11 février 2022

Document de conception

Par Nikolas Ouimet et Yvanoski Sanon

Présentation de conception de Crypto Newbie

Table des matières

[Maquettes du projet 2](#_Toc97119634)

[Page d’accueil 2](#_Toc97119635)

[Page de connexion 3](#_Toc97119636)

[Page de création du compte 4](#_Toc97119637)

[Page des listes 5](#_Toc97119638)

[Page d’achat 6](#_Toc97119639)

[Page d’achat (confirmation) 7](#_Toc97119640)

[Page de vente 8](#_Toc97119641)

[Page de vente (confirmation) 9](#_Toc97119642)

[Page du portfolio 10](#_Toc97119643)

[Page de l’historique des transactions 11](#_Toc97119644)

[Page du profil de l’utilisateur 12](#_Toc97119645)

[Page de gestion d’utilisateurs (Admin) 14](#_Toc97119646)

[Diagramme de cas d’usage 15](#_Toc97119647)

[Diagramme de classes 16](#_Toc97119648)

[Schéma de la structure de données externe (base de données) 17](#_Toc97119649)

[Structures de données 18](#_Toc97119650)

[Patrons de conception 18](#_Toc97119651)

[Expression régulière 19](#_Toc97119652)

[Algorithme 19](#_Toc97119653)

[Calculs mathématiques 19](#_Toc97119654)

# Maquettes du projet

Les maquettes représentent les éléments détaillés de chaque interface graphique du projet.

## Page d’accueil

Avant la connexion :

Une image contenant texte, capture d’écran, moniteur, écran

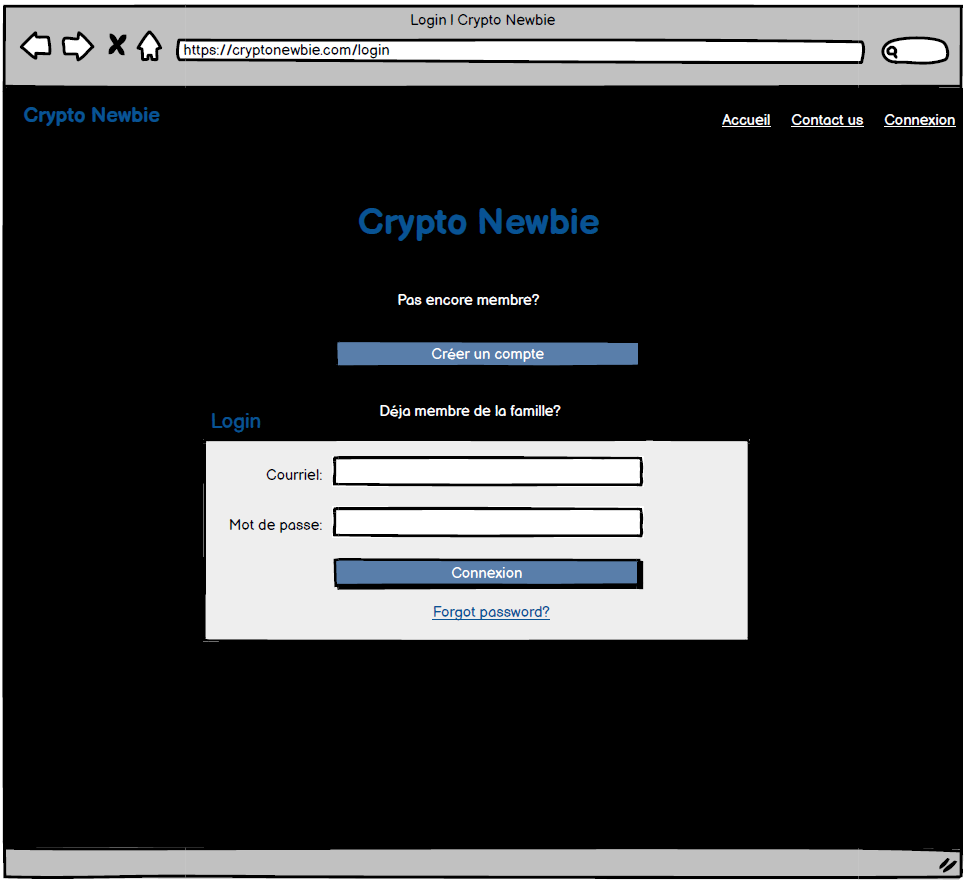
Description générée automatiquement

Après la connexion :

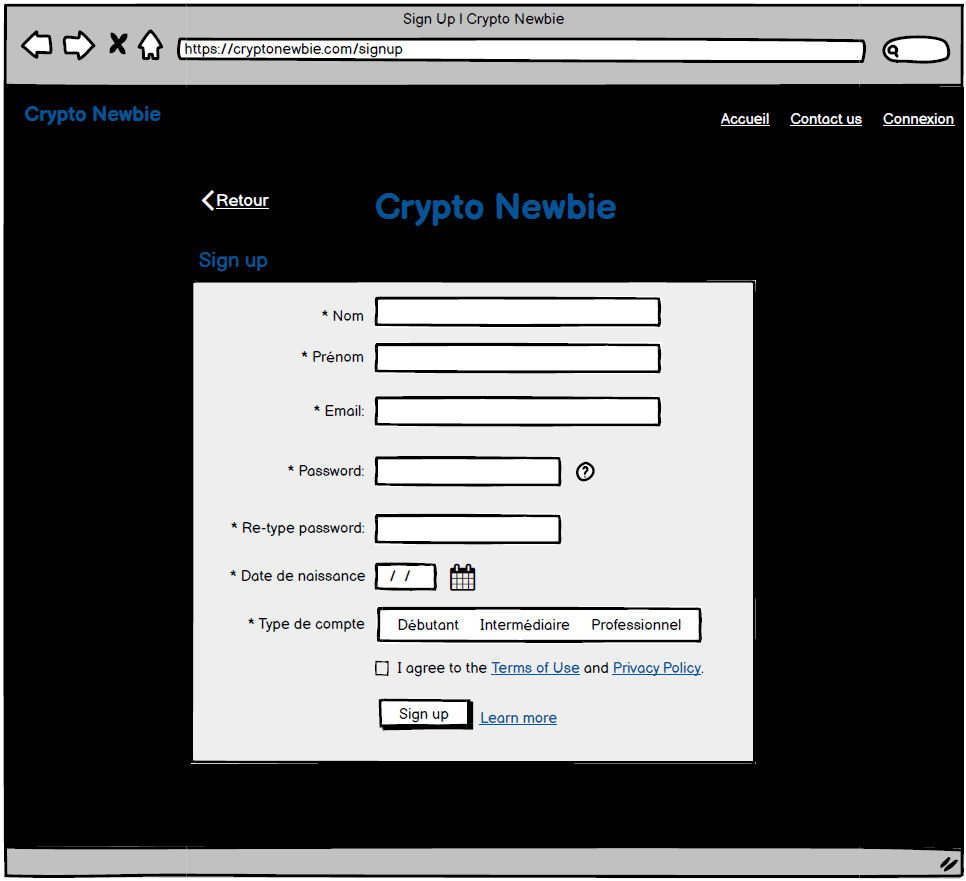
Une image contenant texte, capture d’écran, moniteur, écran

Description générée automatiquement

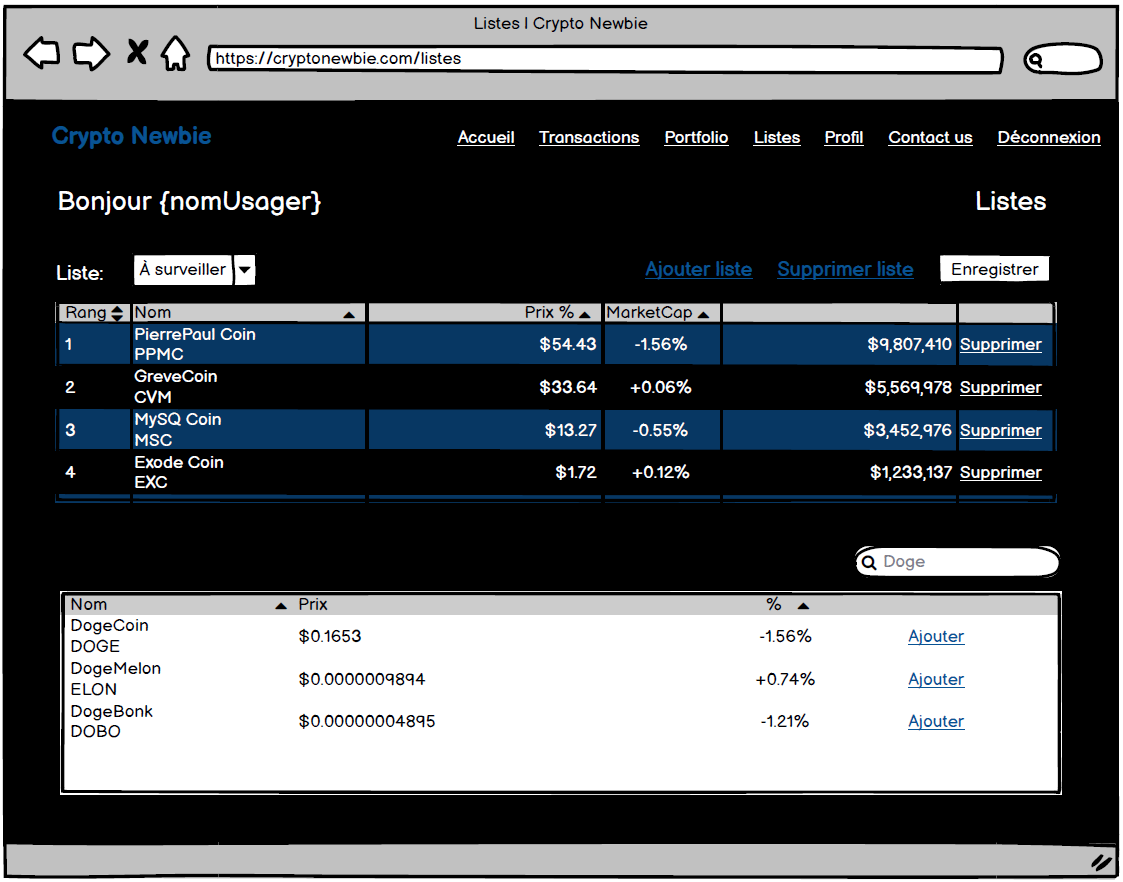
## Page de connexion



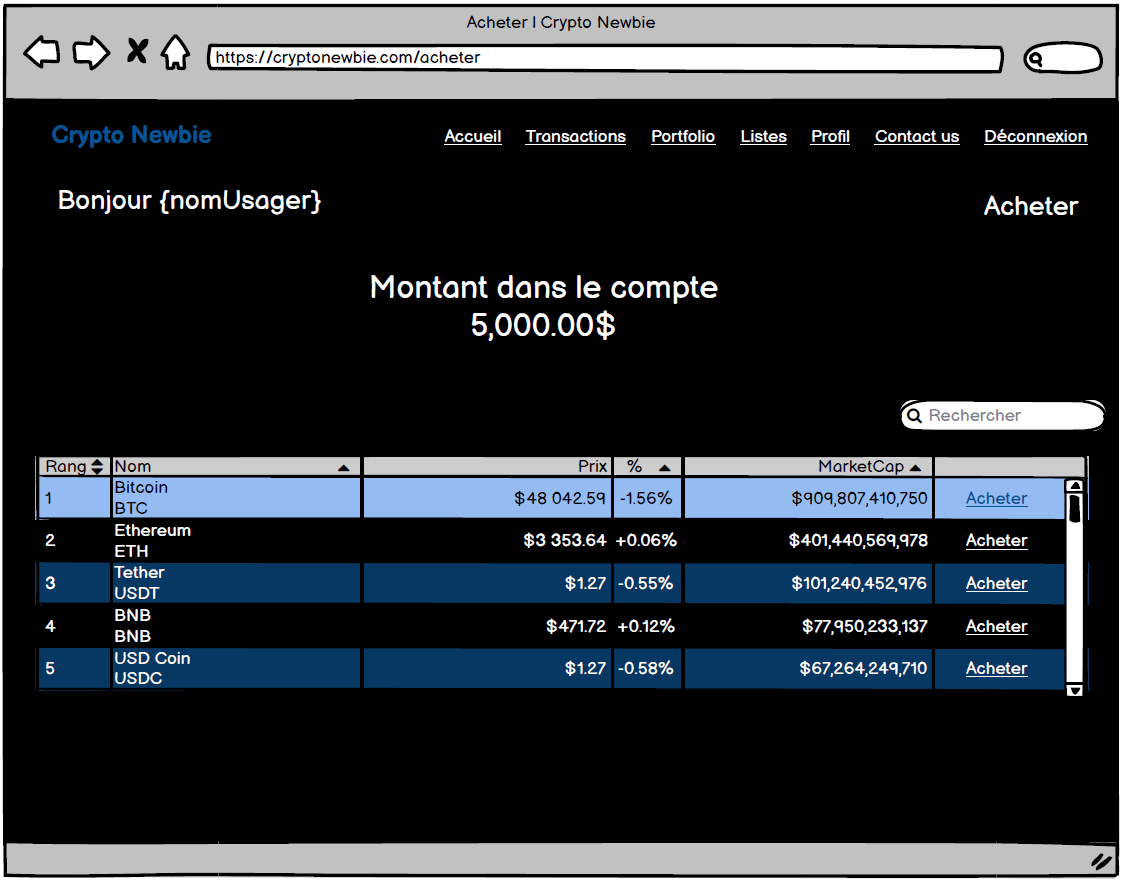
## Page de création du compte



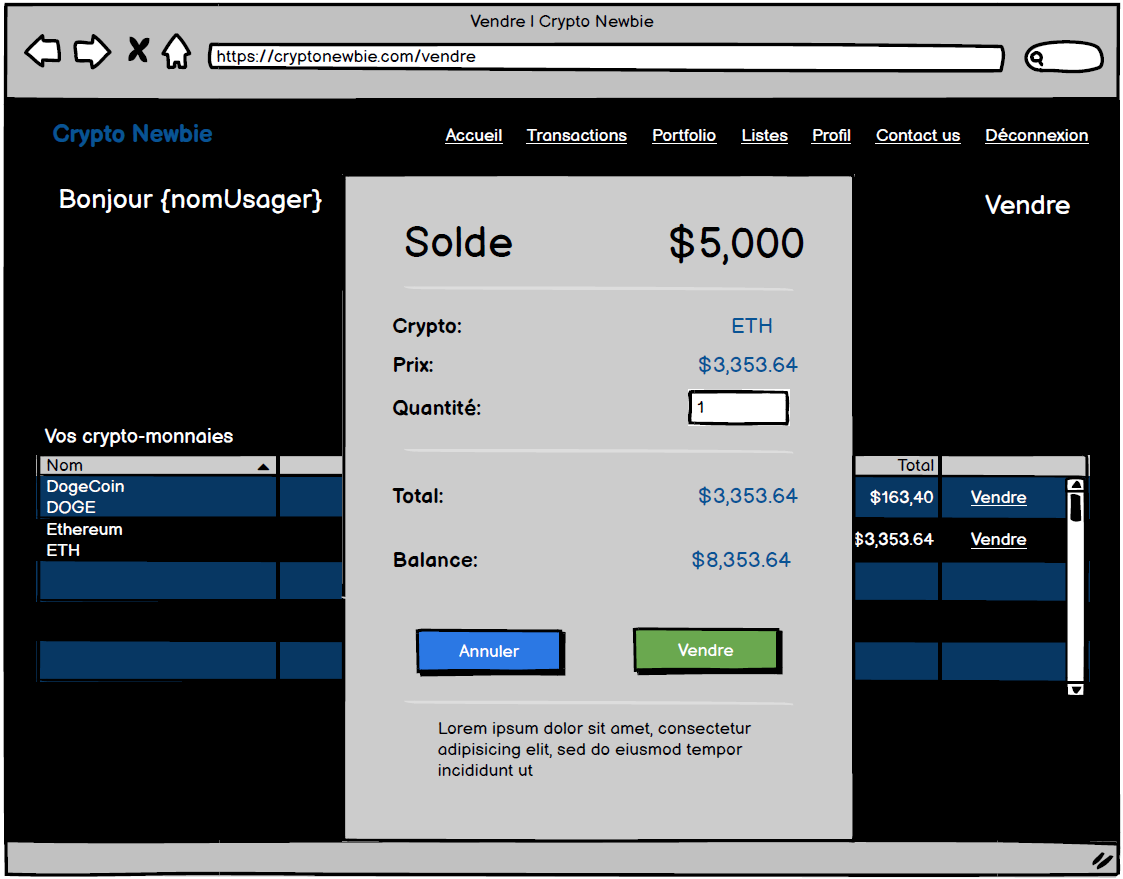
## Page des listes



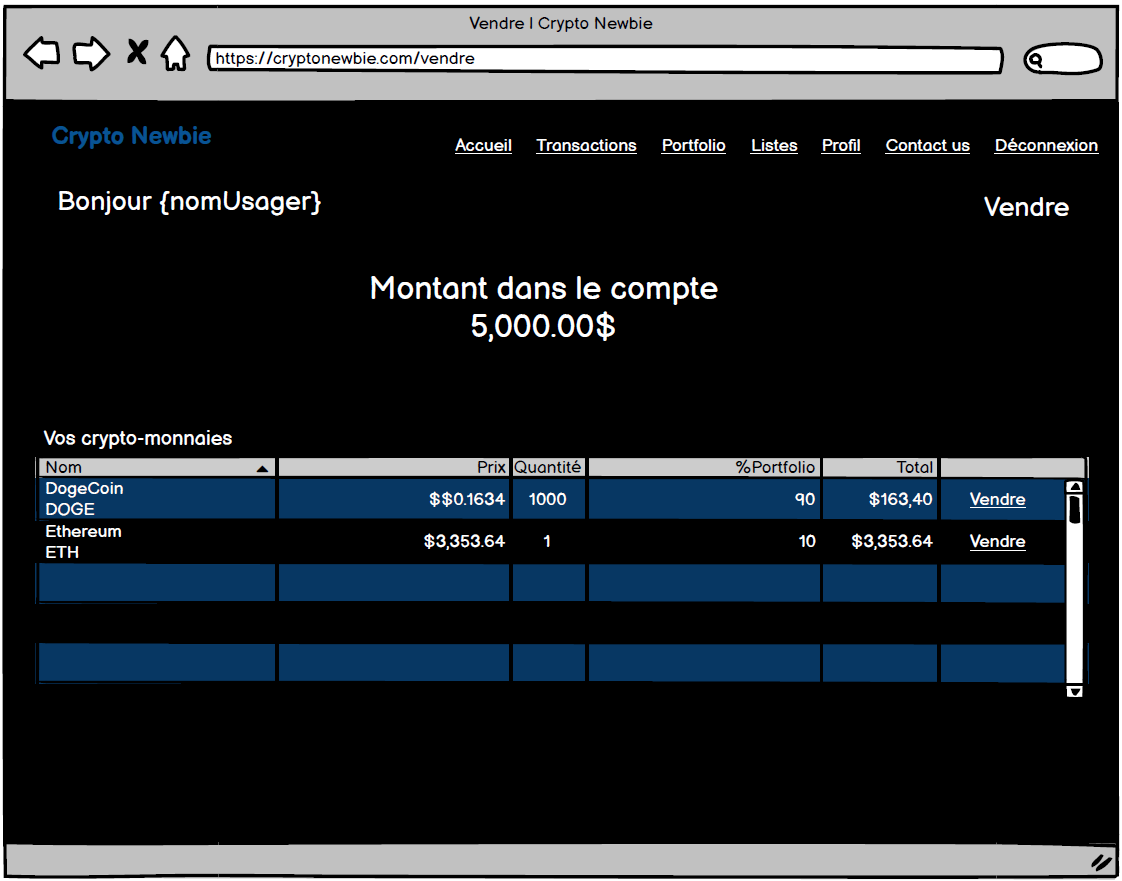
## Page d’achat



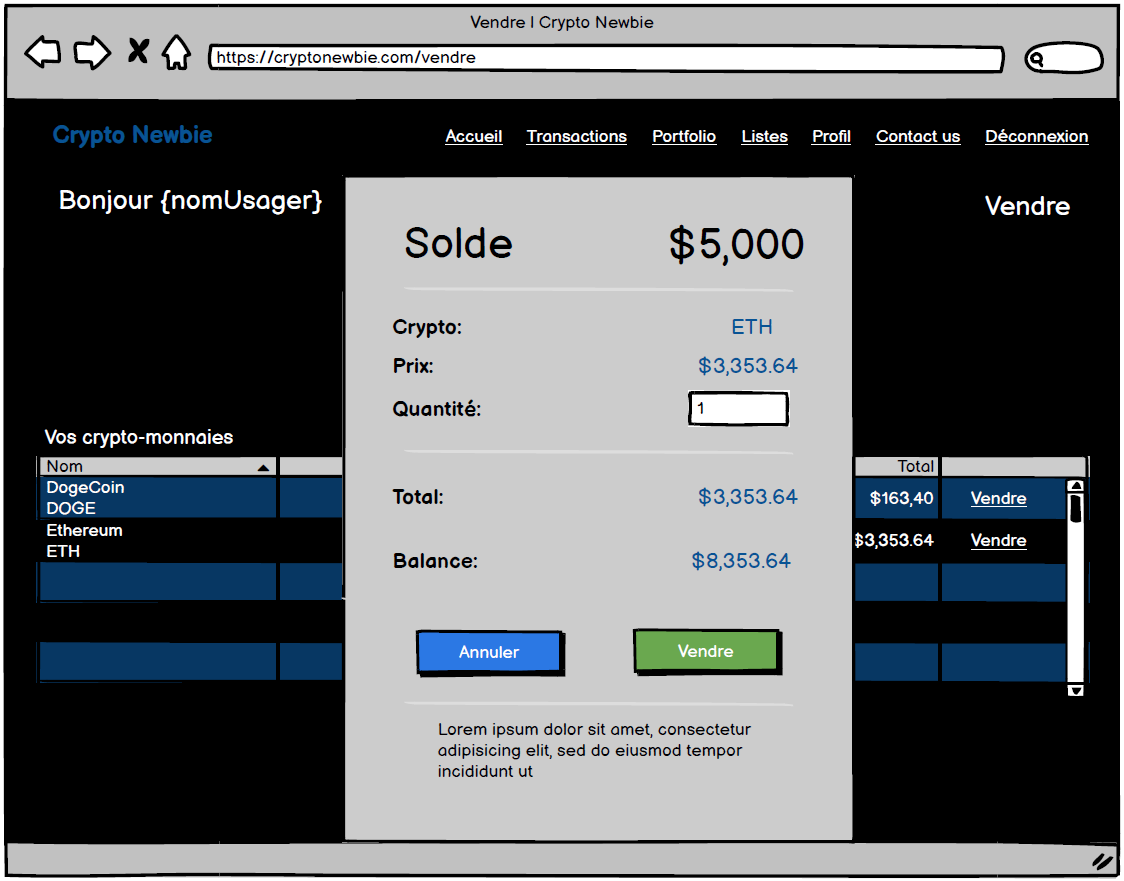
## Page d’achat (confirmation)



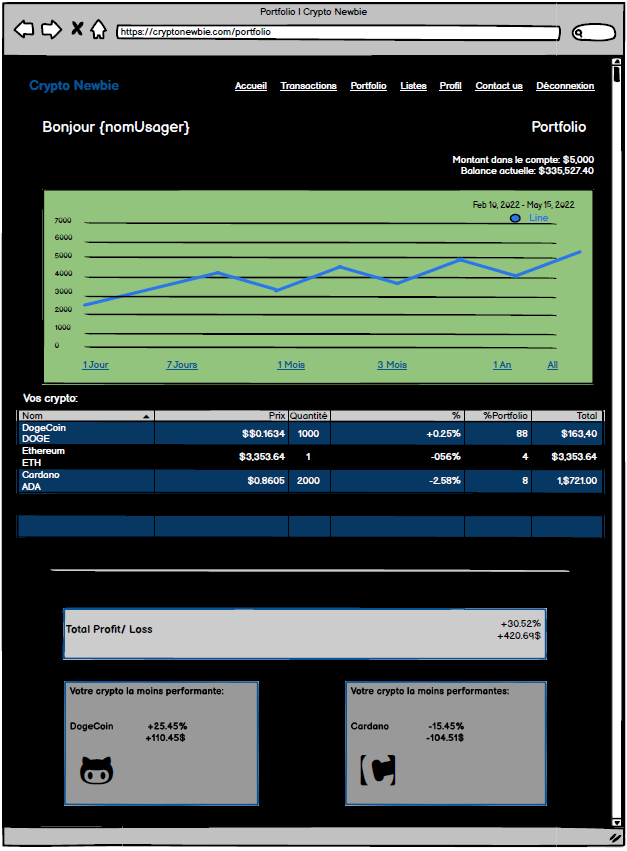
## Page de vente



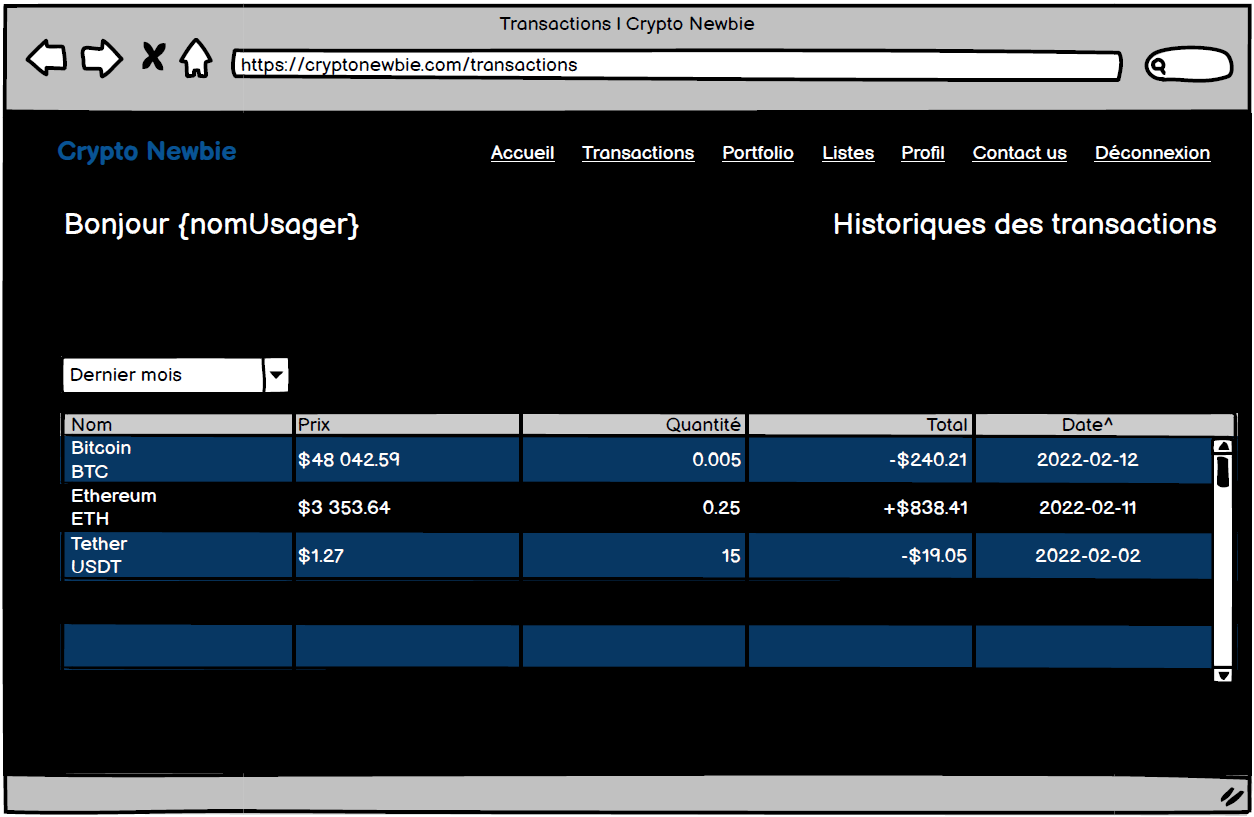
## Page de vente (confirmation)



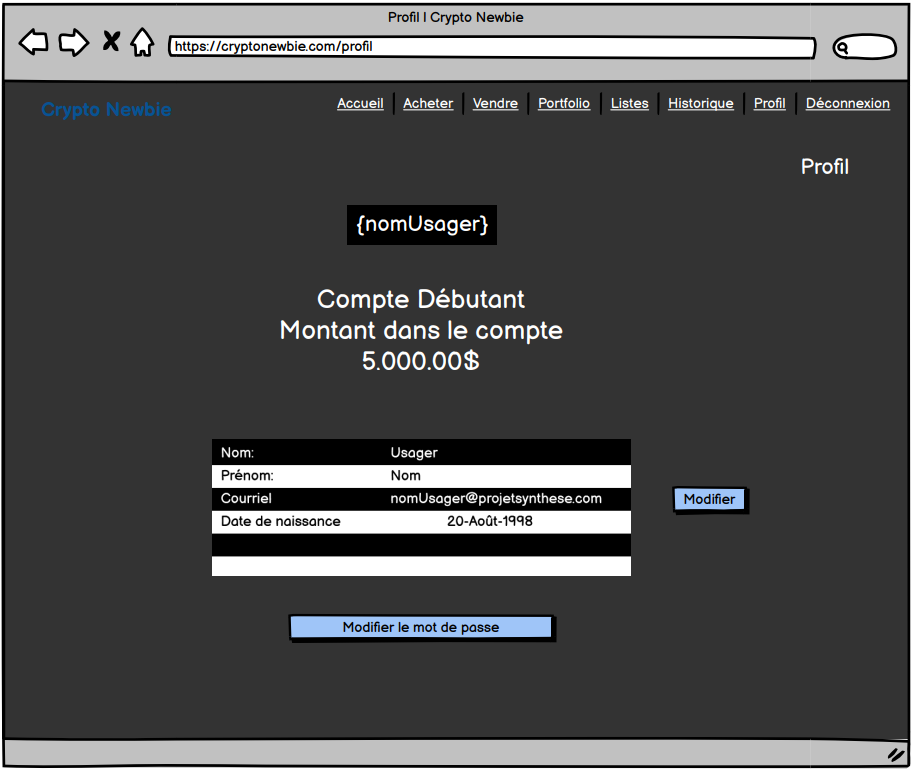
## Page du portfolio



## Page de l’historique des transactions



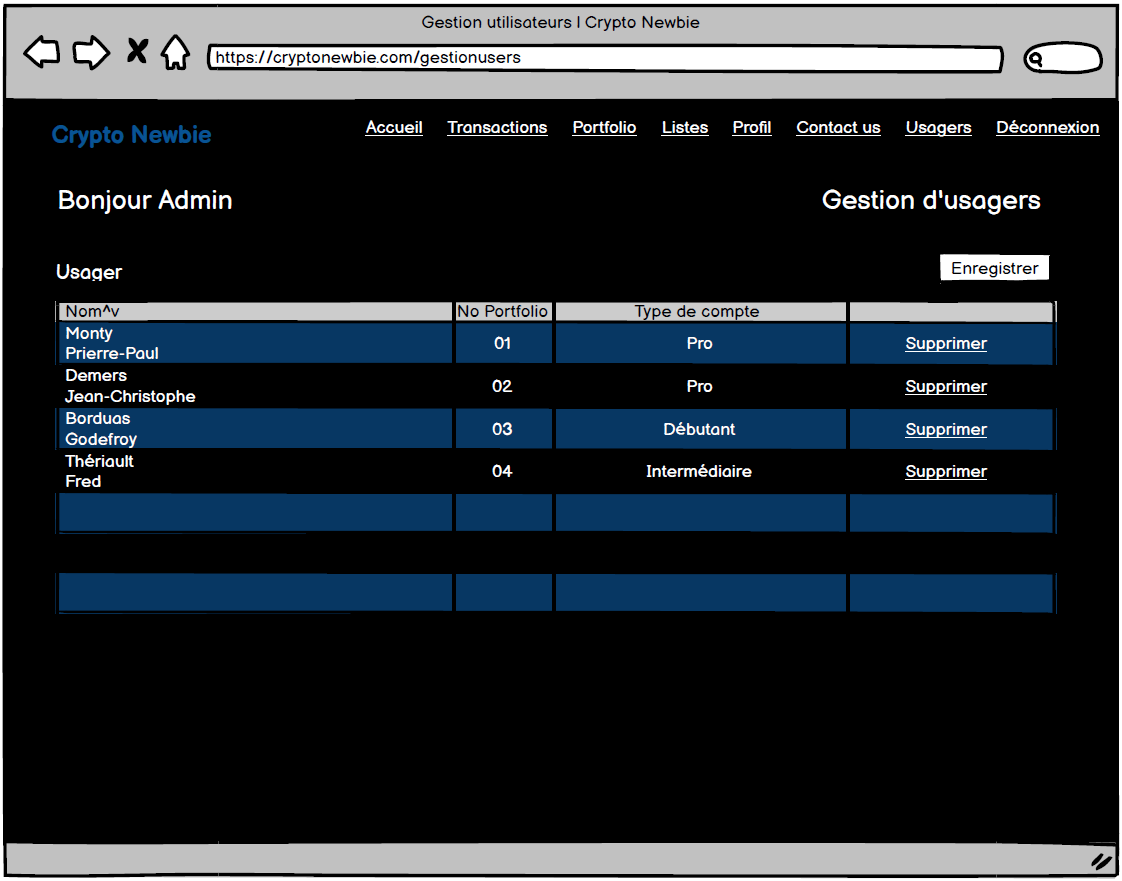
## Page du profil de l’utilisateur



Une image contenant texte

Description générée automatiquement

## Page de gestion d’utilisateurs (Admin)

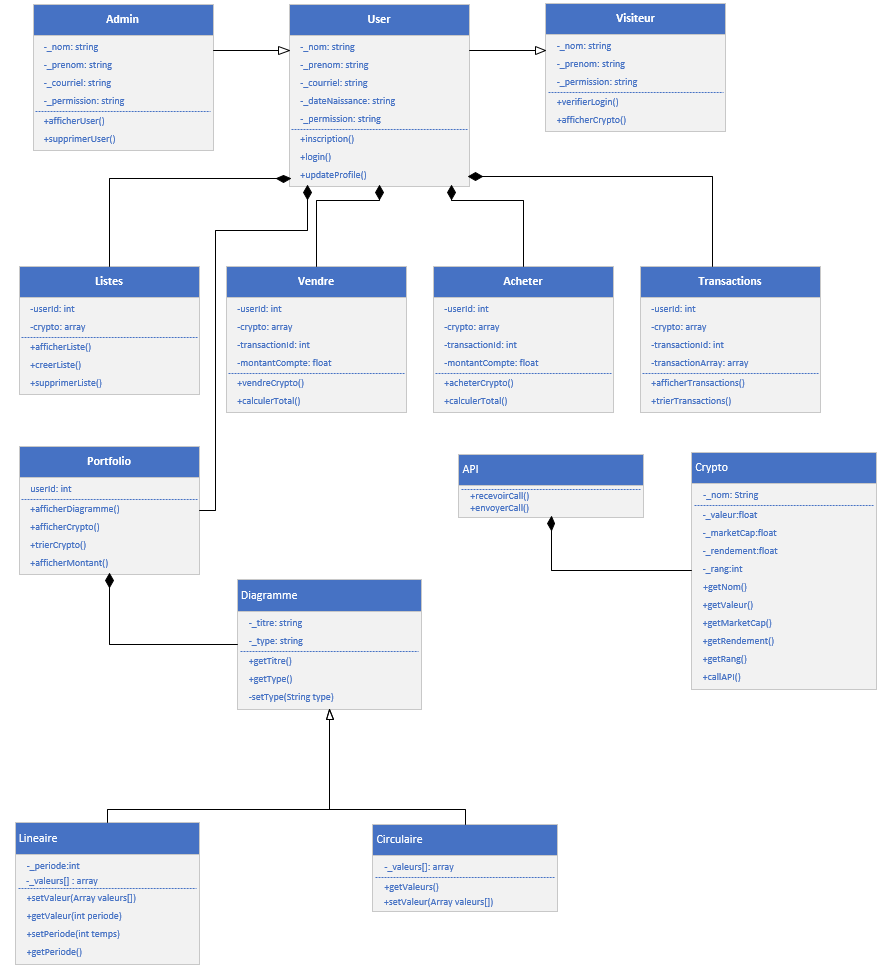


# Diagramme de cas d’usage

Diagram

Description automatically generated

# Diagramme de classes



# Schéma de la structure de données externe (base de données)

Graphical user interface, application, Teams

Description automatically generated

# Structures de données

Pour le projet, nous allons utiliser les quatre structures de données suivantes:

* **Array** : Utilisé pour stocker la liste des crypto-monnaies disponibles. C’est une structure de données simple et facile à implanter. Elle est donc idéale pour une tâche aussi banale.
* **Map** : Utilisation pour lier l’utilisateur à son portfolio. À l’aide de la clé, qui pourrait être l’id de l’inscrit, on pourra facilement accéder aux propriétés de la classe portfolio.
* **Liste** **chainée** : Cette structure sera implémentée par nous. Elle sera utilisée pour stocker la liste des utilisateurs. C’est une structure parfaite pour cette tâche vu qu’elle sera dynamique et s’ajustera automatiquement s’il y a un ajout ou une suppression.

# Patrons de conception

Voici les quatre principaux patrons de conception qui seront utilisé lors de la conception de notre projet :

* **Listener (onClick, onTouch, etc.):** Le listener sera nécessaire afin de gérer les interactions de l’utilisateur à l’interface de la page Web. Par exemple, à la page d’achat, un onClick listener sera utilisé pour faire apparaître une fenêtre de confirmation de la transaction
* **Proxy :** Le proxy sera pratique afin de développer un gestionnaire correspondant à des getters pour l’objet utilisateur.
* **Visitor :** Visitor est un patron de conception qui nous permet de créer des objets pour apporter des modifications dans un autre objet. Par exemple, si on crée un objet « Permission », nous pouvons apporter les modifications de l’objet Permission à l’objet utilisateur afin que l’utilisateur puisse hériter des autorisations et fonction désiré.
* **Strategy :** Strategy permettrait d’encapsuler dans une classe notre algorithme de recherche afin de l’isoler et de pouvoir appeler ses méthodes lorsque nécessaire.

## Expression régulière

Des expressions régulières seront utilisées pour capter la chaine de texte tapée dans la barre de recherche qui servira à trouver une crypto-monnaie.

# Algorithme

Nous implanterons un algorithme de tri rapide qui permettra de changer l’ordre d’affichage des monnaies. L’utilisateur pourra trier par rang, par valeur, par croissance ou par market capitalisation (market cap).

## Calculs mathématiques

Nous ferons l’utilisations de différents calculs mathématiques comme une formule pour trouver la crypto-monnaie la plus performante celle qui performe le mien, des calculs pour suivre les gains ou les pertes d’argent ainsi qu’une représentation s’en pourcentage et des calculs pour les transactions d’achats ou de ventes.