

# AppStats

MACÉ Gabriel 12010249

VELLETAZ Cédric 11908187

ZARCH Vassili 120107374

<b>1. Introduction</b>	<b>3</b>
<b>2. Description de la demande</b>	<b>3</b>
2.1.Les fonctions du produit	3
2.2.Produit du projet	3
2.2.Critères d'acceptabilité et de réception	4
<b>6. Déroulement du projet</b>	<b>5</b>
6.1. Planification	5
6.2.Ressources	6
<b>8.Glossaire</b>	<b>6</b>
<b>9.Références</b>	<b>6</b>

# 1. Introduction

Notre projet est un projet informatique à but académique. Les enjeux principaux sont de poursuivre l'acquisition de nouvelles compétences quant à la gestion de projet, et de réussir à développer un logiciel. De plus, il existe déjà des interfaces ressemblantes à celle de notre projet, il s'agit donc aussi de se différencier.

Programmation Orientée Objet - statistiques descriptives/inférentiels -  
développement - interface

## 2. Description de la demande

### 2.1. Les fonctions du produit

- Permettre d'entrer des données et d'appliquer des tests statistiques sur celles-ci afin d'en ressortir des informations.

- Permettre de tester les connaissances en statistiques de l'utilisateur

- Permettre à l'utilisateur de comprendre à quoi servent les différentes méthodes statistiques.

- Permettre à l'utilisateur de se corriger et de reconnaître ses erreurs.

- Rendre le logiciel exécutable sur n'importe quel PC avec le minimum de programme tiers.

### 2.2. Produit du projet

Le produit de notre projet est un logiciel proposant des applications de méthodes statistiques sur les données entrées par l'utilisateur comportant aussi une partie pédagogique permettant à l'utilisateur de mieux comprendre les statistiques.

#### 2.2.1. Descriptions des risques

Risques portant sur le déploiement logiciel

L'enjeu est de créer un fichier exécutable par l'utilisateur sur les PCs sous Windows, MacOSX et GNU/Linux. Le fichier d'extension .exe est le standard sur windows. Mais sur les machines sous linux, le logiciel de compatibilité wine rend exécutable les fichiers .exe. Si, le but est de développer l'application sur GNU/Linux et MacOS alors développer sur windows n'est pas transversal car la méthode est de créer un package (une forme d'archive). ie, Un

fichier archive .deb sous distribution debian, .rpm sous une distro Fedora, .dmg sous MacOS. Le but ici n'est pas de lister les méthodes possibles car par exemple un package Flatpack est indépendant d'une distro. En sachant que pour déployer un package, c'est intégré les différences entre les utilisateurs, les systèmes d'exploitations qui peuvent dépendre entre elles, les gestionnaires de paquets qui ne sont pas connus par les utilisateurs voir tout le monde notamment sur windows et ainsi l'interaction entre chacune des composantes. Alors, Nous excluons le risque de créer un package dépendant d'un système d'exploitation PC en programmant sous java. En effet, l'adage de Sun la société créatrice de java est: "Write once, run anywhere". Une solution est créer un fichier .jar à partir de javafx exécutable possédant une version supérieur ou égal à java 8.

<https://www.jetbrains.com/help/idea/packaging-javafx-applications.html#troubleshoot>

Risques développement interface graphique:

javafx vs swing

-> le superviseur maîtrise swing mais pas javafx ainsi pas d'aide sur javafx  
solution:

encapsuler la partie swing dans javafx

<https://openjfx.io/javadoc/19/javafx.swing/javafx/embed/swing/package-summary.html>

-> apprendre en autodidacte javafx

-> javafx possède une application pour faire des graphs

<https://github.com/HanSolo/charts>

-> la facilité de créer un projet de type javafx sur intellij

-> javafx possède son propre WindowBuilder dénommer SceneBuilder

<https://gluonhq.com/products/scene-builder/>

Risque portant sur la pédagogie

pédagogie statistique vs pédagogie de l'outil logiciel

## 2.2.2.Critères d'acceptabilité et de réception

- Les données sont toutes bien récupérées.
- Les résultats retournés sont justes.
- Les exercices d'apprentissages ne comportent pas de fautes.

## 3. Contraintes

Prise en main du framework JavaFX.

Un ordinateur comportant java.

## 4. Déroulement du projet

### 4.1. Planification

Représenter les grandes phases du projet et les étapes principales

- A. Définition de la structure du code et des différentes fonctions
- B. Création de la partie exploitation des données :
  - a. Développement des fonctions
  - b. Création de l'interface
- C. Création de la partie pédagogique :
  - a. Développement des fonctions
  - b. Création de l'interface

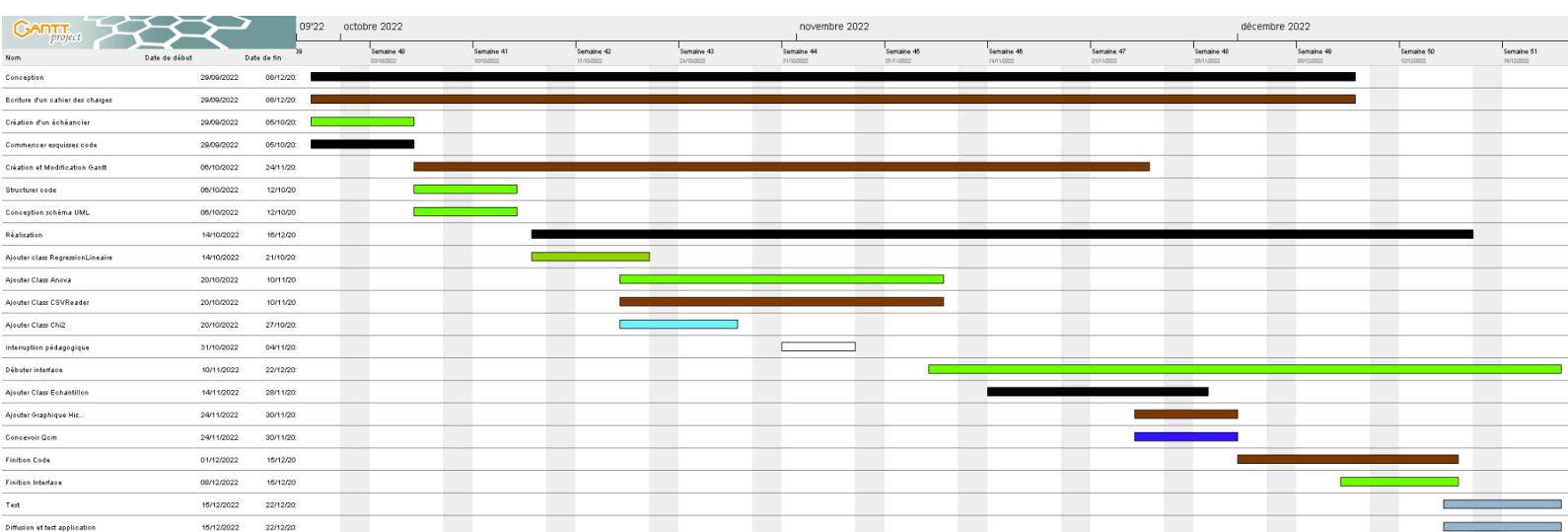


Diagramme Prévisionnel du déroulement du projet

### 4.2. Ressources

Groupe de 3 personnes travaillant sur leur ordinateur personnel.

## 5. Annexes

Lister et joindre au cahier des charges les éventuels documents que le client peut mettre à disposition

## 6. Glossaire

Définit l'ensemble des termes spécialisés du document

## 7. Références

Indique les références bibliographiques vers d'autres documents apportant des informations complémentaires

*Math – The Commons Math User Guide - Statistics*. (s. d.). Consulté le 29 septembre 2022, à l'adresse <https://commons.apache.org/proper/commons-math/userguide/stat.html>

<https://github.com/kermitt/ANOVA> ANOVA réaliser en java

## 8. Index

Liste les mots-clés du document et où les trouver dans celui-ci