

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.
2	04/10/25	JAR et CTR	Finalisation du cahier des charges suite à l'activité d'empathie.
3	18/10/25	JAR et CTR	Relecture du document.

Validation du document :

Version	3	Date	06/10/25
Auteurs et signature	ARBAUT Jean-Baptiste 	TROPEL Célia 	

Sommaire

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

1. Introduction

Ce projet s'inscrit dans le cadre du cours d'Interaction Homme-Machine 2, du Master 2 MIAHS, 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 ou simulation des données.

1.2. Documents de référence

Ce cahier des charges s'appuie sur les retours d'une activité d'empathie que nous avons mené.

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 du projet (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

Fonctionnalité	Description
Présentation générale	Introduction au site et aux fonctionnalités principales.
Visualisation du portefeuille	Affichage de l'évolution du portefeuille utilisateur : bénéfices, pertes, historique des performances.
Chiffres clés globaux	Données synthétiques : capitalisation totale, volume global échangé, tendances principales du marché.
Focus sur une crypto	Affichage détaillé d'une crypto sélectionnée : cours actuel, variation, volume, capitalisation, graphiques.
Tableau récapitulatif	Liste des crypto-monnaies avec informations clés (prix, variation, capitalisation), possibilité de filtrer et comparer.
Informations de la communauté	Espace social avec publication d'articles et commentaires.
Personnalisation de l'accueil	Choix des informations prioritaires affichées (portefeuille, tendances, ou combinaison des deux).

Pour la visualisation du portefeuille, il faut :

- Être connecté à un compte, sinon cette partie ne doit pas être visualisable.
- Un lien facile aux portefeuilles de différentes plateformes (via une clé).
- Une possibilité de personnalisation pour visualiser uniquement ce que la personne souhaite.

5.2. Page détaillée (vue individuelle)

Fonctionnalité	Description
Description de la crypto	Présentation complète : créateur, objectifs, historique, utilité du projet.
Chiffres clés	Prix, variation, capitalisation, volume d'échanges.
Graphiques interactifs	Visualisation du cours, de la capitalisation et du volume. Options de sélection de durée (1 jour à 1 an ou plus), zoom/dézoom, comparaison avec max 3 cryptos.
Prévisions statistiques	Projection des tendances basée sur les données historiques, sans facteurs externes.
Commentaires de la communauté (optionnels)	Permet d'interagir avec la communauté en postant, lisant ou répondant à des commentaires sous les pages détaillées des crypto-monnaies.
Informations pratiques	Liens directs vers plateformes d'achat.
Conversion en devise	Conversion automatique des valeurs en devises courantes (USD, EUR, etc.).
Suggestions de cryptos similaires	Recommandations de cryptos proches (par usage, capitalisation ou popularité).

5.3. Partie Communauté :

Fonctionnalité	Description
Lecture d'articles, publications	Offrir la possibilité de lire des articles communautaires sur les crypto-monnaies.
Possibilité de commenter, de lire les commentaires et d'y répondre	Permet d'interagir avec la communauté en postant, lisant ou répondant à des commentaires sous les articles.

5.4. Autres fonctionnalités souhaitées :

Fonctionnalité	Description
----------------	-------------

Connexion/compte utilisateur	Création et gestion d'un compte, sauvegarde des préférences et personnalisation de l'expérience.
Compatibilité multi-supports	Optimisation pour smartphone (consultation rapide) et ordinateur (analyse détaillée).
Notifications paramétrables	Alertes personnalisables (variation de prix, tendances, nouveaux commentaires).

6. Contraintes du projet

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. Si cela n'est pas assez conséquent, les données seront simulées.

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 interagissent 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 interactive réalisée en Vue.js avec intégration de graphiques HighCharts.	Prototype 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.

Black-end : Le back-end désigne la partie invisible d'une application qui gère la logique, les données et la communication avec le serveur. Il traite les requêtes des utilisateurs, interagit avec la base de données et renvoie les résultats au front-end.

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 : (Appelé aussi basse fidélité) Représentation simplifiée de l'interface (maquette papier).

Prototype concret : (Appelé aussi haute fidélité) Maquette interactive de l'interface (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

Notion	Page(s)
API	3, 7, 8, 9
Back-end	7, 8
Blockchain	4, 8
Capitalisation boursière	4, 8
Front-end	3, 4, 8
HighCharts	1, 2, 3, 5, 7, 8, 9
IHM	3, 4, 5, 7, 8
Prototype abstrait	8
Prototype concret	8
Vue.js	1, 2, 3, 5, 7, 8, 9