Projet de bases de données

420-P58-BB gr. 00005

**Analyse préliminaire**

**Présentation du projet**

Travail présenté à

Jean-François Brodeur

Par

**Kha Pham, Mathieu Larivée,**

**Oussama Lourhmati, Motoki Shintani**

Collège de Bois-de-Boulogne

Mardi 8 octobre 2019

2019

Document d’exigences

PRÉSENTATION DU PROJET

Table des matières

[Sommaire à l’exécutif 3](#_Toc26561948)

[Historique du document 4](#_Toc26561949)

[Sources 5](#_Toc26561950)

[Glossaire 6](#_Toc26561951)

[Description du projet 7](#_Toc26561952)

[Description 7](#_Toc26561953)

[Portée 7](#_Toc26561954)

[Fonctionnement et plateforme web 7](#_Toc26561955)

[Liste des intervenants impliqués dans le projet 8](#_Toc26561956)

[Détail des besoins 9](#_Toc26561957)

[Méthodologie de gestion de projet 11](#_Toc26561958)

[Contexte du projet 12](#_Toc26561959)

[Méthodologie de travail en équipe 15](#_Toc26561960)

[Philosophie de développement 16](#_Toc26561961)

[Nomenclature 16](#_Toc26561962)

[Qualité du code 16](#_Toc26561963)

[Instrumentation 17](#_Toc26561964)

[Documentation 18](#_Toc26561965)

[Assurance qualité 19](#_Toc26561966)

[Sécurité 19](#_Toc26561967)

[Tests 20](#_Toc26561968)

[Analyse des coûts et bénéfices du projet 21](#_Toc26561969)

[Coûts actuels 21](#_Toc26561970)

[Avantages de la solution 21](#_Toc26561971)

[Modèle de la base de données 23](#_Toc26561972)

[Clés de lecture 25](#_Toc26561973)

[Exemples d’interfaces 26](#_Toc26561974)

[Cas d’utilisation 33](#_Toc26561975)

[Diagramme des cas d’utilisation 33](#_Toc26561976)

[Cas d’utilisation 35](#_Toc26561977)

[**Cas d’utilisation 1** – S’authentifier 35](#_Toc26561978)

[Cas d’utilisation 2 – Sélectionner un filtre 36](#_Toc26561979)

[Cas d’utilisation 3 – Visualiser le tableau de bord 37](#_Toc26561980)

[Cas d’utilisation 4 – Ajouter/Modifier/Supprimer un filtre 38](#_Toc26561981)

[Cas d’utilisation 5 – Ajouter/Modifier/Supprimer un graphique 39](#_Toc26561982)

[Cas d’utilisation 6 – Ajouter/Modifier/Supprimer un rapport 40](#_Toc26561983)

[Cas d’utilisation 7 – Gérer les droits d’accès 41](#_Toc26561984)

[Échéancier 42](#_Toc26561985)

[Bibliographie 45](#_Toc26561986)

[Annexes 46](#_Toc26561987)

[Annexe A 46](#_Toc26561988)

[Annexe B 47](#_Toc26561989)

# Sommaire à l’exécutif

Ce document présente un projet de tableau de bord. Ce système permettra l’entreprise *TBS Engineering* d’analyser et interpréter facilement et rapidement les données de vente et d’achat. Ce système s’occupera donc uniquement du côté financier de l’entreprise. Il sera déployé sur une plateforme web et sera développé via APEX d’Oracle.

Ce document donne une description du projet, l’utilité de celui-ci pour l’entreprise concernée, comment il sera développé, combien il coûtera et dans quelles échéances il sera développé. De plus, il indique l’étendue du système et cherche à montrer les avantages au déploiement d’un tel système.

Il présente également un modèle de la base de données sous-jacente au système à l’aide d’un diagramme d’entités-relations standard. La base de données est fournie et maintenue par l’entreprise *JBM Logic Inc*.

Dans ce document, il y aura aussi un diagramme de contexte pour montrer la circulation des flux de données et un diagramme de cas d’utilisation pour donner une vision globale des différentes interactions entre l’utilisateur et notre système.

Il comprendra aussi quelques exemples d’interfaces suite à la présentation du modèle relationnel. Ces interfaces ont pour objectif de montrer ce à quoi peut ressembler le système proposé. Elles permettent aussi de montrer le potentiel et la souplesse d’une telle interface Web.

Une section sera aussi dédiée à l’assurance qualité du système comprenant la sécurité du système, tout ce qui sera testé et la véracité des informations présentées dans les différents graphiques et pages de l’application.

De plus, une section du document sera consacrée aux coûts et bénéfices du projet. En effet, adopter ce système, économiserait non seulement énormément de temps, mais aussi beaucoup d’argent. D’après nos calculs, les bénéfices pourraient facilement atteindre 4560$ par année, et plus de 10 800$ après 5 ans.

Enfin, le reste du document est constitué d’annexes diverses qui ont pour but de documenter la provenance des informations présentes dans les autres sections, ainsi que de compléter ou détailler certaines informations abordées dans les sections principales.

Nous espérons que vous verrez la force et le potentiel derrière le projet que nous vous proposons.

*Note : ce document emploie la graphie rectifiée.*

# Historique du document

Voici un historique des modifications apportées à ce document entre ses différentes versions.

|  |  |  |
| --- | --- | --- |
| Version | Date | Changements |
| 1 | 2019-09-03 | Version initiale. |
| 2 | 2019-09-04 | Ajout du sommaire et modification de l’apparence du document |
| 3 | 2019-09-05 | Ajout de la description du projet |
| 4 | 2019-09-05 | Ajout de la philosophie du projet |
| 5 | 2019-09-05 | Ajout des exemples d’interface |
| 7 | 2019-09-06 | Ajout de la liste des intervenants |
| 8 | 2019-09-09 | Ajout du modèle logique et des clés de lecture |
| 9 | 2019-09-10 | Ajout du contexte du projet (description) |
| 10 | 2019-09-10 | Ajout du diagramme de contexte |
| 11 | 2019-09-11 | Ajout de la méthodologie de gestion de projet |
| 12 | 2019-09-11 | Philosophie de développement, Glossaire |
| 13 | 2019-09-12 | Ajout du modèle logique |
| 14 | 2019-09-12 | Ajout des cas d'utilisation |
| 15 | 2019-09-12 | Modification du glossaire |
|  | 2019-09-18 | Diagramme de contexte |
| 16 | 2019-09-19 | Détail des besoins (compréhension doc TBS) |
| 17 | 2019-09-17 | Ajout de la méthodologie de travail d’équipe |
| 18 | 2019-09-23 | Ajout de l'échéancier (diagramme de Gantt) |
| 19 | 2019-09-24 | Modification de l’historique du document |
| 20 | 2019-09-24 | Modification de la section Nomenclature |
| 21 | 2019-10-02 | Modification de la section Instrumentation |
| 22 | 2019-10-03 | Modification bibliographie |
| 23 | 2019-10-03 | Modification source |
| 24 | 2019-10-03 | Modification détails des besoins |
| 25 | 2019-10-03 | Modification annexe |

*Tableau 1 : historique du document.*

# Sources

Ce tableau rassemble les différentes sources desquelles nous avons pris nos informations pour élaborer ce document ainsi que le projet.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Code | Nom | Date | Description | Location |
| S-001 | Compte-rendu de l’entrevue du 3 septembre 2019 | 2019-09-04 | Les notes de l’entrevue avec Jean-François Brodeur du 3 septembre | Manuscrit, <https://bit.ly/35458Ko>, |
| S-002 | Compte-rendu de l’entrevue du 10 septembre 2019 | 2019-09-10 | Les notes de l’entrevue avec Jean-François Brodeur du 10 septembre | Manuscrit, <https://bit.ly/35458Ko> |
| S-003 | Compte-rendu de l’entrevue du 17 septembre 2019 | 2019-09-17 | Les notes de l’entrevue avec Jean-François Brodeur du 17 septembre | Manuscrit, <https://bit.ly/35458Ko> |
| S-004 | Compte-rendu de l’entrevue du 24 septembre 2019 | 2019-09-24 | Les notes de l’entrevue avec Jean-François Brodeur du 24 septembre | Manuscrit, <https://bit.ly/35458Ko> |
| S-005 | Cours sur le temps | 2019-09-05 | Notes de cours sur le temps, avec Jean-François Brodeur | <https://bit.ly/2IA5Rtd> |
| S-006 | Rapport du client | 2019-10-03 | Détails du besoin du client | <https://bit.ly/2AEq905> |
| S-007 | Entrevue avec le client | 2019-10-03 | Réponses aux questions du client | <https://bit.ly/31Jp5Up> |

*Tableau 2 : les sources du document.*

# Glossaire

|  |  |
| --- | --- |
| Terme | Définition |
| Trello | Un outil de gestion de projet |
| Github | Un outil de gestion des versions (release) du projet. |
| APEX | Un framework permettant de construire une application |
| Diagramme d’entités-relations | Diagramme visualisant la relation entre les tables d’une base de données. |
| Stories | Les tâches à faire |
| Intrant | Les données rentrant dans un processus |
| Extrant | Les informations résultant du processus |
| Processus | Tâche en exécution |
| Diagramme de contexte | Diagramme permettant de visualiser les flux données entre un processus et les utilisateurs. |
| Diagramme de cas d’utilisation | Diagramme permettant de visualiser les interactions entre le système et les utilisateurs. |
| Tableau de bord (dashboard) | Un tableau visualisant les données financières et logistiques de l’entreprise sous plusieurs graphiques. |
| Flux de données | Le mouvement des données au sein d’un système. |
| Soumission | Une proposition de biens et services |
| WO | (Work Order) L’identifiant des projets |
| Integra | Système de gestion de base donnée permettant de faciliter les flux de donnée entre les entités et les processus. |
| Booking | Équipements en production (non fini) |
| Backlog | Rupture de stock (équipements pas encore livrés) |
| Diagramme UML | Diagramme permettant de modéliser la base de données. |

# Description du projet

## Description

Un tableau de bord représente un moyen efficace de communiquer des informations sur les données de gestion de l’entreprises (indicateurs de clé de performance). Dans le monde des affaires, il est utilisé pour évaluer le rendement d’un projet, d’une entreprise, d’un département, etc. sur une période de temps. Bref, il peut être appliqué à plusieurs endroits et pour différents types d’activités.

Grâce à un tableau de bord, les gestionnaires peuvent avoir rapidement un aperçu sommaire ou détaillé sur les activités d’un projet et d’en analyser les résultats. Par exemple, identifier les points forts et les points faibles et prendre les décisions appropriées. Le projet consistera à créer un tableau de bord afin d’aider l’entreprise à gérer ses ventes. Ce tableau de bord sera simple à comprendre afin de faciliter la compréhension des données.

## Portée

La portée de notre projet est de créer un tableau de bord pour l’entreprise *TBS Engineering* en utilisant les données présentes dans la base de données fournie et maintenue dans *Integra e-business*. Ceci consiste à concevoir et créer des graphiques ou des tableaux dynamiques basés sur une dimension temporelle. C’est-à-dire, que l’on affichera les informations nécessaires selon le jour, le mois ou l’année sélectionnée par l’utilisateur. Quant aux informations affichées dans les graphiques et les tableaux dynamiques, elles seront expliquées un peu plus tard dans le document, dans le détail des besoins.

## Fonctionnement et plateforme web

Le système sera déployé sur une plateforme Web via les serveurs Oracle. On utilisera Apex d’Oracle arrimé à INTEGRA e-business™ pour créer le système. Le système doit donner la possibilité à l’utilisateur de voir les ventes effectuées par 2 entités qui sont respectivement la vente d’équipements (machines) et la vente de pièces indépendantes. Ces ventes devront être représentées sur une période de temps. Il y aura aussi la présence des dépenses effectuées par l’entreprise pour permettre ces deux entités d’exister. L’utilisateur sera alors capable de choisir, de comparer les réalisations en chiffre des ventes et des dépenses sur une période de temps donnée et plus encore.

# Liste des intervenants impliqués dans le projet

|  |  |  |
| --- | --- | --- |
| **#** | **Intervenant** | **Rôle** |
| **1** | **Mathieu Larivée (Développeur)** | Développer le tableau de bord pour la compagnie TBS Engineering. |
| **2** | **Oussama Lourhmati (Développeur)** | Développer le tableau de bord pour la compagnie TBS Engineering. |
| **3** | **Motoki Shintani (Développeur)** | Développer le tableau de bord pour la compagnie TBS Engineering. |
| **4** | **Kha Pham (Développeur)** | Développer le tableau de bord pour la compagnie TBS Engineering. |
| **5** | **Jean-François Brodeur** | Mettre en contact le client et les développeurs du tableau de bord. |
| **6** | **TBS Engineering** | Fournir les données nécessaires et administrer le tableau de bord. |
| **7** | **JBM Logic** | S’occuper de la base de données pour la compagnie TBS Engineering. |

# Détail des besoins

Le client a besoin que les données de sa base de données *Integra* soient traitées et par la suite affichées dans un tableau de bord interactif créé avec Apex. Cet outil de développement Web permettra d’afficher les dépenses, les ventes, les profits, un indicateur de performance et d’autres informations utiles à l’analyse des résultats et à la prise de décision.

Par exemple, le gestionnaire aura accès un tableau de bord qui lui permet de voir, pour une période de temps donnée, le nombre de produits vendus, les plus vendus, le coût de la production ou les meilleurs clients de l’entreprise.

Du côté financier, pour représenter les performances globales de l’entreprise, plusieurs graphiques afficheront l’état financier : un graphique affichant le nombre de ventes, le nombre de commandes et le nombres de ventes non-complétées. Un graphique affichant les performances de l’entreprise (ex: le nombre de ventes, le profit des ventes, les dépenses...) sur plusieurs périodes de temps (la dernière année, l’année actuelle, le nombre d’années écoulées depuis une date spécifique). À chaque mois, ces graphiques devront être générés :

* Un graphique affichant l’âge des soumissions et les détails des soumissions triées par pièce et par projet (grande commande, équipements, machines) ;
* Un graphique affichant les soumissions réussies et les soumissions échouées ;
* Un graphique affichant le solde bancaire ;
* Un graphique montrant le top des ventes ordonnées par type (pièce et projet) et les détails des ventes ;
* Un graphique comparant les coûts et les revenus ;
* Un graphique visualisant les tendances des ventes par mois et par année pour les pièces ;
* Une liste montrant les rapports opérationnels (les envois et les livraisons) triés par mois ;
* Un graphique comparant le booking et le backlog (voir glossaire)

Voici d’autres graphiques pertinents qui pourraient être ajoutées au projet :

* Un graphique comparant les revenus attendus et les revenus actuels.
* Un graphique affichant les estimations du coût des projets triés par la date de la proposition, le nom du client, la probabilité d’acceptation, la date d’acceptation et l’étape courante de la vente.
* Un graphique permettant de comparer les objectifs des ventes (estimation des ventes pour une période) et les ventes réalisées (les ventes réelles) triés par catégorie ;

De plus, et afin d’identifier la catégorie des commandes, un identificateur WO  (voir glossaire) est affiché quand l’item est un projet (grande commande, équipements) et il est absent dans le cas d’une simple commande de pièces.

*Voir annexe B*

Pour les projets, le client a besoin d’une vision sur les projets actifs de manière dynamique. Le tableau de bord va donc contenir les différents stades des projets. Cette section devra contenir jusqu’à 20 projets actifs à n’importe quelle instance. Il serait préférable que le projet soit ajouté automatiquement au moment que celui-ci est assigné et retiré de la liste lorsque le projet est mis en vente. Cet élément du tableau de bord utilisera le concept des couleurs des lumières des feux de circulation afin de visualiser l’état ou le statut d’un projet : Vert lorsque le projet est complété, jaune lorsque le propriétaire du projet est en train de travailler sur ce dernier et rouge lorsque rien n’a été fait pour la progression du projet.

Le client voudrait avoir aussi un accès rapide aux économies réalisées grâce aux achats de produits en grand volume. Ceci sera sous forme de rapport dans l’application APEX d’Oracle. Il serait important de pouvoir suivre cette information au moment de la réception dû au manque de code-à-barre individualisé des stocks. Ceci aidera à suivre les efforts sur les initiatives de réduction des coûts matériels du client.

Un autre besoin serait d’avoir un rapport montrant les produits qui se vendent le plus souvent et qui sont donc en mouvement rapide. Ce serait surtout pour les articles qui sont en mouvement plusieurs fois par année, afin de plus facilement planifier et contacter les fournisseurs pour une meilleure tarification en raison du volume.

# Méthodologie de gestion de projet

Pour gérer notre projet, nous utiliserons la méthode Scrum, la méthode Agile. Elle s’agit de la méthode la plus utilisée et documentée de nos jours.

Il existe trois rôles au sein d’une équipe agile :

* Le maître de mêlée (scrum master) qui est le gardien de la méthode,
* Le propriétaire du produit (Product Owner)
* Les membres de l’équipe de développement qui n’ont nécessairement pas de rôles définis à l’avance et qui représentent un groupe de 6 à 10 développeurs.

C’est grâce à ses différents rôles et piliers du scrum qui font la popularité de cette méthode très efficace et productive.

Le déroulement de la méthode scrum se fait en plusieurs étapes. Tout commence avec le Product backlog où le client va donner ses exigences et où va se retrouver l’ensemble des fonctionnalités se nommant « User Story ». La « User Story » contient la fonctionnalité du point de vue utilisateur sous une phrase simple. L’ensemble du projet est divisé en cycles de vie se nommant Sprints. Chaque sprint a une durée de vie courte, habituellement d’un mois au maximum, durant laquelle l’équipe va développer une partie livrable du produit. Avant chaque sprint, l’équipe fait un Sprint Planning Meeting, où elle détermine quel user story sera à traiter et la manière la plus efficace d’élaborer le travail. Les user story sont choisis par ordre de priorité et de complexité. Pour chaque sprint, il y a une réunion d’avancement (Daily Scrum Meeting) quotidienne, pendant environ un quart d’heure, où le scrum master va poser des questions à chaque membre de l’équipe. Cette rétrospection quotidienne permet de mieux connaître l’état du projet, de supporter et stimuler les membres et de s’assurer que le tout va bien. Par la suite, le scrum master va former une courbe d’avancement (Burndown Chart) qui représente les efforts de l’équipe par rapport à l’avancement du projet.

De notre côté, avant chaque sprint, lors du Daily Scrum Meeting, nous utiliserons le « Planning Poker » qui consiste à définir le niveau de difficultés de chaque tâche à faire. Pour cela, chacun des membres de l’équipe devra choisir un nombre (1, 2, 3, 5, 8, 13, 21, 34, 55, 89 ou 144) pour évaluer la difficulté d’une tâche. Par la suite, on va devoir choisir un nombre qui représente le plus la moyenne des nombres que chacun a choisi.

Bref, notre but sera de finir à chaque semaine un certain nombre de “stories” dépendamment du niveau de difficulté qu’on aura choisi.

# Contexte du projet

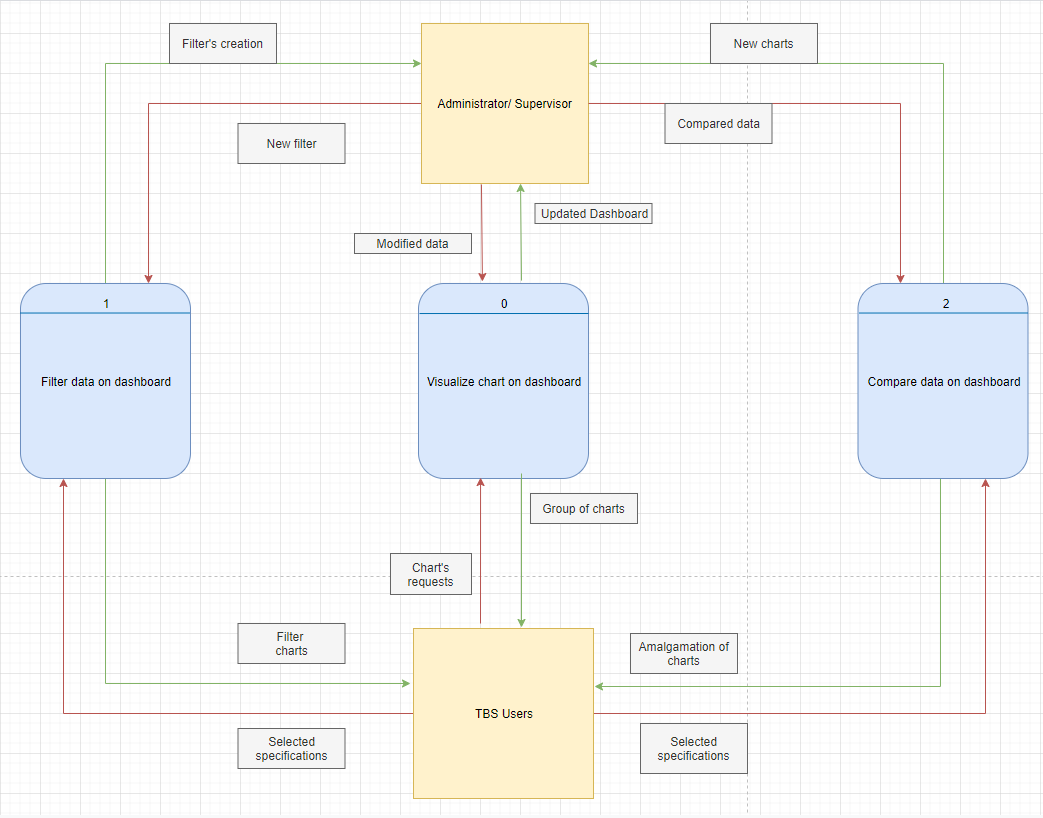
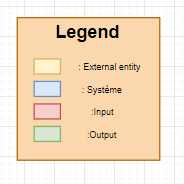
Sachant que les données faisant la source des tableaux de bord et des graphiques à générer sont stockés dans la base de données, le système à réaliser aura comme fonction le traitement et l’exécution des requêtes utilisateur et l’affichage des résultats sous format graphique selon des filtres et des critères bien définis.

Du côté utilisateur, nous pouvons définir deux fonctions principales :

* Administrateur ou Supervisor : c’est un utilisateur qui aura le pouvoir de créer, modifier ou supprimer des nouveaux graphiques, des nouveaux rapports et aussi des nouveaux filtres
* Utilisateur : ce sont plutôt les employés qui vont utiliser ces tableaux de bord pour analyser les ventes et la génération des rapports.

Le diagramme suivant illustre la relation entre les systèmes et les flux de données. Pour ce faire, les entités externes c’est-à-dire les personnes communiquant avec le système vont rentrer les données ou plutôt élaborer une demande d’informations au système *(Managing Dashboard)*. Ce dernier va recevoir les entrées de l’utilisateur, et va les traiter sous format d’une requête à base de différents critères et filtres. Cette requête est ensuite exécutée et les résultats sont envoyés sous format graphique à l’utilisateur final.

Le flux de données est plutôt un traitement des données depuis la base de données en backend et leur affichage dans une page Web sous format de tableau ou de graphique (extrant) pour répondre à des requêtes utilisateurs déclenchées par la définition d’un filtre depuis un browser Web (entrant).



Voici le flux du diagramme :

1. **Administrateur/Superviseur:**
   1. **Intrant:**
      1. Modified data
      2. New filter
      3. Compared data
   2. **Extrant:** 
      1. Updated dashboard
      2. Filter’s creation
      3. New Charts
2. **Utilisateurs TBS:**
   1. **Intrant:** 
      1. Selected specifications
      2. Chart’s request
      3. Selected specifications
   2. **Extrant:**
      1. Filter chart
      2. Group of charts
      3. Amalgamation of charts

Clés de lecture :

* Les utilisateurs de TBS peuvent filtrer les diagrammes.
* Le gestionnaire de tableau de bord renvoie les diagrammes filtrés
* Les utilisateurs de TBS demandent les diagrammes du tableau de bord.
* Le gestionnaire de tableau retourne les diagrammes demandés.
* Les utilisateurs de tableau de bord peuvent comparer les diagrammes entre elles
* Le gestionnaire de tableau de bord renvoie un diagramme avec les comparaisons
* L’administrateur/ le superviseur peut modifier le gestionnaire de tableau de bord.
* Le contenu du gestionnaire de tableau de bord sera mis à jour.
* L’administrateur/ le superviseur peut ajouter un nouveau filtre.
* Le gestionnaire de tableau de bord renvoie un nouveau filtre.
* L’administrateur/ le superviseur peut ajouter une fonctionnalité permettant de comparer les diagrammes.
* Le gestionnaire de tableau de bord renvoie des nouveaux charts comparant les données.

# Méthodologie de travail en équipe

Afin de bien gérer notre travail d’équipe, nous allons utiliser plusieurs outils qui nous aideront à faciliter la communication de nos informations.

Puisque nous allons utiliser des stories afin de bien connaître le travail que nous devons compléter au cours de chaque semaine, nous utiliserons Trello pour gérer ces stories. Trello nous permet de connaître quelles stories ont besoins d’être complété, celles qui sont en cours, ainsi que celles qui sont terminées. Il est aussi possible de mettre des commentaires sur les stories afin de faire savoir aux autres membres de l’équipe ce qu’il se passe et s’il y a un potentiel problème.

Afin de pouvoir modifier les différents composants du projet (requêtes SQL), nous allons utiliser Github, qui nous permettra de centraliser les sources et de facilement avoir accès à notre projet de n’importe quel ordinateur et donc ceci facilitera le travail d’équipe, même si nous ne sommes pas dans la même classe.

Nous allons aussi utiliser APEX d’Oracle afin de développer le tableau de bord pour notre client. Avec nos identifiants APEX, nous pouvons nous connecter à l’application de n’importe quel ordinateur vu que le tout est supporté avec *Integra*.

Nous utilisons Google Drive pour le partage de document et de fichiers tels que les exports de l’application APEX, documents du projet, nos schémas (générés avec Draw.io), notes de cours, etc.

Pour ce qui est de notre méthode de communication, Messenger sera utilisé pour la communication d’information simple et textuelle. Pour ce qui est de la communication verbale, nous utiliserons Discord. Toutefois, le courriel reste l’outil de communication utilisé pour tout échange d’information en mode asynchrone.

En cas de panne, nous avons un backup. En effet, avec Apex, nous pouvons exporter le projet en tout temps. Donc en cas de panne, nous avons la possibilité de restaurer le tout grâce au projet (exporté) que l’on va cette fois importer.

# Philosophie de développement

Le but de la philosophie de développement est de rendre l’application le plus simple à maintenir ou ajouter du contenu le plus facilement possible. Il faut alors respecter plusieurs grandes lignes au sein de l’équipe pour faciliter la compréhension du code.

## Nomenclature

Tout d'abord, il faut respecter des conventions quant à la manière de nommer les tables, les colonnes et les variables qu’on utilisera. On utilisera des noms en majuscules pour tous les noms de tables, vues, etc.

**Exemple** : **CLIENT\_ID** et non “clientId”.

Pour rester dans le thème des noms, ceux-ci devront être significatifs et éviter les noms trop généraux. Il faut par exemple éviter de nommer à une variable le nom “v1”. On doit aussi séparer les mots par des “underscores”, cela permettra une meilleure lisibilité. Les noms de méthodes doivent elles aussi être significatives et doivent commencer par un verbe qui explique ce que celle-ci fait. Dans les versions antérieures d’Oracle, il n’y a pas de différence entre un caractère minuscule et majuscule. C’est pour cette raison que tous nos variables et méthodes seront en majuscules, et que les noms devront être séparés par des “underscores”.

## Qualité du code

Afin d’assurer une qualité et réaliser un projet les standards, nous devons respecter les bonnes pratiques suivantes au niveau programmation :

* Il faut éviter le code trop compliqué et produire, autant que possible, du code simple. Si on se retrouve bloqué avec un bout de code et qu’on ne trouve pas de solution, il faudra toujours chercher à faire quelque chose pour éviter d’alourdir le code.
* Les méthodes devront comporter un maximum de 15 à 20 lignes et elles seront toutes commentées avec l’aide de Javadoc.
* En SQL, il est facile de commencer à faire énormément de lignes de code grâce à la facilité du langage. C’est une force de ce langage mais cela peut aussi très rapidement devenir un désavantage si on commence à écrire de très longs énoncés.
* De plus, pour assurer la qualité du code, nous travaillerons en équipes de 2 lorsque cela est possible, dans une idée de “pair programming”. Le pair programming sert à avoir à la fois une idée générale et spécifique du projet et aussi de faire la révision du code « code review ». En binôme, c’est plus facile de voir les erreurs arriver, les problèmes de structure du code, etc. À court terme, c’est moins efficace que de programmer seul, mais à long terme, cela garantit une production constante avec un code de qualité supérieure.

## Instrumentation

Afin d’améliorer l’analyse du code, nous allons utiliser OraOpenSource, un outil de journalisation (logging tool). Cet outil permet de journaliser le comportement d’un programme, ce qui permet de corriger plus facilement les erreurs et d’améliorer la qualité du code.

En effet, il y a quatre niveaux de journalisation : Debug level, Info level, Error level et Fatal level. Dans le Debug level, on pourrait vérifier le contenu des variables mises en paramètre. Dans Info level, on peut visualiser les résultats d’une méthode. Dans Error level, il est possible de voir les erreurs déclenchées. Dans Fatal level, on a un meilleur aperçu des exceptions non gérées. Avec ces différentes fonctionnalités, les programmeurs auront une meilleure compréhension du code. C’est ce qui va permettre de mieux optimiser le code et de maximiser le temps.

Dans notre cas, afin d’être efficace, nous allons journaliser les exceptions gérées et non gérées, les logins invalides, les données en paramètre, mauvais résultats, un horodateur de l’entrée dans chaque méthode et un horodateur de la sortie de chaque méthode. Les deux derniers points serviront à évaluer la performance de notre application. Selon nous, ce sont les points les plus susceptibles de causer le mauvais fonctionnement d’un programme.

# Documentation

Le français et l’anglais sont, de manière générale, plus faciles à comprendre que n’importe quel langage de programmation. Dans cette optique, chaque méthode PL/SQL sera accompagnée d’un paragraphe la décrivant. Dans ce paragraphe, on trouvera : une description, la valeur de retour (si applicable), les variables en paramètre, les constantes, et plus encore. Pour voir un exemple plus précis, voir annexe A. Avec une bonne documentation, nous pourrons économiser du temps à chaque fois que nous avons besoin de modifier du code. Le défi sera de ne pas oublier de mettre à jour la documentation lorsque nous modifions le code, sans quoi nous compliqueront notre tâche.

Nous allons documenter également l’application Apex. Nous expliquerons en détail où l’information se trouve. Entre autres, nous décrirons comment sélectionner un item différent dans une liste déroulante ou de cliquer un bouton peut modifier le graphique associé. Nous expliquerons que le menu en haut à gauche est la façon principale de se « déplacer » dans l’application. Bref, tout ce que nous jugeons nécessaire et utile sera mentionné. Avec ceci, les utilisateurs et programmeurs ne devraient pas entrer dans une application totalement inconnue. Les utilisateurs devraient naviguer à travers les différents types de diagrammes aisément, et les programmeurs ne devraient pas avoir trop de difficulté à modifier l’interface pour qu’elle soit plus à leur goût.

# Assurance qualité

L’assurance qualité est un aspect primordial pour le développement d’une telle application. Nous mettons en place toutes les pratiques appropriées au projet dès la première phase. Toutes les activités ont été bien définies et validés par Mr Jean-François Brodeur, expert en bases de données et en développement de logiciels. Nous avons défini tous les points importants qui concernent la revue du code, l’audit du logiciel, la vérification, la validation, etc. Nous tenons à ce que toutes les bonnes pratiques soient respectées afin que le produit à développer soit adapté aux attentes de notre client et respectant le budget accordé à ce projet.

Afin d’assurer aussi une bonne expérience utilisateur, tous les graphiques et informations affichés seront tous validés avec l’enseignant. La conception de la base de données a été aussi faite selon les bonnes pratiques. Toutes les requêtes et le code ont été optimisés selon les recommandations de Mr Brodeur.

Des différents tests seront réalisés afin de s’assurer que tous les éléments de la solution sont fonctionnels et bien intégrés.

En plus de commenter le code, nous allons documenter toutes les composantes en mentionnant les références qui y sont liées.

## Sécurité

Notre application sera entièrement sécurisée selon les normes de sécurité du domaine. Tout d’abord, nous allons implémenter des autorisations pour chaque page. En effet, les pages seront accessibles selon le rôle et les privilèges de l’utilisateur. Par exemple, si l’utilisateur est un administrateur, il aura accès à l’ensemble des pages de l’application, y compris la page d’administration où il peut ajuster les réglages. Les rôles qu’un utilisateur peut prendre seraient un administrateur, directeur des finances, directeurs des ventes et directeur de production. Une ségrégation qui a été établie selon les besoins d’accès aux données par chacun des rôles et selon le modèle affaire de l’entreprise.

Pour l’authentification, après trois échecs d’authentification, le compte de l’utilisateur sera verrouillé pendant 1 heure. De plus, tous les mots de passes seront correctement hachés dans la base de données. Donc même avec un accès aux données de la base de données, il sera impossible de connaitre un mot de passe. Il est aussi important de mentionner que tout les authentifications (réussies ou échouées) seront auditées et stockées dans une table (table des logs fructueux et infructueux). Il sera alors toujours possible de voir qui a essayé de se connecter et à quel moment.

Finalement, l’utilisateur aura la possibilité de changer son mot de passe. Pour cela, il faut qu’il se connecte (s’authentifie) et par la suite, il pourra rentrer son nouveau mot de passe au clavier.

## Tests

Chaque fonctions et procédures (requêtes, code) seront testé. En fait, nous allons faire des tests unitaires pour les fonctions et procédures. Le reste sera testé à travers des tests systèmes, donc vérifier si les cliques et les accès aux pages fonctionnent correctement. L’ensemble des tests seront entièrement bien détaillés dans un document dédié aux tests.

# Analyse des coûts et bénéfices du projet

Nous tentons ici d’exposer les avantages de notre solution, en matière de coûts et de bénéfices. Les coûts représentent les dépenses liées au projet. Les bénéfices représentent les avantages ou les profits du projet. Afin de comparer le coût et les bénéfices du projet, nous allons les analyser sous l’angle qualitatif et quantitatif. Le côté qualitatif représente les conséquences subjectives d’un projet. Par exemple, le fait de posséder un tableau de bord pourrait accélérer le processus de rédaction des rapports, bien contrôler les ventes et l’avancement des projets. En revanche, la partie quantitative représente les effets concrets d’un projet comme l'économie de 15 000$ par année. À partir de ces notions, nous pouvons estimer les avantages et les inconvénients de ce projet.

## Coûts actuels

Pour l’instant, nous sommes toujours dans la phase de développement, c’est-à-dire que les coûts reliés au développement ne sont pas à considérer actuellement. Cependant, il est important de faire un suivi de l’application, et donc payer un stagiaire pour la maintenance de l’application, ce qui représenterait une somme annuelle d’environ 6 240$. Cette somme peut sembler grande, mais une telle maintenance pourrait augmenter les ventes de l’entreprise de 2%. En fin de compte, ce serait bénéfique pour l’entreprise.

## Avantages de la solution

Le tableau suivant fait un récapitulatif cumulatif de ces coûts et bénéfices sur une période de cinq ans. Les coûts reliés au développement du projet, sont pour l’instant inexistants vu que les développeurs sont des étudiants et que le tout est produit dans le cadre d’un projet scolaire, mais cela reste à considérer pour les années à venir. En effet, dans les années à venir, il est possible que ce ne soit plus des étudiants qui développent l’application, il faudra alors payer des stagiaires (15$/h à 20$/h) à temps partiel. Sinon, puisque l’entreprise possède déjà l’abonnement Oracle ainsi que la licence, les coûts reliés à cet abonnement ne sont pas compris dans le tableau ci-dessous.

Dans une situation où l’application n’est pas existante, faire un rapport (des ventes, des soumissions, des clients, etc.) prendrait en moyenne 3h à produire. Payer les employés, tels que les gestionnaires de projet, de personnel ou de ventes, pour produire les multiples rapports mensuels et annuels représenterait une somme assez importante pour l’entreprise. En effet, si on prend en compte qu’un rapport prend en moyenne 3h à élaborer par mois, que la moyenne des salaires représente approximativement 30$/h et qu’il y a environ une dizaine de rapport à faire mensuellement, la somme peut atteindre facilement 900$ par mois, c’est-à-dire 10 800$ par année.

Avec l’aide de notre application en développement, cette somme annuelle pourrait représenter le salaire réutilisé pour l’entreprise, ce qui serait un avantage. En effet, après un an, les profits bruts seront de 4560$, et après 5 ans, 10 800$. Donc, adopter un tel système, serait bénéfique pour l’entreprise.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Coûts | 1 an | 2 ans | 3 ans | 4 ans | 5 ans |
| Développement | 0 $ | 3 000 $ | 6 000 $ | 9 000 $ | 12 000 $ |
| Maintenance (par un stagiaire) | 6 240 $ | 12 480 $ | 18 720 $ | 24 960 $ | 31 200 $ |
| Total | 6 240 $ | 15 480 $ | 24 720 $ | 33 960 $ | 43 200 $ |
| Bénéfices | 1 an | 2 ans | 3 ans | 4 ans | 5 ans |
| Salaire réutilisé | 10 800$ | 21 600$ | 32 400$ | 43 200$ | 54 000$ |
| Total | 10 800$ | 21 600$ | 32 400$ | 43 200$ | 54 000$ |
| Grand total | 4 560$ | 6 120$ | 7 680$ | 9 240$ | 10 800$ |

*Tableau 3 : coûts et bénéfices du nouveau système.*

# Modèle de la base de données

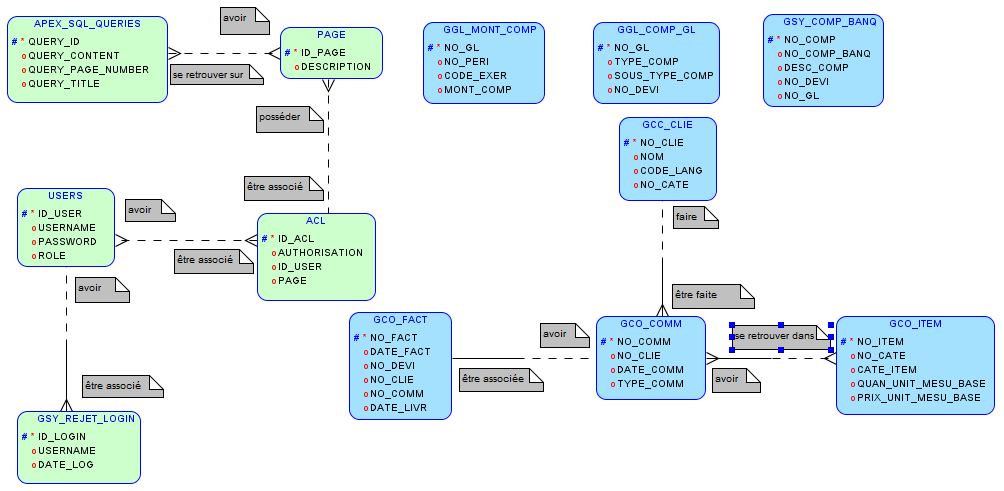
Voici un diagramme entité-association, réalisé à l’aide de la notation standard d’Oracle.

Une entité peut représenter à peu près n’importe quoi, un objet physique (ex: une pièce), quelqu’un (ex: un employé, un fournisseur), une action (ex: une vente). Une association se traduit par deux entités qui interagissent entre elles (un employé peut faire une vente, une pièce peut être vendue par une vente). Alors, un diagramme entité-association est un moyen de représenter comment différentes parties d’un système sont interconnectées.

Le modèle logique suivant est divisé en deux parties.

À gauche, ce sont nos tables. Les tables métrique et *indicateur performance* serviront à personnaliser le tableau de bord. Si les employés de TBS veulent modifier l’échelle d’un diagramme, ils auront seulement besoin de passer par ces tables au lieu d’entrer dans le code. La table *notre client* comprendra les informations de TBS, soit principalement les informations de la personne-ressource.

À droite, ce sont une estimation des tables que TBS possède, avec ce que nous savons sur le projet à date. Leurs tables ainsi que leurs données nous seront fournies de toute façon, mais pour l’instant ce modèle servira de “placeholder”. Cela représente aussi un creuse-méninge pour comprendre comment les différentes parties de leur système interagissent entre elles.



*Figure 2 : diagramme d'entités-relations – et les fréquences ??*

## Clés de lecture

* Un apex\_sql\_query peut se retrouver sur une ou plusieurs pages.
* Une page peut avoir une ou plusieurs apex\_sql\_query.
* Une page peut posséder un acl.
* Un acl peut être associé à une ou plusieurs pages.
* Un user peut avoir un ou plusieurs acl.
* Un acl peut être associé à un ou plusieurs users.
* Un user peut avoir un ou plusieurs gsy\_rejet\_login.
* Un gsy\_rejet\_login doit être associé à un user.
* Un gcc\_clie peut faire une ou plusieurs gco\_comm.
* Une gco\_comm doit être faite par un client.
* Une gco\_fact doit être associée à une gco\_comm.
* Une gco\_comm peut avoir une gco\_fact.
* Un gco\_item peut se retrouver dans une ou plusieurs gco\_comm.
* Une gco\_comm doit avoir un ou plusieurs gco\_item.

# Exemples d’interfaces

Voici quelques exemples de ce à quoi pourrait ressembler l’interface d’utilisation du tableau de bord (Dashboard).

Il y aurait une page pour visualiser les meilleurs clients de la compagnie. Sur un intervalle de temps que l’utilisateur peut sélectionner, il va pouvoir visualiser le nombre de commandes des vingt meilleurs clients et les factures des dix meilleurs clients.

Il y aurait aussi une section pour visualiser les ventes d’équipements et de pièces. Il serait possible de visualiser le total des ventes par année et le total par mois selon l’année sélectionnée par l’utilisateur.

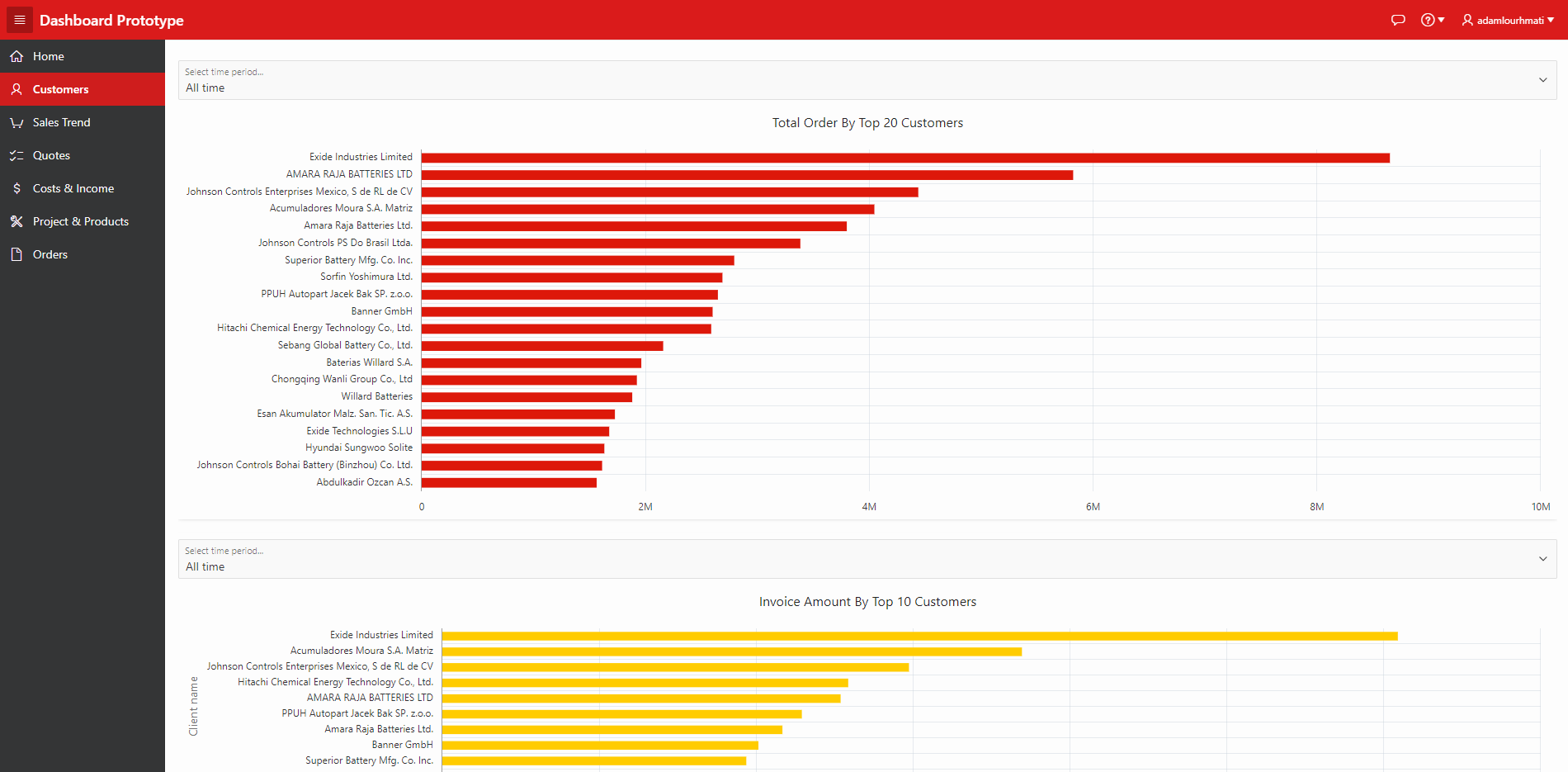
De plus, une page sera consacrée aux soumissions. Tout d’abord, il y aurait un graphique pour l’âge des soumissions, où une couleur représente une catégorie d’âge ; rouge pour les soumissions de plus d’un an, vert pour celles de six à douze mois, jaune pour celle d’un à trois mois et rouge-orange pour celles de trois à six mois. Il y aurait aussi un autre graphique pour visualiser le nombre de soumissions réussies (vert), en cours (jaune), perdues (rouge). L’utilisateur peut aussi choisir le type (pièces ou équipements) et peut cliquer sur l’un des graphiques pour avoir plus de détails.

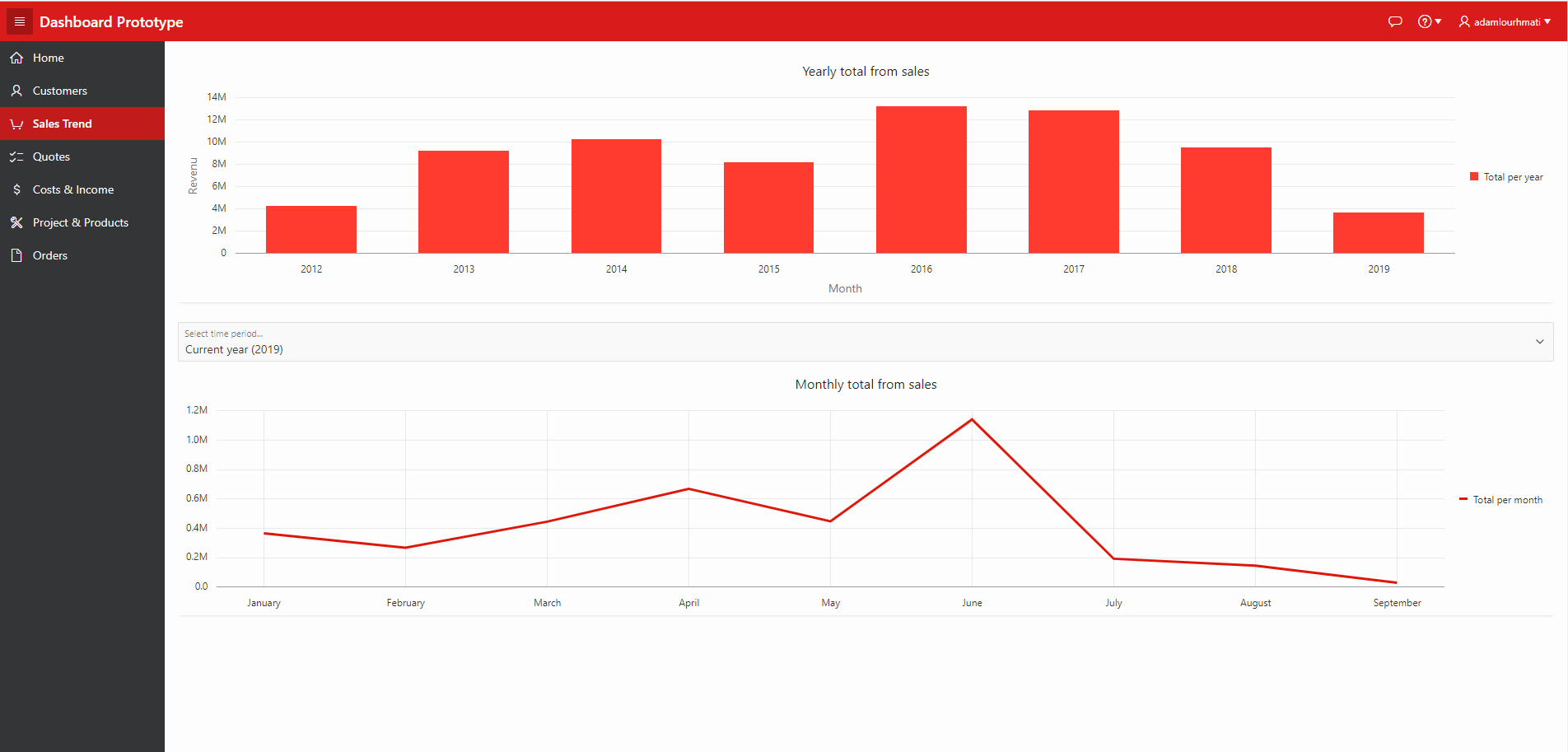
L’utilisateur aura aussi accès à une page des couts (vert), des dépenses (rouge) et des profits (jaune). Le tout sous forme de deux graphiques différents et où il peut choisir l’année de son choix.

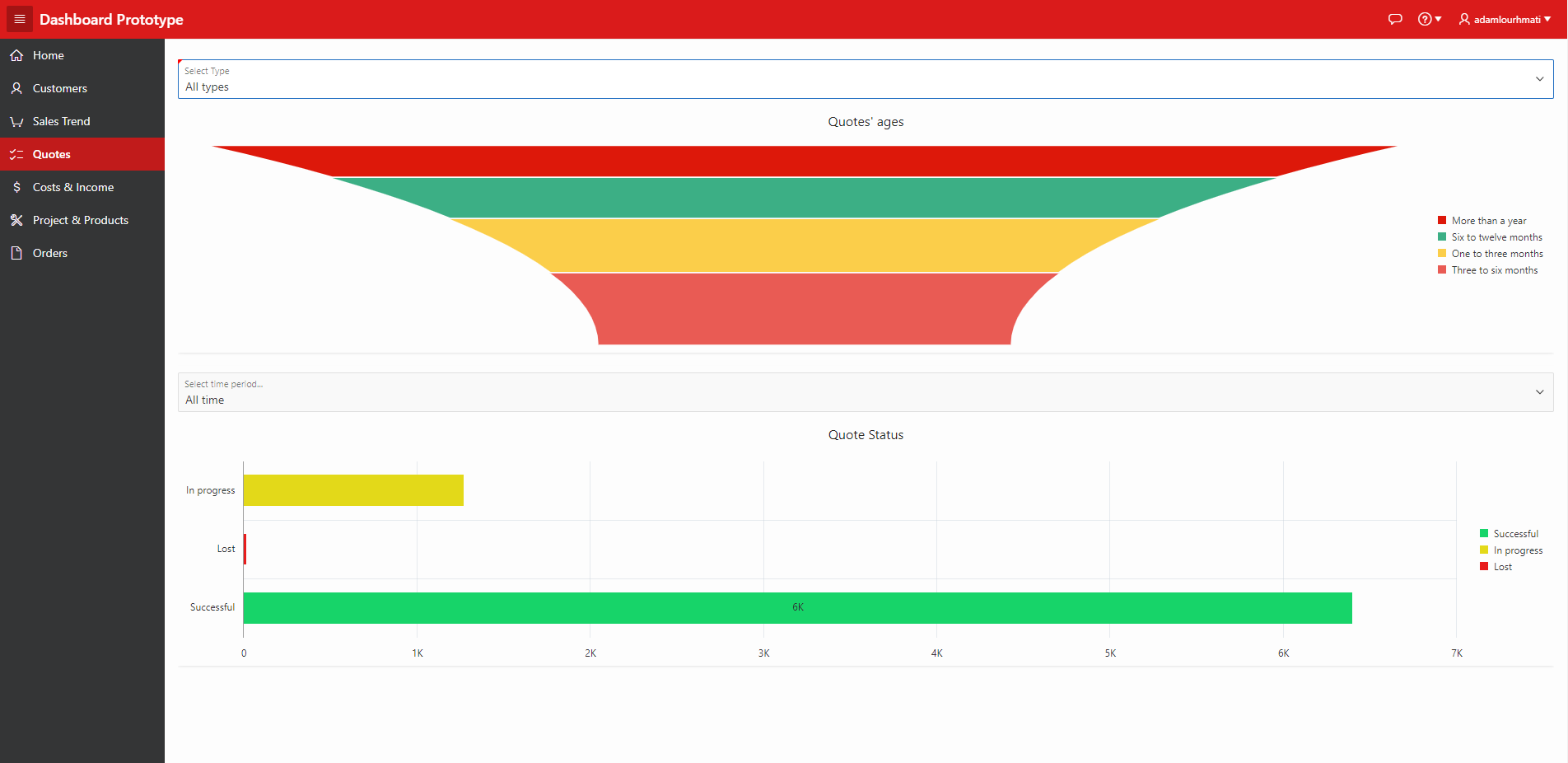
Il y aurait aussi une page pour visualiser les produits ou projets les plus vendues sous forme de deux graphiques. Le premier graphique afficherait les dix items les plus vendues (quantité) avec leurs noms. Le deuxième ce serait les dix items qui ont rapporté le plus d’argent. Dans les deux graphiques, il sera possible de choisir un intervalle de temps et de cliquer pour avoir plus de détails.

Plus encore, l’utilisateur aura aussi accès à une page des bookings (ce qui est en production et prévu) et des backlogs (rupture de stock). Le tout sera illustré sous formes de deux listes où l’utilisateur pourra rechercher un produit en booking ou en backlog en particulier.

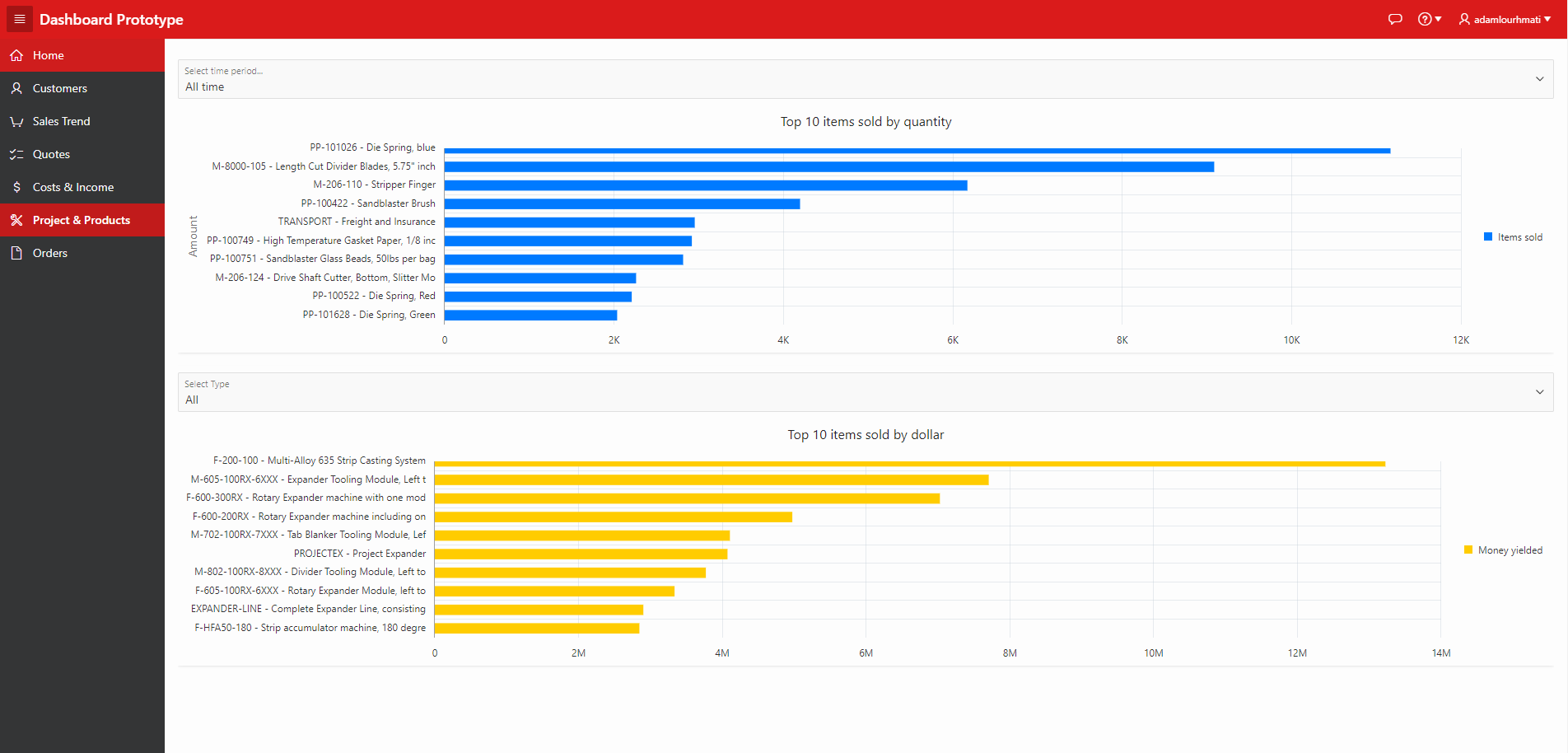
Finalement, en bonus, il y aurait une autre page pour visualiser les estimations des couts des projets et où il y aurait un graphique pour comparer les objectifs de ventes et les ventes réalisées.

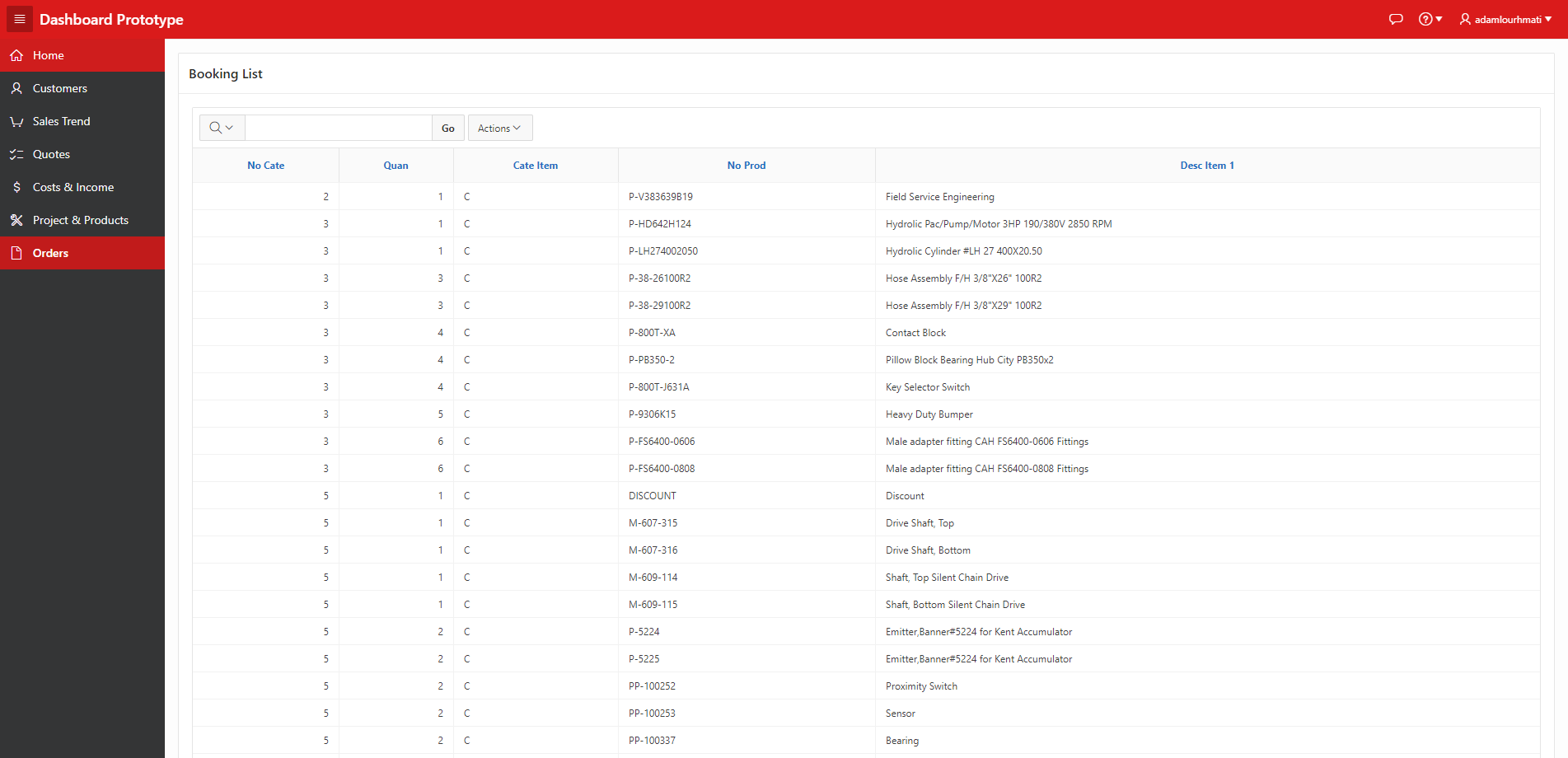










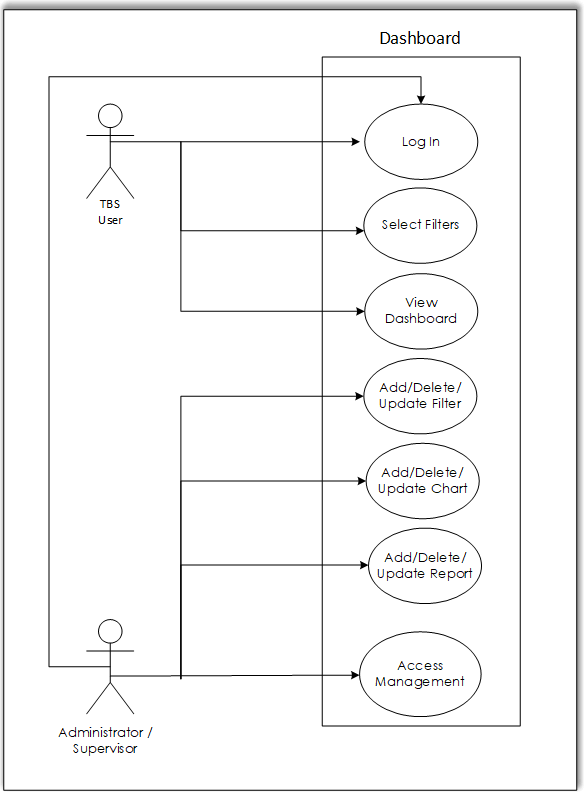


# Cas d’utilisation

Cette section se concentre plus en détail sur les fonctionnalités qu’offre le système. Le diagramme des cas d’utilisation donnera un aperçu des différentes interactions entre les utilisateurs et le système. Par la suite, dans le même document, chacune des interactions dans ce diagramme sera détaillée dans un cas d’utilisation.

## Diagramme des cas d’utilisation

Un diagramme de cas d’utilisation est un diagramme UML servant à donner une vision globale des différentes interactions entre l’utilisateur (humain ou machine) et le système. Dans le diagramme, chaque unité (une ellipse) correspond à un cas d’utilisation, qui correspond à une interaction entre l’utilisateur et le système. Prenons par exemple, le cas d’utilisation View Dashboard, notre utilisateur TBS va interagir avec le système en visualisant le tableau de bord. Ce type de diagramme est souvent utilisé lors des présentations de projets auprès des clients car il permet d’avoir une vue globale sur les différents acteurs (utilisateurs) et quelles interactions ils peuvent avoir avec le système. Bref, le diagramme suivant donne un aperçu de l’ensemble des fonctionnalités du système, qui seront détaillées dans les pages suivantes à travers les cas d’utilisation.



*Figure 13 : diagramme des cas d'utilisation.*

## Cas d’utilisation

Un cas d’utilisation permet de décrire précisément une interaction entre un utilisateur et le système. L’utilisateur veut atteindre un objectif lorsqu’il interagit avec le système. C’est le flux d'événements, dans le cas d’utilisation, qui va décrire les différentes étapes et interactions qui vont permettre à l’utilisateur d’atteindre son objectif. Il y aussi un ou plusieurs flux alternatifs, lorsqu’il y a des erreurs. Chaque cas d’utilisation possède aussi un identifiant unique pour pouvoir s’y référer, et peut avoir des exigences spéciales, pré-conditions ou post-conditions.

## **Cas d’utilisation 1** – S’authentifier

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **CU-01** | **Nom du cas d’utilisation** | **S’authentifier** | **Version** | **1.0** |
| **Brève description** | | Un utilisateur ou un administrateur souhaite s’authentifier au tableau de bord. | | | |
| **Acteurs** | | Utilisateur TBS, Administrateur | | | |
| **Exigences spéciales** | | | | | |
| Exigence spéciale 1. | | Aucune | | | |
| **Pré-conditions** | | | | | |
| Pré-condition 1. | | L’acteur doit avoir un compte déjà défini dans le système | | | |
| **Post-conditions** | | | | | |
| Post-condition 1. | | Aucune | | | |
| **Flux d’évènements** | | | | | |
| Flux de base | | 1. L’utilisateur ou l’administrateur se connecte à la page de login. 2. Entre les informations demandées (nom d’utilisateur, mot de passe) par le système et envoie la demande de connexion. 3. Le système valide les informations et traite la demande. 4. Le système envoie une confirmation à l’utilisateur. | | | |
| Flux alternatif 1 :  Informations invalides | | 1. Le système valide le formulaire et détecte une erreur quant aux informations fournies. 2. Le système retourne le formulaire à l’utilisateur en incluant un message d’erreur. 3. Retour au point deux du flux de base. | | | |

## 

## Cas d’utilisation 2 – Sélectionner un filtre

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | CU-02 | **Nom du cas d’utilisation** | Sélectionner un filtre | **Version** | 1.0 |
| **Brève description** | | Un utilisateur souhaite sélectionner un filtre afin d’afficher les informations selon une période de temps. | | | |
| **Acteurs** | | Utilisateur TBS | | | |
| **Exigences spéciales** | | | | | |
| Exigence spéciale 1. | | Aucune | | | |
| **Pré-conditions** | | | | | |
| Pré-condition 1. | | L’utilisateur doit s’authentifier. | | | |
| **Post-conditions** | | | | | |
| Post-condition 1. | | Le filtre est désormais ajusté selon la période de temps choisie | | | |
| **Flux d’évènements** | | | | | |
| Flux de base | | 1. L’utilisateur accède à la page d’accueil. 2. L’utilisateur ajuste le filtre selon la période de temps voulue 3. Le système valide les modifications et les traite. | | | |
| Flux alternatif 1 :  Période de temps non valide | | 1. Le système valide les modifications et détecte une erreur quant au filtre défini. 2. Le système retourne un message d’erreur à l’utilisateur et lui demande d’ajuster le filtre. 3. Retour au point deux du flux de base. | | | |

## Cas d’utilisation 3 – Visualiser le tableau de bord

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **CU-03** | **Nom du cas d’utilisation** | **Visualiser le tableau de bord** | **Version** | **1.0** |
| **Brève description** | | Un utilisateur souhaite afficher le tableau de bord. | | | |
| **Acteurs** | | Utilisateur TBS | | | |
| **Exigences spéciales** | | | | | |
| Exigence spéciale 1. | | Aucune. | | | |
| **Pré-conditions** | | | | | |
| Pré-condition 1. | | L’utilisateur doit s’authentifier. | | | |
| **Post-conditions** | | | | | |
| Post-condition 1. | | Aucune | | | |
| **Flux d’évènements** | | | | | |
| Flux de base | | 1. Cas d’utilisation : sélectionner un filtre. 2. L’utilisateur accède à la page voulue. 3. Le système valide la demande d’accès et affiche les informations de la page sélectionnée. | | | |
| Flux alternatif 1 :  Connexion interrompue | | 1. La connexion est interrompue avec le serveur. 2. Le système retourne un message d’erreur à l’utilisateur. 3. Retour au point deux du flux de base. | | | |

## Cas d’utilisation 4 – Ajouter/Modifier/Supprimer un filtre

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **CU-04** | **Nom du cas d’utilisation** | **Ajouter/Modifier/Supprimer un filtre** | **Version** | **1.0** |
| **Brève description** | | L’administrateur souhaite ajouter, modifier ou supprimer un filtre | | | |
| **Acteurs** | | Administrateur du système | | | |
| **Exigences spéciales** | | | | | |
| Exigence spéciale 1. | | Aucune | | | |
| **Pré-conditions** | | | | | |
| Pré-condition 1. | | L’administrateur doit s’authentifier. | | | |
| **Post-conditions** | | | | | |
| Post-condition 1. | | Le filtre est désormais ajouté, modifié ou supprimé. | | | |
| **Flux d’évènements** | | | | | |
| Flux de base | | 1. L’administrateur accède à la fonction de modification du filtre. 2. L’administrateur ajoute, modifie ou supprime le filtre voulu 3. Le système valide les modifications et les traite. | | | |
| Flux alternatif 1 :  Données inexistantes dans la base de données | | 1. Le système ne trouve pas les données voulues. 2. Le système retourne un message d’erreur. 3. Retour au point deux du flux de base. | | | |

## 

## Cas d’utilisation 5 – Ajouter/Modifier/Supprimer un graphique

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **CU-05** | **Nom du cas d’utilisation** | **Ajouter/Modifier/Supprimer un graphique** | **Version** | **1.0** |
| **Brève description** | | L’administrateur souhaite ajouter, modifier ou supprimer un graphique | | | |
| **Acteurs** | | Administrateur du système | | | |
| **Exigences spéciales** | | | | | |
| Exigence spéciale 1. | | Aucune | | | |
| **Pré-conditions** | | | | | |
| Pré-condition 1. | | L’administrateur doit s’authentifier. | | | |
| **Post-conditions** | | | | | |
| Post-condition 1. | | Le graphique est désormais ajouté, modifié ou supprimé. | | | |
| **Flux d’évènements** | | | | | |
| Flux de base | | 1. L’administrateur accède à la fonction de modification du graphique. 2. L’administrateur ajoute, modifie ou supprime le graphique voulu 3. Le système valide les modifications et les traite. | | | |
| Flux alternatif 1 :  Données inexistantes dans la base de données | | 1. Le système ne trouve pas les données voulues. 2. Le système retourne un message d’erreur. 3. Retour au point deux du flux de base. | | | |

## Cas d’utilisation 6 – Ajouter/Modifier/Supprimer un rapport

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **CU-06** | **Nom du cas d’utilisation** | **Ajouter/Modifier/Supprimer un Rapport** | **Version** | **1.0** |
| **Brève description** | | L’administrateur souhaite ajouter, modifier ou supprimer un rapport | | | |
| **Acteurs** | | Administrateur du système | | | |
| **Exigences spéciales** | | | | | |
| Exigence spéciale 1. | | Aucune | | | |
| **Pré-conditions** | | | | | |
| Pré-condition 1. | | L’administrateur doit s’authentifier. | | | |
| **Post-conditions** | | | | | |
| Post-condition 1. | | Le rapport est désormais ajouté, modifié ou supprimé. | | | |
| **Flux d’évènements** | | | | | |
| Flux de base | | 1. L’administrateur accède à la fonction de modification du rapport. 2. L’administrateur ajoute, modifie ou supprime le rapport voulu 3. Le système valide les modifications et les traite. | | | |
| Flux alternatif 1 :  Données inexistantes dans la base de données | | 1. Le système ne trouve pas les données voulues. 2. Le système retourne un message d’erreur. 3. Retour au point deux du flux de base. | | | |

## Cas d’utilisation 7 – Gérer les droits d’accès

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **CU-07** | **Nom du cas d’utilisation** | **Gérer les droits d’accès** | **Version** | **1.0** |
| **Brève description** | | L’administrateur souhaite gérer les droits d’accès au tableau de bord | | | |
| **Acteurs** | | Administrateur du système | | | |
| **Exigences spéciales** | | | | | |
| Exigence spéciale 1. | | Aucune | | | |
| **Pré-conditions** | | | | | |
| Pré-condition 1. | | L’administrateur doit s’authentifier. | | | |
| **Post-conditions** | | | | | |
| Post-condition 1. | | Les droits d’accès sont désormais modifiés. | | | |
| **Flux d’évènements** | | | | | |
| Flux de base | | 1. L’administrateur accède à la fonction de modification des droits d’accès. 2. L’administrateur modifie les droits d’accès 3. Le système valide les modifications et les traite. | | | |
| Flux alternatif 1 :  Connexion interrompue | | 1. La connexion est interrompue avec le serveur. 2. Le système retourne un message d’erreur à l’utilisateur. 3. Retour au point deux du flux de base. | | | |

# Échéancier

Voici un échéancier couvrant la réalisation du projet, sous forme d’un diagramme de Gantt. Le diagramme de Gantt est un outil de gestion de projet qui permet de visualiser dans le temps les grandes tâches d’un projet. En bref, il s’agit d’une représentation, sous forme de graphique, de l’avancement du projet. Le diagramme de Gantt sera amélioré (complété plus précisément) au fur et à mesure que le projet avance.

Les grandes étapes du projet sont les suivantes :

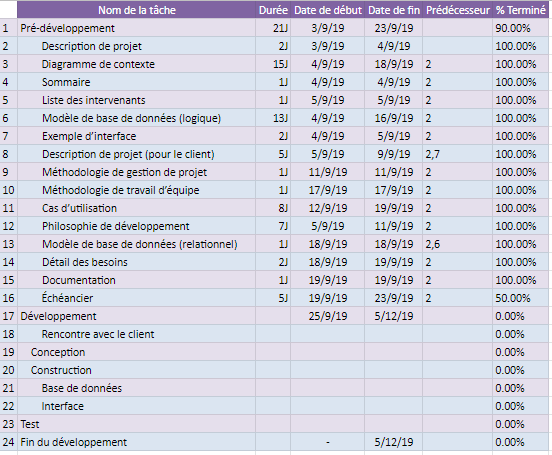
**Pré-développement :** Suite à une analyse des besoins du client *TBS Engineering* ainsi que des différentes informations nécessaires au projet, une proposition de travail du projet de tableau de bord lui sera présentée afin qu’il puisse le valider ou le rejeter.

**Développement :** Après une rencontre avec le client et qu’il ait approuvé la proposition de travail, la phase de la conception du tableau de bord pour les ventes de la compagnie pourra débuter.

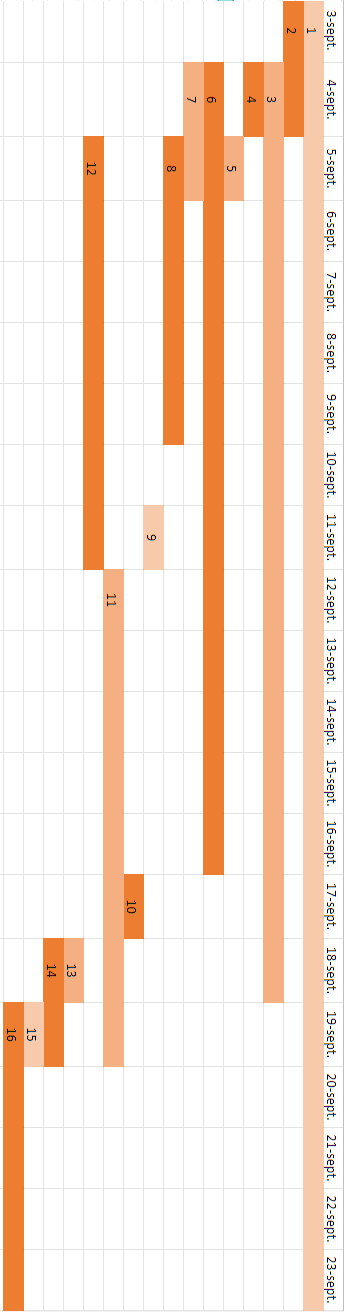
* **Conception :** Analyse des données et des traitements à effectuer.
* **Construction :** La phase de réalisation du projet. On commence le développement du tableau de bord.
  + **Interface :** Réalisation de l’interface qui va composer le tableau de bord.
  + **Base de données :** Table de requête, lien avec environnement Integra.

**Test :** Vérification du bon fonctionnement, des performances et de la fiabilité du tableau de bord.

**Fin du développement :** La phase de livraison du tableau de bord au client afin qu’il puisse valider si le produit répond aux exigences.



*Figure 14 : index du diagramme.*



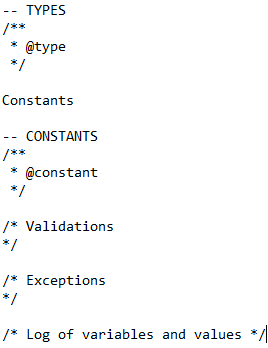
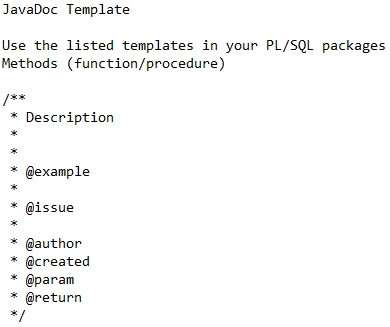
*Figure 15 : diagramme de Gantt.*

# Bibliographie

* « Code to Logging Ratio? [Closed] ». 2008. Stack Overflow.<https://stackoverflow.com/questions/153524/code-to-logging-ratio>.
* JBM Logic. « Business Integra ». Consulté le 3 octobre 2019.<http://www.jbmlogic.com/>.
* TBS engineering. « TBS Engineering ». Consulté le 3 octobre 2019.<http://www.tbseng.co.uk/>.
* Boyer, Clément. 2017. « Méthode Scrum en bref ». SUPINFO International University.<https://www.supinfo.com/articles/single/6150-methode-scrum-bref>.
* « Diagramme des cas d’utilisation ». 2019. Wikipédia.<https://fr.wikipedia.org/wiki/Diagramme_des_cas_d%27utilisation>.
* « Cas d’utilisation ». 2019. Wikipédia.<https://fr.wikipedia.org/wiki/Cas_d%27utilisation>.
* Roel, Carina. 2019. « Débutez l’analyse logicielle avec UML ». OpenClassrooms.<https://openclassrooms.com/fr/courses/2035826-debutez-lanalyse-logicielle-avec-uml/2063941-etape-1-definir-le-contexte-du-futur-logiciel>.
* H. Bouquin, Le Contrôle de gestion, Paris : Presses Universitaires de France, 8e édition, 2008
* Probablement Michel Gervais, Contrôle de gestion et planification de l’entreprise, 2000 (7e édition), page 598
* Le tableau de bord prospectif, Robert S. Kaplan, David P. Norton, Éditions d'Organisation, 2003
* « Le Diagramme de Contexte ». s. d. Système d’information de gestion. Consulté le 16 septembre 2019.<https://geainfolemans.github.io/SGBD/co/contexte.html>.
* Le tableau de bord facile, Daniel Boix, Bernard Féminier, Éditions d'Organisation, 2004
* « SysML- Diagramme de contexte ». 2016.<http://tvaira.free.fr/dev/sysml/sysml-contexte.pdf>.
* Ivar Jacobson, « Object-oriented Development in an Industrial Environment », SIGPLAN Not., vol. 22, no 12,‎ décembre 1987, p. 183–191 (ISSN 0362-1340, DOI 10.1145/38807.38824, https://dl.acm.org/citation.cfm?doid=38807.38824, consulté le 12 juin 2019

# Annexes

## Annexe A



## Annexe B

