

Cahier de charges

---

## **Système de gestion financière**

---

Développement informatique avancé : application

*Réalisé par :*  
Gauthier Adelin  
Classe : 3TL1

30 Juin 2020

## Table des matières

<b>1</b>	<b>Contexte</b>	<b>2</b>
<b>2</b>	<b>Objectifs</b>	<b>2</b>
<b>3</b>	<b>Contraintes techniques</b>	<b>2</b>
<b>4</b>	<b>Les besoins fonctionnels</b>	<b>3</b>
<b>5</b>	<b>Organisation</b>	<b>3</b>

## 1 Contexte

Il peut-être aujourd'hui difficile de gérer son budget tant les sources de revenu et de dépenses sont multiples. Il serait donc intéressant de posséder un outil capable de résumer quotidiennement toutes les transactions réalisées par un utilisateur lambda.

## 2 Objectifs

Le but de ce projet serait de mettre en place un système permettant à l'utilisateur de gérer efficacement ses revenus ainsi que ses dépenses. Cela lui permettra d'avoir une vue d'ensemble sur ses finances et d'en identifier et éliminer les dépenses superflues afin de mieux les répartir, voir même de les éliminer. Cela serait donc bénéfique pour l'utilisateur n'ayant pas la motivation ni l'énergie de s'occuper de son budget qui pourrait économiser son argent. Le système permettrait de rentrer des informations sur les dépenses et sur ses revenus et fournirait un graphique résumant tous ces mouvements financiers.

## 3 Contraintes techniques

Voici les contraintes à satisfaire pour la réalisation de l'application :

**Présence d'une interface graphique :** Celle-ci sera réalisée à l'aide de WindowBuilder. La difficulté résidera essentiellement dans la réalisation du graphique récapitulatif des dépenses et de la légende correspondante.

**Respecter l'architecture MVC :** L'architecture modèle-vue-contrôleur est un motif d'architecture logicielle destiné aux interfaces graphiques. Le motif est composé de trois types de modules ayant trois responsabilités différentes : le modèle qui contient les données à afficher, une vue qui contient la présentation de l'interface graphique, un contrôleur qui contient la logique concernant les actions effectuées par l'utilisateur.

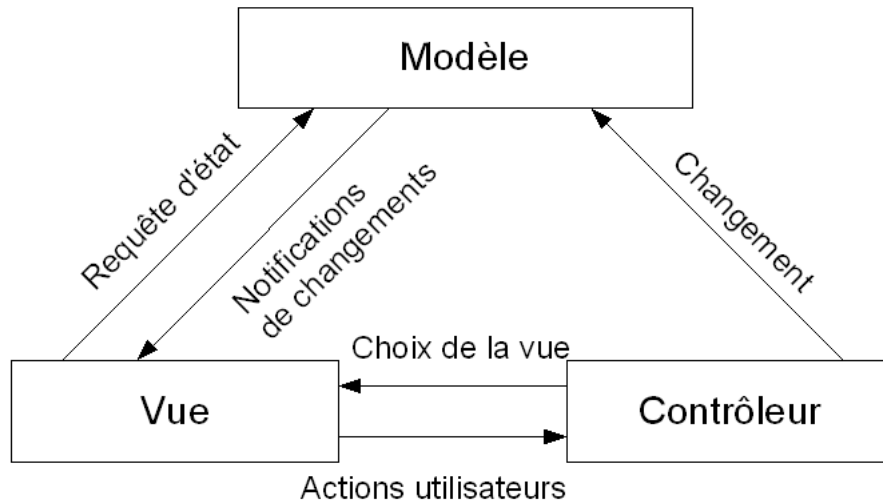


FIGURE 1: Architecture MVC

**Comporte au minimum une Collection ou une Map :** La nature du projet rend inévitable l'utilisation d'une structure de données qui reprendra donc les différents éléments financiers à placer dans le graphique récapitulatif. La technologie que j'utiliserai reste encore à définir mais panchera certainement pour une Map.

**Comporte au minimum un héritage :** La notion d'héritage sera mise en place dans les dépenses et les revenus.

**Comporte au minimum une interface :** La notion d'interface sera mise en place dans les dépenses et les revenus.

## 4 Les besoins fonctionnels

L'utilisateur pourra

- Ajouter un revenu
- Ajouter une dépense
- Consulter le graphique récapitulatif de ses transactions financières personnelles
- Consulter son solde

## 5 Organisation