

# Rapport de stage :

CERDI : Centre d'Études et de Recherches sur le Développement International



**Transfert d'un modèle de Valeur actualisée nette (VAN) développé sur Excel vers une page web : Une application au taux effectif moyen d'imposition (TEMI) en Afrique.**

Matthias GAYDU



IUT de Clermont-Ferrand

Département Informatique

Année 2017-2018

**Autorisation de diffusion :**

J'autorise la diffusion de mon rapport sur l'intranet de l'IUT

## **Remerciement :**

Je remercie le CERDI (UCA-CNRS) de m'avoir accueilli en tant que stagiaire informatique du 3 mai au 6 juillet 2018 et de m'avoir accordé sa confiance pour la réalisation de ce projet de transfert d'un modèle de valeur actualisée nette développé sur Excel vers une page WEB.

Je remercie Grégoire Rota Graziosi d'avoir dirigé ce stage ainsi que l'équipe du CERDI pour avoir été là lorsque j'en avais besoin.

Je remercie Florent Madelaine de m'avoir suivi et accompagné en tant qu'encadrant.

## Sommaire :

### Intro

#### I) Présentation de l'entreprise

#### II) Présentation du stage

##### 1- Environnement

##### 2- Existant

##### 3- Objectif

##### 4- Cibles

#### III) Développement

##### 1- Emploi du temps

###### a) date début-fin

###### b) Gantt

- 1 agile+ reverse engineering

- 2 Gantt prévisionnel

- 3 Gantt réel

- 4 analyse des écarts

##### 2- Établissement d'un choix technologique

##### 3- Le modèle existant : sa structure

- a) La base fiscale

- b) La base économique

- c) La création des données du modèle

##### 4- Le modèle créé

- a) Explication du modèle (diagramme class un peu moins complet que celui dans doc)

- b) Explication du détail de certain calcul (je parle des calculs clé comme le calcul des valeurs actuelles nettes, taux de rendement interne, taux d'effectif marginaux)

- c) API utilisée

##### 5-Des objectifs graphiques

##### 6- Différents cas d'utilisation de l'application (2 en fait)

- a) Premier cas de résultat avec le résultat entre 1 entreprise et 1 pays mais avec du détail

- b) Deuxième cas avec le graphique de comparaison et le tableau des TEMI (sans détails)

## Introduction :

J'effectue mon stage au sein du CERDI à Clermont-Ferrand. Le CERDI est un centre de recherche et d'étude (un établissement de recherche) qui entretient de multiples relations de collaboration scientifique ou d'expertise avec la Banque mondiale, le Fonds monétaire international, la Commission de l'Union Européenne, l'Agence française du développement ou le Ministère de l'Europe et des affaires étrangères.

Le CERDI a créé un modèle sur Excel sur les valeurs actualisées nettes qui permet de simuler les taux effectifs moyens d'imposition dans douze pays Africains situés en Zone franc. Ce modèle qui suit l'approche de la Banque Mondiale (Doing Business) a été adapté à la fiscalité des pays de l'Union économique et monétaire de l'Ouest africain (UEMOA : Bénin, Burkina Faso, Côte d'Ivoire, Mali, Niger, Sénégal, Togo) et de la Communauté économique et monétaire de l'Afrique Centrale (CEMAC : Cameroun, Congo, Gabon, République de Centrafrique, Tchad).

L'outil de simulation existant prend la forme d'un fichier Excel composé d'une base de données et d'un modèle de flux financiers sur cinq années (valeur actuelle nette) d'une entreprise en distinguant les principaux impôts et le profit net de l'investisseur. Le fichier Excel en pièce jointe présente le modèle, les caractéristiques de l'entreprise et la fiscalité des 16 pays africains.

L'objectif de ce stage est donc de prendre ce modèle qui est développé sur Excel et de le mettre sur une page web.

J'ai donc opté de refaire le modèle en mettant en œuvre un site entièrement front-end (c'est-à-dire que les calculs et les données n'ont pas besoin d'un serveur pour se faire). Cette page doit être rattachée au site internet comme étant un lien de page et a pour finalité de permettre aux personnes qui utilisent ce modèle de ne pas être obligées de demander au CERDI le modèle Excel, mais d'y accéder directement sur son site.

La demande du maître de stage était d'ajouter une visualisation des TEMI par histogramme afin que l'on puisse voir les TEMI d'une entreprise sur les pays ainsi qu'une extraction en Excel (xls) des valeurs des TEMI afin de permettre à un chercheur ou décideur public d'utiliser les résultats de leur simulation.

## I) Présentation de l'entreprise



Cette image est le logo officiel du centre CERDI

Le CERDI est un centre de recherches dédié à l'étude des processus de développement international. Il a été fondé en 1976 et il rassemble des économistes, chercheurs du CNRS ou enseignants-chercheurs de l'Université Clermont Auvergne, dont les domaines de compétences sont variés et couvrent aussi bien les aspects microéconomiques que macroéconomiques du développement. Les pays à revenu faible ou intermédiaire constituent le champ d'application des travaux menés au CERDI.

Des formations (Magistère, Masters, Doctorat) spécialisées en développement international sont adossées au CERDI et bénéficient d'un lien étroit entre l'enseignement et la recherche.

Le CERDI travaille de manière régulière avec les institutions nationales (Agence Française de Développement, Ministères), régionales (banques de développement, institutions d'intégration régionale) et internationales (Nations Unies, Fonds Monétaire International, Banque Mondiale) en charge des politiques de développement et de leur financement.

Le CERDI collabore avec la Fondation pour l'Etude et la Recherche sur le Développement International (FERDI) et l'Institut du Développement Durable et des Relations Internationales (IDDRI) dans le cadre du Laboratoire d'Excellence (Labex) IDGM+.

Ce laboratoire a pour but de promouvoir une recherche d'excellence afin de réexaminer les politiques de développement international.

Il a été évalué A+ par l'Agence d'Evaluation de la Recherche et de l'Enseignement Supérieur en 2011 (rapport d'évaluation).

Le CERDI entretient de multiples relations de collaboration scientifique ou d'expertise avec la Banque Mondiale, le Fonds Monétaire International, la Commission de l'Union Européenne, l'Agence Française du Développement ou le Ministère de l'Europe et des Affaires Etrangères.

Le CERDI est actuellement le premier pourvoyeur français de fonctionnaire internationaux, depuis sa création, une cinquantaine de doctorants du CERDI travaillent et ont travaillé pour les deux institutions de Bretton Woods : le FMI (fonds monétaire international) et la banque mondiale (Doing Banque).

Le CERDI est composé de 30 enseignant-chercheurs et de 82 doctorants.

Cette établissement fait des recherches sous trois grand axes :

- Le financement du développement
- Les trajectoires de développement durable
- Intégration des pays en développement dans l'économie mondialisée

Le stage se déroule sur de l'existant fait à partir du troisième axe c'est-à-dire sur des données venant de recherche faite sous l'axe d'une intégration des pays en développement dans l'économie mondial.

## II) Présentation du stage

### 1) Environnement

Le service dans lequel je suis installé est une « bulle » permettant au doctorants, stagiaires, étudiants de travailler. C'est ce qu'on appelle une salle en open-space constitué d'un côté de grandes tables, de l'autre des bureaux individualisés séparés par des sous-cloisons. Je suis du côté des bureaux individualisés.

Les outils utilisés pour ce projet sont :

- ➔ Web Storm IDE, un logiciel qui permet de coder en JavaScript, html et un « sous-Logiciel » de PHP Storm entrant dans la gamme d'intelliJ.
- ➔ Notepad++ au tout début, Un éditeur de texte open source
- ➔ StarUML pour faire la partie analyse UML du projet
- ➔ Team Gantt un site internet permettant de faire les différents Gantt
- ➔ ENT UCA mail pour communiquer avec mes différents tuteurs
- ➔ GitHub me permettant d'avoir un répertoire distant et accessible de partout.

Pour vous donner un historique de l'utilisation des outils, j'ai commencé à coder avec Notepad++. Mais même si l'éditeur implémente un code couleur pour les langages utilisés, il est loin de faciliter la programmation.

Pour la conception, j'ai utilisé StarUML pour faire les différents diagrammes UML.

J'ai rapidement pris la décision de regarder quelles IDE me permettraient d'avoir un confort de programmation certain pour continuer de coder. Je me suis rapidement intéressé à la gamme des IDE de IntelliJ car j'avais déjà utilisé des IDE de cette gamme et je la trouve bien faite pour des codes (Ils ont des IDE spécifiques à la majorité des langages de programmation connus). J'ai donc cherché l'IDE spécifique au langage du projet (JavaScript). Et j'ai trouvé WEB Storm. Le problème de l'IDE en lui-même, c'est qu'il faut une licence mais il s'avère que Web Storm, est intégré dans un IDE de la gamme nommé PHP Storm, dont j'avais obtenu une License par le biais de ma formation peu de temps avant. Je pouvais donc utiliser cette IDE. J'ai rapidement cherché des modules pour pouvoir absolument tout faire à partir de l'IDE (création de diagramme UML avec le module plantUML...).

De plus l'IDE a des fonction d'intégration VCS (pour les répertoires distants) permettant la facilitation de l'outil GitHub.

L'outil Team Gantt est une interface web permettant de faire des Gantt pour des projets. L'outil est simple d'utilisation et propose des Gantt suffisamment clairs, le seul bémol est que le Gantt qu'il génère est accompagné du logo sûrement pour des raisons commerciales (L'objectif du site est lucratif avec une version premium).



L'outil UCA mail avec la messagerie Zimbra est un outil qui est connu de tous, c'est le seul outil mis en place pour la communication distante avec les différents tuteurs de ce stage que ce soit avec mon maître de stage au CERDI ou les tuteurs de l'IUT. L'outil en lui-même est juste une messagerie (une boîte mail).

GitHub permet d'avoir un répertoire distant. Pour faire simple c'est une zone de stockage distante. Pour faire compliqué, GitHub propose des serveurs Distant pour l'utilisation de l'outil GIT, ce dernier est un logiciel qui permet de stocker un ensemble de fichiers tout en permettant de conserver la chronologie de toutes les modifications effectuées : un VCS (Version Control System). Cet outil m'a permis d'avoir mon code sur le PC fixe mis à disposition par le CERDI et d'autres outils que j'utilise tels que mon PC personnel ou autre. De plus, GitHub permet le téléchargement en fichier compressé zip permettant d'échanger directement avec des personnes qui peuvent télécharger et tester le projet.

## 2) Existant

Un modèle a été développé sur Excel pour apprécier le taux effectif moyen d'imposition d'une entreprise dans les 16 pays de la zone Franc (Afrique). Ce modèle qui suit l'approche de la Banque Mondiale (Doing Business) a été adapté à la fiscalité des pays de l'Union économique et monétaire de l'Ouest Africain (UEMOA : Bénin, Burkina Faso, Côte d'Ivoire, Guinée Bissau, Mali, Niger, Sénégal, Togo) et de la Communauté économique et monétaire de l'Afrique Centrale (CEMAC : Cameroun, Congo, Gabon, Guinée Equatoriale République de Centrafrique, Tchad).

L'outil de simulation existant prend la forme d'un fichier Excel composé d'une base de données et d'un modèle de flux financiers sur cinq années (valeur actuelle nette) d'une entreprise en distinguant les principaux impôts et le profit net de l'investisseur. Le fichier Excel en pièce jointe présente le modèle, les caractéristiques de l'entreprise et la fiscalité des 16 pays africains. D'autres langages informatiques pourraient être utilisés.

### 3) Objectif

Le CERDI souhaite mettre en ligne un modèle qui permet de simuler la charge fiscale effective qu'une entreprise supporte. Ce modèle a été développé pour apprécier le taux effectif moyen d'imposition d'une entreprise dans les 12 pays de la zone Franc (Afrique). Ce modèle qui suit l'approche de la Banque Mondiale (Doing Business) a été adapté à la fiscalité des pays de l'Union Economique et Monétaire de l'Ouest Africain (UEMOA : Bénin, Burkina Faso, Côte d'Ivoire, Mali, Niger, Sénégal, Togo) et de la Communauté Economique et Monétaire de l'Afrique Centrale (CEMAC : Cameroun, Congo, Gabon, République de Centrafrique, Tchad). Il pourrait être étendu aux autres pays de l'espace OHADA.

L'objectif de ce projet est donc de mettre en place ce modèle développé sur Excel sur une interface WEB ce qui permettrait un accès direct à ce modèle et une plus grande visibilité.

D'un point de vue quantitatif, on aura besoin de voir le modèle complet et donc le site est constitué de 3 pages :

La première sera constituée d'un formulaire avec les paramètres d'entrées. Dès que ce formulaire sera validé, sera affiché un récapitulatif des données de l'entreprise simulées avec les coefficients de Djankov ainsi que les données fiscales du pays sélectionné.

Une deuxième page avec le modèle en lui-même de l'entreprise en fonction du pays, elle permettra de voir le TEMI ainsi que les flux de trésorerie générés par l'entreprise ainsi que les Valeurs actuelles nettes générées par ces derniers.

Une troisième page générera un graphique avec les pourcentages du taux effectif moyen d'imposition de différents pays et entreprises sélectionnés préalablement dans un nouveau formulaire. Ce graphique sera accompagné d'un tableau des valeurs de TEMI qui pourrait être exporté au format Excel.

L'objectif du modèle est de permettre à un utilisateur de faire une simulation du TEMI avec les coefficients de Djankov dans 12 pays d'Afriques.

Il y aura une dernière page qui permettra d'ajouter un nouveau pays avec des taux que l'utilisateur devra entrer.

#### 4) Cibles du projet

Les cibles de ce projet sont donc des entrepreneurs, des entreprises qui souhaitent s'installer dans les pays d'Afrique (UEMOA et CEMAC) afin de leur permettre de savoir dans quels pays, d'un point de vue fiscal, il vaudrait mieux installer leur entreprise.

Autres cibles du projet, les gouvernements de ces pays, cela leur permet de vérifier si leur régime fiscal leur permet d'attirer les entreprises plus ou moins que leurs voisins, ou voire par rapport au reste du monde. Cela leur permet donc d'adapter leur régime fiscal afin d'être plus ou moins attractif aux yeux des entrepreneurs du monde entier.

Les dernières cibles sont les populations locales de ces pays, afin de leur permettre d'entreprendre et afin d'évaluer les risques engendrés par les charges fiscales, ils auront directement le pourcentage imposable.

### III) Développement

#### 1) Emploi du temps

##### a) Les dates

Le stage a commencé le mercredi 2 mai 2018 dans les locaux du CERDI entre huit heures et neuf heures.

Un rapport de stage doit être rendu le mercredi 13 juin 2018 au professeur référent du stage, au maître de stage et au Jury de la soutenance constituée de professeurs.

Le stage se termine le vendredi 6 juillet 2018 à dix-huit heures dans les locaux du CERDI.

Une soutenance de stage est prévue le 5 juillet 2018 à 10 heure afin d'évaluer les compétences utilisées lors du stage.

##### b) Gantt

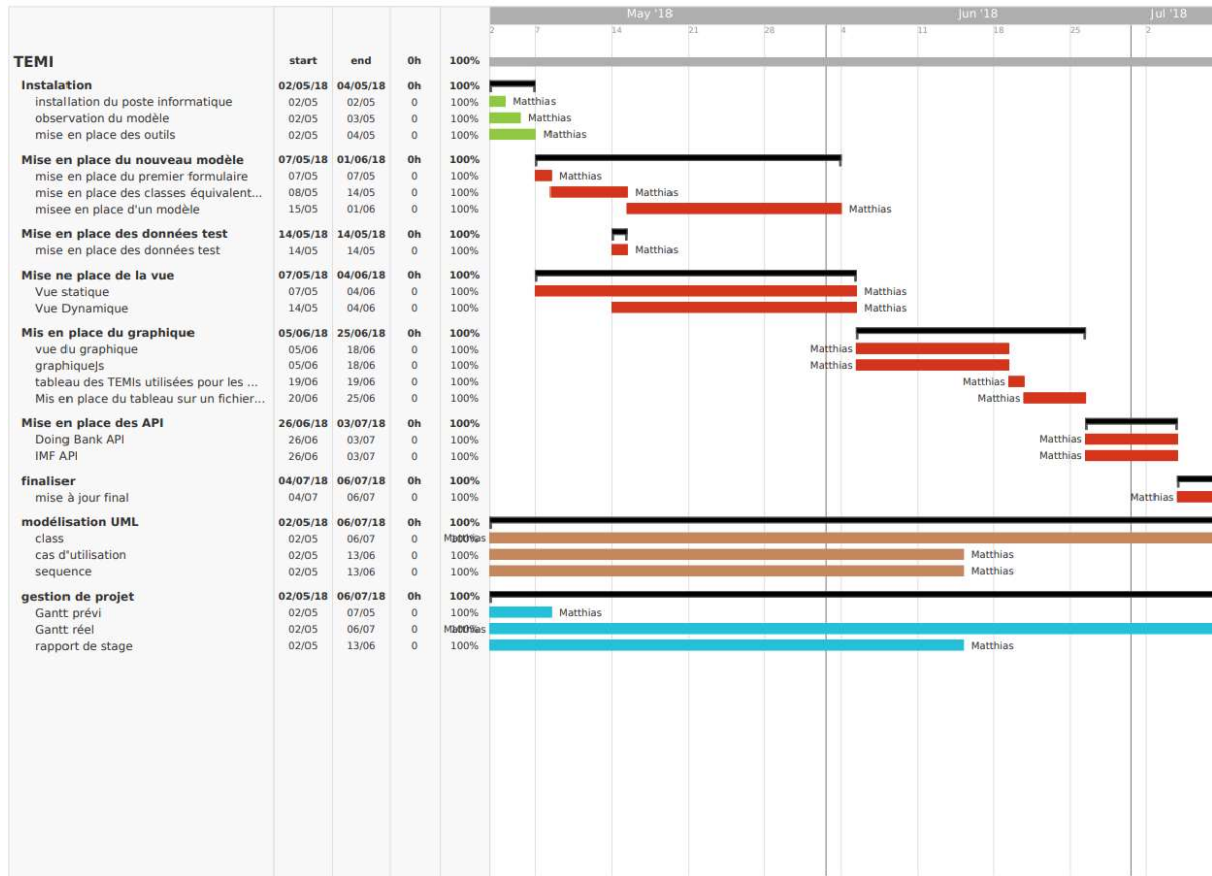
##### 1) Méthodes utilisées

Pour faire le site, j'ai utilisé le reverse engineering :

Les objectifs du site étaient détaillés dans un document contexte rédigé par le maître de stage, mais j'avais du mal à imaginer le rendu que pourrait avoir le site ni même la technologie avec laquelle j'allais faire cette interface web. J'ai pensé à utiliser cette méthode pour me permettre d'adapter la technologie et donc le modèle que je vais mettre en place en fonction des limites des technologies utilisées avant que les difficultés (dû aux limites) m'empêchent d'avancer. Pour les rendus des vues, la partie statique de ces dernières sont codées avec un langage balisé et simple nommé HTML, du coup, elles permettent des changements rapides favorisant la mise en place de cette méthode.

Etant le seul à travailler sur le projet, une gestion de projet telle qu'un GANTT prévisionnel ne fut pas concrètement mis en place lors du lancement du projet mais fut quand même pensé. Cette réflexion a été modélisée à la fin du projet pour en définir les écarts. La méthode utilisée n'est pas concrètement une méthode AGILE mais s'en rapproche. Un emploi du temps des tâches fut mis en place puis pensé et adapté en fonction des avancées et des retards tout le long du projet.

## 2) Gantt Prévisionnel



Cette image représente le Gantt prévisionnel, elle fut traitée par Team Gantt et rognée par mes soins

Étant donné que le Gantt prévisionnel ne fut pas concrètement pensé dès le début, il a été modélisé avec les temps limites équivalents pensée lors du départ du projet, chaque phase étant réfléchie avant chaque tâche. Les phases ont été adaptées à la durée du stage.

On peut voir dans ce Gantt prévisionnel les grandes parties organisationnelles de ce projet :

- ➔ L'installation ;
- ➔ Le développement ;
- ➔ La modélisation ;
- ➔ La gestion du projet en lui-même.

J'avais prévu à peu près 3 jours pour l'installation étant donné que je ne connaissais pas les technologies que j'allais utiliser : entre les déterminer, trouver les meilleurs outils accessibles pour ces technologies, les installer et évaluer rapidement les options, j'ai pensé que 3 jours étaient nécessaires pour installer tous les outils nécessaires au développement.

La modélisation et la gestion de projet doivent se faire tout le long du projet, on ne peut pas modéliser concrètement un modèle sans penser à toutes les spécificités des langages qui sont utilisés lors de l'élaboration du projet.

Le développement doit se faire uniquement après avoir installé son poste de travail (programmer avec des outils non adaptés c'est comme enfoncer un clou sans marteau, c'est beaucoup plus long et fastidieux à faire).

On peut voir que le développement se fait en plusieurs étapes :

- ➔ La mise en place du modèle devrait être effectuée du 7 mars au 1 juin. C'est la phase principale du projet, elle permet, à partir des paramètres présélectionnés par l'utilisateur, d'obtenir un résultat et une valeur du TEMI avec des résultats détaillés pour chacun des impôts étudiés.
- ➔ La mise en place de la vue : pour moi, la vue se fait du début jusqu'à la fin du projet, je l'ai programmée du 7 mai au 4 juin car je me suis laissé 2 -3 jours pour figurer le projet. Il y a deux types de vues : une vue statique, la partie purement HTML qui doit commencer dès le départ du développement jusqu'à la fin en parallèle au reste, et une partie dynamique qui commence en même temps que la mise en place des premières données.
- ➔ Il y a donc une partie dans le développement qui consiste à mettre en place des données. Cette tâche n'existant que pour utiliser réellement le modèle, elle nécessite pas peu de temps d'où le fait qu'elle soit mise sur un jour.
- ➔ La mise en place d'un graphique m'avait paru au début compliqué et mérité sa sous partie, elle doit être développée uniquement une fois que le modèle est mis en place, pour moi, elle commence donc à partir du 04-06 et dure environ 3 semaines.
- ➔ La mise en place des APIs était pour moi abordable, je ne l'avais juste jamais fait donc le temps de se documenter, j'avais prévu une semaine de mise en place.  
Elles permettent de récupérer des valeurs actualisées au dernière année utilisé par le modèle pour le TEMI
- ➔ Il me reste donc du 4 juin au 6 juin pour mettre en œuvre les dernières fonctionnalités et les dernières mises au point.

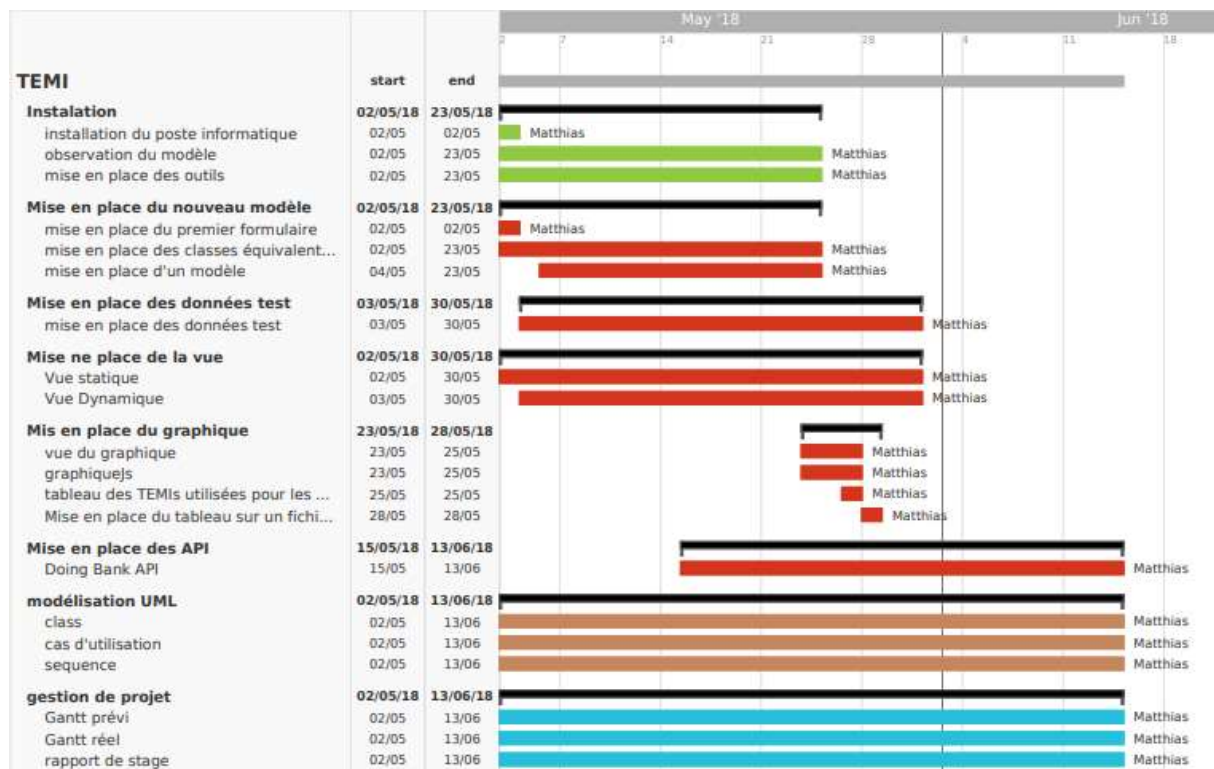
On peut voir ensuite dans la modélisation deux grandes dates : le 13 juin et le 06 juillet. En utilisant le Reverse Engineering, je n'avais pas encore d'idée de modèle. Je pouvais donc attaquer les grandes lignes de chaque diagramme mais ces derniers seraient raffinés tout le long du projet.

Pour certains diagrammes, pas besoin d'attendre la fin du projet pour avoir le diagramme complet. J'ai donc décidé que j'arrêterai de modifier ces diagrammes sur une date-clé du projet, le jour du rendu du rapport : le 13 juin 2018.

Pour d'autres diagrammes, ce ne sera pas fini tant que le développement est en cours, la modification de ces diagrammes se finira donc en même temps que le projet. Pour la gestion de projet, on voit plusieurs cas :

- ➔ La mise en place du Gantt prévisionnel: la gestion de projet se fait en parallèle au reste du projet, je n'ai pas réellement pensé à mettre en place un Gantt prévisionnel, mais le fait de me donner des dates limites à chaque début de tâche est équivalent. J'ai évalué le temps de ces estimations avec la création du Gantt à 1 semaine mais plus pour prévenir que c'est espacé dans le temps que fait de façon intensive.
- ➔ Le Gantt réel de n'importe quel projet est mis à jour chaque fois qu'une tâche est finie et/ou commence. Il permet d'avoir une estimation réelle du temps passé sur les différentes tâches et non une approximation comme le Gantt prévisionnel. Il doit donc être mis à jour tout le long du projet et ce jusqu'à la fin de ce dernier.
- ➔ Le rapport de stage : une tâche évaluée permettant à un jury d'évaluer la qualité du stage, et d'avoir un aperçu sur le travail effectué lors de ce stage. Ce dernier document doit et sera rendu le 13 juin 2018, il sera donc commencé dès le début du stage et sera terminé le jour où il devra être rendu.
- ➔ Il y a aussi la soutenance d'évaluation du stage, mais cette dernière se situant fin aout, elle n'est pas accessible par le Gantt.

### 3) Gantt Réel



Cette image représente le Gantt réel, elle fut traitée par Team Gantt et rognée par mes soins

Le Gantt Réel actuel n'est pas celui qui est définitif.

Les tâches non terminées sont :

- ➔ L'API de la Banque mondiale
- ➔ Le diagramme de classe
- ➔ Le Gantt Réel

On peut y voir les mêmes tâches que sur l'ancien mais avec des tâches supprimées comme par exemple, les finalités.

Comme pour finaliser, il faut avoir terminé le projet et que je ne l'ai pas fini, je n'ai pas pu les mettre dans le Gantt.

Le Gantt Réel se termine le 13 juin car je ne peux pas faire un Gantt réel sur des événements qui se passent dans le futur. On peut y voir un Gantt Réel très différent du Gantt prévisionnel à cause d'une désorganisation sûrement due au fait que je suis seul et que ma curiosité m'a fait attaquer beaucoup de chose en même temps, les différences seront expliquées dans la partie analyse ci-dessous.

#### 4) Analyse

On va analyser les écarts par partie :

Au niveau de l'installation, un jour d'observation sur l'ancien modèle n'est pas suffisant. J'ai réellement eu besoin d'analyser et de comprendre le modèle tout au long de la réalisation du modèle.

La mise en place de tous les outils se fait rarement dès le départ du projet, car on est souvent confronté à des problèmes inattendus qui font choisir de nouveaux outils plus tard. Dans mon cas, les derniers outils sont des modules pour faire le tableau Excel et le graphique qui ont été mise en place lors du début de ces deux tâches.

Ces deux écarts de date sont dus aux faits expliqués précédemment.

Au niveau de la gestion de projet, le Gantt réel ne peut pas donner des tâches qui ne se sont pas encore déroulé et donc ne peut dépasser la date du rendu de ce rapport.

Le Gantt prévisionnel est quelque chose qui fut instauré en y posant des limites. Il a été repensé au début de chaque nouvelle tâche, il n'était pas réfléchi et concrétisé la première semaine mais tout le long du stage.

Au niveau de la modélisation UML, étant donné que le diagramme de class s'affinera tout au long du projet mais ne peut être considéré comme terminé, on a donc l'impression que le cas change, mais en réalité, c'est juste que le Gantt Réel s'arrête le 13 juin.



Au niveau du développement, il s'avère que mon empressement m'a fait attaquer le 3 mai en même temps que l'installation de mes outils, c'est dû au fait que le PC installé, le temps d'installer les outils, j'avais utilisé mon PC personnel et j'avais réfléchi sur les différentes technologies. Etant donné que j'avais eu l'occasion de parler avec mon maître de stage et que j'avais compris la demande de ce dernier (même si je ne visualisais pas le résultat). J'avais donc compris quelles contraintes j'avais pour faire cette page Internet et j'ai donc cherché une technologie. La technologie que j'utilise m'est « venue naturellement » - j'utilise du JavaScript, HTML/CSS - probablement parce que ce sont les technologies primitives les plus courantes. Après avoir trouvé pourquoi utiliser ces technologies (les avantages qui y sont associés sont décrits plus bas dans le projet), j'ai donc commencé le développement en mettant en place le formulaire et en commençant les classes équivalentes aux bases de données présentes sur le modèle Excel.

La majorité de ces classes ont été faites en quelques jours, ce qui m'a permis de commencer le modèle en lui-même plus tôt que prévu afin de me permettre de préétablir des tests et d'observer les premiers résultats rapidement. Le modèle fut attaqué alors que les classes n'étaient pas finies car ces classes l'ont été rapidement mais modifiées tout au long de la création du modèle. Le modèle attaque donc le 4 mai au lieu du 15 mai. Cela m'a permis d'adapter les classes ont pu être adapté au modèle.

La mise en place des données étant techniquement directement associée au class cité précédemment, la mise en place des données commença rapidement mais s'éternisa parce que à chaque modification, les données devaient être mises à jour.

La vue change mais c'est dû aux dépendances expliquées dans le Gantt prévisionnel est au fait que le développement ait commencé plus tôt.

On peut voir aussi que la mise en place du graphique commence plus tôt et dure moins longtemps, le module Chart.js m'a bien arrangé, il y avait juste certains points qu'il fallait arranger et les données à agencer. Du coup, le temps imparti fut beaucoup plus court que ce qui était prévu.

L'utilisation de l'API a commencé plus tôt car je pouvais me permettre de l'attaquer et surtout qu'il y a une donnée qui me manquait. Je l'ai repris que récemment à partir du 30 mai.

## 2) Un choix technologique

Le sujet fut donné par une personne n'étant pas dans l'informatique et qui m'a très clairement spécifié que je n'avais pas de contrainte technique en m'exposant le fait que mon travail serait rattaché par le biais d'un lien url (comme si c'était un lien externe au site).

Je n'avais donc aucune contrainte ou spécification technique sur le choix de conception, ou encore de développement. Je n'avais aussi aucune contrainte technique au niveau des méthodes de travail à utiliser.

Il fallait donc que je trouve quelle technologie utiliser ou encore de quelle manière j'allais m'y prendre.

Au niveau des méthodes, j'avais du mal à me projeter sur un résultat de fin stage. Je voyais à peu près les étapes à suivre mais je ne visualisais pas le rendu final. Il a donc fallu trouver une méthode qui permet de travailler sur le projet sans en apercevoir le rendu final. Pour cela, le Reverse Engineering est la meilleure méthode car elle permet de faire la phase de développement avant la phase de conception permettant donc d'avoir le résultat avant de concevoir le modèle.

Pour la gestion du projet, je n'avais pas de contrainte écrite, la seule contrainte que j'ai, c'est le fait de réaliser le projet en 10 semaines, du 3 mai au 6 juillet. J'entrapercevais les différentes étapes du projet. Lors du développement, à chaque nouvelle étape dans l'élaboration des pages, je me fixais un temps limite sur lequel je voulais me consacrer en fonction d'un indice de difficulté estimé par le ressenti sur le type de problème que je pouvais rencontrer. Il y a donc « de base » aucune prévision réelle mise en place, on peut donc parler de méthode agile. Mais avec une mise en commun de chacun de ces temps limites sur chacune des étapes du projet, on peut voir un planning prévisionnel qui est modélisé par le Gantt présenté plus haut.

Pour le développement, Il n'y avait pas de contrainte technologique comme spécifié plus haut et il faut savoir que pour développer une page web, des langages de programmation jusqu'au Framework, il en existe une infinité plus ou moins réputés. Il a d'abord fallu que je regarde quel type de site je voulais faire. Il s'agit ici d'un modèle économique constitué principalement d'opérations simples. J'ai donc opté pour une application sans ce que l'on appelle des appels serveur car j'ai considéré que le modèle n'avait pas besoin d'une unité centrale avec des vitesses de calcul importantes pour permettre la fluidité des pages mises en place et donc j'ai développé un site entièrement Front-end et prévu aucun Back-end.

Avant de développer, il m'a fallu choisir une technologie qui permet de développer un modèle (qui est une simulation économique) sans que ce dernier ne permet pas de faire toutes les objectifs demandées dans le laps de temps imparti (par exemple, elle n'intègre pas une bibliothèque qui permet de faire des graphiques, etc....). Il a donc fallu chercher quelque chose de modulable dans un premier temps, et ensuite suffisamment réputé pour donner accès à un nombre de modules importants.

J'ai donc opté pour le JavaScript. C'est un langage de programmation réputé pour faire de la mise en place de vues dynamiques. Il est actuellement présent sur une majorité de site internet et c'est un langage complet. Il est spécialisé dans le Front-end mais peut permettre le back-end avec node.js (Si le modèle avait été trop demandeur en ressources au final, il m'aurait fallu intégrer du back-end et donc migrer vers une solution qui n'aurait pas forcément été possible avec d'autres langages que j'ai choisi).

Pour la vue, j'ai utilisé du HTML, un langage de balisage qui permet de fabriquer des vues sur une interface web, accompagné de CSS. Le Framework Bootstrap est utilisé car le Framework propose une mise en forme fluide et propre

Pour la conception, j'ai décidé de rester simple et efficace en utilisant les différents diagrammes de conception des normes UML mis en place à un niveau universel. Je considère qu'ils permettent de modéliser mon travail de façon simple et efficace.

Pour la documentation, étant donné que la documentation des classes et des fonctions se fait avec la JSDoc est intégrée et normalisée au langage JavaScript. Les autres documents nécessitent soit l'utilisation du logiciel Word permettant le format de modification docx et le format final pdf, soit des images qui sont les différents diagrammes UML mis en place lors de la modélisation du projet. Ces documents permettent d'établir les documentations nécessaires au suivi du projet et à la maintenance future du projet.

### 3) Modèle existant

Le modèle existant est scindé en plusieurs parties :

- ➔ Une feuille d'accueil avec des listes déroulantes permettant d'entrer les différents paramètres nécessaires à la simulation et un résumé des différentes valeurs de simulation nécessaires.
- ➔ Une base de données (saisie dans le tableur directement) sur les différentes bases fiscales des pays développés plus loin dans le rapport.
- ➔ Une base de données (saisie dans le tableur directement) sur les différents facteurs économiques utiles au modèle.
- ➔ Un outil de sélection en fonction des paramètres entrés dans une page accueil
- ➔ Une table des macros : c'est-à-dire une zone où le nom de toutes les variables inter-feuille sont écrites.
- ➔ Un modèle détaillé avec toutes les valeurs permettant d'arriver au résultat voulu qui est la valeur d'un Taux d'effectif moyen d'imposition (TEMI).

#### a) La base fiscale

Plusieurs paramètres sont utiles à l'élaboration de la simulation, tout d'abord, au niveau fiscal, le PIB/habitant est donné pour chaque pays. Le modèle comporte aussi le PNB/habitant.

La base de données comporte aussi les taux des impôts sous le régime général des impôts de chaque pays.

Les impôts utilisés sont le CFE (contribution forfaitaire employer), l'IS (impôt sur les sociétés), l'IMF (impôt minimum forfaitaire), l'IRVM (impôt sur les revenus mobilières), l'IRC (impôt sur les revenus des créances), la TVA sur le pétrole (taxe sur la valeur ajoutée du pétrole).

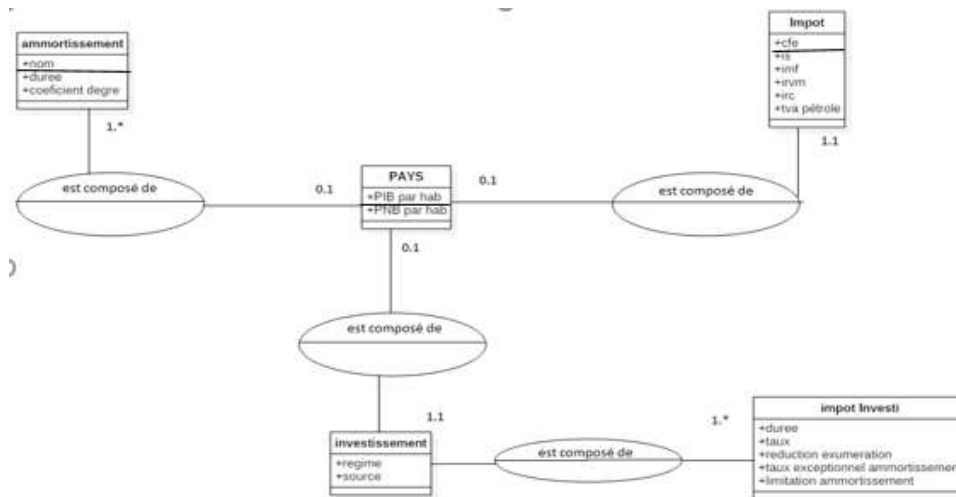
Le code général des impôts n'étant pas le seul régime fiscal, il a fallu préalablement déterminer le nombre de régimes des investissements ainsi que ceux que l'on choisit.

La formulation de plusieurs hypothèses a été nécessaire afin de déterminer à quel régime privilégié l'entreprise modélisée peut prétendre. L'investissement concerne une entreprise nouvelle. Il ne s'agit pas d'un investissement d'extension des capacités de production d'une entreprise déjà existante. Le montant d'investissement éligible au code des investissements correspond à l'actif immobilisé de l'entreprise, soit  $145 \times \text{PNB/tête}$ . L'entreprise est située dans la capitale du pays. L'entreprise crée 60 emplois. L'entreprise n'est pas exportatrice. Elle vend la totalité de sa production sur le marché national. L'entreprise commence sa phase d'exploitation : les cinq années modélisées sont les cinq premières années d'exploitation de l'entreprise nouvelle, l'investissement a déjà eu lieu. Ces hypothèses conduisent généralement à retenir le régime du code des investissements le moins avantageux fiscalement (souvent appelé régime A), car la taille de l'entreprise est insuffisante pour être éligible aux régimes supérieurs.

La base de données contient donc le nom des régimes suivant les hypothèses précédentes de chaque pays.

La base de données contient donc la durée, le taux et la réduction/exonération de chacun des impôts présélectionnés plus haut.

Il contient aussi les amortissements pour chaque pays avec la durée ainsi que le coefficient dégressif de chaque pays



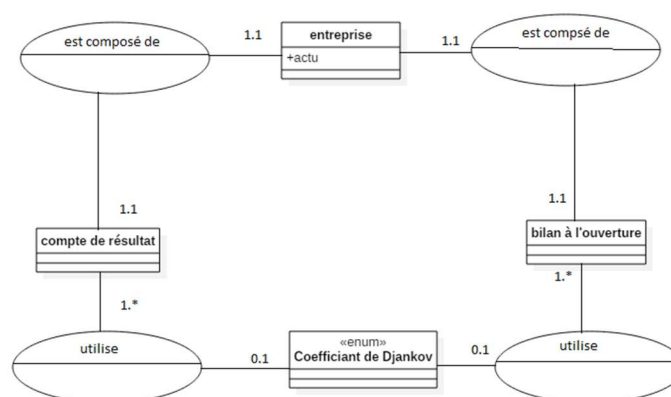
Cette image représente le mcd de la base de donnée fiscale du modèle Excel, elle a été créée localement avec l'aide de StarUML et de Paint

## b) La base économique

Plusieurs paramètres sont utiles à l'élaboration de la simulation, tout d'abord, au niveau économique, cette dernière correspond principalement aux données de l'entreprise simulées car l'entreprise n'est pas créée avec une quantité de paramètres mais avec les coefficients de Djankov mis en place en 2010, ces coefficients sont multipliés par le PIB/tête ou le PNB/tête des pays et permet la création de l'entreprise.

L'entreprise sera alors ensuite modélisée par son compte de résultat, son bilan à l'ouverture, et ses dividendes.

La base de donnée d'un point de vue économique comporte aussi le taux d'actualisation.



Cette image représente le mcd de base de données économique du modèle Excel, elle a été créée localement avec l'aide de StarUML et de Paint

## c) Le modèle

Le modèle va faire la simulation sur 5 années consécutives.

Le modèle permet à partir des paramètres sélectionnés par l'utilisateur sur la page d'accueil (qui vont chercher les données équivalentes dans les bases de données prédéfinies) et va calculer le TEMI en passant par plusieurs étapes. La première étape concerne l'entreprise, il va calculer l'investissement du pays par rapport aux achats de biens et services de l'entreprise (multiple du PIB/habitant ou PNB/habitant par les coefficients de Djankov).

Le modèle va ensuite calculer les différents amortissements de chaque achat de biens et/ou de services de l'entreprise. En fonction de la durée et du coefficient dégressif.

Le modèle va ensuite calculer les taxations réelles établies sur l'entreprise par rapport à l'impôt sélectionné, ainsi que le facteur d'actualisation sur les 5 années. On finira par avoir les coûts réels, impôts et taxes payés par l'entreprise et donc la somme totale des impôts payés par l'entreprise.

Pour calculer ces impôts, le modèle est obligé de calculer chaque donnée sur ces 5 années permettant de calculer l'impôt (sachant qu'on a que les taux d'imposition pour chacun), on va donc rapidement pouvoir accéder aux flux de trésorerie avant et après les impôts. Ces derniers permettent de trouver la VAN (valeur actuelle nette) qui permettent d'obtenir grâce aussi à une règle de trois le TEMI.

#### 4) Le nouveau modèle

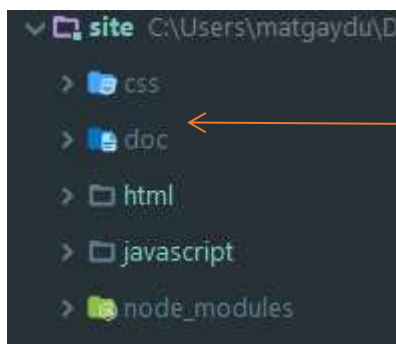
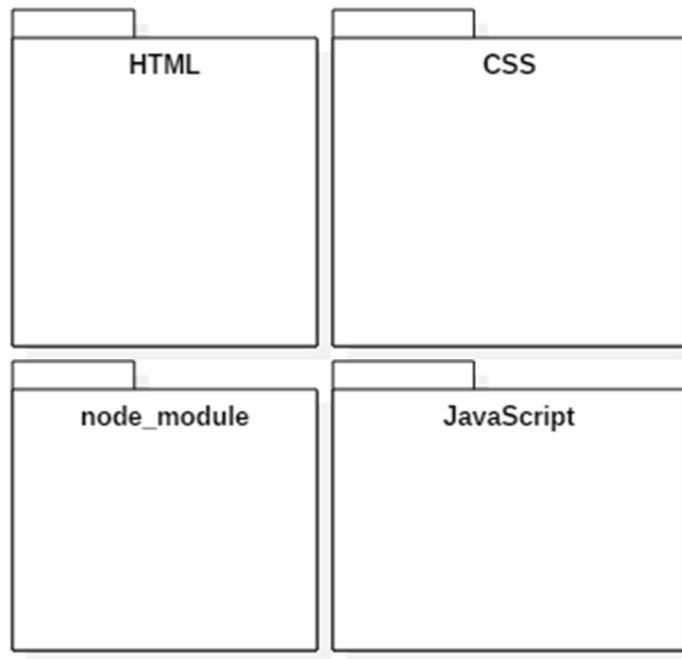
Le résultat attendu est une page web qui sera attachée au site du CERDI et qui présente les simulations de TEMI pour un ou plusieurs pays donnés au choix de l'utilisateur.

Le modèle créé permet en entrée plusieurs paramètres comme le pays, le régime fiscal, le taux d'actualisation et le taux de marge.

Le modèle établi le même résultat que le modèle Excel.

Certaines données de la Banque mondiale (comme le PIB...) sont téléchargées automatiquement. Une visualisation sous forme d'histogramme est possible. Une extraction en xls est aussi possible pour que les chercheurs ou décideurs publics puissent utiliser le résultat de leur simulation de politique fiscale.

Pour cette partie du rapport, on va arrêter l'analyse économique et étudier l'aspect informatique.



Ce dossier est spécifique à un module ajouté à mon environnement Web Storm et n'interagit pas avec le site.

Ces images représentent le diagramme de package du nouveau modèle, la 1<sup>re</sup> a été créée localement avec l'aide de StarUML et la deuxième est une capture d'écran rognée par rapport au logiciel Web Storm

On peut voir que le code est organisé de façon suffisamment simple :

Il y a un dossier HTML comportant les fichiers HTML d'une vue que l'on va définir comme statique.

Il y a un dossier JavaScript contenant le modèle en lui-même ainsi que des méthodes de chargement de vue définies comme étant dynamiques

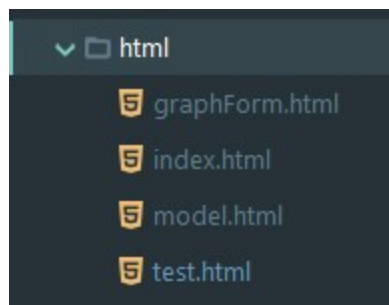
Il y a un dossier CSS comportant les fichiers CSS pour la mise en forme des fichiers HTML, car bien que le Framework bootstrap permette une mise en forme simple, il m'a fallu ajouter un peu de style.

Il y a un dossier node\_modules, ce dossier m'a permis d'ajouter des modules au JavaScript (car j'avais besoin de fonctions supplémentaires pour l'extraction Excel ou

pour le graphique), je ne développerai pas cette partie (pour plus d'informations sur ces modules, voir l'annexe).

Dans le dossier HTML, on peut voir 4 fichiers :

- ➔ Un index c'est-à-dire l'accueil du projet.
- ➔ La page test est une page qui me permettait d'afficher certains tests, la page en elle-même n'est pas importante.
- ➔ Il y a le fichier model, où il y a la partie statique du modèle.
- ➔ Le fichier nommé « graphForm » qui permet d'avoir la partie où il y a le graphique des TEMIs.



Cette image représente une capture d'écran rognée par rapport au logiciel Web Storm

On peut donc observer la mise en place de plusieurs dossiers en interne au modèle:

- ➔ Le dossier Start comporte les scripts et algorithmes nécessaires au démarrage de l'appli et communs à toutes les pages html, elle contient :
  - Script.js qui contient les fonctions utilisées lors des clics sur les boutons, Il va donc déterminer l'état principal de l'algorithme.
  - Le package comporte aussi un fichier index.js qui permet de mettre en place les valeurs à initier malgré les retours.
- ➔ Le dossier data comportant la class Data qui permet de centraliser les données et de créer le modèle.
- ➔ Le dossier manufacturing comportant la class Manufactor permettant d'ajouter les class qui permettent de recueillir les données des anciennes bases de données (développés sur Excel).
  - Elle va donc permettre de créer des entreprises qui sont représentées d'une part par les coefficients de Djankov qui sont prédéfinis, d'autre part par les deux taux (actualisation et marge) qui sont définis par l'utilisateur.
- ➔ Le dossier model composé lui aussi d'un autre dossier, de la classe modèle et de la classe modèle Manager.
  - La class modèle permet de réunir tous les données, des détails au résultat, qui permettent d'accéder au TEMI comme par exemple le prix réel des différents impôts ou l'amortissement de l'entreprise...
  - La class model Manager contient la création de tous les tableaux et valeurs nécessaires pour faire le modèle.
  - Le dossier, nommé another-function, contient la class MyMath et la fonction pour créer le fichier Excel.
    - La class MyMath regroupe toutes les fonctions qui permettent de faire des calculs plus compliqués que des additions,



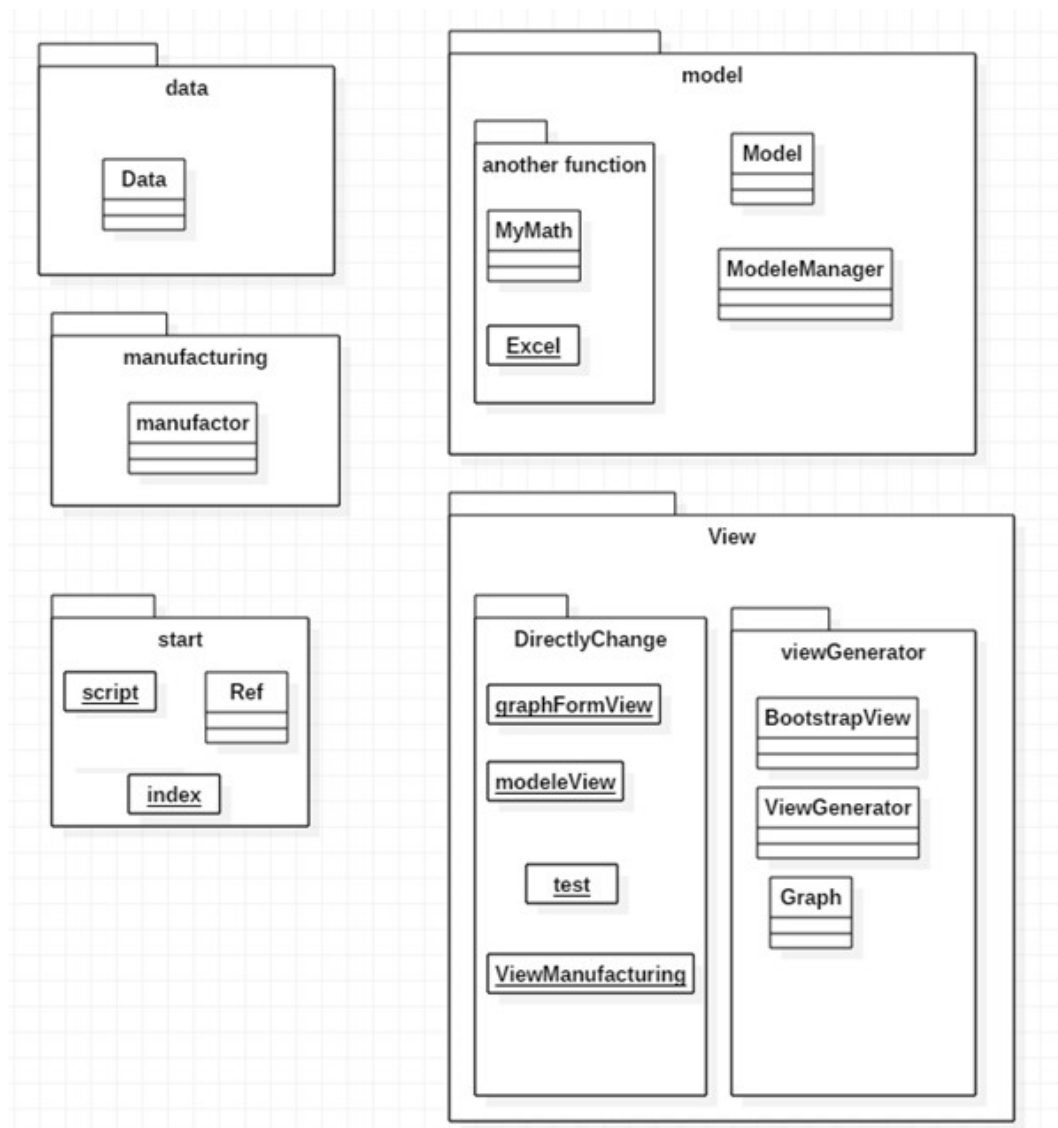
multiplications, soustractions, et divisions et n'étant pas dans la class Math, prédéfini en JavaScript. Elle permet par exemple de calculer des formules telle que celle de la VAN...

- La fonction du fichier Excel va utiliser le module excellentexport qui utilise l'API de EXCEL pour créer un fichier XLS dans lequel on retrouvera soit le tableau HTML fait de la même manière, soit un tableau prédéfini en JavaScript.

➔ Le dossier View permet de modifier la vue en fonction du modèle, il est composé de deux autres dossiers :

- Le dossier directlyChange pour lequel il y a les fonctions qui permettent de créer dynamiquement la vue, le fichier utilisé dépend du fichier html équivalent, la modification de la vue fonctionne par des IDs qui s'ils ne sont pas trouvés dans le fichier HTML actuellement utilisé, génèrent des erreurs, Il y a donc quatre fichiers comme dans le dossier HTML :
  - View manufacturing va changer, créer ou modifier la vue de l'index.
  - ModeleView lui va gérer: il va mettre en place les éléments des résultats dans la vue en fonction des items sélectionnés (Il va utiliser des fonctions et des classes qui seront décrites plus bas).
  - Test va interagir directement avec la page test, c'est sur cette page que j'ai établi certains essais avant de mettre en place le programme.
  - GraphFormView permet de mettre en place la vue qui crée le graphique, comme pour ModèleView, elle va utiliser des fonctions et des classes qui seront décrites en dessous.
- Le dossier viewGenerator qui va contenir la class BootstrapView, Graph et ViewGenerator :
  - BootstrapView est une class qui permet de créer un code HTML utilisant le Framework Bootstrap, la classe n'étant pas finie car elle ne me permet que de créer des éléments utiles à mon projet et non à tout le Framework, elle pourrait devenir un objet de travail supplémentaire voire même un projet complet.
  - Graph est une class qui utilise le module Chart.js, elle va créer les éléments nécessaires pour créer le graphique, et en utilisant les class du module créer ce qu'on appelle en HTML un canvas. Cet outil HTML ne permet pas seulement de faire des graphiques, le canvas est l'outil HTML qui permet de définir une zone de « dessins ».  
En plus de créer le graph, comme elle possède toutes les données nécessaires à la création du tableau des TEMI, elle va générer le tableau en créant sa version HTML. Elle va donc créer ce tableau et mettre en place le bouton qui permet d'utiliser la fonction Excel. Elle va donc mettre le graphique et le tableau et la sortie Excel en place.
  - ViewGenerator est une class qui va créer la vue en fonction du modèle, elle va utiliser au ¾ du temps la classe BootstrapView afin de répondre à ces besoins. Pour faire simple, elle va trier les informations utilisées et utiliser l'agencement d'une ou de plusieurs fonctions internes à la class BootstrapView, et créer le

HTML qui crée la vue. Cette fonction sera celle qui interagira avec le modèle et les différents fichiers du dossier directlyChange.



Cette image représente une capture d'écran rognée par rapport au logiciel StarUML

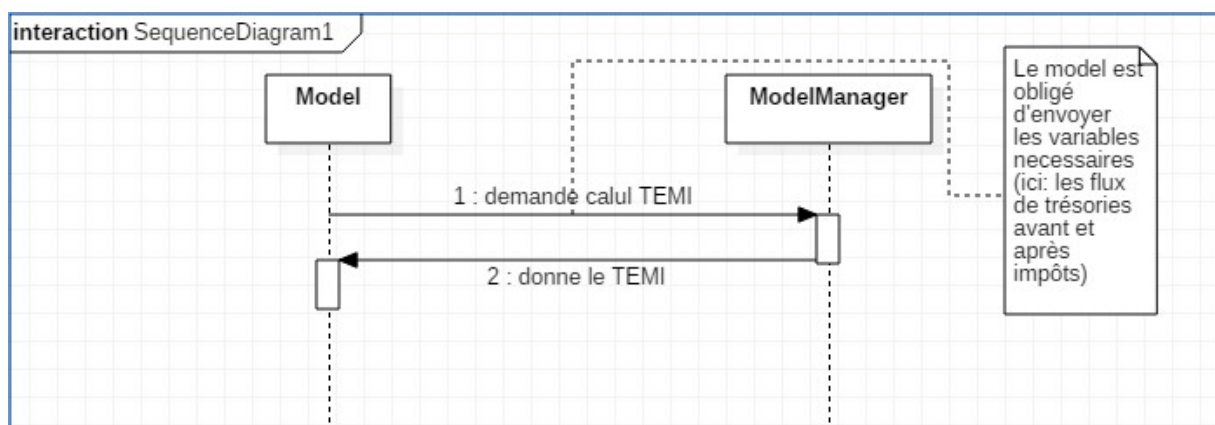
En regardant le schéma ci-dessus, on peut voir que le projet contient des fonctions, les relations n'ont pas été posées car en réalité, l'objet référence de Start qui est un fichier contient la déclaration de toutes les instances uniques.

Ce que j'appelle **les instances** uniques c'est les instances de classes que l'on n'a pas forcément besoin en plusieurs exemplaires dans le modèle. Attention, je ne parle pas de **singleton**, référence ne va jamais vérifier s'il existe une instance de ces instances. Les classes dont je parle sont MyMath, ModeleManager, BootstrapView, Manufactor, Ref et Graph.

Cela nous permet d'avoir une instance et donc de ne pas les déclarer ailleurs inutilement (donc de ne pas avoir des dizaines d'instance à droite à gauche) et d'économiser de la mémoire dans la pile d'exécution.

Pour le modèle, cela dépend des données et les données (DATA) dépendent des choix de l'utilisateur. Comme le graphique est un outil de comparaison entre plusieurs TEMI, il faut donc trouver plusieurs modèles. Du fait que les données doivent attendre les choix de l'utilisateur avant d'être instancié et du fait qu'il y a possibilité (même si ce n'est pas l'optimum car un modèle instancié comporte un nombre important d'instance, pour que ce soit optimum d'un point de vue mémoire, il faudrait que seule le TEMI soit sauvegarder et que le reste des variables soit retiré de la pile d'exécution).

Comment, le ModelManager peut gérer tous les modèles s'il est seul. Eh bien le ModelManager ne contient pas les modèles mais des méthodes qui avec des variables d'entrée, permettent de calculer et de retourner au modèle les valeurs nécessaires à la création du modèle.



Cette image représente une capture d'écran rogné par rapport au logiciel StarUML

Cette image montre un exemple d'interaction entre le model et le model Manager (celle du calcul que l'on qualifiera de principale). Les interactions entre les deux entités se font comme cela.

Le Model par contre, il possède une instance de la Classe Data qui lui donne les éléments créés avec les paramètres entrés par l'utilisateur.

Au niveau du graphique, on va avoir le droit à plusieurs étapes.

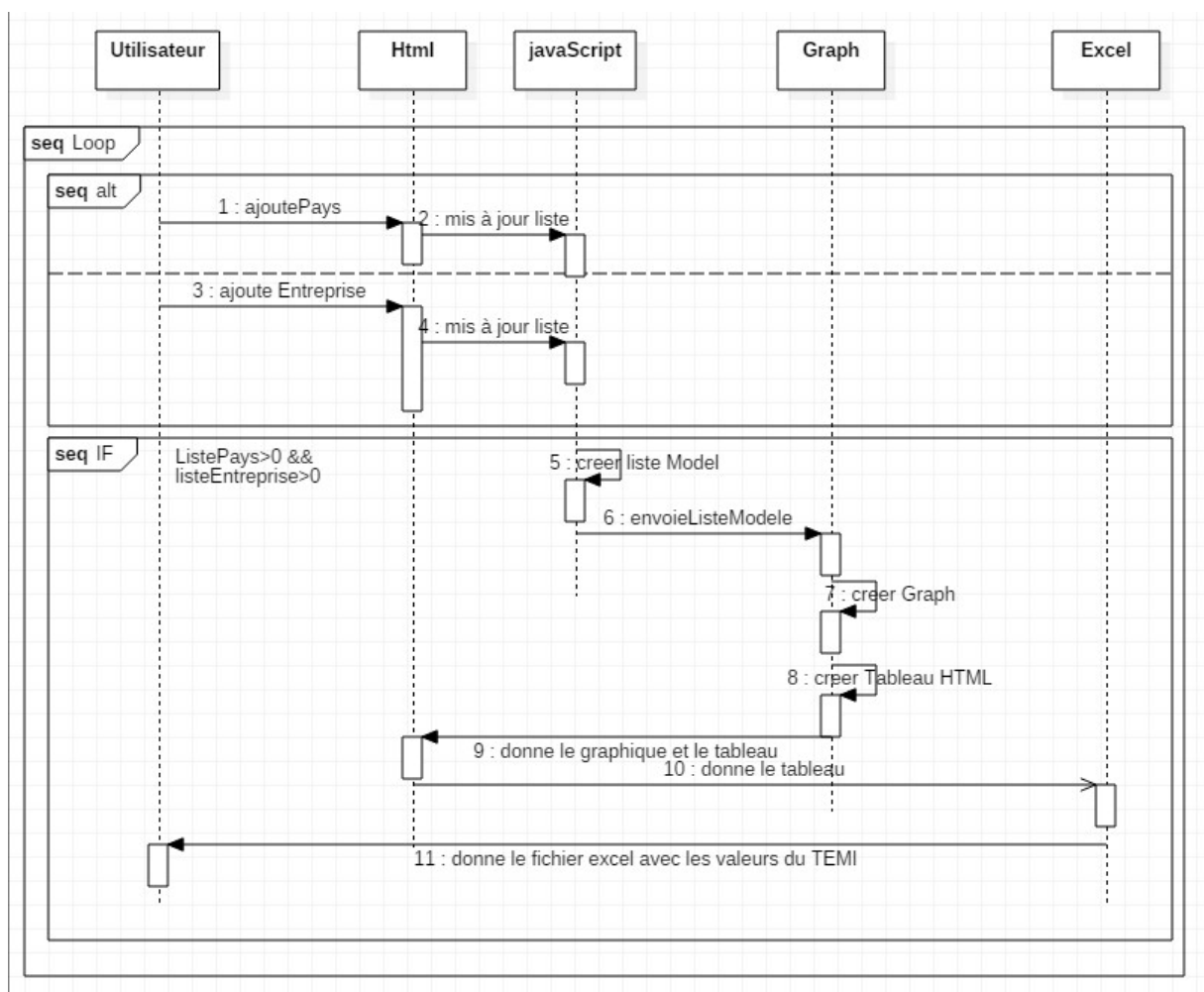
On va considérer que l'utilisateur aura déjà passé celle du choix du taux d'actualisation et du régime commun à chaque entreprise et à chaque pays qui vont être choisis dans la partie que nous allons étudier.

Dans cette partie, l'utilisateur sera face à un formulaire où il y aura une liste déroulante, une zone de texte, et deux boutons, chaque bouton est à côté soit de la liste, soit de la zone de texte.

La liste correspond à la liste des pays et la zone de texte permet à l'utilisateur d'entrer des taux de marges, le bouton à côté de la liste permet donc d'ajouter des pays et le bouton à côté de la zone de texte permet d'ajouter des entreprises avec des réussites plus ou moins différentes (le taux de marge fait varier le nombre d'achats de l'entreprise et donc de modifier son profit. Un bénéfice plus ou moins important influe le TEMI).

Au niveau du modèle, lors de ces ajouts une liste pour chacun de ces items sera mise à jour (soit ajout, soit suppression).

Quand au moins 1 pays sera sélectionné et au moins une entreprise prédéfinie, une liste de modèles va se mettre à jour en créant les modèles entre chacun des pays pour chaque entreprise définie. La class graph sera appelée, à chaque fois cette liste sera modifiée et va créer le graphique avec uniquement les TEMI intégrés dans les modèles. La class va en profiter pour insérer tous les TEMI dans un tableau, va appeler la Class BootstrapView pour faire le tableau HTML et va ajouter un bouton qui appelle la fonction pour créer le fichier EXCEL.



Cette image représente une capture d'écran rognée par rapport au logiciel StarUML

Cette image est un diagramme séquence, ce genre de diagramme est utilisé pour expliquer le fonctionnement d'un algorithme. Le schéma explicite ce que j'ai présenté pour la génération du graphique au-dessus de ce dernier.

Nous allons maintenant regarder certains calculs qui sont déduits d'une formule ainsi que l'algorithme que j'ai mis en œuvre pour les utiliser.

On va se consacrer sur 2 calculs principalement, le TRI et la VAN (qui sont liés).

Tout d'abord, au niveau de VAN, on a la formule suivante :

$$\text{Valeur Actuelle Nette (VAN)} = \sum_{t=1}^N \frac{\text{Cash flow}_t}{(1 + \text{taux})^t}$$

Image de la formule prise sur le site « [financedemarche.fr](http://financedemarche.fr) »

Le Cash-Flow est l'ensemble des valeurs obtenu en calculant les flux de trésorerie de l'entreprise, le taux est le taux d'actualisation de l'entreprise. T représente le temps.

D'un point de vue algorithmique, on va mettre en place une boucle qui incrémente une valeur représentant le nombre d'années et qui ajoute les sommes des valeurs trouvées. A cette valeur, on soustrait le prix d'investissement de l'entreprise, et on obtient la VAN du flux de trésorerie étudié.

La VAN est une mesure de la rentabilité d'investissement, elle permet de calculer le TEMI en faisant (la VAN du flux de trésorerie sans impôt – la VAN du flux de trésorerie avec impôt) / (la VAN du flux de trésorerie sans impôt).

Le deuxième calcul intéressant est le TRI (taux de rendement interne), c'est le taux d'actualisation pour lequel la VAN est égale à 0.

On calculera le TRI par tâtonnement en essayant plusieurs taux d'actualisation jusqu'à avoir une VAN=0. D'un point de vue algorithmique, cette méthode ne permettrait pas d'être efficace et ou de trouver un TRI au plus proche de la réalité mais un arrondi.

Pourtant en regardant la formule, en remplaçant le taux par le TRI et en considérant que la VAN=0, il est difficile de trouver une formule mathématique magique. Sans cette dernière, il va nous falloir utiliser la première méthode.

On a donc choisi l'arrondi du TRI de 0,01%. On va donc calculer le TRI à 0.01 près.

Pour se faire, on va commencer avec un taux à 1%, calculer la VAN, et tant que cette dernière est supérieure à 0, ajouter 1 aux taux, Une fois que la VAN est

inférieure, on va enlever aux taux 0.01 tant que cette dernière reste dans l'état indiqué. Si elle redevient positive, on ajoute 0.01 et on aura le TRI.

Pour comprendre, on va dire que le premier taux testé est 1. Que l'on augmente rapidement ce TAUX si la VAN est supérieure à 0 en sachant que le taux c'est le TRI en lui-même que l'on cherche par tâtonnement. D'après la formule, le taux étant au dénominateur, augmenter le taux fera réduire la VAN, en sachant que cette dernière est soustraite à l'investissement, la VAN passera forcément en négatif sur une actualisation forte. L'objectif est donc de trouver le niveau où le taux d'actualisation qui fait passer la VAN en positif et en VAN négatif. Une fois ce point d'actualisation trouvé, étant donné que l'on a augmenté rapidement le taux, il va falloir affiner de manière à ce que l'on trouve la VAN la plus proche possible de zéro. On va donc affiner le taux de manière à ce qu'il soit arrondi au chiffre voulu (qui est de 0.1). Mais comme la VAN que l'on a trouvé est négative, au lieu d'augmenter le taux, on va le diminuer.

Comme le chiffre trouvé est le moment où le VAN repasse en positif, on va augmenter le taux de 0.1.

Lors de la conception du site, on va utiliser certaines données comme par exemple, le PIB de chaque pays.

Afin d'utiliser ces données qui s'actualisent de manière automatique, on va utiliser l'API de la banque mondiale (DOING BANK) afin de récupérer certaines valeurs en fonction des pays utilisés. Cela permet d'avoir ces données actualisées chaque année et donc d'avoir les valeurs réelles des différentes données chaque année.

## 5) Les objectifs graphiques

Il n'y a pas vraiment de contrainte au niveau du graphisme. Il était prévu que mon projet soit rattaché comme étant un lien externe au site. Me laissant donc le choix de la disposition et des couleurs de la vue. Ayant le choix, j'ai donc décidé d'utiliser le Framework bootstrap car ce dernier fait une très grande partie sur la mise en forme d'un projet.

Il m'a donc permis de ne pas perdre de temps sur la vue et donc de me concentrer sur le développement du modèle en lui-même. De plus, j'ai mis en place un certain nombre de fonctions permettant de créer la vue en utilisant ce Framework de façon assez simple.

Il existe différentes façons d'utiliser le Framework en lui-même. La méthode la plus connue est de télécharger les bibliothèques du Framework et de l'intégrer au site. Je me suis servi des différents liens externes donnés par le site de W3School.

Utiliser un lien plutôt que d'utiliser les bibliothèques que l'on intègre aux projets à ses avantages et ses inconvénients.

Ces liens permettent d'alléger le site (et donc laisse la possibilité au site d'avoir un nombre de possibilité d'hébergement plus important à un prix raisonnable) mais empêche en regardant le site sans avoir internet, la mise en page du site en lui-même.

De mon côté, n'ayant aucune idée de la mémoire que je devais utiliser, j'ai préféré au début utiliser un lien et de ne pas regarder si le site allait être donné à un utilisateur qui n'as pas internet. De plus, l'api utilisé a besoin d'une connexion internet pour aller chercher les données. Une connexion internet et donc inévitable.

J'ai mis que très peu de couleur. Au niveau couleur, il y a celle de bootstrap et de temps en temps des pointes de jaune, bleu et rouge dans les tableaux du modèle pour la mise en page.

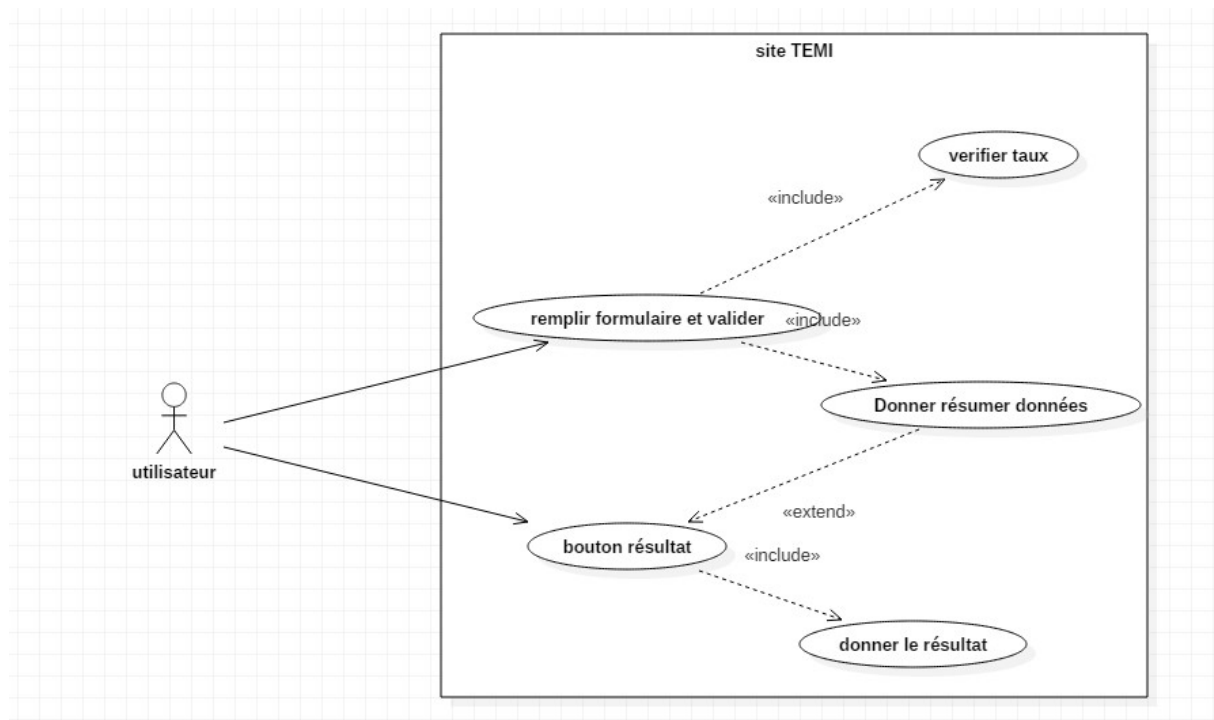
La mise en page est principalement en noir et blanc. Cette dernière n'ayant pas de rapport avec le site officiel, une nouvelle mise en page du même style que le CERDI sera mise en place, elle aura donc comme couleur principale le bleu.

Le site doit être ergonomique, c'est-à-dire qu'il faut qu'il soit facile de comprendre les informations diffusées. J'ai mis en place le modèle de base comme le modèle Excel était mis en œuvre. J'avais du mal à savoir quelle information était plus importante. Après discussion, j'ai fini par savoir, par exemple, quelle information devait être mise en place à quel endroit.

Le site doit être responsive, c'est-à-dire que l'affichage doit s'adapter à la taille du support sur lequel il est visionné.

Le Framework bootstrap a déjà intégré l'idée que le site doit être responsive et fait un énorme travail dessus, mais il reste quelques points à améliorer comme les tableaux ou encore les boutons.

## 6) Différents cas d'utilisation



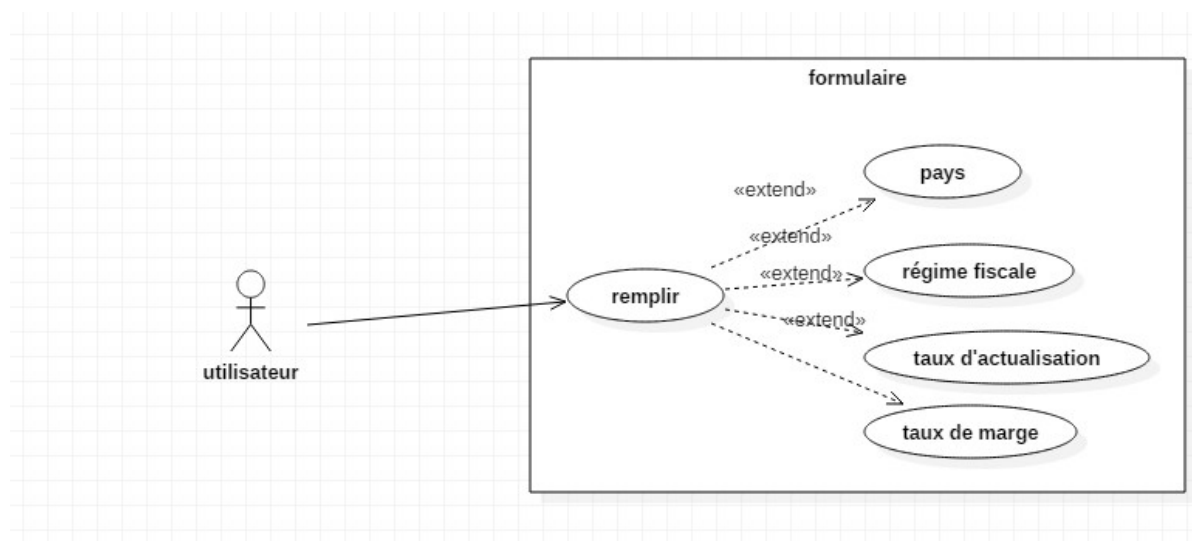
Cette image représente une capture d'écran rognée par rapport au logiciel StarUML

Cette image représente le cas d'utilisation du cas numéro 1.

Le cas numéro 1 permet à l'utilisateur d'accéder à un modèle.

Le formulaire est composé de plusieurs paramètres d'entrée comme indiqué sur l'image en dessous :

- Le pays sur lequel l'utilisateur veut faire le modèle.
- Le régime fiscal sur lequel l'utilisateur veut faire le modèle.
- Le taux d'actualisation que veut utiliser l'utilisateur pour faire le modèle.
- Le taux de marge que veut utiliser l'utilisateur pour faire le modèle.



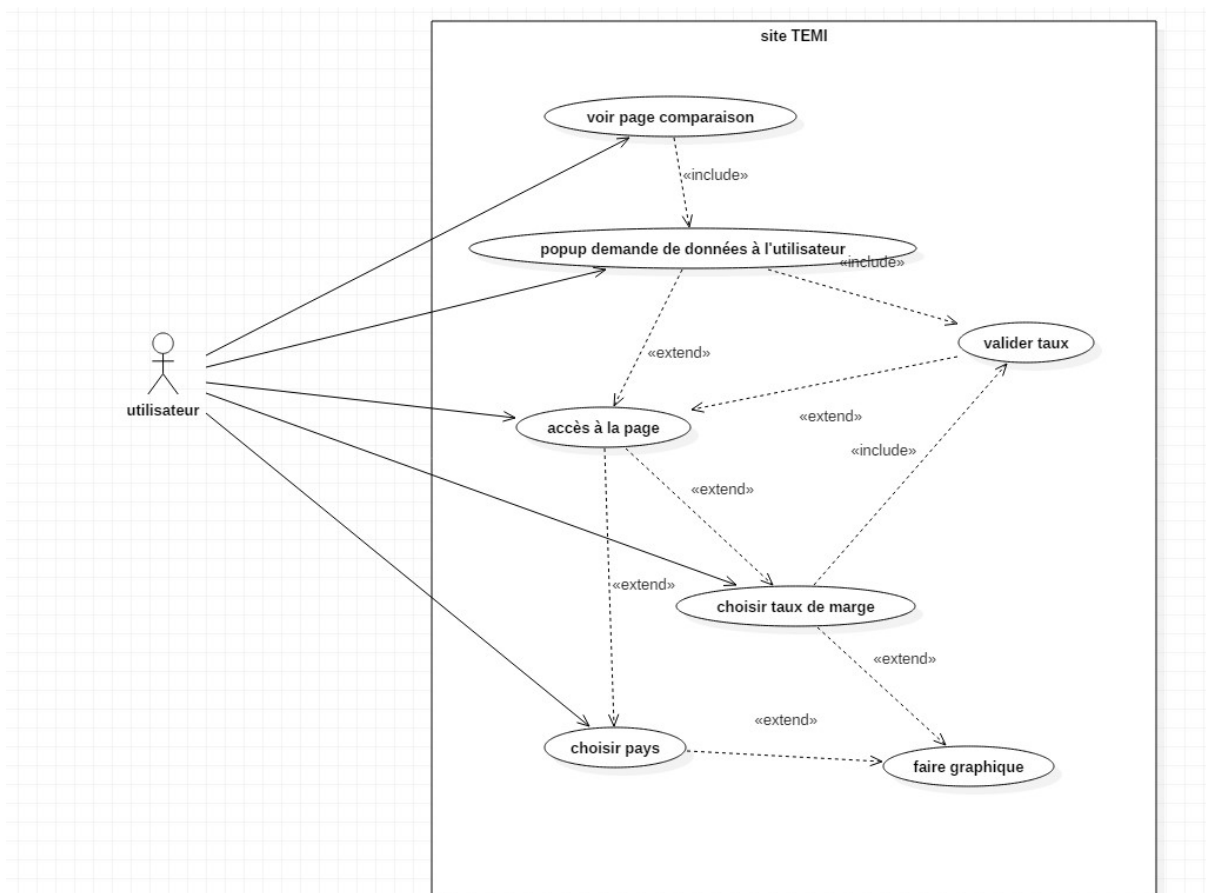


Cette image représente une capture d'écran rognée par rapport au logiciel StarUML

Pour passer à l'étape suivante, il faut remplir tout le formulaire, le régime et le pays étant des listes déroulantes, le premier item de chaque liste sera sélectionné par défaut. Il n'y a pas de taux d'actualisation et de taux de marge par défaut. Ces derniers seront donc vérifiés. S'ils ne sont pas remplis ou remplis avec n'importe quoi, ils ne seront pas validés et l'utilisateur reviendra à la case départ.

Une fois le formulaire rempli, le site va automatiquement donner un résumé organisé de toutes les données utiles au modèle. Et va mettre en place un bouton « voir résultat » qui permet, en cliquant dessus, de voir le résultat du TEMI et certaines valeurs de calcul.

Ce bouton « voir le résultat » donne donc du TEMI ainsi que tous les détails sur les impôts et l'amortissement. En plus de la valeur du TEMI, on aura accès aux TRI, aux taux effectif marginal d'imposition.

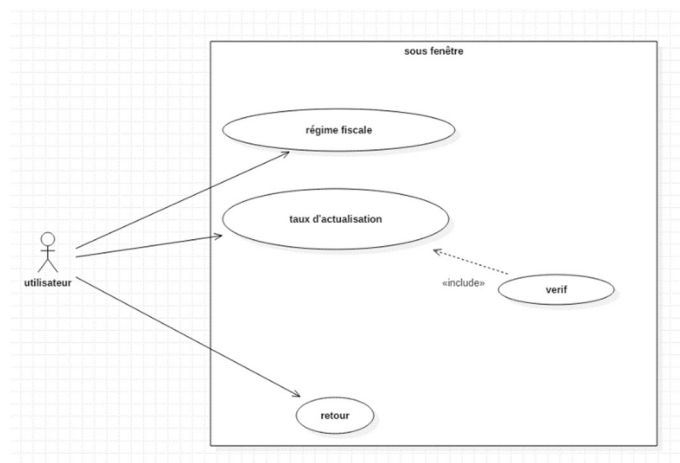


Cette image représente une capture d'écran rognée par rapport au logiciel StarUML

On voit ici le deuxième cas d'utilisation du site.

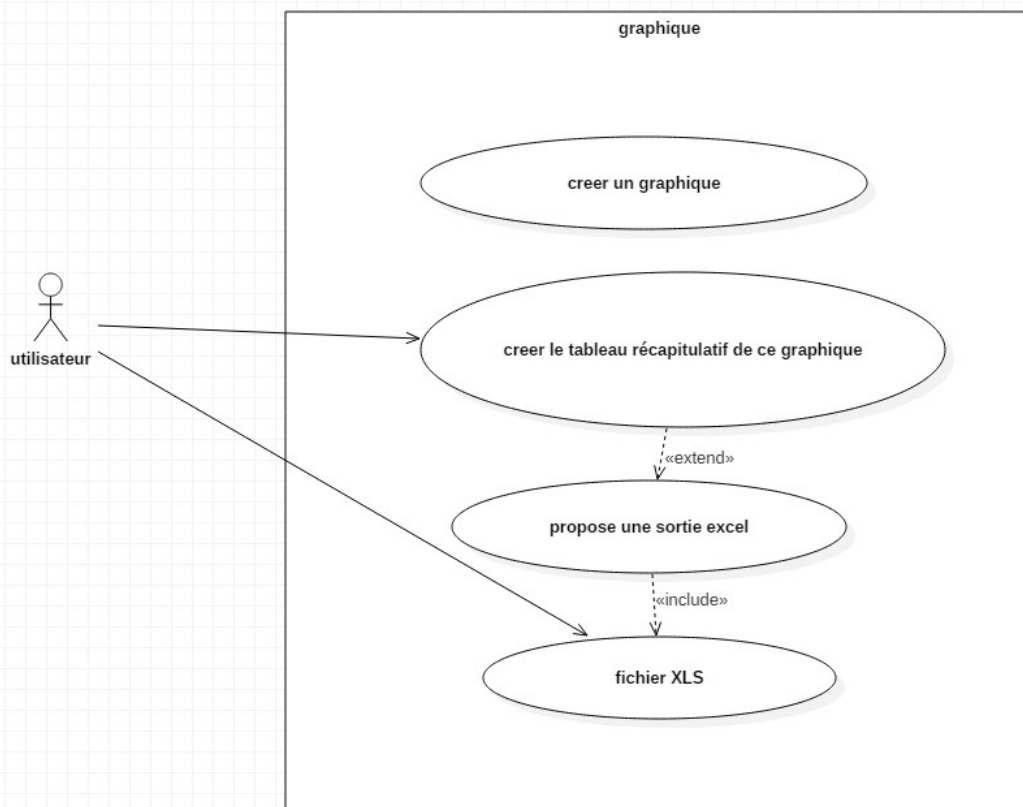
Pour accéder à ce deuxième cas, il faut obligatoirement passer par le bouton « démarrer comparaison » qui met en place une sous-fenêtre dans lequel il faut

sélectionner un régime fiscal et entrer un taux. Si ce taux n'est pas correct, on sera de retour à la case départ.



Cette image représente une capture d'écran rognée par rapport au logiciel StarUML

Il y a aussi la possibilité de faire un retour en arrière. Une fois que cette sous-fenêtre est remplie, l'utilisateur a accès à un formulaire sur lequel il peut choisir 1 ou plusieurs pays et un ou plusieurs taux de marge qui déterminera les différentes entreprises. Dès qu'au moins une entreprise et un pays est sélectionné, un graphique sera généré avec des fonctionnalités que nous allons voir en dessous.



Cette image représente une capture d'écran rognée par rapport au logiciel StarUML

On peut voir sur ce cas d'utilisation que le graphique et le tableau vont être créés et donnés à l'utilisateur. Ces deux vont utiliser les valeurs de TEMI calculées au préalable pour se construire.

Il y aura ensuite la possibilité de récupérer ces TEMI dans un fichiers XLS.

## **Bilan technique :**

La demande initiale était le transfert d'un modèle XLS sur une interface WEB. Une demande sur laquelle il n'y avait aucune contrainte.

J'ai donc accompli tout au long de mon stage ce site avec lequel j'ai acquis certaines compétences principalement en JavaScript.

J'ai donc apporté au CERDI la possibilité d'avoir l'un de leurs nombreux modèles sur un site et donc de permettre à leur « client » d'accéder à un de leur modèle sans forcément télécharger un fichier XLS incompréhensible.

Le fait d'être la seule personne à travailler sur le projet a permis aux autres personnels travaillant dans l'entreprise de continuer leur activité quotidienne. En travaillant donc sur ce modèle au sein de cette entreprise, j'ai apporté un plus sans gêner personne. Donc sans ralentir la productivité habituelle du CERDI.

Le projet n'est actuellement pas fini mais le sera je pense à la fin de ce stage. Malgré le fait que le projet sera terminé, un nombre incroyable de choses pourraient être améliorées. Il pourrait par exemple être possible de changer les valeurs des coefficients qui simulent l'entreprise. Cela permettrait à toute personne de simuler sa propre entreprise avec des coefficients qu'il aura lui-même élaborés (avec les coefficients de DJANKOV par défaut bien évidemment).

On pourrait encore dans le fichier XLS rentrer tous les modèles de toutes les entreprises avec les détails de calculs pour chaque pays.

L'utilisateur pourrait avoir la possibilité d'ajouter un pays ou plusieurs supplémentaire.

On aurait pu mettre en place un système de connexion avec la possibilité pour un utilisateur d'avoir ses tableaux et ses valeurs de paramètres favoris afin de permettre à ce dernier une meilleure productivité de démonstration.

Le modèle pourrait s'étendre à la mise en place d'une interprétation des résultats en fonctions des comportements des valeurs du TEMI.

## **Conclusion :**

Ce stage m'a apporté des connaissances concernant le JavaScript.

Lors de la formation en DUT informatique, nous avons eu 7 semaines de cours en programmation web coté client de quoi apprendre les bases de ce type de programmation. Cette période et lors de ces cours, nous avons passé 5 semaines sur JS et deux semaines sur le Framework Angular. De mon côté j'avais à peine les bases en JS si ce n'est le fait que j'étais en retard comparé à un débutant qui s'intéresse un minimum au sujet.

Ce stage m'a permis de combler mes lacunes concernant les bases de ce langage et m'a m'apporter certaines compétences comme l'utilisation du stockage du navigateur ou l'utilisation des APIs avec les objets spécifiques aux langages...

Etant donné que j'ai utilisé du JS, du CSS et du HTML. D'un point de vue informatique, ce stage m'a permis d'apprendre des choses que dans ces domaines. Mais il m'a permis d'apprendre des choses dans d'autre secteurs que dans le secteur informatique, J'ai appris beaucoup concernant le monde des finances avec les différents calculs et les définitions tout le long de ce stage (cash-flow, TRI, VAN, ...).

D'un point de vue professionnel, le stage ne m'a pas permis d'apprendre l'esprit d'équipe mais il m'a permis d'avoir un aperçu de ce que pourrait attendre un client d'un prestataire informatique. Etant donné que mon maître de stage n'est pas informaticien, il m'a expliqué ce qu'il voulait et à partir de ces explications, j'ai essayé de construire le projet en lui-même.

Le fait d'être seul m'a permis aussi d'améliorer mon organisation et mon autonomie. Je pense que je suis déjà autonome comme je l'ai prouvé pour mes travaux étudiant, mais le côté organisation est pour moi un vrai calvaire. Hors lors de ce stage, je devais mener un projet seul d'A à Z. L'organisation est donc primordiale lors de la réalisation de ce genre de projet. Ce stage m'a donc forcé à mettre en place une organisation minimum.

## Résumée en anglais :

My internship took place in CERDI (Center for Studies and Research on International Development). My subject was how to implement a WEB app for a model existing on an Excel File.

To do this, I used the Reverse Engineering method because working this way gave me a draw about the result that I did not have at the beginning of the project.

The warning Time was not established at the beginning of the project but all a new Task appears. A limit Time was set.

I have implemented a web site which is an only Front-End interface. The Technology used is a JavaScript Bootstrap, HTML and CSS. These technologies are modular and designed to implement the type of application that I developed.

I built a three-page website with two services:

- ➔ A tool to compute a TEMI (the average effective tax rate) with a lot of details.
- ➔ A tool to compare more TEMIs.

The site has a particularity, it uses an API from all the Institutions of Bretton Woods.

The tool to compute a TEMI gives a page with all compute results on a 5 years period.

For the other Tool, we can see a graph, a table and a XLS export about a TEMI value requested by the User.

For the Grap, we use a JavaScript modul. The JS modul name is chart.js. another module is installed to have the XLS export File, it's ExcellentExport .

The site will be realised on time but a lot of things could be improve in term of fonctionnalities or optimization.

That's how my internship went, constituting an enriching experience for me.

## **Bibliographie/webographie :**

<http://www.hceres.fr/> : rapport d'évaluation de l'Agence d'Evaluation de la Recherche et de l'Enseignement Supérieur

<http://ferdi.fr/> :site du FERDI

<http://iddri.org/> :site du IDDRI

<http://cerdi.org/> : site du CERDI

<https://www.teamgantt.com/> : Le site de team Gantt

<http://financedemarche.fr/finance/la-valeur-actuelle-nette-dun-actif-comment-la-calculer> : Le site financedemarche.fr

<https://www.w3schools.com/> : le site de W3schools

A partir de maintenant tous les liens seront des images utilisé dans l'annexe

<https://www.programmation-facile.com/mettre-en-forme-texte-html5/>

<https://www.humancoders.com/formations/css3>

<https://www.grafikart.fr/formations/debuter-javascript>

[https://upload.wikimedia.org/wikipedia/commons/thumb/f/fb/Singleton\\_UML\\_class\\_diagram.svg/500px-Singleton\\_UML\\_class\\_diagram.svg.png](https://upload.wikimedia.org/wikipedia/commons/thumb/f/fb/Singleton_UML_class_diagram.svg/500px-Singleton_UML_class_diagram.svg.png)

## Lexique :

**IDE :** Un environnement de développement intégré (Integrated Development Environment): Un logiciel qui permet de faciliter le développement d'un projet informatique.

**VCS:** logiciel de gestion de version (Version contrôle system): un logiciel qui permet de stocker un ensemble de fichiers en conservant la chronologie de toutes les modifications qui ont été effectuées dessus.

**Technologies primitives:** J'entends par primitives des technologies que l'on pourrait nommer comme étant les plus anciennes des technologies les plus connues.

**Framework:** En programmation informatique, un framework (appelé aussi cadre applicatif, cadre d'applications, socle d'applications ou encore infrastructure de développement) désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).

**Bootstrap :** Bootstrap est ce que l'on appelle Framework, principalement basée sur la vue d'un site, il propose des feuilles de style qui permettent d'établir des fonds. Le développeur doit de cette manière faire très peu de CSS et utilise plus le Framework, cela lui permet d'aller plus vite.

**Front-End/Back-end :** Un site internet comporte deux faces lors du déploiement de ce dernier, Une partie nommée Front-End, qui est la partie côté client, c'est ce que le Pc le navigateur de l'utilisateur va voir et faire. Le côté back-end, plus du côté serveur, c'est le modèle sur lequel se base le site, ce qui se fait en arrière-plan

**Norme UML:** Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

**Cotisation foncière des entreprises (CFE) :** Cet impôt est payé par les entreprises, dans notre cas, elle est calculée par rapport au prix de la main d'œuvre de l'entreprise.

**Impôt société (IS):** Cet impôt est payé par les entreprises, elle est calculée par rapport au bénéfice sur lequel il est possible de faire payer des impôts.

**Impôts minimum forfaitaire (IMF) :** Cet impôt est payé par les entreprises, elle se calcule par rapport au chiffre d'affaire de l'entreprise, uniquement lorsque l'entreprise ne peut pas payer l'impôt société, elle paye l'IMF.

**Impôt sur les revenus mobilières (IRVM) :** cet impôt est payé par les entreprises, cette taxe se calcule par rapport au bénéfice imposable sur lequel on enlève l'impôt IS ou IMF selon le cas.

**Impôt sur les revenus des créances :** cet impôt est payé par les entreprises, cette impôt se calcule par rapport au revenu généré par les créances client, dans notre cas on prendra la colonne des charges financières.



Taxes sur la valeur ajoutée du pétrole: tous le mondes connais la TVA, cette taxe qui est sur n'importe quelle bien ou service consommable qui se calculs à partir de la valeur ajouté d'un produit. Dans ce modèle, on prend cette taxe sur le pétrole.

Les coefficients de Djankov sont des coefficients mise en place par l'économiste Simeon Djankov, ils permettent de simuler une entreprise. Il suffit juste de les multiplié par le PIB/habitant ou le PNB/habitant du pays dans lequel on simule l'entreprise.

PIB/tête et PNB/tête: IL y a dans le monde plusieurs indicateur permettant d'évalué le développement mondiale, le PIB et PNB par tête en font partie, IL permettent d'évalué la répartition de la richesse nationale parmi les habitants si cette dernière serait répartie équitablement. C'est donc une moyenne des richesses possédé par les habitants.

Compte de Résultat, un document comptable synthétisant l'ensemble des charges et des produits d'une entreprise ou autre organisme ayant une activité marchande, pour une période donnée, appelée exercice comptable.

Le bilan à l'ouverture : Un bilan comptable est un document qui synthétise à un moment donné ce que l'entreprise possède, appelé l'« actif » (terrains, immeubles, etc.) et ses ressources, appelées le « passif » (capital, réserves, crédits, etc.). Le bilan est l'un des documents composant les états financiers. Le bilan à l'ouverture est le bilan au moment où l'entreprise commence son activité professionnelle.

Taux d'actualisation : un taux minimal de rentabilité en dessous duquel un investisseur considère qu'il n'a pas d'intérêt à investir ses capitaux. Ceci veut dire qu'à partir d'un taux d'actualisation donné l'investisseur peut prendre une décision d'investir ou non.

Flux de trésorerie (cash-flow) est un mouvement, entrée ou sortie, de liquidités.

Les instances sont en informatique des objets qui seront déterminer par des classes. Cette objet aura un comportement, un état prédéterminé par cette classe.

Singleton : le nom d'un pattern design qui sera expliqué en annexe.

Pile d'exécution est une structure de données de type pile qui sert à enregistrer des informations au sujet des fonctions actives dans un programme informatique. Une structure de donnée de type Pile est une manière d'organisé les données assez particulières, elle consiste à entassé es donnée dans ce qu'on va appeler une liste, la dernière donnée de cette liste sera quoi qu'il arrive la première sortie (last in first out).

Microéconomique : la branche de l'économie qui modélise le comportement des agents économiques (consommateurs, ménages, entreprises, etc.) et leurs interactions notamment sur les marchés.

Macroéconomique : l'approche théorique qui étudie l'économie à travers les relations existantes entre les grands agrégats économiques, le revenu, l'investissement, la consommation, le taux de chômage, l'inflation, etc.

Bretton Woods : Les Accords de Bretton Woods, parfois également nommés selon la Conférence de Bretton Woods, sont des accords économiques ayant dessiné les grandes lignes du système financier international mis en place après 1944. Les

accords ont donné naissance à deux organismes internationaux : la Banque internationale pour la reconstruction et le développement ou BIRD, aujourd'hui l'une des composantes de la Banque mondiale ; le Fonds monétaire international ou FMI.

## Annexe :



Cette image représente le langage HTML 5 qui m'a permis de faire une partie de la vue



Cette image représente le langage avec lequel je vais faire ma feuille de style



Cette image représente le langage avec lequel j'ai fait le modèle

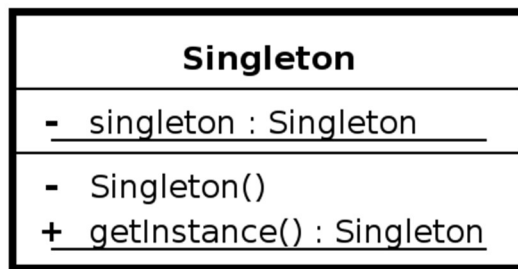
Dans mon développement le long de ce rapport, j'ai parlé de vue statique et vue dynamique, ce que j'entendais par vue statique, c'est la vue qui, peu importe le choix de l'utilisateur, restera inchangé quoi qu'il arrive. Cette dernière est généralement faite dans des fichiers HTML de base.

La vue dynamique est la vue qui sera mise en place lorsque l'utilisateur va faire tels ou tels comportements, elle permet d'établir donc le vrai comportement de la vue, il y a plusieurs façons de faire cette vue, dans notre cas, la vue sera mise en place par le langage JAVASCRIPT.

Le singleton, lorsque j'ai expliqué mon modèle j'ai cité ce que l'on appelle un design pattern mais qu'est-ce que c'est ?

Déjà un design pattern ou patron de conception pour les français est la meilleure solution connue à un problème de conception récurrent. C'est-à-dire qu'en programmation, on est souvent confronté à un même type de problème. Ces patrons permettent d'avoir au développeur des techniques pour résoudre le problème. Ils nous montrent généralement la meilleure solution, c'est-à-dire la solution la plus propre de résoudre le problème.

Le singleton est un patron de conception, il résout donc un problème : Certainement, le développeur a besoin d'avoir une seule instance par rapport à une classe car sinon le programme va se mettre à avoir des problèmes et à avoir un mauvais comportement qui pourrait s'apparenter à des bugs.



Cette image est tirée de Wikipédia

Ce schéma représente le Singleton, elle utilise les normes pour faire un diagramme de classe en UML.

Ça façon de procédé est simple, c'est lui qui va gérer l'instance de l'objet ciblé, Il va posséder une déclaration de l'objet, et une méthode de classe public.

Cette méthode va retourner l'instance est donc vérifié si cette dernière n'est pas déjà créée.

Afin d'avoir un aperçu de l'état actuel de la vue, je vous mets ci-dessous une capture d'écran de la page d'arrivé :

