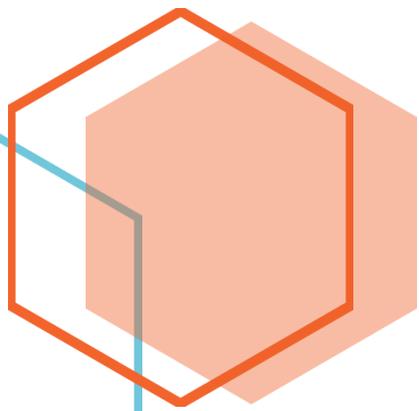




REQUIREMENT SPECIFICATIONS

Requirement Specification Document For Crypto Currency Exchange Project.

This document contents functional and non-functional requirement of crypto currency exchanger project.



Cryptocurrency Exchange

How it works?

Crypto coin exchange platforms match buyers with sellers. Like a traditional stock exchange, traders can opt to buy and sell bitcoin by inputting either a market order or a limit order. When a market order is selected, the trader is authorizing the exchange to trade his coins for the best available price in the online marketplace. With a limit order set, the trader directs the exchange to trade coins for a price below the current ask or above the current bid, depending on whether s/he is buying or selling

Customers

Usually designate crypto participants as either makers or takers. When a buyer or seller places a limit order, the exchange places adds it to its order book until the price is matched by another trader on the opposite end of the transaction. When the price is matched, the buyer or seller who set the limit price is referred to as a maker. A taker is a trader who places a market order that immediately gets filled.

Current Market

All cryptocurrency exchanges have transaction fees that are applied to each completed buy and sell order carried out within the exchange. The fee rate is dependent on the volume of cryptocurrency transactions that is conducted. For example, bitcoin exchange Poloniex has its rate ranging from 0 to 0.25%, GDAX fees range from 0 to 0.30%, Kraken's fees range from 0 to 0.36%, and Paxful charges 1% of the amount of a sale to the seller but buyers don't get charged.

Exchange Facts



A cryptocurrency exchange is a digital marketplace where traders can buy and sell cryptocurrencies using different fiat currencies or altcoins. A crypto currency exchange is an online platform that acts as an intermediary between buyers and sellers of the cryptocurrency.

Cryptocurrency Funds: Funds are pools of professionally managed cryptocurrency assets which allows public buy and hold cryptocurrency via the fund. One such fund is GBTC.

Cryptocurrency Brokers: These are website-based exchanges that are like the currency exchange at an airport. They allow customers to buy and sell cryptocurrencies at a price set by the broker (generally at the market price plus a small premium). Coinbase is an example of this type of exchange.



1. Briefing

1.1. Project description

The objective of this project is the implementation of a web application (including website) which will provide customers a platform for buying and selling crypto currencies. The specifics of this portal are that the customer can view current price of crypto currency, purchase and sell it. Also user can place bid and ask for price for trade. As example to support the programming work for the portal www.cryptopia.co.nz/Exchange, <https://bittrex.com/>, <https://www.zebpay.com/> etc. can be used. The overall objective is to implement a similar portal.

1.2. Realization

This document will specify the detailed requirements which have to be realized in the frame of this project.

Hereby the functionalities and a rough draft of the template/layout will be provided. However basically there are no fixed guidelines provided in terms of the design.

It is requested that the stakeholders of project must bring in own experience and ideas. Please note that a modification of this specification during the implementation phase will be allowed and for the same compensation for efforts will be provided, e.g. especially modification/adding of filter criteria, small design/template modifications, changes which become visible and deem to be necessary during the test phase for logic and marketing/functional purposes have to be additionally implemented. Non-functional requirements, which are part of best practice standards in terms of programming will be seen as self-explaining for the realization of the project, e.g. clean code, documentation, performance, security and design.

If there are open questions during implementation of the API's the individual interface providers will provide contact points. Stakeholder or company will provide necessary credentials and documentation.



2. Definitions

2.1. Buyer

Buyer is portal user who will purchase currency through portal

2.2. Seller

Seller is portal user who will sell crypto currency through portal

2.3. Trade Engine

Trade engine used to define trading strategy of portal and it will decide which transaction / trade will process as per matching criteria

2.4. Order Book

An order book is the list of orders that a trading venue uses to record the interest of buyers and sellers in a particular currency.

2.5. KYC / AML

Know your customer allows tracking user's real identity of user preventing money laundering and frauds

2.6. Hot Wallet / Cold Wallet

Hot wallet is wallet, which stores real cryptocurrency on active node, and it is available for immediate currency transaction. Cold wallet is offline wallet, which will store currencies offline.

2.7. Liquidity

Liquidity means stock of currency, fiat money needed to run the business

2.8. Bid/ Offer

A bid price is the highest price that a buyer (i.e., bidder) is willing to pay for a good. It is usually referred to simply as the "bid".

Ask price, also called offer price, offer, asking price, or simply ask, is the price a seller states they will accept.

3. System environments

The web application (inclusive website) must be available on a test system and on productive environment (LIVE system).

The liquidity providers like banks, currency brokers and suppliers provide for its interfaces test credentials. Based upon these test trades can be made. The LIVE system has to be setup with an SSL certificate. In addition, other security programs added.



4. Details

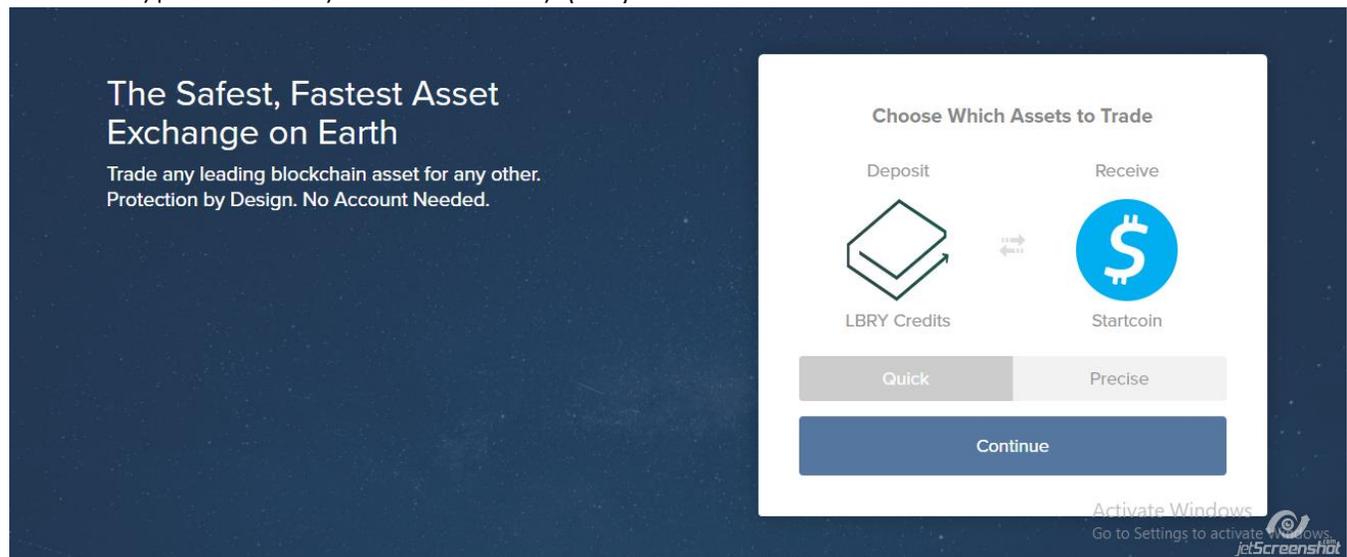
This section should provide an overview of the all the process and the individual functionalities of the web application/website.

4.1. Homepage / Landing Page:

Home page will be consist of following sections:

Home section:

Should demonstrate engaging idea and overview about exchange, it should have current cost of crypto currency in fiat currency (INR).



How It works:

Details about how to purchase currency from this exchange site



EXCHANGER



Recent Trades

Table showing how trades are working here. Moreover, what are recent transactions? It will look something like this image below.

Recent Transactions					
BTC → ALT		ALT → BTC		ALT → ALT	
→ 0.046 BTC to ETH	11 mins ago	→ 0.985 ETH to BTC	5 mins ago	→ 5.292 ETH to XMR	3 mins ago
→ 0.213 BTC to XMR	11 mins ago	→ 0.07 ETH to BTC	6 mins ago	→ 0.285 LTC to FUN	3 mins ago
→ 0.046 BTC to ETH	13 mins ago	→ 0.060 ETH to BTC	6 mins ago	→ 378.531 DGB to RDD	3 mins ago
→ 0.073 BTC to BCH	13 mins ago	→ 1.763 ETH to BTC	6 mins ago	→ 0.019 ETH to DASH	4 mins ago

@jsharps

Please note our base currency is INR and all above pair will be show as crypto currency to INR.

Footer:

Footer of website will include Newsletter section, links for site, Support section.

Subfooter: It will have copyright section and social icons.

4.2. Other Navigations and subpages

FAQ:

This page will include question and answers about portal and how to use it.

Contact Us

Company address, google map integration and contact form will be available on this page

About Us

Details about company , team etc will be provided in this section.

Action Buttons

Sign In and Signup buttons will be available in navigation bar.



4.3. General user flow of this application

User will visit website

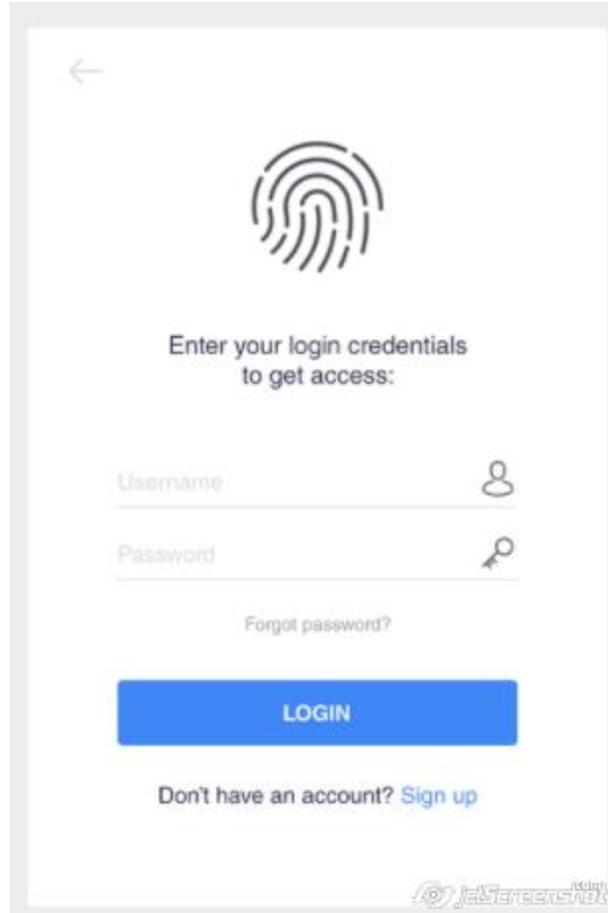
Signup through registration form

A screenshot of a mobile registration form on a blue background. At the top left is a white back arrow. In the center is a white person icon. Below the icon is the text "Hey there! You can start by creating a new account". The form contains four input fields: "Username" with the value "BubuDragos" and a red eye icon, "Email address", "Password", and "Repeat password". Below the fields is a white "SIGN UP" button. At the bottom, there is a link: "Do you have an account? Login". A "jetScreenshots.com" watermark is visible in the bottom right corner.

On successful signup user will get an email to activate an account, once user click on activate account which user will get account activated from DB and user will be redirected to login screen.

On login screen user will be prompt to put username and password provided at the time of registration. For security feature we will add up captcha which will ensure spam attacks can be avoided through Bot machines.

EXCHANGER



←



Enter your login credentials
to get access:

Username 

Password 

[Forgot password?](#)

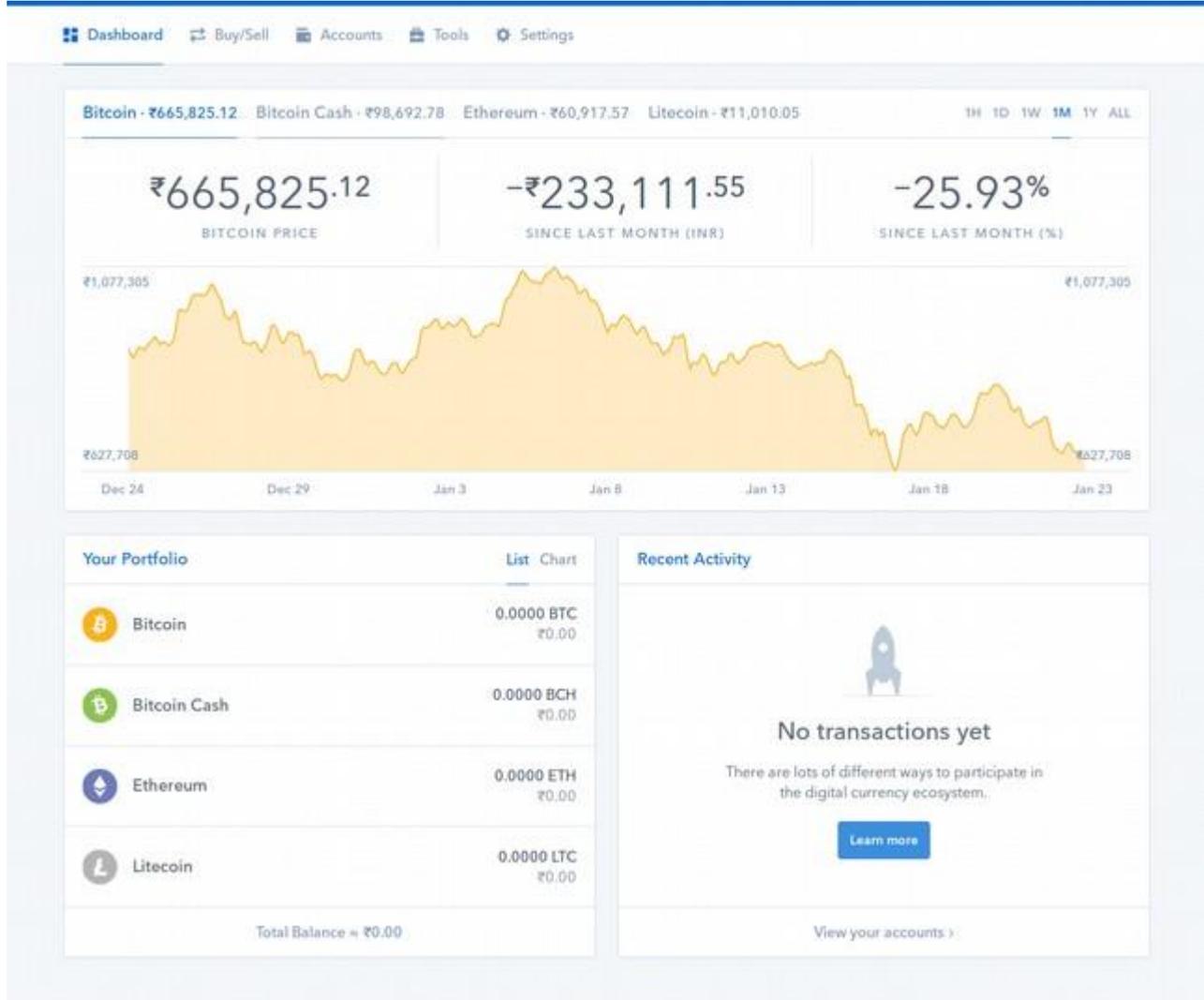
LOGIN

Don't have an account? [Sign up](#)





4.4. Dashboard:



4.4.1. General:

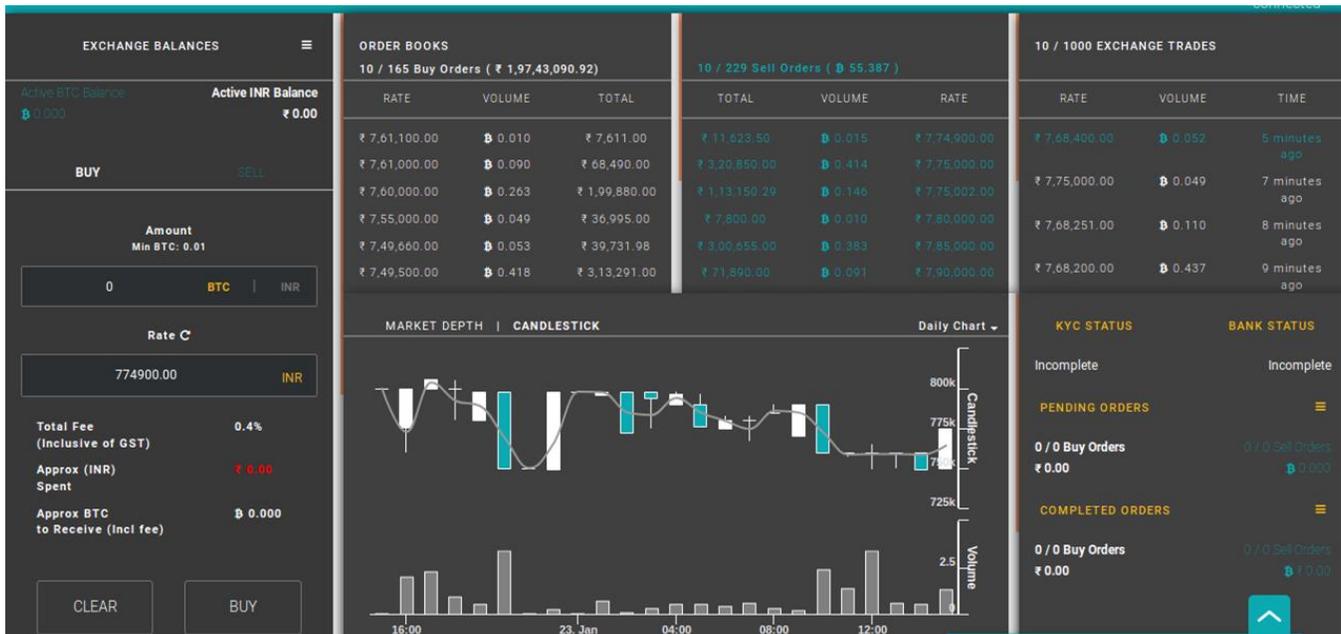
The dashboard has to be displayed following the login screen on successful authentication of user of portal and is the main element of the user interface after granting access. The dashboard must show recent activities and status of cryptocurrency's price, which platform is trading with in INR.

4.4.2. Detailed requirement:

- In case during the phase if any details can't be fetched it should show "Oops something went wrong" in particular part of window
- Chart should be available as per currency ticker tabs so that user can navigate on other currency and fetch data about the same
- There should be option to see historical data of that crypto currency so that customer can track progress of currency.
- When user log's in for first time there won't be any transaction so system should display no transaction in that window.



4.5. Buy/ Sell



4.5.1. General

The Buy/Sell is primarily consists of order book data and recently completed transactions. A focus has to be dedicated to the structure and design as these are very important criteria in order to attract customers for purchasing or selling.

4.5.2. Detailed Requirement

- Exchange balance section will show active users balance in crypto currency and INR.
- Buy Crypto Currency will be form, which will allow user to put up amount of currency he is looking to purchase. System will automatically calculate fees that user need to pay for trade in INR and amount of crypto currency he will get after final trade.
- Sell Crypto Currency will be form, which will allow user to put up amount of currency he is looking to sell. System will automatically calculate fees that user need to pay for trade in that currency and amount of INR he will get after final trade.
- Candlestick graph will be available on this screen will fetch data of trades done in past and will show to end user
- We will show details of successful trades done in past on this page. It will have details of trade in

4.6. User Account

4.6.1. General

User account will consists of forms which will allow him to complete KYC and other part like name , password change etc.

4.6.2. Detailed requirement

- User will have access to profile page which will display his details registered with site at the time of registrations which he can change, only name can be edited from this form. Username and email can not be edited.



- For KYC we will be providing input form which will allow user to upload his identification documents using which his identification can be stored with us.

5. Other Development

5.1. Trade Engine:

5.1.1. *General*

A currency exchange is a system for buyers and sellers of different currencies to exchange different types of currency. The other matching module matches buy and sell orders, creates transactions to record the process, and updates the customers account balances.

5.1.2. *Detailed requirement*

A customer enters the quantity and prices of the orders and clicks "buy" or "sell"

- Website creates record in OrderBook with Pending order status. The order is filed to be processed. [PlaceBuyBid
Then, the Order Matching Service iterates through the list of pending orders. [public int PlaceBuyBid(int customerId, decimal quantityOfBTC, decimal pricePerBTC, DateTime ?expirationDate = null) and public int PlaceSellOffer(int customerId, decimal quantityOfBTC, decimal pricePerBTC, DateTime ?expirationDate = null)] For each order:
- Re-verify order status to ensure that it is still pending/active. If expired, the order is Cancelled
- Validate the order funding. [public bool ValidateOrderFunding(Order order)] The customer must have sufficient assets to process the order. If not, the order is Suspended (it may be re-activated if funds become available later.) If order passes validation:
 - o the order status is changed to Active
 - o the assets needed to pay for the order are added to the Frozen balance. The prevents the customer from placing orders on more assets than he has. This feature may be removed later – we can allow spending greater than the available balance if we check the balance before processin the transaction.
- The Order Matching Service tries to find a match to the buy or sell order. To find a match, we search all Active orders which match the specified price. [ISpecification IsMatchingOrderQuery(decimal price, int orderTypeId, int wantAssetTypeId,int offerAssetTypeId, bool? isMarketOrder)]
 - o *If the order is a buy, we look for a price less than or equal.
 - o *If the order is a sell, we look for a price greater than or equal.
 - o *If it's a market order, we find the highest (sell) or lowest (buy) price.

We sort the matches ascending for buy orders, and descending for sell orders. Then we sort by date if the price matches. [ISpecification IsMatchingOrderQuery(decimal price, int orderTypeId, int wantAssetTypeId,int offerAssetTypeId, bool? isMarketOrder)]
- We load the top 3 matches into memory. The reason we take 3 matches is in case one of the orders fails validation, we can try with the order 2 orders.
- We compare the order and the match. This is a double check with C# – the order should have been already matched by the database query above. The orders should match:

EXCHANGER



- assets (\$/BTC, order types (buy/sell), non-two market orders, and have matching prices (as above). [OrderComparisonResult CompareOrders(Order firstOrder, Order secondOrder)]
- If the orders comparison succeeds, we generate a Transaction to record the match [Transaction GetTransactionForTwoOrders(OrderComparisonResult comparisonResult)].
 - *A_Order is the buy order
 - *B_Order is the sell order
 - (A and B are used because I'm still not sure whether ordering should be Buy/Sell, chronological, or something else)
 - If the order quantities do not match exactly, we must generate a split order with the remainder of the larger order.
 - We run [ActivateTakeProfitAndStopLossOrders(Order order)] – TODO – should not be done here, but scheduled. (See separate post for advanced orders – take profit, stop loss, trigger, stop order, etc.)
 - We process the transaction to record the result [public Order ProcessTransaction(Transaction transaction)] (note: this module is a database transaction)
 - a: add the transaction and the split orders to the DB
 - b: for both orders in the transaction:
 - o subtract the debit access (customers balance of \$ or BTC)
 - o increment the credit asset (customers balance of \$ or BTC)
 - o record the commission in the commission account
 - o unfreeze frozen balances
 - o save changes
 - Repeat the process on each split order, until there is either no remaining quantity (entire order is fulfilled) or we run out of matching orders; the remaining split order stays open as an active order
foreach (OrderProcessResultModel n in ProcessOrder(splitOrder.OrderId))
{
yield return n;
}
 - If the remaining quantity is 0, set the status to Completed

EXCHANGER



5.1.3. DB schema

