

# *Cahier des Charges*

---

*Visualisation de données de crypto-monnaies  
avec le package Highcharts de Vue.js*

*ARBAUT Jean-Baptiste*

*TROPEL Célia*

*M2 MIASHS IC*

*Université Grenoble Alpes*

# Cahier des Charges

*Visualisation de données de crypto-monnaies avec le package Highcharts de Vue.js*

## Versionnage :

Version	Date	Auteur(s)	Commentaires
1	26/09/25	JAR et CTR	Attente des retours de l'activité d'empathie.

## Validation du document :

Version		Date	
Auteurs et signature	ARBAUT Jean-Baptiste		TROPEL Célia

## Sommaire

<b>1. Introduction.....</b>	<b>3</b>
1.1. Objectifs et méthodes.....	3
1.2. Documents de référence.....	3
<b>2. Guide de lecture.....</b>	<b>3</b>
<b>3. Contexte du projet.....</b>	<b>4</b>
<b>4. Concepts de base.....</b>	<b>4</b>
<b>5. Description de la demande.....</b>	<b>5</b>
5.1. Page principale.....	5
5.2. Page détaillée (vue individuelle).....	5
<b>6. Contraintes.....</b>	<b>5</b>
6.1. Contraintes de coûts.....	5
6.1. Contraintes de délais.....	6
6.2. Contraintes matérielles.....	6
6.3. Contraintes techniques.....	6
<b>7. Déroulement du projet.....</b>	<b>6</b>
<b>8. Glossaire.....</b>	<b>7</b>
<b>9. Références.....</b>	<b>7</b>
<b>10. Index.....</b>	<b>7</b>

# 1. Introduction

Ce projet s'inscrit dans le cadre du cours d'Interaction Homme-Machine 2, du Master 2 MIASHS, parcours Informatique et Cognition.

Ce document a pour objectif de définir le périmètre, les besoins et les modalités de réalisation d'un prototypage d'une application web de visualisation et d'analyse des crypto-monnaies.

## 1.1. Objectifs et méthodes

Voici les objectifs du projet :

- Mener un développement en respectant les principes ergonomiques vue en cours.
- Concevoir une interface claire et intuitive pour la consultation de données sur les crypto-monnaies.
- Proposer des représentations graphiques et des tableaux adaptés à différents niveaux de détail.

Pour cela, les méthodes suivantes seront utilisées :

- Méthodologie de conception IHM : Scénarios d'utilisation, modèle de l'utilisateur cible, modèle de tâches et prototypages avec différents degré de fidélité.
- Développement front-end en Vue.js.
- Utilisation du package HighCharts Vue.js pour les représentations graphiques.
- Récupération des données via une API.

## 1.2. Documents de référence

- 

# 2. Guide de lecture

Ce guide est destiné à être lu par l'enseignant encadrant le module et les étudiants participant au projet. Voici les différentes parties de ce document, chacune doit être lue par toutes les parties prenantes du projet :

- Une introduction présentant le contexte et les objectifs.
- Les concepts de base nécessaires à la compréhension technique.
- La description de la demande, qui détaille les fonctionnalités attendues.
- Les contraintes (coûts, délais, matérielles et techniques).
- Le déroulement du projet, qui reprend la méthodologie de conception et les livrables associés.
- Des sections de référence (glossaire, bibliographie, index).

### 3. Contexte du projet

Dans le cadre du module Interaction Homme-Machine 2, il est demandé de développer l'IHM d'un réseau social, d'une application ou d'un site web permettant à la fois de découvrir une nouvelle technologie et de rester en cohérence avec les projets professionnels futurs des étudiants.

Nous travaillons en binôme et nous avons tous deux une dominante en analyse et visualisation de données dans notre parcours antérieur. Ce projet constitue donc une passerelle entre le développement web, la conception IHM et la visualisation de données, ainsi cela est en concordance avec nos deux projets professionnels.

Le choix du thème s'est porté sur les crypto-monnaies, car ce domaine offre un large potentiel en matière de visualisation de séries temporelles et de données clés. En s'intéressant aux plateformes existantes, nous avons retenu que les interfaces sont souvent trop complexes pour les utilisateurs novices,

Notre projet vise ainsi à proposer une interface :

- Compréhensible par un débutant.
- Ergonomique, avec une navigation simple et lisible.
- Analytique, où la visualisation des données est placée au centre de l'expérience utilisateur.

### 4. Concepts de base

Dans cette partie, nous présentons les concepts nécessaires à la compréhension du projet et du présent document.

Une crypto-monnaie est un actif numérique reposant sur une blockchain, qui est une technologie de registre distribué permettant de stocker et sécuriser les transactions de manière décentralisée. Les principaux indicateurs utilisés dans l'analyse des crypto-monnaies sont :

- Le cours, c'est-à-dire le prix actuel de la crypto-monnaie
- Le volume, qui correspond au montant échangé sur une période donnée.
- La capitalisation boursière, qui représente la valeur totale du marché d'une crypto-monnaie (le prix par le nombre de tokens en circulation).

Afin d'aider les utilisateurs à anticiper l'évolution des cryptos, l'application proposera également des prévisions statistiques, c'est-à-dire des projections basées uniquement sur des données historiques, sans prise en compte de facteurs externes (politiques, sociaux, géopolitiques, etc.).

Sur le plan technique, le projet se concentre sur la conception du front-end, c'est-à-dire la partie visible de l'application et son interaction avec l'utilisateur. Pour cela, nous utilisons :

- Vue.js, un framework JavaScript permettant de développer des interfaces web réactives et modulaires.
- HighCharts, une bibliothèque de visualisation de données intégrée à Vue.js, adaptée aux séries temporelles et aux graphiques financiers.

Ce travail s'inscrit dans le champ de l'Interaction Homme-Machine (IHM), dont l'objectif est de concevoir des interfaces ergonomiques, simples d'utilisation et centrées sur les besoins des utilisateurs.

## 5. Description de la demande

### 5.1. Page principale

- Présentation générale du site web.
- Chiffres clés globaux : capitalisation totale, volume global, tendances.
- Focus sur une crypto sélectionnée : valeurs synthétiques et graphiques.
- Tableau récapitulatif : liste des cryptos avec informations clés (prix, variation, capitalisation).

### 5.2. Page détaillée (vue individuelle)

- Description de la crypto.
- Chiffres clés : prix, variation, capitalisation, volume.
- Graphiques interactifs :
  - Affichage du cours, capitalisation et volume.
  - Sélection de la durée (personnalisée, ex : 7 jours, 1 mois, 1 an).
  - Zoom/dézoom.
  - Comparaison avec jusqu'à 3 autres cryptos.
  - Prévisions statistiques.
- Où l'acheter : liens vers plateformes.
- Conversion en devise (USD, EUR, etc.).
- Suggestions de cryptos similaires.

## 6. Contraintes

### 6.1. Contraintes de coûts

Le projet ne dispose pas de budget dédié : seuls des outils et frameworks gratuits ou open-source doivent être utilisés.

L'API choisie pour les données devra proposer un accès gratuit ou en version limitée suffisante pour les besoins du prototype. Par exemple, CoinGecko offre un accès libre à ses données dans la limite des 50 requêtes par minute.

### 6.1. Contraintes de délais

Le projet doit être réalisé selon le calendrier du module Interaction Homme-Machine 2. Les livrables doivent être produits dans l'ordre défini : cahier des charges, scénarios, modèles, prototypes.

### 6.2. Contraintes matérielles

L'application doit être consultable sur les navigateurs web les plus courants (Chrome, Firefox, Edge).

### 6.3. Contraintes techniques

Utilisation de Vue.js et HighCharts pour la visualisation, sans développement de back-end complet. Les rendus doivent respecter les principes d'IHM vus en cours.

## 7. Déroulement du projet

Le projet se déroule en différentes étapes liées aux livrables attendus :

Numéro d'étape	Contenu et livrable	Livrable
1	Brainstorming des idées et choix du sujet en lien avec le module IHM 2 et nos projets professionnels.	/
2	Conception et réalisation d'activités d'empathie pour identifier les besoins et attentes des utilisateurs cibles.	/
3	Définition formelle des besoins : cadrage du projet, description des fonctionnalités, contraintes et objectifs.	Cahier des charges
4	Élaboration de parcours types pour représenter la manière dont différents utilisateurs interagiront avec l'application.	Scénarios d'utilisation
5	Définition du profil type de l'utilisateur cible (caractéristiques, attentes, besoins).	Modèle utilisateur cible
6	Analyse et formalisation des tâches que l'utilisateur devra effectuer pour atteindre ses objectifs.	Modèle de tâches
7	Conception d'une première maquette abstraite : structure générale de l'interface, organisation des éléments, sans détails graphiques.	Prototype abstrait
8	Tests utilisateurs des maquettes avec les scénarios d'usage : validation de l'ergonomie, identification des points d'amélioration.	/
9	Développement du prototype concret : maquette	Prototype

	interactive réalisée en Vue.js avec intégration de graphiques HighCharts.	concret
10	Tests utilisateurs du prototype interactif : vérification de la navigation, de la lisibilité et des graphiques.	/

## 8. Glossaire

**API** : Interface de Programmation Applicative, qui permet à une application de communiquer avec une autre.

**Blockchain** : Technologie de registre distribué permettant de stocker et sécuriser des transactions de manière décentralisée.

**Capitalisation boursière** : Valeur totale d'une crypto-monnaie, calculée comme le prix multiplié par le nombre de tokens en circulation.

**Front-end** : Partie visible d'une application web, avec laquelle l'utilisateur interagit.

**HighCharts** : Bibliothèque JavaScript spécialisée dans la visualisation de données.

**IHM (Interaction Homme-Machine)** : Discipline visant à concevoir des interfaces ergonomiques, simples et adaptées aux utilisateurs.

**Prototype abstrait** : Représentation simplifiée de l'interface (ex. maquette papier).

**Prototype concret** : Maquette interactive de l'interface (ex. réalisée en Vue.js).

**Vue.js** : Framework JavaScript permettant de développer des interfaces web dynamiques et modulaires.

## 9. Références

- Support de cours Interaction Homme-Machine 1 et 2 du Master MIASHS, parcours Informatique et Cognition, de l'UGA.
- Documentation officielle [Vue.js](#), [HighCharts](#).
- Documentation [CoinGecko API](#).

## 10. Index