



Projet de Fin d'Étude

Diplôme Universitaire de Technologie
Filière Informatique
Option
Génie Logiciel

Initiation à la technologie blockchain: Ethereum use case (Smart contract) et implémentation d'une application (Mobile) de tracking de la valeur d'un portefeuille.

Par:

Mohammed SADOK & Ahmed Houssam BOUZINE

Soutenu le 15/06/2022 devant le jury composé de :

Pr. Mehdi TMIMI

EST

Encadrant

Pr. Bader NEJJAR

EST

Examineur

Année Universitaire : 2021/2022

Table of Contents

Remerciement.....	5
Introduction	6
Chapitre 1 : Cadre et contexte du projet	7
1.1. Introduction	8
1.2. Enjeux et contexte du projet	8
1.3. Présentation de l'organisme d'accueil.....	8
1.4. Présentation sur les Crypto-monnaies et Blockchain	9
1.4.1 Les crypto-monnaie c'est quoi ?	9
1.4.2 Blockchain c'est quoi ?	9
1.4.3 Smart contracts c'est quoi ? :.....	11
Chapitre 2 : Présentation et cahier des charges.....	12
2.1. Introduction.....	13
2.2. Présentation du projet.....	13
2.3. Objectifs et périmètres du projet	14
2.4. Public ciblé	14
2.5. Contenu de l'application.....	14
2.6. Charte graphique : exigences et limites.....	15
2.7. Arborescence de l'application	18
2.8. Planification du projet.....	19
2.8.1 Diagramme de GANTT.....	20
2.8.2 Méthode de PERT	20
Chapitre 3 : Technologies et outils utilisés	21
3.1. Introduction	22
3.1. Technologies implémentées	22
3.2.1. Introduction :	22
3.2.2. Javascript :.....	22
3.2.3. NodeJS :.....	23

3.2.4.	Express Js :	23
3.2.5.	MySQL :	24
3.2.6.	React native :	24
3.2.7.	Expo :	25
3.2.	Outils d'implémentation	25
3.3.	Problématiques et solutions technologiques :	28
Chapitre 4 : Architecture et Conception		30
4.1.	Introduction	31
4.2.	Architecture de l'application	31
Couche présentation (premier niveau) :		31
Couche métier / business (deuxième niveau) :		31
Couche accès aux données (troisième niveau) :		32
4.3.	Conception de l'application	33
4.3.1	Pourquoi UML ?	33
4.3.2	Règles de gestion	34
4.3.3	Diagramme cas d'utilisation	34
4.3.4	Diagramme de classes	35
Chapitre 5 : Présentation de l'application		38
5.1.	Introduction	39
5.1.	Etape d'authentification	39
5.2.	Etape d'inscription	40
5.3.	Profile	41
5.4.	Market	45
5.5.	Home	48
Bibliographie		53

Table d'abréviation

IDE : Integrated Development Environment.

UML : Unified Modeling Language.

HTML : Hypertext Markup Language.

CSS : Cascading Style Sheets.

JS: JavaScript.

HTTP: HyperText Transfert Protocol.

URL: Uniform Resource Locator.

MySQL: My Structured Query Language.

GNU: GNU's Not UNIX.

UWP: Universal Windows Platform.

IOS: initialement iPhone OS.

PHP: PHP Hypertext Preprocessor.

API : Application Programming Interface

TCP: Transmission Control Protocol

POO : Programmation orientée objet

Remerciement

A cette occasion favorable nous adressons nos sincères remerciements et grâces à tous les gens qu'ils ont nous aidé à réaliser ce projet de fin d'étude.

Nous exprimons nos sincères remerciements à Mr TMIMI Mehdi ; notre encadrant pour son aide, son soutien, son guidage, son écoute, et son esprit ouvert de partage de connaissances au cours de cette période de la réalisation du projet.

Nous remercions le corps professionnel et pédagogique de l'Ecole Supérieure de Technologie de Fès qui veille sans cesse à ce que notre formation soit de qualité.

Nous exprimons également notre gratitude aux membres du jury, qui nous ont honorés en acceptant de juger ce modeste travail.

Introduction

Ce rapport constitue le fruit du travail que nous avons réalisé pendant 4 mois dans le cadre du projet de fin d'études au sein de l'Ecole Supérieure de Technologie de Fès pour appliquer l'ensemble des connaissances que nous avons acquies durant ces deux ans de formation.

Notre projet avait comme thème : Initiation à la technologie blockchain : Ethereum use case (Smart contract) et implémentation d'une application (Mobile) de tracking de la valeur d'un portefeuille.

Afin de comprendre la démarche que nous avons utilisé pour mener ce projet à son terme, notre rapport se structure de la façon suivante :

Tout d'abord, dans une première partie nous présenterons le contexte général du projet. Après, on passera à la définition générale de l'aspect de notre projet, c'est-à-dire ce qui existe et ce que notre projet va apporter. Puis dans une seconde partie, nous présenterons les outils et les technologies avec lesquels nous avons travaillé. Ensuite dans une troisième partie, nous allons aborder la tâche technique, avant que nous décrivions une démonstration du fonctionnement du projet.

Chapitre 1 :

Cadre et contexte du projet

1.1. Introduction

Comme tout premier chapitre celui-ci, s'intéressera à la présentation du cadre du projet en tant que projet de fin d'étude, tout en présentant l'organisme d'accueil et le contexte du projet.

1.2. Enjeux et contexte du projet

Dans le cadre de la formation au sein de l'Ecole Supérieure de Technologie de Fès , les étudiants de la deuxième année afin de valider leurs dernier semestre sont obligés de réaliser un projet de fin d'étude encadré par un professeur de la filière et présenter devant des jury lors de la soutenance .

Dans ce cadre le thème de notre projet de fin d'étude est : Initiation à la technologie blockchain: Ethereum use case (Smart contract) et implémentation d'une application (Mobile) de tracking de la valeur d'un portefeuille.

1.3. Présentation de l'organisme d'accueil



Figure 1 : logo de l'école supérieure de technologie

L'Ecole Supérieure de Technologie de Fès, fait partie de l'université Sidi Mohamed Ben Abdellah, délivre le DUT : Diplôme Universitaire de Technologie, qui est parfaitement bien reconnu et hautement apprécié.

Depuis l'année universitaire 2014-2015 l'ESTF est habilité à délivrer le diplôme de licence professionnelle.

Les formations dispensées représentent des solides bases pour l'avenir des diplômés grâce à la connaissance et l'ouverture sur l'entreprise marocaine, alliés à des installations et laboratoires technologiques de haut niveau et en s'appuyant sur un potentiel humain expérimenté.

Ces formations sont de nature professionnalisantes qui offrent une double opportunité. Les lauréats peuvent s'insérer rapidement dans la vie active comme ils peuvent intégrer des cycles d'études supérieures.

Plusieurs partenariats ont été noués par EST Fès pour améliorer la qualité de sa formation.

Actions intégrées – Projet Ikonos : Malte, Jordanie, Tunisie, Egypte, Algérie – Conventions de coopération avec les universités Sfax, Tunisie – Poitiers, France – Ohridski, Bulgarie – Valahia, Roumanie – Parme, Italie – Orsay, Paris – Reims.

1.4. Présentation sur les Crypto-monnaies et Blockchain

1.4.1 Les crypto-monnaie c'est quoi ?

La crypto-monnaie est un système de paiement numérique qui ne dépend pas des banques pour vérifier les transactions. C'est un système peer-to-peer qui permet à n'importe qui, n'importe où, d'envoyer ou de recevoir des paiements.

Les paiements en crypto-monnaie existent uniquement sous forme d'entrées numériques dans une base de données en ligne décrivant des transactions spécifiques. Lorsqu'on transfère des fonds en crypto-monnaie, les transactions sont enregistrées dans un grand livre public.

La crypto-monnaie est stockée dans des portefeuilles numériques, est nommée par ce nom car elle utilise le cryptage pour vérifier les transactions. Cela signifie qu'un codage avancé est impliqué dans le stockage et la transmission des données de crypto-monnaie entre les portefeuilles et vers les registres publics. Le but du cryptage est d'assurer la sécurité et la sûreté.

1.4.2 BlockChain c'est quoi ?

Les blockchains permettent de stocker et d'échanger de la valeur sur internet sans intermédiaire centralisé.

Elles sont le moteur technologique des crypto-monnaies, du Web Décentralisé et de son corollaire, la finance décentralisée. Une blockchain constitue une base de données qui contient l'historique de tous les échanges effectués entre ses utilisateurs depuis sa création. Cette base de

données est sécurisée et distribuée : elle est partagée par ses différents utilisateurs, sans intermédiaire, ce qui permet à chacun de vérifier la validité de la chaîne.

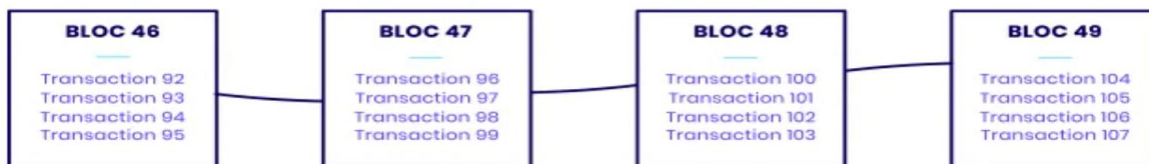


Figure 2: Exemple d'une chaîne des blocs

Une blockchain peut donc être assimilée à un grand livre comptable public, anonyme et infalsifiable. Comme l'écrit le mathématicien Jean-Paul Delahaye, il faut s'imaginer « **un très grand cahier, que tout le monde peut lire librement et gratuitement, sur lequel tout le monde peut écrire, mais qui est impossible à effacer et indestructible.** »

Certaines entreprises envisagent parfois la création de blockchains dites « privées », dont l'accès et l'utilisation sont limités à un certain nombre d'acteurs. Cela permet d'expérimenter en interne mais avec des limites en termes d'innovation (écosystème restreint) et de ROI (coût de l'infrastructure à mettre en place).

Toute blockchain publique fonctionne nécessairement avec une monnaie ou un token (jeton) programmable. Bitcoin est un exemple de monnaie programmable. Les transactions effectuées entre les utilisateurs du réseau sont regroupées par blocs. Chaque bloc est validé par les nœuds du réseau appelés les "mineurs", selon des techniques qui dépendent du type de blockchain. Dans la blockchain Bitcoin cette technique est appelée le "Proof-of-Work", preuve de travail, et consiste en la résolution de problèmes algorithmiques. Une fois le bloc validé, il est horodaté et ajouté à la chaîne de blocs. La transaction est alors visible pour le récepteur ainsi que l'ensemble du réseau.

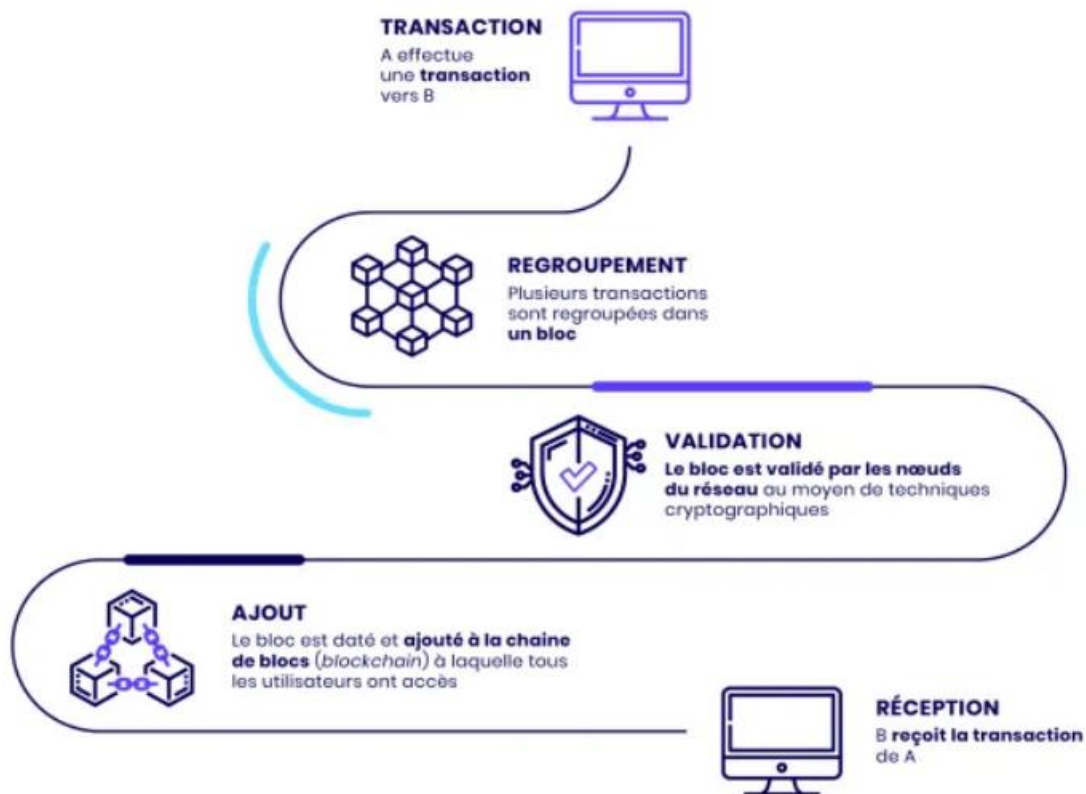


Figure 3: Schéma représentant le cycle d'envoi d'une transaction

1.4.3 Smart contracts c'est quoi ? :

Smart Contract ou "contrat intelligent" est simplement un programme exécuté sur la blockchain d'Ethereum. C'est un ensemble de code (fonctions) et de données (état) qui réside à une adresse spécifique sur la blockchain Ethereum. Il dispose d'un solde et peut envoyer des transactions sur le réseau. Cependant, il n'est pas contrôlé par un utilisateur, mais est plutôt déployé et exécuté comme un programme.

Un contrat intelligent peut définir des règles, comme un contrat normal, et les appliquer automatiquement via le code. Les contrats intelligents ne peuvent pas être supprimés par défaut et les interactions avec eux sont irréversibles.

Chapitre 2 :

Présentation et cahier des charges

2.1. Introduction

Ce chapitre sera réservé à l'étude fonctionnelle de l'application à savoir les différents aspects fonctionnels de notre projet.

On commencera notre chapitre par révéler les différentes motivations de la création de l'application puis on présentera les aspects fonctionnels de l'application, c'est-à-dire le but du projet. Ensuite on passera à la population intéressée par l'application et en terminera par déclarer l'ensemble des aspects fonctionnels de l'application et les exigences du choix de la charte graphique.

2.2. Présentation du projet

Le développement scientifique et informatique a mené à la création d'une nouvelle monnaie électronique pour effectuer des transactions sans nécessité d'une banque centrale, cette monnaie est utilisable au moyen d'un réseau informatique décentralisé. Elle utilise des technologies de cryptographie et associe l'utilisateur aux processus d'émission et de règlement des transactions, elle est connue sous le nom 'cryptomonnaie'.

Cette crypto-monnaie repose sur une blockchain ; un registre distribué qui répertorie l'ensemble des actions du réseau depuis l'origine. Les informations à ajouter sont appelées transactions, et sont groupées dans des blocs. Une transaction peut par exemple être un transfert de crypto-monnaie d'une personne à une autre.

Chaque utilisateur de ces crypto-monnaies doit obligatoirement avoir un portefeuille (wallet) là où il peut suivre l'état de sa monnaie et si sa valeur a augmenté ou non, et il peut effectuer des transactions (buy/sell).

D'où l'idée de l'implémentation d'une application (Mobile) de tracking de la valeur d'un portefeuille, dans cette application l'utilisateur dès qu'il crée son compte, il peut suivre les portefeuilles qu'il a créé, effectuer des transactions, consulter le prix et voir des statistiques à propos de toutes les crypto-monnaies.

2.3. Objectifs et périmètres du projet

Le projet opte à créer une application mobile pour suivre la valeur d'un portefeuille de cryptomonnaie ; tout utilisateur a le droit de :

- Créer un compte personnel puis créer plusieurs portefeuilles avec le même compte utilisateur.
- Effectuer des transactions de vente et d'achat des crypto-monnaies.
- Suivre l'état du portefeuille ; l'augmentation ou la diminution des valeurs du crypto monnaies.
- Suivre le marché des crypto-monnaies et voir des statistiques à propos de chaque crypto-monnaie.

2.4. Public ciblé

L'application concerne la population intéressée par ce domaine de crypto-monnaies; pour qu'ils puissent suivre d'une façon simple leur portefeuilles et avoir des informations à propos du marché des crypto monnaies.

2.5. Contenu de l'application

Après l'étape d'inscription puis l'authentification, l'utilisateur dans son propre espace peut naviguer entre ses portefeuilles, le marché actuel des crypto-monnaies et l'espace où il peut effectuer des transactions.

2.5.1 Étape d'inscription

Tout utilisateur voulant profiter de l'application doit passer par l'étape d'inscription en fournissant certaines informations personnelles.

2.5.2 Étape d'authentification

Tout utilisateur de l'application doit s'identifier via un identifiant (adresse email) et un mot de passe, d'où la nécessité d'une étape d'authentification.

2.5.3 Espace principale

- L'utilisateur peut chercher les prix des crypto-monnaies via le marché actuel.
- L'utilisateur peut suivre l'état du portefeuille ; l'augmentation ou la diminution des valeurs du crypto monnaies.
- L'utilisateur peut effectuer des transactions de vente ou d'achat au moment réel via ses portefeuilles.
- L'utilisateur peut suivre le marché des crypto-monnaies et voir des statistiques à propos de chaque crypto- monnaie :

2.6. Charte graphique : exigences et limites

2.6.1 Objectif de la charte :

La charte graphique est le document de référence dans la construction des applications. Elle présente les principes généraux graphiques, interactifs et ergonomiques applicables aux applications. Elle garantit ainsi une homogénéité de l'identité du Groupe à travers ses activités, ses supports.

2.6.2 Les grands enjeux de l'application:

L'application a été pensée autour de 3 dimensions essentielles :

- Dimension « Home »

Cet espace qui représente l'espace d'accueil pour l'utilisateur il doit être bien organisé, divisé en plusieurs parties, et contient les informations principales à propos du compte. Ces informations sont le nom de l'utilisateur, leur gain total auprès de tous les portefeuilles, les transactions effectuées par chaque cryptomonnaie, et les prix des crypto-monnaies les plus élevés.

- Dimension « Market »

Cet espace est responsable de l’affichage des prix de tous les crypto-monnaies donc il doit être sous forme d’une liste contenant les informations de chaque cryptomonnaie, pour le prix et le nom qui sont les plus importants ils doivent être en gras et en majuscules, pour que l’utilisateur arrive à les voir simplement, et pour le pourcentage il doit être clairement présenté avec une couleur d’arrière-plan qui montre le taux d’évolution de cette cryptomonnaie

Pour l’espace réservé à les transactions il faut qu’il contient 2 boutons pour vendre et acheter les crypto-monnaies avec deux différents couleurs.

- Dimension « Profile »

Cet espace profile doit contenir une liste des paramètres qui permet la modification du compte.

2.6.3 LOGO :

- Règles d’utilisation du logo.



Figure 4 : logo de l’application

Le logo de notre application (figure 4) est affiché au-dessus il contient deux éléments :

- Un texte « cryptowall » qui se compose de deux mots « crypto » comme référence à cryptomonnaie et « wall » comme référence à wallet (portefeuille).
- Une icône qui représente une intersection de la lettre « C » avec une feuille ; la lettre « C » est le début du mot « Cryptomonnaie » et la feuille est une référence aux feuilles d’argents ce qui montre qu’il y a une égalité entre les crypto-monnaies et les feuilles d’argents et ils peuvent jouer le même rôle économique.

- **Couleurs du logo**

Ce Logo contient 3 couleurs :

La couleur du texte : noir, la couleur de la feuille : bleu-ciel, et la couleur de la lettre « C » est en bleu.

2.6.4 Palette de couleurs

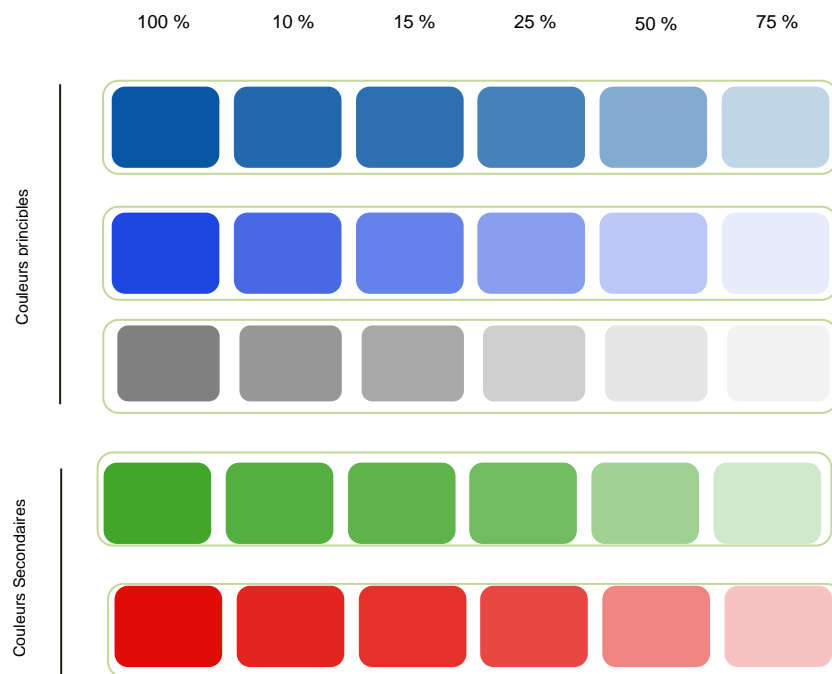


Figure 5 : Palette des couleurs

Utilisation du couleur :

Les couleurs principales seront utilisées pour les arrières plans et les boutons essentiels de l'application (bouton d'inscription, d'authentification, et les boutons pour effectuer des transactions)

Les couleurs secondaires seront utilisées pour les arrières plans des divisions contenant les pourcentages, le vert montre l'augmentation du prix et le rouge montre sa diminution. Le rouge est utilisé aussi pour les actions qui demandent l'attention de l'utilisateur comme la suppression d'un portefeuille ou l'action de la déconnexion.

2.6.5 Typographies :

- **Mulish**

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h i j k l m n o p q r s t u v w x y z
. , : @ [] ; ! / ? % ^ \$ & é » ' - è _ ç à () = * +

2.7. Arborescence de l'application

Parmi les étapes de la conception d'une application quelconque, il y a l'arborescence de l'application, qui est une étape très importante, car elle conditionne les modes d'accès à l'information et la navigation sur l'application, elle est donc importante pour l'ergonomie d'une application mobile.

Le schéma suivant (figure 6) illustre l'arborescence de notre application :

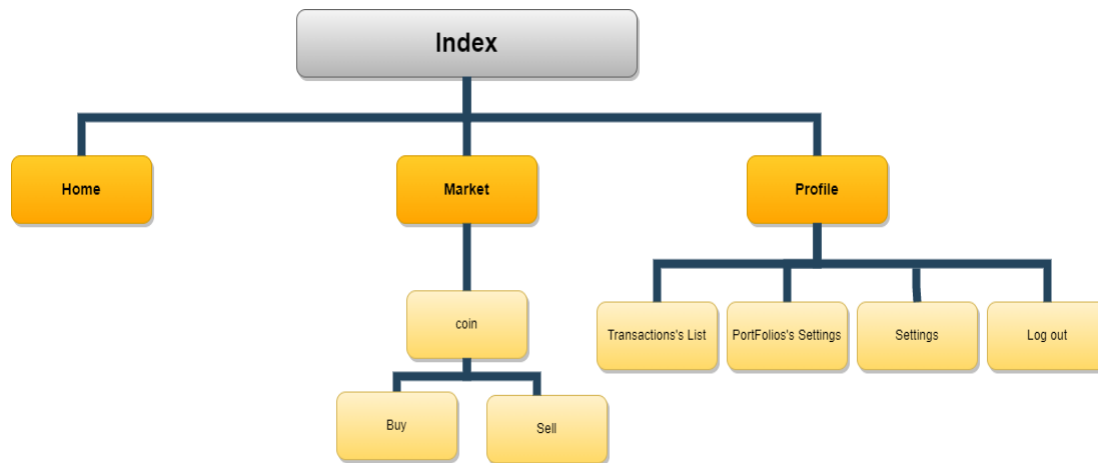


Figure 6 : Arborescence de l'application

Dans l'application il y a trois axes principaux :

- **Home** : Cet axe représente une table de bord contenant les informations essentielles à propos du compte utilisateur et ses actions.
- **Market** : Cet axe contient une liste de tous les crypto-monnaies avec le prix actuel et les informations propres à chacune de ces crypto-monnaies :

Avec une simple clique sur une crypto-monnaie une nouvelle interface s'affiche contenant les informations de cette cryptomonnaie et deux boutons permettant de se diriger vers une autre interface permettant d'effectuer des transactions.
- **Profile**: Cet axe contient l'espace de l'utilisateur là où il peut accéder à la liste de ces transactions et à leurs portefeuilles, aussi il aura la possibilité de modifier ces informations et de se déconnecter de son compte.

2.8. Planification du projet

La planification du projet est l'activité qui consiste à déterminer et à ordonnancer les tâches du projet, à estimer leurs charges et à déterminer les profils nécessaires à leur réalisation . L'objectif principale de la planification (le planning) est suivre et communiquer l'avancement du projet, et déterminer si les objectifs des tâches sont réalisés ou dépassés .

2.8.1 Diagramme de GANTT

Le diagramme de GANTT est la technique et représentation graphique permettant de renseigner et situer dans le temps les tâches et ressources du projet .Voici le diagramme de GANTT que nous avons suivi :

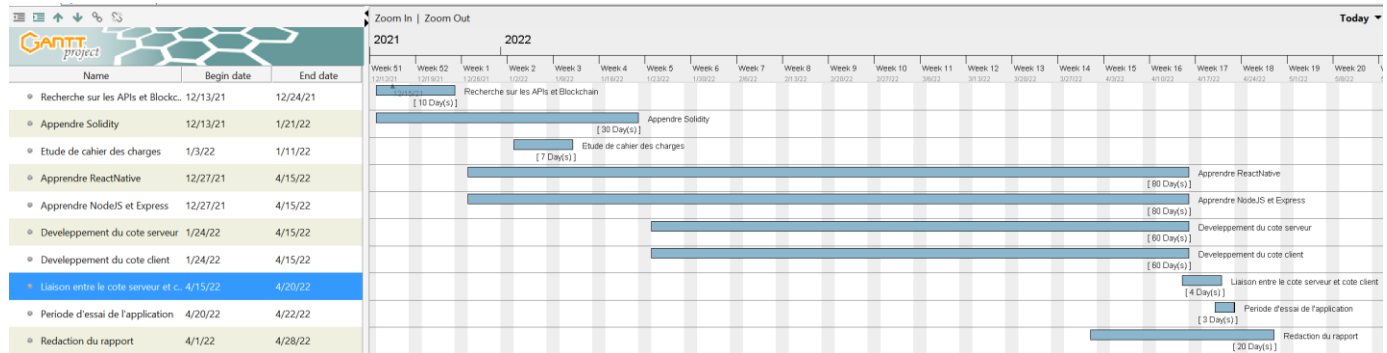


Figure 7 : Diagramme de GANTT

2.8.2 Méthode de PERT

La méthode PERT (Program Evaluation and Review Technique) utilise une représentation en graphe pour déterminer la durée minimum d'un projet, connaissant la durée de chaque tâche et les contraintes d'enchaînement.

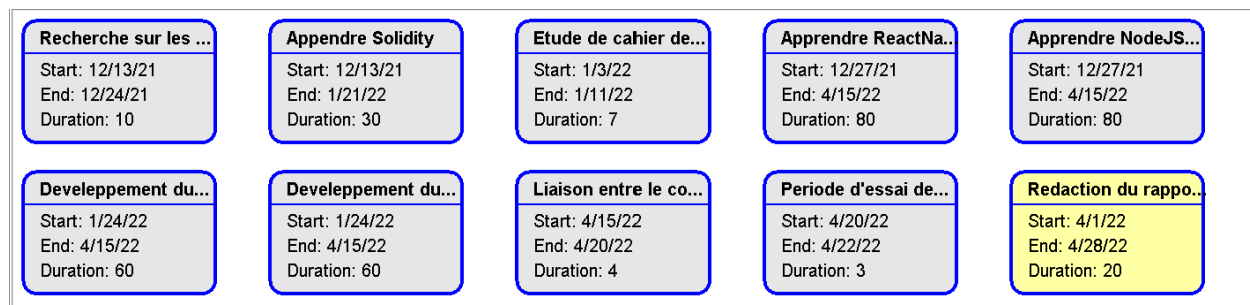


Figure 8 : Diagramme de PERT

Chapitre 3 :

Technologies et outils utilises

3.1. Introduction

Dans ce chapitre on abordera plutôt l'aspect technique implémenté par notre application, à l'inverse du deuxième chapitre consacré à l'aspect fonctionnel; on commencera par présenter notre choix de technologies, pourquoi ? Et pour quel but ? Puis on présentera l'ensemble des technologies et outils utilisés depuis la conception jusqu'à la réalisation.

3.1. Technologies implémentées

3.2.1.Introduction :

Les avantages et la grande communauté de JavaScript nous amènent à choisir ce langage dans les deux parties côté client et côté serveur ; dans le côté client nous avons utilisé React native qui donne la possibilité que le développement soit hybride et dans le côté serveur on a utilisé le Framework express Js en se basant dans les deux parties sur NodeJS.

Dans la partie suivante une présentation des technologies ainsi que les langages utilisés :

3.2.2.Javascript :



Figure 9 : Logo de JavaScript

JavaScript est un langage de programmation dynamique utilisé pour le développement Web, dans les applications Web, pour le développement des jeux et bien plus encore il permet d'implémenter des fonctionnalités dynamiques sur des pages Web qui ne peuvent pas être réalisées uniquement avec HTML et CSS.

3.2.3. NodeJS :



Figure 10 : Logo de NodeJS

Node.js est un environnement d'exécution asynchrone, fonctionne sur différentes plateformes open source et multiplateforme permettant d'exécuter du code JavaScript en dehors d'un navigateur...

✓ Points forts de node js :

- Facile à démarrer et peut être utilisé pour le prototypage et le développement agile.
- Fournit des services rapides et hautement évolutifs.
- Utilise JavaScript partout, il est donc facile de créer des services back-end à l'aide de Node.js
- Code source plus propre et cohérent.
- Grand écosystème pour bibliothèque open source.

3.2.4. Express Js :



Figure 11 : Logo d'Express

Express est un framework Web rapide, affirmé, essentiel et modéré de Node.js , fournit un ensemble robuste de fonctionnalités pour développer des applications Web et mobiles.

✓ Fonctionnalités de Express:

- Il peut être utilisé pour concevoir des applications Web monopage, multipage et hybrides.
- Il permet de configurer des middlewares pour répondre aux requêtes HTTP.
- Il définit une table de routage qui est utilisée pour effectuer différentes actions basées sur la méthode HTTP et l'URL.
- Il permet de restituer dynamiquement des pages HTML en passant des arguments aux modèles.

3.2.5. MySQL :



Figure 12 : Logo de MySQL

MySQL est un système de gestion de bases de données relationnelles (RDBMS) largement utilisé, gratuit et open-source, idéal pour les petites et les grandes applications, distribué sous double licence, une licence publique générale GNU et une propriétaire selon l'utilisation qui en est faite. La première version de MySQL est apparue en 1995 et l'outil est régulièrement entretenu.

3.2.6. React native :

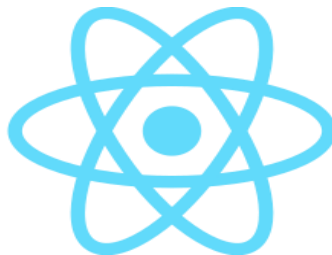


Figure 13 : Logo de React Native

React Native est un framework d'applications mobiles, open source, créé par Facebook, utilisé pour développer des applications mobiles pour Android, iOS et UWP en permettant aux développeurs d'utiliser React avec les fonctionnalités natives de ces plateformes.

3.2.7.Expo :



Figure 14 : Logo d'Expo

Expo, c'est à la fois un framework et une plateforme qui simplifient la création et le déploiement d'applications mobiles avec React Native. Expo embarque de nombreux outils utiles et des bibliothèques natives pour React Native. Il gère aussi la mise à jour de ces bibliothèques. C'est donc un moyen de démarrer facilement et rapidement son projet.

3.2. Outils d'implémentation

3.3.1. Visual Studio Code :



Figure 15 : Logo de Visual Studio Code

Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS. Les fonctionnalités incluent la prise en charge du débogage, la mise en évidence de la syntaxe, la complétion intelligente du code, les snippets, la refactorisation du code et Git intégré.

3.3.2. WampServer :



Figure 16 : Logo de WampServer

WampServer est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs, un interpréteur de script, ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

3.3.3. Power AMC :



Figure 17 : Logo de PowerAMC

PowerAMC est un logiciel de modélisation. Il permet de modéliser les traitements informatiques et leurs bases de données associées.

Créé par SDP sous le nom AMC*Designor, racheté par Power soft, ce logiciel est produit par Sybase depuis le rachat par cet éditeur en 1995. Hors de France, la version internationale est commercialisée par Sybase sous la marque Power Designer.

3.3.4. Postman



Figure 18 : Logo de Postman

Postman est une application permettant de tester des API, créée en 2012 par Abhinav Asthana, Ankit Sobti et Abhijit Kane² à Bangalore pour répondre à une problématique de test d'API partageable. D'abord module complémentaire de Google Chrome, puis client lourd, et finalement client léger, elle est à présent utilisée par plus de 500 000 entreprises dans le monde et a son siège à San Francisco

3.3.5. Figma



Figure 19 : Logo de Figma

Figma est un éditeur de graphiques vectoriels et un outil de prototypage. Il est principalement basé sur le web, avec des fonctionnalités hors ligne supplémentaires activées par des applications de bureau pour macOS et Windows. Les Figma Mirror companion apps pour Android et iOS permettent de visualiser des prototypes Figma sur des appareils mobiles. L'ensemble des fonctionnalités de Figma est axé sur l'utilisation dans la conception de l'interface utilisateur et de l'expérience utilisateur, en mettant l'accent sur la collaboration en temps réel.

3.3. Problématiques et solutions technologiques :

- Les limites du protocole HTTP:

Les websockets sont une solution pour les limites du protocole HTTP. Une WebSocket est une spécification d'un protocole définie dans la RFC 6455, publiée en décembre 2011 permettant l'échange de données entre un client et un serveur d'une manière asynchrone, bidirectionnelle en mode full duplex utilisant une connection TCP.

Une websocket est efficace car elle require moins de bande passante , elle ne require pas d'entete dans chaque message, la latence est reduite , et elle permet de recevoir des donnees en temps reel. Ce dernier c'était l'un des problematiques de notre sujet ; car l'obtention du prix actuel des cryptomonnaies avec le protocole HTTP c'était impossible a cause des limites du protocole HTTP d'où l'idée de l'utilisation de cette technologie.

Les étapes d'utilisation pour la mise en oeuvre d'une websocket sont :

- établir une connexion.
- envoyer des messages côté client et serveur (bi-directionnel) indépendamment les uns des autres (full duplex).
- fin de la connexion.

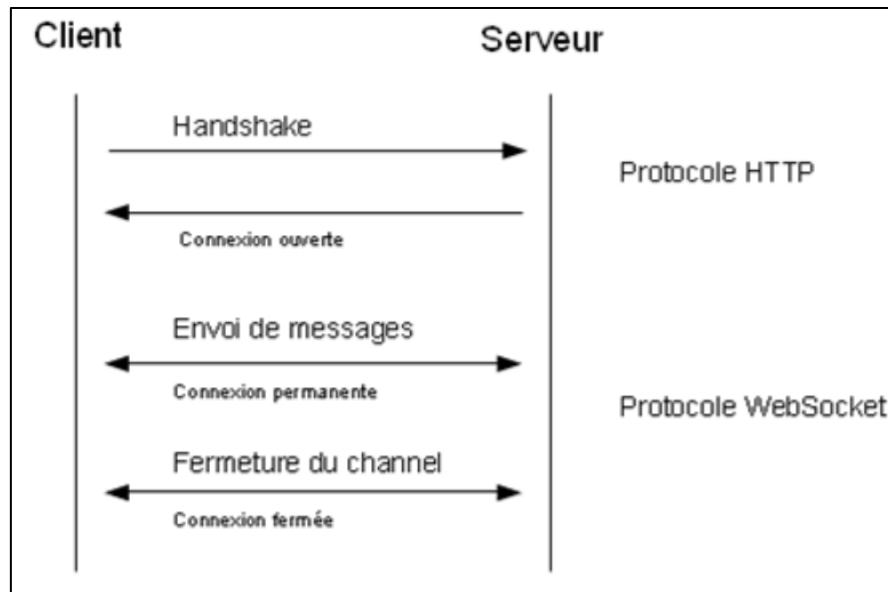


Figure 20 : Mise en œuvre d'une web socket

Le protocole HTTP n'est utilisé que pour établir la connexion d'une WebSocket : une fois la connexion établie le protocole HTTP n'est plus utilisé au profit du protocole WebSocket.

- Maintenir L'application sans arrêt :

PM2 est un gestionnaire de processus de production pour les applications Node.js avec un équilibreur de charge intégré. Il permet de maintenir les applications en vie pour toujours, de les recharger sans temps d'arrêt et de faciliter les tâches d'administration système courantes.

- Partager le serveur localhost :

Ngrok est un proxy inverse qui crée un tunnel sécurisé entre un point de terminaison public et un service Web exécuté localement. ngrok capture et analyse tout le trafic sur le tunnel pour une inspection et une relecture ultérieures.

Chapitre 4 :

Architecture et Conception

4.1. Introduction

Dans ce chapitre, on discutera la conception de l'application, on parlera d'abord de l'architecture de l'application; ensuite de l'UML qui a été la base essentielle de notre travail puisqu'il nous a donné la chance de se servir de deux diagrammes : diagramme de cas d'utilisation et diagrammes de classes.

4.2. Architecture de l'application

L'architecture 3-tiers (de l'anglais tiers signifiant étages ou niveaux) est un modèle logique d'architecture applicative qui vise à séparer très nettement trois couches logicielles au sein d'une même application ou système, à modéliser et présenter cette application comme un empilement de trois couches, étages, niveaux ou strates dont le rôle est clairement défini :

- La présentation des données : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur.
- le traitement métier des données : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative.
- l'accès aux données persistantes : correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive.

Les trois couches de l'architecture 3-tiers sont :

Couche présentation (premier niveau) :

Elle correspond à la partie de l'application visible et interactive avec les utilisateurs. On parle d'Interface Homme Machine.

En informatique, elle peut être réalisée par une application graphique ou textuelle. Elle peut aussi être représentée en HTML pour être exploitée par un navigateur web ou en WML pour être utilisée par un téléphone portable.

Couche métier / business (deuxième niveau) :

Elle correspond à la partie fonctionnelle de l'application, celle qui implémente la « logique », et qui décrit les opérations que l'application opère sur les données en fonction des requêtes des utilisateurs, effectuées au travers de la couche présentation. Les différentes règles de gestion et de contrôle du système sont mises en œuvre dans cette couche.

La couche métier offre des services applicatifs et métier à la couche présentation. Pour fournir ces services, elle s'appuie, le cas échéant, sur les données du système, accessibles au travers des services de la couche inférieure. En retour, elle renvoie à la couche présentation les résultats qu'elle a calculés.

Couche accès aux données (troisième niveau) :

Elle consiste en la partie gérant l'accès aux gisements de données du système. Ces données peuvent être propres au système, ou gérées par un autre système. La couche métier n'a pas à s'adapter à ces deux cas, ils sont transparents pour elle, et elle accède aux données de manière uniforme.

Voici un schéma représentant l'architecture de notre application :

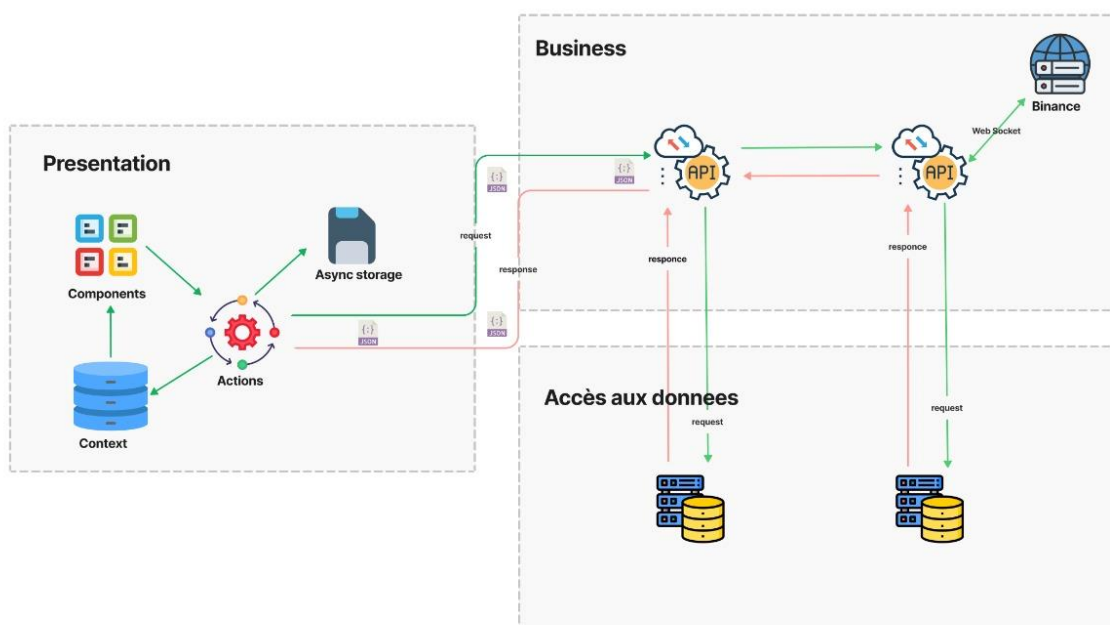


Figure 21 : schéma de l'application

Dans notre projet, pour la couche présentation elle est présentée sous un projet front end réalisé avec React Native , et pour la couche business nous avons utilisé Express JS.

4.3. Conception de l'application

Afin de concevoir notre application, nous avons opté pour UML comme langage de conception et de modélisation.

4.3.1 Pourquoi UML ?

UML est apparu au cours des années 90 et tient sa force de la fusion des méthodes de BOOCH, de la méthode OMT et OOSE qui préexistaient et dont il a su tirer le meilleur. En pratique, il s'agit, comme pour tout formalisme, d'un ensemble de diagrammes permettant dès lors que l'on en comprend la signification, de décrire simplement un ensemble de données, d'actions ou de traitements, permettant ainsi la création d'une analyse complète pour réaliser un projet informatique (une application mobile dans notre cas).

UML :

- Un langage de Modélisation (standard) ;
- basé sur une approche solide Objectory ;
- un support important de stratégies et d'heuristiques ;
- adapté au développement Objet ;
- adapté au développement Itératif ;
- adaptable au processus d'une équipe ;
- supporté par des outils.

UML & POO :

- UML utilise une approche objet des logiciels ;
- les entités métier sont identifiées comme des objets ;
- les modules opérationnels sont aussi vus comme des objets ;
- la construction se fait avec une approche Yoyo (Top-down, Bottom-Up).

Adaptable :

- UML n'impose pas de méthode de développement. C'est un standard de communication contenant un certain nombre d'artefacts utiles ;
- il est donc possible d'intégrer l'utilisation d'UML à un processus de développement existant.

Pour toutes ces raisons nous avons choisi UML comme langage de conception lors de la réalisation de notre projet de fin d'étude.

La norme UML actuelle comporte 13 diagrammes. Cependant seuls 2 diagrammes nous seront réellement utiles : Le diagramme des cas d'Utilisation, et le diagramme de classes.

Le diagramme de classes est le plus utile, c'est un formalisme pour représenter les concepts usuels de l'orienté objet. Le diagramme des cas d'utilisation permet de décrire les besoins de l'utilisateur final de l'application.

4.3.2 Règles de gestion

- Un utilisateur peut avoir un seul compte.
- Un compte peut avoir plusieurs portefeuilles.
- Un portefeuille appartient à un seul compte.
- Pour chaque portefeuille l'utilisateur peut effectuer plusieurs transactions.
- Une transaction se fait pour une seule cryptomonnaie.
- Une crypto monnaie peut avoir plusieurs prix.
- La transaction peut être sous forme de deux types : vent ou achat.

4.3.3 Diagramme cas d'utilisation

Les cas d'utilisation (use cases) décrivent sous forme d'actions et de réactions, le comportement d'un système du point de vue d'un utilisateur. Ils sont utiles lors de l'élaboration du cahier des charges ou du document de spécifications des besoins du logiciel. Ce diagramme comprend les acteurs, le système et les cas d'utilisation.

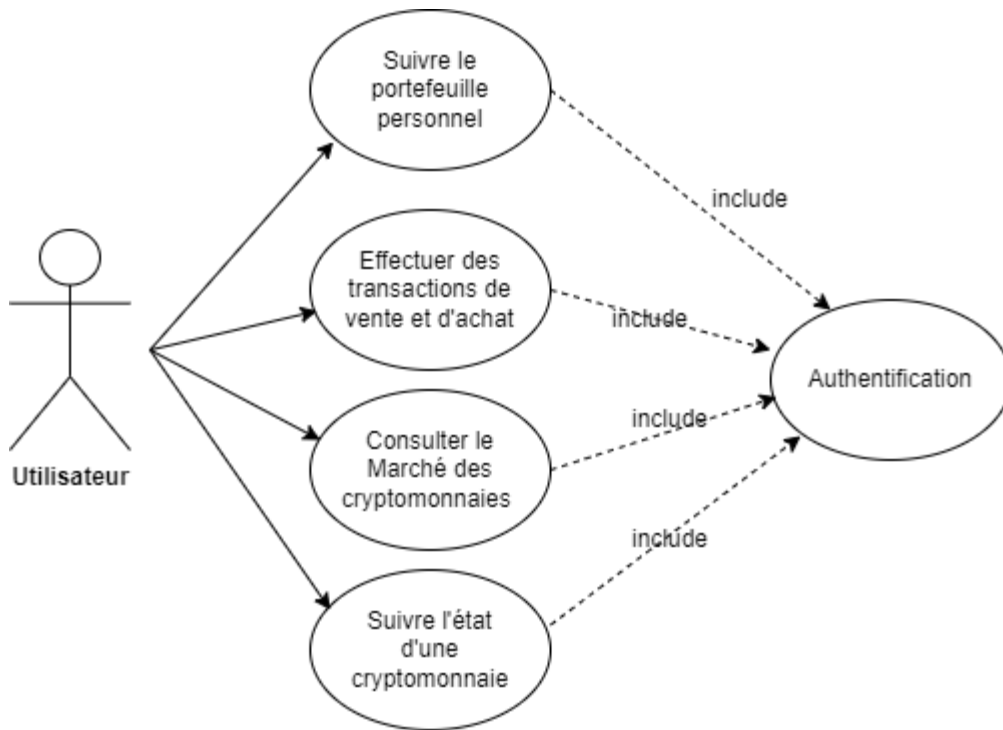


Figure 22 : Diagramme cas d'utilisation

Ce diagramme ci-dessus montre la totalité des tâches principales pour l'utilisateur de l'application il peut consulter le marché des cryptomonnaies , suivre son portefeuille, effectuer des transactions de vent ou d'achat, et suivre l'état et les statistiques des cryptomonnaies et tous ces tâches sont procédées obligatoirement par une authentification.

4.3.4 Diagramme de classes

Le diagramme de classes d'UML saisit la structure statique d'un système en montrant les objets dans le système, les relations entre les objets, les attributs et les opérations qui caractérisent chaque classe d'objets. C'est le plus important des modèles d'UML.

Il est totalement indispensable quel que soit la méthode utilisée, généralement le diagramme de classes défini la structure interne et les liens entre les différentes classes du système mis en œuvre.

Nous examinerons plus en détails ce diagramme dans le schéma suivant :

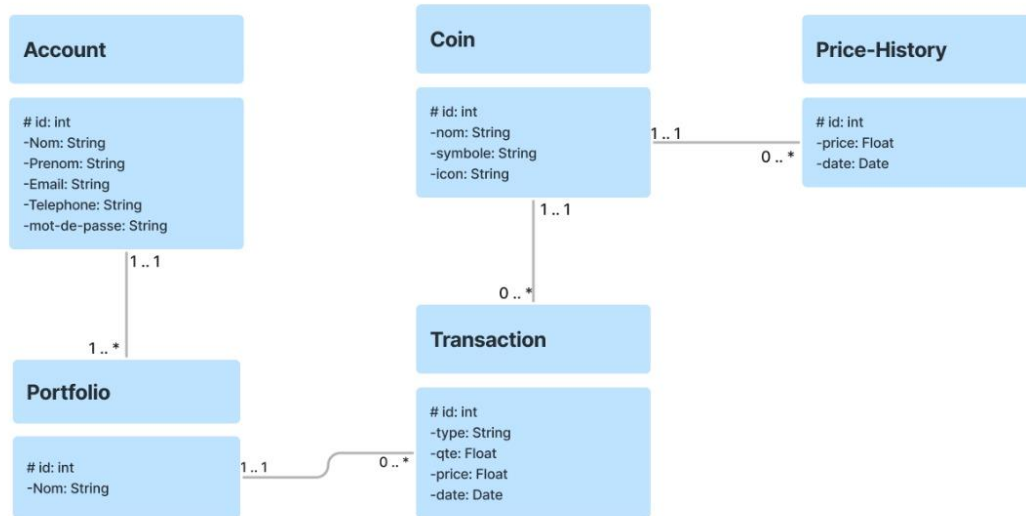


Figure 23 : Diagramme de classes

En ce qui concerne notre projet, nous avons créé 5 classes, listées ci suit :

- **Classe Account** : cette classe réservée pour le compte utilisateur qui est obligatoire pour utiliser l'application, elle contient les attributs suivants : le nom, le prénom, l'email, le téléphone, le mot de passe, et un id pour identifier l'utilisateur, lors de l'inscription les données nécessaires sont ces attributs et ils sont enregistrés dans une table qui s'appelle « Account ».
- **Classe Portfolio** : Cette classe contient comme attributs : l'id pour identifier le portefeuille et le nom du portefeuille, lorsqu' on transforme le diagramme de classes en modèle relationnel cette classe se transforme en une table qui contient une clé étrangère de la table « Account ».
- **Classe Transaction** : cette classe contient les attributs suivants : le nom, le type de la transaction : buy ou sell, la date de la transaction, le prix et la quantité. Lors

de la transformation de ce diagramme en modèle relationnel il va se traduit en une table qui contient deux clés étrangères la première vers la table « portfolio » et l'autre vers « coin ».

- Classe **Coin** : cette classe contient les attributs suivants : l'id, le nom, le symbole, et une icône.
- Classe **Price-history** : cette classe contient les attributs suivants : l'id, le prix et la date, lors de la transformation de cette ce diagramme en modèle relationnel il va se traduit en une table qui contient ces attributs et une clé étrangère vers la table « coin ».

Chapitre 5 :

Présentation de l'application

5.1. Introduction

Ce chapitre vise l'étude et la description détaillée de notre application, celui là est divisé en cinq espaces: inscription, authentification, home, market, et Profile. Pour mieux éclaircir les choses, cette présentation sera accompagnée avec une suite de captures d'écran.

5.1. Etape d'authentification

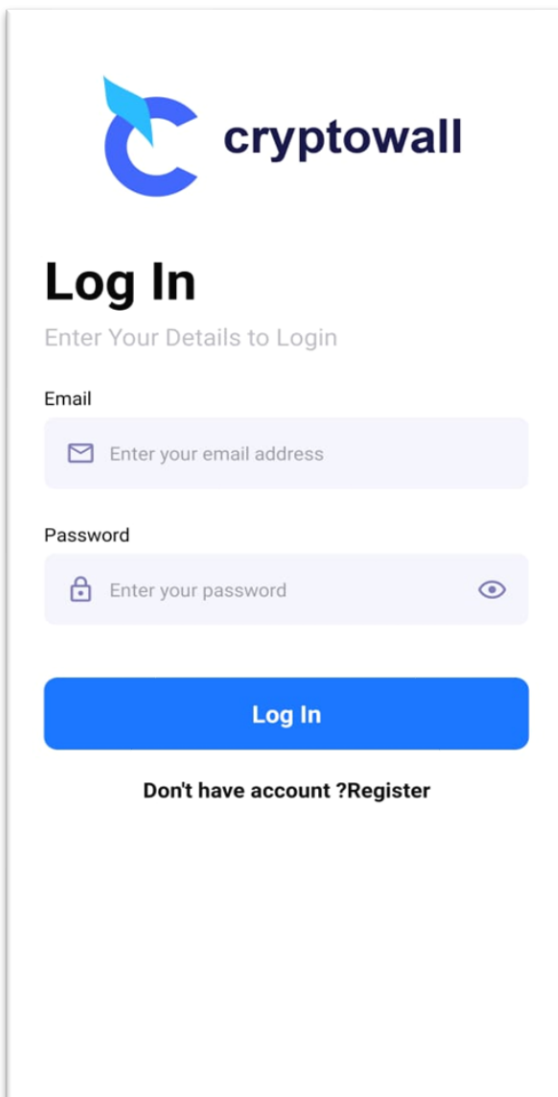
The image shows a mobile application login screen for 'cryptowall'. At the top is the logo, a blue stylized 'C' with a leaf-like shape, followed by the text 'cryptowall'. Below the logo is the title 'Log In' in a large, bold, black font. Underneath the title is the subtitle 'Enter Your Details to Login' in a smaller, grey font. There are two input fields: the first is labeled 'Email' and contains a placeholder text 'Enter your email address' with an envelope icon; the second is labeled 'Password' and contains a placeholder text 'Enter your password' with a lock icon and a toggle eye icon. Below these fields is a large blue button with the text 'Log In' in white. At the bottom, there is a link that says 'Don't have account ?Register'.

Figure 24 : Etape d'authentification

Cette étape d'authentification (figure 24) est obligatoire pour les utilisateurs ayant un compte pour qu'ils puissent utiliser l'application .

Ils doivent s'authentifier en utilisant l'adresse mail et le mot de passe convenable a cet adresse email .

Si l'une des informations est incorrecte une erreur est afficher et l'utilisateur doit saisir ces données correctement pour qu'il puisse s'authentifier.

Si l'utilisateur n'a aucun compte utilisateur il doit s'inscrire en cliquant sur « Register » , une interface d'inscription s'affiche avec un formulaire à remplir pour créer un compte utilisateur.

Cette interface (figure 24) se compose de deux parties ; une contient le logo de l'application et l'autre contient le fomulaire de l'authentification.

5.2. Etape d'inscription

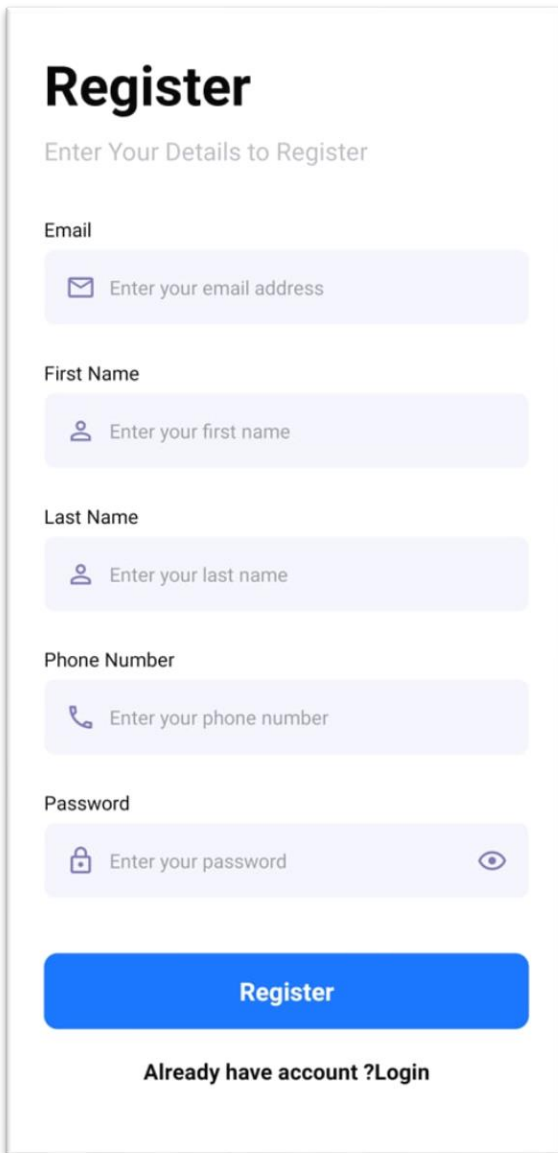
The image shows a 'Register' form with a title 'Register' and a subtitle 'Enter Your Details to Register'. It contains five input fields: 'Email' with an envelope icon, 'First Name' with a person icon, 'Last Name' with a person icon, 'Phone Number' with a phone icon, and 'Password' with a lock icon and a toggle eye icon. Below the fields is a blue 'Register' button and a link 'Already have account ?Login'.

Figure 25 : Etape d'inscription

L'étape de l'inscription est importante pour utiliser l'application ; chaque utilisateur doit remplir un formulaire (figure 25) qui contient : le nom, le prénom, l'adresse email, le numéro de téléphone, et un mot de passe afin qu'il puisse s'identifier prochainement avec ce mot de passe lors de l'étape de l'authentification .

Après que l'utilisateur va remplir ces champs et cliquer sur le bouton « Register » d'inscription un compte va être créer pour cet utilisateur et il peut utiliser l'application juste avec l'adresse email et le mot de passe.

Si l'adresse email choisie par l'utilisateur lors de l'inscription est déjà existante pour un autre compte,l'utilisateur doit remplir ce champs de nouveau par une autre adresse email.

Si déjà l'utilisateur avait un compte il peut cliquer sur le bouton « Login » directement et s'authentifier avec leur adresse email et leur mot de passe.

Après cette étape d'inscription l'utilisateur peut utiliser l'application facilement il peut créer le nombre qu'il veut des portefeuilles avec ce compte qu'il a créé et effectuer des transactions d'achats ou de ventes avec les crypto-monnaies existantes sur le marché.

5.3. Profile

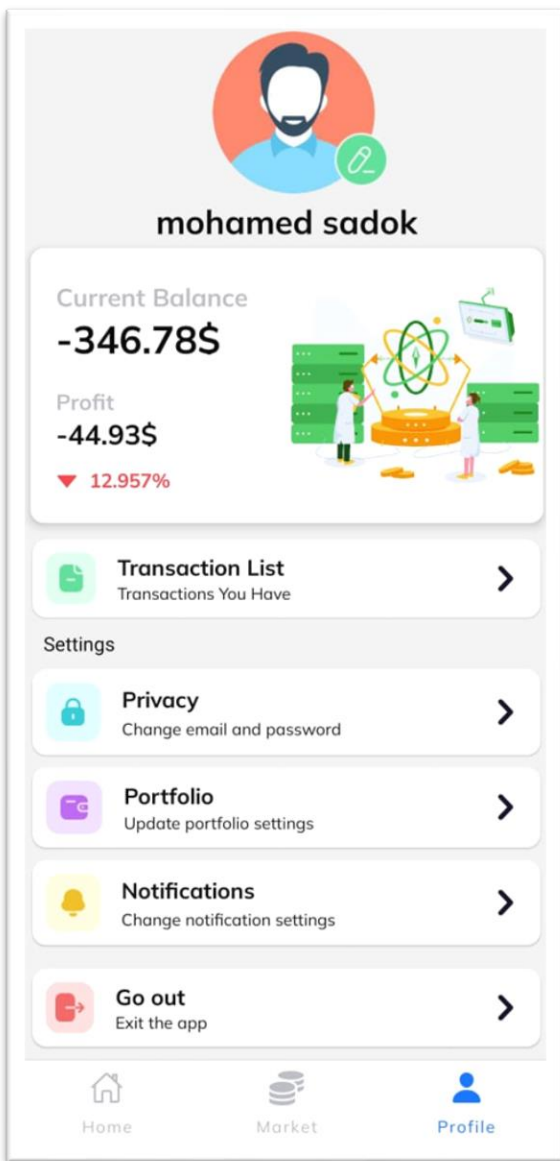


Figure 26 : Interface du Profile

Cette interface (figure 26) présente l'espace profile de l'utilisateur où il peut modifier les informations de ces portefeuilles et voir la liste des transactions propres à chaque portefeuille.

Nous pouvons diviser cette interface en 2 parties ; la première qui est en haut contient le nom complet de l'utilisateur, le gain qui a eu d'après ces portefeuilles et un lien vers la liste des transactions propres au portefeuille choisi, au-dessous de cette partie se trouve une liste des paramètres pour modifier les informations de l'utilisateur ou des portefeuilles et en bas il y a un bouton pour la déconnexion.

Lorsqu'on clique sur la liste des transactions une nouvelle interface s'affiche contenant les transactions effectuées propres au portefeuille choisi par défaut.

Chaque clique sur « Portfolio » implique l'apparition d'une interface qui permet l'affichage des portefeuilles de ce compte.

En bas il y a un bouton pour la déconnexion permet à l'utilisateur de se déconnecter de son compte utilisateur.

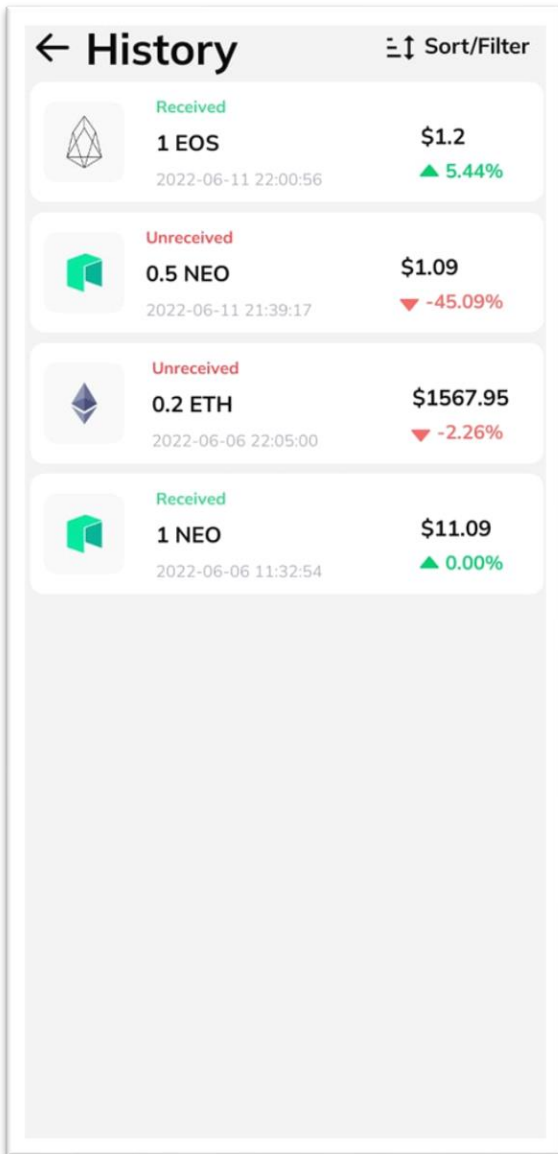


Figure 27 : Interface de la liste des transactions

Après qu'on choisit un portefeuille comme nous avons dit, l'id de ce portefeuille s'enregistre globalement, et après une clique sur le bouton « Transaction List » de l'interface profil (figure 26) s'affiche cette interface (figure 27) qui contient la liste de toutes les transactions d'achat ou de ventes effectuées pour ce portefeuille.

Chaque élément de cette liste représente une transaction ; il contient l'icône et la quantité de la cryptomonnaie, la date, et le prix de cette transaction, ainsi qu'un pourcentage qui montre le pourcentage de gain ou de perte après cette transaction.

Cette liste contient tous les transactions effectuée même s'ils contiennent la même cryptomonnaie.

Si on veut classer les transactions on doit cliquer sur « sort/filter » se trouvant en haut à gauche.

Et pour retourner à la page précédente il faut cliquer sur la flèche qui se trouve en haut à gauche.

Lors de l'ajout d'une transaction elle sera ajoutée dans la base de données et pour qu'elle soit visible dans cette liste présentée dans cette interface il faut l'actualiser en faisant un glissement de haut vers le bas et comme cela la transaction sera ajoutée à la fin de la liste et si on veut qu'elle soit classée il faut refaire le classement de toutes les transactions.

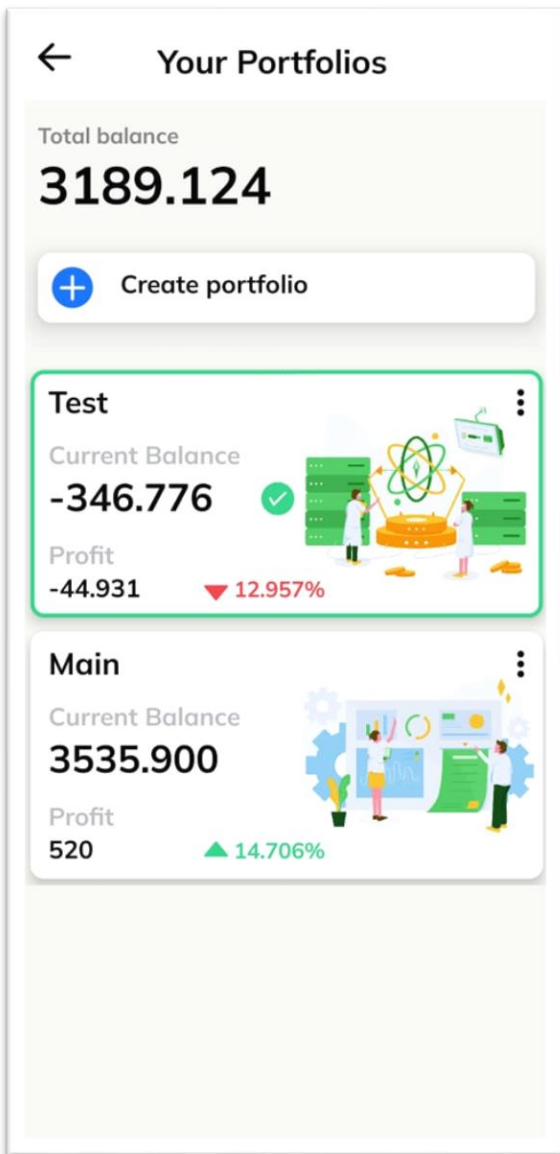


Figure 28 : interface de la liste des portefeuilles

Lorsqu'on clique sur le bouton « Portfolio » de la liste se trouvant dans l'interface de « Profile » (figure 26), il y a l'apparition de cette interface (figure 28) qui représente la liste de tous les portefeuilles créés avec le compte utilisateur.

Chaque portefeuille s'affiche comme un élément de la liste avec la somme de gain de ce portefeuille.

A droite de chaque élément il y a une icône de trois points, par chaque clique il y a l'apparition d'une division permettant la possibilité de la modification et la suppression d'un portefeuille, ainsi qu'une image pour différencier les portefeuilles.

En haut se trouve un bouton « create portfolio » pour la création d'un nouveau portefeuille.

Lorsqu'on clique sur ce bouton un formulaire va être affiché avec les champs nécessaires pour créer un nouveau portefeuille relatif à ce compte utilisateur.

Pour différencier les portefeuilles il est mieux de créer chaque portefeuille avec un nom différent au lieu de choisir le même nom pour deux portefeuilles au plus.

Lorsqu'on clique sur un portefeuille l'id de ce portefeuille s'enregistra globalement ; si on effectue des transactions ils vont être affectés à ce portefeuille, même dans la partie qui représente la liste des transactions, seules les transactions de ce portefeuille qui vont être listées, et il sera entouré par une ligne verte pour montrer qu'il est le portefeuille par défaut.

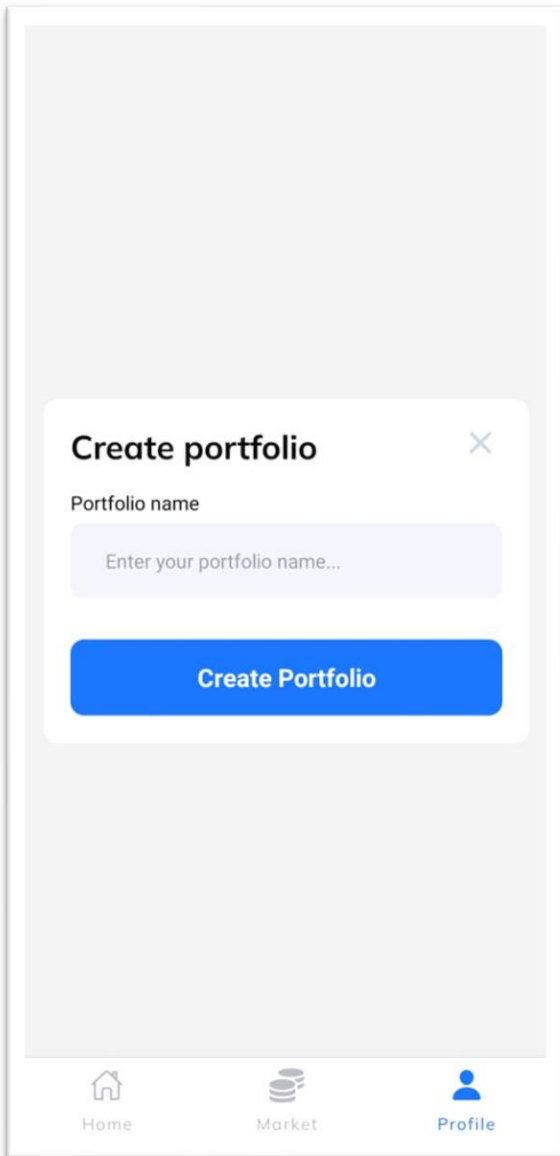


Figure 29 : Modal pour la création d'un portefeuille

Au cours de l'étape de la création d'un portefeuille cette interface s'affiche (figure 29) contient un formulaire contenant un seul champ pour le nom du portefeuille et un bouton pour soumettre le formulaire

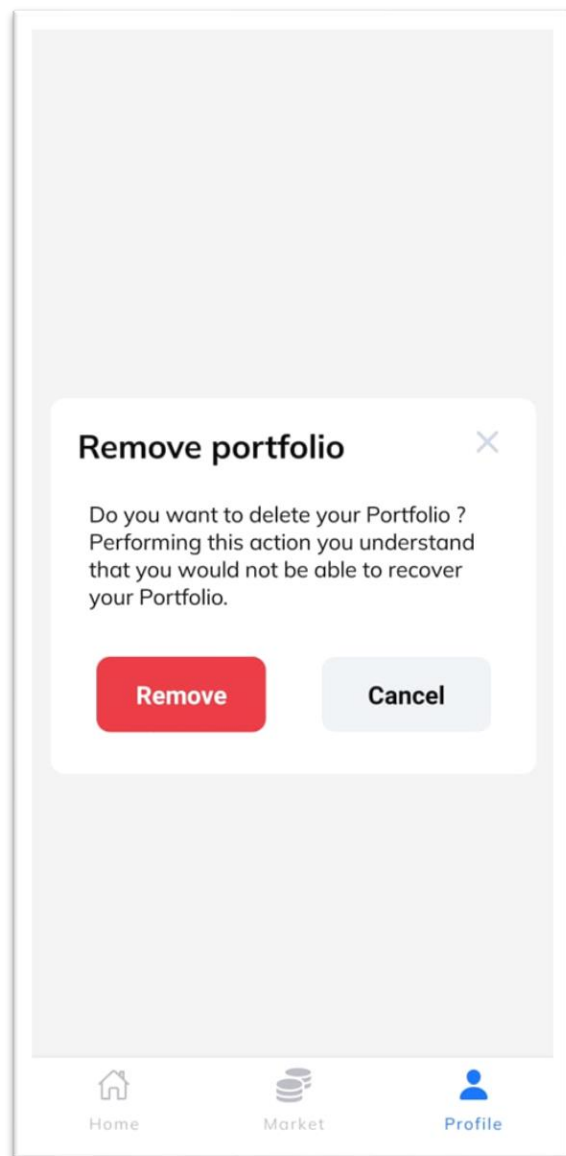


Figure 30 : Modal pour confirmer la suppression d'un portefeuille

Après qu'on clique sur le bouton de la suppression un modal (figure 30) s'affiche contient un bouton pour confirmer la suppression et un autre pour l'annuler.

5.4. Market

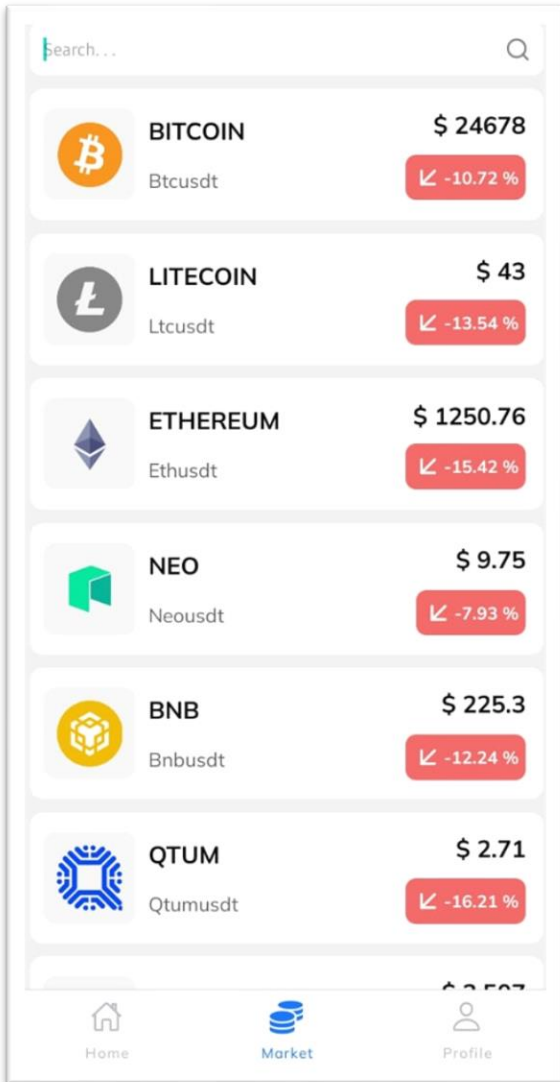


Figure 31 : La liste des transactions

Cette interface (figure 31) présente le marché des crypto-monnaies ; elle contient une liste contenant la majorité des crypto-monnaies existantes publiques et privées avec leurs prix en temps réel en dollar.

Chaque changement en temps réel d'une cryptomonnaie implique le changement du prix visualisé dans cette liste.

Pour chaque élément de cette liste qui représente une cryptomonnaie il y a une icône à gauche de la cryptomonnaie, au milieu il y a le nom de cette cryptomonnaie et à droite il y a le prix actuel en dollar et un pourcentage qui est le taux de changement du prix de cette cryptomonnaie par rapport au prix d'avant 24h à ce moment s'il y a une augmentation du prix la couleur de la division contenant le pourcentage s'affiche en vert sinon elle s'affiche en rouge.

En haut de cette interface il y a une zone de recherche ; si l'utilisateur veut chercher le prix d'une cryptomonnaie il peut simplement taper le nom dans la zone de recherche et les informations de cette cryptomonnaie vont être affichées avec le prix en temps réel.

Si l'utilisateur veut savoir des informations détaillées à propos d'une cryptomonnaie, il peut simplement cliquer sur cette cryptomonnaie dans la liste globale cela implique l'affichage d'une nouvelle interface propre à cette cryptomonnaie comme affiche dans la figure suivante :

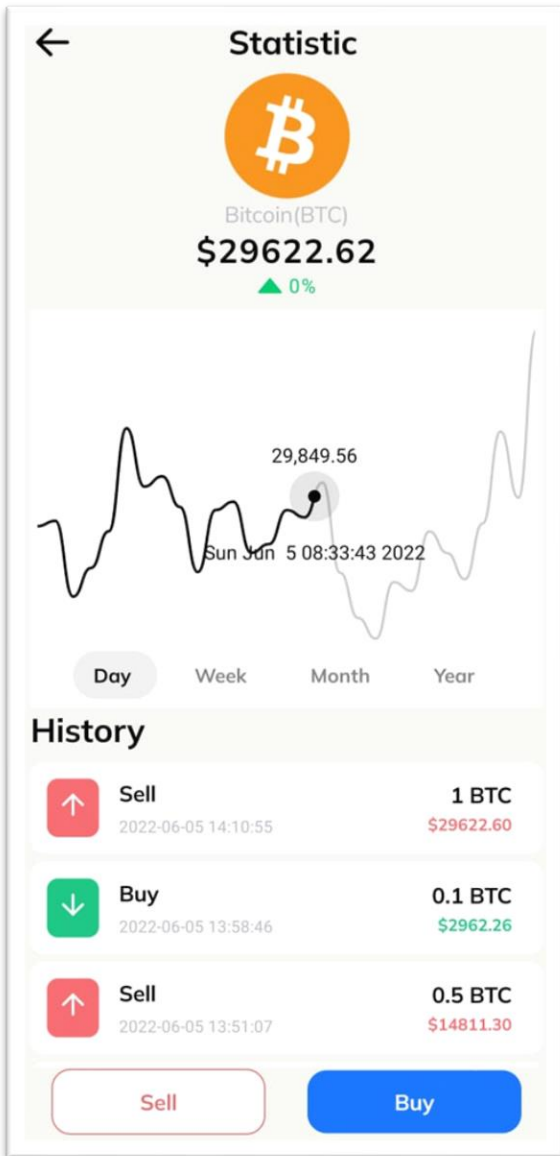


Figure 32 : Statistiques d'une cryptomonnaie

Cette interface (figure 32) représente les informations propres au cryptomonnaie choisie ; le prix, le pourcentage d'évolution, et les transactions effectuées pour cette cryptomonnaie, ...

Cette interface (figure 33) se compose de 4 rubriques :

- La première contient les mêmes informations qui se trouvent dans le market ; le nom du cryptomonnaie, son prix actuel, et le pourcentage d'évolution :
- La deuxième contient un graphe présentant les statistiques d'évolution de cette cryptomonnaie pendant une durée déterminée cette durée peut être un jour, une semaine, un mois ou une année, on peut déterminé cette durée en cliquant sur les boutons qui se trouvent au-dessous du graphe.

Le bouton « Day » est consacré pour une durée d'une journée avant, « Week » est consacré pour une durée d'une semaine avant, « Month » est consacré pour une durée d'un mois avant, « Year » est consacré pour une durée d'une année avant.

- La troisième présente la liste des transactions effectuées pour cette cryptomonnaie pour le portefeuille par défaut. La différence entre cette liste et la liste des transactions qui se trouve dans le profile c'est que cette liste contient seulement les transactions de cette cryptomonnaie choisie par contre l'autre liste contient les transactions de tous les cryptomonnaies du même portefeuille.

- En bas la quatrième rubrique contient deux boutons pour effectuer des transactions le premier pour vendre une quantité de cette cryptomonnaie et le second pour acheter une quantité.

Lorsqu'on clique sur l'un des boutons se trouvant en bas de la figure 32 s'affiche cette interface (figure 33) pour effectuer une transaction d'achat ou de vente selon sur quel bouton qu'on a cliqué.

Le formulaire (figure 33) contient 3 champs ; la quantité, le prix de la cryptomonnaie ce prix est déterminé d'après le prix actuel de cette cryptomonnaie mais on peut le changer en remplissant ce champ avec un autre prix, et le troisième champ est calculé depuis les deux derniers champs car il représente le prix total de cette transaction.

Après qu'on effectue une transaction elle sera enregistrée avec la date de sa création, son type, le prix total, Et elle va être affichée dans la troisième rubrique convenable à la liste des transactions pour cette cryptomonnaie (figure 32) et dans la liste de tous les transactions effectuées (figure 27) pour le portefeuille par défaut.

The screenshot shows a mobile application interface for Bitcoin transactions. At the top, there is a header labeled "Statistic" with a Bitcoin icon. A modal form is displayed in the center, containing three input fields: "Quantity" (0.00), "Price Per Coin" (39804.72), and "Total Spent" (0.00\$). Below the fields is a blue button labeled "Add Transaction". The background shows a list of transactions with "Sell" and "Buy" buttons at the bottom.

Figure 33 : Formulaire pour effectuer une transaction

Pour chaque portefeuille il est calculé le profit gagné ou perdu auprès des transactions effectuées pour ce dernier, ce profit est afficher dans la liste des portefeuilles et dans l'interface du profile.

Chaque transaction effectuée va être ajoutée à « Assets » qui regroupe l'ensemble des transactions propres à la même cryptomonnaie avec le gain ou la perte d'après les transactions.

5.5. Home

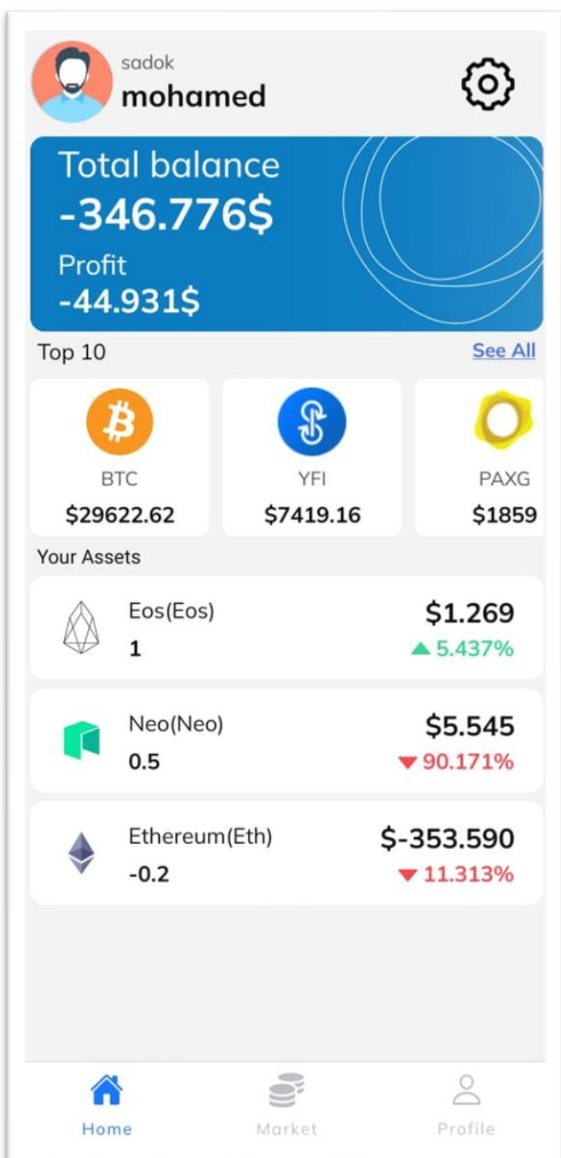


Figure 34 : interface Home

Cette interface (figure 34) représente l'accueil de cette application, elle se compose de 3 parties essentielles la première porte des informations à propos du compte utilisateur qui a fait l'authentification ; ces informations sont le nom et la somme de gain calculés pour le portefeuille choisi par défaut.

La deuxième partie contient une liste des 10 crypto-monnaies les plus élevés selon leur prix en temps réel.

La troisième partie contient une liste qui se compose de plusieurs éléments chaque élément représente la somme des transactions effectuées pour une seule cryptomonnaie ; s'il y a des transactions de ventes la somme totale va augmenter et dans le cas des transactions d'achat elle va diminuer.

Cet élément contient aussi le pourcentage de gain d'après cet « asset » ainsi que sa quantité totale pour ces transactions.

Conclusion

Dans ce rapport, nous avons exposé les étapes de conception et de développement de notre projet de fin d'étude qui consiste à l'implémentation d'une application (mobile) de tracking de la valeur d'un portefeuille.

Pour mener à bien le projet, nous diviser le projet en plusieurs tâches et travailler d'une façon organisée en respectant les priorités de nos besoins.

Ce projet de fin d'étude nous a permet d'accroître notre connaissance en développement hybride, développer notre manière du travail (utilisation d'une architecture, travail en groupe via Git), ainsi que se familiariser avec de nouveaux frameworks, nouvelles technologies, et nouveau sujet.

Nous gardons de ce projet un excellent souvenir, il constitue désormais une expérience professionnelle valorisante et encourageante pour notre avenir.

Pour conclure nous remercions tous les gens qu'ils ont nous aidé pour réaliser ce travail.

Table of Contents

Remerciement.....	5
Introduction	6
Chapitre 1 : Cadre et contexte du projet	7
1.1. Introduction	8
1.2. Enjeux et contexte du projet	8
1.3. Présentation de l'organisme d'accueil.....	8
1.4. Présentation sur les Crypto-monnaies et Blockchain	9
1.4.1 Les crypto-monnaie c'est quoi ?	9
1.4.2 BlockChain c'est quoi ?	9
1.4.3 Smart contracts c'est quoi ? :.....	11
Chapitre 2 : Présentation et cahier des charges	12
2.1. Introduction.....	13
2.2. Présentation du projet.....	13
2.3. Objectifs et périmètres du projet	14
2.4. Public ciblé	14
2.5. Contenu de l'application.....	14
2.6. Charte graphique : exigences et limites.....	15
2.7. Arborescence de l'application	18
2.8. Planification du projet.....	19
2.8.1 Diagramme de GANTT.....	20
2.8.2 Méthode de PERT	20
Chapitre 3 : Technologies et outils utilisés	21
3.1. Introduction	22
3.1. Technologies implémentées	22
3.2.1. Introduction :	22
3.2.2. Javascript :	22
3.2.3. NodeJS :	23
3.2.4. Express Js :	23
3.2.5. MySQL :	24
3.2.6. React native :	24

3.2.7. Expo :	25
3.2. Outils d'implémentation	25
3.3. Problématiques et solutions technologiques :	28
Chapitre 4 : Architecture et Conception	30
4.1. Introduction	31
4.2. Architecture de l'application.....	31
Couche présentation (premier niveau) :	31
Couche métier / business (deuxième niveau) :	31
Couche accès aux données (troisième niveau) :	32
4.3. Conception de l'application	33
4.3.1 Pourquoi UML ?	33
4.3.2 Règles de gestion	34
4.3.3 Diagramme cas d'utilisation.....	34
4.3.4 Diagramme de classes.....	35
Chapitre 5 : Présentation de l'application	38
5.1. Introduction	39
5.1. Etape d'authentification	39
5.2. Etape d'inscription	40
5.3. Profile.....	41
5.4. Market.....	45
5.5. Home.....	48
Bibliographie.....	53

Table des figures

Figure 1 : logo de l'école supérieure de technologie	8
Figure 2: Exemple d'une chaine des blocs	10
Figure 3: Schéma représentant le cycle d'envoi d'une transaction.....	11
Figure 4 : logo de l'application.....	16
Figure 5 : Palette des couleurs.....	17
Figure 6 : Arborescence de l'application.....	19
Figure 7 : Diagramme de GANTT.....	20
Figure 8 : Diagramme de PERT.....	20
Figure 9 : Logo de JavaScript.....	22
Figure 10 : Logo de NodeJS.....	23
Figure 11 : Logo d'Express.....	23
Figure 12 : Logo de MySQL	24
Figure 13 : Logo de React Native	24
Figure 14 : Logo d'Expo	25
Figure 15 : Logo de Visual Studio Code.....	25
Figure 16 : Logo de WampServer.....	26
Figure 17 : Logo de PowerAMC.....	26
Figure 18 : Logo de Postman.....	27
Figure 19 : Logo de Figma	27
Figure 20 : Mise en œuvre d'une web socket.....	29
Figure 21 : schéma de l'application	32
Figure 22 : Diagramme cas d'utilisation.....	35
Figure 23 : Diagramme de classes.....	36
Figure 24 : Etape d'authentification	39
Figure 25 : Etape d'inscription	40
Figure 26 : Interface du Profile	41
Figure 27 : Interface de la liste des transactions	42
Figure 28 : interface de la liste des portefeuilles.....	43
Figure 29 : Modal pour la création d'un portefeuille	44
Figure 30 : Modal pour confirmer la suppression d'un portefeuille.....	44
Figure 31 : La liste des transactions	45
Figure 32 : Statistiques d'une cryptomonnaie	46
Figure 33 : Formulaire pour effectuer une transaction	47
Figure 34 : interface Home	48
Figure 35 : splash screen.....	54
Figure 36 : confirmation de la deconnexion	55

Bibliographie

Web socket for binance , URL : <https://github.com/binance/binance-spot-api-docs/blob/master/web-socket-streams.md>

La documentation de node js : URL : <https://nodejs.org/en/docs/>

La documentation d'Express : URL : <https://expressjs.com/fr/>

La documentation de React Native : URL : <https://reactnative.dev/docs/getting-started>

La documentation de Expo : URL : <https://docs.expo.dev/>

La documentation de PM2 : URL : <https://pm2.keymetrics.io/>

Binance : URL : <https://www.binance.com/en>

CoinGecko : URL : <https://www.coingecko.com/fr>

Annexe

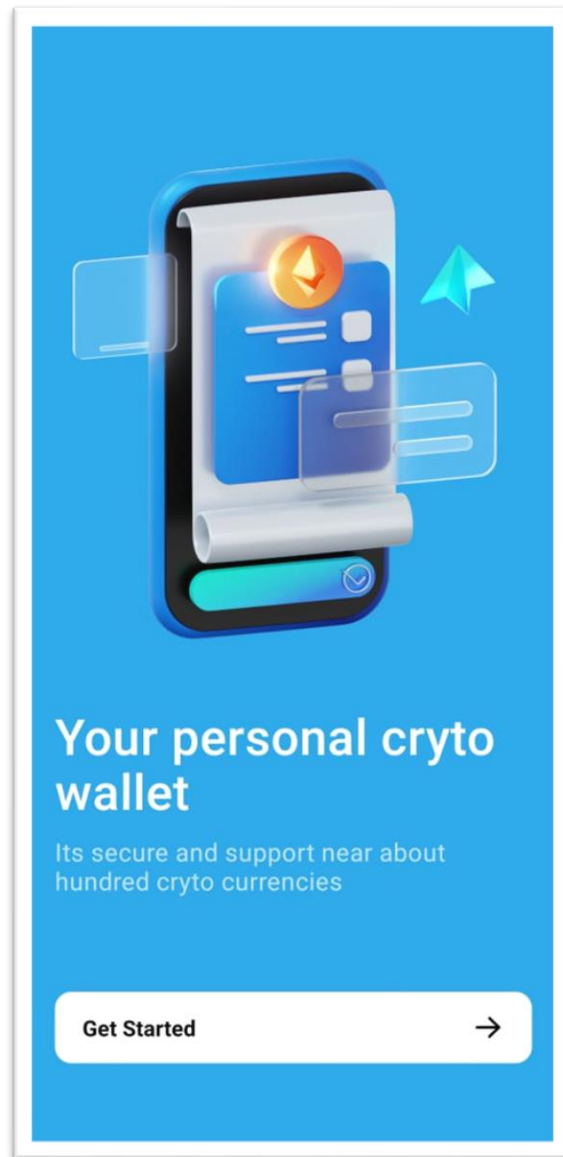


Figure 35 : splash screen

Cette interface (figure 35) s'affiche lors du premier usage de l'application après l'installation

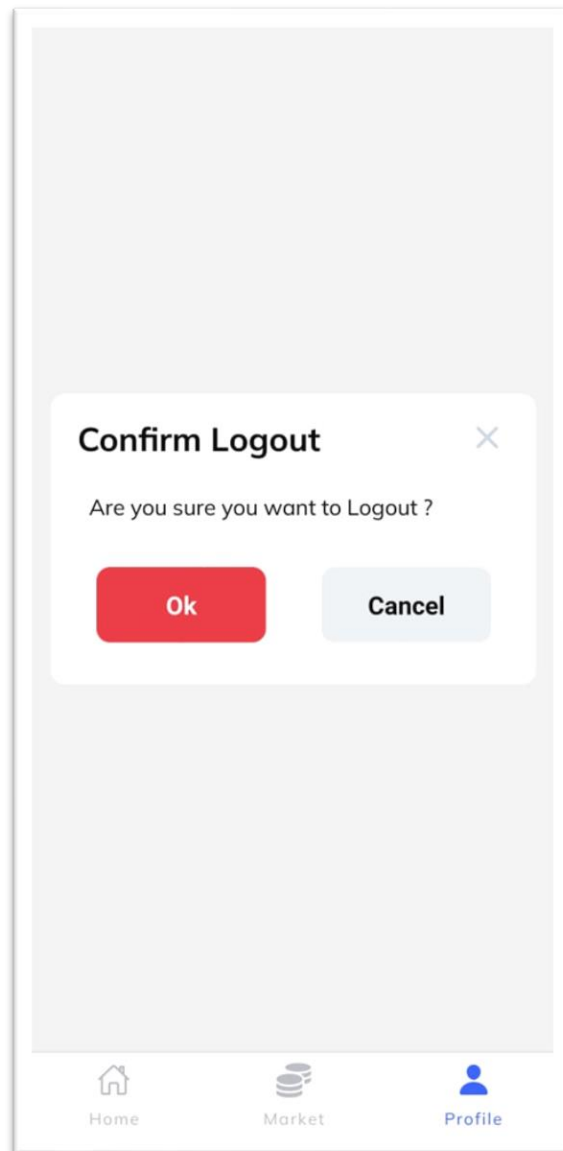


Figure 36 : confirmation de la deconnexion

Lors de la déconnexion s'affiche cette interface qui contient deux boutons le premier pour confirmer la déconnexion et le deuxième pour annuler la déconnexion