

REPUBLIQUE DU SENEGAL

Un Peuple – Un But – Une Foi

**MINISTERE DE L’ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE**

**UNIVERSITE DE THIES**



**U.F.R DE SCIENCES ET TECHNOLOGIE**

Mémoire de fin de Cycle pour l’obtention de

**Licence Informatique Option Génie Logiciel**

*Sujet*: Application pour tracer des Graphes et Diagrammes

Présenté par Sous la direction de :

**Abou SOW M. Ousmane SALL**

**Année Académique 2015/2016**

# Dédicaces

# Remerciements

# Sigles et Abréviations

# Avant-propos

1. SOMMAIRE

Dédicaces..............................................................................................I

Remerciements…………………………………………………..….II

Abréviations et sigles………………………...……………………III

Explication du titre de l’application……………………………...IV

Introduction………………………………………………….………1

Chapitre 1 : Présentation Général……………………………..….2

* 1. Objectif du document…………………………………………………3
  2. Portée du système………………………………………………………4
  3. Définition du sujet
  4. Problématique
  5. Objectifs

Chapitre 2 : Analyse Des besoins

* 1. Besoins fonctionnels
  2. Besoins non fonctionnels
  3. Principales fonctionnalités du futur système
  4. Etude détaillé des modules

Chapitre 3 : Conception de la solution

* 1. Structure de la solution
  2. Choix des outils et technologie utilisé

Chapitre 4 : Implémentation

* 1. Technologies utilisés
  2. Les captures

Conclusion

Document de référence

Introduction :

De nos jours les professeurs éprouvent un besoin de faire des graphes pour mieux visualiser certaines données numériques. Et vu que le nombre d’outils qui pourront leur permettre de réaliser cela est très restreint, la nécessité de créer une application de bureau (application s’exécutant sur un ordinateur sans l’aide du navigateur) pouvant tracer des graphiques ainsi que des diagrammes deviennent primordial.

Toute fois l’application présentera diverses fonctionnalités comme le fais de photographier des graphes ou de les exporter sous fichier PDF.

Il s’y ajoute que l’utilisateur aura la possibilité de choisir une image simple ou en 3D mais néanmoins l’application ne pourra pas prendre en compte les fonctions mathématiques (cosinus, sinus, etc.).

Ainsi l’application s’inscrit dans la logique de remédier à tous les obstacles qu’un professeur peut rencontrer pour tracer un graphe ou un diagramme

Chapitre 1 : Présentation Générale

* 1. Objectif du document

L’objectif du document est de représenter les fonctionnalités et les limites de l’application « **TAG** » afin d’avoir une idée plus claire de ce qu’elle sera. Il permet de bien définir les besoins et conséquemment de s’assurer que les développeurs ont la même compréhension sur ce qui sera implanté et sur ce qui ne le sera pas.

* 1. Portée du système

Dans l'ensemble il s'agit de développer un système de générateur de graphes. Dans cette phase du projet, on considère seulement la partie du système qui crée des graphiques à barres, circulaire ou linéaires. Le système ne prend pas en charge les fonctions mathématiques.

* 1. Définition du sujet

Le sujet consiste à utiliser un système informatisé dans le but de créer des diagrammes. Le principe est le suivant : l’utilisateur saisi des données numériques dans le système, ce dernier génère des graphes en barres, circulaire, linéaires et autres. Ainsi l’utilisateur a plusieurs options sur le format du diagramme (en 3D ou simple).

* 1. Problématique

Des fois on a besoin de représenter graphiquement les données pour une visualisation plus optimale. ET il est difficile de vouloir tracer des graphes à la main, cela amène même une manque d’exactitude des données. Il s’y ajoute aussi la difficulté de tracer des graphiques riches, élégants et colorés à la main.

Ainsi, il est rare de trouver des outils qui nous permettent de réaliser des graphiques voulu facilement. Les données a représenté sont parfois énormes et on a du mal à les biens représenté.

* 1. Objectif

Sur ces faits l’objectif se limite à la mise en place d’une plateforme de générateur de graphe, ce qui permettra de réaliser les opérations suivantes :

* De choisir le type de graphes désiré
* De choisir les caractéristiques du graphe
* D’entrer facilement les données numériques
* De sauvegarder le graphe généré en image ou PDF
* De coller le diagramme dans un document

Chapitre 2 : Analyse des données

* 1. **Besoin fonctionnel**

La plateforme devra permettre de réaliser les opérations suivant :

* De choisir le type de données à saisir
* De choisir le type de graphe
* De paramétrer les caractéristiques du graphe
* Saisir les données dans les champs au niveau de paramètre de données
* D’ajouter ou de diminuer les champs
* De générer des diagrammes
* De pouvoir les modifié
* Exporter le graphe généré dans un document
* De sauvegarder le graphe sous format PDF ou image
  1. **Besoin non fonctionnel**

Il s'agit des besoins qui caractérisent le système. Ce sont des besoins en matière de performance, de type de matériel ou de type de conception. Ces besoins peuvent concerner les contraintes d'implémentation (langage de programmation, type SGBD, de système d’Exploitation, etc.). Il faudra aussi noter que les informations gérées à travers les applications devront être accessibles uniquement aux personnes selon leurs profils respectifs.

* 1. **Principales fonctionnalités du système**

Sous la forme d’un diagramme de cas d’utilisation UML, nous présentons à la **figure1** les fonctionnalités de haut niveau de notre système

****

**Figure 1 :** **Diagramme du cas d’utilisation du système**

* 1. **Etudes détaillé des modules**

Nous proposons dans les lignes qui suivent une description textuelle détaillée de chacun des cas d’utilisation.

* **Choisir type de données du graphe :**

L’utilisateur doit choisir le type de données qu’il veut entrer dans le menu

* **Choisir type de graphe :**

Après il choisit le type de graphe qu’il désire par exemple diagramme en barre, circulaire etc.

* **Paramétrer les caractéristiques du graphe**

Ici l’utilisateur paramètre les caractéristiques du graphe

* **Saisir des données**

C’est la partie ou l’utilisateur saisi les données du graphe

* **Modifier des données**

Il peut aussi modifier les données qu’il vient de saisir

* **Ajouter ou réduire des champs**

Ici il pourra ajout le nombre de champs qu’il veut

* **Générer graphe**

C’est la partie qui lui permet de visualité le graphe

* **Modifier les caractéristiques du graphe généré**

Il peut aussi modifier les caractéristiques du graphe généré

* **Sauvegarder graphe sous format PDF ou image**

C’est la partie qui lui permet de sauvegarder le graphe

* **Quitter**

Il peut quitter l’application à tout moment au niveau du menu

**Un diagramme de séquence** permet de cacher les différentes interactions d’objets dans le cadre d’un scénario d’un diagramme de cas d’utilisation. Nous l’avons utilisé pour détaillé chacun des cas d’utilisation. Nous allons choisir un cas d’utilisation qui nous semble plus importants pour en donner une description détaillée. Nous allons ainsi faire un diagramme de séquence dans les lignes qui suivent.

**Scénario :**

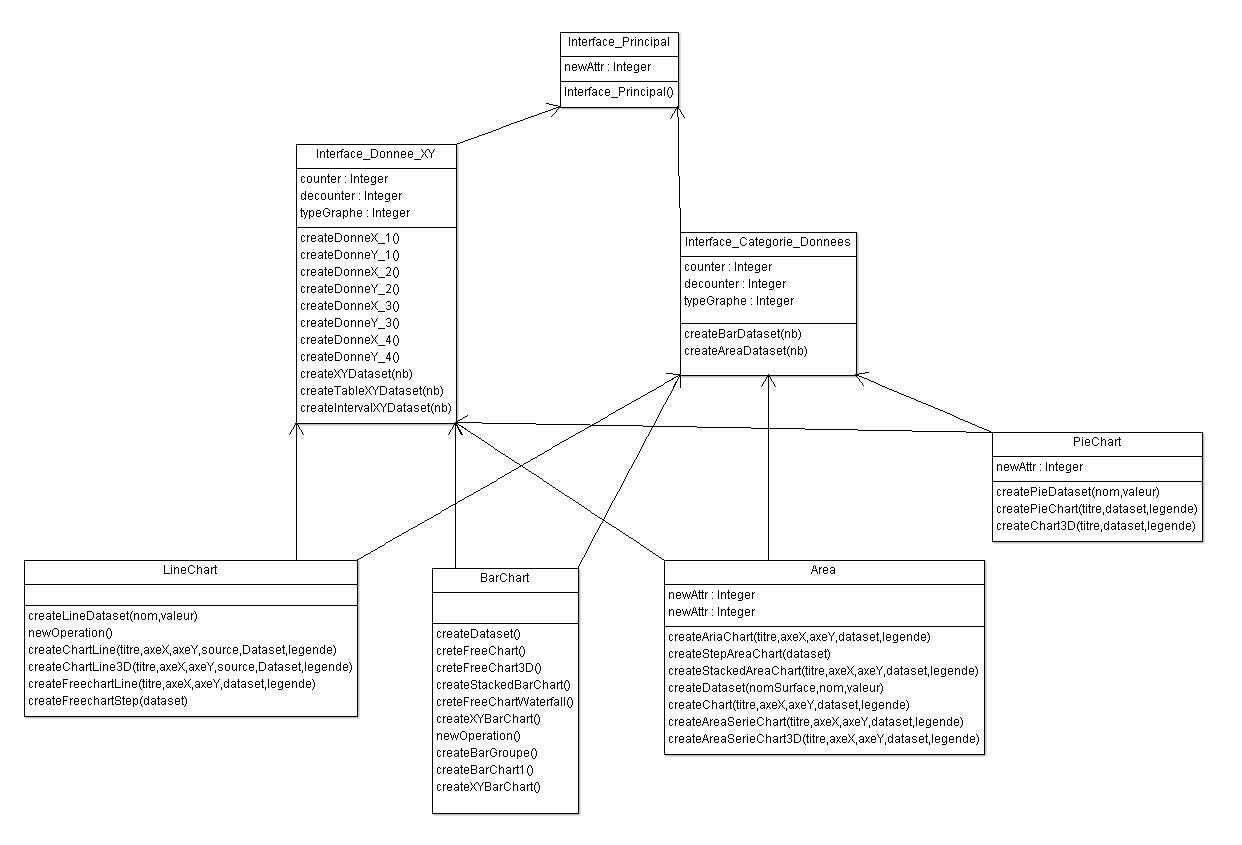
**Nom du cas :**

**Principale acteur :**

**Scénario normal :**

* L’utilisateur choisir les types de données du graphe dans le menu principal
* Le système ouvre une interface ou l’utilisateur agi
* L’utilisateur choisi le type de graphe, les caractéristiques de ce dernier
* L’utilisateur saisi les données, click sur valider
* Le système génère un diagramme en barres

Après avoir étudié les diagrammes de séquences de certains cas d’utilisation, nous présentons sous la forme d’un **diagramme de classe** la structure statique du système en termes de classe et de relation entre ces classes. L’intérêt du diagramme de classe est de modéliser les différentes entités du système d’informations.



**Description textuelle de chaque classe :**

* Interface\_Principal :

La classe Interface\_Principal représente interface ou l’utilisateur interagi avec l’application,

* Interface\_Donnees\_XY :

La classe Interface\_Donnees\_XY représente la partie de l’application ou

l’utilisateur saisi les données de type l'axe du domaine comme axe X et

l'axe de portée comme Axe Y

* Interface\_Categorie\_Donnees :

La classe Interface\_Donnees\_XY représente la partie de l’application ou l’utilisateur saisi les autres type données d

* LineChart :

Représente tous les graphes de type linéaire

* BarChart :

Représente tous les diagrammes en barres de l’application

* AreaChart :

Représente tous les diagrammes de type surface de l’application

* PieChart :

Représente tous les diagrammes circulaires de l’application

Chapitre 3 : Conception de la solution

**3.1** **Structure de la solution**

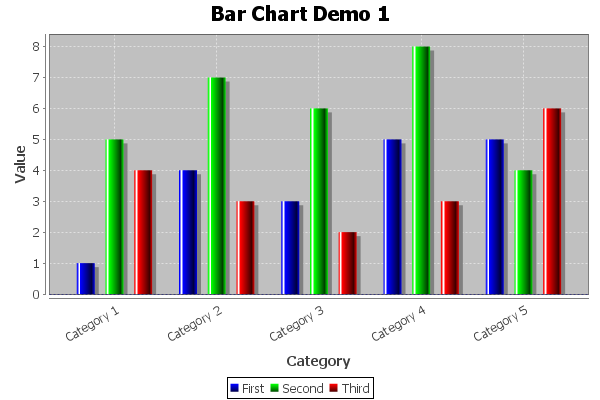
Notre solution est une **application Desktop** qui est un logiciel applicatif qui tourne sur PC. Un logiciel est un ensemble de programmes qui vont être exécutés par la machine. C’est un programme directement utilisé par l’utilisateur pour réaliser une tache. Typiquement, un [éditeur de texte](https://fr.wikipedia.org/wiki/%C3%89diteur_de_texte), un [navigateur web](https://fr.wikipedia.org/wiki/Navigateur_web), un [lecteur multimédia](https://fr.wikipedia.org/wiki/Lecteur_multim%C3%A9dia), un [jeu vidéo](https://fr.wikipedia.org/wiki/Jeu_vid%C3%A9o), sont des applications. Les applications s'exécutent en utilisant les services du [système d'exploitation](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) pour utiliser les [ressources matérielles](https://fr.wikipedia.org/wiki/Mat%C3%A9riel_informatique).

Contrairement aux applications web, les applications Desktop ne nécessite pas de connexion internet pour marcher. Il suffit juste de l’installer directement sur son propre ordinateur. Le principal composant de l’application desktop est l’interface utilisateur.

* 1. **Présentation de l’application TAG**

Ici nous présentons un exemple en image de l’interaction entre l’utilisateur et l’application TAG





Utilisateur graphe généré par l’application TAG

**Figure** : Schéma de l’application desktop TAG

* 1. **Choix des outils et technologie utilisé**
     1. JAVA

C'est un langage très utilisé, notamment par un grand nombre de programmeurs professionnels, ce qui en fait un langage incontournable actuellement. Java est un langage de programmation moderne développé par **Sun Microsystems** (aujourd'hui racheté par **Oracle**)

Une de ses plus grandes forces est son excellente portabilité : une fois votre programme créé, il fonctionnera automatiquement sous Windows, Mac, Linux, etc.

Parmi les nombres avantages qu’offre la technologie JAVA, nous avons choisi **java swing** pour gérer l’interface graphique de l’application TAG. En effet swing est une [bibliothèque graphique](https://fr.wikipedia.org/wiki/Biblioth%C3%A8que_graphique) pour le [langage de programmation Java](https://fr.wikipedia.org/wiki/Java_%28langage%29). Il offre la possibilité de créer des [interfaces graphiques](https://fr.wikipedia.org/wiki/Graphical_user_interface) identiques quel que soit le [système d'exploitation](https://fr.wikipedia.org/wiki/Syst%C3%A8me_d%27exploitation) sous-jacent,

* + 1. l'API JFreeChart

FreeChart est une bibliothèque de graphiques gratuite à 100% Java qui permet aux développeurs d'afficher facilement des graphiques de qualité professionnelle dans leurs applications.

Il est conçu pour une utilisation dans des applications, Applets, servlets et JSP. JFreeChart est distribué avec le code source complet sous réserve des termes De la licence GNU Lesser General Public License, qui permet à JFreeChart d'être utilisé

JFreeChart peut générer des diagrammes à secteurs, des diagrammes à barres (réguliers et empilés, avec un ect 3D optionnel), les diagrammes en ligne, les diagrammes de dispersion, les diagrammes de séries chronologiques (y compris les moyennes mobiles, les graphiques à haut-bas-ouverture-fermeture et les parcelles de chandelier), les diagrammes de Gantt, les diagrammes de compteur (cadran, boussole et thermomètre), diagrammes de symbole, les parcelles de vent, les diagrammes de combinaison et plus. Les caractéristiques supplémentaires comprennent:

* Les données sont accessibles à partir de toute mise en œuvre des interfaces définies;
* Exporter vers des formats d'image PNG et JPEG (ou vous pouvez utiliser la bibliothèque Image IO de Java pour exporter à n'importe quel format pris en charge par Image IO);
* Exporter vers n'importe quel format Graphics2D Mise en œuvre, notamment :
* SVG
* PDF
* Outils;
* Zoom interactif (zone de glissement et / ou roulette de la souris) et panoramique;
* Événements de la souris de diagramme (ceux-ci peuvent être utilisés pour les diagrammes drill-down ou les pop-ups d'information);
* Annotations;
* Génération de cartes d'image HTML;
* Fonctionne dans les applications, servlets, JSP (grâce au projet Cewolf1) Et applets;
* Distribué avec le code source complet sous réserve des conditions du grand public GNU Licence (LGPL);

Vu que l’API **JFreeChart** est très utile pour mettre en œuvre des programmes pour tracer des diagrammes, nous avons choisi cette technologie qui est capital pour la réalisation de notre application TAG.

* 1. **Conception détaillé des fonctionnalités de l’application**

La conception est constituée de deux parties principales :

* La première partie du code est relative à l’interface graphique de l’utilisateur. Avec l’aide de java swing, nous utilisons un JFrame pour l’interface utilisateur avec des panels, par exemple on a un panel de paramétrage pour les caractéristiques du graphe avec des saisis de textes et des choix que l’utilisateur pourra faire, un panel de saisi de données du graphe des champs de données de saisie et un boutons pour permettre à l’utilisateur de transmettre les données.
* La deuxième partie du code est responsable du traitement métier de l’application. Une fois que l’utilisateur saisi les informations et appui sur le bouton créer graphe, des classes créer avec du langage java se chargent de récupérer les informations dans des variables avec des contrôles sur ces dernières, Après cela les informations sont envoyés aux méthodes fourni par l’API **JFreeChart,** ces méthodes génèrent le type de graphe choisi qui sera visualiser grâce à un JFrame