



SHAHEED ZULFIKAR ALI BHUTTO
INSTITUTE OF SCIENCE AND TECHNOLOGY

Final Year Project Report

ZABx Blockchain, Coin and Exchange

Project Team:

Mohammad Areeb Faisal 1912231

Fatima Mohiuddin 1912224

8/1/2023

Project Supervisor:

Dr Imran Amin

Submitted in the partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer

Science in the

Faculty of Computing and Engineering Sciences

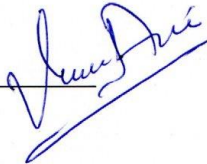
PLAGIARISM FREE CERTIFICATE

This is to certify that we, **Mohammad Areeb Faisal** S/O Faisal Rasheed and **Fatima Mohiuddin** D/O Mohiuddin Ahmed Ansari, are the members of FYP group **ZABx Blockchain, Coin and Exchange** under registration numbers **1912231** and **1912224** respectively, at the Department of Computer Science at SZABIST, Karachi. We certify that our FYP documentation has been reviewed by our advisor and the work presented is our own.

Name of Advisor: **Dr. Imran Amin**

Designation: Head of Computer Science

Signature: _____



Declaration of Authorship

We, Mohammad Areeb Faisal (1912231) and Fatima Mohiuddin (1912224), declare that this report titled, "ZABx Blockchain, Coin and Exchange" and the work presented in it are our own. We confirm that:

This work was done wholly or mainly while in candidature for a bachelor's degree at this University.

Where any part of this report has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated. Where we have consulted the published work of others, this is always clearly attributed.

Where we have quoted from the work from others, the source is always given. With the exceptions of quotations, this report is entirely our work. We have acknowledged all main sources of help.

Where the report is based on work done by ourselves jointly with others, we have made clear exactly what was done by others and what we have contributed ourselves.

Signed:

Mohammad Areeb Faisal 1912231
Fatima Mohiuddin 1912224

Date: 1st August 2023

Project Description

The ZABx Blockchain, Coin, and Exchange is a ground-breaking platform created to empower SZABIST students and other developers through providing them access to a decentralized environment for deploying their dApps (decentralized applications) as virtual or crypto assets. This cutting-edge blockchain infrastructure is accompanied by a native cryptocurrency known as "ZABx coin," as well as a user-friendly exchange that enables smooth buying, selling, and earning of other cryptocurrencies, including ZABx coin.

The fundamental goal of ZABx Blockchain, Coin, and Exchange is to build a thriving ecosystem where SZABIST students and developers could successfully and securely deploy their dApps. The platform provides a specialized environment for developing virtual assets and smart contracts with the goal of promoting innovation, cooperation, and learning within the blockchain community.

A comprehensive platform called ZABx Blockchain, Coin, and Exchange enables the implementation of dApps on the blockchain, acting as virtual or crypto assets, empowering SZABIST students and developers. The platform uses ZABx coin, the ecosystem's native cryptocurrency, as the main medium of exchange for transactions, offering a user-friendly exchange for simple trading of different cryptocurrencies. Through dApp interactions, task completion, and network stability contributions, users can gain ZABx coin. ZABx Blockchain, Coin, and Exchange possess a unique business model with potential revenue streams. Partnerships with real vendors enable the platform to offer exciting rewards and incentives sponsored by these vendors.

Keywords: challenges, dApps, crypto assets, ZABx coin, exchange, virtual assets, blockchain ecosystem, innovation.

Acknowledgement

In the name of ALLAH the most beneficent and merciful who gave us the knowledge and courage to work on this research area.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of our project.

We would first like to thank our supervisor Dr. Imran Amin of the Computer Science faculty at Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology. The door to his office was always open whenever we ran into a trouble spot or had a question about our research or writing. He consistently helped, cooperated and motivated us throughout the research.

We would like to thank to our teachers who guided us in the light of their knowledge and experience. We would also like to express our gratitude to our loving parents and family members who helped and gave us encouragement. Furthermore, we would also like to acknowledge with much appreciation the crucial role of the staff of SZABIST, who gave the permission to use all required equipment and the necessary materials to complete the project.

At the end, we would like to thank Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology for providing us with such an inspiring environment. The quality education, the cooperative faculty members and the challenging environment have always motivated and boosted the confidence level of each and every student who has been a part of Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology.

Table of Contents

Declaration of Authorship	3
Project Description	4
Acknowledgement	5
.....	11
PROPOSAL	11
.....	11
1. Introduction	11
2. Objective.....	11
3. Problem Description.....	11
4. Methodology	12
5. Project Scope	12
6. Feasibility Study	13
i. Risks Involved:.....	13
ii. Resource Requirement:.....	13
7. Solution Application Areas	13
8. Tools/Technology	13
9. Expertise of the Team Members	14
10. Milestones	14
10.1 Preliminary Investigation	14
10.2 Design Blockchain:.....	14
10.3 Design Coin and Other Smart Contracts:.....	14
10.4 Design Database:	14
10.5 Front-End User Interface for Web Module	15
10.6 Integration.....	15
10.7 Testing	15
11. Project Schedule	16
12. Work Breakdown Structure	18
.....	19
SYSTEM REQUIREMENTS SPECIFICATIONS	19
.....	19
1. Introduction	19
1.1. Purpose	19
1.2. Document Conventions	19
1.3. Intended Audience and Reading Suggestions	19
1.4. Product Scope	20

1.5. References	20
2. Overall Description.....	20
3. External Interface Requirement.....	22
3.1. User Interfaces	22
3.2. Hardware Interfaces.....	22
3.3. Software Interfaces	22
3.4. Communications Interfaces	23
4. System Features	23
4.1. Connect MetaMask	23
4.2. Buying Coin.....	24
4.3. Selling Coin	25
4.4. Showing all transaction history	26
4.5. Earning Coin	27
4.6. Transferring Coin.....	28
5. Other Nonfunctional Requirements	29
Performance Requirements.....	29
Safety Requirements	29
Security Requirements	29
Software Quality Attributes	29
Business Rules.....	29
6. Diagrams	30
6.1. Entity Relationship Diagram.....	30
6.2. Domain Model Diagram.....	30
.....	31
SYSTEM DESIGN SPECIFICATIONS	31
.....	31
7. Introduction	31
7.1. Purpose of this document.....	31
7.2. Scope of the development project.....	31
7.3. Definitions, acronyms, and abbreviations	31
7.4. References	32
7.5. Overview of document	32
8. System architecture description	33
8.1. Section Overview.....	33
8.2. General Constraints	33
8.3. Data Design	33

8.4.	Program Structure	34
8.5.	Alternatives Considered	35
9.	Detailed Description of Components	35
9.1.	Section Overview.....	35
9.2.	Component and Detail	35
10.	User Interface Design	38
10.1.	Section Overview.....	38
10.2.	Interface Design Rules	38
10.3.	Detailed Description	39
11.	Reuse and Relationships to other products.....	43
12.	Design Decisions and Tradeoffs	43
13.	Pseudocode For Components.....	43
14.	Appendices	51
14.1.	Class Diagram	51
14.2.	Object Diagram.....	52
14.3.	State Chart Diagram	53
14.4.	Activity Diagram.....	58
14.5.	System Sequence Diagram	63
14.6.	Communication Diagram.....	66
	66
14.7.	Use Case Diagram	69
14.8.	Component Diagram.....	70
14.9.	Deployment Diagram	70
14.10.	System Block Diagram	71
15.	Testing	71
15.1.	Introduction.....	71
15.2.	Purpose of this document.....	71
15.3.	Test Cases	72
16.	User Manual.....	75
16.1.	Home Page	75
16.2.	Crypto News.....	76
16.3.	Earn Coins.....	77
16.4.	Trading.....	78
16.5.	Searching cryptocurrency.....	79
16.6.	Contact.....	80
17.	Iteration Plan.....	81

17.1.	FYP 1	81
17.2.	FYP 2	81
18.	Meeting Log Form	82
19.	Appendix A: Glossary	85

PROPOSAL

1. Introduction

Our project revolves around the emerging technology “Blockchain”. Which is a system of recording transactions in digital ledger that is duplicated and distributed across the entire network of computer systems on the blockchain. Each block in the chain contains a number of transactions, and every time a new transaction occurs on the blockchain, a record of that transaction is added to every participant’s ledger. The decentralized database managed by multiple participants is known as Distributed Ledger Technology (DLT). Our project is to deploy a blockchain to build a platform for dApp developers, an ERC 20 token which can be used to send or receive payments through our blockchain and an exchange in which buying, selling and trading of our and other coins can be done.

2. Objective

Our objective is to deploy a blockchain which is a public, layer 2 blockchain for the development/deployment of all the dApps made by students of SZABIST and other educational institutions and also for the deployment for other dApp developers as well. Moreover, we also aim to launch our cryptocurrency for usage by the student, teacher and other faculty members and general public. We also plan to launch our own exchange to help to buy, sell, send, receive our cryptocurrency.

3. Problem Description

We are making a public, layer 2 blockchain, cryptocurrency & exchange for SZABIST and the general public because of ease of use as if all transactions are done in SZABIST through our cryptocurrency it will be secure, speedy and less costly to vendors as compared to POS terminal. It can also help with traceability and visibility of all the transactions done through blockchain. Our blockchain can help secure all the data present in our servers by decentralizing or distributing it into different nodes. Moreover, it can help with all the dApps made by SZABIST students with reduced GWEI/gas fees if deployed on our blockchain. Where there are immense benefits of a blockchain there are also some issues/challenges we will have to face like Awareness of blockchain which means many students which are not of Computer Science background do not know much about blockchain or cryptocurrency or exchanges. Which means that we will have to train or raise awareness about blockchain before implementing. Second challenge or issue we will face is the initial & maintenance cost of the blockchain like the IPFS server which will hold all of the transaction’s hashes etc. Last issue or threat we will or can face is a wallet on our exchange may be hacked

or a node can be hacked due to human error which can lead to some security issues in blockchain.

4. Methodology

We plan to use Geth and Lighthouse to create and deploy our own public and layer 2 blockchain. We will configure Geth and Lighthouse Client to the Goerli testnet (as it is the only testnet that has Proof of Stake Consensus Mechanism). It works when a user generates a transaction specifying the sender, recipient, and the amount to be transferred. The transaction may also include additional data or instructions based on the blockchain's functionality. Instead of submitting the transaction directly to the main blockchain, Layer 2 solutions enable off-chain validation. Validators or a group of selected participants in the PoS network validate the transaction's correctness, authenticity, and sender's available balance. Once the transaction is validated off-chain, a cryptographic proof is generated to represent the transaction's validity. This proof is periodically committed to the main blockchain as a summary or aggregated representation of multiple off-chain transactions. The main blockchain periodically updates the state of the Layer 2 blockchain based on the committed proofs. This update includes the new balances and state changes resulting from the off-chain transactions. It ensures that the main chain maintains an accurate representation of the Layer 2 blockchain's state. The finality of the transaction occurs when the committed proof is included in a block on the main blockchain. At this point, the transaction is considered settled and irrevocable. The Layer 2 blockchain participants can trust the validity and finality of the transaction without requiring individual confirmation on the main chain. Next, we will need to create a validator node which requires 32.05 goerli faucets. After that we will create the genesis.json file which will set all the protocols of the new blockchain. Next, we will configure Geth and lighthouse client to our new blockchain. Next, we plan to introduce our own cryptocurrency which will be an ERC 20 Token on our blockchain and the transactions of all our cryptocurrency will be validated through our blockchain. We also will make an exchange, which will fetch trading and graphing APIs to display and provide various functions to our users like users can purchase, sell and trade other coins as well as our coins. These transactions will also be validated through our blockchain.

5. Project Scope

Our Blockchain, coin and Exchange has multiple scopes mentioned below;

- **Cyber security:** It will reduce the chances of an attack on data of our university as it verifies data and encrypts it using cryptography technology.
- **Financial:** It helps in saving money from the middlemen as well as the time needed for processing and validating transactions. It works on a distributed database that eases the operations and ensures tight security.

6. Feasibility Study

We will be able to cover all the scopes of our project and will strictly follow our project schedule:

- i. Risks Involved: Security and Privacy challenges, scalability and lack of adoption are some of the risks we might encounter in deploying our blockchain. Moreover, while establishing our own exchange we might face some risk of criminal activities and its high cost which we can overcome by keeping full track of our customers so that illegal activities does not take place on our exchange. Funding is also an issue as to take a blockchain to the main net takes a lot of funding. Boosting the profile of our exchange will also be a great task which we will have to overcome by bringing top-level clients so that our exchange profile can be boosted. We also dependent on external APIs which is also a risk that if their service is disrupted it can also affect our services.
- ii. Resource Requirement: IPFS Server/PC, nodes, blocks, committers, consensers, endorser, APIs for exchange, Solidity and Chain Code.

7. Solution Application Areas

The project is of real value as it is an emerging technology which helps in secure and speedy transactions, traceability and visibility of transactions and also helps all the blockchain developers as it offers less fees than other blockchain. We are targeting all the blockchain developers (from SZABIST and other universities) and also other professional blockchain developers who are making or interested in making dApps. We are also targeting all the people in our university to use our cryptocurrency for all their needs. Moreover, it can also be used for trading purposes using our exchange.

8. Tools/Technology

- **Geth**: A full Ethereum node implementation used for interacting with the Ethereum network and participating in blockchain consensus.
- **Lighthouse Client**: an open-source Ethereum 2.0 client which provides node implementation for participating in the Ethereum 2.0 Beacon Chain and Shard Chains consensus.
- **PC/Server**: A server or PC to store all transactions hashes.
- **Chain Code/ Solidity**: For smart contract development.
- **Web3**: libraries for the front end
- **GO Lang/Java**: To make Blocks etc.
- **React JS**: For development of front end of the exchange
- **Trading View API**: To show graphs of cryptocurrencies
- **Binance WebSocket's API**: To show the live list of all cryptocurrencies

- **MetaMask:** To provide Wallet connection
- **Bybit API:** To fulfill all buy/sell orders
- **Coin Gecko API:** To provide news articles to users

9. Expertise of the Team Members

The team members pre-equipped with the level of knowledge of blockchain, needed for the successful completion of this project. We have studied the relevant course by now in our university and also, we had taken courses from Udemy and studied from YouTube in the respective discipline to increase our knowledge. This project is of equal interest to both team members.

10. Milestones

10.1 Preliminary Investigation:

1. Research About Blockchain, Coin and Exchange
2. Examine the system's goals, restrictions, and scope
3. Analyze Resources that are available, advantages, and feasibility
4. Make a preliminary project proposal and submit it.

10.2 Design Blockchain:

1. Analyze Uses and Requirements
2. Consensus Mechanism Selection
3. Data Structure and Transaction Format
4. Cryptographic Algorithms Selection
5. Network Topology
6. Smart Contracts and Chain code
7. Tokenomics
8. Prototype Development
9. Deployment and Launch

10.3 Design Coin and Other Smart Contracts:

1. Define Coin Specifications and Use Case
2. Create ERC-20 Smart Contract Template
3. Implement Coin Smart Contract Logic
4. Determine Token Distribution and Supply
5. Conduct Security Audit and Code Review
6. Test and Deploy Coin Smart Contract on Testnet
7. Integrate Coin into Wallets and Exchanges

10.4 Design Database:

1. Examine the needs for data storage and retrieval
2. Create an ER Diagram for the proposed system.

3. Examining the finished ER diagram

10.5 Front-End User Interface for Web Module

1. Examine the needs for user input.
2. Create a web-based graphical user interface.
3. Develop Form Prototype
4. Code Implementation
5. Create Web Forms for User Input
6. Examine the web form's user input.
7. Deliverables of Project
8. Examine the project's SDS and SRS.
9. SRS and SDS project deliverables

10.6 Integration

1. Web connectivity API Integration
2. Smart Contracts Integration

10.7 Testing

1. Unit Testing
2. Performance Testing
3. Module Testing
4. Integration Testing
5. System Testing

11. Project Schedule

1. FYP-1

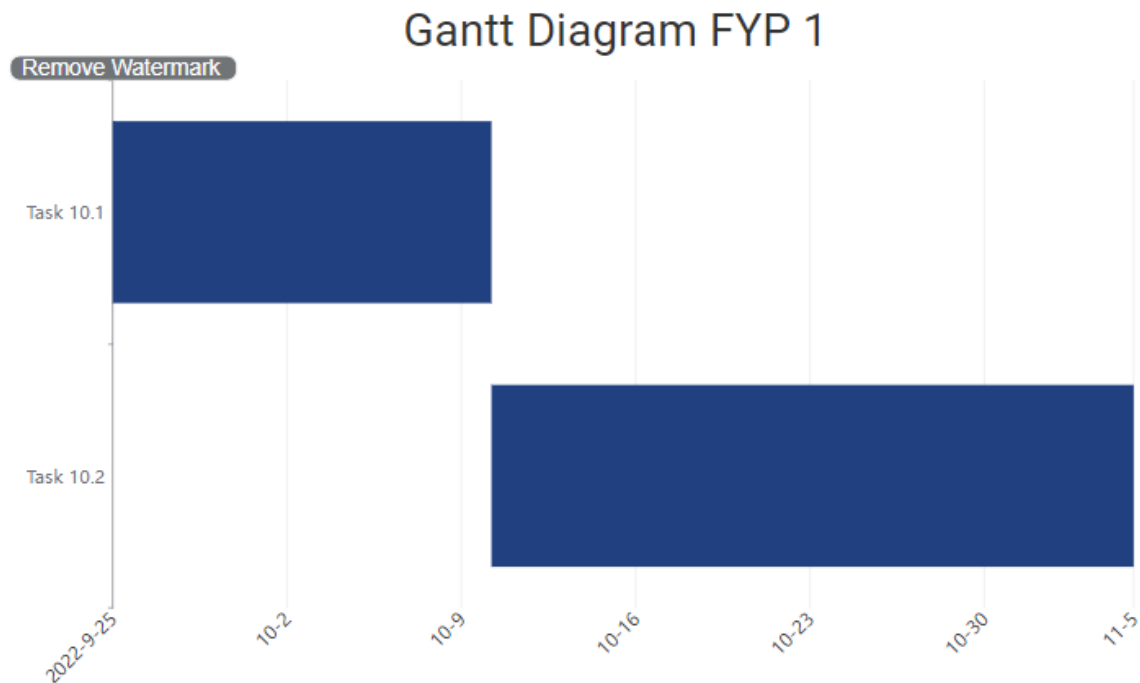


Figure 1. Mid Term FYP 1 Schedule

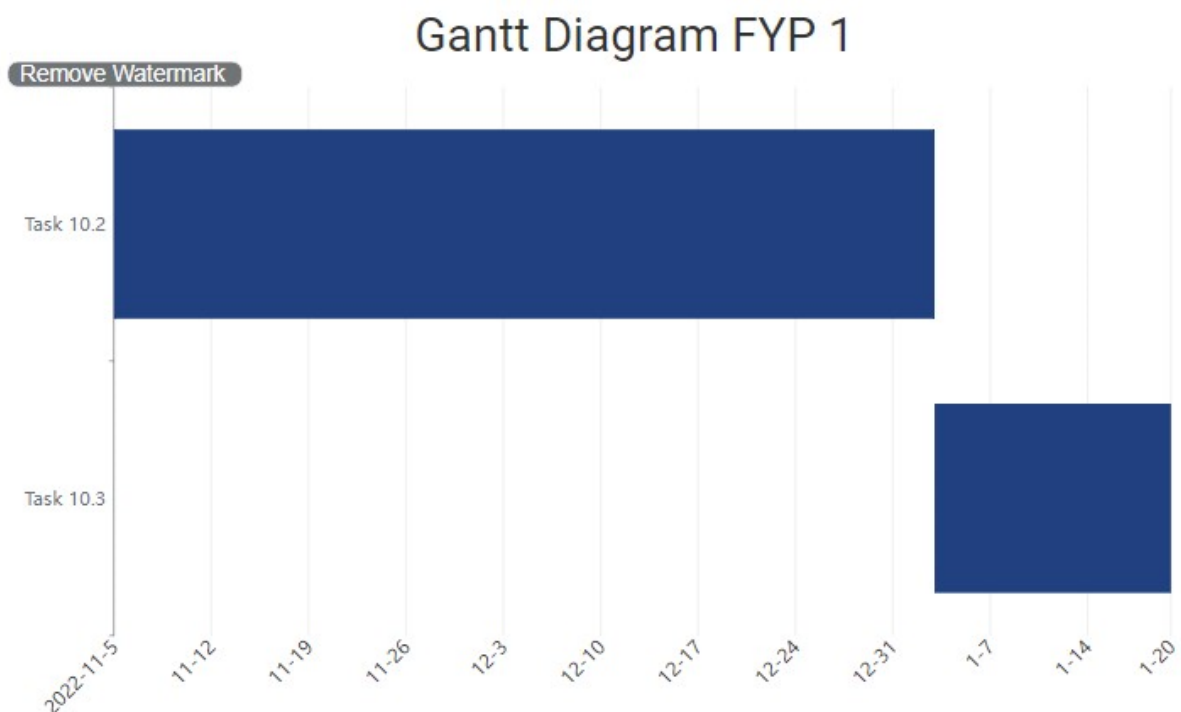


Figure 2. Final Term FYP 1 Schedule

2. FYP-2

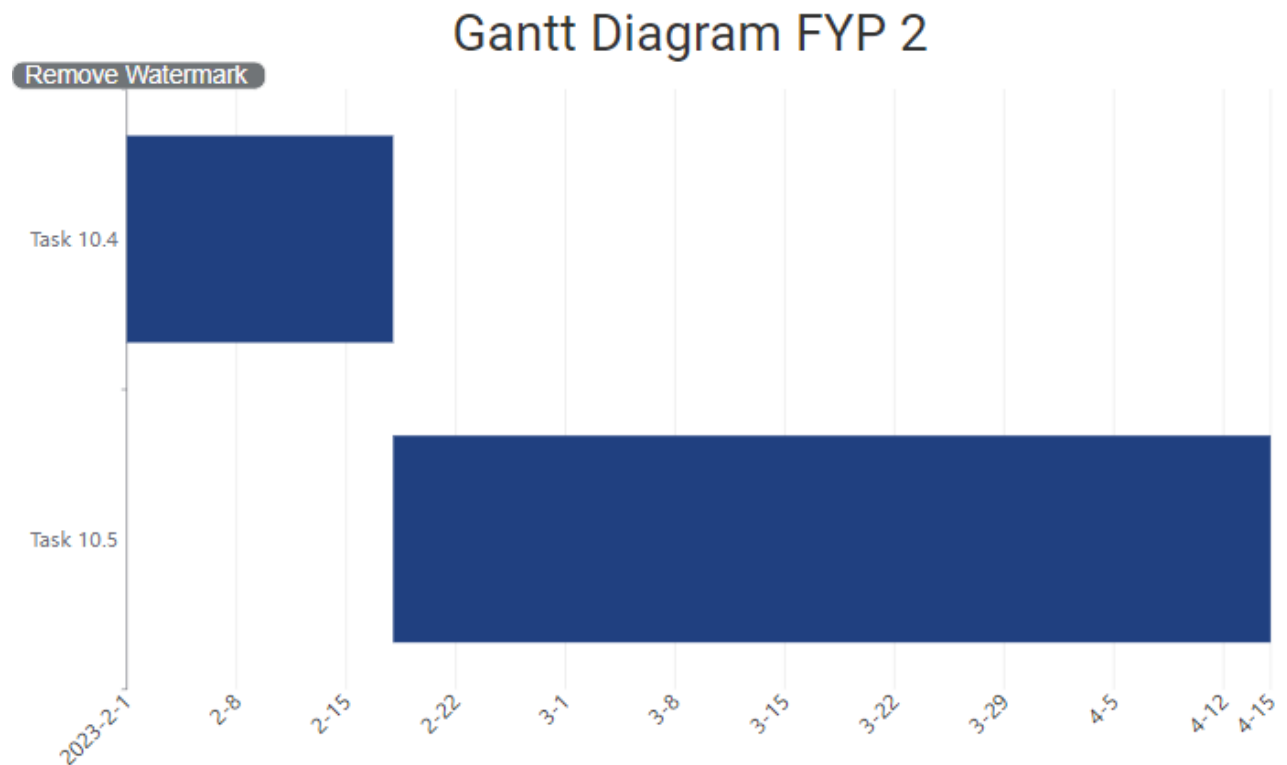


Figure 3. Mid Term FYP 2 Schedule

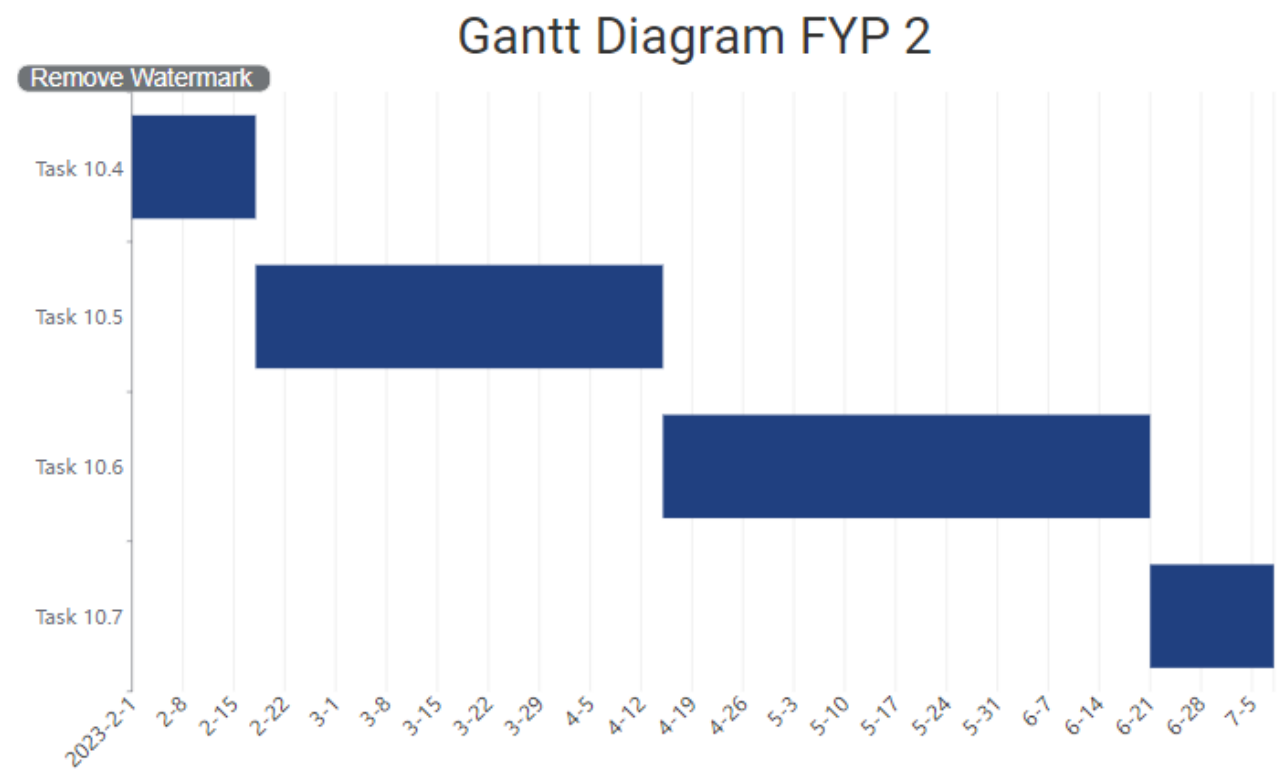


Figure 4. Final Term FYP 2 Schedule

12. Work Breakdown Structure

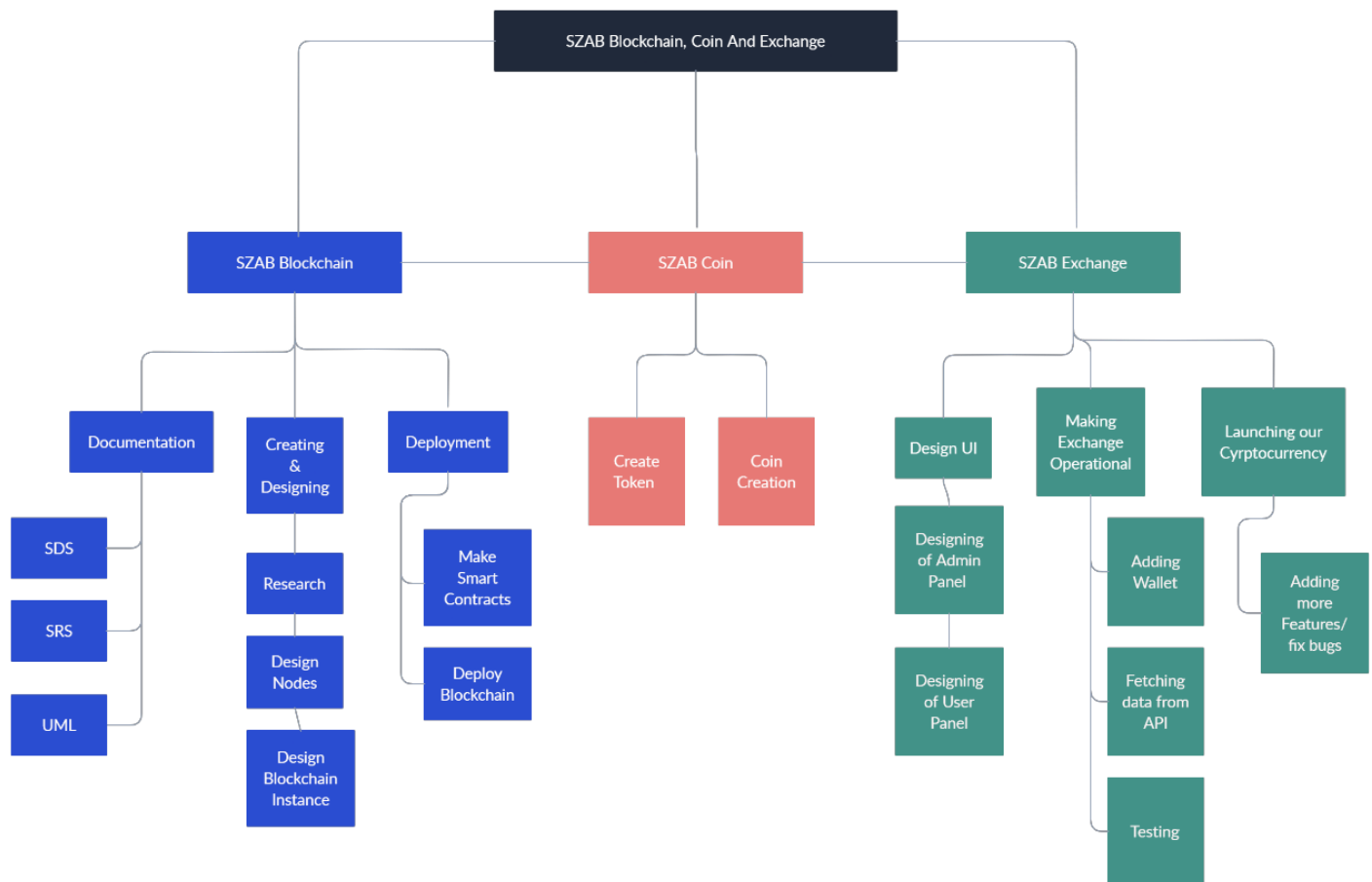


Figure 5. Work Break Down Structure

13. References

- <https://dataconomy.com/2022/05/blockchain-implementation-challenges/>
- <https://youtu.be/iTV89Tqfmqk>
- <https://youtu.be/Y177TCUc4q0>
- <https://www.cryptyk.io/security-risks-public-private-blockchains/>
- <https://www.blockchaincommunity.in/blog/future-scope-of-blockchain-technology-in-india>
- <https://www.vskills.in/certification/blog/future-scope-and-opportunities-of-blockchain-technology-in-2022/>
- <https://www.mantralabsglobal.com/blog/challenges-of-blockchain/>
- <https://www.makeuseof.com/problems-with-blockchain-technology/>
- <https://www.techtarget.com/searchcio/feature/Top-10-benefits-of-blockchain-technology-for-business?amp=1>
- <https://www.blocknative.com/blog/ethereum-validator-lighthouse-geth>
- <https://github.com/binance/binance-spot-api-docs/tree/master>

SYSTEM REQUIREMENTS SPECIFICATIONS

1. Introduction

1.1. Purpose

As blockchain is one of the most emerging technologies nowadays. It is an open ledger which can be accessed by several parties at once. It has numerous benefits but one of its most important benefits is that the information once recorded is hard to change without an agreement from all the parties involved. Blockchain helps in the verification and traceability of multistep transactions needing verification and traceability. It can provide secure transactions, reduce compliance costs, and speed up data transfer processing. Blockchain technology can help contract management and audit the origin of a product. Our purpose is to develop our own blockchain is because it is more secure and transaction are speedy as we can increase transaction per speed and it highly scalable and as we can define the block size and our one block is equal to the one transaction on layer 1 and our data is secured and you are responsible for your own data it increases traceability and visibility is increased as you can keep record and of all the transactions as well as you can trace them.

1.2. Document Conventions

Format of this SRS is as follows:

- Font-Family: Arial.
- Font-Size: Main headings 17, Other Headings 13 and Normal text 12.
- Text-Alignment: Justify.

1.3. Intended Audience and Reading Suggestions

This document is intended for all the new blockchain developers, marketing staff, users and testers. For blockchain developers it contains how our blockchain works, what is our purpose, uses & scope of blockchain and how it is made, and which protocols or consensus mechanism is used etc. For marketing staff, the purpose, and uses of the blockchain is important and how will it affect the user in what way. For users it is what way they will be affected and the benefits of using our blockchain etc. For testers it contains all the test cases from which the blockchain has passed and if any are missing, they can be rectified. Moreover, for testers it also contains how the blockchain works. The sequence the readers should follow is from the start till the end so that they can understand everything which we have built and can be used accordingly.

1.4. Product Scope

The scope of our project is to deploy our own blockchain and develop our own coin and an exchange as due to its encryption feature blockchain is secure it will reduce the chances of various attack on our data of our dApps as it verifies data and encrypts it using cryptographic technology. It will also help in saving money from the middlemen as it is less costly, and it will also reduce the time needed for processing and validating transactions as the authenticity of a transaction is verified and confirmed by participants. As it works on a distributed database that eases the operation and ensures tight security.

1.5. References

- <https://dataconomy.com/2022/05/blockchain-implementation-challenges/>
- <https://youtu.be/iTV89Tqfmqk>
- <https://youtu.be/Y177TCUc4q0>
- <https://www.cryptyk.io/security-risks-public-private-blockchains/#:~:text=Private%20Blockchain%20Risks,on%20their%20ledger%20through%20communication>
- <https://www.blockchaincommunity.in/blog/future-scope-of-blockchain-technology-in-india>
- <https://www.vskills.in/certification/blog/future-scope-and-opportunities-of-blockchain-technology-in-2022/>
- <https://www.mantralabsglobal.com/blog/challenges-of-blockchain/>

2. Overall Description

2.1. Product Perspective

The context and origin of the product being specified in this SRS. Our final year project is a replacement for certain existing digital certification in blockchain network.

HOW BLOCKCHAIN WORKS

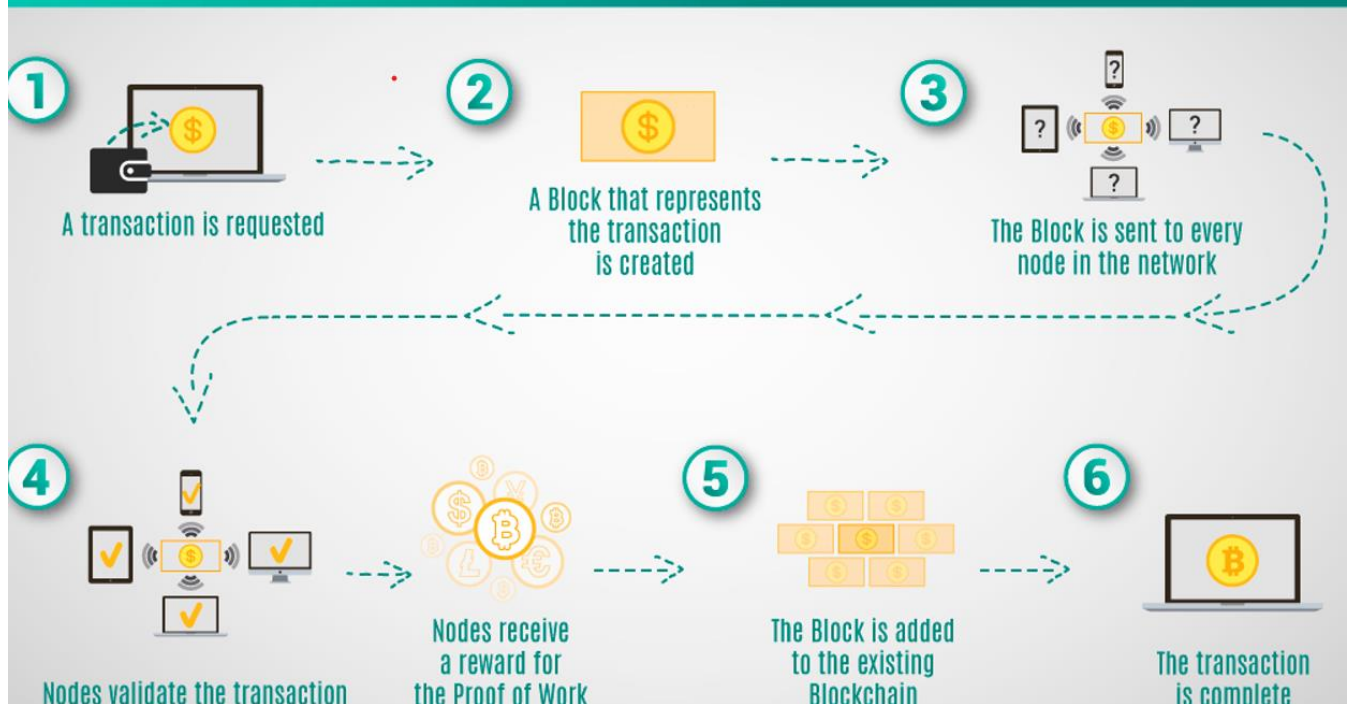


Figure 6. Shows how a Blockchain works

2.2. Product Functions

- Buying Coin
- Connect MetaMask
- Selling Coin
- Earn Coin
- Transferring Coin
- Transaction History

2.3. User Classes and Characteristics

In this SRS, there are three user classes which are the Blockchain Developers, Crypto Traders and Normal End User who can buy, sell and trade coins through our exchange, stake our coins for long term profit, deploy smart contract and dApps and they can also use it for economic activities.

2.4. Operating Environment

The environment on which we will be operating our blockchain and cryptocurrency on Ethereum Virtual Machine/ Geth. Our backend of our exchange will be running on the Hardhat and NodeJS which will be connected to Web3 JS and the front end will be running on React JS and Vanilla JS.

Language Requirement:

- Node JS

- Web3 JS
- React JS
- Vanilla JS
- Hardhat

Database:

We will make our own distributed database on which people will store their transactions which can be done through dApps, trading our coins and via our exchange. We will also use MySQL to store data of the Earn Module in our Exchange as if we use the distributed database, it will cost us money so we will use MySQL to store data of the user used for giving users airdrop.

3. External Interface Requirement

3.1. User Interfaces

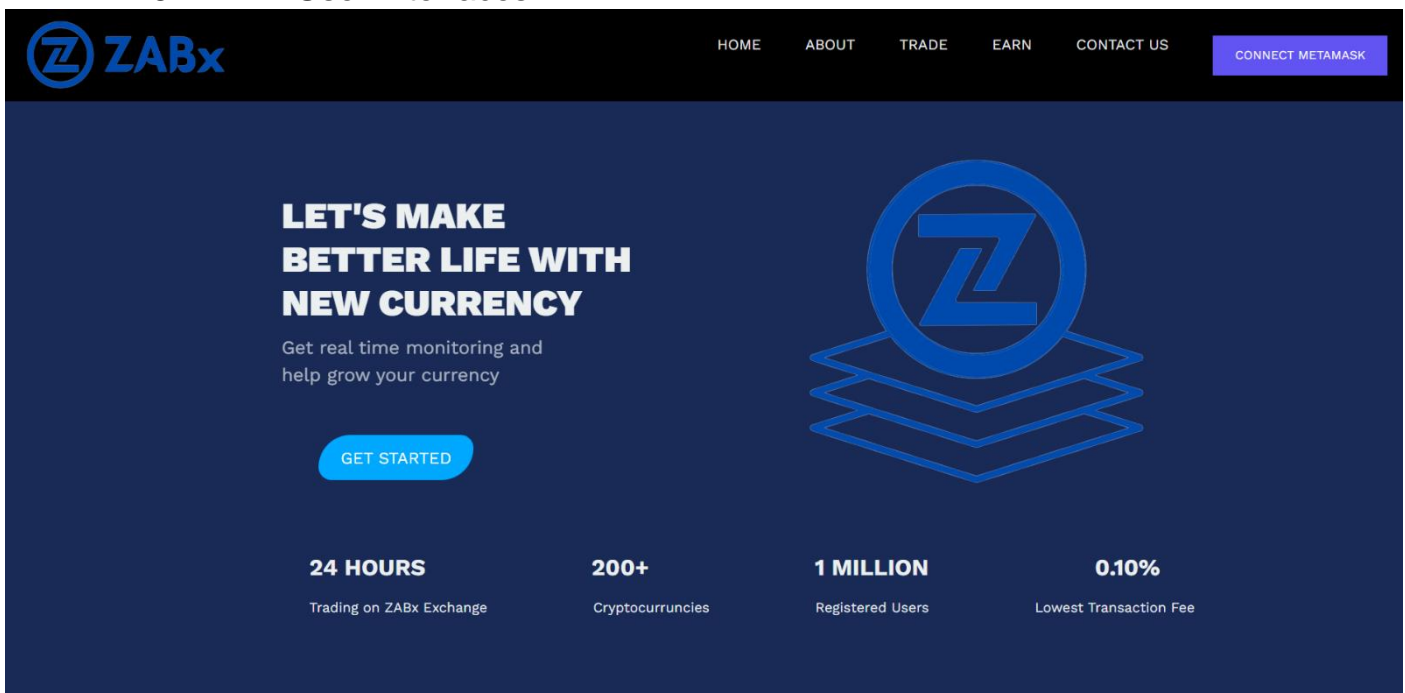


Figure 7. User Interface of SZAB Exchange

3.2. Hardware Interfaces

- Window 10 and higher.
- Linux for Blockchain Node
- RAM 16 GB – 32 GB.
- Storage Space 2 TB.
- Bandwidth 500 Mbps.

3.3. Software Interfaces

- Ethereum Virtual Machine/Geth for connecting the nodes.
- For making smart contract we will be using Solidity.

- Web3 JS for connecting to the Ethereum node.
- Node JS and Hardhat for backend.
- React JS and Vanilla JS for frontend.
- Digital wallet provider.

3.4. Communications Interfaces

Communication in the transaction is encrypted via cryptographic hashing functions like SHA256 which is a mathematical function that transforms or map a given set of data into a bit string of fixed size, also known as the hash value. Hash functions are used for cryptocurrency, password security, and message security, and for our exchange we will be using HTTP protocol for fetching data from trading APIs, and it will be adding authentication and more security to our blockchain and a digital wallet to store cryptocurrency on our exchange and for interacting with dApps.

4. System Features

4.1. Connect MetaMask

Preconditions:

Users installed MetaMask Extension and have made wallet on MetaMask.

Happy Path:

Actor Action	System Response
1) User clicks on the Connect MetaMask Button.	2) dApp checks for the Web3 Provider
4) User enters MetaMask password.	3) dApp asks for MetaMask password.
6) User selects the account to connect and click on authorize button	5) dApp displays the menu for which account to connect
	7) dApp displays the wallet address on the button and Account is logged in

Table 1. Use Case for Connect MetaMask

Alternate Path:

- 3) If the password is not entered, then wait for the user's action.
- 4) If the password is incorrect, ask again to enter the valid password.
- 6) If the user does not select any account or click on the authorize button, then wait for the user's action.

Post-Conditions:

The user is connected to MetaMask

4.2. Buying Coin

Preconditions:

Users must have a stable internet connection and MetaMask account must be connected.

Happy Path:

Actor Action	System Response
2) Users select the cryptocurrency he or she wants to buy	1) dApp shows the list of all cryptocurrencies which can be bought on the exchange
4) User enters the amount to buy	3) dApp shows all the trading graphs and data of the cryptocurrency
8) User confirms transactions and enter password	5) dApp verifies if the user has enough balance to buy cryptocurrency
10) User enters OTP code	6) dApp calculates gas fees
	7) dApp asks for users' approval.
	9) dApp verifies password and generates OTP and send to email or phone number
	11) dApp verifies OTP code and transactions is added to the block
	12) dApp updates portfolio and shows transaction details

Table 2. Use Case for Buying Coin

Alternate Path:

- 2) If coins are not entered, then wait for the user's selection.
- 10) If cancel is selected, Show the main page again.

Post-Conditions:

Coin purchased.

4.3. Selling Coin

Preconditions:

Users must have a stable internet connection and MetaMask account must be connected.

Happy Path:

Actor Action	System Response
2) Users select the cryptocurrency he or she wants to sell	1) dApp shows the list of all cryptocurrencies user has.
4) User enters the amount to sell	3) dApp shows all the trading graphs and data of the cryptocurrency
8) User confirms transactions and enter password	5) dApp verifies if the user has enough balance of selected cryptocurrency
10) User enters OTP code	6) dApp calculates gas fees
	7) dApp asks for users' approval.
	9) dApp verifies password and generates OTP and send to email or phone number
	11) dApp verifies OTP code and transactions is added to the block
	12) dApp updates portfolio and shows transaction details

Table 3. Use Case for Selling Coin

Alternate Path:

- 2) If coins are not entered, then wait for the user's selection.
- 10) If cancel is selected, Show the main page again.

Post-Conditions:

Coin sold.

4.4. Showing all transaction history

Preconditions:

Users must have a stable internet connection and MetaMask must be connected.

Happy Path:

Actor Action	System Response
1) Users selects show all transaction history	2) dApp checks which account is connected or if the account is connected or not
	3) Retrieves the transaction history
	4) Display All transaction history

Table 4. Use Case for showing all transaction history

Alternate Path:

- 1) If nothing is selected, then wait for the user's selection.
- 2) If cancel is selected, Show the main page again.

Post-Conditions:

Transaction history showed.

4.5. Earning Coin

Preconditions:

Users must have a stable internet connection and MetaMask should be connected.

Happy Path:

Actor Action	System Response
2) User enters Full name, phone num, email, twitter URL and telegram username	1) dApp shows the form for earning coins
	3) dApp validates if MetaMask is connected and if the form is filled correctly
	4) dApp stores the data in the database
	5) dApp call the smart contract function
	6) dApp then shows the coin that have been transferred to the user
	7) dApp shows Main Menu

Table 5. Use Case for Earning Coins

Alternate Path:

- 1) If nothing is selected, then wait for the user's selection.
- 2) If cancel is selected, Show the main page again.

Post-Conditions:

User details stored and Coins transferred into account.

4.6. Transferring Coin

Preconditions:

Users must have a stable internet connection and user must be signed in from their MetaMask account.

Happy Path:

Actor Action	System Response
1) User clicks on MetaMask Extension	3) dApp shows the list of all cryptocurrencies which can be sent on the exchange
2) User clicks the send and enters receiver address	5) dApp verifies if the user has enough balance to send cryptocurrency
4) User confirms which currency to send and in how much quantity	6) dApp calculates gas fees
8) User confirms transaction and enter password	7) dApp asks for users' approval.
	9) dApp updates portfolio and shows transactions details

Table 6. Use Case for transferring coin

Alternate Path:

- 3) If nothing is selected, then wait for the user's selection.
- 5) If balance is not enough, then gives an error and wait for the user's correction.
- 6) If balance is not enough, then gives an error and wait for the user's correction.
- 7) If nothing is selected, then wait for the user's selection.
- 8) If user does not confirm, then wait for the user's selection.

Post-Conditions:

Coin transferred.

5. Other Nonfunctional Requirements

Performance Requirements

We will be needing a stable internet connection so that our blockchain and exchange can run constantly and there should be no laggy and bugs in our exchange and the transactions should be speedy in our blockchain and should be approved as soon as possible.

Safety Requirements

We will be banning certain addresses that are reported continuously for fraudulent activities in order to take safety precautions.

Security Requirements

We will be adding two factor authentication and verification for each user using our exchange to make it secure from frauds and we will also be using cryptographic hashing functions, public and private keys for authenticating the transactions.

Software Quality Attributes

- **Privacy:** Even if a single node is hacked then it will not affect the entire blockchain as all nodes are connected to each other and all the transactions are shared with every node.
- **Security:** Even if the entire blockchain is hacked then data is still secured as it is in hash form whereas hacking a blockchain is next to impossible.
- **Transparency:** In blockchain is the key of transparency as the information of all the transactions is shared between every node.
- **Scalability:** We can define the block size our on our blockchain as one block is equal to the one transaction on layer 1 so it is highly scalable.
- **Speedy:** We can make our transaction authenticate faster on our blockchain as we introduce off chain rollups.

Business Rules

To earn tokens, the user email address, wallet address, phone number, twitter URL and telegram username has to be unique. Along with this one user cannot access the contents of other users every user will have their unique private key which they cannot share with anyone and to every user's exchange account a wallet must be connected.

6. Diagrams

6.1. Entity Relationship Diagram

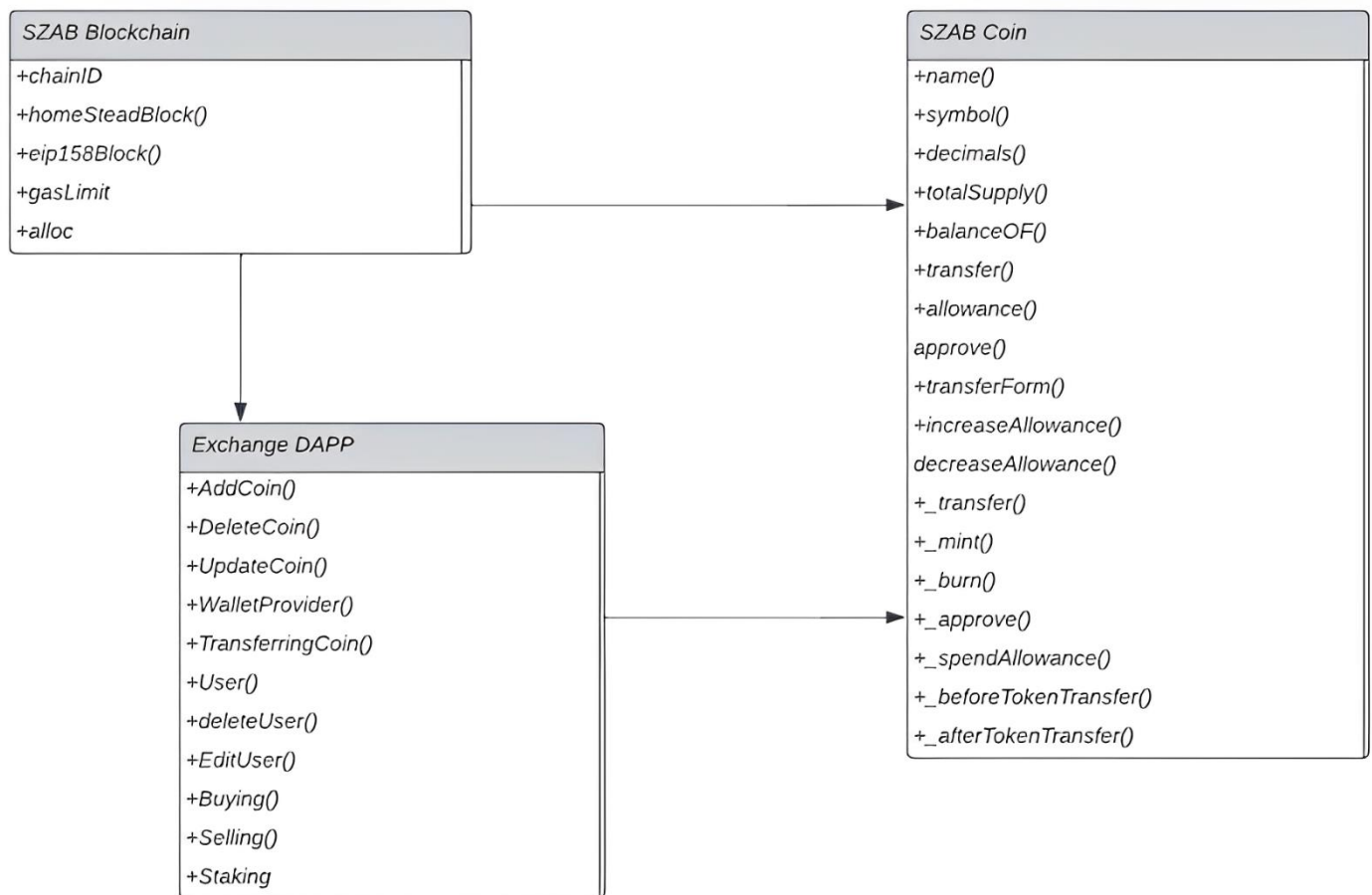


Figure 8. Entity Relationship Diagram

6.2. Domain Model Diagram

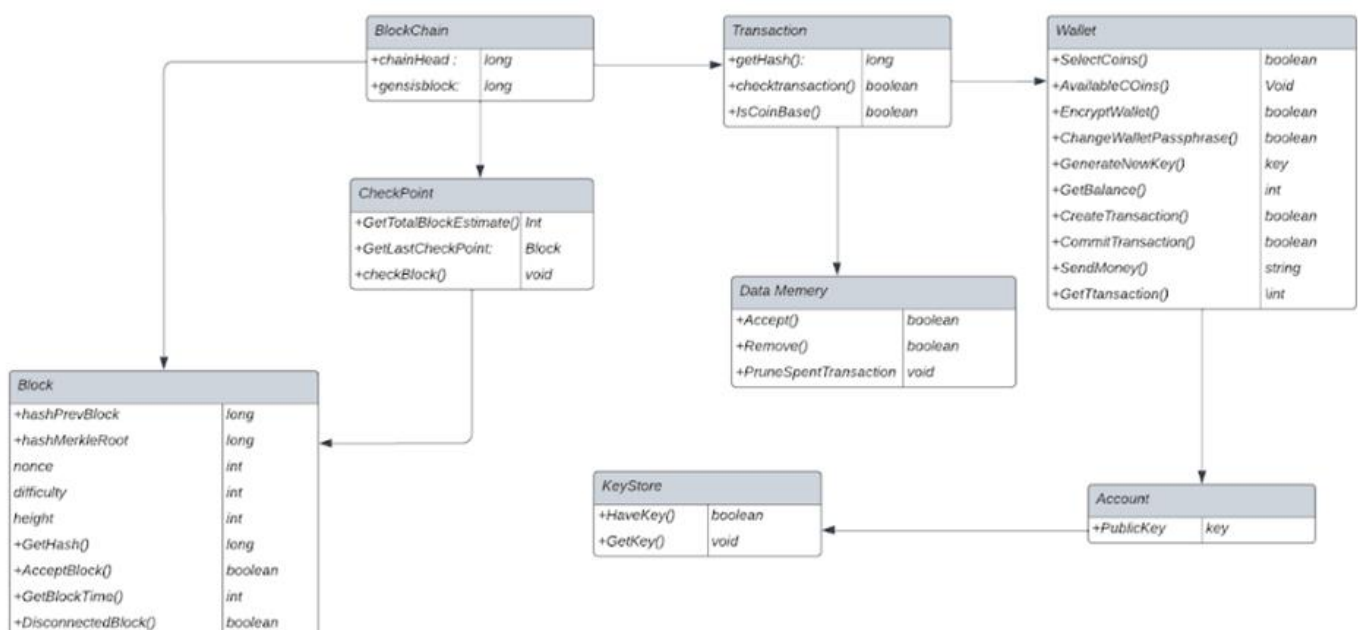


Figure 9. Domain Model Diagram

SYSTEM DESIGN SPECIFICATIONS

7. Introduction

7.1. Purpose of this document

The purpose of this document is to provide a detailed overview of the system architecture of Blockchain, coin and an exchange. This document will provide every minor detail and information about the deployment of blockchain, coin and exchange and our motive behind its deployment. This document will also provide multiple overviews of the system design. System Design Specification describes the high-level architecture of our Blockchain, coin and exchange to the low-level view of even the tiniest component. In our project. Basically, SDS is used to define and describe all the architectural interface, data, and component level design of our Blockchain, coin and an exchange. Within the SDS there are graphical documentation of the software design of our project including class diagram, object diagram, state chart diagram, component diagram, deployment diagram and other supporting requirement information.

7.2. Scope of the development project

The purpose of this document is to provide a detailed overview of the system architecture of Blockchain, coin and an exchange. This document will provide every minor detail and information about the deployment of blockchain, coin and exchange and our motive behind its deployment. This document will also provide multiple overviews of the system design. System Design Specification describes the high-level architecture of our Blockchain, coin and exchange to the low-level view of even the tiniest component. In our project. Basically, SDS is used to define and describe all the architectural interface, data, and component level design of our Blockchain, coin and an exchange. Within the SDS there are graphical documentation of the software design of our project including class diagram, object diagram, state chart diagram, component diagram, deployment diagram and other supporting requirement information.

7.3. Definitions, acronyms, and abbreviations

- **API:** Application Programming Interface
- **SDS:** Software Design Specification
- **SRS:** Software Requirement Specification
- **SQL:** Structured Query Language
- **UI:** User Interface
- **dApp:** Decentralized Application

- **WebSocket:** Live data API
- **Solidity:** Language to write smart contract
- **ERC 20:** Ethereum Request for Comment 20
- **Hardhat:** Backend tool to compile, edit, debug, and deploy smart contracts

7.4. References

- <https://dataconomy.com/2022/05/blockchain-implementation-challenges/>
- <https://youtu.be/iTV89TqfmGk>
- <https://youtu.be/Y177TCUc4g0>
- <https://www.cryptyk.io/security-risks-public-private-blockchains/#:~:text=Private%20Blockchain%20Risks,on%20their%20ledger%20through%20communication>
- <https://www.blockchaincommunity.in/blog/future-scope-of-blockchain-technology-in-india>
- <https://www.vskills.in/certification/blog/future-scope-and-opportunities-of-blockchain-technology-in-2022/>
- <https://www.mantralabsglobal.com/blog/challenges-of-blockchain/>

7.5. Overview of document

- **Section 1:** In this section an overview of the document is given by explaining the purpose of the document along with the scope of the project with terms and references. There is also a brief overview of each section within this document.
- **Section 2:** This section will provide an overview of the system architecture along with the general constraints that will have an impact on our system design along with the structure of the data design along with the relation between the various components and the alternatives considered.
- **Section 3:** This section gives the detail overview of all the components used in the deployment of our blockchain, coin and exchange and a detailed description of the components about its purpose, function, dependencies, resources, interface and data used.
- **Section 4:** This section provides an overview of the design decisions, justification of the conventions and standards that were used to design the interface and the APIs used in the development of the environment.
- **Section 5:** This section will explain the philosophy of reuse and how it affects our project and its implementation.
- **Section 6:** This section explains the designs that were abandoned during the development of our project as they lacked feasibility.
- **Section 7:** In this section the pseudo code for our component is provided.
- **Section 8:** In this section we have placed all the artifacts that are required.

8. System architecture description

8.1. Section Overview

In this section we are going to describe the system architecture in detail and we will also be describing the constraints and usability of our project. It will also describe the data design which is used for the explanation of some major features. It will give the overview of the structure and will give the detail description of the features and the tools and languages used in the deployment of the blockchain, coin and exchange furthermore it will also give an overview the alternatives considered in the deployment process.

8.2. General Constraints

- **Performance and Scalability:** Ensuring that the custom Layer 2 blockchain can handle a high volume of transactions per second and scales effectively to accommodate a growing user base is crucial for a successful platform.
- **Security:** Designing a secure architecture with robust cryptographic protocols, smart contract audits, and secure key management to protect user assets and data from potential attacks and vulnerabilities.
- **Interoperability:** Ensuring that the Layer 2 blockchain and dApp can interact with other blockchains and networks to enable seamless cross-chain asset transfers and transactions, increasing the platform's versatility.
- **User Experience:** Designing an intuitive and user-friendly experience for interacting with the blockchain and dApp to encourage adoption among both experienced and novice users, improving overall user engagement.
- **Regulatory Compliance:** Ensuring that the architecture complies with relevant regulatory frameworks, particularly in terms of KYC and AML requirements for the crypto exchange dApp, to avoid legal issues and build trust.
- **Economic Model:** Designing a sustainable economic model for the cryptocurrency that incentivizes network participants, validators, and users while maintaining stability and avoiding potential economic attacks, ensuring the long-term viability and value of the cryptocurrency.

8.3. Data Design

The data design of the "Crypto Exchange" Decentralized application involves the following components:

- **Blockchain Data:** Smart Contracts will be deployed on our own blockchain for buying, selling, transferring, earning of coins. It is stored as transactions on the blockchain. It is shown in the figure below:

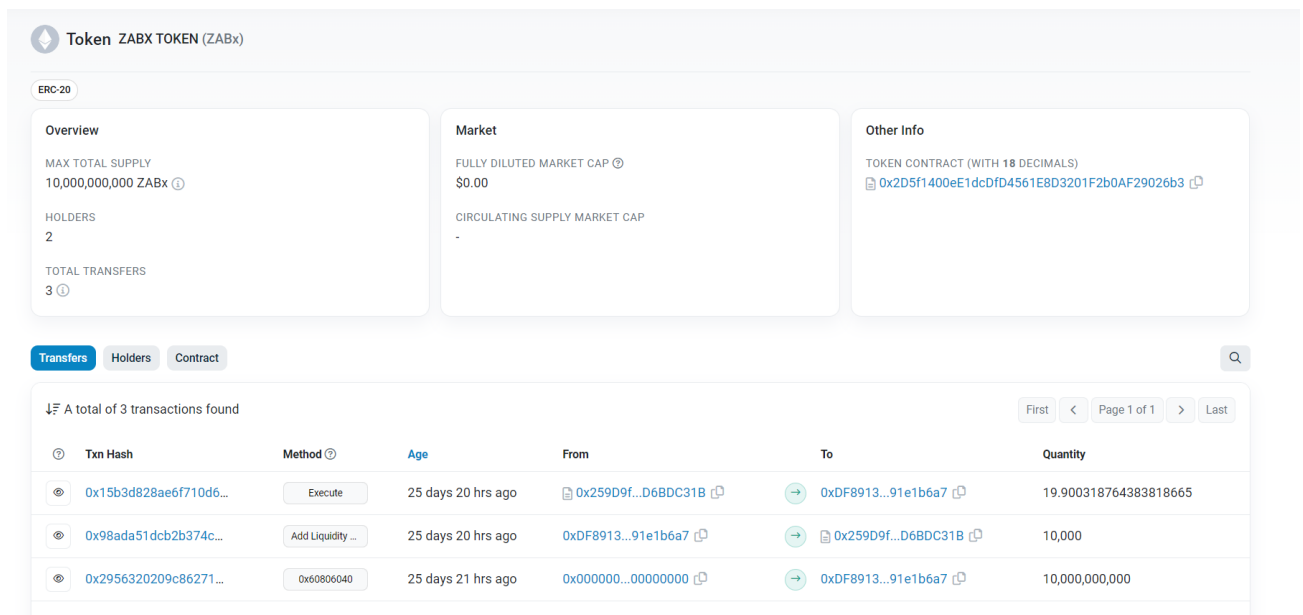


Figure 10. Blockchain Data

- **External Files:** Files related to user will be uploaded and stored on MySQL Database as file storage.

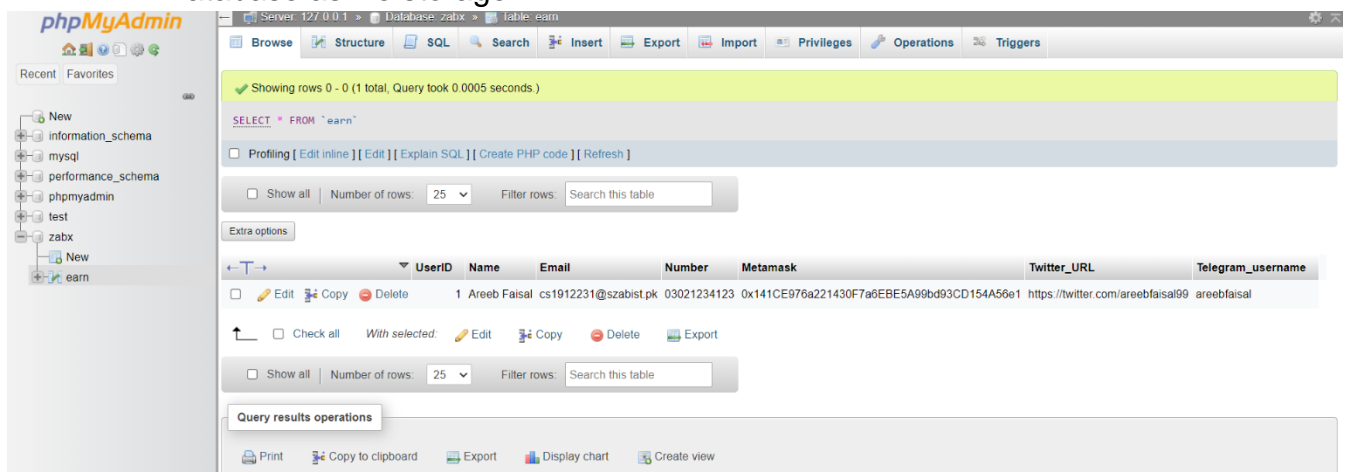


Figure 11. MySQL Database

8.4. Program Structure

For the deployment of the blockchain, first we installed Geth and then we configured lighthouse, which is a client. And Geth and then we collected faucets in our account in order to become the validator and then through we request. We requested to be the validator when we became the validator, then we configured the client and get again with some changes for creating a node. Then all the data of the blockchain was stored on this node and then we created a beacon chain in which we added the Genesis Block. And that's how our blockchain was created. Then in order to show the

transaction on our blockchain we deployed our native coin by making smart contracts using ERC 20 standard through solidity language. It was deployed on the Goerli test net and it was also deployed for Uniswap so that it can show on MetaMask. In order to provide a medium for the transaction of all the coins or to communicate with our blockchain we deployed exchange. Also helps in buying, selling or transferring our coins and also, other cryptocurrencies as well.

8.5. Alternatives Considered

First, we decided to use Hyperledger Fabric for the deployment of our blockchain but then we did not because it did not support public chain and it did not supported layer 2 blockchain. We also tried to code our own blockchain in GO language or Java but it was not possible as it would take too much time and was complex.

9. Detailed Description of Components

9.1. Section Overview

In this section we will provide detailed description about how each component or feature will perform under certain circumstances. Moreover, it will also give us the idea on how user will proceed along with these functions and also about the purpose of these functions along with its resources, interface and other details.

9.2. Component and Detail

9.2.1. Buying Coin:

Identification	Buying coin
Type	A Form
Purpose	It will help users to buy crypto currencies.
Function	It facilitates the process of placing buy orders, matching buyers with sellers and executing transactions
Subordinate	It will show all the transaction details.
Dependencies	It relies on MetaMask without which coins cannot be bought and also Bybit API is also needed to fulfill the buy orders of the user.
Interfaces	Display multiple fields to buy coins using buy orders
Resources	Blockchain and Wallet resources is required
Processing	Once the user enters all the details, the provided data is sent to Bybit API as a buy order and also stored on ZABx Blockchain.

Data	Transaction Hash, Wallet Addresses (From and To), Gas Fees, Value, Transaction Fee and Block Number etc will be stored on ZABx Blockchain.
-------------	--

9.2.2. Selling Coin:

Identification	Selling coin
Type	A Form
Purpose	It will help users to sell crypto currencies.
Function	It facilitates the process of placing sell orders, matching buyers with sellers and executing transactions
Subordinate	It will show all the transaction details.
Dependencies	It relies on MetaMask without which coins cannot be sold and also Bybit API is also needed to fulfill the sell orders of the user.
Interfaces	Display multiple fields to sell coins using sell orders
Resources	Blockchain and Wallet resources is required
Processing	Once the user enters all the details, the provided data is sent to Bybit API as a sell order and also stored on ZABx Blockchain.
Data	Transaction Hash, Wallet Addresses (From and To), Gas Fees, Value, Transaction Fee and Block Number etc will be stored on ZABx Blockchain.

9.2.3. Earning Coin

Identification	Earn coin
Type	A Form
Purpose	It will help users to earn crypto currencies.
Function	It facilitates the process for earning crypto currencies and executing transactions
Subordinate	It will show all the transaction details.
Dependencies	It relies on MetaMask without which coins cannot be earned
Interfaces	Display multiple fields to earn coins
Resources	Database Access, Blockchain and Wallet resources is required

Processing	Once the user enters all the details, the provided data is stored in the database and then smart contract transfer function is called and coins are sent and this transaction is stored on ZABx Blockchain.
Data	All the details about the user like name, phone num, email, Wallet Address, twitter URL and Telegram username is stored in the database whereas the transactions details like Transaction Hash, Wallet Addresses (From and to), Gas Fees, Value, Transaction Fee and Block Number etc. will be stored on ZABx Blockchain.

9.2.4. Show Transaction History

Identification	Show Transaction History
Type	A Component
Purpose	It will help users to see all the transactions
Function	It facilitates the process for showing all the transactions to user which helps them to keep track of all the transactions they did like a bank statement.
Subordinate	N/A
Dependencies	It relies on MetaMask without which transaction history cannot be shown.
Interfaces	Display list of all the transactions done
Resources	Blockchain and Wallet resources is required
Processing	Once the user clicks on the MetaMask icon and clicks on the show all the transactions. It fetches all the data of the wallet address from blockchain and displays the data.
Data	All the details about the user like balance, Transaction hashes, Block Numbers, Timestamps, Receiver Wallet Addresses, Value of Transactions, Value of transactions and also the transaction fees.

9.2.5. Connect MetaMask

Identification	Connect MetaMask
Type	A Component
Purpose	It will help users to use the exchange and perform various functions.
Function	It facilitates to connect and use the user's crypto assets via executing transactions

Subordinate	N/A
Dependencies	It relies on Internet connection and MetaMask extension is required
Interfaces	Display multiple fields to earn coins
Resources	Stable and Good Internet Resources
Processing	Once the user clicks the button, it detects any Web3 provider and when it detects MetaMask, it requests the users to authorize to use MetaMask account and when it is authorized the account is connected.
Data	It only takes the wallet address and private key to authorize transactions which is stored in database or blockchain.

10. User Interface Design

10.1. Section Overview

In this section, we have discussed the set of standards we have used to build our exchange and also the specification and the details about the user interface of our exchange. Moreover, we have also discussed the user design rules in this section that we have used to make our user interface more please and less complex for the users.

10.2. Interface Design Rules

Our user interface is made from ReactJS, HTML and CSS and the data is sent from the frontend to the MySQL database and ZABx Blockchain (using Truffle, Web3). The frontend also displays the data from Coin Gecko API (For crypto news) and Binance WebSocket API (For list of crypto currencies).

10.2.1. The Shortcut Principle:

We have developed a good hierarchy in order to make things clear and simple for the users so that the users can access all parts of our exchange with the minimum number of clicks as possible.

10.2.2. The Simplicity Principle:

We have designed our exchange in such a way that it is easy for the users to perform simple and common task simply and clearly in the user's language we have not made anything complicated or difficult for the user to understand or find.

10.2.3. The Organized Principle:

The design of the user interface is organized in a manner that all related things are placed together separating the unrelated things and making the resemblance between the similar things and differentiating between the dissimilar things.

10.3. Detailed Description

When you open our website, you can connect/disconnect MetaMask through it by clicking on the connect MetaMask Button and when MetaMask is connected it shows the wallet address and when it is clicked it shows the disconnect button. On the interface you can also see the roadmap of the development process of our blockchain, coin and exchange. You can also read crypto news articles through our interface. You can also Earn our native coin “ZABx” using the Earn tab, by filling the form, connecting the MetaMask and by clicking Claim Rewards button you can receive rewards instantly to your MetaMask account and it shows the transaction details. You can also trade coins via the Trade tab on the interface which when clicked shows all the currencies your can trade in and it is also shows table of all the necessary information along with it which is updated every 2 seconds. When you search or click on your desired coin you will see the graphs with all the indicators and you will see the trade menu which has a buy or sell option and it has a form which can be filled with necessary information to buy or sell any coin but you will have to connect MetaMask for it. Upon completing the transaction, it shows the details of the transactions. You can contact us through the information given on our website or directly send us a message using the contact us tab on our website.

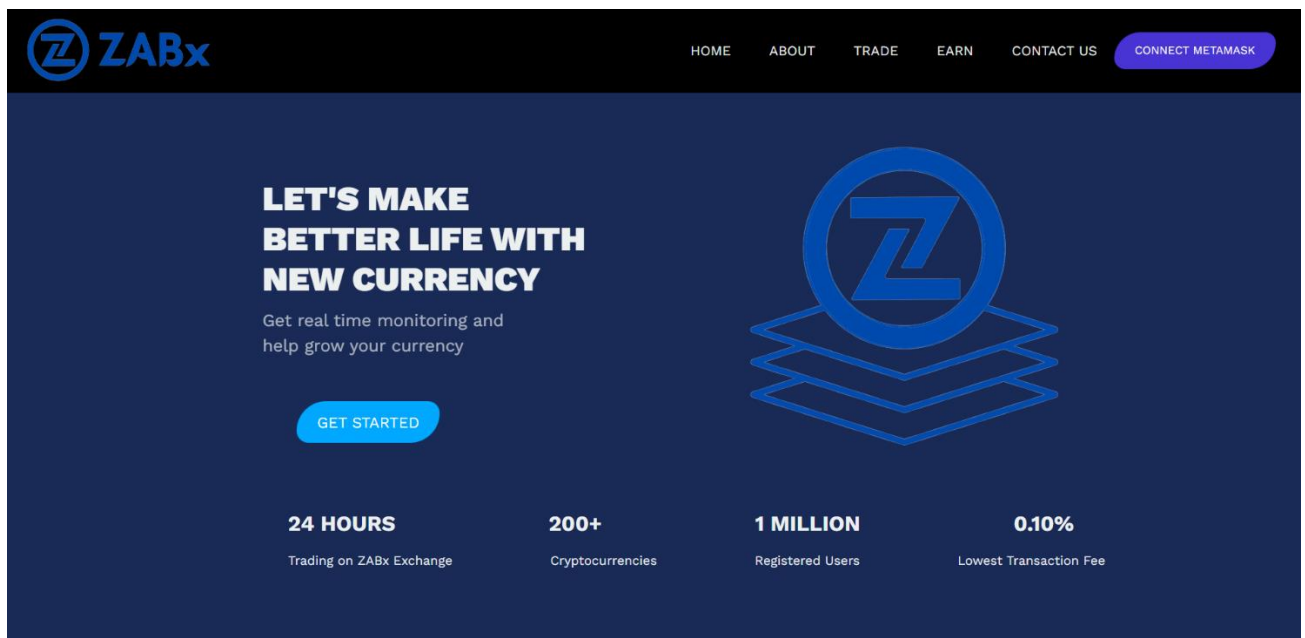



Figure 12. Home Page


CRYPTO NEWS



Uniswap: Critical Price Increase!

Uniswap (UNI) price has recently reclaimed the \$6.50 milestone for the first time since March 2023. Will investors resist for more gains amid increasing selling...


[Read More](#)



Breaking: Ripple Ruling Rejected by Federal Judge


In a surprising twist to the ongoing Securities and Exchange Commission (SEC) case against Ripple, U.S. District Judge Jed Rakoff has repudiated a significant aspect...

[Read More](#)



Should the ECB tighten its leash on EU banks? Expert insight


The European Central Bank (ECB) stands at a crossroads where its actions will dictate the direction of the Eurozone's banking sector. Faced with significant risks...




DeSantis Accuses Biden of 'War on Bitcoin,' Vows to Stop It if Elected President

Presidential candidate and Florida Governor Ron DeSantis (R-Fla.) called out the Biden Administration for its "war" on bitcoin and


Figure 13. Crypto News on Home Page

HOME ABOUT TRADE EARN CONTACT US [CONNECT METAMASK](#)


YOUR TRUSTED CRYPTO EXCHANGE




Allows you to restrict account and addresses that can access your account.



Your transaction data is secured only you have access to your personal information.




We have implemented fraud detection to keep your wallet safe.




FRAUD PREVENTION

Always upfront no unfair rates no hidden fee.



We provide fingerprint and face ID for quick and easy access to your account.

Figure 14. About Page



[HOME](#)
[ABOUT](#)
[TRADE](#)
[EARN](#)
[CONTACT US](#)
[CONNECT METAMASK](#)

To Earn Free Coins

Name

Email

Phone Number


Twitter URL

Telegram Username

Not Connected

[CLAIM YOUR REWARD](#)

Figure 15. Earn Page



[HOME](#)
[ABOUT](#)
[TRADE](#)
[EARN](#)
[CONTACT US](#)
[CONNECT METAMASK](#)

#	Name	Price	Volume (24H)	Change (24H)
1	BTCUSD	29207.29000000	22445.22477000	-0.163% +
2	ETHUSD	1952.87000000	173434.86980000	-0.670% +
3	BNBUSD	240.90000000	335239.19500000	-0.660% +
4	NEOUSD	8.65000000	153375.49000000	-1.593% +
5	LTCUSD	91.87000000	770880.49900000	-1.890% +
6	ADAUSD	0.30810000	64834735.80000000	-0.996% +
7	XRPUSD	0.69280000	342887348.00000000	-1.842% +
8	TUSDUSD	0.99850000	119987791.00000000	0.000% -
9	XLMUSD	0.15250000	78030153.00000000	-1.866% +
10	ONTUSD	0.19360000	4771360.00000000	0.577% +
11	ETCUSD	19.45000000	469389.49000000	-1.284% +
12	NULSUSD	0.20160000	522959.00000000	-2.041% +
13	USDCUSD	1.00000000	251927349.00000000	0.000% -
14	LINKUSD	7.49100000	359136.94000000	-0.253% +
15	WAVESUSD	1.90700000	2654865.13000000	-2.654% +
16	FETUSD	0.20700000	21456265.00000000	-1.004% +

Figure 16. List of all available coins

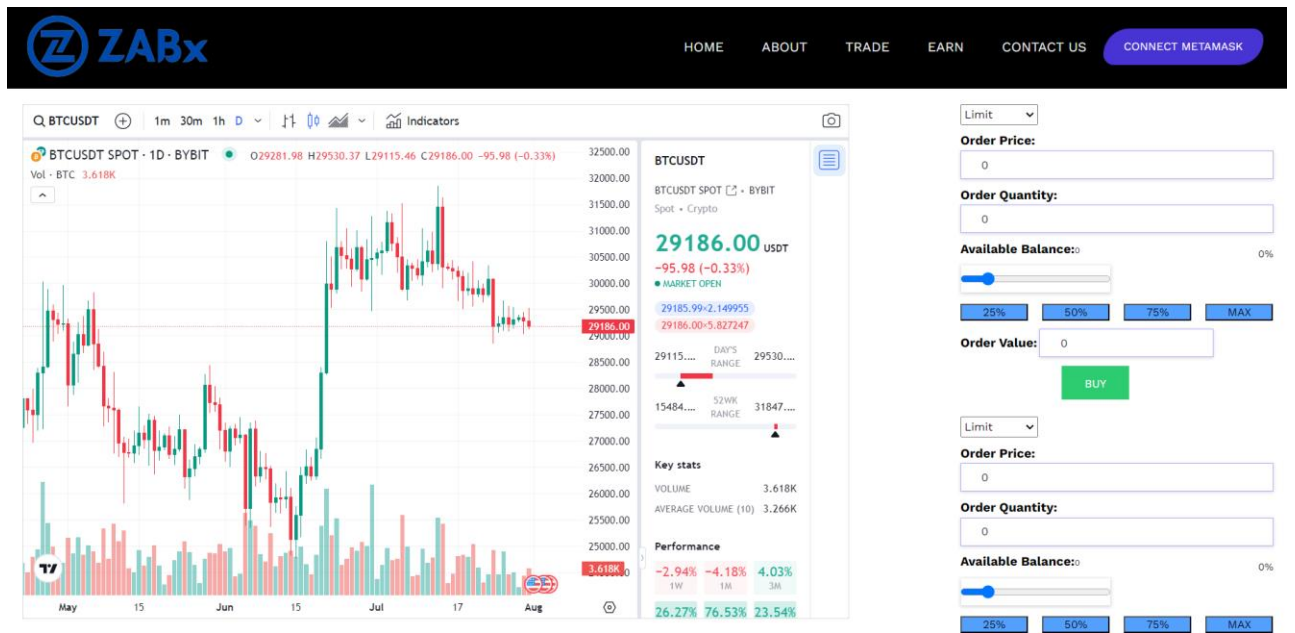


Figure 17. Trade Page

The screenshot shows the ZABx contact page. The top navigation bar is identical to the trade page. The main content area features a contact form with the following elements:

- Text:** "Feel Free to Contact us"
- Form Fields:**
 - Name:** Input field.
 - Email:** Input field.
 - Message:** Large text area for the contact message.
- Submit Button:** A blue button labeled "send".

Figure 18. Contact Us Page

11. Reuse and Relationships to other products

The ability to reuse existing code makes it possible for developers to work in a more efficient manner across a variety of applications.

11.1. Blockchain:

We used the Geth which has multiple functions like mining etc. It helped us interact with the Ethereum Blockchain which is in itself a platform which has been used by many other Blockchain Designers as well.

11.2. Crypto Currency:

We used ERC-20 standard which is a standard for any token to be deployed on Ethereum Blockchain which has been used by many other Blockchain Developers as well.

11.3. Exchange dApp:

We used MetaMask integration, which is a browser extension for Ethereum transactions. It helps to allow first user authentication and then transaction processing. We have also used the widely used database MySQL to store data of the user. We have also utilized Ethers.js library which is open-source library which helps to interact with Ethereum smart contracts for transaction handling.

12. Design Decisions and Tradeoffs

12.1. Design Decisions:

- ✓ Made a Layer 2 blockchain to speed up the processing and validation of transactions by saving our time.
- ✓ Made a PoS Blockchain to use lesser computer resources
- ✓ Made a Public Blockchain so that it is accessible to everyone
- ✓ Used MetaMask as Web3 Wallet Provider
- ✓ Used Ethers.js to handle and interact with smart contract.
- ✓ Used MySQL to store some data of user
- ✓ Used ReactJS for frontend

12.2. Tradeoffs:

- ✗ Making a Layer 2 Blockchain was complex and connecting to Ethereum was also time consuming
- ✗ Making a PoS Blockchain was time consuming and had to do a lot of research.
- ✗ Making a Public Blockchain required more resources than a Private Blockchain.

13. Pseudocode For Components

13.1. Trading

Function Trade:

```
// State variables initialization
sym = useParams()
orderType = 'limit'
buyOrderPrice = 0
buyOrderQuantity = 0
buySliderValue = 0
buyTriggerPrice = 0
sellOrderPrice = 0
sellOrderQuantity = 0
sellSliderValue = 0
sellTriggerPrice = 0
availableBalance = 0
isMetamaskConnected = false

// Function to handle order type change
Function handleOrderTypeChange(event):
  setOrderType(event.target.value)
  If event.target.value is 'market':
    setBuyOrderPrice(0)
    setSellOrderPrice(0)

// Function to handle buy order price change
Function handleBuyOrderPriceChange(event):
  setBuyOrderPrice(parseFloat(event.target.value))

// Function to handle sell order price change
Function handleSellOrderPriceChange(event):
  setSellOrderPrice(parseFloat(event.target.value))

// Function to handle buy order quantity change
Function handleBuyOrderQuantityChange(event):
  setBuyOrderQuantity(parseFloat(event.target.value))

// Function to handle sell order quantity change
Function handleSellOrderQuantityChange(event):
  setSellOrderQuantity(parseFloat(event.target.value))

// Function to handle sell slider change
Function handleSellSliderChange(value):
  setSellSliderValue(value)
  maxSliderValue = 100
  sellOrderQuantity = (availableBalance * value) / maxSliderValue
  setSellOrderQuantity(sellOrderQuantity)

// Function to handle buy slider change
```

```

Function handleBuySliderChange(value):
    setBuySliderValue(value)
    maxSliderValue = 100
    buyOrderQuantity = (availableBalance * value) / maxSliderValue
    setBuyOrderQuantity(buyOrderQuantity)

// Function to handle buy trigger price change
Function handleBuyTriggerPriceChange(event):
    setBuyTriggerPrice(parseFloat(event.target.value))

// Function to handle sell trigger price change
Function handleSellTriggerPriceChange(event):
    setSellTriggerPrice(parseFloat(event.target.value))

// Function to handle buy slider button click
Function handleBuySliderButtonClick(percentage):
    maxSliderValue = 100
    value = (maxSliderValue * percentage) / 100
    setBuySliderValue(value)
    buyOrderQuantity = (availableBalance * value) / maxSliderValue
    setBuyOrderQuantity(buyOrderQuantity)

// Function to handle sell slider button click
Function handleSellSliderButtonClick(percentage):
    maxSliderValue = 100
    value = (maxSliderValue * percentage) / 100
    setSellSliderValue(value)
    sellOrderQuantity = (availableBalance * value) / maxSliderValue
    setSellOrderQuantity(sellOrderQuantity)

// Function to handle buy or sell action
Function handleBuyOrSell():
    print('Executing order...')
    If availableBalance is greater than or equal to buyOrderValue and
sellOrderValue:
        If orderType is 'buy':
            buyCoin()
        Else if orderType is 'sell':
            sellCoin()
    Else:
        setErrorMessage('Insufficient balance')

// Calculate buy and sell order values
buyOrderValue = buyOrderQuantity * buyOrderPrice
sellOrderValue = sellOrderQuantity * sellOrderPrice

// Fetch balance using useEffect hook

```

```

Function fetchBalance():
  Try:
    If Web3 is available:
      Create web3 instance
      Get the current account address
      Fetch balance of the account
      Set isMetamaskConnected to true
      Set availableBalance with the fetched balance
    Else:
      Throw 'Metamask not found' error
  Catch error:
    Print 'Error connecting to Metamask:', error

// Call fetchBalance using useEffect hook
useEffect:
  Call fetchBalance()

// Render the JSX components
Return (
  <Wrapper>
    <TradingViewWrapper>
      <TradingViewWidget />
    </TradingViewWrapper>
    <BuySellWrapper>
      // Render buy and sell components
    </BuySellWrapper>
  </Wrapper>
)

```

13.2. TradingView

```

Function TradingViewWidget:
  // Get the symbol (parameter) from the URL
  sym = get value of 'sym' parameter from the URL using useParams()

  // Create a reference to the createWidget function
  onLoadScriptRef = create a new reference

  // useEffect hook to load TradingView script and create the widget
  useEffect:
    // Assign the createWidget function to the onLoadScriptRef
    onLoadScriptRef.current = createWidget

  // Check if the TradingView script is already loading
  If tvScriptLoadingPromise does not exist:
    // Create a promise to load the TradingView script
    tvScriptLoadingPromise = new Promise:
      // Create a script element and set its attributes

```

```

script = create a new script element
script.id = 'tradingview-widget-loading-script'
script.src = 'https://s3.tradingview.com/tv.js'
script.type = 'text/javascript'
script.onload = resolve the promise

// Append the script to the document head
Append script to the document head

// When the TradingView script is loaded, call the createWidget
function
When tvScriptLoadingPromise is resolved:
  If onLoadScriptRef.current exists:
    Call onLoadScriptRef.current()

// Clean up function: remove the onLoadScriptRef when the
component unmounts
Return a function:
  Set onLoadScriptRef.current to null

// Function to create the TradingView widget
Function createWidget:
  // Check if the element with the specified ID exists and 'TradingView'
is available in window
  If element with ID 'tradingview_d836d' exists and 'TradingView' is
available in window:
    // Create a new TradingView widget with specified configuration
new TradingView.widget:
  width: 980
  height: 610
  symbol: sym
  interval: "D"
  timezone: "Asia/Karachi"
  theme: "light"
  style: "1"
  locale: "en"
  toolbar_bg: "#f1f3f6"
  enable_publishing: false
  allow_symbol_change: true
  details: true
  container_id: "tradingview_d836d"

// Render the JSX component
Return a div with className 'tradingview-widget-container':
  Div with id 'tradingview_d836d'

```

13.3. Earn:

```

Function Earn:
// State variables initialization
account = null
email = ""
phoneNumber = ""
twitter = ""
telegram = ""
name = ""

// Function to handle claiming rewards
Function handleClaimReward():
    If email is empty or not a valid email:
        Show an alert with "Please enter a valid email address."
        Return

    If twitter is empty or not a valid Twitter URL:
        Show an alert with "Please enter a valid Twitter URL."
        Return

    If phoneNumber is empty or not a valid phone number:
        Show an alert with "Please enter a valid phone number."
        Return

    Print 'Name:', name
    Print 'Email:', email
    Print 'Phone Number:', phoneNumber
    Print 'Twitter URL:', twitter
    Print 'Telegram Username:', telegram

    ethereumProvider = detectEthereumProvider()
    If ethereumProvider exists:
        web3 = create Web3 instance using ethereumProvider
        contract = create new contract instance with ERC20_ABI and
CONTRACT_ADDRESS

        amount = convert 2000 coins to wei

        contract.methods.transfer(account, amount).send({ from: account },
function(error, hash):
    If error exists:
        Print 'Transfer error:', error
    Else:
        Print 'Transaction hash:', hash

Else:
    Print 'Please install MetaMask.'

```



```

// Clear form fields
clearFormFields()

// Function to validate email format
Function validateEmail(email):
  emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/
  Return result of emailRegex test on email

// Function to validate Twitter URL format
Function validateTwitterURL(url):
  twitterRegex = /^(?:https?:\V)?(?:www\.)?twitter\.com\V(?:#!\V)?[a-zA-Z0-9_]+(?:\Vw+)*$/
  Return result of twitterRegex test on url

// Function to validate phone number format
Function validatePhoneNumber(phoneNumber):
  phoneNumberRegex = /^\d{10}$/
  Return result of phoneNumberRegex test on phoneNumber

// Function to clear form fields
Function clearFormFields():
  Set name to empty string
  Set email to empty string
  Set phoneNumber to empty string
  Set twitter to empty string
  Set telegram to empty string

// UseEffect hook to detect Ethereum provider and update account
state
useEffect:
  detectProvider():
    Try:
      ethereumProvider = detectEthereumProvider()
      If ethereumProvider exists and ethereumProvider has
selectedAddress property:
        Set account to ethereumProvider.selectedAddress
      Else:
        Set account to null
    Catch error:
      Print "Error detecting Ethereum provider:", error

  Call detectProvider()

// Render the JSX components
Return (
  <Container>

```

```
<FormWrapper>
  <Form>
    // Form inputs for name, email, phone number, Twitter URL, and
Telegram username
    // Account information display
    // Button to claim rewards
  </Form>
</FormWrapper>
</Container>
)
```

14. Appendices

14.1. Class Diagram

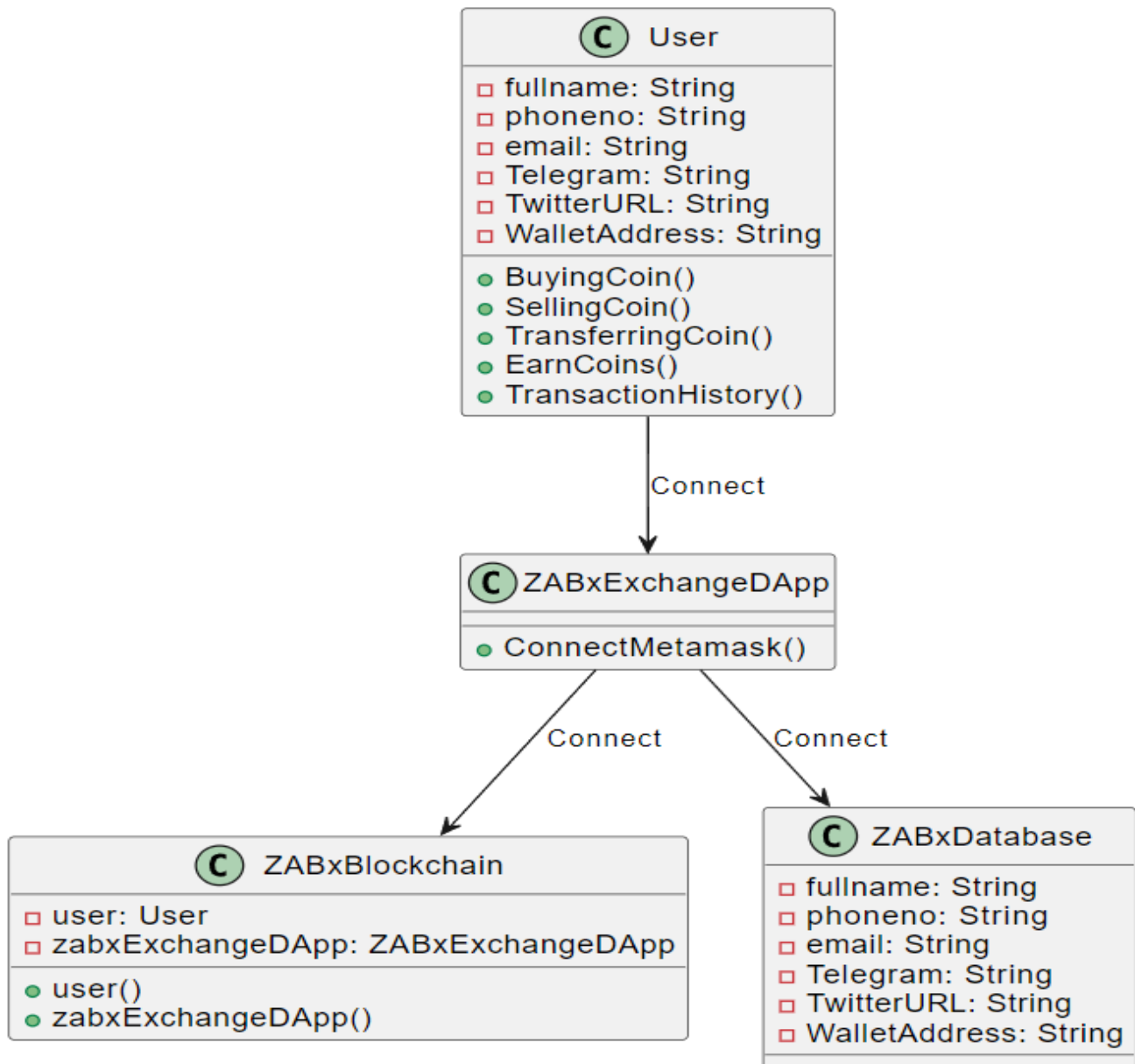


Figure 19. Class Diagram

14.2. Object Diagram

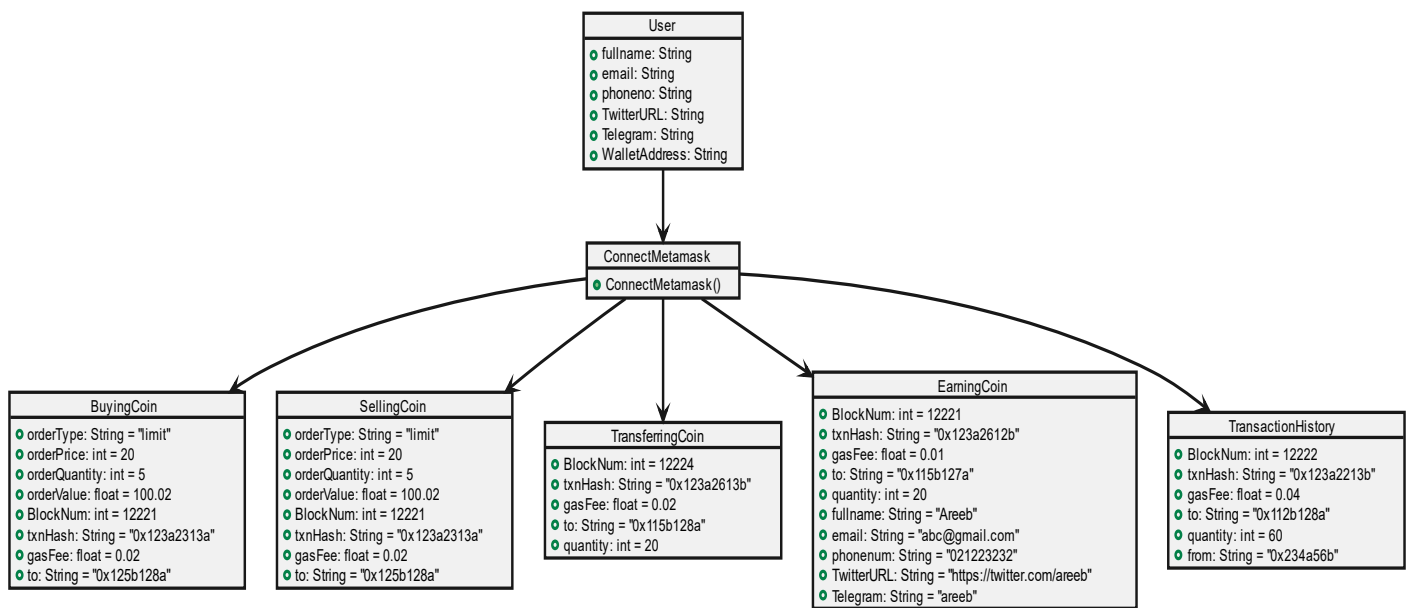


Figure 20. Object Diagram

14.3. State Chart Diagram

14.3.1. Buying Coin

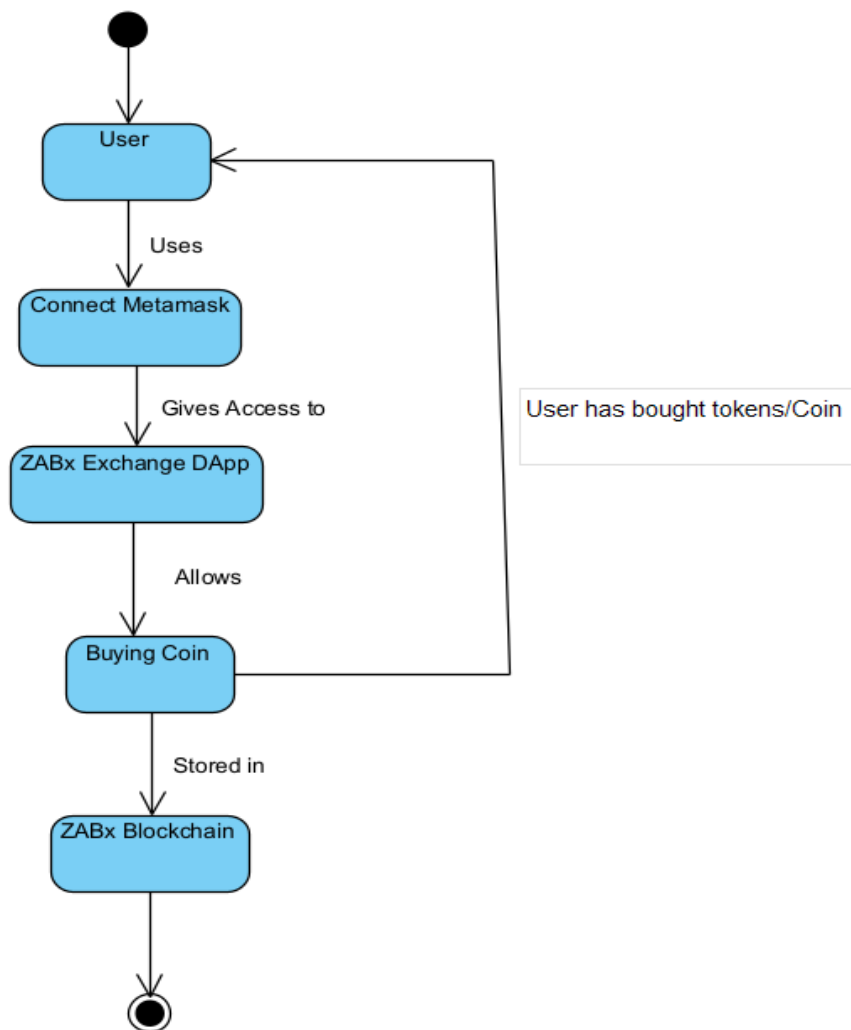


Figure 21. Buying Coin Statechart Diagram

14.3.2. Selling Coin

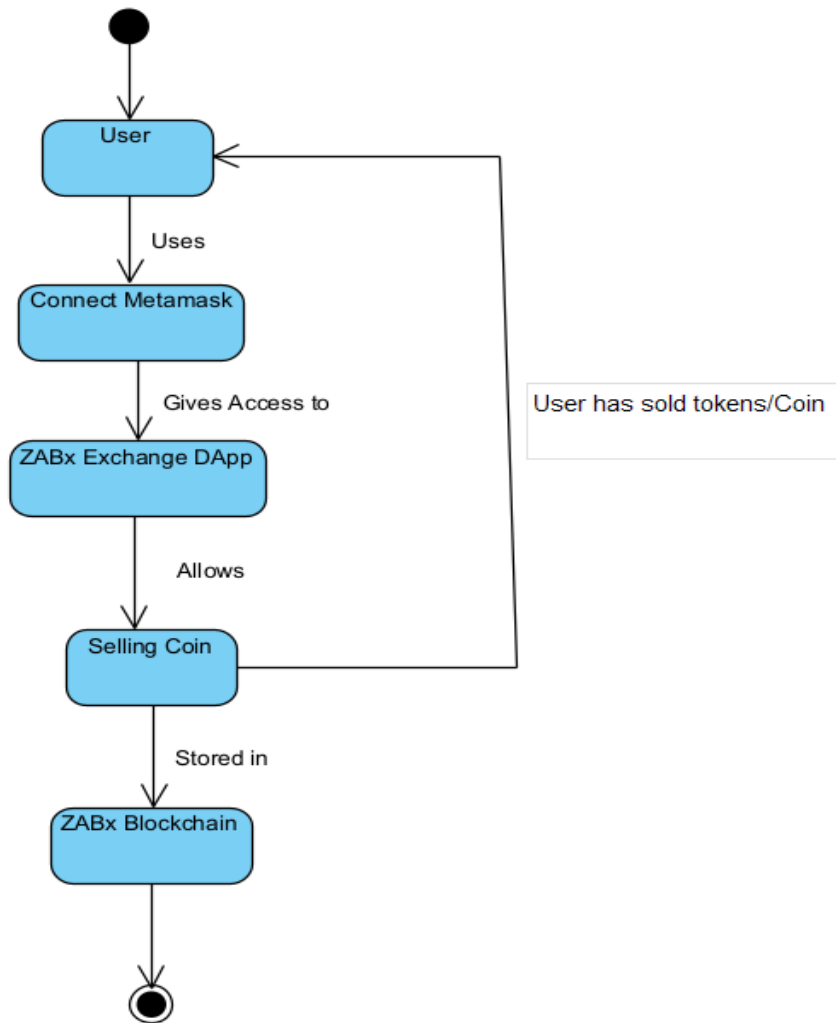


Figure 22. Selling Coin Statechart Diagram

14.3.3. Earning Coin

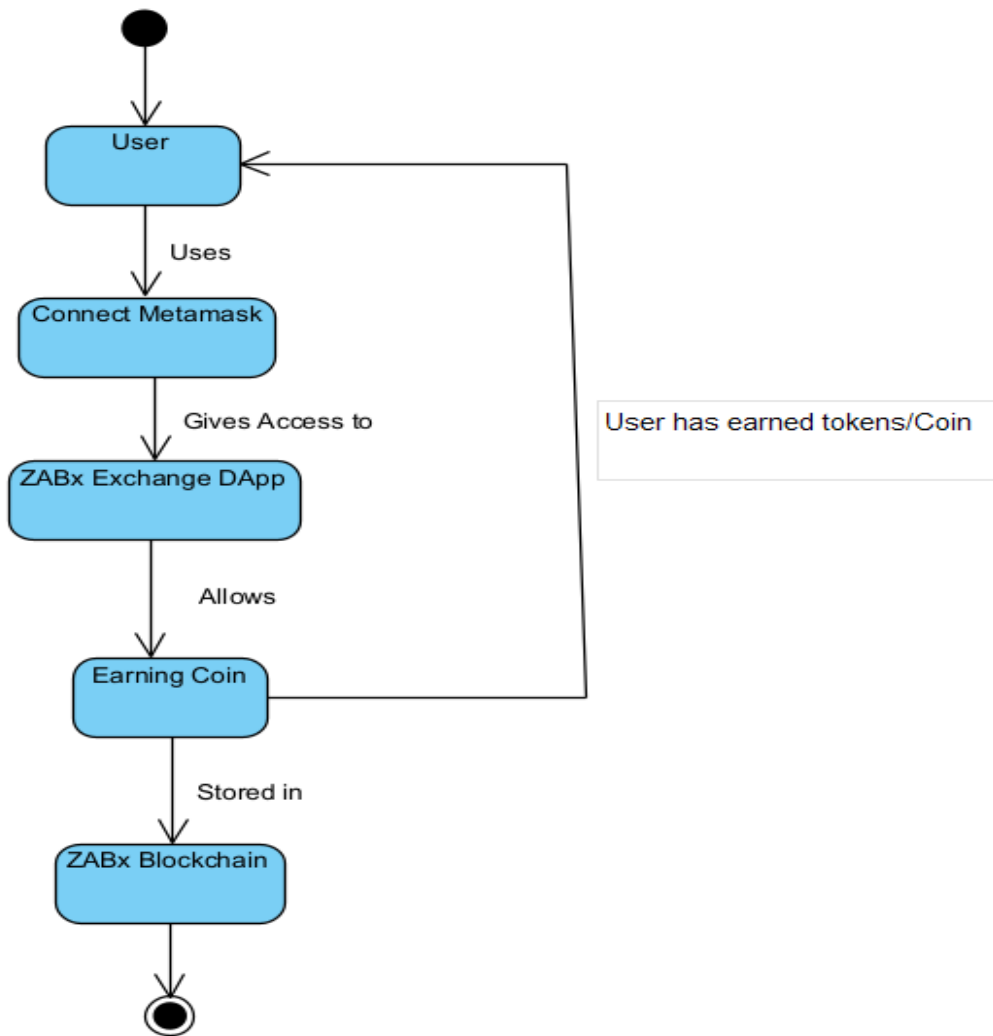


Figure 23. Earning Coin Statechart Diagram

14.3.4. Show Transaction History

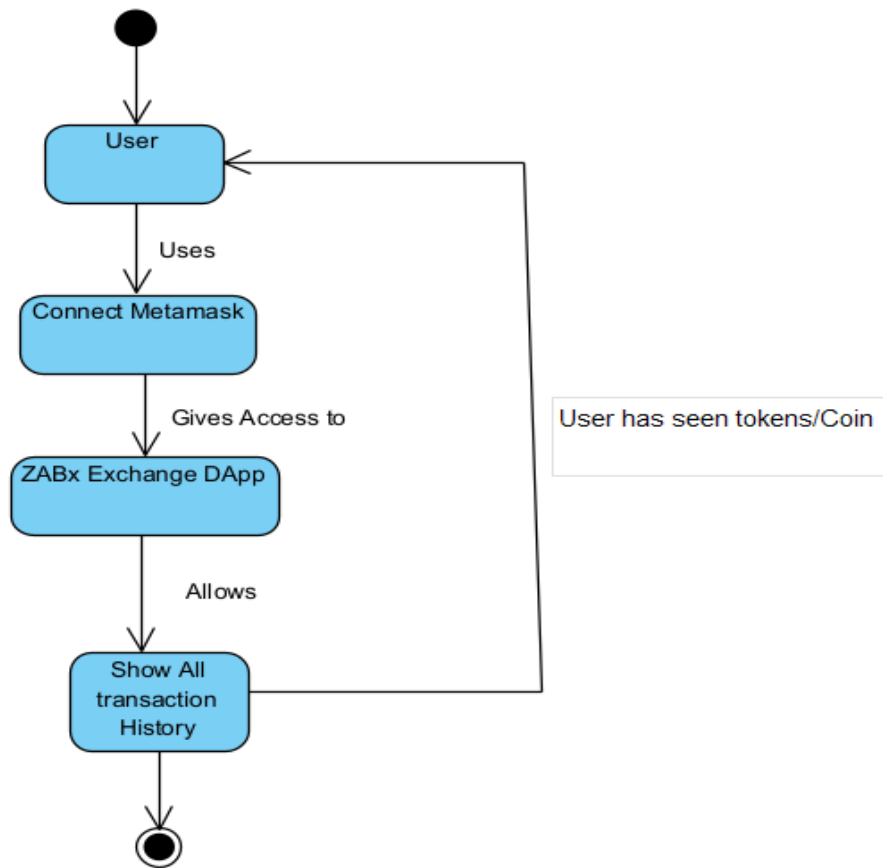


Figure 24. Show Transaction History Statechart Diagram

14.3.5. Transferring Coin

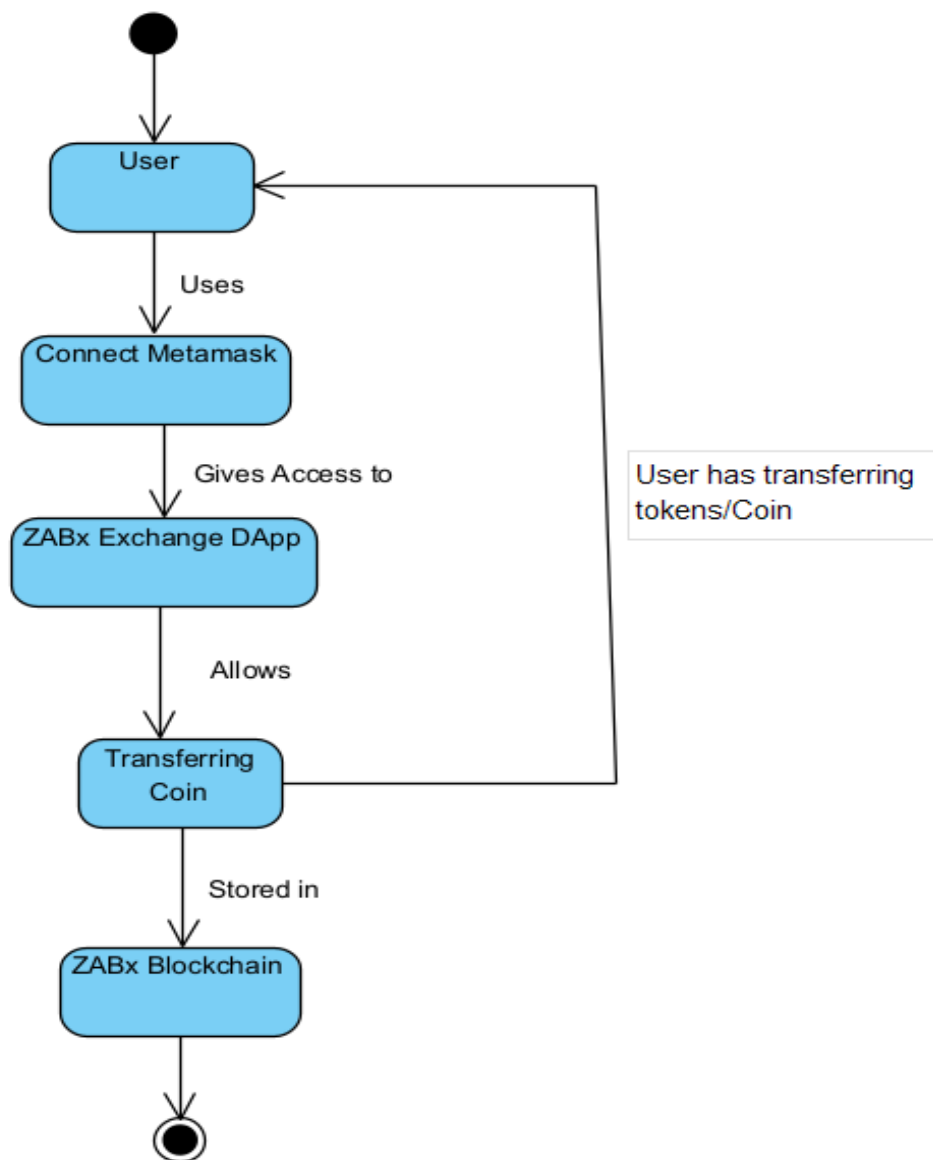


Figure 25. Transferring Coin Statechart Diagram

14.4. Activity Diagram

14.4.1. Buying Coin

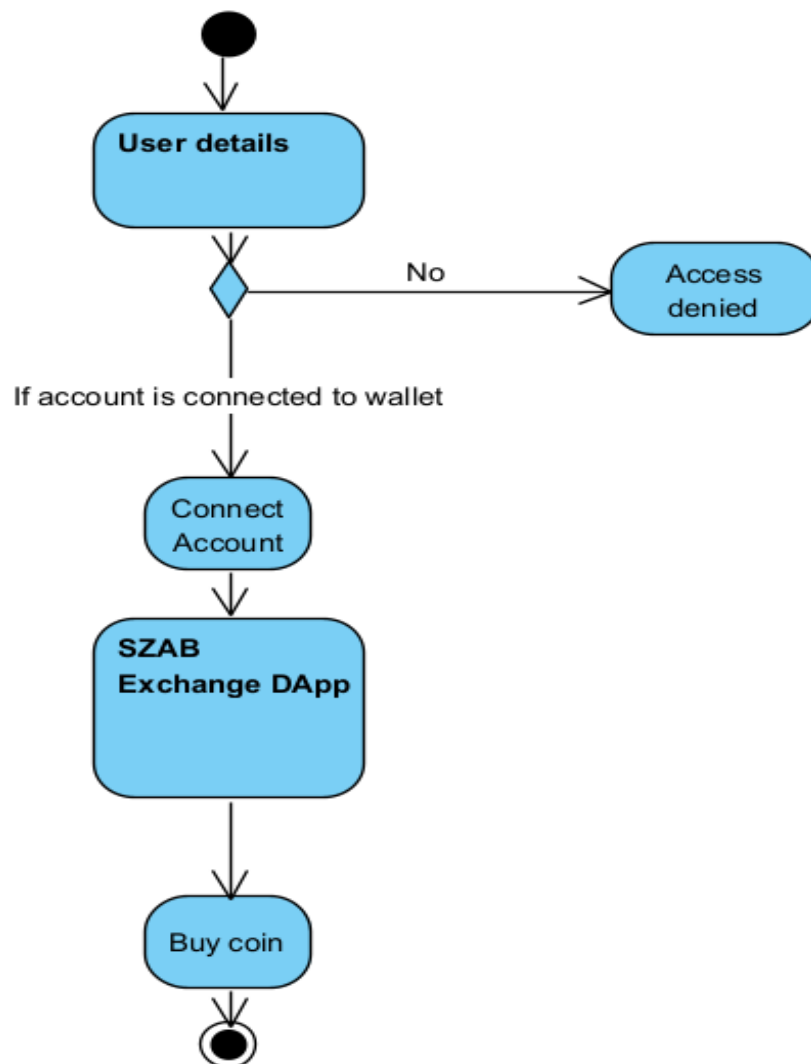


Figure 26. Buying Coin Activity Diagram

14.4.2. Selling Coin

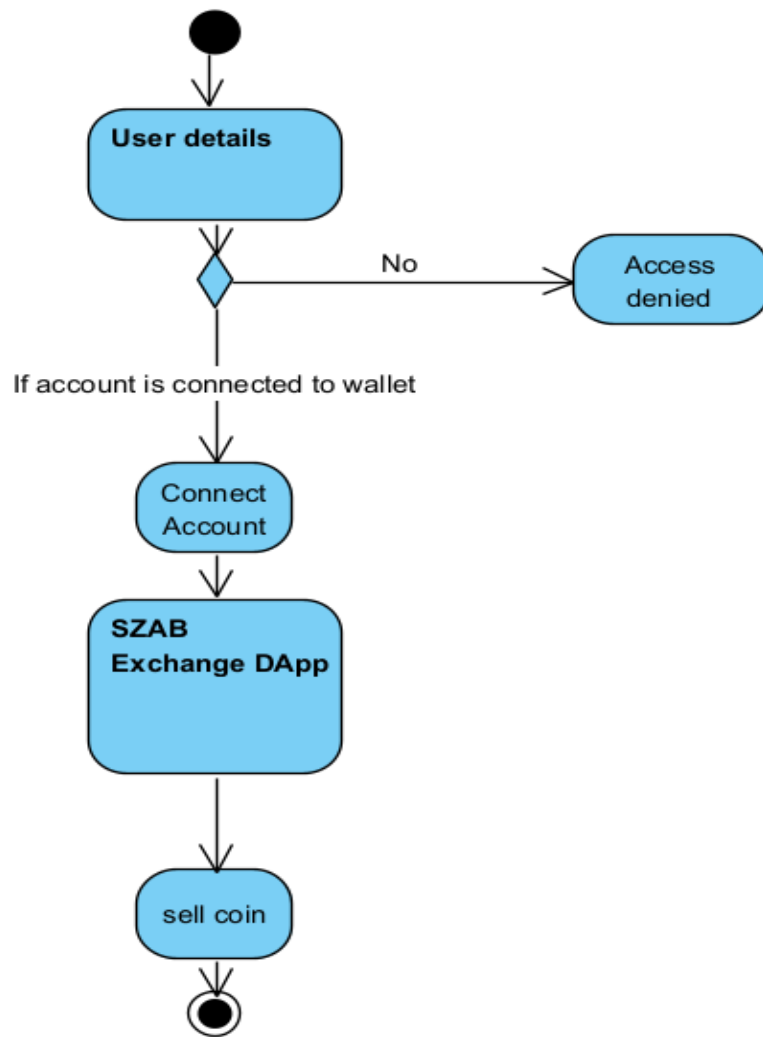


Figure 27. Selling Coin Activity Diagram

14.4.3. Transferring coin

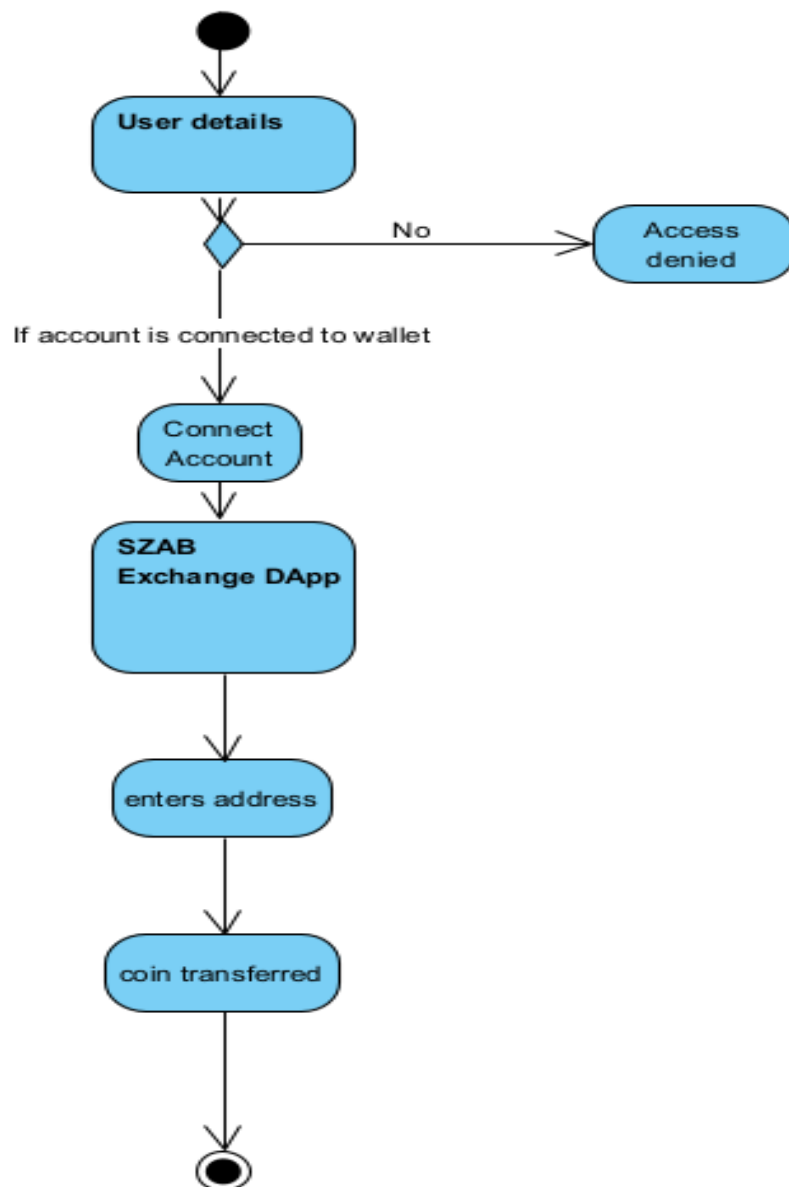


Figure 28. Transferring Coin Activity Diagram

14.4.4. Showing Transaction history

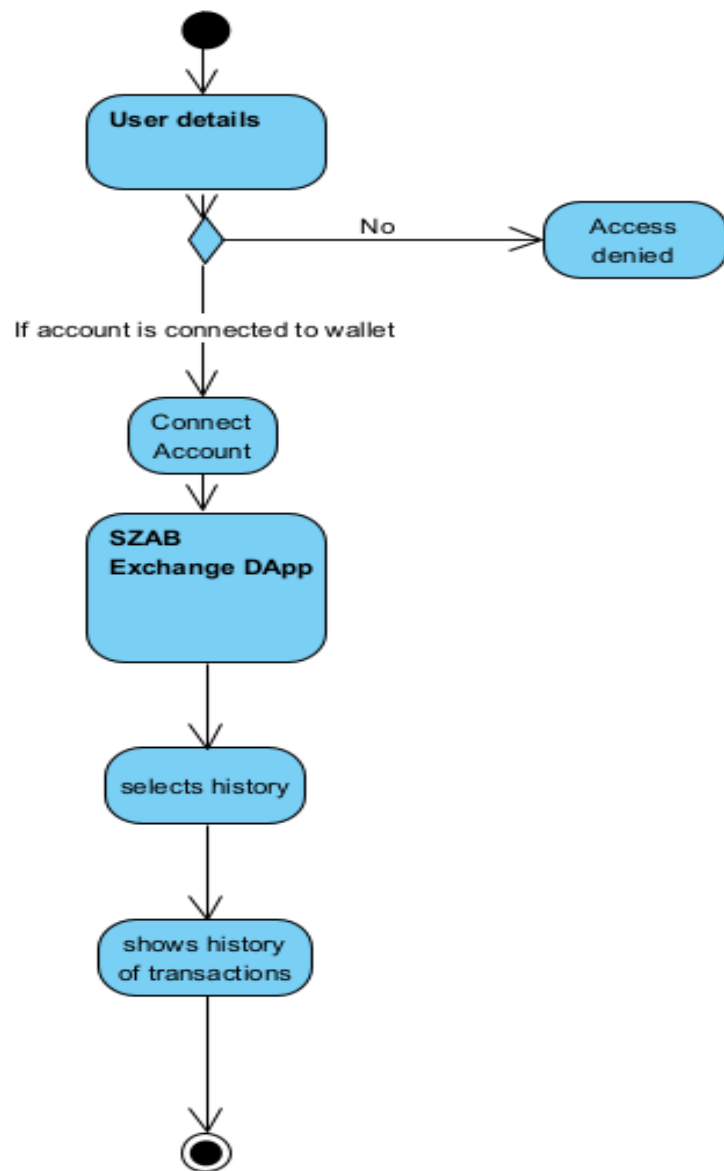


Figure 29. Transaction History Activity Diagram

14.4.5. Earning Coin

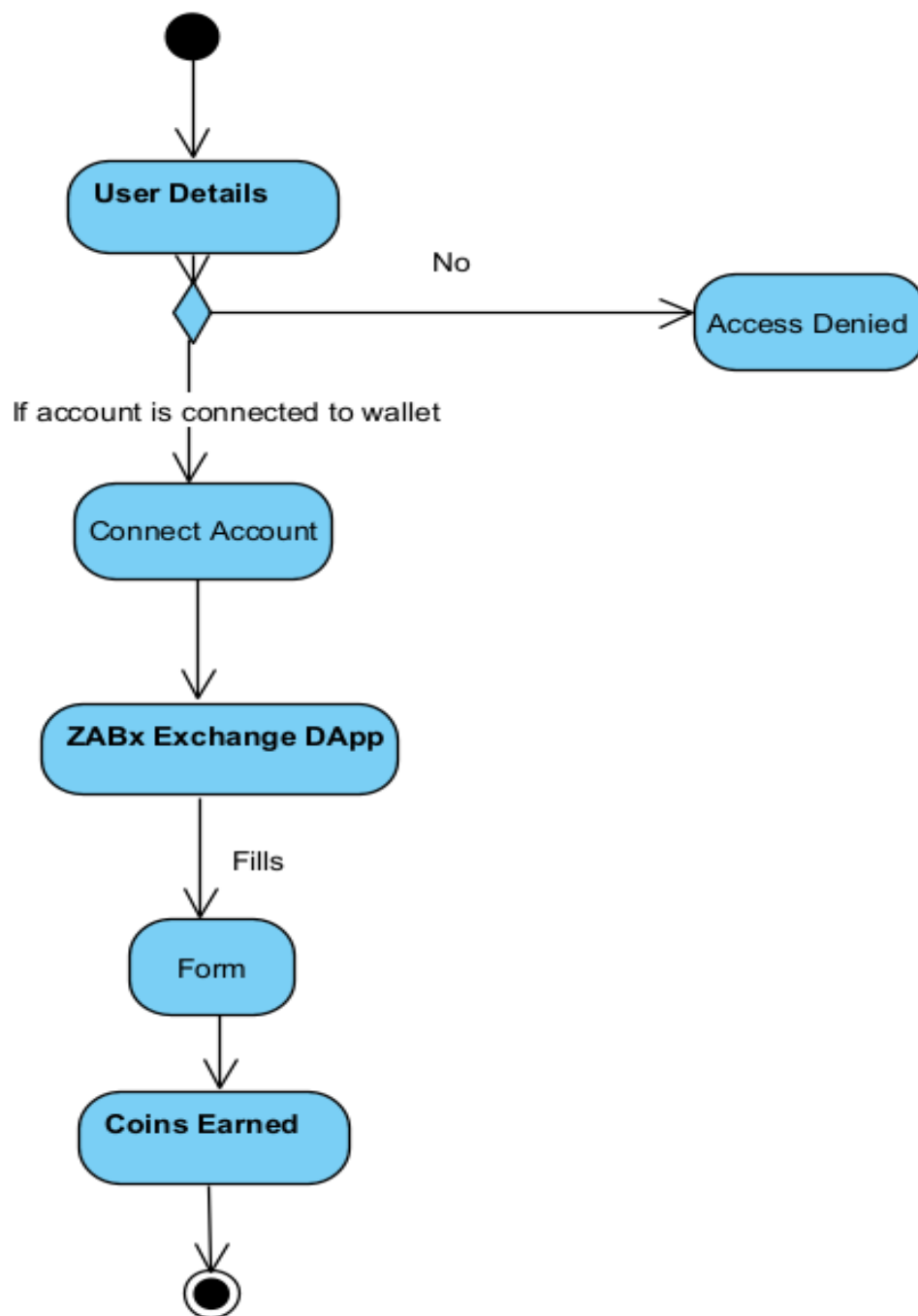


Figure 30. Earning Coin Activity Diagram

14.5. System Sequence Diagram

14.5.1. Buying Coin

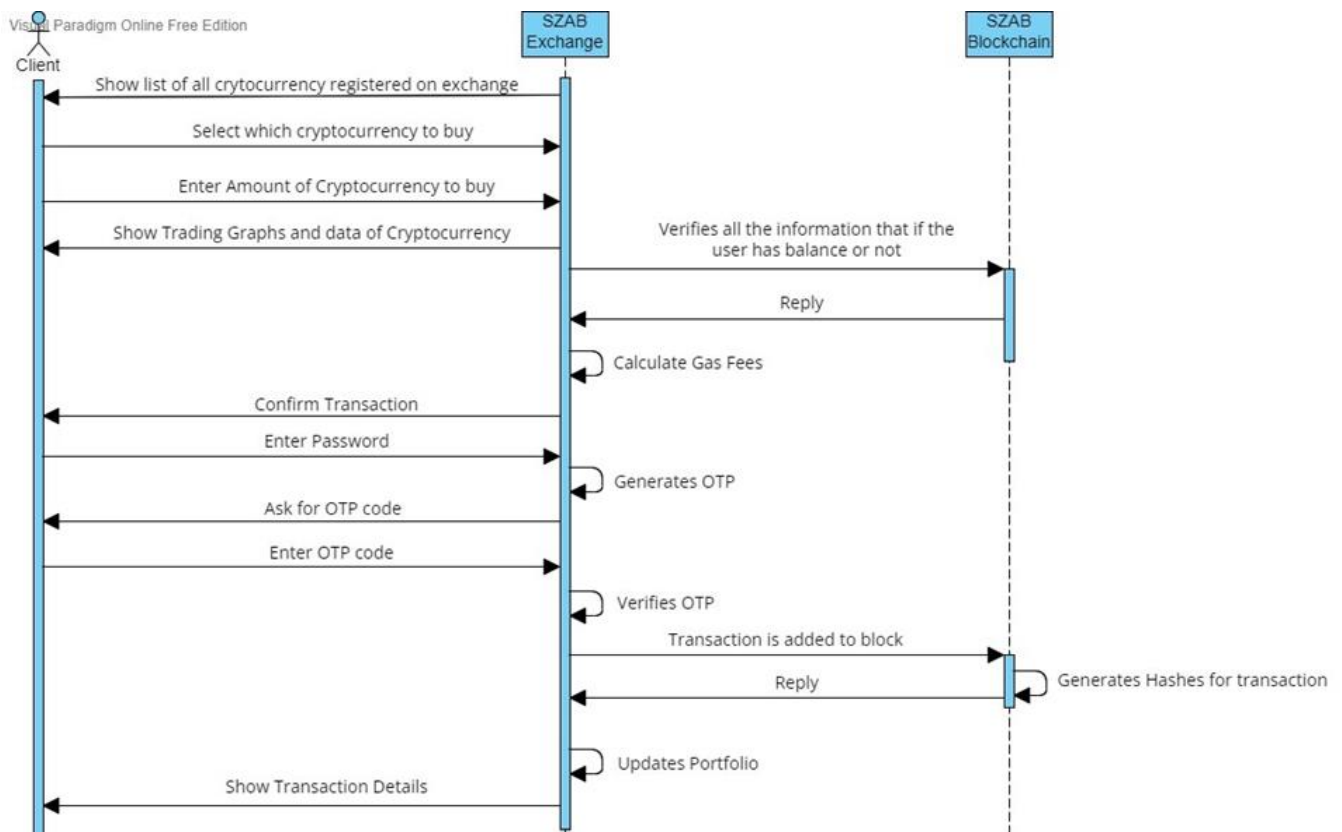


Figure 31. System Sequence Diagram for Buying Coin

14.5.2. Selling Coin

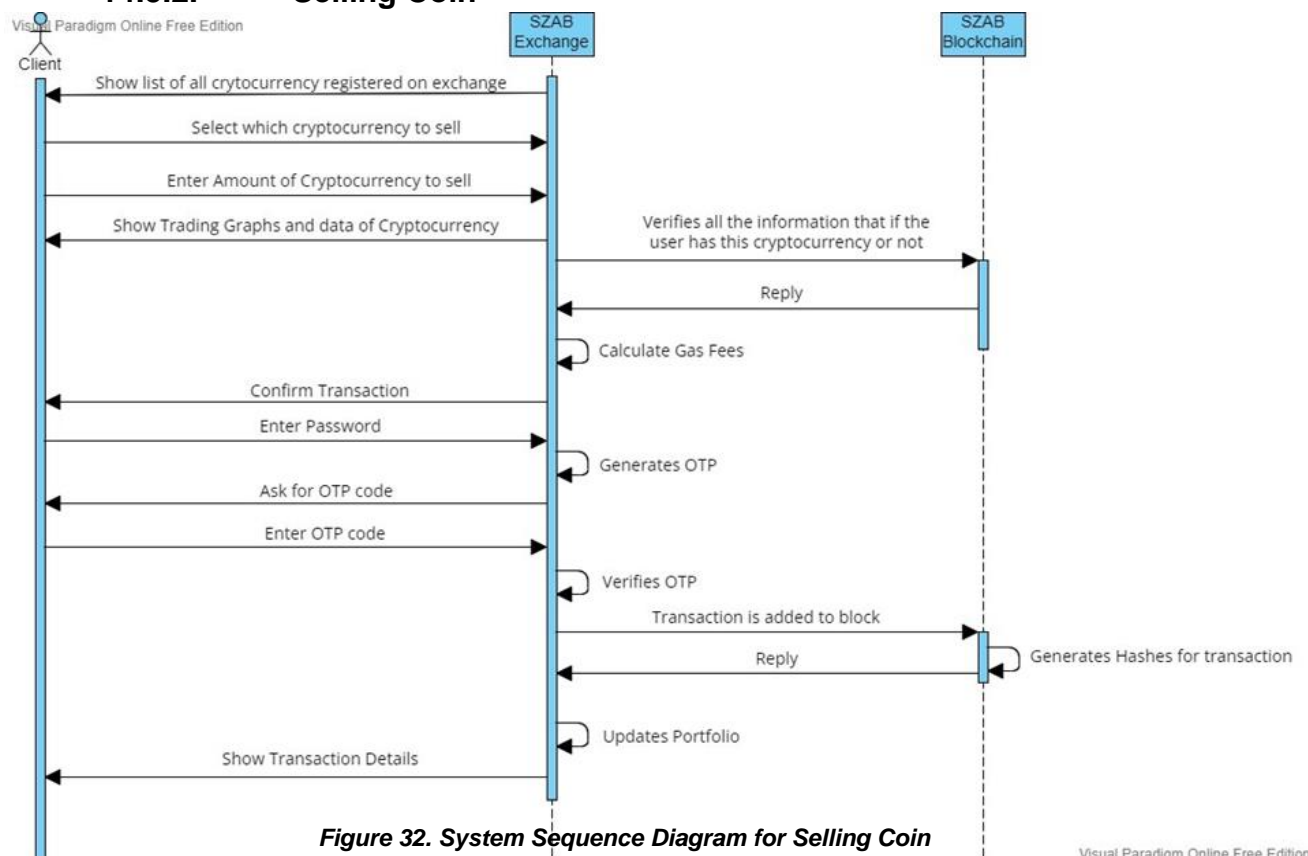


Figure 32. System Sequence Diagram for Selling Coin

14.5.3. Earning Coin

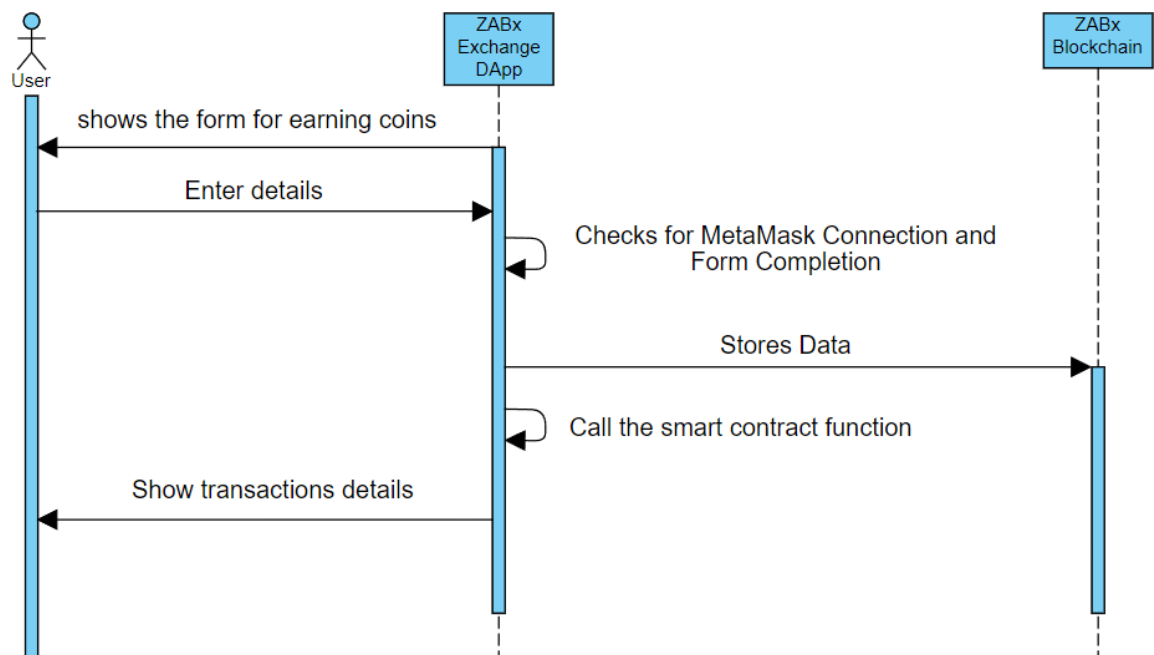


Figure 33. System Sequence Diagram for Earning Coin

14.5.4. Show Transaction History

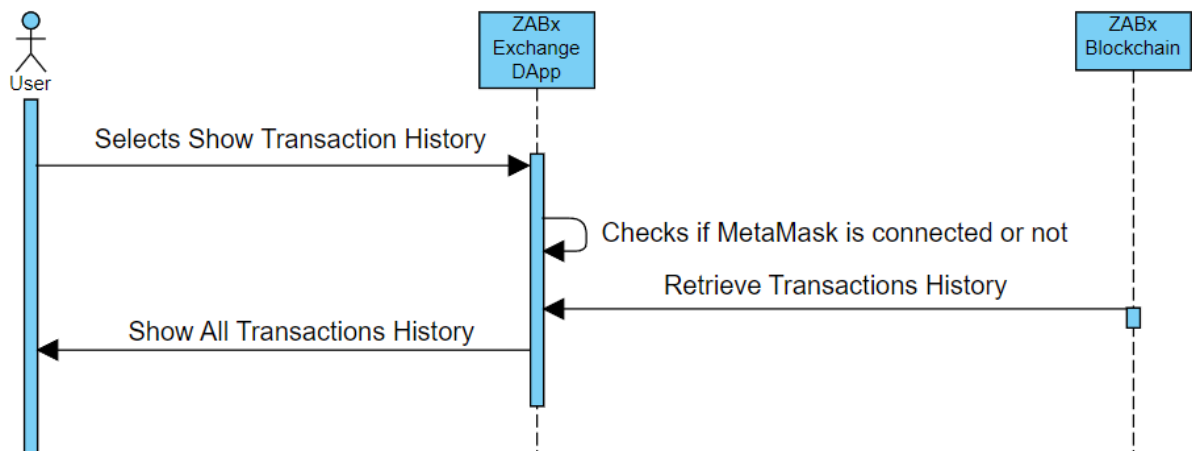


Figure 34. System Sequence Diagram for Showing Transaction History

14.5.5. Transferring Coin

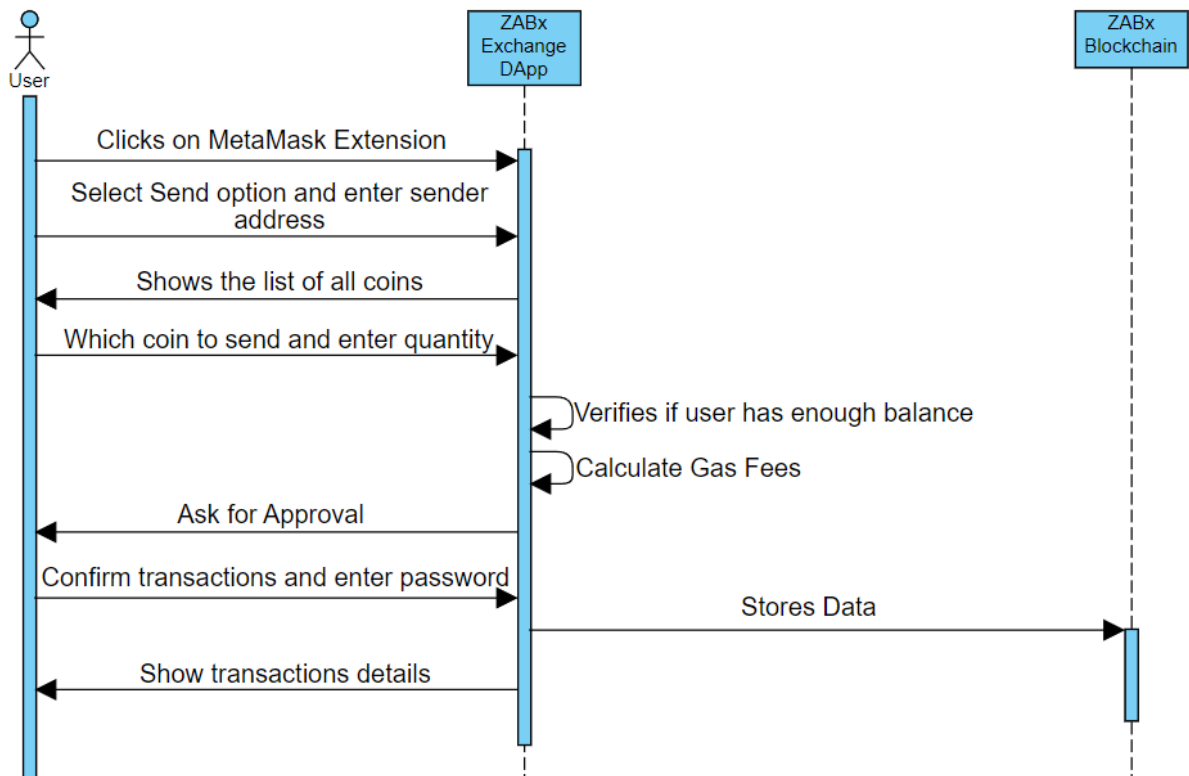


Figure 35. System Sequence Diagram for Transferring Coins

14.5.6. Connect MetaMask

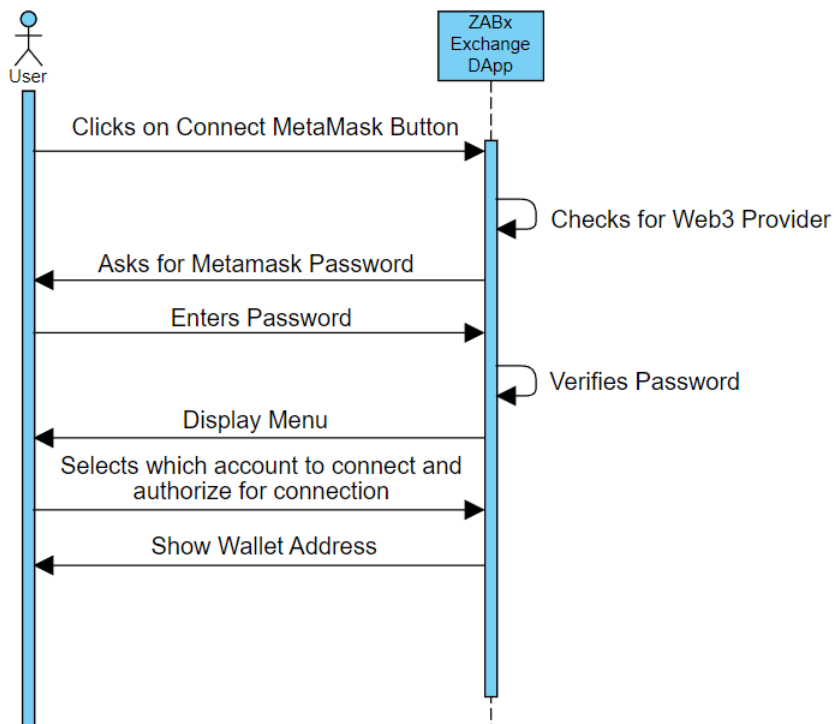


Figure 36. System Sequence System for Connect MetaMask

14.6. Communication Diagram

14.6.1. Buying Coin

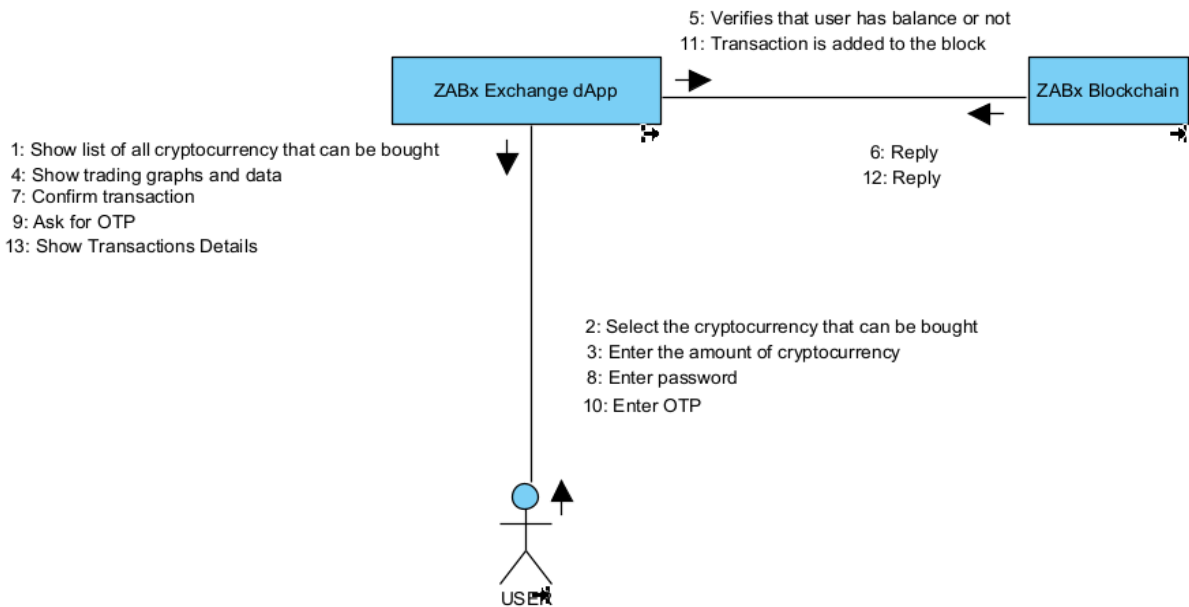


Figure 37. Buying Coin Communication Diagram

14.6.2. Selling Coin

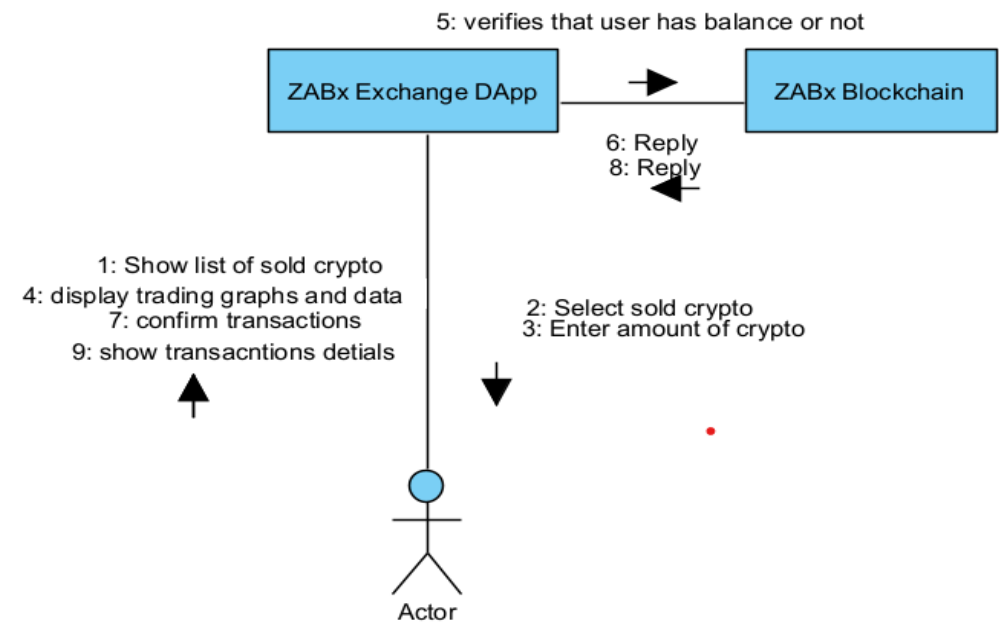


Figure 38. Selling Coin Communication Diagram

14.6.3. Connect MetaMask

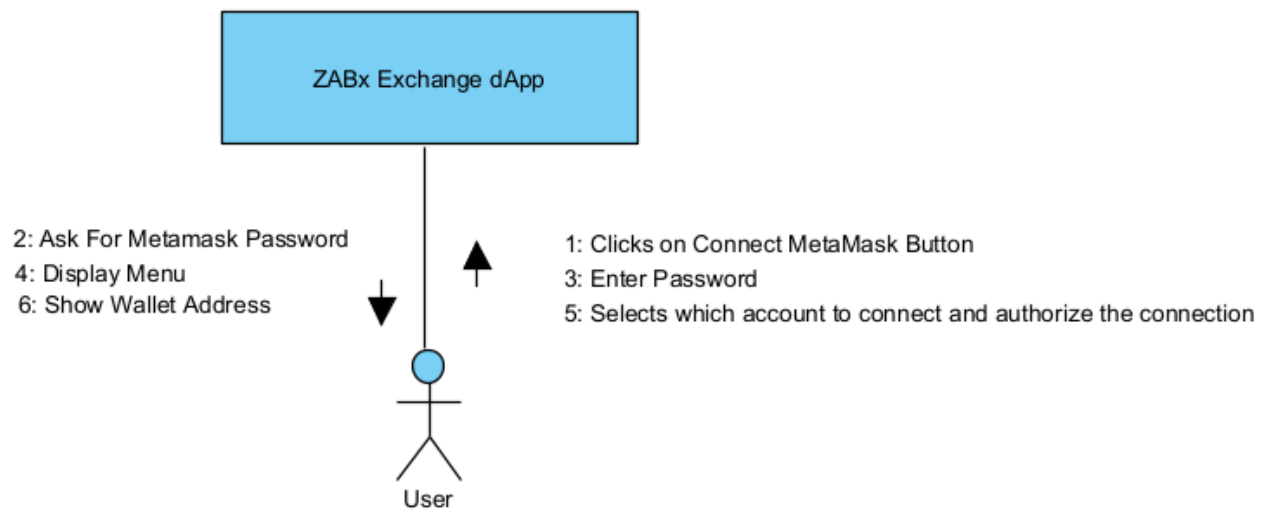


Figure 39. Connect Metamask Communication Diagram

14.6.4. Show All Transactions History

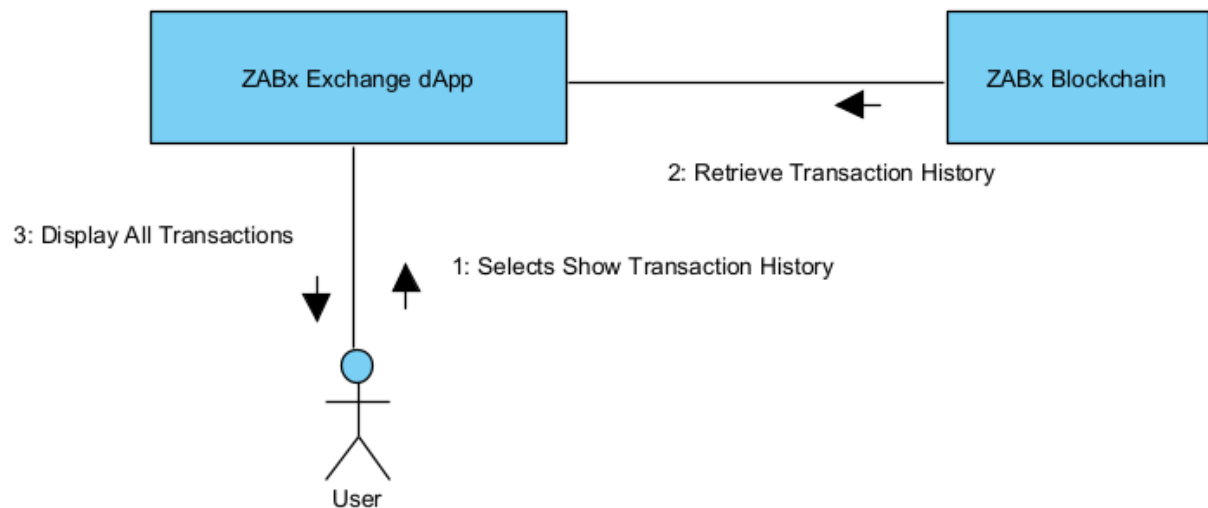


Figure 40. Show All Transactions Communication Diagram

14.6.5. Transferring Coin

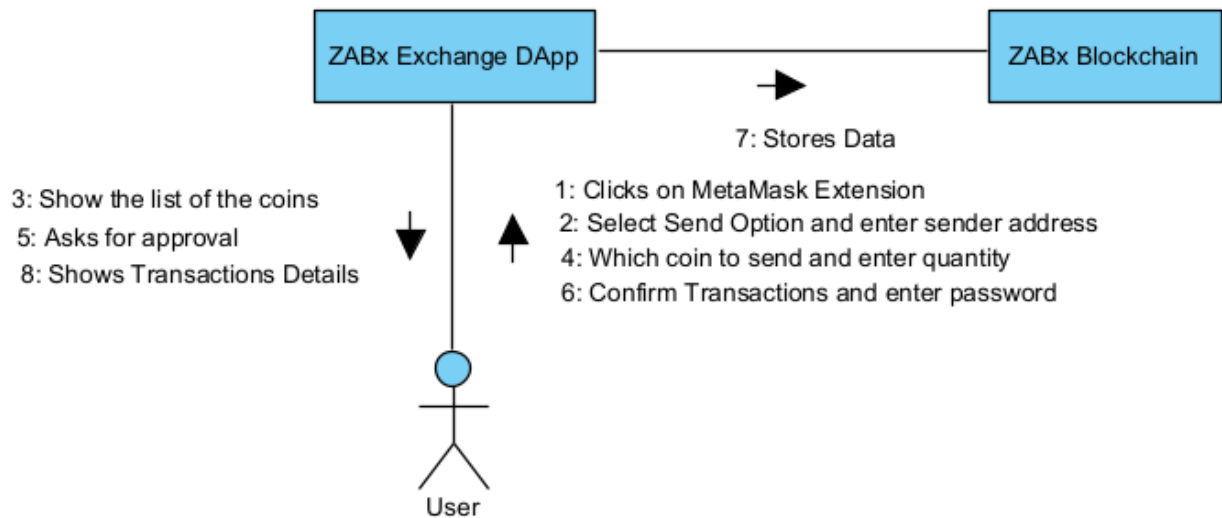


Figure 41. Transferring Coin Communication Diagram

14.6.6. Earning Coin

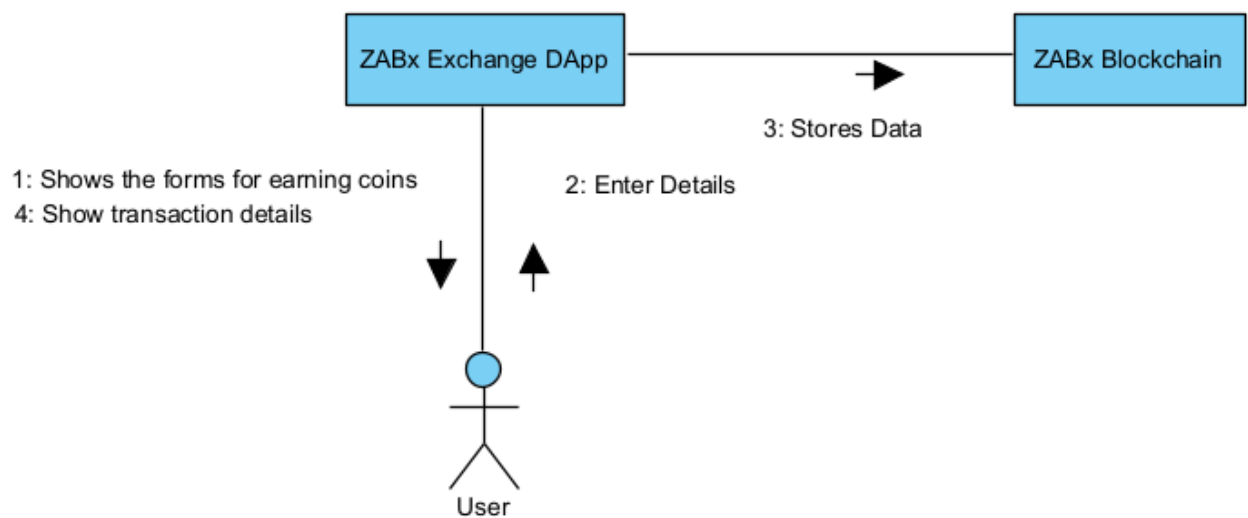


Figure 42. Earning Coin Communication Diagram

14.7. Use Case Diagram

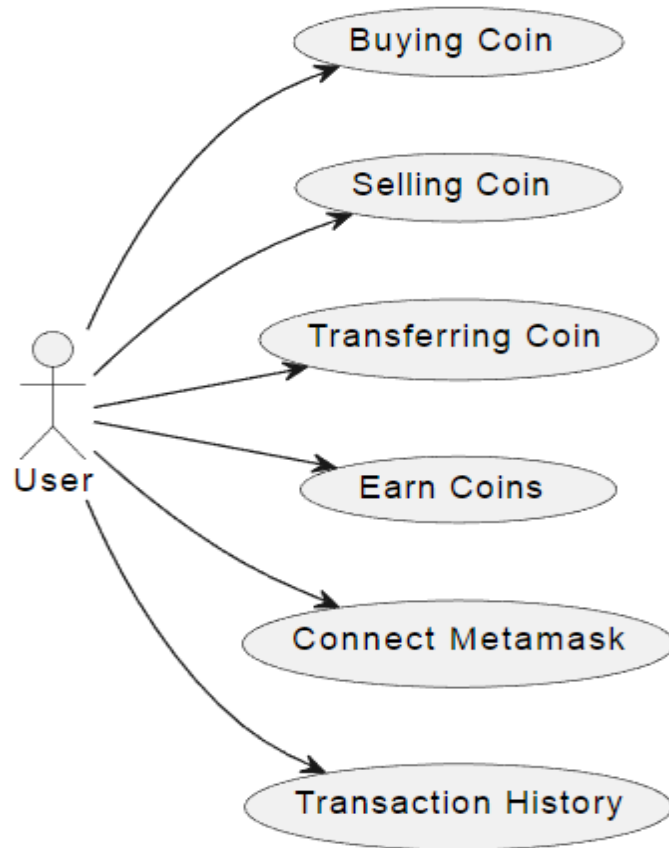


Figure 43. Use Case Diagram

14.8. Component Diagram

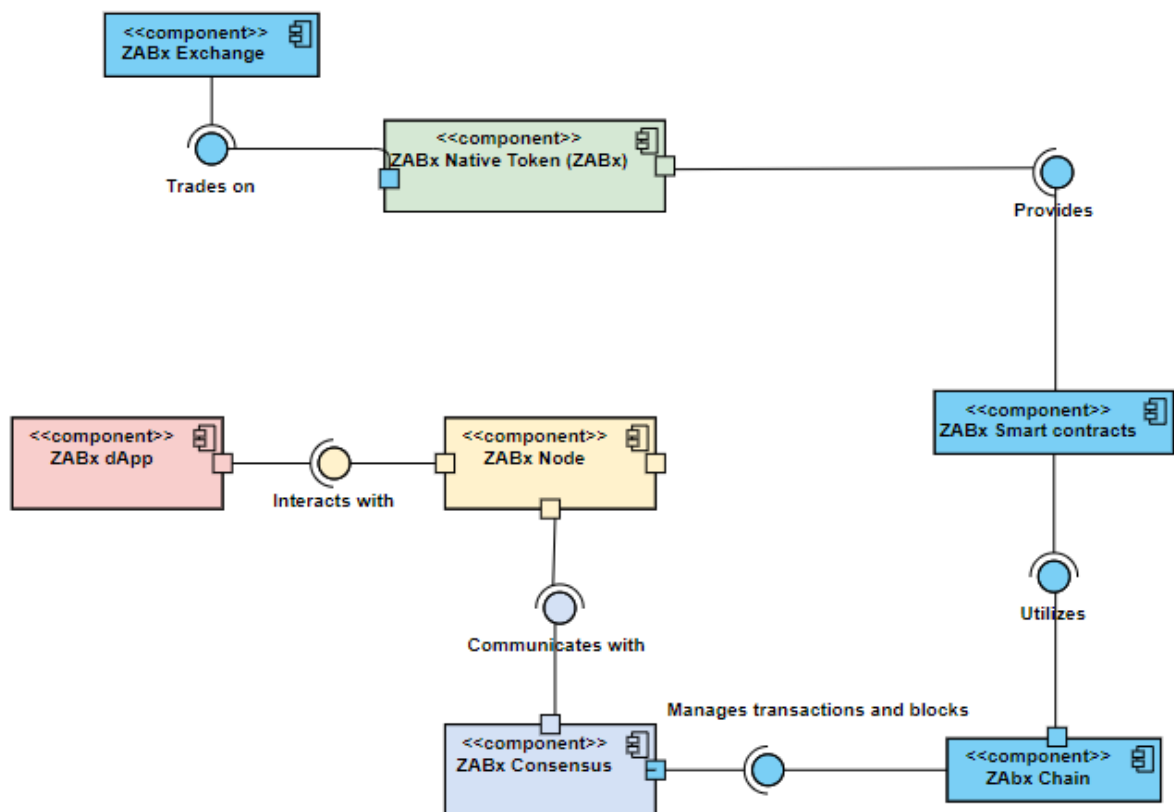


Figure 44. Component Diagram

14.9. Deployment Diagram

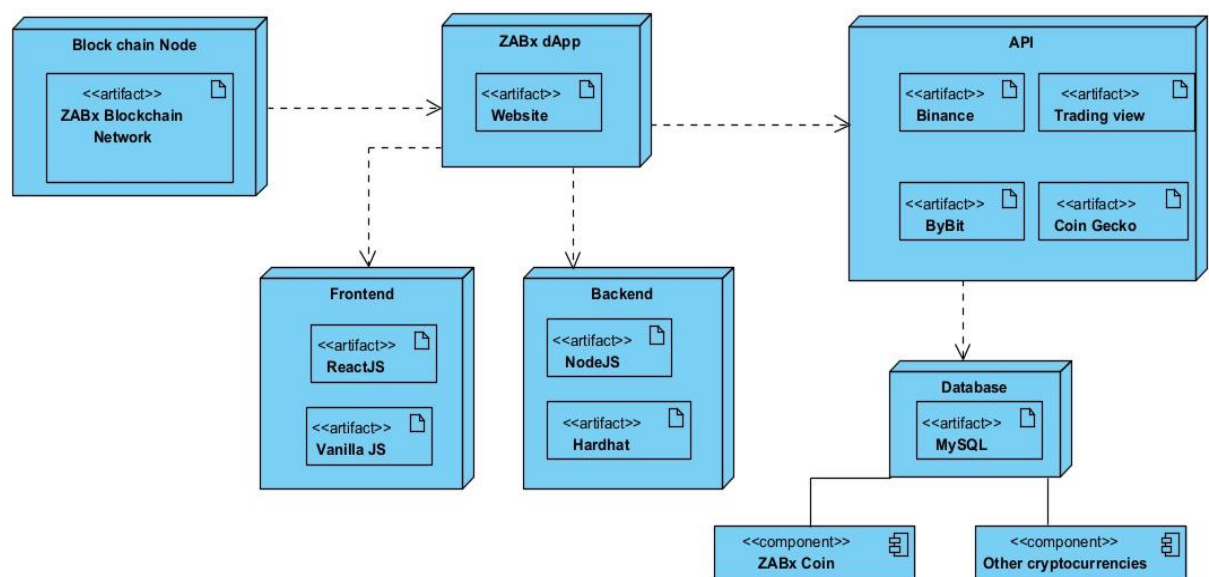


Figure 45. Deployment Diagram

14.10. System Block Diagram

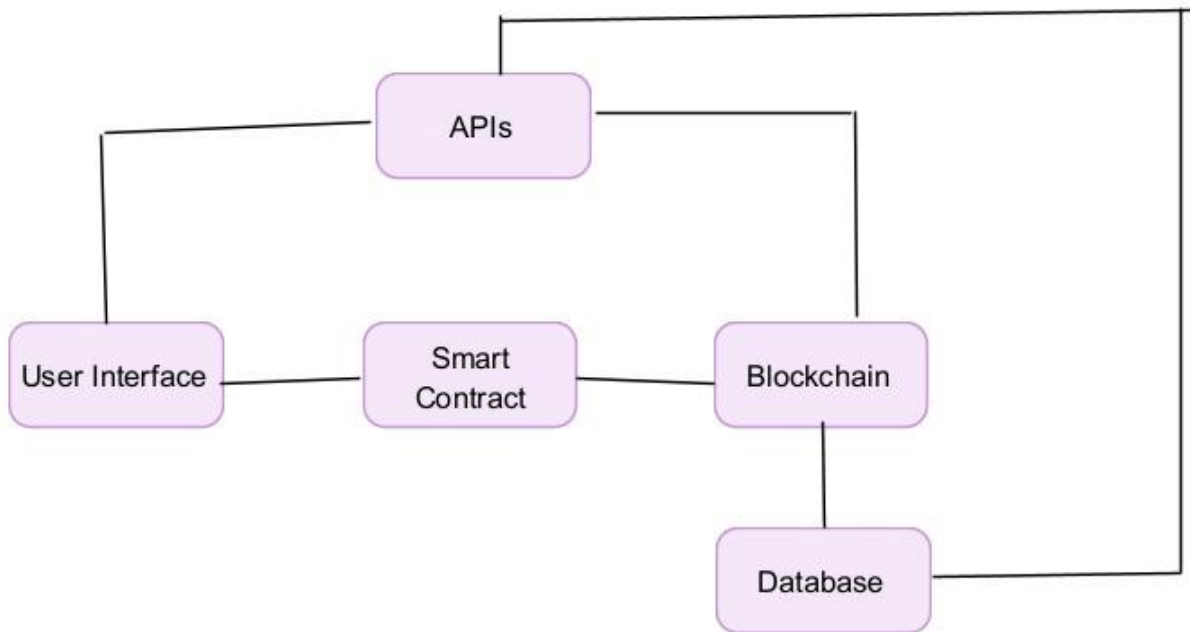


Figure 46. System Block Diagram

15. Testing

15.1. Introduction

Our project is ZABx Blockchain, Coin and Decentralized Crypto Exchange dApp. The main features include connect MetaMask using Web3 JS, buying, selling, transferring and earn coins using smart contracts and APIs and it all is stored our own Public, Layer 2, Public Blockchain The application uses React, Hardhat, EtherJS, NodeJS etc.

15.2. Purpose of this document

Testing is an essential process in software development with the primary purpose of identifying and rectifying defects before software release. It ensures that the software meets requirements, functions as intended, and provides a positive user experience. Testing aims to detect bugs, assure quality, validate and verify software, improve user experience, mitigate risks, and enhance stability and reliability. Additionally, it helps achieve compliance with standards and regulations. Test cases, detailed instructions for verification, play a crucial role in testing. They outline specific scenarios, steps, test data, and expected outcomes to assess the software's behavior accurately. Properly designed test cases aid in effective regression testing and documentation, facilitating the delivery of high-quality software products to end-users.

15.3. Test Cases

We have only used our FYP which is only done in the second part of FYP and all of them are listed below:

- Connect MetaMask
- Cryptocurrency

All of these test cases are shown in this document below:

Test ID	Test Case name	Test case summary	Test case steps	Expected results	Actual result	Pass/Fail
1	Connection of Metamask	This test case verifies the functionality of a user connecting to the ZABx wallet.	Launch the ZABx wallet application.	The application should launch successfully without any errors.	The user has successfully connected to the ZABx wallet application.	Pass
			On the login screen, enter valid credentials (username and password) of an existing user.	The user should be able to enter valid login credentials.	The user's dashboard is displayed, showing relevant wallet information.	Pass
			Click on the "Login" button.	Upon clicking the "Login" button, the user should be authenticated, and the user dashboard should be displayed.	The user's wallet balance is accurately displayed, reflecting the correct amount of ZABx coins and other cryptocurrencies held by the user.	Pass
			The application should authenticate the user credentials and navigate to the user dashboard.	The wallet balance should be shown accurately, reflecting the user's ZABx coin and other cryptocurrency holdings.	The recent transaction history associated with the user's wallet is visible and accurately shows successful transactions.	Pass
			Test that the user	The user should be able	The user can log out from the	Pass

			can log out from the wallet.	to log out from the wallet, and the application should return to the login screen.	ZABx wallet, and the application returns to the login screen.	
--	--	--	------------------------------	--	---	--

Table 2. Connect MetaMask Test Case

Test ID	Test Case name	Test case summary	Test case steps	Expected results	Actual result	Pass/Fail
2	Cryptocurrency	This test case is used to check the cryptocurrency functionality	User should have a Metamask account	User should be connected to wallet	Digital wallet is opened	Pass
			User should buy the coin	Coin should be transfer in wallet	A successful message is displayed	Pass
			User can sell the coin	The deducted coins should be showed and also the amount in which it is sold	A successful transaction message	Pass
			User can transfer coin to another user	The dApp will transfer coin to another user	A transaction slip is displayed	Pass
			User can check the transaction information	The transaction data should be showed	The transaction history is displayed	Pass

Table 3. Cryptocurrency Test Case

16. User Manual

16.1. Home Page

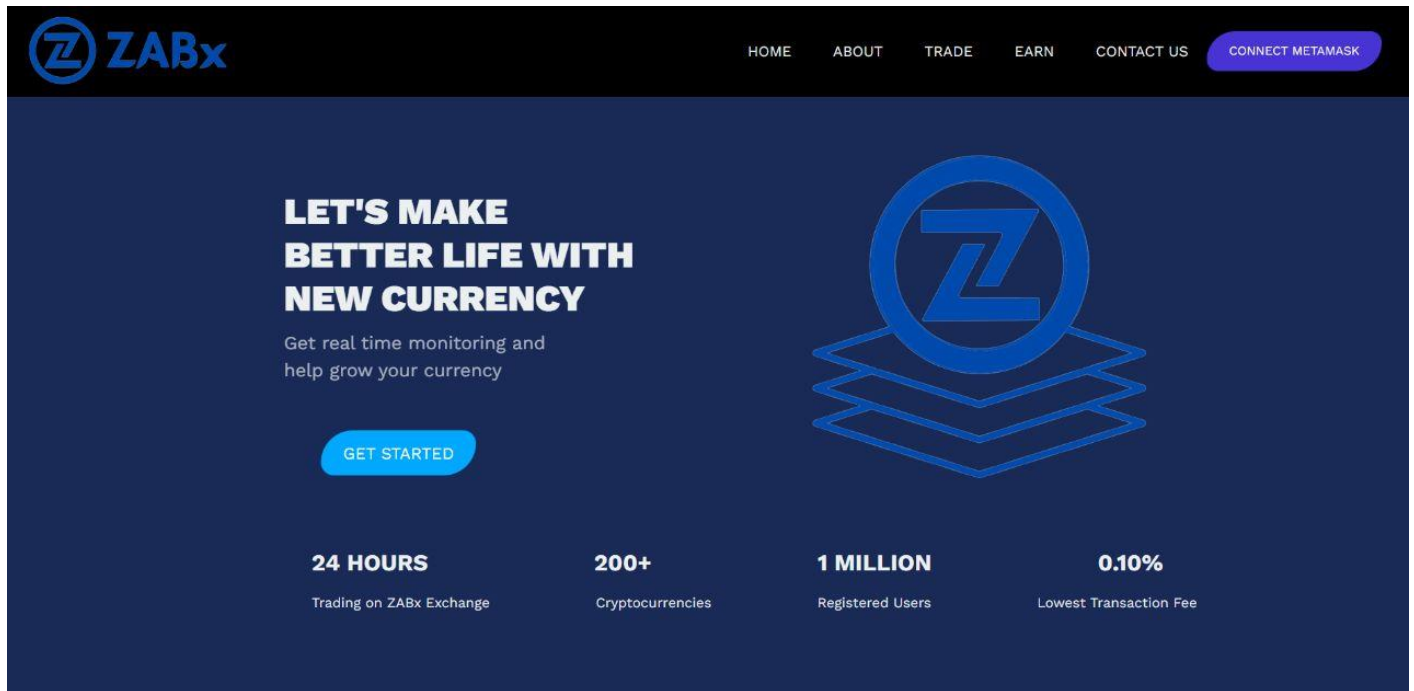


Figure 47. Home Page

- Main page promoting the start of the crypto journey with a prominent "Get Started" button
- Requirement to connect to Metamask, a popular cryptocurrency wallet, to proceed with the journey
- Navigation bar featuring key sections: Home, About, Trade, Earn, and Contact Us
- Seamless integration with Metamask for easy and secure access to the Dapp's features
- Clear and user-friendly interface to ensure a smooth onboarding process for beginners in the crypto space.

16.2. Crypto News

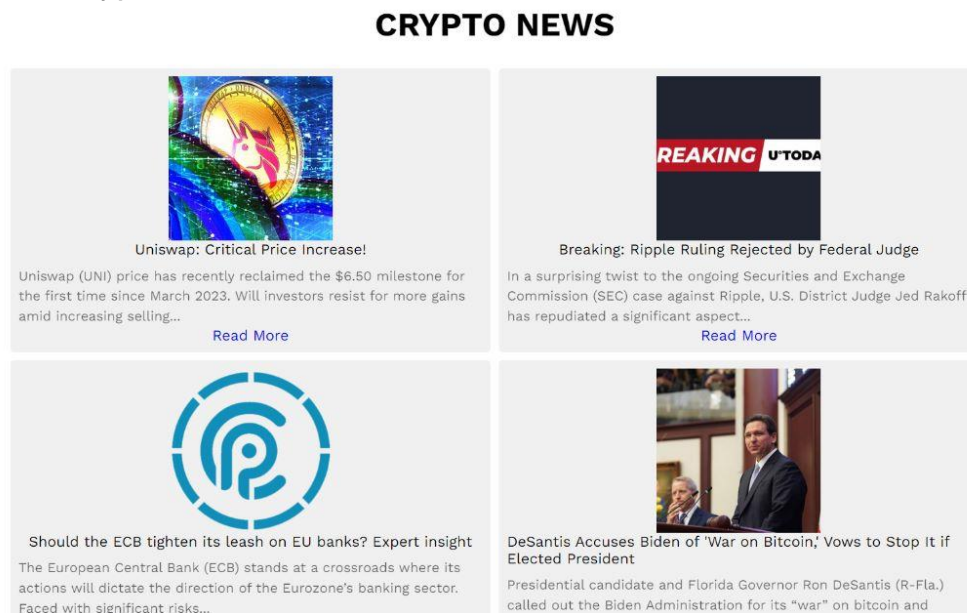


Figure 48. Crypto News

- The Dapp offers authentic news about cryptocurrencies to users.
- Users can access real-time and reliable news directly within the Dapp.
- This feature helps traders stay informed about the latest developments in the cryptocurrency market.
- Having access to trustworthy news enhances the trading experience and decision-making process.
- Users can make well-informed decisions based on the latest information and market trends.
- The Dapp ensures that users have all the necessary news and updates at their fingertips while trading on the platform.
- By providing news within the Dapp, users can save time and effort, avoiding the need to switch between different sources for information.

16.3. Earn Coins

ZABx HOME ABOUT TRADE EARN CONTACT US CONNECT METAMASK

To Earn Free Coins

Name
Areeb

Email
cs1912231@szabist.pk

Phone Number
3002756664

Twitter URL
https://twitter.com/areebfaisal99

Telegram Username
AreebFaisal

Not Connected

CLAIM YOUR REWARD

Figure 49. Earn Coins Page

- The UI displays an option to earn free ZAB coins, a cryptocurrency, by filling in specific fields.
- The required fields for earning the reward are Name, Email, Phone Number, Twitter URL, and Telegram Username.
- After filling in the required fields, users can click on the "Claim Your Reward" button to initiate the process.
- To receive the earned coins, users must be connected to their cryptocurrency wallet.
- If the user is not connected to their wallet, they will be prompted to connect before proceeding with the claim.
- The process ensures that only users with connected wallets can receive the ZAB coins as a reward.
- This feature encourages user engagement and participation in the Dapp's ecosystem while promoting the use and adoption of the ZAB coin.
- The Dapp's rewarding mechanism adds an incentive for users to provide accurate information and promotes a sense of trust and transparency in the platform.

16.4. Trading

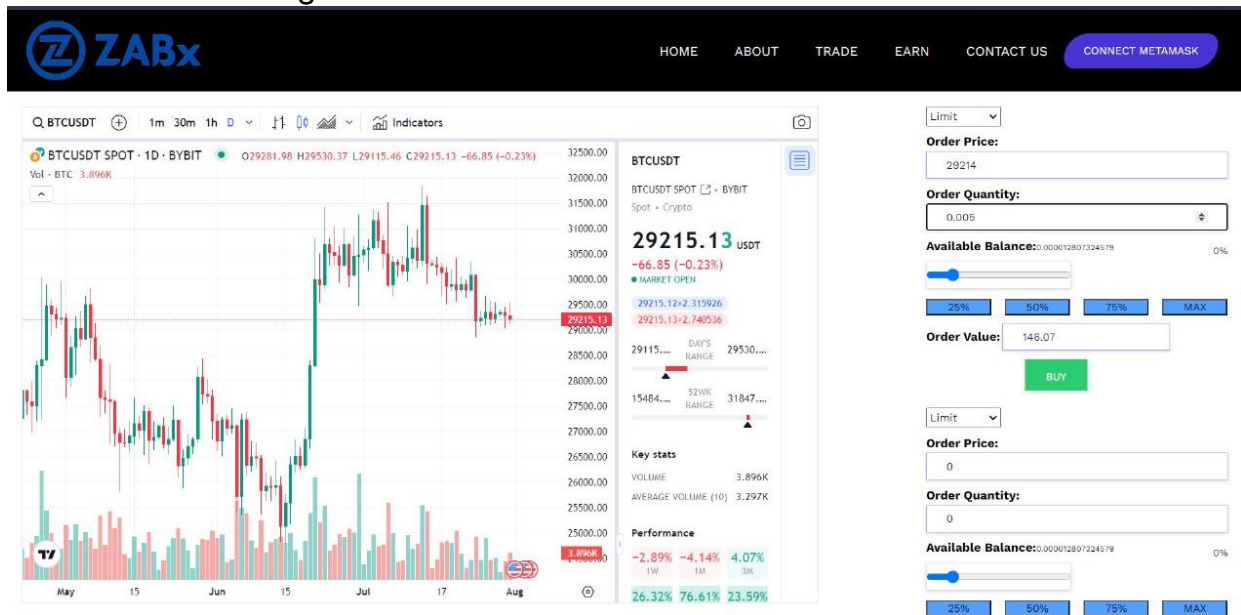


Figure 50. Trading Page

- The UI displays a trading chart for a cryptocurrency, providing users with visual insights into its price trends and historical data.
- Users can analyze the chart to make informed trading decisions based on the cryptocurrency's performance.
- The "Buy" button allows users to purchase the cryptocurrency directly from the Dapp.
- Clicking on the "Buy" button opens a user-friendly interface to input the desired quantity and complete the purchase.
- The "Sell" button enables users to sell their existing cryptocurrency holdings through the Dapp.
- Clicking on the "Sell" button initiates the selling process, and users can input the quantity they wish to sell.
- The trading chart and buttons are designed for easy navigation and seamless trading experience within the Dapp.
- The Dapp ensures that all transactions are secure and transparent, offering a reliable platform for users to trade cryptocurrencies conveniently.
- By providing a trading chart and straightforward buy and sell buttons, the Dapp empowers users to actively participate in the cryptocurrency market with ease.

16.5. Searching cryptocurrency

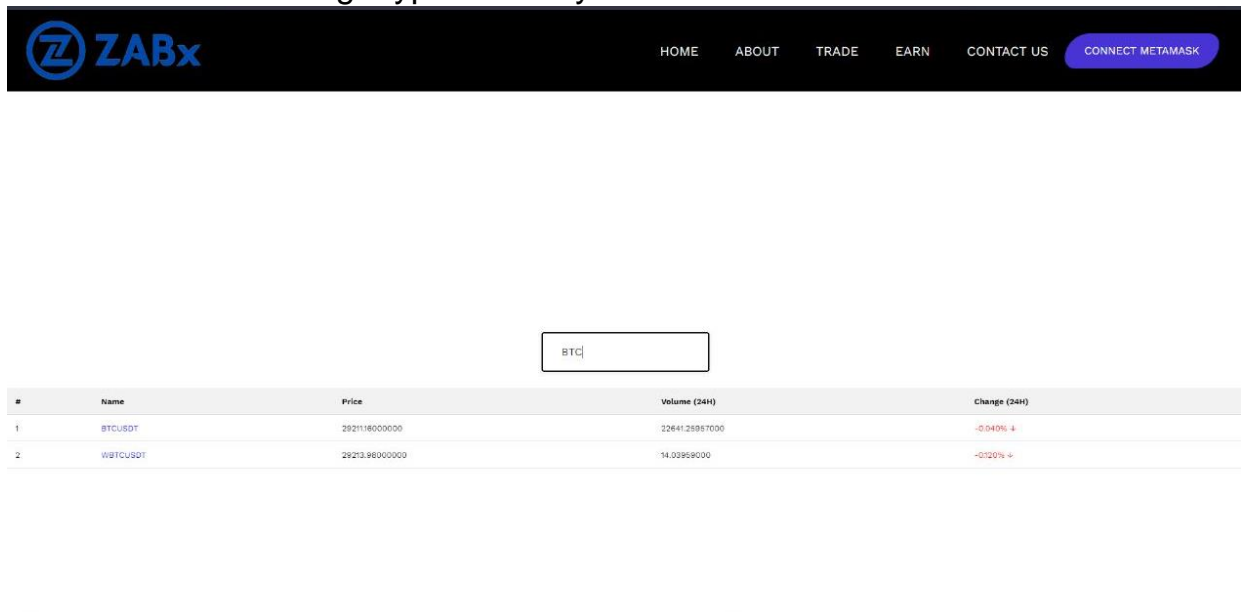
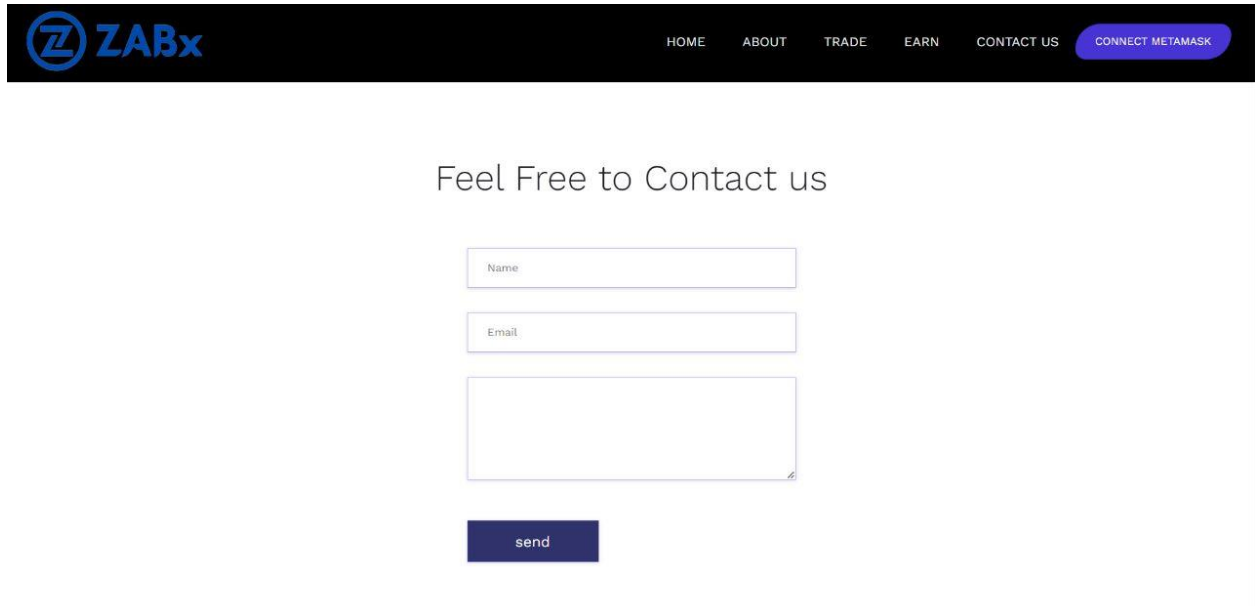


Figure 51. Searching Coins

- The UI features a search option that allows users to search for specific cryptocurrencies.
- Users can input the name or symbol of the cryptocurrency they want to search for.
- Upon entering the search query, the Dapp displays detailed information about the searched cryptocurrency, including its current price and trading volume.
- This feature helps users quickly find and access relevant information about their desired cryptocurrencies.
- The search functionality enhances the user experience by providing easy access to data without the need for manual browsing or scrolling through the entire list of cryptocurrencies.
- Users can use the search option to monitor and track multiple cryptocurrencies efficiently.
- The displayed information, such as prices and volumes, helps users make well-informed decisions regarding their crypto investments and trading strategies.

16.6. Contact



Feel Free to Contact us

Name

Email

send

Figure 52. Contact Us Page

- The UI provides a contact form for users to reach out to the owners of the Dapp.
- The contact form includes fields for users to enter their Name, Email, and Message.
- Once users have filled in the required information, they can click the "Send" button to submit their message.
- The Dapp's owners will receive the submitted message in their designated communication channel

17. Iteration Plan

17.1. FYP 1


S.No.	Features	FYP-I Iterations			
		Monthly Iteration-I	Monthly Iteration-II	Monthly Iteration-III	Monthly Iteration-IV
F1	Connect Metamsk	Requirements			
		Design			
		Implementation			
			Testing		
F2	Buying Coin	Requirements (50%)	Requirements (100%)		
			Design		
			Implementation(100%)		
					Testing
F3	Selling coin		Requirements		
			Design		
			Implementation		
F4	Transferring Coin				Requirements(100%)
			Design(50%)	Design(100%)	
				Implementation(50%)	
					Implementation
...
Output Features		F1	F2, F3,F4	F3	F2, F4

17.2. FYP 2

S.No.	Features	FYP-II Iterations			
		Monthly Iteration-I	Monthly Iteration-II	Monthly Iteration-III	Monthly Iteration-IV
F1	Earn coins	Requirements			
			Design		
			Implementation		
				Testing	
F2	Transaction History	Requirements (50%)	Requirements (100%)		
				Design	
				Implementation(100%)	
					Testing
...
Output Features		F1,F2	F1,F2	F1,F2	F2

18. Meeting Log Form

18.1. FYP 1



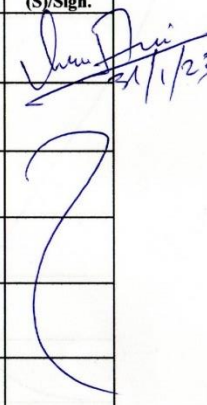
SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE & TECHNOLOGY KARACHI CAMPUS

Form IV: Student Log Form

Title: ZABx Blockchain, Coin and Exchange

Supervisor: Dr Imran Amin Batch/Sec: _____ Group #: 32

Reg. # (Group members): M. Areeb Faisal-1912231 & Fatima Mohiuddin-1912224

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	General discussion meeting	07/10/22	Completed	 21/1/23
2	General discussion meeting 2	18/10/22	Completed	
3	Started working on research	25/10/22	Completed	
4	SRS Discussion	8/11/22	Completed	
5	Discussion on the development of coin and blockchain	15/11/22	Completed	
6	Whitepaper Discussion	22/11/22	Completed	

**SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE &
TECHNOLOGY KARACHI CAMPUS**

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
7	Discussion on post mid task	29/11/22	Completed	
8	Mid term Discussion	02/12/22	Completed	
9	Developing Smart Contract for the coin	13/12/22	Completed	
10	Installation of geth	20/12/22	Completed	
11	Developing smart contract for genesis block	27/12/22	Completed	
12	Collecting faucets	03/01/23	Completed	
13	Developing Node	10/01/23	Completed	
14	Discussion on SRS and SDS	17/01/23	Completed	
15	Final update	30/01/23	Completed	

**SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE &
TECHNOLOGY KARACHI CAMPUS**

Supervisor's Authentication (Completed report):



Dated: 31/1/23

FYP Coordinator Authentication:

Dated: _____

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	General discussion meeting	07/10/22	Completed	
2	Contract discussion meeting 2	13/10/22	Completed	
3	Contract meeting	23/10/22	Completed	
4	SPC Discussion	30/10/22	Completed	
5	Discussion on SRS and SDS	06/11/22	Completed	
6	Whisperer Discussion	20/11/22	Completed	

18.2. FYP 2



SHAHEED ZULFIKAR ALI BHUTTO INSTITUTE OF SCIENCE & TECHNOLOGY KARACHI CAMPUS

Form IV: Student Log Form

Title: ZABx Blockchain, Coin & Exchange

Supervisor: Dr Imran Amin

Batch/Sec: D Group #: 32

Reg. # (Group members): 1912231 , 1912224

Sr.	Task Assigned	Due	Task Completed (S)	Date (S)/Sign.
1	Discussion about Exchange	6/3/23	Completed	<i>Imran</i>
2	Vm Crash	7/3/23	Completed	
3	Coding Update	8/3/23	Completed	
4	NOC Create	13/3/23	Completed	
5	Update about Blockchain and UI	16/3/23	Completed	
6	Mid Update	30/3/23	Completed	



7	Blockchain Update	11/4/23	Completed	
8	Earn feature Update	17/4/23	Completed	
9	Trade feature Update	27/4/23	Completed	
10	Trading list Discussion	9/5/23	Completed	
11	Blockchain Connectivity	16/5/23	Completed	
12	General Discussion	29/5/23	Completed	
13	UI Discussion	5/6/23	Completed	
14	Coin Deployment	9/6/23	Completed	
15	Blockchain and Exchange Discussion	6/7/23	Completed	Jun

Supervisor's Authentication (Completed report):

[Signature]

Dated: _____

FYP Coordinator Authentication: _____

Dated: _____

19. Appendix A: Glossary

N/A

20. Appendix B: Turn it Report

Turnitin Originality Report

1 by Faisal Areeb

From Student (Fall2023)



- Processed on 02-Aug-2023 16:58 PKT
- ID: 2140389417
- Word Count: 10158

Similarity Index

10%

Similarity by Source

Internet Sources:

N/A

Publications:

N/A

Student Papers:

10%

sources:

- 1 1% match (student papers from 06-Jan-2017)
[Submitted to Higher Education Commission Pakistan on 2017-01-06](#)
- 2 1% match (student papers from 26-Jul-2023)
[Submitted to Technological University Dublin on 2023-07-26](#)
- 3 1% match (student papers from 23-May-2023)
[Submitted to WHU - Otto Beisheim School of Management on 2023-05-23](#)
- 4 1% match (student papers from 08-Apr-2021)
[Submitted to Baze University on 2021-04-08](#)
- 5 < 1% match (student papers from 30-Mar-2017)