Software Requirement and Design Specifications

CRYPTEX

Cryptocurrency Trading Simulator

Version: 1.00

Course Code	CS3004-Software Design and Analysis
Instructor	Javeria Farooq
Project Team	Qusai Ezzy K214866 Muhammad Mushtaq K213273 Muhammad Shaheer Luqman K214655
Submission Date	December 4th, 2023

Table of Contents

1. Introduction

1.1. Purpose of Document

The purpose of the Software Requirement and Design Specifications document is to:

1. **Facilitate Communication**: Ensure a shared understanding among stakeholders about the project's goals and features.
2. **Serve as a Reference**: Provide a guide for the development team throughout the project lifecycle.
3. **Basis for Development**: Outline functional and non-functional requirements, serving as a foundation for development.
4. **Support Decision-Making**: Assist in making informed choices regarding design patterns, technologies, and other critical aspects.
5. **Validation and Verification**: Act as a basis for validating and verifying whether the final product meets specified requirements.
6. **Project Management**: Aid in project planning and tracking by providing a clear scope and roadmap.
7. **Document Constraints**: Outline constraints, assumptions, and dependencies that may impact development.
8. **Basis for Testing**: Provide information for creating test cases and ensuring the software functions are as intended.

1.2. Intended Audience

The intended audience for the Software Requirement and Design Specifications document includes:

1. **Developers**: Responsible for coding and implementation.
2. **Designers**: Involved in UI/UX design.
3. **Project Managers**: Oversee project planning and execution.
4. **Testing Teams**: Ensure software meets requirements and quality standards.
5. **Client/Owners**: Commissioning the development, ensuring requirements are met.
6. **System Architects**: Design the overall system architecture.
7. **Technical Writers**: Create user manuals and documentation.
8. **Business Analysts**: Ensure software aligns with business objectives.
9. **Investors/Stakeholders**: Have a financial interest in the project.
10. **Regulatory Compliance Teams**: Ensure compliance with relevant regulations.

1.3. Definition of Terms, Acronyms and Abbreviations

Not Applicable [Self Explanatory Words Used]

1.4 Document Convention

Font Used	Arial
Heading Font Size	16
Subheading Font Size	12
Sub-subheading Font Size	11
Text Size	10

2. Overall System Description

2.1. Project Background

In recent years, the financial landscape has witnessed a revolutionary transformation with the advent of cryptocurrencies. These digital assets, built on decentralized blockchain technology, have disrupted traditional financial systems by offering a secure, transparent, and decentralized alternative for financial transactions and asset management. As the world gradually embraces the potential of cryptocurrencies, there arises a pressing need for comprehensive systems that can efficiently manage, analyze, and provide insights into this rapidly evolving ecosystem.

2.2. Project Scope

Exchange Platform Development: Building a user-friendly and secure exchange platform that allows users to buy, sell, and trade various cryptocurrencies.

Trading Engine: Designing a robust trading engine capable of executing orders swiftly and efficiently, supporting various order types (market, limit, stop-limit, etc.).

Wallet Management: Implementing a wallet system for users to securely store their cryptocurrencies, supporting multiple assets and providing deposit and withdrawal functionalities.

Market Data Aggregation: Collecting and displaying real-time market data, including price charts, order book data, trade history, and other relevant market statistics.

User Account Management: Creating a system for user registration, authentication, KYC (Know Your Customer) verification, and account management with security measures like 2FA (Two-Factor Authentication).

2.3. Not In Scope

This section outlines functionalities that are intentionally excluded from the scope of the current project. The following are not in scope:

1. **Advanced Trading Strategies:** Implementation of advanced trading strategies, such as algorithmic trading, high-frequency trading, etc., is not within the current project's scope. The focus is on basic trading functionalities.
2. **Mobile Application Development:** While mobile responsiveness is considered, the development of dedicated mobile applications (iOS, Android) is not in scope. The project will primarily focus on a responsive web application.
3. **Social Media Integration:** Integrating social media platforms for user authentication or sharing is not in scope for the current project.
4. **Advanced Reporting and Analytics:** Detailed reporting and analytics beyond basic transaction history and portfolio values are not within the scope. This includes complex data visualizations and statistical analyses.
5. **Integration with External Services beyond APIs:** Integrations with services beyond the specified external APIs, such as integration with specific third-party financial services, are not considered in scope.

2.4. Project Objectives

1. **Replicating Binance's Core Functionalities:**
    a. **Objective**: Develop a comprehensive exchange platform mirroring Binance's key functionalities, including trading, wallet management, market data display, and user account security.
    b. **Result**: Create a user-friendly and secure platform that allows users to trade various cryptocurrencies, manage their wallets, access real-time market data, and ensure robust account security similar to Binance.
2. **High-Performance Trading Engine:**
    a. **Objective**: Build a fast and efficient trading engine capable of executing orders swiftly and accurately, ensuring a seamless trading experience for users.
    b. **Result**: Deliver a trading engine that supports various order types, handles high trading volumes, and maintains low latency, providing users with a reliable trading platform.
3. **Secure Wallet Management System:**
    a. **Objective**: Implement a secure wallet system enabling users to securely store and manage their cryptocurrencies within the platform.
    b. **Result**: Provide users with a secure and easy-to-use wallet management system, ensuring the safety of their digital assets.
4. **Real-time Market Data Display:**
    a. **Objective**: Collect and display real-time market data, including price charts, order book data, trade history, and other relevant statistics.
    b. **Result**: Offer users access to comprehensive and accurate market data for informed trading decisions.
5. **User Account Security Measures:**
    a. **Objective**: Incorporate robust security measures, including authentication, encryption, and account protection, to ensure user account security.
    b. **Result**: Safeguard user accounts against unauthorized access and ensure the security of personal and transactional information.
6. **User-Friendly Interface and Experience:**
    a. **Objective**: Design an intuitive and responsive user interface to enhance user experience and accessibility across various devices.
    b. **Result**: Provide users with a seamless and intuitive platform interface for efficient trading and interaction with the system

2.5. Stakeholders

1. **Traders/Investors**: Users engaging in buying, selling, and trading cryptocurrencies on the platform. They seek a reliable, user-friendly, and secure trading environment.
2. **Market Analysts**: Professionals interested in accessing real-time market data, trends, and analytics for informed decision-making and analysis.
3. **Wallet Managers**: Users interested in securely managing their digital assets within the platform's wallet system.
4. **Compliance Officers**: Individuals responsible for ensuring the platform adheres to basic security measures and user verification processes.
5. **Developers/Engineers**: Technical personnel involved in the software development lifecycle, including frontend and backend developers, responsible for creating and maintaining the platform's functionalities.
6. **Database Administrators**: Professionals managing the database architecture, ensuring efficient data storage and retrieval.

2.6. Operating Environment

1. **Hosting Infrastructure:**

   Web Servers: Utilizing reliable and scalable web servers (e.g., AWS EC2 instances, Google Cloud VMs) to host the website.

   Storage: Employing cloud-based storage solutions for storing website data, media files, and other necessary content.

2. **Network Environment:**

   Internet Connectivity: Reliable and high-speed internet connectivity to ensure the website is accessible and responsive to users.

3. **Database Management:**

   Database System: Employing a scalable and efficient database system (such as MySQL) for storing user data, transaction records, and other website-related information.

4. **Web Technologies:**

   Frontend Technologies: Developing the website's frontend using HTML, CSS, and JavaScript frameworks (e.g., React) for an interactive and responsive user interface.

   Backend Technologies: Implementing server-side scripting languages (e.g., Node.js, PHP) and frameworks to handle business logic, user authentication, and database interactions.

2.7. System Constraints

1. **Software Constraints:**
   a. Compatibility with various operating systems (Windows, macOS, Linux, etc.).
   b. Integration with specific programming languages and frameworks.
   c. Dependence on third-party APIs for market data and blockchain interactions.
2. **Hardware Constraints:**
   a. Minimum system requirements for end-users accessing the platform.
   b. Server infrastructure scalability to handle potential spikes in user activity.
3. **Cultural Constraints:**
   a. Multilingual support for users from diverse linguistic backgrounds.
4. **Legal Constraints:**
   a. Compliance with financial and cryptocurrency regulations.
   b. Data protection and privacy laws adherence.
5. **Environmental Constraints:**
   a. Consideration for the noise level in the environment where the software will be accessed.
6. **User Constraints:**
   a. User interface design considerations based on the target audience (e.g., traders, investors).
   b. Accessibility features for users with different abilities.
7. **Off-the-Shelf Component Constraints:**
   a. Limitations imposed by third-party tools or components integrated into the platform.
8. **Concurrency Display Constraint:**
   a. The platform may be unable to show the concurrent user access due to technical limitations or design considerations.

2.8. Assumptions & Dependencies

Assumptions:

1. **Market Stability:** Assuming a relatively stable cryptocurrency market environment for consistent trading activities. Extreme market volatility might affect user behaviors and transaction volumes.
2. **Regulatory Compliance:** Assuming the legal and regulatory landscape pertaining to cryptocurrencies remains stable, allowing for adherence to compliance requirements during system operations.
3. **User Adoption:** Assuming a steady adoption rate of the cryptocurrency exchange platform by traders, investors, and users interested in cryptocurrency trading.
4. **Security Measures:** Assuming the effectiveness of implemented security measures in safeguarding user data, protecting against cyber threats, and ensuring platform integrity.

Dependencies:

1. **Database Installation:** The database should be pre-installed to support data storage and retrieval.
2. **Required Software:** Essential software components such as React, Node.js, and XAMPP should be installed to facilitate the development and deployment of the platform.
3. **External APIs:** Dependence on reliable third-party APIs for market data aggregation, blockchain interactions, or other essential functionalities integrated into the platform.
4. **Internet Connectivity:** Assuming users have reliable internet access for seamless interaction with the website, including trade executions, data access, and account management.

3. External Interface Requirements

3.1. Hardware Interfaces

The cryptocurrency exchange system primarily operates on standard web-based hardware. It requires servers with adequate processing power, memory, and storage to handle high-frequency transactions and real-time data processing. The system does not have specific hardware dependencies beyond these standard requirements.

3.2. Software Interfaces

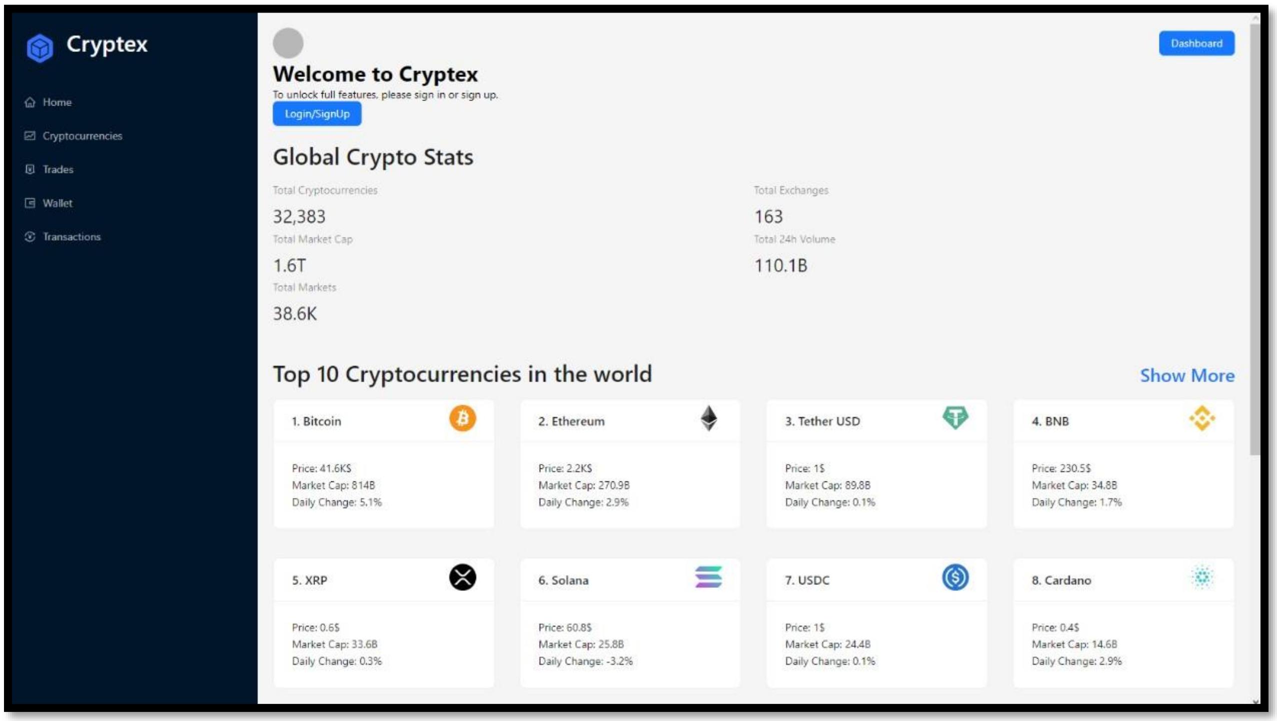
Database: The system interacts with a database management system (e.g., MySQL, PostgreSQL) to store transactional data, user information, market data, and other relevant information. It exchanges data such as user accounts, orders, market prices, etc., with the database.

Operating System: The software runs on a server operating system (e.g., Linux distributions like Ubuntu Server) for web hosting and application deployment.

Web Server: Utilizes web server software (e.g., Nginx, Apache) to handle HTTP requests, manage web application deployment, and ensure optimal performance.

Third-Party APIs: Engages with external APIs provided by market data providers, blockchain networks, payment gateways, etc., to fetch real-time market data, execute transactions, or interact with external systems for additional functionalities.

Security Tools/Libraries: Integrates security tools and libraries for encryption, user authentication, and secure communication protocols, ensuring data security and user privacy.


Cryptex

Home
Cryptocurrencies
Trades
Wallet
Transactions

Dashboard

Welcome to Cryptex

To unlock full features, please sign in or sign up.

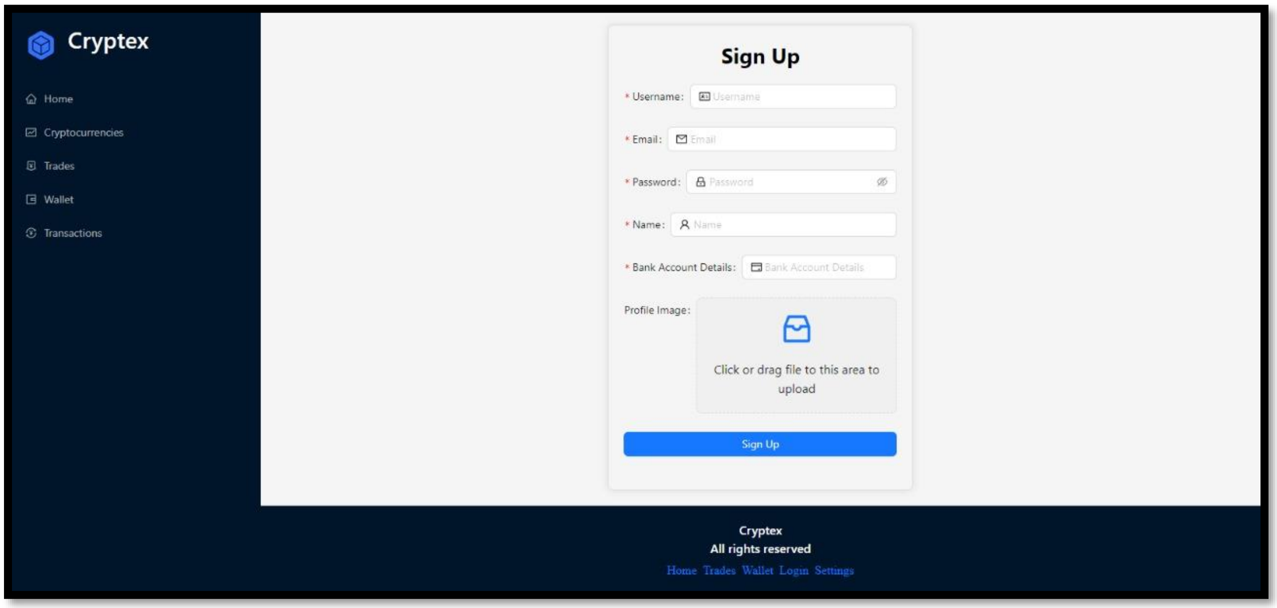
Login/SignUp

Global Crypto Stats

Total Cryptocurrencies 32,383	Total Exchanges 163
Total Market Cap 1.6T	Total 24h Volume 110.1B
Total Markets 38.6K	

Top 10 Cryptocurrencies in the world — Show More

Coin	Price	Market Cap	Daily Change
1. Bitcoin	41.6K$	814B	5.1%
2. Ethereum	2.2K$	270.9B	2.9%
3. Tether USD	1$	89.8B	0.1%
4. BNB	230.5$	34.8B	1.7%
5. XRP	0.6$	33.6B	0.3%
6. Solana	60.8$	25.8B	-3.2%
7. USDC	1$	24.4B	0.1%
8. Cardano	0.4$	14.6B	2.9%


Cryptex

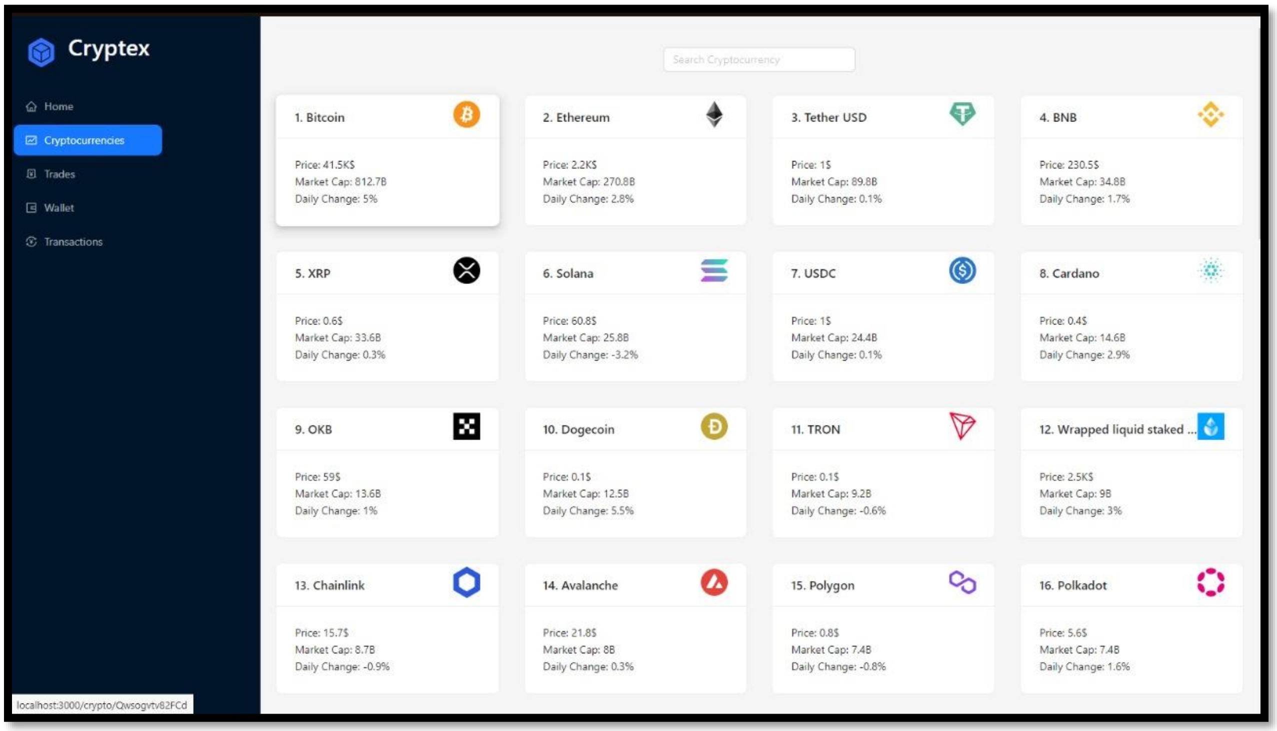
Home
Cryptocurrencies
Trades
Wallet
Transactions

Sign Up

* Username:	Username
* Email:	Email
* Password:	Password
* Name:	Name
* Bank Account Details:	Bank Account Details
Profile Image:	Click or drag file to this area to upload

Sign Up

Cryptex
All rights reserved
Home Trades Wallet Login Settings



1. Bitcoin
Price: 41.5K$
Market Cap: 812.7B
Daily Change: 5%

2. Ethereum
Price: 2.2K$
Market Cap: 270.8B
Daily Change: 2.8%

3. Tether USD
Price: 1$
Market Cap: 89.8B
Daily Change: 0.1%

4. BNB
Price: 230.5$
Market Cap: 34.8B
Daily Change: 1.7%

5. XRP
Price: 0.6$
Market Cap: 33.6B
Daily Change: 0.3%

6. Solana
Price: 60.8$
Market Cap: 25.8B
Daily Change: -3.2%

7. USDC
Price: 1$
Market Cap: 24.4B
Daily Change: 0.1%

8. Cardano
Price: 0.4$
Market Cap: 14.6B
Daily Change: 2.9%

9. OKB
Price: 59$
Market Cap: 13.6B
Daily Change: 1%

10. Dogecoin
Price: 0.1$
Market Cap: 12.5B
Daily Change: 5.5%

11. TRON
Price: 0.1$
Market Cap: 9.2B
Daily Change: -0.6%

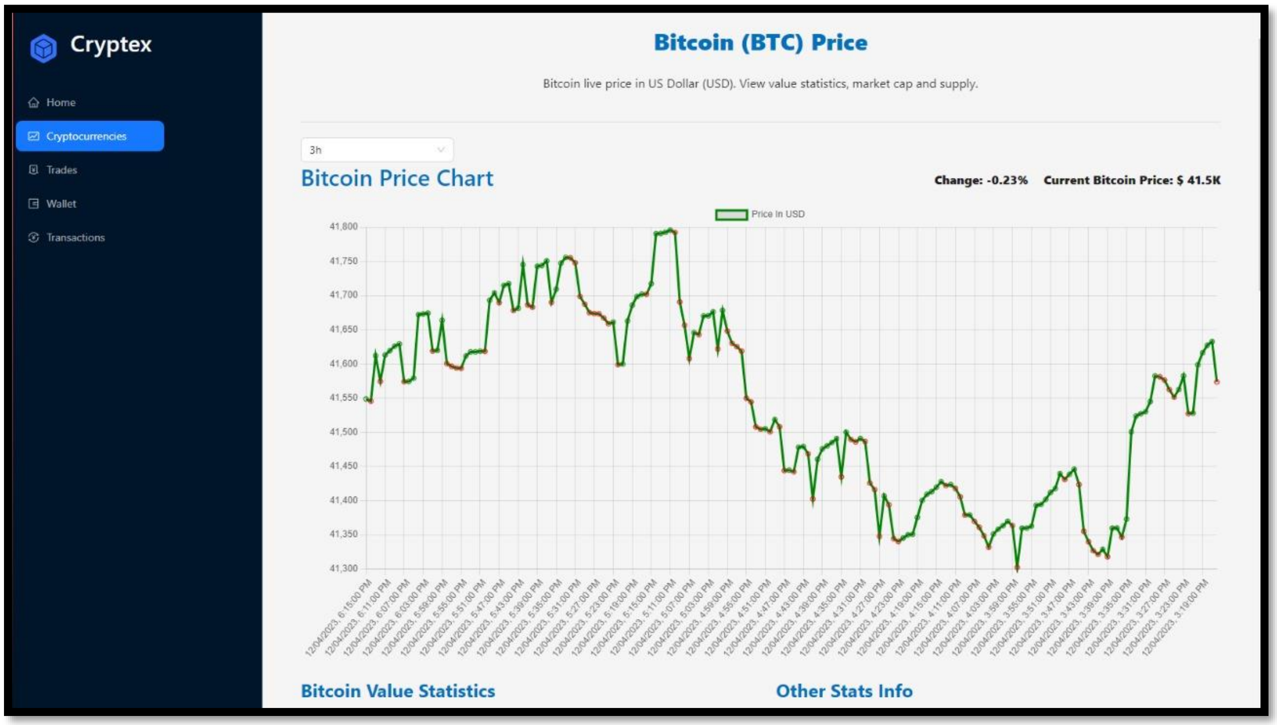
12. Wrapped liquid staked ...
Price: 2.5K$
Market Cap: 9B
Daily Change: 3%

13. Chainlink
Price: 15.7$
Market Cap: 8.7B
Daily Change: -0.9%

14. Avalanche
Price: 21.8$
Market Cap: 8B
Daily Change: 0.3%

15. Polygon
Price: 0.8$
Market Cap: 7.4B
Daily Change: -0.8%

16. Polkadot
Price: 5.6$
Market Cap: 7.4B
Daily Change: 1.6%



Bitcoin (BTC) Price

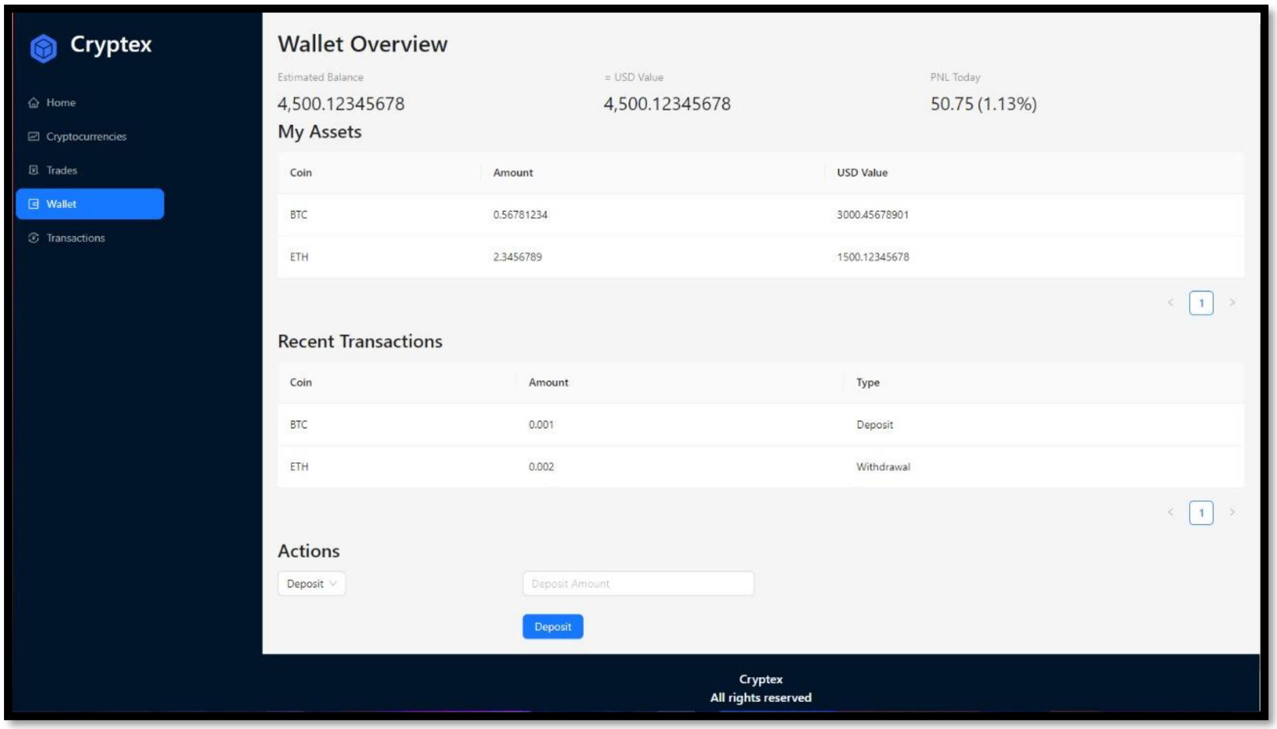
Bitcoin live price in US Dollar (USD). View value statistics, market cap and supply.

Bitcoin Price Chart

Change: -0.23% **Current Bitcoin Price: $ 41.5K**

Bitcoin Value Statistics

Other Stats Info



Wallet Overview

Estimated Balance 4,500.12345678 | ≈ USD Value 4,500.12345678 | PNL Today 50.75 (1.13%)

My Assets

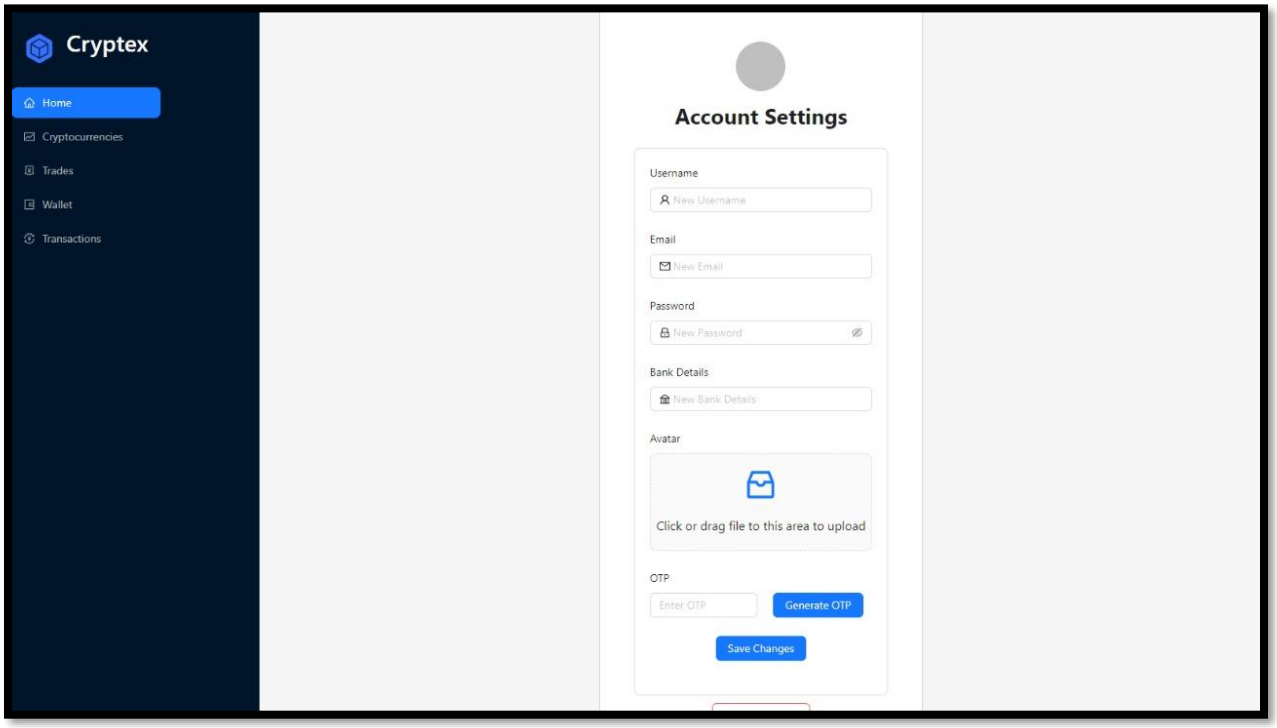
Coin	Amount	USD Value
BTC	0.56781234	3000.45678901
ETH	2.3456789	1500.12345678

Recent Transactions

Coin	Amount	Type
BTC	0.001	Deposit
ETH	0.002	Withdrawal

Actions

Deposit | Deposit Amount



Account Settings

Username: New Username

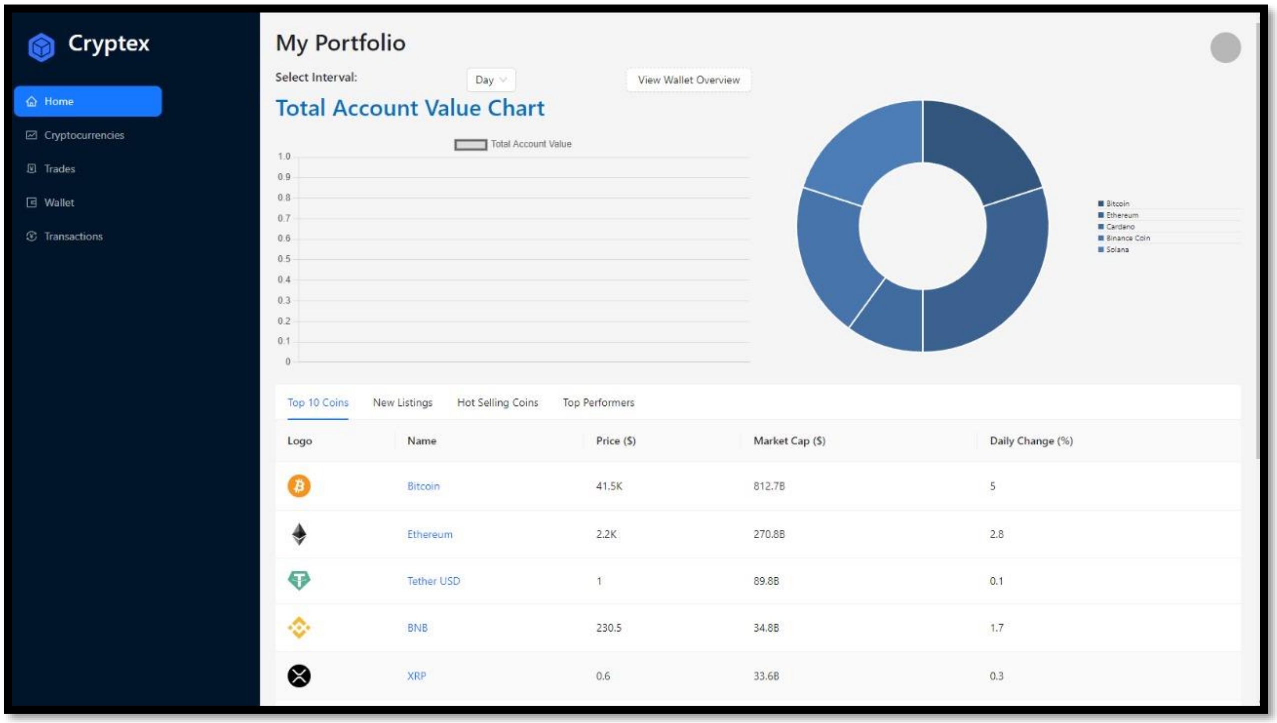
Email: New Email

Password: New Password

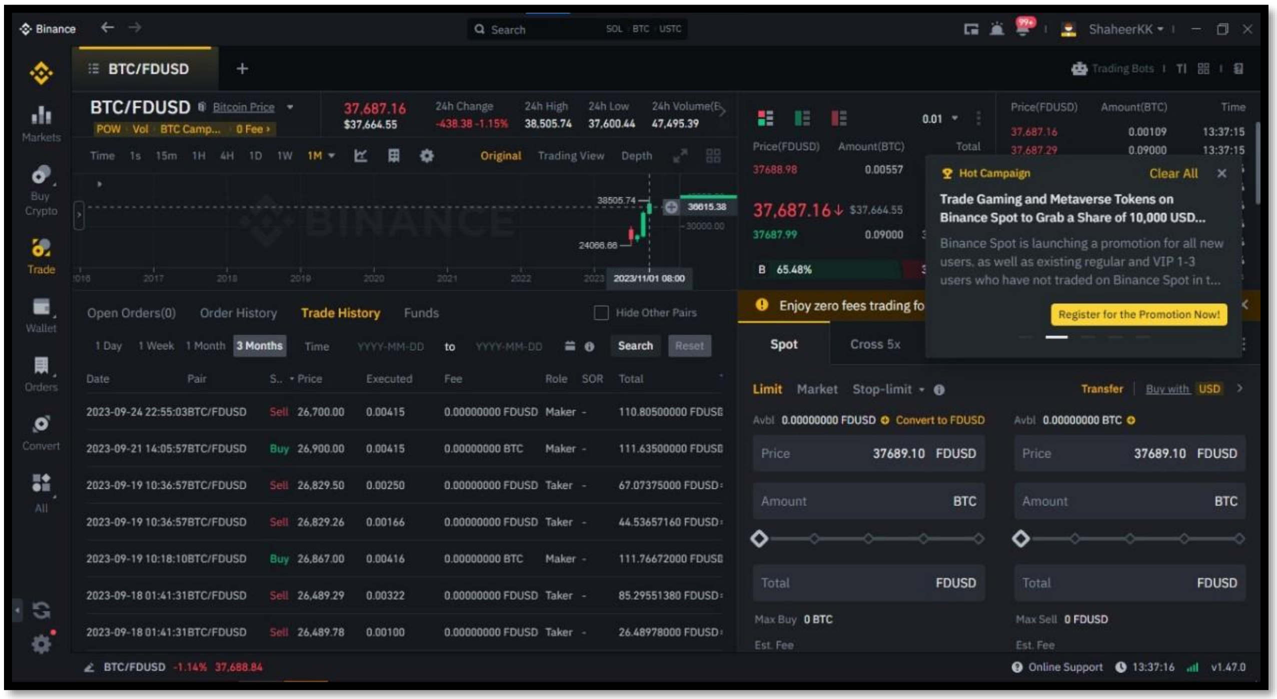
Bank Details: New Bank Details

Avatar: Click or drag file to this area to upload

OTP: Enter OTP



Logo	Name	Price ($)	Market Cap ($)	Daily Change (%)
	Bitcoin	41.5K	812.7B	5
	Ethereum	2.2K	270.8B	2.8
	Tether USD	1	89.8B	0.1
	BNB	230.5	34.8B	1.7
	XRP	0.6	33.6B	0.3



Trade Gaming and Metaverse Tokens on Binance Spot to Grab a Share of 10,000 USD...

Binance Spot is launching a promotion for all new users, as well as existing regular and VIP 1-3 users who have not traded on Binance Spot in t...

Register for the Promotion Now!

Open Orders(0) Order History **Trade History** Funds

1 Day 1 Week 1 Month **3 Months** Time YYYY-MM-DD to YYYY-MM-DD Search Reset

Date	Pair	S..	Price	Executed	Fee	Role	SOR	Total
2023-09-24 22:55:03	BTC/FDUSD	Sell	26,700.00	0.00415	0.00000000 FDUSD	Maker	-	110.80500000 FDUSD
2023-09-21 14:05:57	BTC/FDUSD	Buy	26,900.00	0.00415	0.00000000 BTC	Maker	-	111.63500000 FDUSD
2023-09-19 10:36:57	BTC/FDUSD	Sell	26,829.50	0.00250	0.00000000 FDUSD	Taker	-	67.07375000 FDUSD
2023-09-19 10:36:57	BTC/FDUSD	Sell	26,829.26	0.00166	0.00000000 FDUSD	Taker	-	44.53657160 FDUSD
2023-09-19 10:18:10	BTC/FDUSD	Buy	26,867.00	0.00416	0.00000000 BTC	Maker	-	111.76672000 FDUSD
2023-09-18 01:41:31	BTC/FDUSD	Sell	26,489.29	0.00322	0.00000000 FDUSD	Taker	-	85.29551380 FDUSD
2023-09-18 01:41:31	BTC/FDUSD	Sell	26,489.78	0.00100	0.00000000 FDUSD	Taker	-	26.48978000 FDUSD

BTC/FDUSD -1.14% 37,688.84

Trading Page not created as of yet, but will be similar to this.

3.3. Communications Interfaces

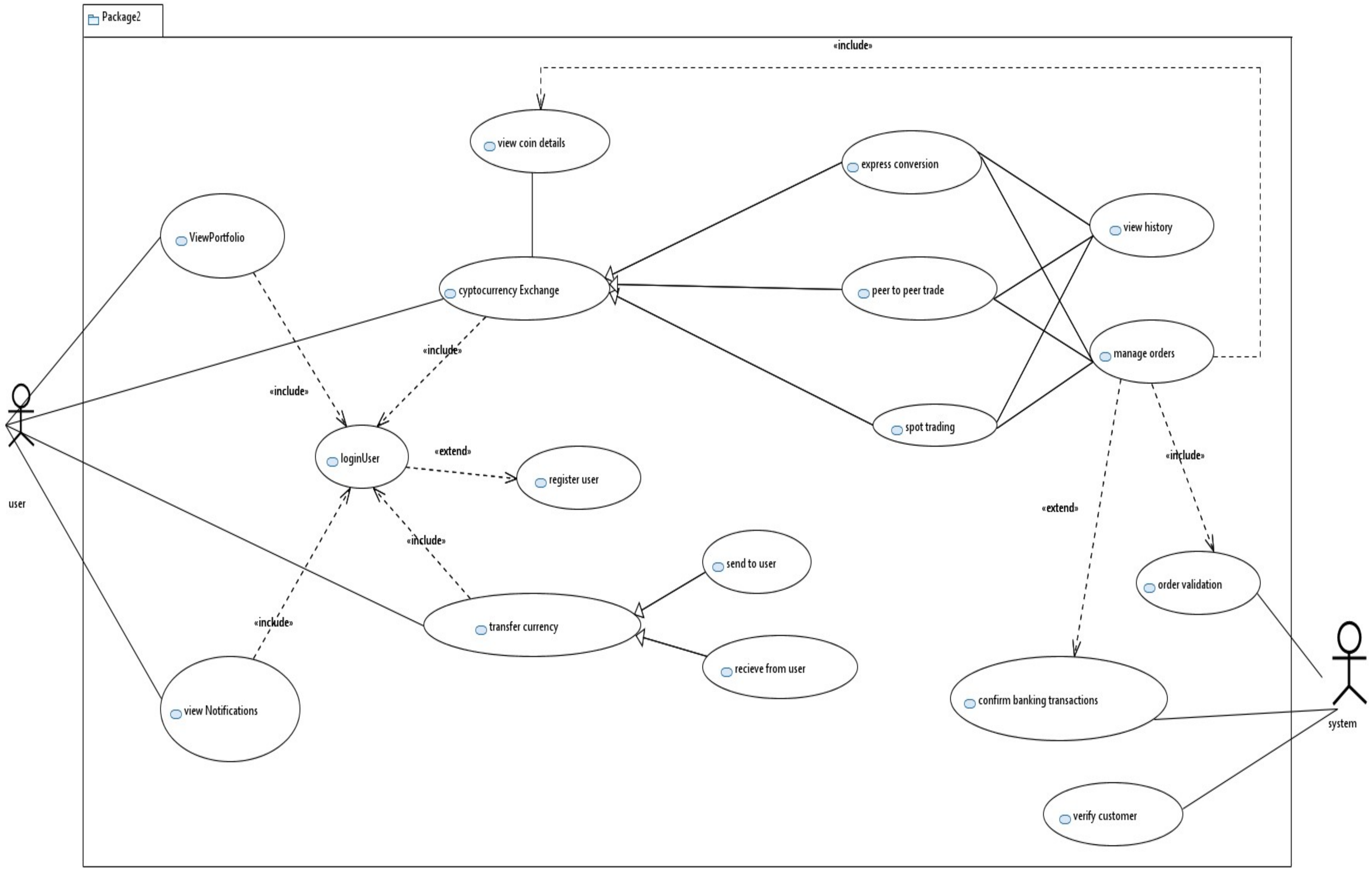
1. **Web Browser Interface:**
    - Requirement: The system requires compatibility with standard web browsers (Chrome, Firefox, Safari, etc.) for user interaction.
    - Message Formatting: Uses HTML, CSS, and JavaScript for web page rendering and interaction.
2. **API Communication:**
    - Requirement: Interacts with external APIs for fetching real-time market data, executing transactions, etc.
    - Protocols: Utilizes Redux Toolkit React for API interactions.
    - Message Formatting: Utilizes JSON or XML for data exchange between the system and external APIs.
3. **Network Communications Standards:**
    - Requirement: Relies on standard networking protocols for data transmission and communication between servers and users.
    - Protocols: Uses HTTP/HTTPS for web-based communication and TCP/IP for general network communications.
4. **Data Transfer Rates and Synchronization:**
    - Requirement: Ensures efficient data transfer rates for real-time market data and transaction processing.
    - Synchronization: Requires synchronization mechanisms to update market prices, transaction statuses, and user account information in near-real-time across the system.
5. **Authentication and Authorization:**
    - Requirement: Implements secure authentication mechanisms for user logins.
    - Encryption: Ensures encryption of sensitive user data, credentials, and tokens during authentication and authorization processes.

4. Functional Requirements

4.1. Functional Hierarchy

1. **User Management:**
    a. User Registration:
        i. Validate user input data.
        ii. Generate unique user identifiers.
        iii. Store user details securely.
    b. User Authentication:
        i. Authenticate users during login.
        ii. Implement secure password handling.
2. **Trading Operations:**
    a. Order Placement:
        i. Allow users to place buy/sell orders.
        ii. Validate order parameters and user balances.
        iii. Match buy and sell orders in the market.
3. **Market Data Presentation:**
    a. Fetch and display real-time market data.
    b. Provide trading charts, order books, etc., for analysis.
4. **Wallet and Asset Management:**
    a. Wallet Creation:
        i. Generate wallet for different cryptocurrencies.
        ii. Associate wallet with user accounts.
    b. Portfolio Monitoring:
        i. Display user holdings and values.
        ii. Calculate and update portfolio values in real-time.
    c. Transaction Handling:
        i. Transaction Execution:
            1. Facilitate trades between users.
        ii. Transaction History:
            1. Record and display transaction logs.
            2. Offer filters and search options for transaction history.
5. **Admin Functions:**
    a. User Account Management:
        i. Admin tools for user account oversight.
        ii. User suspension, deletion, or privilege management.
    b. System Configuration:
        i. Configure system settings and parameters.
        ii. Manage market pairs, trading fees, etc.
6. **Notification System:**
    a. Notification Generation:
        i. Generate alerts for completed transactions, order status changes, etc.
        ii. Deliver notifications via email or platform alerts.
    b. User Notification Preferences:
        i. Allow users to manage notification settings.

4.2. Use Cases


Package2
«include»
view coin details
express conversion
ViewPortfolio
view history
cyptocurrency Exchange
peer to peer trade
manage orders
«include»
«include»
spot trading
loginUser
«extend»
register user
«include»
«extend»
user
«include»
send to user
order validation
«include»
transfer currency
recieve from user
confirm banking transactions
view Notifications
system
verify customer

4.2.1. Log-in/Sign-up

Use Case name: Log-in/Sign-up	
Use Case Description: This use case describes the process wherein an existing user accesses the cryptocurrency exchange platform by providing valid credentials to log into their account.	
Primary actor: User	**Other actors:** System
Stakeholders: User	
Relationships **Includes:** • view portfolio • view notifications • cryptocurrency Exchange • Transfer currency **Extends:** • Register User	
Pre-conditions: • The user possesses valid login credentials (username/email and password). • The user has registered an account on the cryptocurrency exchange platform.	
Flow of Events: 1. The user navigates to the login page of the cryptocurrency exchange platform. 2. The system presents the login interface prompting the user to enter their registered username/email and password. 3. The user enters the login credentials and submits the form for authentication. 4. The system verifies the entered credentials against the stored data in the database. 5. If the credentials match an existing user account: 6. The system grants access to the user's account. 7. The user is redirected to their dashboard or the main page of the platform. 8. If the credentials are invalid or not found: 9. The system prompts the user with an error message indicating incorrect credentials. 10. The user is given the option to retry login or reset the password.	
Alternative and exceptional flows: If the entered credentials do not match any registered account: The system denies access and prompts the user to re-enter correct credentials or reset the password.	
Post-conditions: Upon successful authentication, the user gains access to their account and the platform's features based on their permissions and role.	

4.2.2. View Portfolio

Use Case name: View Portfolio	
Use Case Description: This use case describes the process wherein a registered user accesses and views their cryptocurrency portfolio within the cryptocurrency exchange platform.	
Primary actor: User **Stakeholders:** User	**Other actors:** System
Relationships ▪ Includes**:** • none **Extends:** • none	
Pre-conditions: The user is logged into their registered account on the cryptocurrency exchange platform. The user has holdings or transactions within their portfolio.	
Flow of Events: 1. The user navigates to the "Portfolio" section or dashboard of the cryptocurrency exchange platform. 2. The system presents the user's portfolio interface displaying various details such as: 3. Cryptocurrency holdings (coins/tokens). 4. Quantity or amount of each cryptocurrency held. 5. Current valuation or market value of the holdings. 6. Historical performance or trends, if available. 7. The user reviews their portfolio: 8. They view the list of cryptocurrencies they currently hold. 9. They observe the quantity or amount of each cryptocurrency held. 10. They check the current valuation or market value of their holdings. 11. They may explore historical performance charts or trends to track changes. 12. The user may perform additional actions, such as: 13. Initiating buy/sell orders based on their portfolio analysis. 14. Reviewing transaction history or details of specific holdings, if available.	
Alternative and exceptional flows: • If the user has no holdings or transactions in their portfolio: • The system displays a message indicating an empty portfolio or suggests actions to start trading or acquiring cryptocurrencies.	
Post-conditions: The user has reviewed the details of their cryptocurrency portfolio, allowing them to make informed decisions or take actions related to their holdings within the platform.	

4.2.3. View Coin details

Use Case Name : View Coin details	
Use Case Description : This use case describes the process wherein a user accesses and reviews detailed information about a specific cryptocurrency within the cryptocurrency exchange platform.	
Primary actor: User	**Other actors:** System
Stakeholders: User:	.
Relationships: **Includes:** Manage orders **Extends:** None	
Pre-conditions: • The user is logged into their registered account on the cryptocurrency exchange platform. • The platform provides access to detailed information about individual cryptocurrencies	
Flow of Events: • The user navigates to the "Coin Details" or similar section within the platform. • The system presents the user interface to search or select a specific cryptocurrency for detailed information. • The user selects a cryptocurrency of interest from the available list. • The system displays comprehensive details about the selected cryptocurrency: • Basic Information: Such as name, symbol, market cap, circulating supply, etc. • Price Information: Current price, price history, market trends, etc. • Technical Details: Algorithm, consensus mechanism, block time, etc. • Historical Performance: Charts or graphs depicting historical price data, volume, etc. • Community and Development: Links to community forums, social media, official websites, development progress, etc.	
Alternative And Exceptional Flows: • If the selected cryptocurrency details are not available: • The system displays a message indicating that details for the specific coin are not accessible now.	
Post-conditions: • The user has obtained comprehensive information about the selected cryptocurrency, enabling them to make informed decisions regarding their investments or interests.	

4.2.4. Transfer Currency

Use Case Name : Transfer Currency	
Use Case Description : This use case describes the process wherein a registered user initiates a transfer of cryptocurrency from their wallet to another user's wallet within the cryptocurrency exchange platform.	
Primary actor: User	**Other actors:** System
Stakeholders: Sender User Reciever User	
Relationships: **Includes:** Login/sign-up **Extends:** None	
Pre-conditions: • The sender user is logged into their registered account on the cryptocurrency exchange platform. • Sufficient balance of the selected cryptocurrency is available in the sender user's wallet.	
Flow of Events: • The sender user navigates to the "Transfer" or "Send Currency" section within the platform. • The system presents the interface to initiate a transfer, prompting the sender to input necessary details: 1. Receiver's Wallet Address: Unique wallet address of the recipient user. 2. Amount to Transfer: Quantity or amount of the cryptocurrency to be transferred. • The sender user enters the required details for the transfer. • The system validates the input: 1. Verifies the correctness of the recipient's wallet address format. 2. Validates the availability of the required balance in the sender's wallet. • If the input details are valid: 1. The system processes the transfer of cryptocurrency from the sender's wallet to the recipient's wallet. 2. The sender user receives a confirmation of the successful transfer. • The system updates the sender and recipient wallet balances accordingly.	
Alternative And Exceptional Flows: • If the entered recipient's wallet address is invalid or incorrectly formatted: The system prompts the sender user to re-enter the correct recipient's wallet address.	
Post-conditions: • The sender user has successfully transferred cryptocurrency from their wallet to the recipient's wallet, and both parties receive confirmations of the transaction.	

4.2.5. Peer to Peer trading

Use Case Name : Peer to Peer trading

Use Case Description :

This use case describes the process wherein two registered users engage in direct trading of cryptocurrencies within the cryptocurrency exchange platform without involving the platform's order book.

Primary actor: User	**Other actors:** System
Stakeholders: Trading Users (Buyer and Seller): Aim to execute a direct trade of cryptocurrencies.	

Relationships:

Includes: None

Extends: None

Pre-conditions:

- Both the buyer and seller users are logged into their registered accounts on the cryptocurrency exchange platform.
- Both users possess the required cryptocurrencies in their respective wallets for the trade.

Flow of Events:

- The buyer user expresses interest in purchasing a specific cryptocurrency from the seller.
- The seller user confirms their willingness to sell the requested cryptocurrency directly to the buyer.
- Both users agree on the terms of the trade:
    1. Cryptocurrency Type and Quantity: Agreed amount and cryptocurrency to be traded.
    2. Price or Exchange Rate: Mutually agreed-upon rate or price for the trade.
    3. Transaction Details: Any additional terms or details related to the trade, if applicable.
- The buyer and seller users navigate to the "Peer-to-Peer Trading" section or interface within the platform.
- The system presents an interface for the users to input trade details:
    1. Cryptocurrency Type and Quantity: The agreed-upon cryptocurrency and quantity for the trade.
    2. Trade Terms Confirmation: Confirmation of the trade terms agreed upon by both parties.
- The buyer and seller users confirm the trade details and initiate the transaction.
- The system validates the trade request:
    1. Ensures both users have the required balance for the trade.
    2. Verifies the accuracy of the entered trade details.
- If the trade details are valid:
    1. The system executes the peer-to-peer trade, transferring the agreed cryptocurrencies between the users' wallets.
    2. Both buyer and seller users receive confirmations of the successful transaction.

Alternative And Exceptional Flows:

- If there is a discrepancy in the agreed trade terms:
- The system prompts the users to review and reconfirm the trade details before execution.

Post-conditions:

- Both the buyer and seller users have completed a successful peer-to-peer trade of cryptocurrencies, and their wallet balances reflect the updated transactions.

4.2.6. Express Conversion

Use Case Name : Express Conversion	
Use Case Description : This use case describes the process wherein a registered user swiftly converts one cryptocurrency to another at the current market rate within the cryptocurrency exchange platform.	
Primary actor: User	**Other actors:** System
Stakeholders: User	
Relationships: **Includes:** None **Extends:** None	
Pre-conditions: • The user is logged into their registered account on the cryptocurrency exchange platform. • Sufficient balance of the source cryptocurrency is available in the user's wallet for conversion.	
Flow of Events: • The user navigates to the "Express Conversion" or similar section within the platform. • The system presents the user interface to perform express conversion: 1. The user selects the source cryptocurrency they wish to convert. 2. The user specifies the target cryptocurrency they want to receive. 3. The system displays the current market rate or exchange rate for the conversion. • The user inputs the amount or quantity of the source cryptocurrency they want to convert. • The system calculates the estimated amount of the target cryptocurrency to be received based on the provided quantity and current market rate. • The user confirms the conversion details and initiates the express conversion. • The system validates the conversion request: 1. Ensures the availability of the required balance for the conversion. 2. Verifies the accuracy of the entered conversion details. • If the conversion details are valid: 1. The system executes the express conversion, swiftly converting the specified amount of the source cryptocurrency to the target cryptocurrency at the current market rate. 2. The user receives the equivalent amount of the target cryptocurrency in their wallet.	
Alternative And Exceptional Flows: • If the user's balance for the source cryptocurrency is insufficient: • The system notifies the user and prompts them to adjust the conversion amount or ensure sufficient balance for the conversion	
Post-conditions: • The user has successfully performed an express conversion, swiftly converting one cryptocurrency to another at the current market rate, receiving the equivalent amount in their wallet.	

4.2.7. Spot trading

Use Case Name : Spot trading	
Use Case Description : This use case describes the process wherein a registered user engages in spot trading of cryptocurrencies by placing orders based on the current market prices within the cryptocurrency exchange platform.	
Primary actor: User	**Other actors:** System
Stakeholders: User	

Relationships:

Includes: None

Extends: None

Pre-conditions:

- The user is logged into their registered account on the cryptocurrency exchange platform.
- The user possesses a sufficient balance of the cryptocurrency they intend to trade.

Flow of Events:

- The user navigates to the "Spot Trading" or "Market" section within the platform.
- The system presents the user interface for spot trading:
  1. The user selects the cryptocurrency pair (e.g., BTC/ETH) they wish to trade.
  2. The system displays the current market prices, including bid and ask prices for the selected pair.
- The user chooses the type of order they want to place:
  1. Buy Order: The user specifies the amount of cryptocurrency they want to purchase at the best available market price.
  2. Sell Order: The user specifies the quantity of cryptocurrency they want to sell at the best available market price.
- The user inputs the quantity or amount of cryptocurrency for the order.
- The system calculates the estimated cost (for buy orders) or estimated proceeds (for sell orders) based on the specified quantity and current market rates.
- The user confirms the order details and submits the spot trading order.
- The system validates the order request:
  1. Ensures the availability of the required balance or quantity for the order.
  2. Verifies the accuracy of the entered order details.
- If the order details are valid:
  1. The system executes the spot trading order, matching the user's buy or sell order with the best available market price.
  2. The user's account is updated with the purchased cryptocurrency (for buy orders) or the equivalent value in their currency (for sell orders).

Alternative And Exceptional Flows:

- If the market conditions change rapidly:
- The system notifies the user of any price changes before executing the order, allowing them to review and confirm the updated details.

Post-conditions:

- The user has successfully placed a spot trading order, either buying or selling cryptocurrencies based on the best available market prices within the cryptocurrency exchange platform.

4.2.8. View History

Use Case Name : View History	
Use Case Description : This use case describes the process wherein a registered user accesses and reviews their transaction history or activity log within the cryptocurrency exchange platform.	
Primary actor: User	**Other actors:** System
Stakeholders: User	
Relationships: **Includes:** None **Extends:** None	
Pre-conditions: The user is logged into their registered account on the cryptocurrency exchange platform.	
Flow of Events: • The user navigates to the "History" or "Transaction History" section within the platform. • The system presents the user interface displaying historical transaction data or activity log: 1. The interface includes options to filter or search for specific types of transactions or a specific time period. 2. It displays details such as transaction date/time, transaction type (buy, sell, transfer), cryptocurrency involved, quantity, value, and any associated fees. • The user reviews their transaction history: 1. They can scroll through the list of past transactions or use filters to find specific transactions. 2. They examine transaction details to track past trades, transfers, purchases, or sales made within the platform. • The user may perform additional actions based on the historical data: 1. They might use the information for record-keeping, accounting, or tax purposes. 2. They may analyze past transactions to make informed decisions for future trades or activities.	
Alternative And Exceptional Flows: • If the user has no transaction history or activity log: • The system displays a message indicating an absence of recorded transactions.	
Post-conditions: The user has reviewed their transaction history or activity log, allowing them to track and analyze past activities conducted within the cryptocurrency exchange platform.	

4.2.9. Manage Orders

Use Case Name : Manage Orders	
Use Case Description : This use case describes the process wherein a registered user interacts with the cryptocurrency exchange platform to manage their existing buy or sell orders.	
Primary actor: User	**Other actors:** System
Stakeholders: User:	
Relationships: **Includes:** view coin details **Extends:** None	
Pre-conditions: • The user is logged into their registered account on the cryptocurrency exchange platform. • The user has active buy or sell orders that they wish to manage.	
Flow of Events: • The user navigates to the "Orders" or "Manage Orders" section within the platform. • The system presents the user interface displaying their existing buy or sell orders: • The interface lists active orders along with details such as order type, cryptocurrency pair, quantity, price, status, etc. • Options to modify or cancel orders are available for each listed order. • The user selects an order they want to manage. • The system provides options to modify or cancel the selected order: Modify Order: The user can adjust the quantity or price of the order before execution. Cancel Order: The user can choose to cancel the selected order if it has not been executed yet. • The user confirms the modification or cancellation of the order. • The system validates the request: Ensures that the order can be modified or canceled according to platform rules. Verifies the accuracy of the entered modification or cancellation details. • If the request is valid: For modification: The system updates the order details as per the user's modifications. For cancellation: The system cancels the order, removing it from the active orders list.	
Alternative And Exceptional Flows: • If the selected order cannot be modified or canceled: • The system notifies the user of the reason for inability (e.g., order already executed) and prompts them to recheck their actions.	
Post-conditions: The user has successfully managed their buy or sell orders, either modifying their parameters or canceling them within the cryptocurrency exchange platform.	

4.2.10. Order validation

Use Case Name : Order validation	
Use Case Description : This use case describes the process wherein the cryptocurrency exchange platform validates and processes a user's buy or sell order request.	
Primary actor: System	**Other actors: -**
Stakeholders: User	
Relationships: **Includes: Manage order** **Extends:** confirm banking transaction	
Pre-conditions: The user has initiated a buy or sell order request on the cryptocurrency exchange platform.	
Flow of Events: • The user places a buy or sell order request on the platform: For a buy order: Specifies the cryptocurrency type, quantity, and price they are willing to pay. For a sell order: Specifies the cryptocurrency type, quantity, and price they are willing to sell for. • The system receives the order request and validates it: Checks the availability of the required balance or quantity for the buy/sell order. Verifies that the specified price aligns with current market rates (for limit orders). Validates the order against platform-specific rules and limitations. • If the order passes validation: For a buy order: The system checks for available sell orders matching the user's buy criteria and executes the trade. For a sell order: The system checks for available buy orders matching the user's sell criteria and executes the trade. • If the order does not pass validation: The system rejects the order, providing a reason for rejection (e.g., insufficient balance, invalid price, etc.). Notifies the user about the rejection of the order request.	
Alternative And Exceptional Flows: • If there are fluctuations in market prices between order placement and execution: 1. The system may notify the user about price changes and request confirmation for the modified prices before executing the trade	
Post-conditions: • If the order is validated and executed, the user's buy/sell request is processed, and the corresponding cryptocurrency or equivalent value is exchanged accordingly. • If the order is not validated, the user is informed about the rejection of the order request, and no trade is executed.	

4.2.11. Confirm Banking Transaction

Use Case Name: Confirm Banking Transaction	
Use Case Description : This use case describes the process wherein a user confirms banking transactions related to depositing or withdrawing funds on the cryptocurrency exchange platform.	
Primary actor: User	**Other actors:** System
Stakeholders: User	
Relationships: **Includes:** **Extends:** Manage order	
Pre-conditions: • The user has initiated a deposit or withdrawal transaction on the cryptocurrency exchange platform using banking methods.	
Flow of Events: • The user initiates a deposit or withdrawal transaction through banking methods: • Deposit: Specifies the amount to be deposited into the platform from their bank account. • Withdrawal: Requests a withdrawal of a certain amount from their platform account to their linked bank account. • The system processes the user's transaction request and interfaces with the banking system: • For deposits: The system initiates a request to debit the specified amount from the user's bank account and credit it to their platform account. • For withdrawals: The system initiates a request to debit the specified amount from the user's platform account and credit it to their bank account. • The banking system executes the transaction and sends a confirmation request to the user for validation. • The user receives the confirmation request: • For deposits: Verifies the details of the deposit transaction and confirms the transfer from their bank account to the platform. • For withdrawals: Validates the details of the withdrawal transaction and confirms the transfer from the platform to their bank account. • The user confirms the banking transaction through the provided confirmation method (e.g., OTP, PIN, confirmation button). • The system receives the user's confirmation: • For deposits: Confirms the successful deposit of funds into the user's platform account. • For withdrawals: Confirms the successful transfer of funds from the user's platform account to their bank account.	
Alternative And Exceptional Flows: • If the user does not confirm the banking transaction within a specified time: 1. The system may timeout and mark the transaction as pending, requiring user reconfirmation or intervention.	
Post-conditions: • Upon user confirmation, the deposit or withdrawal of funds through banking transactions is completed, and the respective balances on the platform and the user's bank account are updated accordingly.	

4.2.12. Verify a user

Use Case Name : Verify a user	
Use Case Description : This use case describes the process wherein the cryptocurrency exchange platform verifies the identity of a customer to ensure compliance with regulatory requirements and enhance security.	
Primary actor: System	**Other actors:** user
Stakeholders: User, System	
Relationships: **Includes:** None **Extends:** None	
Pre-conditions: • The user has registered an account on the cryptocurrency exchange platform. • The user has initiated the process of identity verification.	
Flow of Events: • The user initiates the identity verification process on the platform: Navigates to the "Verify Identity" or similar section within their account settings. Selects the type of identity document to be used for verification. • The system provides a user interface to upload the necessary identity documents: The user uploads clear and valid images or scans of the selected identity document. The system may prompt the user to provide additional information or details. • The system processes the submitted documents: Validates the authenticity of the uploaded documents. Checks for completeness and clarity of the information provided. • If the submitted documents pass the initial validation: The system marks the identity verification process as pending. May request additional information or clarification from the user if needed. • The platform initiates a background check or verification process: Utilizes third-party verification services if applicable. Cross-references the provided information with existing databases. • The system updates the verification status: If the verification is successful, the user's account status is updated to "Verified." If the verification fails, the user is notified of the reason for failure and any actions needed.	
Alternative And Exceptional Flows: • If the submitted documents are unclear or incomplete: The system notifies the user and prompts them to re-upload or provide additional information. • If the background check results are inconclusive or require manual intervention: The platform may initiate a manual review process with involvement from the compliance team.	
Post-conditions: • Upon successful verification, the user's identity is confirmed, and their account is updated to a "Verified" status, allowing them to access additional features or higher transaction limits on the cryptocurrency exchange platform.	

5. Non-functional Requirements

5.1. Performance Requirements

Speed and Responsiveness:

- Requirement: The system should respond to user actions within milliseconds to provide real-time data updates and order execution.
- Benchmark: Achieve an average response time of under 200 milliseconds for critical functionalities such as order placement and market data retrieval.

Capacity and Throughput:

- Requirement: The platform should support a substantial number of transactions per second (TPS) to accommodate market demand.
- Benchmark: Maintain a throughput of at least 1,000 transactions per second during peak trading hours.

Reliability and Availability:

- Requirement: Ensure high system availability, minimizing downtime and service interruptions.
- Benchmark: Aim for 99.99% uptime (less than 1 hour of downtime annually) for system maintenance and updates.

Data Integrity and Security:

- Requirement: Guarantee the integrity and security of user data, transactions, and system operations.
- Benchmark: Implement encryption standards (e.g., SSL/TLS) for secure data transmission and storage, ensuring no data breaches or unauthorized access.

Fault Tolerance and Disaster Recovery:

- Requirement: Implement fault-tolerant measures and backup systems to ensure data integrity and continuity in case of system failures.
- Benchmark: Set up a disaster recovery plan to recover data and services within a specified timeframe (e.g., less than 4 hours) in case of a catastrophic failure.

5.2. Safety Requirements

Asset Protection and Security Measures:

- Requirement: Implement robust security measures to protect user accounts, funds, and sensitive data from unauthorized access or breaches.
- Safeguards: Utilize encryption protocols, multi-factor authentication, and secure storage mechanisms for private keys and user data.

Transaction Integrity and Verification:

- Requirement: Ensure transactional integrity and accuracy to prevent financial losses or unauthorized transactions.
- Safeguards: Implement transaction verification protocols, real-time monitoring, and confirmation mechanisms to prevent fraudulent activities.

User Account Safety:

- Requirement: Safeguard user accounts from unauthorized access, phishing attacks, or account takeovers.
- Safeguards: Educate users about security best practices, enable security features like email confirmations for critical actions, and regularly audit user sessions for anomalies.

Regulatory Compliance and Policies:

- Requirement: Adhere to relevant regulatory standards and industry best practices to ensure compliance and mitigate legal risks.
- Safeguards: Implement Know Your Customer (KYC) and Anti-Money Laundering (AML) procedures, adhere to data protection regulations (e.g., GDPR), and obtain necessary licenses or certifications in compliance with local financial laws.

Emergency Shutdown Procedures:

- Requirement: Establish emergency procedures to quickly halt trading or transactions in case of system vulnerabilities or security threats.
- Safeguards: Develop contingency plans and automated triggers to suspend operations, freeze withdrawals, or halt trading in emergencies.

Monitoring and Incident Response:

- Requirement: Implement continuous monitoring and incident response mechanisms to detect and respond to security breaches or anomalies promptly.
- Safeguards: Set up intrusion detection systems, conduct regular security audits, and maintain incident response teams for rapid mitigation of security incidents.

5.3. Security Requirements

User Authentication and Authorization:

- Requirement: Implement robust authentication mechanisms to verify user identities securely.
- Safeguards: Utilize multi-factor authentication, strong password policies, and token-based access controls. Implement role-based access control (RBAC) for authorization to ensure users access only permitted functionalities based on their roles.

Data Encryption and Storage:

- Requirement: Ensure sensitive data (like user credentials, private keys, and transaction details) is encrypted both in transit and at rest.
- Safeguards: Utilize strong encryption algorithms (e.g., AES) for data encryption. Store sensitive information in secure, encrypted databases and employ secure key management practices.

Transaction Integrity and Audit Trails:

- Requirement: Ensure the integrity of transactions and maintain detailed audit logs for traceability and accountability.
- Safeguards: Implement mechanisms to record and timestamp transactions securely. Maintain immutable logs for auditing and forensic analysis if required.

Security Testing and Incident Response:

- Requirement: Regularly conduct security assessments, penetration testing, and vulnerability scans.
- Safeguards: Establish an incident response plan, including procedures for detecting, reporting, and responding to security incidents promptly.

5.4. User Documentation

User Manuals:

Comprehensive guides explaining platform features, functionalities, and step-by-step instructions on how to use various aspects of the exchange, including account setup, trading, portfolio management, etc.

Online Help System:

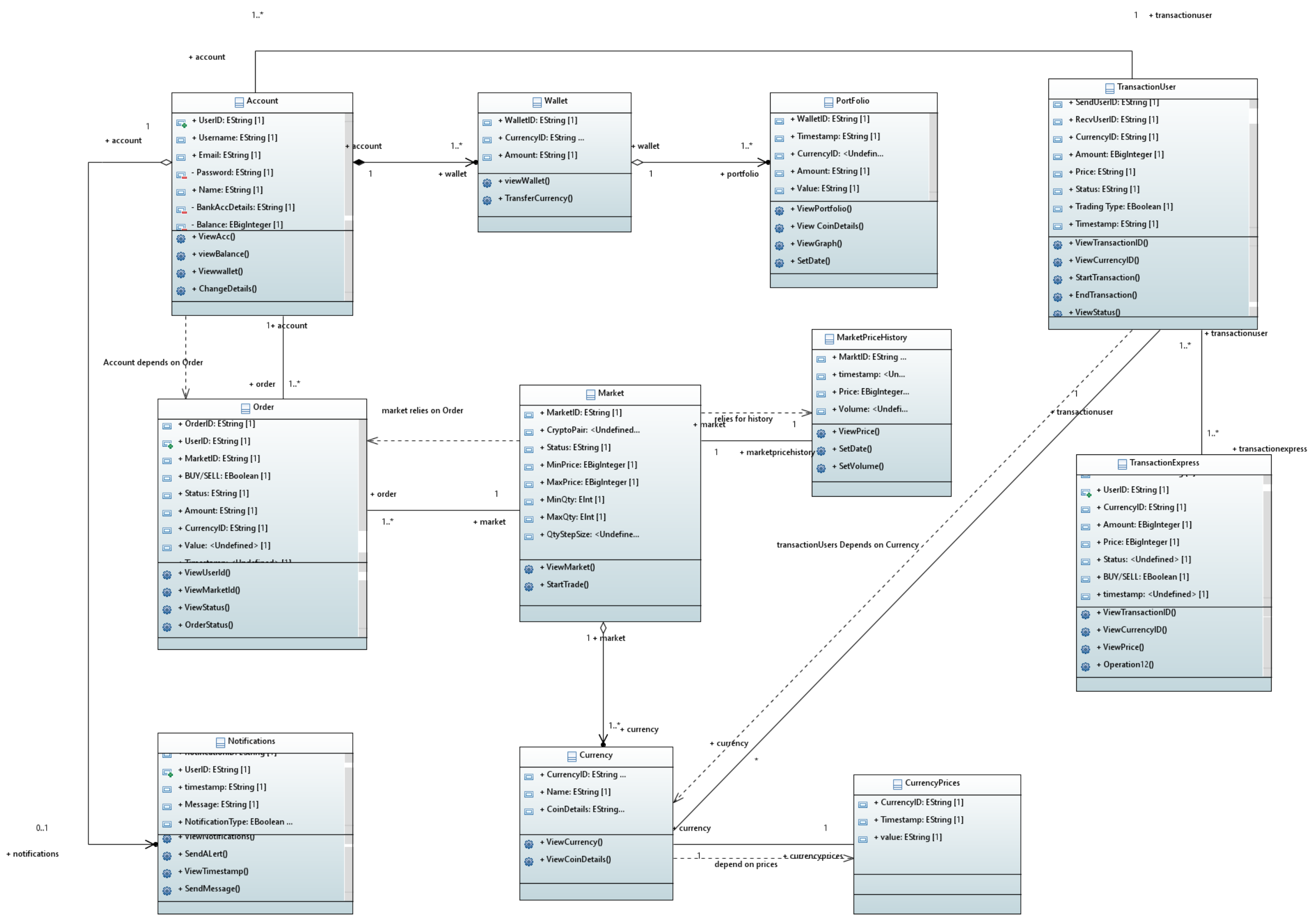
Interactive online help accessible within the platform, offering context-sensitive guidance. It provides immediate assistance related to the user's current task or query.

Tutorials and How-To Guides:

Step-by-step tutorials demonstrating specific processes or workflows within the platform, aiding users in understanding complex functionalities or new features.

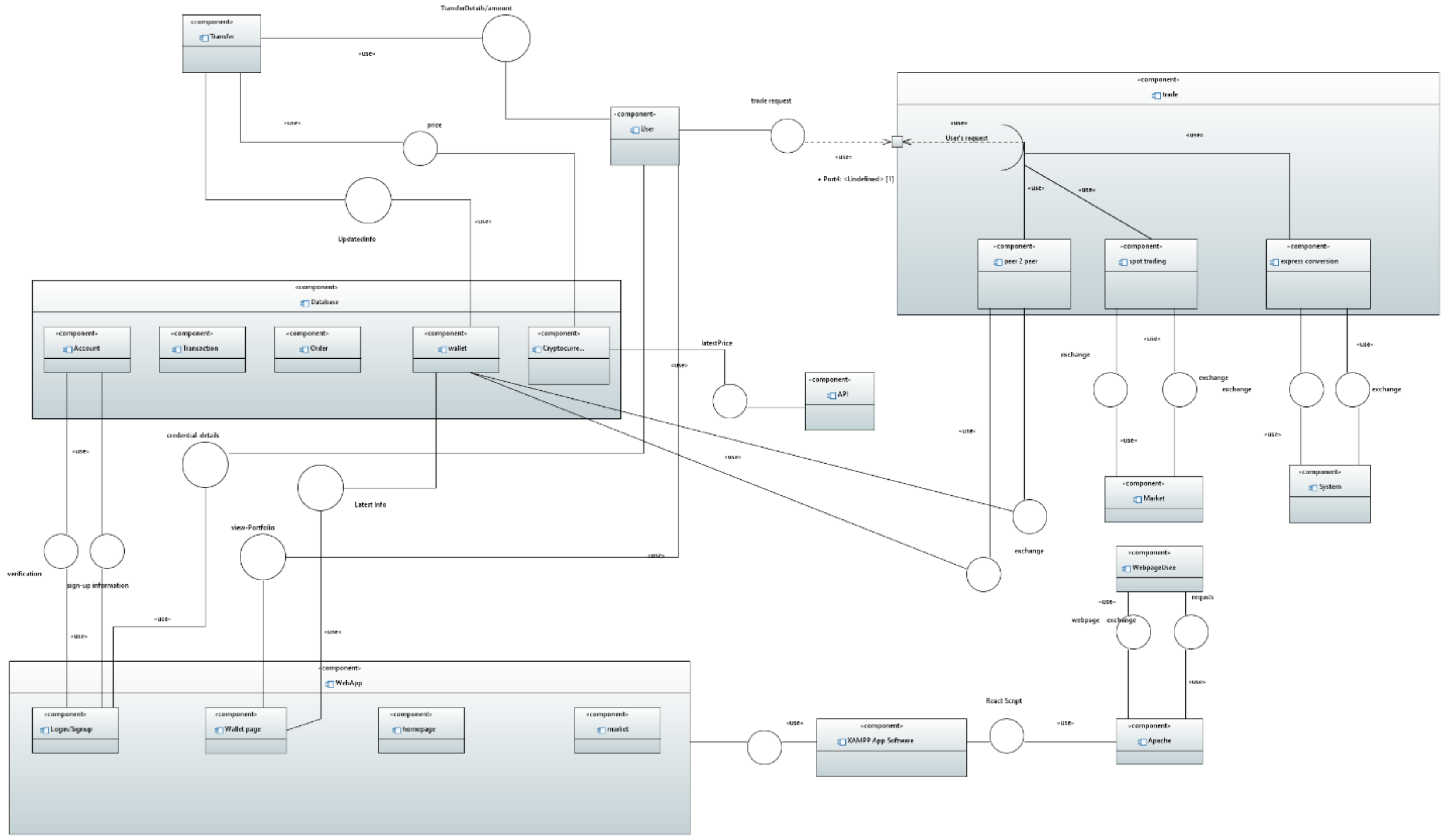
6. System Architecture

6.1. System Level Architecture

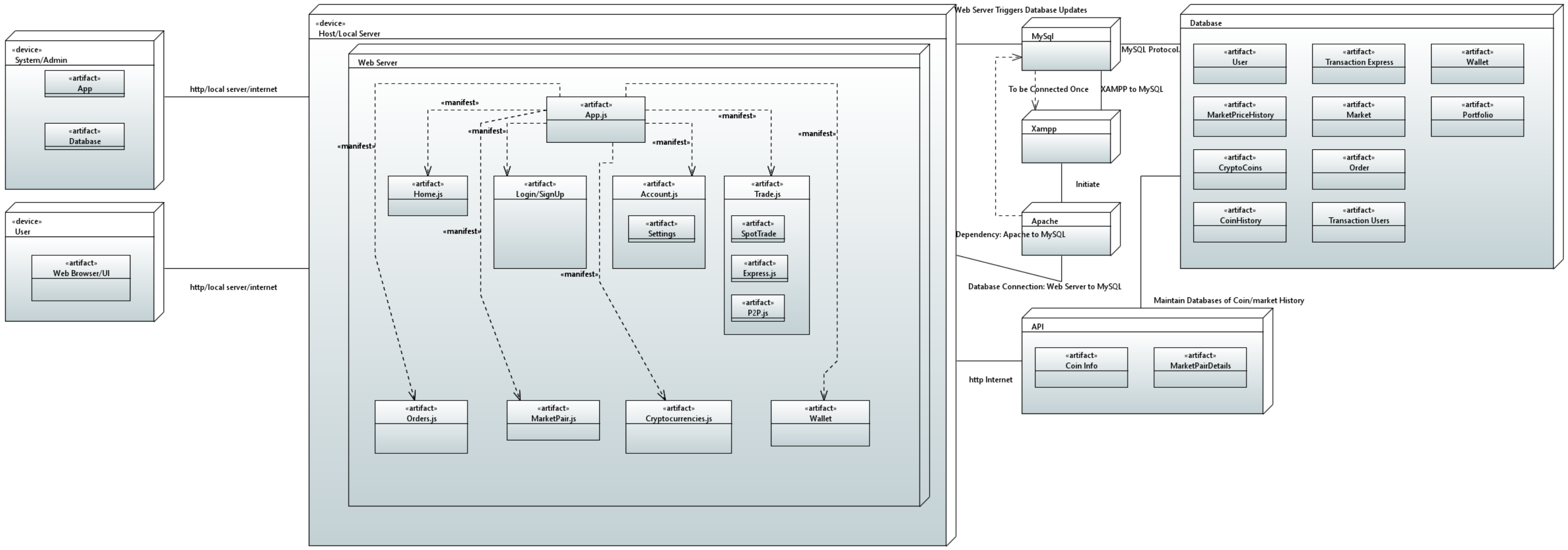


6.2. Software Architecture

6.2.1 Component Diagram:



6.2.2 Deployment Diagram:



7. Design Strategy

Future System Extension or Enhancement:

- The architecture is designed with modularity and scalability in mind to easily accommodate future extensions or enhancements to the Crypto trading app.
- Modular components allow for the seamless integration of new features without disrupting the existing system.

System Reuse:

- Leveraging industry-standard design patterns and frameworks to promote component reusability.
- Reusable modules are identified and encapsulated, facilitating efficient integration into different parts of the system.

User Interface Paradigms:

- Adhering to intuitive and user-friendly interface paradigms to enhance the overall user experience.
- Implementing responsive design principles to ensure accessibility across various devices and screen sizes.

Data Management (Storage, Distribution, Persistence):

- Employing a distributed and scalable database architecture to handle the storage and distribution of market data.
- Implementing data persistence mechanisms that ensure the security and integrity of user and transaction data.

Concurrency and Synchronization:

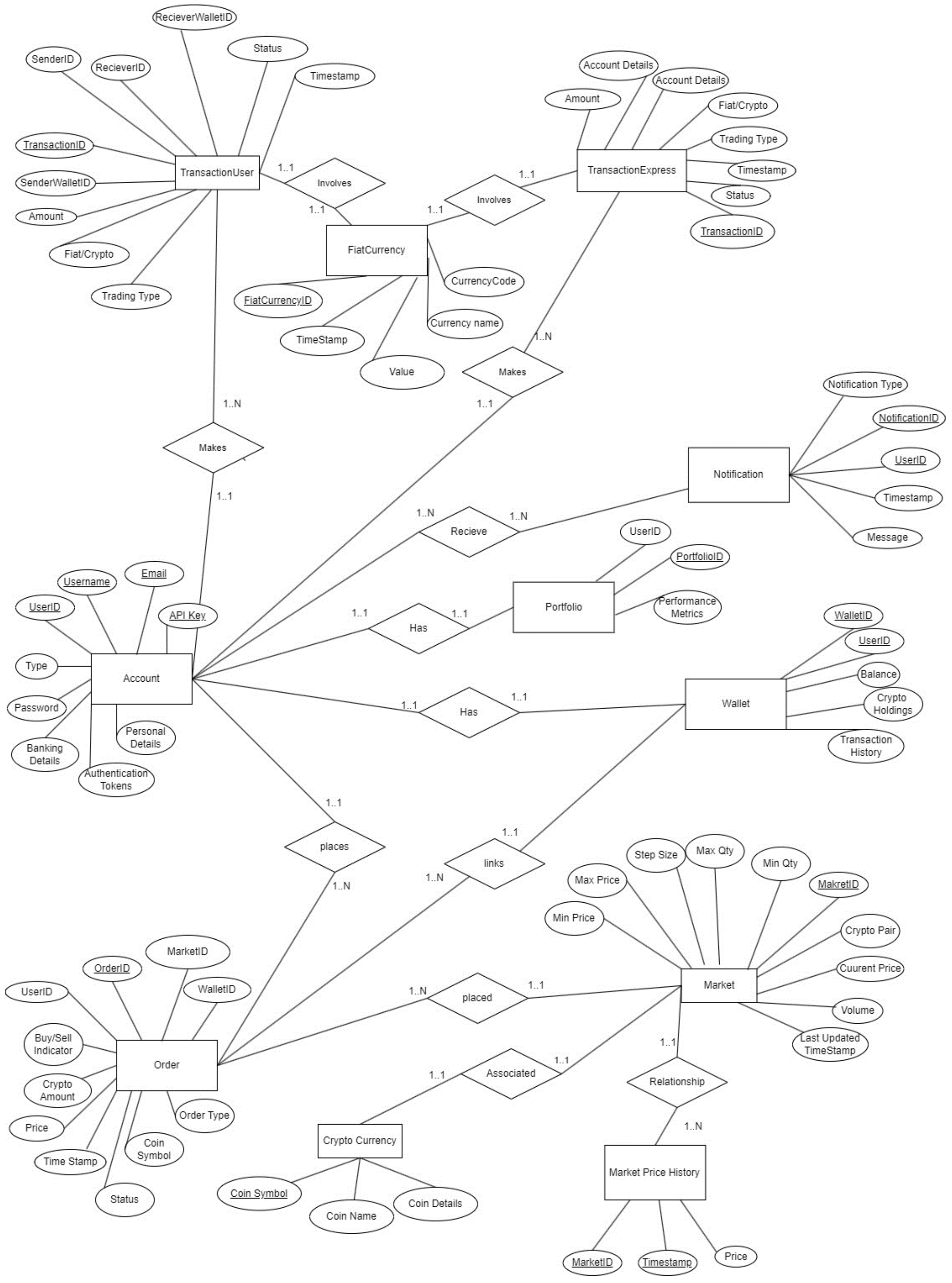
- Addressing the high concurrency demands of a trading platform through careful design of concurrency control mechanisms.
- Prioritizing data consistency and system stability through effective synchronization strategies.

Reasoning and Trade-offs:

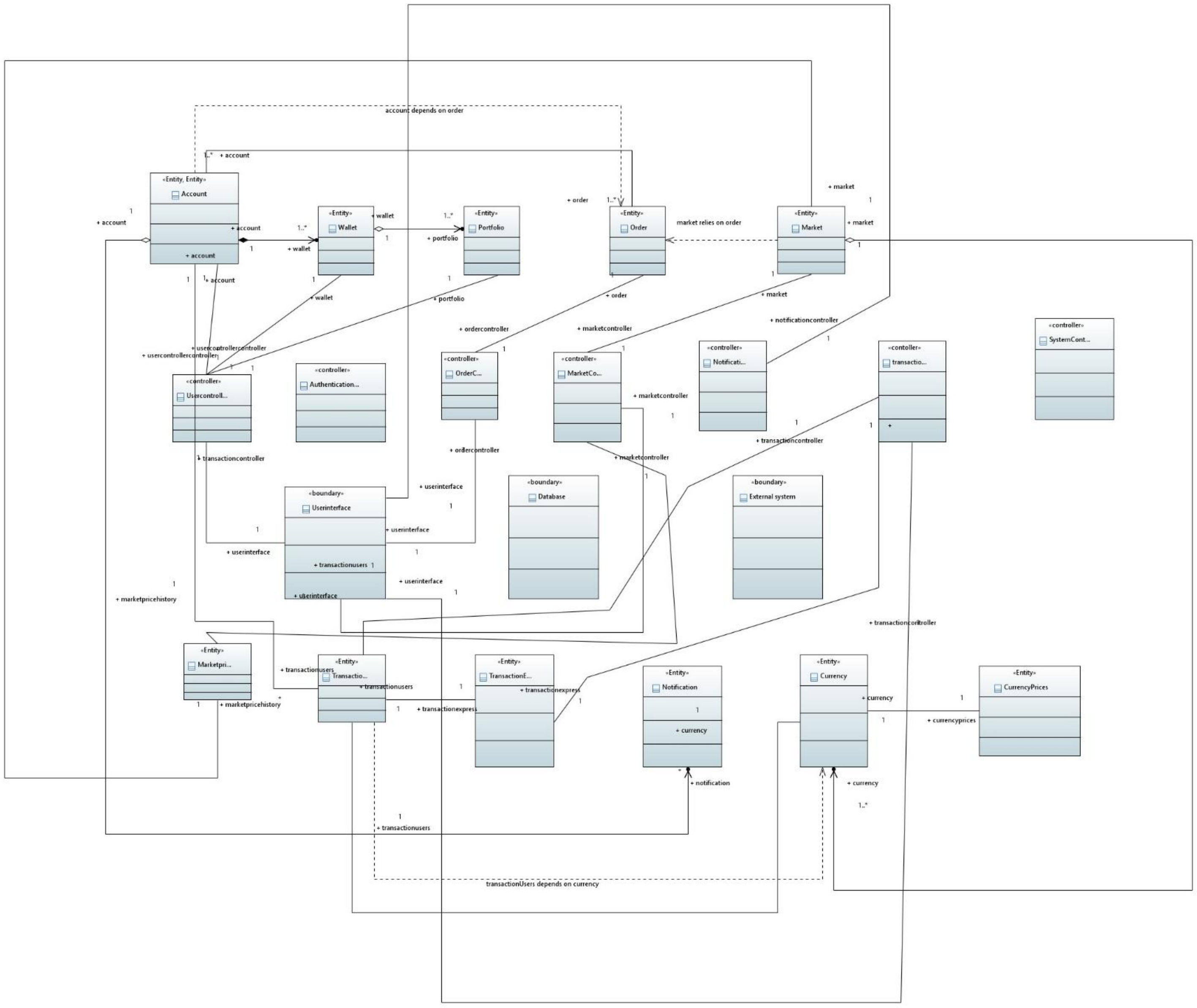
- Design decisions are guided by the principle of achieving a balance between performance and reliability.
- Trade-offs are made with a focus on optimizing system responsiveness while maintaining the security and integrity of user data.

8. Detailed System Design

8.1. Database Design ER Diagram



8.1.1 ECB Diagram (Optional)



8.2. Data Dictionary

8.2.1. Account

< Data 1>	
Name	Account
Alias	Write other names used for the first entry.
Where used/how used	This table stores information about user accounts. It is used for user authentication and tracking user-related data.
Content description	This table contains user-related information, including UserID, Username, Email, Password, Name, Bank Account Details, and the Balance derived from the Wallet.

Column Name	**Description**	**Type**	**Length**	**Nullable**	**Default Value**	**Key Type**
UserID	Unique identifier for a user	INT	-	NO	-	PK
Username	User's chosen username	VARCHAR	255	NO	-	-
Email	User's email address	VARCHAR	255	NO	-	-
Password	User's hashed password	VARCHAR	255	NO	-	-
Name	User's full name	VARCHAR	255	YES	NULL	-
BankDetails	Details of the user's bank	VARCHAR	255	YES	NULL	-
Balance	User's account balance Derived from Wallet	DECIMAL	10,2	YES	0.00	-

8.2.2

< Data 2>						
Name	Portfolio					
Alias	Write other names used for the first entry.					
Where used/how used	This table stores information about the portfolio of each user. It is used to track the historical data of a user's cryptocurrency holdings.					
Content description	The Portfolio table contains WalletID, Timestamp, CurrencyID, Amount, and Value.					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
WalletID	Unique identifier for a wallet	INT	-	NO	-	PK, FK
Timestamp	Timestamp of the portfolio entry	DATETIME	-	NO	CURRENT_TIME	PK
CurrencyID	Identifier for the currency	INT	-	NO	-	PK/FK
Amount	Amount of the currency in portfolio	DECIMAL	10,2	NO	0.00	-
Value	Value of the portfolio entry	DECIMAL	10,2	YES	NULL	-

8.2.3

< Data 3>						
Name	**Wallet**					
Alias	This table stores information about user wallets. It is used to track the amount of different currencies held by users.					
Where used/how used	This table stores information about user accounts. It is used for user authentication and tracking user-related data.					
Content description	The Wallet table contains WalletID, CurrencyID, and Amount.					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
WalletID	Unique identifier for a wallet	INT	-	NO	-	PK/FK
CurrencyID	Identifier for the currency	INT	-	NO	-	PK/FK
Amount	Amount of the currency in wallet	DECIMAL	10,2	NO	0.00	-

8.2.4

< Data 4>	
Name	Order
Alias	Write other names used for the first entry.
Where used/how used	This table stores information about user orders in the cryptocurrency market. It is used to track buy and sell orders.
Content description	The Orders table contains OrderID, UserID, MarketID, Buy/Sell Indicator, Status, OrderType, CurrencyID, Amount, Value, and Timestamp.

Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
OrderID	Unique identifier for an order	INT	-	NO	-	PK
UserID	User placing the order	INT	-	NO	-	FK
MarketID	Identifier for the market	INT	-	NO	-	FK
Buy/Sell	Indicator for buy or sell	VARCHAR	4	NO	-	-
Status	Status of the order	VARCHAR	20	NO	-	-
OrderType	Type of the order (e.g., limit, market)	VARCHAR	20	NO	-	-
CurrencyID	Identifier for the currency	INT	-	NO	-	FK
Amount	Amount of the currency in the order	DECIMAL	10,2	NO	0.00	-
Value	Value of the order	DECIMAL	10,2	YES	NULL	-
Timestamp	Timestamp of the order	DATETIME	-	NO	CURRENT_TIME	-

8.2.5

< Data 5>	
Name	Market
Alias	Write other names used for the first entry.
Where used/how used	This table stores information about cryptocurrency markets. It is used to track market details such as status, price limits, and quantity limits.
Content description	The Market table contains MarketID, Crypto Pair, Status (Active/Dead), Min Price, Max Price, Min Qty, Max Qty, and Qty Step Size. Will get essential for each market pair .

Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
MarketID	Unique identifier for a market	INT	-	NO	-	PK
CryptoPair	Pair of cryptocurrencies traded	VARCHAR	50	NO	-	-
Status	Status of the market (Active/Dead)	VARCHAR	10	NO	Active	-
MinPrice	Minimum price for the market	DECIMAL	10,2	YES	NULL	-
MaxPrice	Maximum price for the market	DECIMAL	10,2	YES	NULL	-
MinQty	Minimum quantity for the market	DECIMAL	10,2	YES	NULL	-
MaxQty	Maximum quantity for the market	DECIMAL	10,2	YES	NULL	-
QtyStepSize	Step size for quantity changes	DECIMAL	10,2	YES	NULL	-

8.2.6

< Data 6>	
Name	MarketPriceHistory
Alias	Write other names used for the first entry.
Content description	This table stores historical price and volume data for each market. It is used to track the price and volume changes over time.
Where-used/howused	For the user to analyze which market pair has best chance of making them a profit.

Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
MarketID	Identifier for the market	INT	-	NO	-	PK/FK
Timestamp	Timestamp of the price/volume entry	DATETIME	-	NO	CURRENT_TIME	-
Price	Price of the market at the given time	DECIMAL	10,2	NO	0.00	-

8.2.7

< Data 7>	
Name	TransactionUsers
Alias	Write other names used for the first entry.
Where used/how used	This table stores information about user transactions, including transfers and trades. It is used to track the movement of currencies between users.
Content description	The TransactionUsers table contains TransactionID, SenderUserID, ReceiverUserID, CurrencyID, Amount, Price, Status, Trading Type (Transfer/Trade), and Timestamp..

Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
TransactionID	Unique identifier for a transaction	INT	-	NO	-	PK
SenderUserID	User initiating the transaction	INT	-	NO	-	FK
ReceiverUserID	User receiving the transaction	INT	-	NO	-	FK
CurrencyID	Identifier for the currency	INT	-	NO	-	FK
Amount	Amount of the currency in the transaction	DECIMAL	10,2	NO	0.00	-
Price	Price per unit currency in the transaction	DECIMAL	10,2	YES	NULL	-
Status	Status of the transaction	VARCHAR	20	NO	-	-
TradingType	Type of transaction (Transfer/Trade)	VARCHAR	20	NO	-	-
Timestamp	Timestamp of the transaction	DATETIME	-	NO	CURRENT_TIME	PK

8.2.8

< Data 8>	
Name	TransactionExpress
Alias	Write other names used for the first entry.
Where used/howused	This table stores information about user transactions, including transfers and trades. It is used to track the movement of currencies between users.
Content description	The TransactionExpress table contains TransactionID, UserID, CurrencyID, Amount, Price, Status, Buy/Sell Indicator, and Timestamp.

Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
TransactionID	Unique identifier for the transaction	INTEGER	-	NO	-	PK
UserID	User identifier associated with the transaction	INTEGER	-	NO	-	FK
CurrencyID	Identifier for the currency involved	INTEGER	-	NO	-	FK
Amount	Amount of the transaction	DECIMAL	-	NO	-	-
Price	Price of the transaction	DECIMAL	-	NO	-	-
Status	Status of the transaction (e.g., completed, pending)	VARCHAR	50	NO	-	-
Buy/Sell Indicator	Indicator for Buy or Sell action	VARCHAR	10	NO	-	-
Timestamp	Timestamp of the transaction	TIMESTAMP	-	NO	-	PK

8.2.9

< Data 9>						
Name	Notification					
Alias	Write other names used for the first entry.					
Where-used/howused	This table stores information about user transactions, including transfers and trades. It is used to trackthe past history of the user using notifications.					
Content description	The TransactionExpress table contains TransactionID, UserID, CurrencyID, Amount, Price, Status, Buy/Sell Indicator, and Timestamp.					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
NotificationID	Unique identifier for the notification	INTEGER	-	NO	-	PK
UserID	User identifier receiving the notification	INTEGER	-	NO	-	PK
Timestamp	Timestamp of the notification	TIMESTAMP	-	NO	-	PK
Message	Message content of the notification	TEXT	-	YES	-	-
Notification Type	Type of notification (e.g., email, alert)	VARCHAR	50	NO	-	-

8.2.10

< Data 10>						
Name	Currency					
Alias	Write other names used for the first entry.					
Where-used/howused	Processes involving currency data, this data is received from the API.					
Content description	The Currency that is used in trading / displaying charts and much more.					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
CurrencyID	Unique identifier for the currency	INTEGER	-	NO	-	PK
Name/Symbol	Name of the currency	VARCHAR	20	NO	-	-
CoinDetails	Details about the currency (e.g., BTC, DOGE)	TEXT	-	YES	-	-

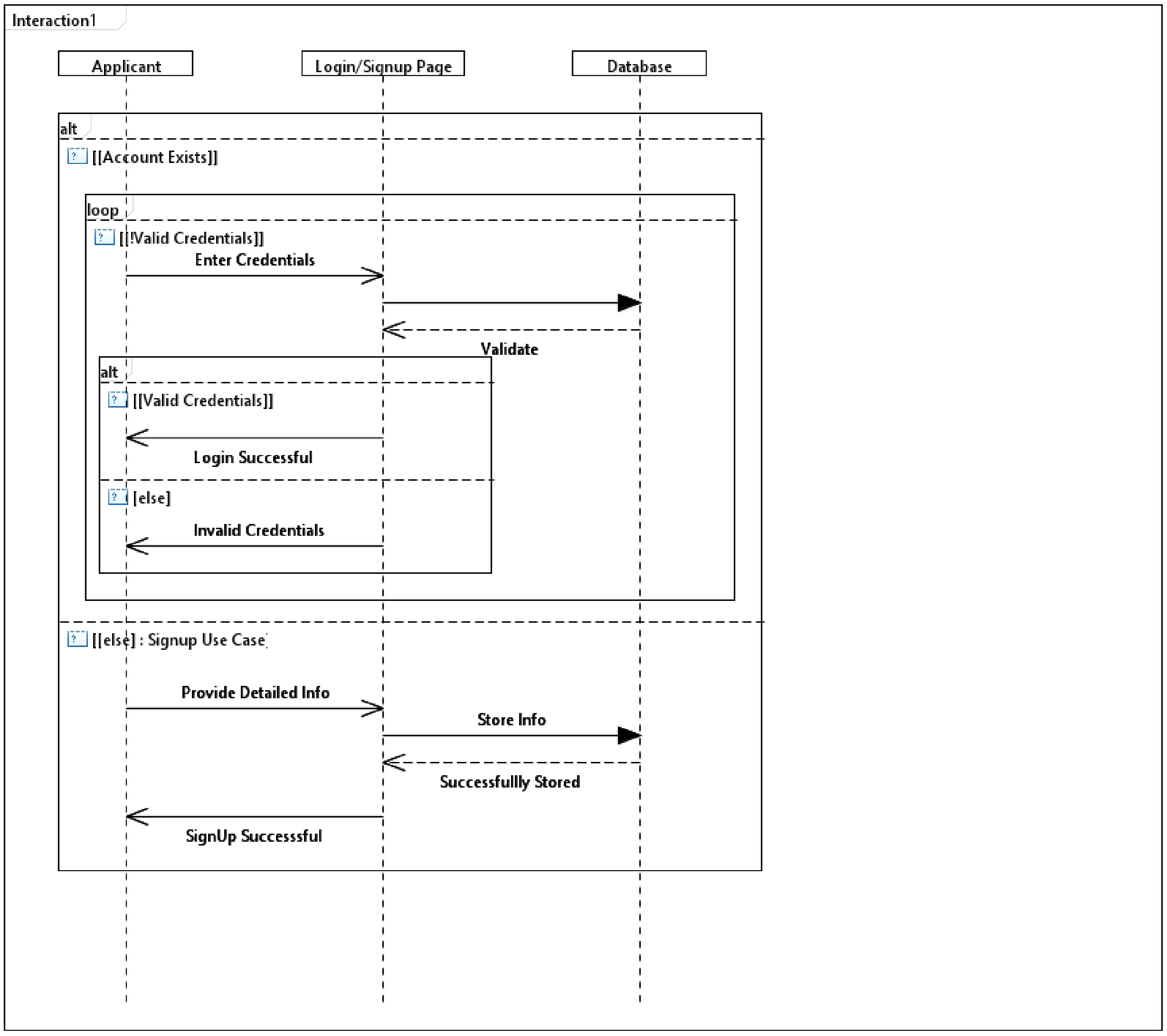
8.2.11

< Data 11>	
Name	CurrencyPricesHistory
Alias	Write other names used for the first entry.
Where-used/howused	Processes involving currency data at different time stamps, this data is received from the API.
Content description	The Currency that is used in trading / displaying charts and much more. Users are also able to know by the past history to know which currency to invest/sell.

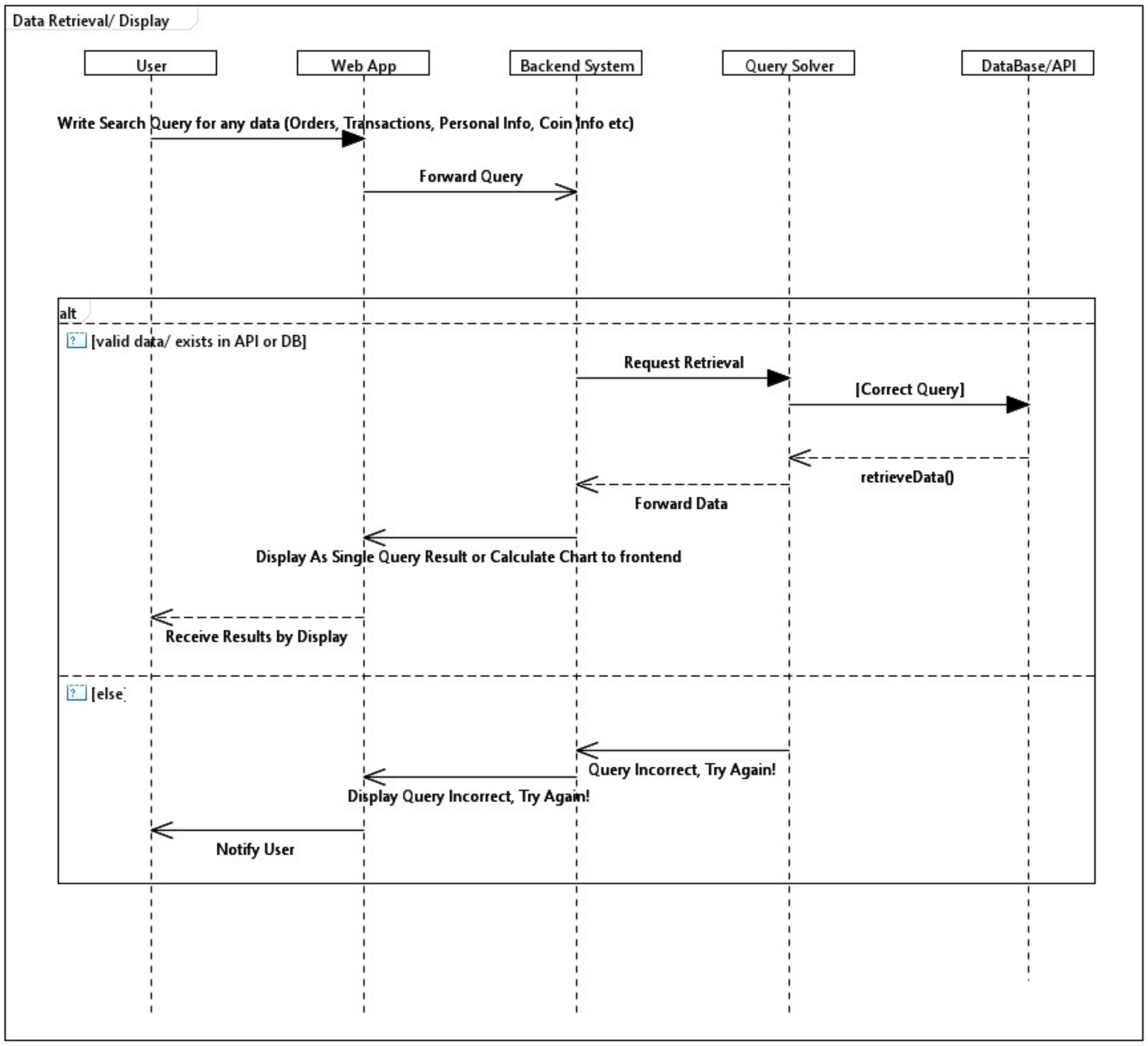
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
CurrencyID	Identifier for the currency	INTEGER	-	NO	-	PK/FK
Timestamp	Timestamp of the currency price	TIMESTAMP	-	NO	-	PK
Value	Value of the currency at the specified timestamp	DECIMAL	-	NO	-	-

9. Application Design

9.1.2. Login/Signup Sequence Diagram

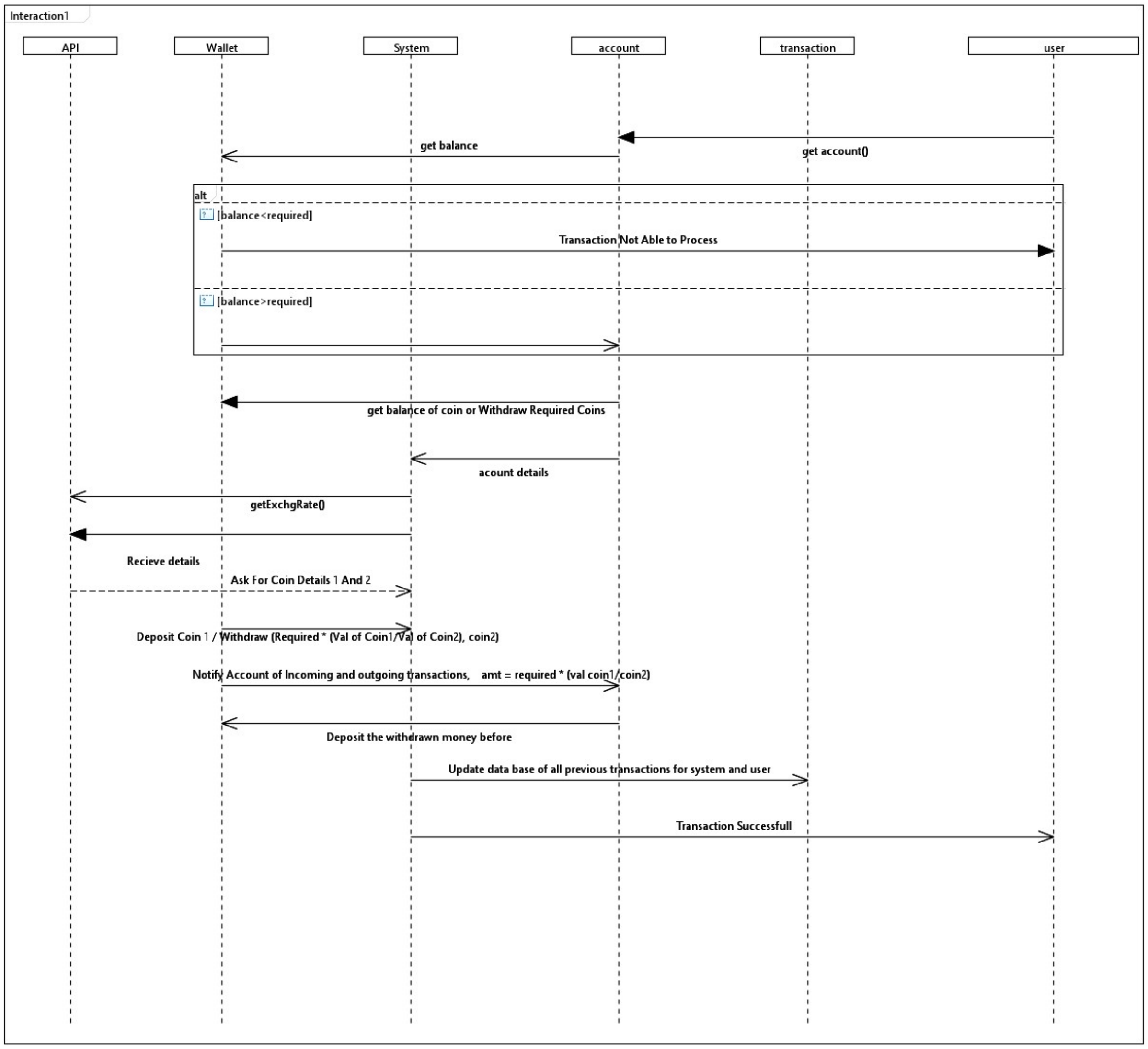


9.1.2.1 Data Retrieval Sequence Diagram


Data Retrieval/ Display
User
Web App
Backend System
Query Solver
DataBase/API
Write Search Query for any data (Orders, Transactions, Personal Info, Coin Info etc)
Forward Query
alt
[valid data/ exists in API or DB]
Request Retrieval
[Correct Query]
retrieveData()
Forward Data
Display As Single Query Result or Calculate Chart to frontend
Receive Results by Display
[else]
Query Incorrect, Try Again!
Display Query Incorrect, Try Again!
Notify User

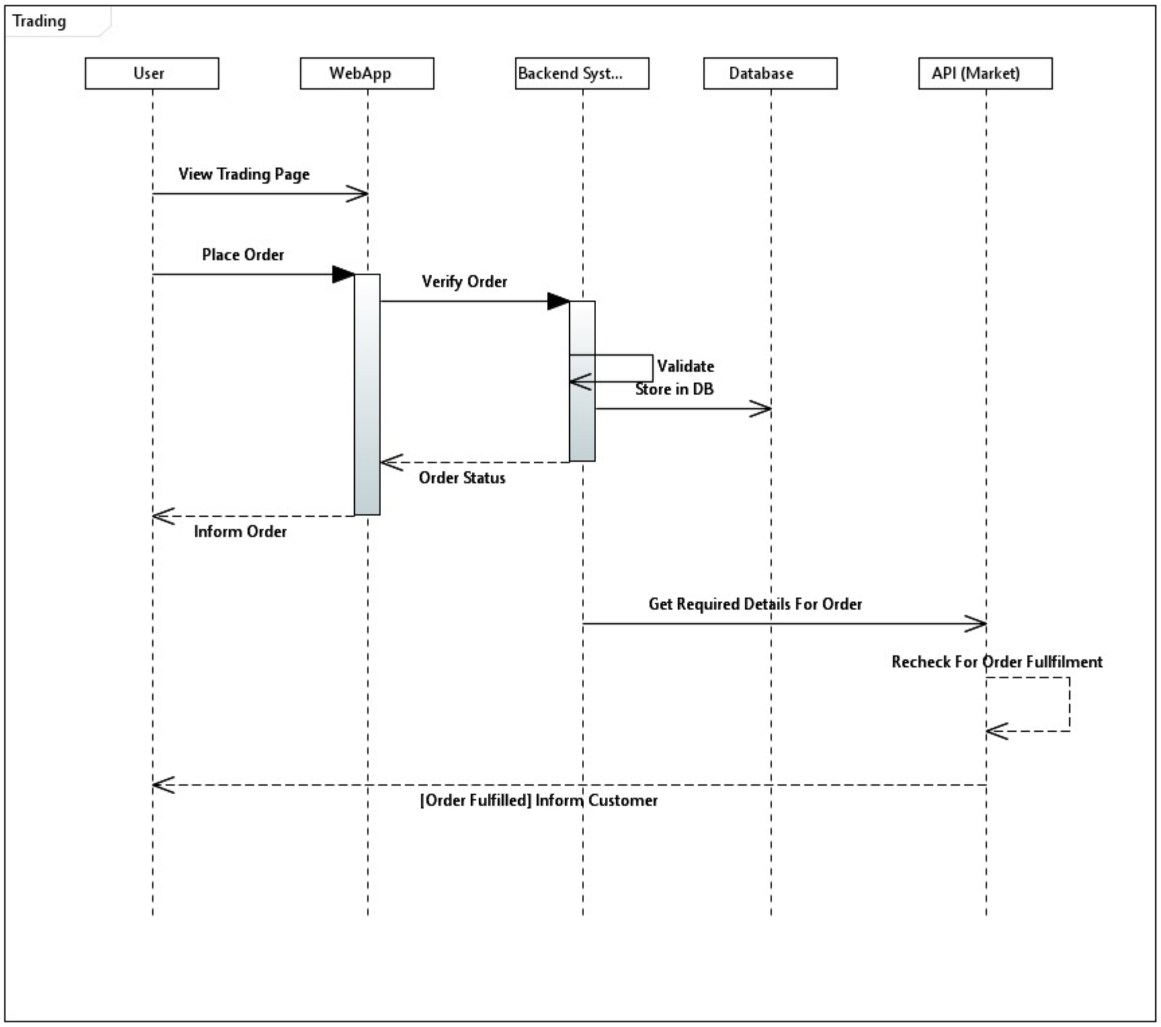
Data retrieval sequence diagram depicts the interactions and sequence of steps involved when a system or component requests and retrieves specific information from a database/API. It serves as a visual aid for understanding the process of fetching data and facilitates communication about data retrieval procedures among stakeholders.

9.1.2.2 Transfer Process Sequence Diagram


Interaction1
API
Wallet
System
account
transaction
user
get account()
get balance
alt
[balance<required]
Transaction Not Able to Process
[balance>required]
get balance of coin or Withdraw Required Coins
acount details
getExchgRate()
Recieve details
Ask For Coin Details 1 And 2
Deposit Coin 1 / Withdraw (Required * (Val of Coin1/Val of Coin2), coin2)
Notify Account of Incoming and outgoing transactions, amt = required * (val coin1/coin2)
Deposit the withdrawn money before
Update data base of all previous transactions for system and user
Transaction Successfull

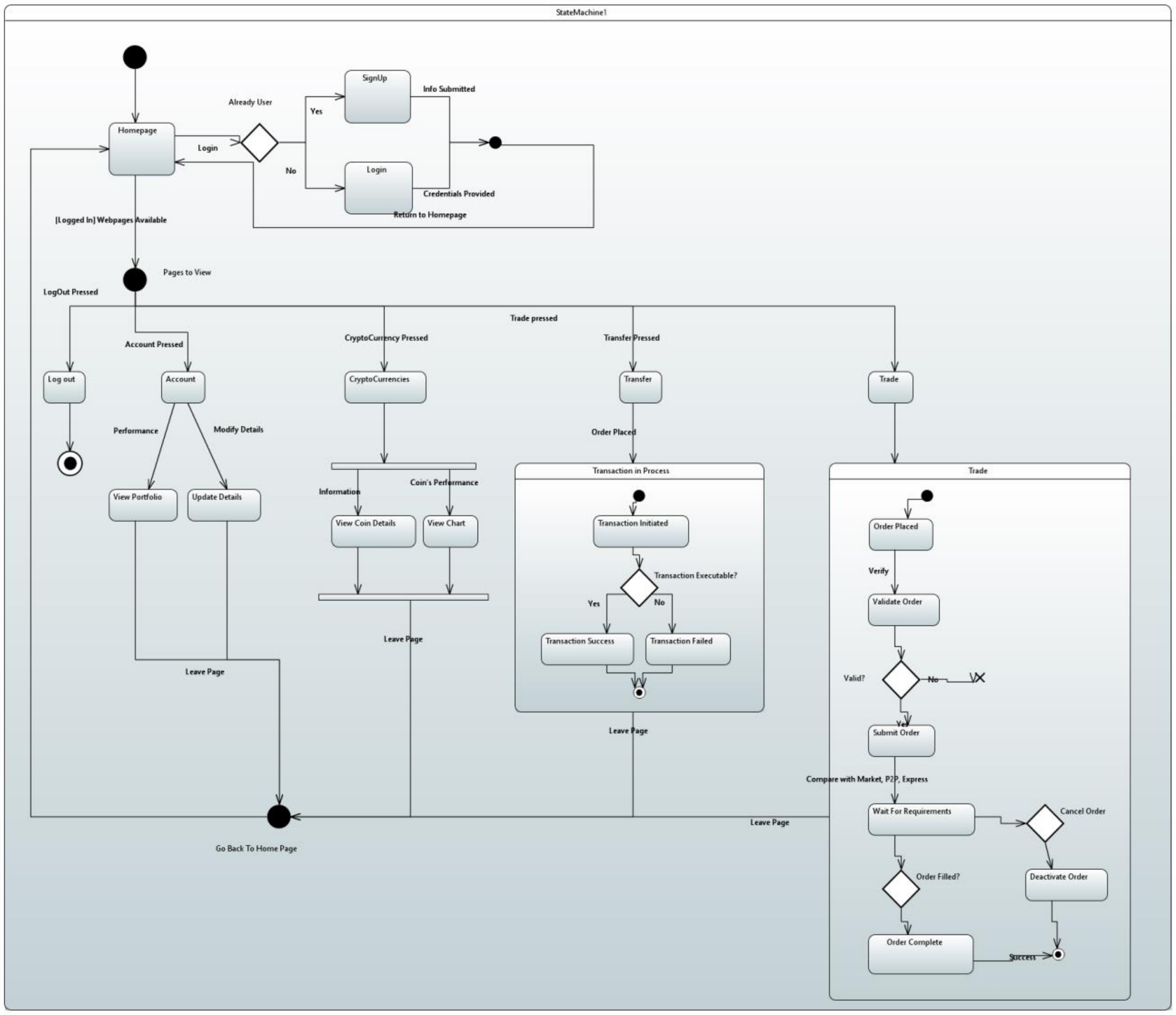
.
Transfer process sequence diagram visually represents the interactions and sequence of steps involved when initiating and completing a fund or asset transfer within a system. It serves as a tool for understanding, communicating, and analyzing the process of transferring funds/assets among different entities or accounts.

9.1.2.3 <Sequence Diagram n>



In the crypto trading platform, the user experience is seamlessly orchestrated through a dynamic trading sequence that encompasses Spot Trading, Express Trading, and Peer-to-Peer (P2P) Trading functionalities. In Spot Trading, users initiate buy or sell orders, and the system processes transactions in real-time, ensuring efficient order matching and confirmation. Express Trading streamlines the process for quick transactions, as users specify the cryptocurrency and amount, and the system rapidly matches them with suitable counterparts, simplifying the overall trading flow. Peer-to-Peer (P2P) Trading introduces a decentralized dimension, allowing users to directly engage in transactions. The buyer expresses an intent to purchase a specific cryptocurrency amount, and the seller offers to sell, with the system facilitating negotiation, agreement, and trade execution. This integrated trading experience ensures versatility, speed, and flexibility for users, catering to diverse trading preferences within a secure and user-friendly environment.

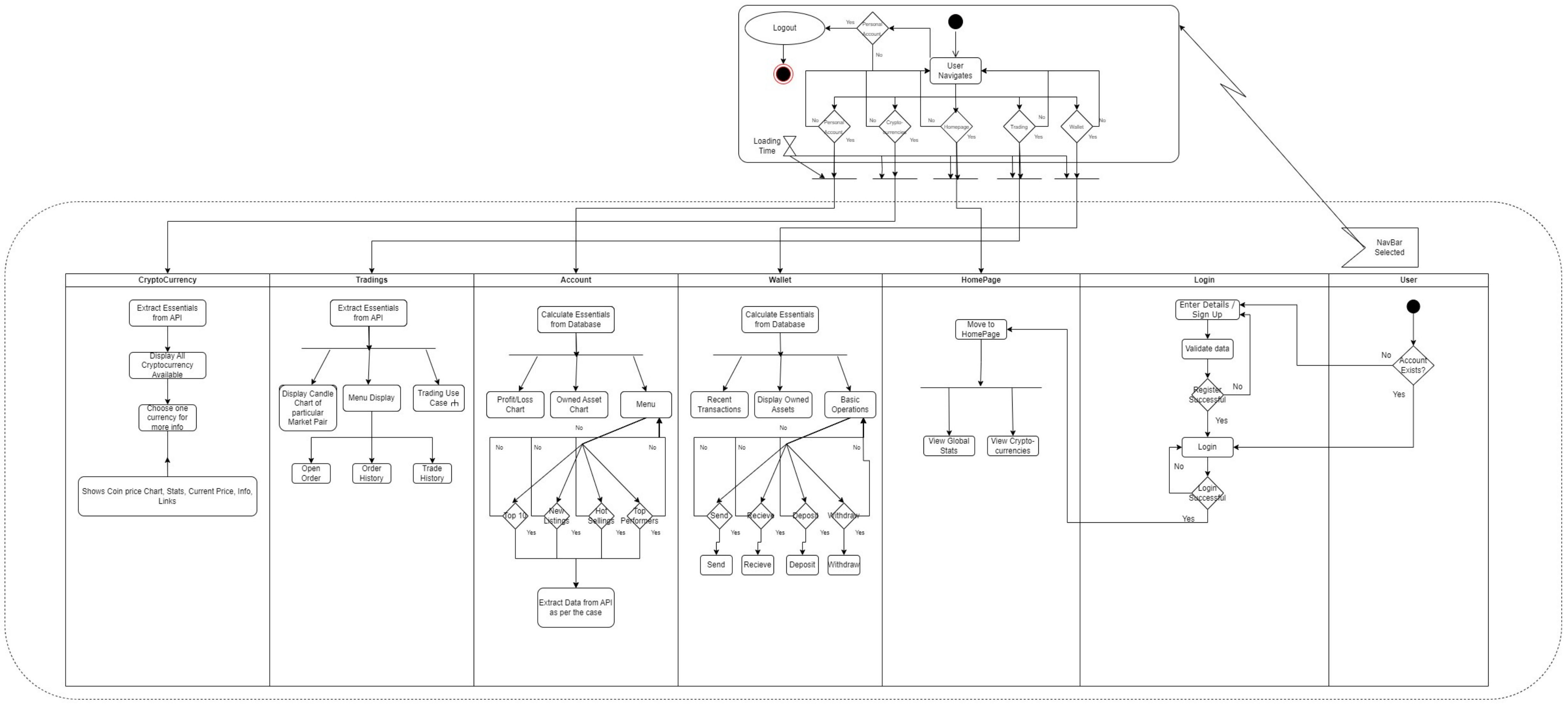
9.1.3. State Diagram



9.1.4. Activity Diagram

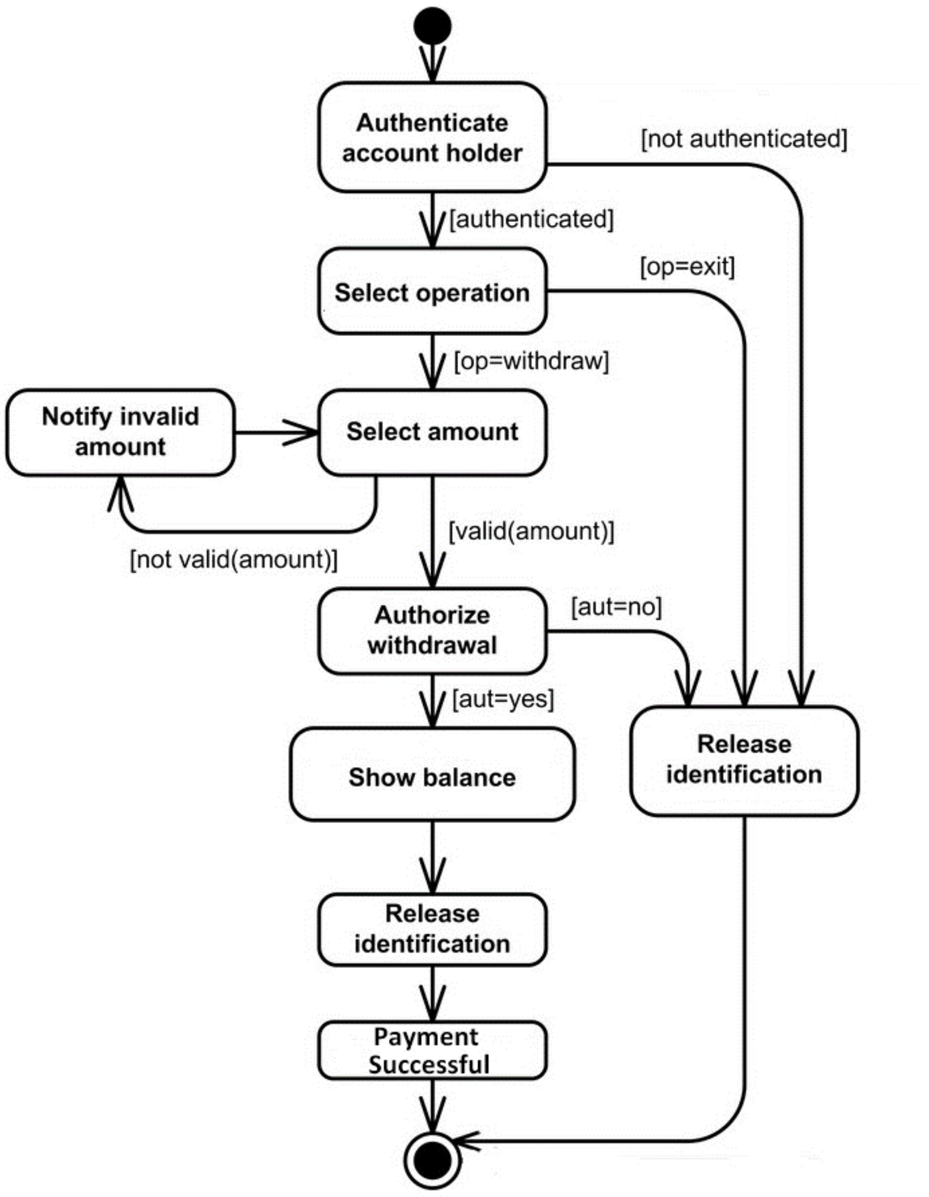
9.1.4.1 Base Activity

This Base diagram shows the flow of activity when the User's goes live on website , he can navigates through many tabs like transfer, home , wallet , account , cryptocurrency.



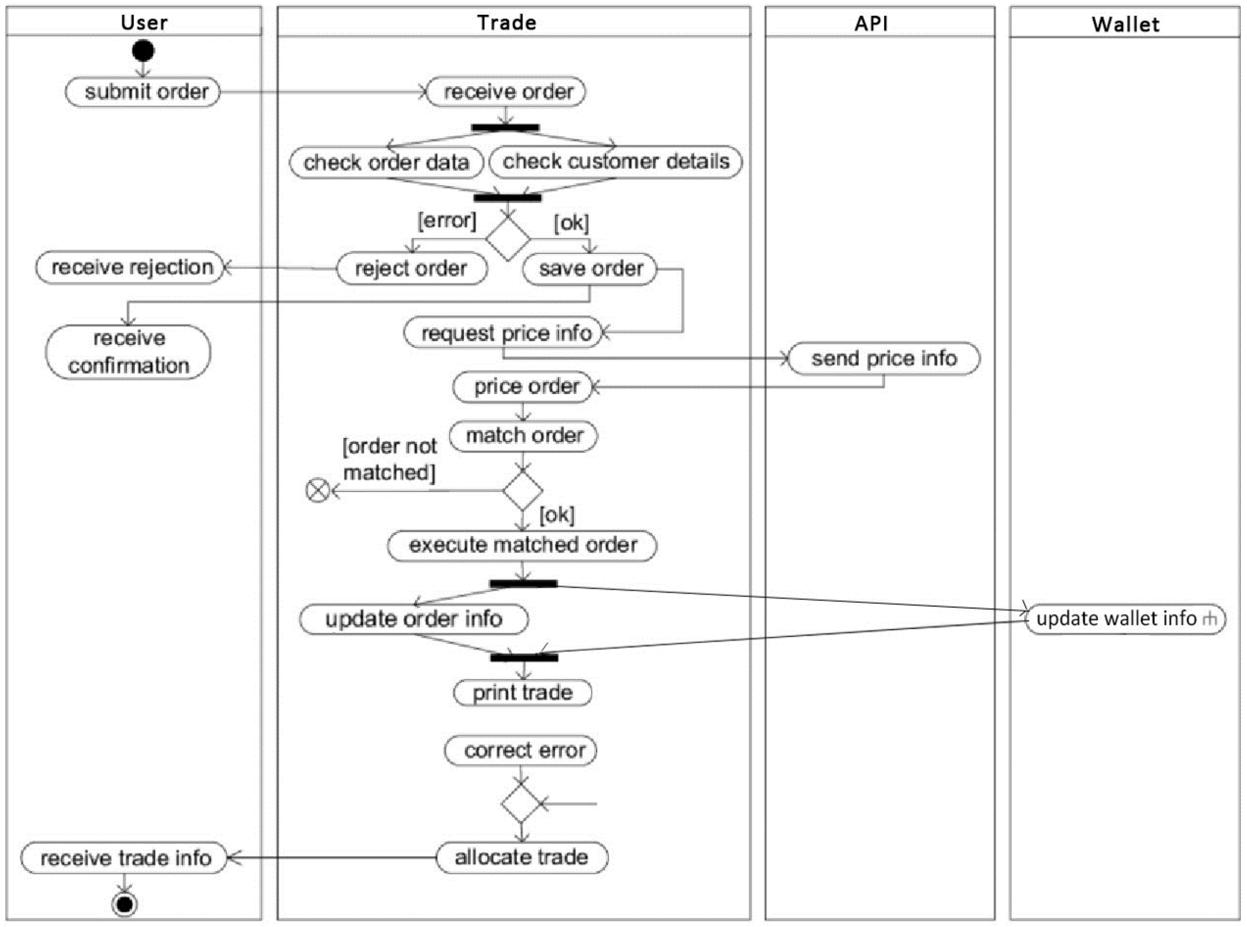
9.1.4.2 Transaction Activity diagram

A transaction activity diagram visually represents the sequence of actions, interactions, and steps within a transactional process. It illustrates the flow of activities between entities, showcasing how transactions occur and how different components interact within a system. This diagram helps in understanding, analyzing, and optimizing transactional processes by presenting a clear visual depiction of the sequence of events, decision points, and completion of transactions. It serves as a valuable communication tool, aiding in conveying complex transactional processes to stakeholders and identifying areas for improvement or optimization within the system.



9.1.4.3 Trade Activity Diagram

A trade activity diagram visually outlines the sequence of actions and interactions within a trading process. It illustrates how different entities interact during a trade, depicting the steps involved in initiating, executing, and completing a trade. This diagram aids in understanding and analyzing trade processes, facilitating communication about trade-related activities among stakeholders, and identifying areas for process improvement or optimization within the system.



10. References

Python-Binance Documentation

- Author: Binance Devs
- Date: Updated 11th Aug 2023
- Title: Python-Binance Documentation
- URL: https://python-binance.readthedocs.io/en/latest/index.html

Coinranking API Details on RapidAPI

- Author: Coinranking
- Date: Updated 8 months ago (As of 4th Dec 2023)
- Title: Coinranking API Details
- URL: https://rapidapi.com/Coinranking/api/coinranking1/details

YouTube Video: "Build and Deploy a React Cryptocurrency App"

- Author: JavaScript Mastery
- Date: Sep 18, 2021
- Title: Build and Deploy a React Cryptocurrency App and Master Redux Toolkit in One Video
- URL: https://www.youtube.com/watch?v=9DDX3US3kss

Inspiration from Binance

- Source: Binance (Cryptocurrency Exchange)
- Date: July 15, 2017
- Context: Design and functionality inspiration

11. Appendices

Term	Definition
Crypto Trading Website	An online platform that facilitates the buying and selling of cryptocurrencies.
Data Encryption	The process of encoding data to ensure the confidentiality and integrity of sensitive information.
Use Case Diagram	A visual representation illustrating how different actors interact with the crypto trading website to achieve specific goals.
Activity Diagram	A graphical representation of the flow of data within the system, showing processes, data stores, and data flow.
System Architecture	The overall structure of the crypto trading website, including components, modules, and their interactions.
Database Schema	The structure of the database, specifying tables, relationships, and constraints.
Third-Party Integrations	External services and APIs integrated into the crypto trading website to enhance functionality.
Security Protocols	Measures implemented to protect user data, transactions, and the overall system.
Performance Requirements	Expectations regarding system response times, scalability, and concurrent user support.
Other Diagrams such as sequence, deployment, component, class diagram	Detailed scenarios and steps to validate the functionality and performance of the crypto trading website.