

UNIVERSITA’ POLITECNICA DELLE MARCHE

*Facoltà di Ingegneria*

PROGETTO:

REALIZZAZIONE DI UN’APP PER MONITORARE E GESTIRE INVESTIMENTI IN CRIPTOVALUTE

*Relazione di:*

*De Ritis Riccardo*

*Esame di Programmazione mobile A.A. 2021/2022*

*Corso di Laurea Triennale Ingegneria Informatica e dell’Automazione*

#### Introduzione

* Obiettivo

L’obiettivo di tale progetto è la realizzazione di un’app che consente di gestire il tuo portafoglio e ottenere accesso in tempo reale ai movimenti di prezzo di tutte le criptovalute presenti sul mercato. Il mio compito è quello di documentare e implementare un’app che possa essere un utile strumento per tutte quelle persone che si avvicinano a questo nuovo mondo ancora in evoluzione e rendere semplici quelle operazioni che potrebbero risultare molto complicate agli occhi di quest’ultimi. Questo è reso possibile grazie all’utilizzo di diverse API relative ai portali online più importanti, per favorire sia la maggior precisione e affidabilità possibile nei dati che vengono mostrati e sia per dare all’utente tutte le informazioni possibili e necessarie affinché non debba andare altrove per trovare altre informazioni d’interesse.

L’app in questione consente inoltre di scambiare criptovalute utilizzando l’Exchange più grande in termini di volumi, famoso ed affidabile che ci sia: Binance.

* Approccio allo sviluppo

Per quanto riguarda tale punto è stato scelto un approccio agile, ossia metodi iterativi per lo sviluppo rapido del software. Tale approccio si concentra sul codice anziché sulla progettazione e ogni requisito è stato subito implementato in modo da far evolvere rapidamente il software per soddisfare nuove esigenze.

#### Sviluppo in Android

* Descrizione del sistema

Il progetto proposto dovrà quindi consentire la possibilità di : monitorare il prezzo e le prestazioni di ogni criptovaluta in particolare volumi, capitalizzazione e variazioni di prezzo; monitorare volumi e prestazioni del mercato e della “dominance” di Bitcoin; fornire un grafico a candele in tempo reale messo a disposizione da TradingView sulla maggior parte delle criptovalute con informazioni aggiuntive come prezzo massimo/minimo raggiunto durante il giorno; inserire le criptovalute maggiormente seguite all’interno di una watchlist; ricercare ogni tipo di criptovaluta per nome o simbolo; mostrare informazioni dettagliate sulla maggior parte delle criptovalute come la categoria e il settore che ricoprono, la problematica nel mondo reale che intendono risolvere e altre informazioni dettagliate; mostrare le ultime news di rilevanza sul mercato delle criptovalute e fornire l’accesso all’URL per andare a leggerle nel dettaglio; fornire una sezione di login dove l’utente potrà accedere, una volta registratosi e inserito le API di interesse, alla parte relativa alla gestione del portafoglio dove verranno mostrati informazioni utili come un grafico a torta, la propria asset allocation e l’importo totale.

* Glossario dei termini

|  |  |
| --- | --- |
| Termine | Descrizione |
| Criptovaluta | È un’unità di valore digitale, transata e resa sicura in modo decentralizzato sulla blockchain. |
| Bitcoin Dominance | È la percentuale della capitalizzazione di Bitcoin rispetto all’intero mercato delle criptovalute. |
| Grafico a candele | È una rappresentazione grafica dei prezzi di un titolo rappresentato per mezzo di rettangoli verticali che si ottiene dalla relazione tra prezzo di apertura, massimo, minimo e chiusura. |
| TradingView | È un servizio che offre grafici di borsa interattivi e permette di condividere idee e strategie pubblicamente. |
| Asset Allocation | È un processo con il quale si decide in che modo distribuire il proprio capitale fra i diversi possibili investimenti. |
| Exchange (centralizzati) | Sono piattaforme che facilitano l’acquisto e la vendita di asset digitali e sono supervisionati da un’azienda. |

* Requisiti funzionali

|  |  |
| --- | --- |
| Requisito | Descrizione |
| RF1 Visualizzazione lista criptovalute | L’app dovrà visualizzare la lista delle prime 100 criptovalute ordinate per capitalizzazione. |
| RF2 Visualizzazione prestazioni mercato | L’app dovrà visualizzare le prestazioni relative alla capitalizzazione, volumi e dominance di Bitcoin. |
| RF3 Ricerca criptovalute | L’app dovrà gestire la ricerca di qualsiasi criptovaluta. |
| RF4 Visualizzazione grafico a candele | L’app dovrà gestire il click su una criptovaluta e mostrare un grafico a candele sull’andamento del prezzo e varie informazioni aggiuntive. |
| RF5 Visualizzazione dettagli criptovaluta | L’app dovrà visualizzare dettagli specifici su una particolare criptovaluta selezionata. |
| RF6 Inserimento criptovaluta nei preferiti | L’app dovrà permettere all’utente di inserire qualsiasi criptovaluta nella watchlist. |
| RF7 Visualizzazione lista preferiti | L’app dovrà visualizzare una lista contenente tutte le criptovalute inserite nella watchlist. |
| RF8 Visualizzazione delle news | L’app dovrà visualizzare una serie di news rilevanti sul mercato delle criptovalute. |
| RF9 Visualizzazione news sul web | L’app dovrà gestire il click su una news e riportarlo alla fonte principale dove visualizzerà l’articolo per intero. |
| RF10 Visualizzazione pagina utente | L’app dovrà visualizzare le informazioni relative all’utente e permettere anche il reset della password e delle chiavi di accesso al wallet. |
| RF11 Registrazione utente | L’app dovrà gestire la registrazione dell’utente. |
| RF12 Pagina di accesso utente | L’app dovrà consentire all’utente di effettuare l’accesso per sbloccare la sua funzionalità di gestione del portafoglio. |
| RF13 Visualizzazione wallet | L’app dovrà visualizzare il proprio wallet con grafici e statistiche sull’asset allocation. |
| RF14 Scambio criptovalute | L’app dovrà consentire lo scambio effettivo di criptovalute tra tutte le “coppie” presenti su Binance. |

* Requisiti non funzionali

|  |  |
| --- | --- |
| Requisito | Descrizione |
| RNF1 Utilizzo di servizi esterni (API) | L’app utilizzerà diverse API con accesso limitato, dopodiché i dati aggiornati non saranno più reperibili. |

* Descrizione dei casi d’uso

1/2) Visualizzazione lista criptovalute/prestazioni mercato

Questo caso d’suo si verifica all’avvio dell’applicazione dove verrà mostrata una schermata contenente sia la lista delle criptovalute e sia le prestazioni del mercato.

1. Ricerca criptovalute

Questo caso d’uso si verifica quando l’utente vuole cercare una criptovaluta di suo interesse e digitando il suo nome o simbolo sulla barra di ricerca verranno mostrato tutte le cripto che iniziano per quel nome.

1. Visualizzazione grafico a candele

Questo caso d’uso si verifica nel momento in cui l’utente clicca su una criptovaluta nella lista o nella ricerca, verrà aperta una schermata dove verrà mostrato un grafico a candele.

1. Visualizzazione dettagli criptovaluta

Questo caso d’uso si verifica quando l’utente, si trova già nella schermata relativa al grafico a candele e clicca su un bottone, a questo punto si aprirà una schermata contenente tutti i dettagli relativi a quella criptovaluta.

1. Inserimento criptovaluta nei preferiti

Questo caso d’uso si verifica quando l’utente, trovandosi nella schermata principale, clicca sul numero che si riferisce alla classifica delle prime cento criptovalute.

1. Visualizzazione lista preferiti

Questo caso d’uso si verifica quando l’utente naviga nell’app attraverso un Widget nella parte bassa della schermata e clicca sulla sezione “Preferiti”.

Verrà mostrata la lista delle criptovaluta che abbiamo precedentemente messo nei preferiti.

1. Visualizzazione delle news

Questo caso d’uso si verifica quando l’utente naviga nell’app attraverso un Widget nella parte bassa della schermata e clicca sulla sezione “News”.

Verrà mostrata la lista delle news di maggiore rilevanza.

1. Visualizzazione delle news su Web

Questo caso d’uso si verifica quando l’utente, trovandosi nella schermata “News” clicca su una news e verrà riportato sulla pagina ufficiale contenente tutto l’articolo.

1. Visualizzazione pagina utente

Questo caso d’uso si verifica quando l’utente effettua il login nella schermata “Login”, dopo essersi registrato. Verrà mostrata una schermata dove l’utente potrà visualizzare tutti i suoi dati ed eventualmente resettare la password e reimpostare le API.

1. Registrazione utente

Questo caso d’uso si verifica quando l’utente, trovandosi nella schermata “Login” clicca su “Registrati”. Verrà mostrata una schermata contenente tutti i campi da riempire e anche un link tutorial dove l’utente potrà creare le API di Binance.

1. Pagina di accesso utente

Questo caso d’uso si verifica quando l’utente naviga nell’app attraverso un Widget nella parte bassa della schermata e clicca sulla sezione “Login”.

Verrà mostrata una schermata dove l’utente potrà effettuare il login, resettare la password nel caso fosse dimenticata e registrarsi.

1. Visualizzazione wallet

Questo caso d’uso si verifica quando l’utente effettua il login e attraverso un Widget nella parte bassa della schermata e clicca sulla sezione “Wallet”.

Verrà mostrata una schermata contenente un grafico a torta relativo alla propria asset allocation, insieme all’importo totale di ogni criptovaluta posseduta.

1. Scambio criptovalute

Questo caso d’uso si verifica quando l’utente, trovandosi nella schermata “Wallet” clicca su un “Converti”. Verrà mostrata una schermata dove l’utente potrà selezionare la criptovaluta da acquistare/vendere in tutte le coppie presenti su Binance.

* Architettura

Non è stata utilizzata nessuna architettura in particolare, però sono stati seguiti alcuni punti fondamentali per una progettazione corretta dell’architettura al fine di rendere più facile e comprensibile lo sviluppo dell’app. Il primo punto chiave è stato quello di separare i componenti base dell’app, ovvero le Activity dalla memorizzazione dei dati questo perché la parte relativa all’interfaccia non è affidabile al fine di memorizzare i dati a causa di eventi imprevedibili. Il secondo punto fondamentale, quindi, è stato quello di realizzare dei modelli di dati per tutte le informazioni che devono essere gestite, indipendentemente dalle View. Questo oggetto che modella i dati e comunica con la View prende il nome di ViewModel. Grazie a quest’ultima possiamo conservare i dati necessari all’interfaccia utente durante tutta la vita del componente che la contiene. Nel progetto, in particolare, sono state utilizzate classi con lo scopo di memorizzare dati, che prende il nome di Data class, con all’interno un particolare oggetto chiamato Companion Object, che consente di avere una classe interna statica dando accesso alla sua singola istanza, in modo che tutte le componenti che comunicano con quella classe, accedono alla stessa e sola istanza.

* Mockup

Un mockup è una realizzazione a scopo illustrativo o meramente espositivo di un

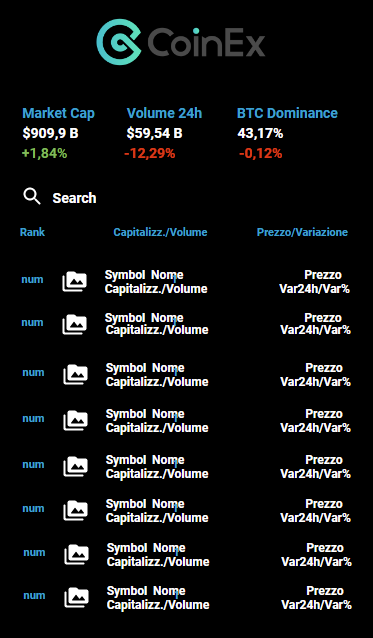
oggetto o un sistema, senza le complete funzioni dell'originale; può rappresentare la

totalità o solo una parte dell'originale di riferimento. Nelle pratiche dell’ingegneria del

software vengono utilizzati per dare un’idea al cliente su quale sarà il risultato del

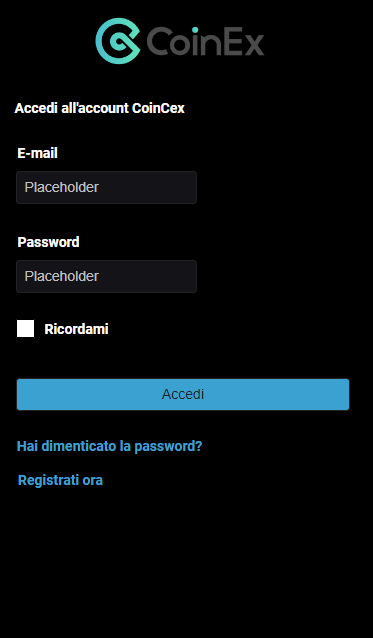
processo software.

Di seguito viene riportato alcune delle schermate di rilevanza dell’applicazione.

 Immagine che contiene testo, elettronico, scuro

Descrizione generata automaticamente

Figura 1: Pagina principale (Market) Figura 2: Pagina News

 Immagine che contiene testo

Descrizione generata automaticamente

Figura 3: Pagina Login Figura 4: Accesso eseguito

Immagine che contiene testo, monitor, schermo, nero

Descrizione generata automaticamente

Figura 5: Pagina Wallet

* Sviluppo

Per quanto riguarda lo sviluppo dell’app, come detto prima, è stato utilizzato un approccio agile. È stata sviluppata come prima parte quella relativa ai modelli dei dati. Sono state testate tutte le funzioni di acquisizione dei dati come quelli relativi ad ogni criptovaluta, al mercato e alle news affinché una volta completato questo ci si potesse concentrare sulla parte dell’interfaccia. Una volta costruiti questi Fragment, è stata realizzata l’Activity principale con all’interno un FrameLayout pensato per ospitare i Fragment precedentemente creati. Successivamente è stata realizzata una BottomNavigationBar che consentisse la navigazione fra questi fragment. Una volta fatta questa prima parte, mi sono concentrato sulla parte del Login, prima implementando una semplice interfaccia e poi andando a realizzare il Database su Firebase. Dove aver compreso la documentazione, ho realizzato il database e subito implementato all’interno dell’app. Nel nostro caso servirà solamente la memorizzazione delle principali informazioni dell’utente e anche di 2 campi per la gestione del portafoglio. Implementato ciò, sono passato alla parte un po' più complessa sia per quando riguarda la chiamata API da fare, quella usata per ottenere le informazioni sul portafoglio, sia a causa delle scarse informazioni dell’API stessa, perché si è dovuto combinare più dati insieme al fine di dare una maggiore precisione ai dati stessi e sia per implementare la funzione che consentisse di acquistare/vendere criptovalute direttamente dall’app senza passare per Binance. Tutte queste chiamate, in quest’ultima parte, sono di tipo POST e dovevano essere crittografate secondo l’algoritmo hash HMACSHA256. Una volta realizzato ciò, è stata testata l’app sia da me che da alcune persone per verificarne l’efficacia e le prestazioni. Di seguito riporto alcune porzioni di codice che sono state spiegate precedentemente.