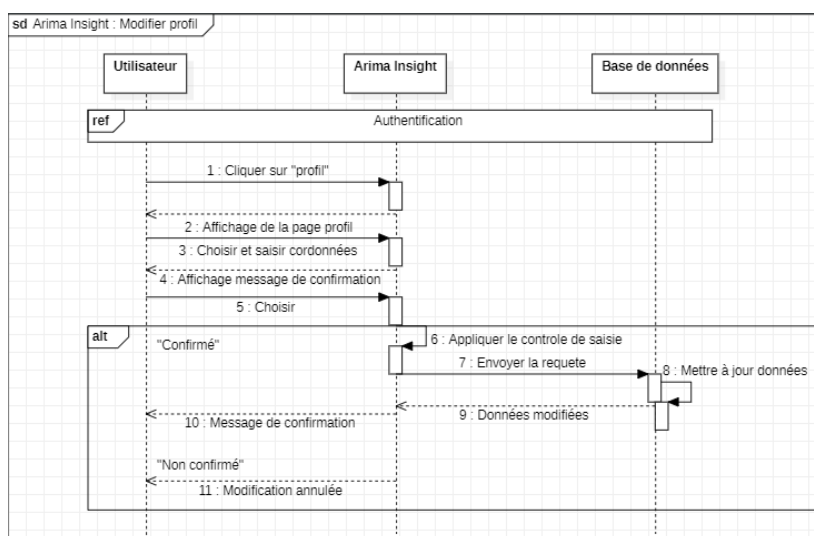


Figure 5.21 – Diagramme de séquence "ARIMA INSIGHT : Modifier profil"

Ce tableau résume les actions spécifiques de chaque acteur impliqué dans le processus de modification de profil sur "ARIMA INSIGHT", ainsi que les pré-conditions nécessaires pour que chaque étape soit réalisée avec succès :

Table 5.8 – Description diagramme de séquence "ARIMA INSIGHT : Modification de profil"

Acteur	Scénario	Pré-condition
Utilisateur	<ol style="list-style-type: none"> 1. L'utilisateur choisit ses préférences et saisi les nouvelles coordonnées. 2. Le système demande une confirmation. 3. L'utilisateur confirme la modification. 	L'utilisateur est connecté à l'application ARIMA INSIGHT.
ARIMA INSIGHT	<ol style="list-style-type: none"> 1. ARIMA INSIGHT reçoit la requête pour afficher la page de profil. 2. ARIMA INSIGHT affiche la page de profil. 3. ARIMA INSIGHT applique les changements au profil de l'utilisateur après confirmation. 4. ARIMA INSIGHT enregistre les modifications dans la base de données. 	Le système est opérationnel.
Base de données	<ol style="list-style-type: none"> 1. La base de données reçoit les modifications de profil de la part d'ARIMA INSIGHT. 2. La base de données met à jour les informations de profil. 	La base de données est disponible et opérationnelle.

Développement

Les figures ci-jointes illustrent les interfaces de gestion de profil sur "ARIMA INSIGHT".

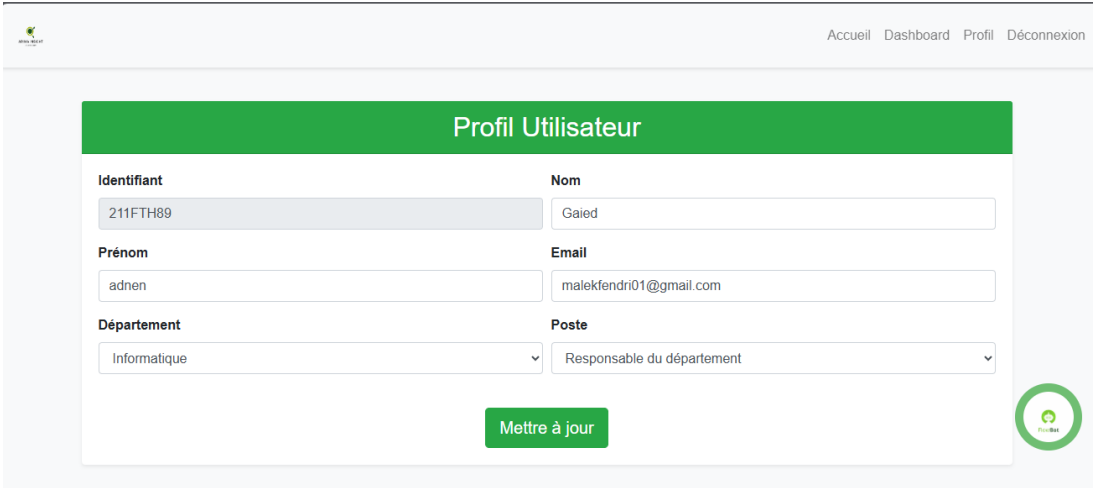


Figure 5.22 – Page profil "ARIMA INSIGHT"

Volet administrateur

- Ajout utilisateur
- Le diagramme de séquence suivant illustre le scénario d'ajout utilisateur sur "ARIMA INSIGHT".

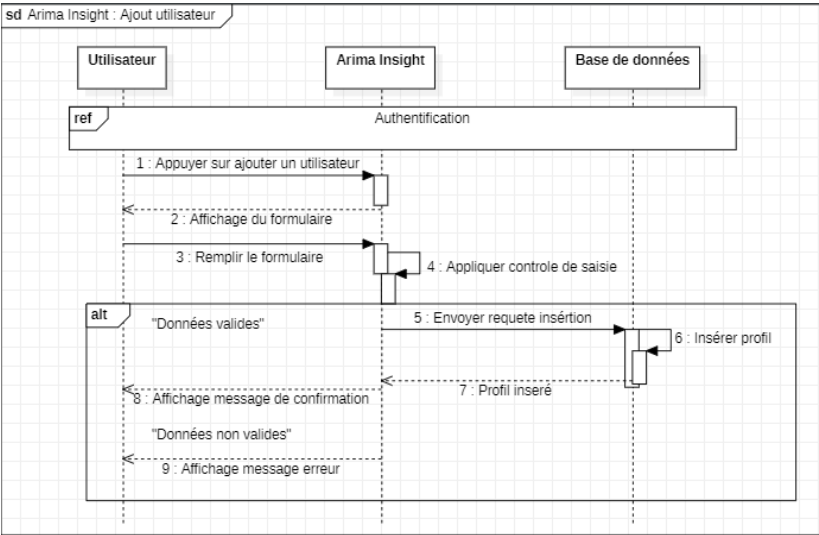


Figure 5.23 – Diagramme de séquence "ARIMA INSIGHT : Ajout utilisateur"

Ce tableau résume les actions spécifiques de chaque acteur impliqué dans le processus d'ajout utilisateur sur "ARIMA INSIGHT", ainsi que les pré-conditions nécessaires pour que chaque étape soit réalisée avec succès :

Table 5.9 – Description diagramme de séquence "ARIMA INSIGHT : Ajouter un utilisateur"

Acteur	Scénario	Pré-condition
Utilisateur	<ol style="list-style-type: none"> 1. L'utilisateur appuie sur "ajouter un utilisateur". 2. Le formulaire d'ajout s'affiche. 3. L'utilisateur remplit le formulaire. 4. L'utilisateur clique sur "soumettre". 	L'utilisateur est connecté à l'application ARIMA INSIGHT.
ARIMA INSIGHT	<ol style="list-style-type: none"> 1. ARIMA INSIGHT reçoit la requête pour afficher le formulaire. 2. ARIMA INSIGHT affiche le formulaire. 3. ARIMA INSIGHT applique le contrôle de saisie sur les données du formulaire. 4. ARIMA INSIGHT envoie la requête d'insertion à la base de données. 5. ARIMA INSIGHT affiche un message de confirmation ou d'erreur. 	Le système est opérationnel.
Base de données	<ol style="list-style-type: none"> 1. La base de données reçoit la requête d'insertion. 2. La base de données insère les nouvelles données de l'utilisateur. 	La base de données est disponible et opérationnelle.

Développement

Les figures ci-jointes illustrent les étapes de l'ajout d'un utilisateur sur "ARIMA INSIGHT".

ARIMA INSIGHT

AccueilAjoutSuppressionDéconnexion

Ajouter un utilisateur

Identifiant

Nom

Prénom

Département

Informatique

Poste

Responsable du département

Mot de passe

Figure 5.24 – Page ajout utilisateur "ARIMA INSIGHT "

Suppression profil

Le diagramme de séquence suivant illustre le scénario de suppression profil sur "ARIMA INSIGHT".

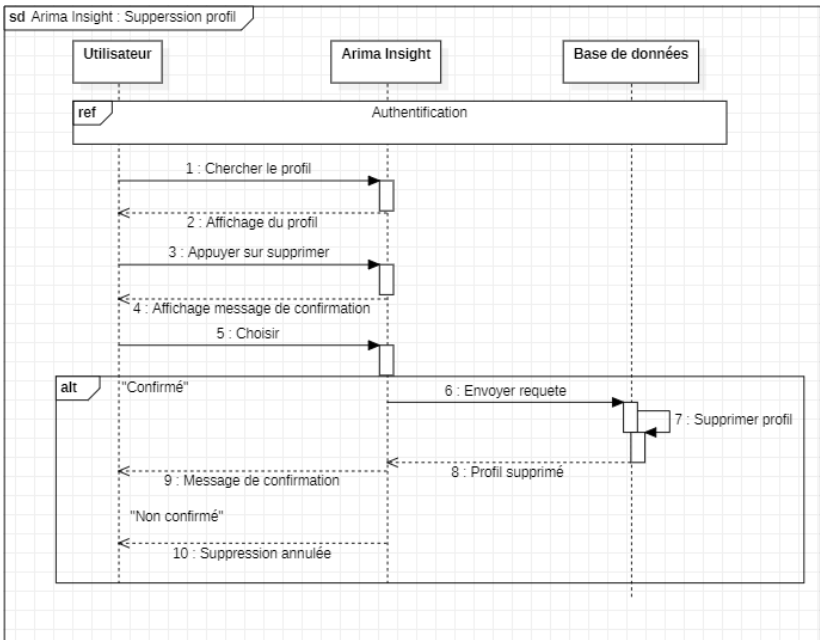


Figure 5.25 – Diagramme de séquence "ARIMA INSIGHT : Suppression profil"

Ce tableau résume les actions spécifiques de chaque acteur impliqué dans le processus de suppression profil sur "ARIMA INSIGHT", ainsi que les pré-conditions nécessaires pour que chaque étape soit réalisée avec succès

Table 5.10 – Description diagramme de séquence "ARIMA INSIGHT : Suppression de profil"

Acteur	Scénario	Pré-condition
Utilisateur	<ol style="list-style-type: none"> 1. L'utilisateur cherche le profil à supprimer. 2. L'utilisateur affiche les informations du profil. 3. L'utilisateur appuie sur "supprimer". 4. L'utilisateur confirme la suppression. 	L'utilisateur est connecté à l'application ARIMA INSIGHT.
ARIMA INSIGHT	<ol style="list-style-type: none"> 1. ARIMA INSIGHT affiche les informations du profil sélectionné. 2. ARIMA INSIGHT affiche une boîte de dialogue de confirmation. 3. ARIMA INSIGHT envoie la requête de suppression à la base de données après confirmation. 4. ARIMA INSIGHT affiche un message de confirmation ou d'erreur. 	Le système est opérationnel.
Base de données	<ol style="list-style-type: none"> 1. La base de données reçoit la requête de suppression. 2. La base de données supprime les informations du profil. 	La base de données est disponible et opérationnelle.

Développement

Les figures ci-jointes illustrent le processus de suppression d'un utilisateur sur "ARIMA INSIGHT" :

La figure 5.26 montre la page de gestion des utilisateurs, où les administrateurs peuvent visualiser les informations des utilisateurs existants et accéder à l'option de suppression. après avoir cliqué sur supprimer une alerte de confirmation est affichée .



Figure 5.26 – Page suppression utilisateur "ARIMA INSIGHT "

Conclusion

Ce chapitre a détaillé le processus de création d'un chatbot, en couvrant ses composants essentiels. Nous avons également couvert le développement de l'application web Arima Insight, en soulignant ses fonctionnalités principales et les étapes nécessaires à sa réalisation.

Conclusion générale et perspectives

Dans un monde où la quantité de données générées augmente de façon exponentielle, l'importance de l'informatique décisionnelle n'a jamais été aussi cruciale. Les entreprises doivent pouvoir analyser rapidement et efficacement ces données pour prendre des décisions éclairées et stratégiques. Notre projet s'inscrit dans cette dynamique, en fournissant une solution intégrée qui répond aux besoins actuels en matière de gestion des flux financiers et d'assistance utilisateur.

Nous avons développé un système décisionnel complet et robuste, capable de collecter, transformer et restituer des données de manière claire et pertinente. Notre projet, mené au sein de l'organisme d'accueil "Assurances Biat", a débuté par une phase de maîtrise des données, où nous avons analysé et généré des données synthétiques pour alimenter notre entrepôt de données. Nous avons choisi un modèle en étoile pour notre entrepôt, facilitant ainsi l'organisation et l'accès aux données. Cette modélisation nous a permis de structurer les informations de manière optimale, assurant ainsi une meilleure performance des requêtes et une facilité de navigation à travers les différents axes d'analyse.

Grâce à l'intégration de Power BI, nous avons pu créer des tableaux de bord interactifs qui offrent une vision globale et détaillée des opérations financières. Ces tableaux de bord permettent de visualiser les encaissements et décaissements, d'analyser les performances par branche, produit et réseau de distribution, et de suivre les variations des montants nets. Ils constituent un outil essentiel pour les décideurs, leur permettant d'identifier rapidement les tendances et anomalies, et d'ajuster leurs stratégies en conséquence. Cette capacité à transformer des données brutes en insights exploitables est un atout majeur pour toute organisation cherchant à rester compétitive et agile.

En parallèle, nous avons développé FlowBot, un chatbot intelligent conçu pour répondre aux questions des utilisateurs et offrir une assistance technique en temps réel. FlowBot est capable de comprendre les intentions des utilisateurs, de fournir des informations précises sur les transactions financières et de signaler les problèmes techniques, améliorant ainsi l'expérience utilisateur et l'efficacité du support. L'intégration de ce chatbot permet de réduire les délais de réponse et d'améliorer la satisfaction des utilisateurs, tout en libérant les ressources humaines pour des tâches à plus forte valeur ajoutée.

Notre application web "Arima Insight" centralise toutes ces fonctionnalités, offrant une plateforme unique où les utilisateurs peuvent accéder aux tableaux de bord, interagir avec FlowBot et gérer leurs profils. L'application inclut des fonctionnalités d'authentification, de réinitialisation de mot de passe, et de gestion des utilisateurs pour les administrateurs, assurant ainsi une sécurité et une flexibilité optimales. Cette plateforme intégrée permet non seulement une gestion centralisée des données, mais aussi une accessibilité accrue pour les utilisateurs finaux, facilitant ainsi l'adoption et l'utilisation quotidienne de l'outil.

Pour réaliser ce projet, nous avons suivi une méthodologie rigoureuse, débutant par la compréhension et l'analyse des besoins, puis la conception et la modélisation de l'entrepôt de données, la mise en place

des processus ETL, et enfin le développement et le déploiement des différentes composantes du système. Chaque étape a été soigneusement planifiée et exécutée, garantissant la qualité et la fiabilité de notre solution. Cette approche méthodique nous a permis de surmonter les défis techniques et organisationnels, assurant ainsi la réussite de notre projet.

En conclusion, notre projet réalisé au sein des Assurances Biat répond aux exigences actuelles de la BI en offrant une solution intégrée et complète pour la gestion des flux financiers. Grâce à un système décisionnel performant, un chatbot intelligent et une application web interactive, nous avons créé un outil puissant et polyvalent qui permet aux entreprises de prendre des décisions éclairées et de gérer efficacement leurs opérations. Notre approche méthodique et l'utilisation de technologies avancées assurent que notre solution est prête à répondre aux défis futurs de l'informatique décisionnelle. Nous sommes convaincus que notre projet apportera une valeur ajoutée significative à notre organisme d'accueil et servira de modèle pour des initiatives similaires dans d'autres contextes.

Pour l'avenir, nous envisageons plusieurs améliorations et extensions pour notre solution : l'amélioration du chatbot FlowBot pour intégrer et travailler avec des fichiers ainsi que pour prendre en charge les interactions vocales, ce qui améliorera encore l'efficacité et l'accessibilité de l'assistance utilisateur. De plus, nous prévoyons de déployer "Arima Insight" à plus grande échelle, en assurant une disponibilité continue et une sécurité renforcée. Le déploiement permettra à un plus grand nombre d'utilisateurs de bénéficier de ses fonctionnalités avancées. Ces perspectives visent à rendre notre solution encore plus complète et adaptée aux besoins futurs des entreprises, renforçant ainsi leur capacité à prendre des décisions basées sur des données précises et pertinentes.

Bibliographie

- [1] Assurances Biat, <https://www.assurancesbiat.com.tn/>, Mars 2024.
- [2] Assurances, <https://www.economie.gouv.fr/facileco/comprendre-assurance-risques>, Mars 2024.
- [3] Besoins fonctionnels et non fonctionnels, <https://savoir.plus/besoins-fonctionnels-non-fonctionnels/>, Mars 2024.
- [4] Méthode agile, <https://bubbleplan.net/blog/agile-scrum-gestion-projet/>, Avril 2024.
- [5] Kanban, <https://www.appvizer.fr/magazine/operations/gestion-de-projet/kanban>, Avril 2024.
- [6] SCRUM, <https://toolapp.fr/methode-agile-scrum/>, Avril 2024.
- [7] GANTT, <https://www.gantt.com/fr/>, Avril 2024.
- [8] Systèmes décisionnels, <https://datavalue-consulting.com/systeme-information-decisionnel/>, Avril 2024.
- [9] ETL, <https://www.talend.com/fr/resources/guide-etl/>, Avril 2024.
- [10] IA, <https://www.oracle.com/fr/artificial-intelligence/what-is-ai/>, Avril 2024.
- [11] NLP, <https://www.ibm.com/topics/natural-language-processing>, Avril 2024.
- [12] RASA, <https://rasa.com/docs/rasa/arch-overview/>, Avril 2024.
- [13] Architecture de RASA, <https://rasa.com/blog/the-rasa-masterclass-handbook-episode-7/>, Avril 2024.
- [14] Chatbots, <https://www.groupeonepoint.com/fr/nos-publications/comment-developper-son-chatbot-avec-rasa/>, Avril 2024.
- [15] Frameworks, <https://www.lemon-interactive.fr/actualites/developpement/tout-comprendre-framework-web-avantages-inconvenients/>, Mai 2024.
- [16] Flask, <https://flask.palletsprojects.com/en/latest/>, Mai 2024.
- [17] Conception, <https://grim.developpez.com/cours/businessintelligence/concepts/conception-datawarehouse/>, Avril 2024.
- [18] Planification, <https://help.talend.com/fr-FR/administration-center-user-guide/8.0/scheduling-execution-of-job>, Mai 2024.
- [19] Chatgpt, <https://chatgpt.com/>, Mai 2024.
- [20] Mesures Power BI, <https://learn.microsoft.com/fr-fr/power-bi/transform-model/desktop-tutorial-create-measures>, Mai 2024.