FUNCTIONAL SPECIFICATION DOCUMENT

DOCUMENT VERSION 1.1

20.11.2020

APPROVALS

Approval Date	Approved Version	Approver Role	Approver
20.11.2020	Version 1	Project Lead	Veera Sarma
20.11.2020	Version 1	Project Manager	Jenicks
20.11.2020	Version 1	Developer	Alwin Mathew

AUTHOR

Name	Role	Department
N.Gnaneshwaran	Project Associate	Project coordinates

Table of Content

Contents	Page No
1.**Introduction**	5
2.**Purpose of the document**	5
3 **Project Scope.**	5
4. **Terms/Acronyms and Definitions**	6
5**.Technical Stack**	8
6.**Modules & features**	8
6.1. **Registration**	9
6.2. **Login**	10
6.3. **Derivative Trade**	11
6.3.1 **Code overview**	12
6.3.2 **Derivative DFD**	15
6.3.3 **Example for Derivative Trade**	16
6.3.4 **Database overview**	17
6.4 **Spot Trade**	19
6.4.1 **Spot DFD**	20
6.4.2 **Code Overview**	21
6.4.3 **Database overview**	22

6.5 **Calculations for Derivative Trade & Spot Trade**	22
6.5.1 **Spot Calculations**	22
6.5.2 **Derivative Calculations**	23
6.6. **Account Modules**	25
6.6.1 **Trading Wallet & Exchange Wallet**	25
6.6.2 **KYC**	26
6.6.3 **Preferences**	26
7.**Conclusion**	27

1.Introduction:

Bitengen is developed has highly scalable crypto exchange platform and have leverage options up to 100x leverage trading with access to the peer-to-peer funding market and here two types of trading are devised first one is derivative Trade and second is spot trade and important features in derivative Trade are user have sufficient Balances in BTC and were user can make trade in any other cryptos pairs we have five major pairs BTC,ETH,BCH,XRP,LTH and spot trade we have BTC ,ETH,INR and USDT markets.

2.Purpose of Document:

This Document provides detailed information on how the Exchange platform will function and the requested behaviours. This document is created based on the high-level requirements identified in the Business Requirements Document and provides traceability on the functional specifications back to the business requirements. Included in this document will be the detailed functional requirements including use cases, system inputs and outputs, process flows, diagrams, and mock ups.

3.Project Scope:

A Exchange is devised with peer to peer Trading implemented using blockchain technology to gain decentralization, transparency, and immutability.

Instant trade and multiple order form supported for spot and derivative trade: Quick and fast trade Perpetuals Contract are implemented for derivative trading and user has provision to trade with different crypto pairs BTC/USD, ETH/USD, XRP/USD, BCH/USD.

Instant Wallet: once the user completed the registration process the separate wallet are automatically created for all crypto pairs implemented in this platform.

Fund Security: (Non-custodial crypto wallet): We don't keep a backup of your private keys to ensure you provide a completely decentralized crypto wallet that is safe from any type of hacks. User assets are completely safe and secure in the non-custody environment.

Secure Withdrawal: Secure Transaction meth ology were implemented for withdrawing the Fiats and cryptos Such as: MPin ,2FA.

Mpin:

The M-Pin client-server protocol, which features two-factor client authentication as an alternative to Username/Password add extra security for all fiat and crypto withdraws were implemented **user** can access this feature in preference section.

Two Factor Authentication (2 FA): The other significant security measure is Two Factor Authentication implementation This is mandory recommended for all fiat and crypto withdraws. This means users are required to use a password and a registered mobile phone to authenticate the account at login and to withdraw funds from the exchange.
Easy KYC: Simple KYC required for minimum caping of the coins for daily withdrawal, users are able to do the KYC in very less time.

Finance:

The staking concept in which the user can deposit in to different investment plans devised by platform and only USD is allowed to user to make investment.

4.Terms/Acronyms and Definitions:

Limit:

A **limit order** is one of the most basic order types. It allows the traders to specify an amount and price they are able and willing to buy or sell.

Example: If the best ask price is 316 and I want to buy lower than that at 315, then I would place a limit buy order at 315. If a seller is willing to take my bid, my limit order will be matched and executed at 315.

Market:

A market order is an order type that executes immediately against the best price available. A market sell will match the best available bids on the order book, while a market buy will

match against the best available asks. Market orders are often used when rapid execution is **prioritized** over the price at which the order executes.

When placing this order, the trader prefers the trade to happen immediately at the current best prices available.

Good-Till-Cancelled:

A **Good-Till-Cancelled** (GTC) **order** is an **order** to buy or sell a stock that lasts **until** the **order** is completed or **cancelled**

Immediate or cancel:

An **Immediate or Cancel (IOC) order** requires all or part of the order to be executed immediately, and any unfilled parts of the order are cancelled. Partial fills are accepted with this type of **order** duration, unlike a fill-or-kill **order**, which must be filled **immediately** in its entirety or be **cancelled**.

Fill or kill

A "fill or kill" order is an order that must be filled immediately in its entirety, at a specific price (or better), or it is cancelled (killed).

Post only:

The post-only limit order option ensures the limit order will be added to the order book and not match with a pre-existing order. If your order would cause a match with a pre-existing order, your post-only limit order will be cancelled.

Reduce only:

Reduce-only orders serve to strictly reduce your position size by dynamically reducing or adjusting your limit order's contract quantity to match the contract size of the open position.

5.Technical Stack:

Frame Work	MEAN
Front End	React, Html, CSS
Backend	Mongo dB, Node

MongoDB: A document-oriented, No-SQL database used to store the application data. In our platform the MongoDB are uses for Database.

NodeJS: The JavaScript runtime environment. It is used to run JavaScript on a machine and Nodejs is used in our back-end functionalities.

ExpressJS: A framework layered on top of NodeJS, ExpressJS is used in our Planform for connecting API.

ReactJS: It is used to build UI components, In this platform entire front-end Part was developed using this ReactJS

6.Modules & Features

In this platform we have Five Major Modules:

1.Registration

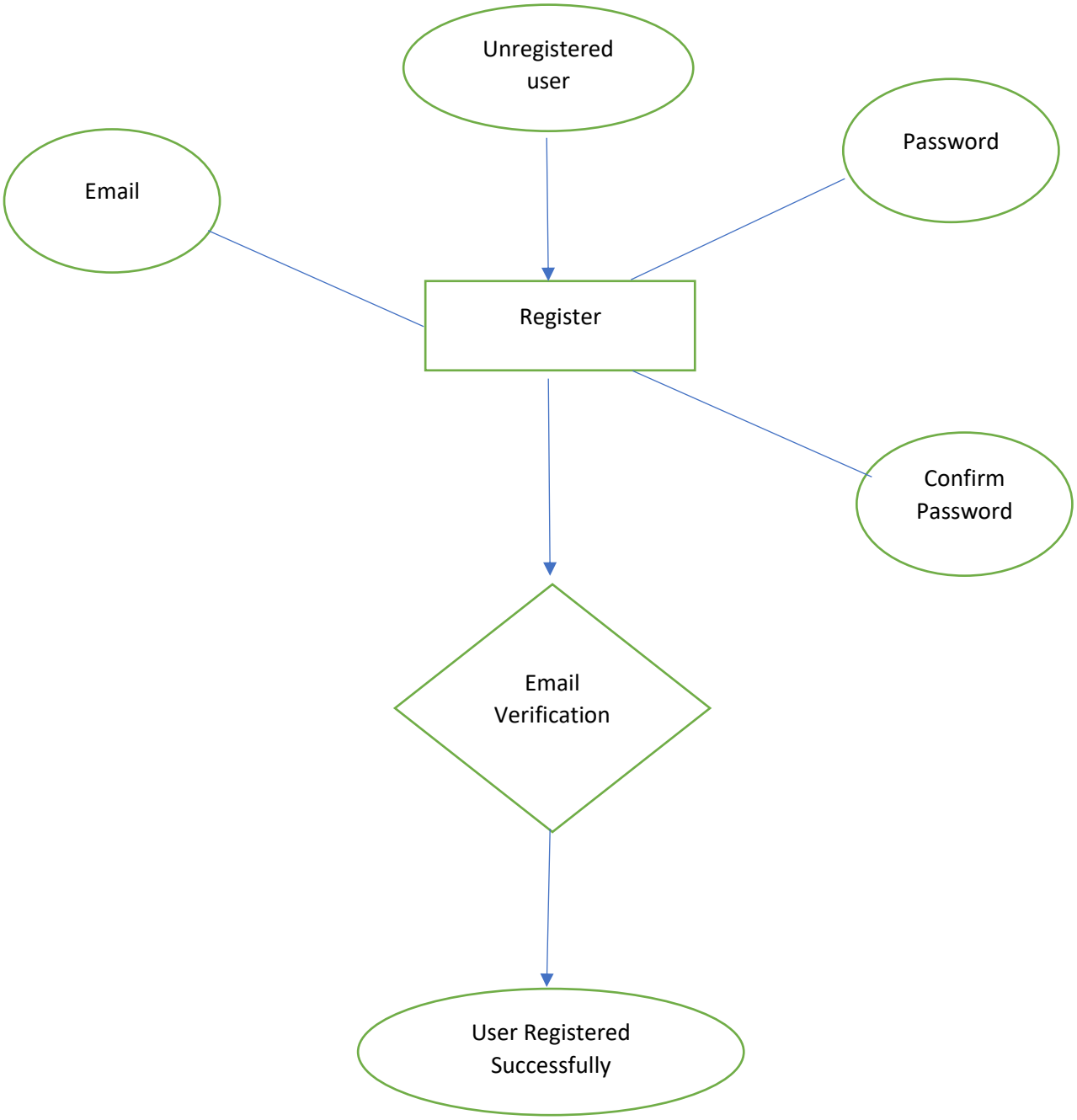
2.Login

3.Derivative

4.spot

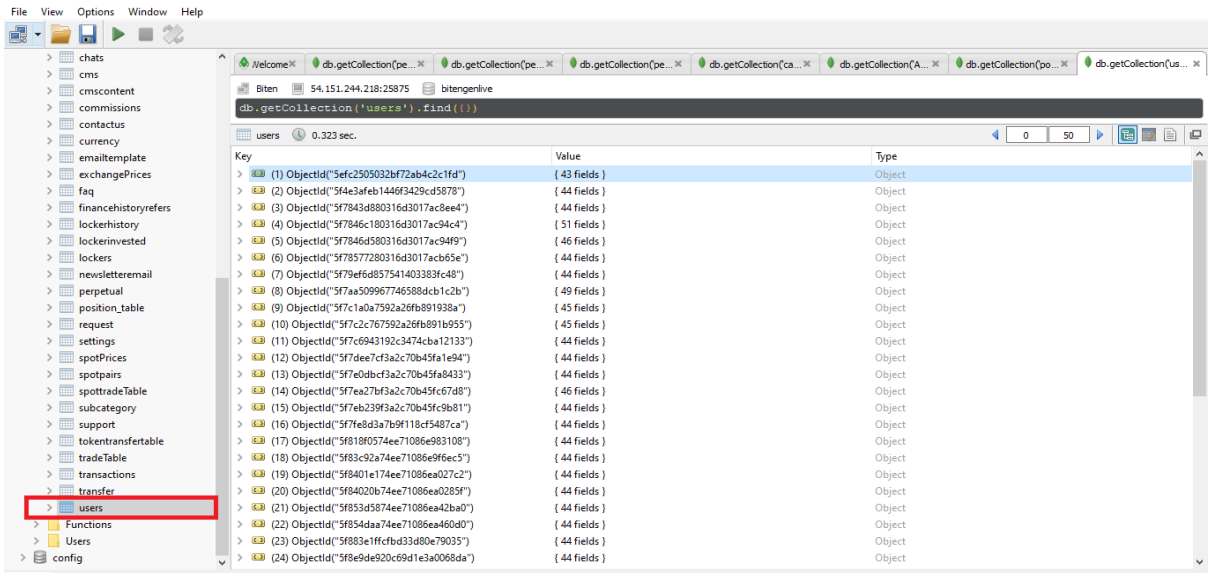
5.Account

6.1. **Registration:**



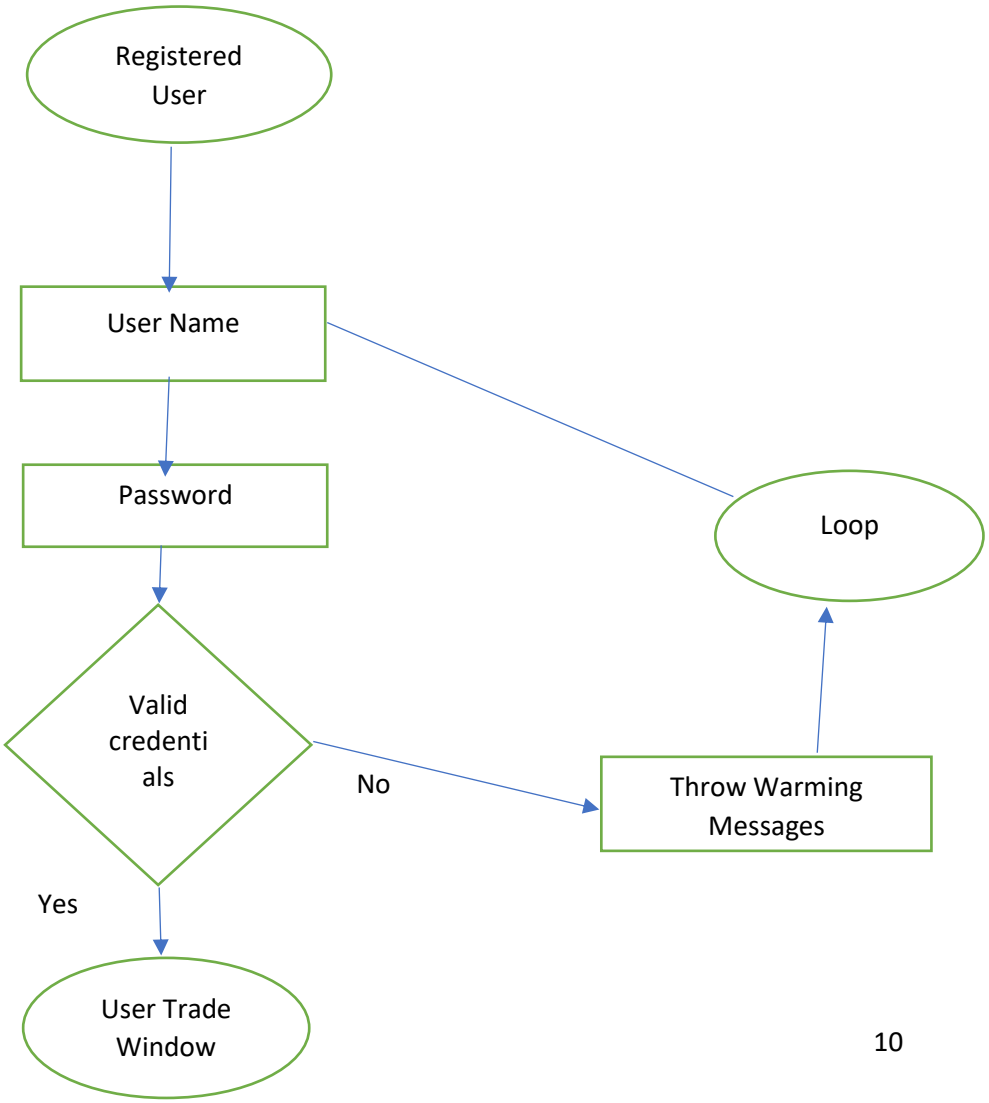
In register section the unregister user can proceed with providing email and strong password and after successfully submit of registration, confirmation email will be trigger to respective user mail and once user complete the email verification the account will successfully activated

User Table: All registered user stored in User table and unique object id is generated for all registered user.



6.2. **Login:**

In this section Registered user are able to login with register Credentials and all registered are directed to User Dashboard and invalid credentials getting from login section will throw a warming message.



6.3. **Derivative Trade:**

Perpetual contracts methodology is implemented for derivative trading for following crypto pairs BTC/USD, ETH/USD, LTH/USD, XRP/USD, LTH/USD and leverage options included up to 100x and in derivative trading has different options for placing a order for both **sell** and **buy** to place a order

Example: options such as order Post only, Reduce Only, GoodTillCancelled ,FillorKill, ImmediateOrCancel .

Order Type:

Limit: User has provision to set Leverage and contracts quantity and set the order price. When the best traded price reaches of set of user order limit price, the order will be executed.

Market: User are able to set their leverage and contracts quantity, but not the executed price, and user want to buy or sell immediately can use the market order features.

Buy Order: Allow order only user enter Price below or Equivalent Market Price.

Sell Order: Allow order only user enter price above or equivalent Market Price.

Graph: The market inflations and deflations of selected crypto pair are showed and moving Average are shown default. User can select different trade view types.

Order Book: Buy order and sell order are showed with top Ten best price with Qty.

Recent Trade: All executed trade data with Price and Qty with time will be showed.

Dashboard: User sell and buy order top prices are displayed once the user execute order and the market price inflation and deflations also shown above the specific cryptos and Green colour represent buy and red colour represent sell order.

Position: Once the buy or sell order are executed then the position will be open and user has provision to close the position with market price after positions are open.

Closed P & L It displays profit or loss of user position at the current market price after close of positions, In this tab the entire the data of exit and entry price and contract details, Quantity, Exit type whether short or long and the close position time will be also showed.

6.3.1 Code Overview:

The component for derivatives trading For this route need to pass the Pair as the parameter

In this component we need to display the trading related records all are get by the each separate services

In the **getData** function the services are called

Depends on the pair parameter in the URL, the records are getting from backend

Onchange function - The calculation for the Margin impact, And for the liquidation price calculation, Distance between the Mark price and liquidation price all these are in the onchange function

Before make the order placing we show the all details to user once confirm from user then the order placing will occur

Leverage Option

For the derivatives trading user can choose the leverage, depends on the leverage only the Margin impact and Liquidation price will work

order Placing function - For make the order

By the Bitcoin balance only the trade will occur it means user can trade eny pair in derivatives with the balance of the bitcoin

We have the four order types Market, Limit, Stop, Take profit

Market

- The trade will do with current price.
- If user don't have the balance the it will shows the message like insufficient balance

Limit Order

- User can enter the Price and the trade will do depends on this price
- Here we have condition like user cannot enter higher than the Spot price in Buy section and user cannot enter lower than the Spot price in Sell side

Take profit

This order type for closing the open position like if user open the long position the take profit order for the short type by closing the position when mark price reach the take profit price Here user can only enter the profitable price

Order book

GetTradeData this is the service to get the order book data

Recent trades

GetTradeData this is the service to get the Recent trades data

Position details

In this section user open positions are display like Contract, type, Value, Price, Liquidation price, Profit and loss, initial margin,

All the position data’s are get from the getuserTradeData service and make the calculation before render process.

Here provide the option to close the position with the market order, The new order will

place and the position will close the profit and loss will update in the users balance

Closed positions

In this section the users closed positions are displayed it contains the Quantity, Entry price, Exit price, Date and time, Profit and loss details

Pending orders

These are the open orders of the particular user For this we get the data from getuserTradeData service

Filled order

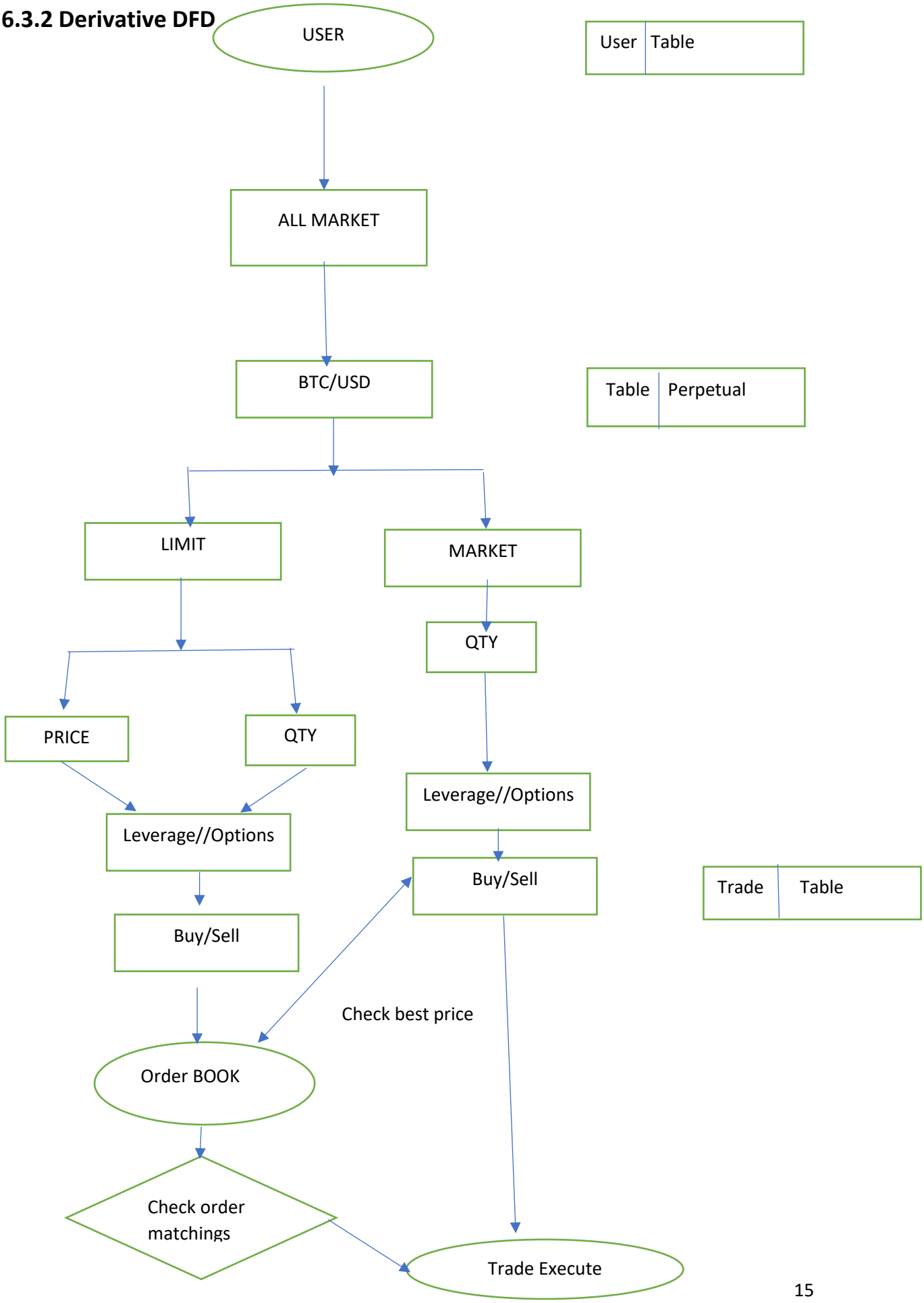
These are the trade history of the particular user For this we get the data from getuserTradeData service

History

It contains the all status of the trade like New, Filled, Partial, Cancel For this we get the data from getuserTradeData service

For the ticker price we get the getspotPricevalue service

6.3.2 Derivative DFD



6.3.3 Example: Now we going to make trade with two different registered Account

User 1: Tom@yopmail.com

User 2: Jack@yopmail.com

Bitengen platform was implemented uses fair price marking to avoid market manipulation.

Select the Pair from Market**: BTC** /USD

Enter the Price and Quantity ie

Current market price: 17959.61 USD

User Jack :(Sell order)

Account Balance: 99.99248292 BTC (1771893.00 USD)

Sell order Data:

Qty :1 BTC

Price: **17959.61**

Leverage :1x

Order: **Sell**

After order sell order executed Balances :98.9924695 BTC

User Tom (Buy Order)

Current Account Balance: 87.96735979 BTC (1597495.51 USD)

Given Data:

Qty:1 BTC

Price :17959.61

Leverage: 1x

Order: **Buy**

After Order buy order executed -Balances: 88.97718313BTC (1595487.84 USD)

Buy Order**:** Qty :**1** BTC price :**17959.61**

Once the order is placed the balances will be ducted from Account and trade engine will find the best price for execution.

Currently the order status is pending since there is no match order in order book.

Margin Impact is 1.00094738BTC

Examine the margin impact calculation:

Margin Impact = Margin impact = (Amount * Price) / Leverage + (0.075%) (this is $ value we need to convert as BTC)

Given Data :

Qty:1 BTC

Price :17959.61

Leverage: 1x

So 1X17959.61 /1+(0.075%)USD // convert in to BTC

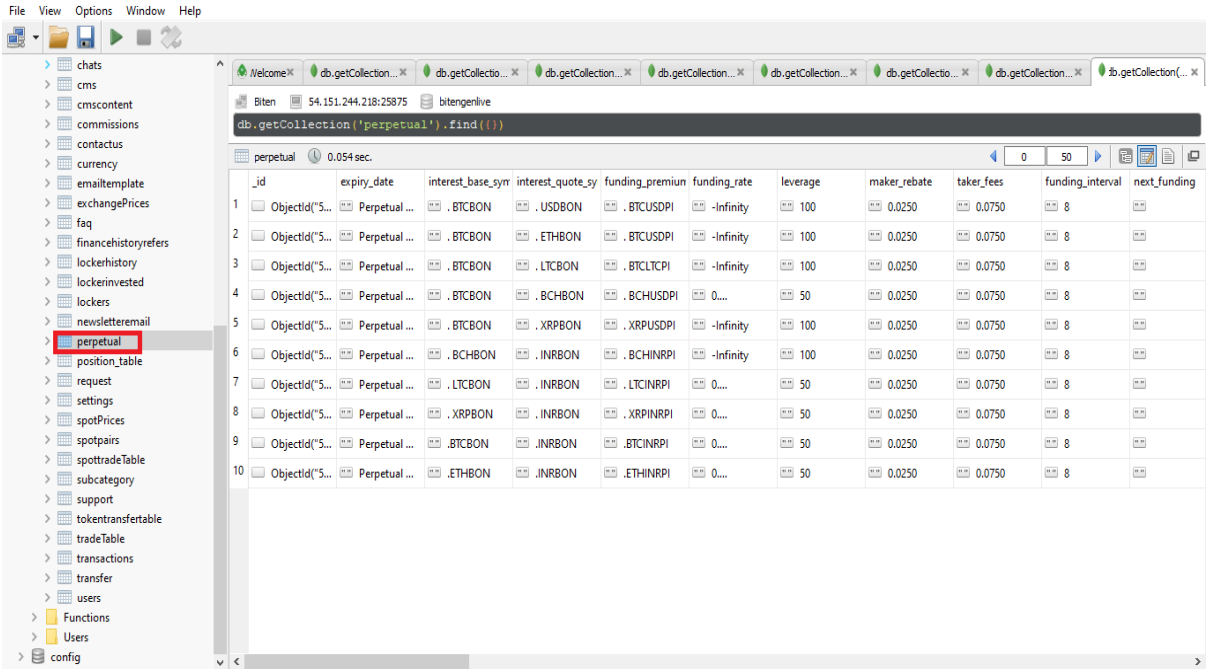
17959.61+134 USD // Convert to BTC

18.093.61 USD

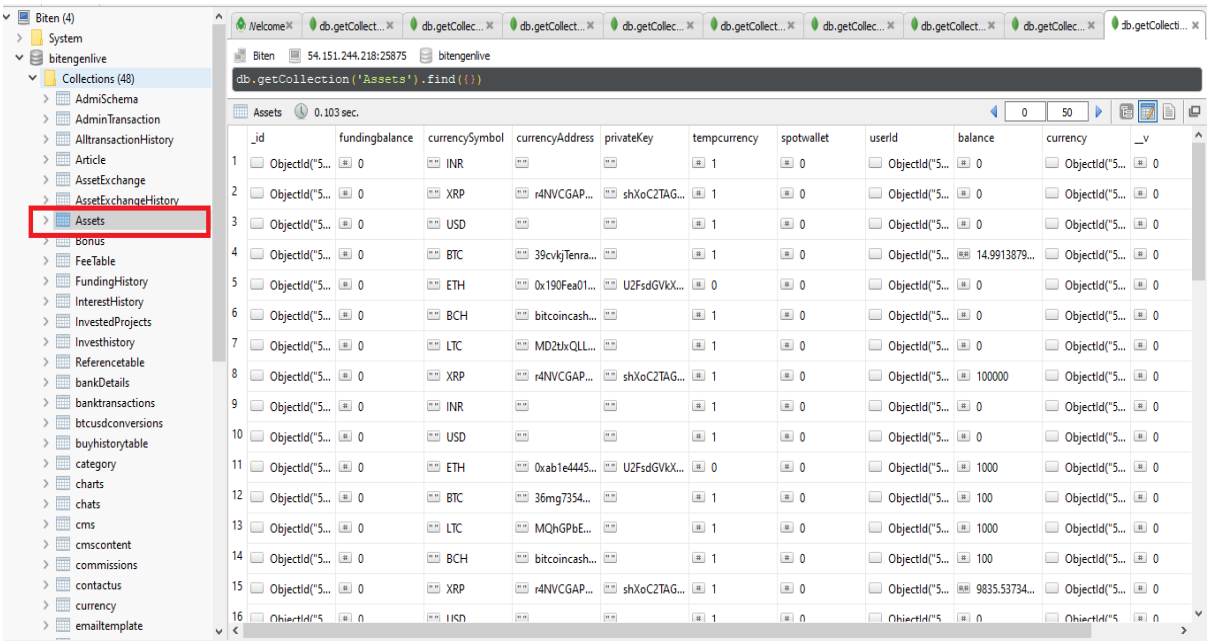
1.00094738 approx.

6.3.4 Database overview:

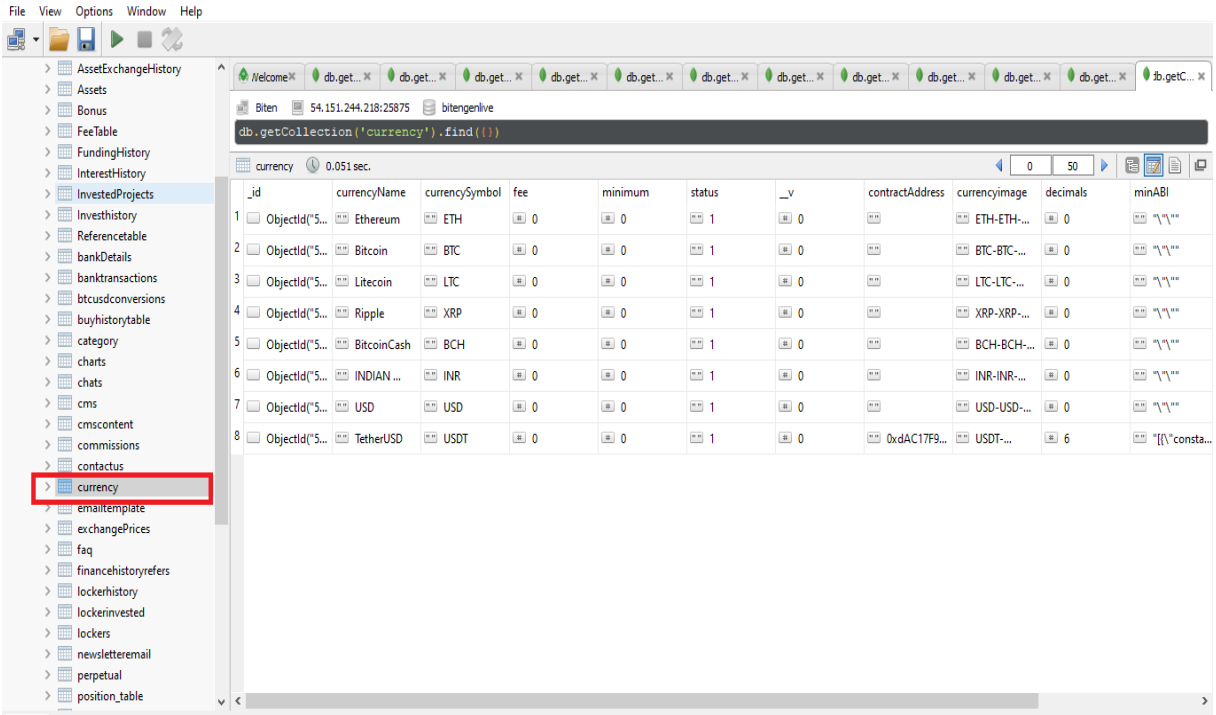
Perpetual Table: It contains the pairs of derivatives trades



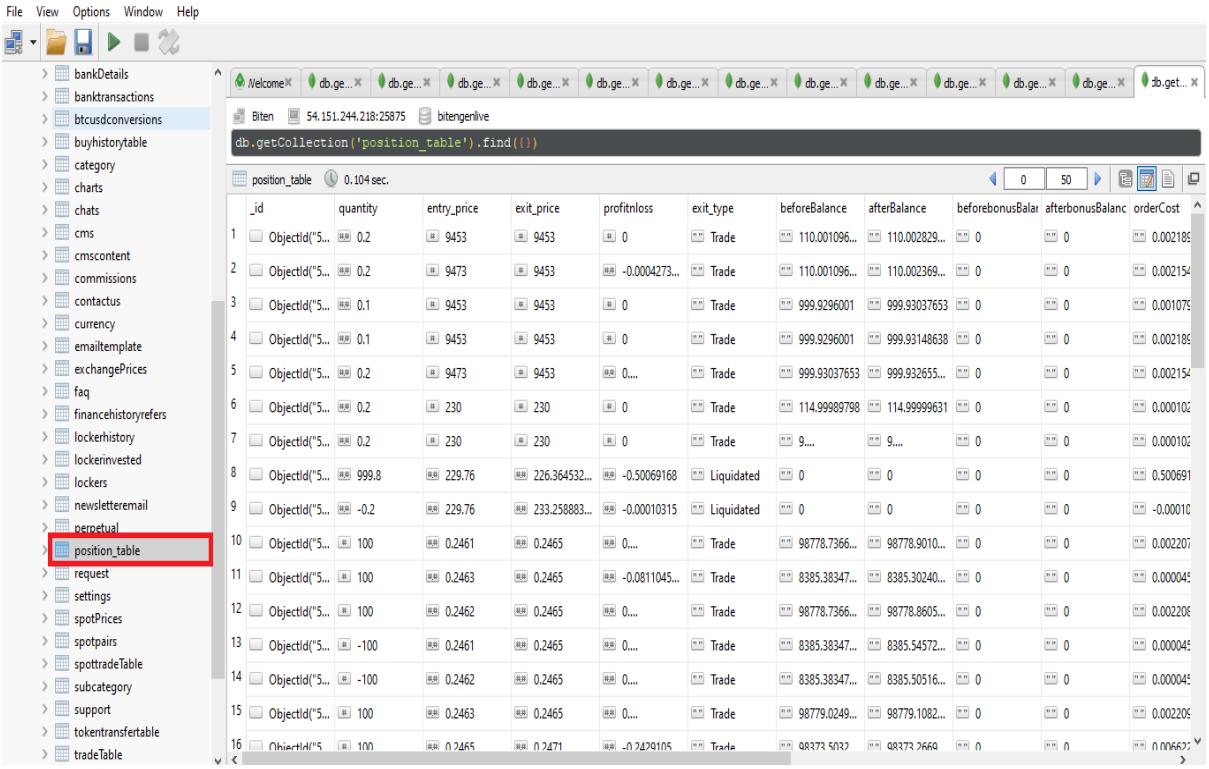
Asset Table: It contains all user wallet data and all user deposit fiat and crypto are automatically updated here .



Currency Table: Fait and Crypto currency are added here and we add new currency via this currency table .



Position Table: user positions data are store here and dynamic data of trade positions made by all registered user are fetched here.



6.4.Spot Trade:

Spot trade has unique feature were the user has sufficient balances in one coin can execute trade with different coins ie if user has BTC, user has provision to execute the trade with ETH, LTC, XRP, BCH and here BTC act as base currency**.**

Spot trade has unique feature were the user has sufficient balances in one coin can execute trade with different coins ie if user has BTC, user has provision to execute the trade with ETH, LTC, XRP, BCH and here BTC act as base currency**.**

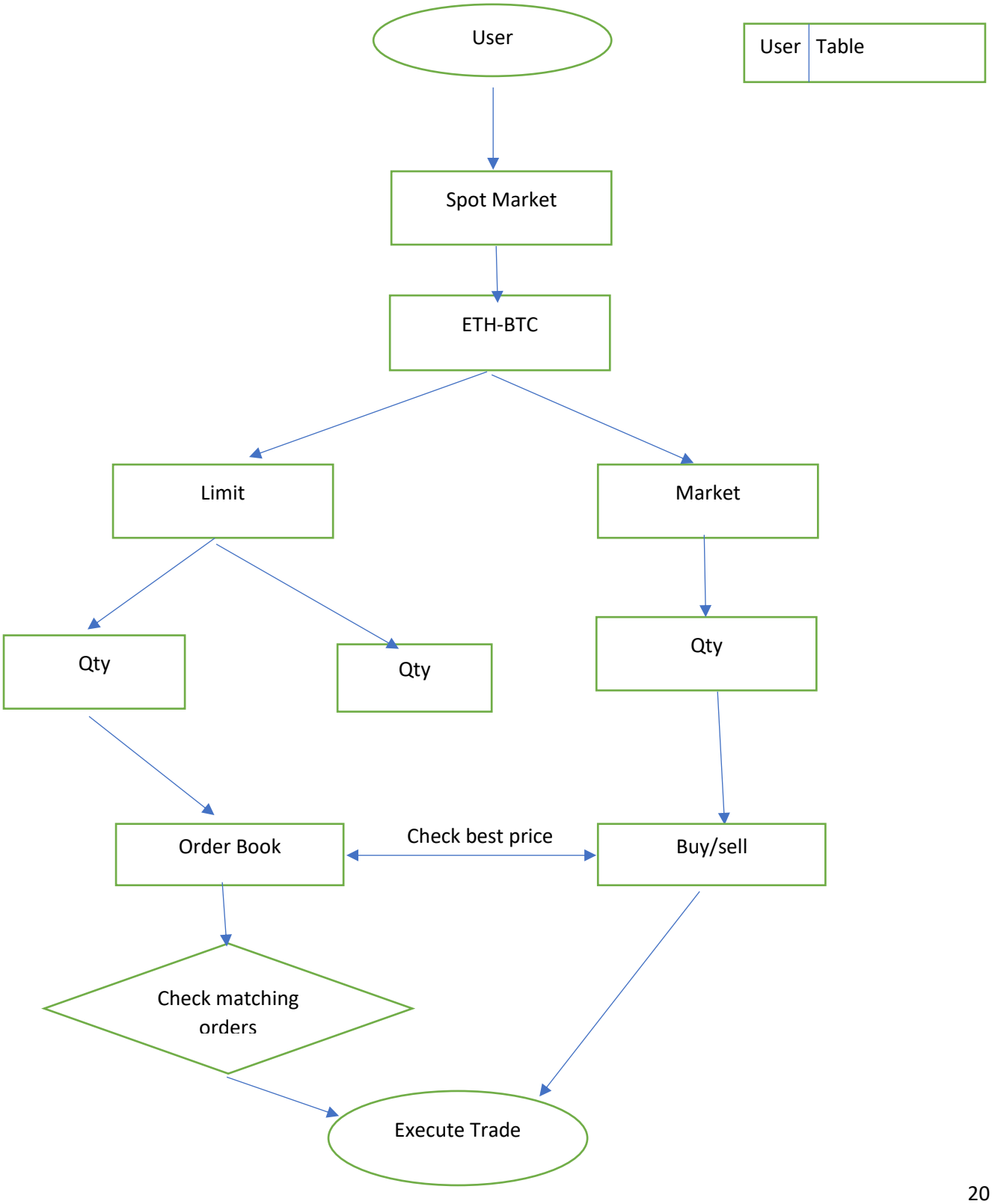
The trade is implemented for BTC, ETH, INR, USDT Markets.

Pairs - ETH/BTC, LTC/BTC, XRP/BTC, BCH/BTC, BTC/ETH, LTC/ETH, XRP/ETH, BCH/ETH, BTC/INR, ETH/INR.

Open Order: The user pending order are shown here with user exact qty and price and filled execution data and order status and order time will be displayed.

Order History: Every order history where status in partially or executed all data will be shown. Here also two order types are permitted:

4.1 Spot DFD



6.4.2 Code Overview:

Spot:

This is the component for Spot trading For this route need to pass the Pair as the parameter In this component we need to display the trading related records all are get by the each separate services .

In the **getData** function the services are called Depends on the pair parameter in the URL, the records are getting from backend

Onchange function - The calculation for order value is in this function

orderPlacing function - For make the order

We have the two order types Market and Limit

Market

The trade will do with current price All the Market orders executed by best available sell or buy order listed in order book If user don't have the balance the it will shows the message like insufficient balance

Limit Order

User can enter the Price and the trade will do depends on this price Here we have condition like user cannot enter higher than the Spot price in Buy section and user cannot enter lower than the Spot price in Sell side

Order book

GetspotTradeData this is the service to get the order book data

Recent trades

GetspotTradeData this is the service to get the Recent trades data

open orders

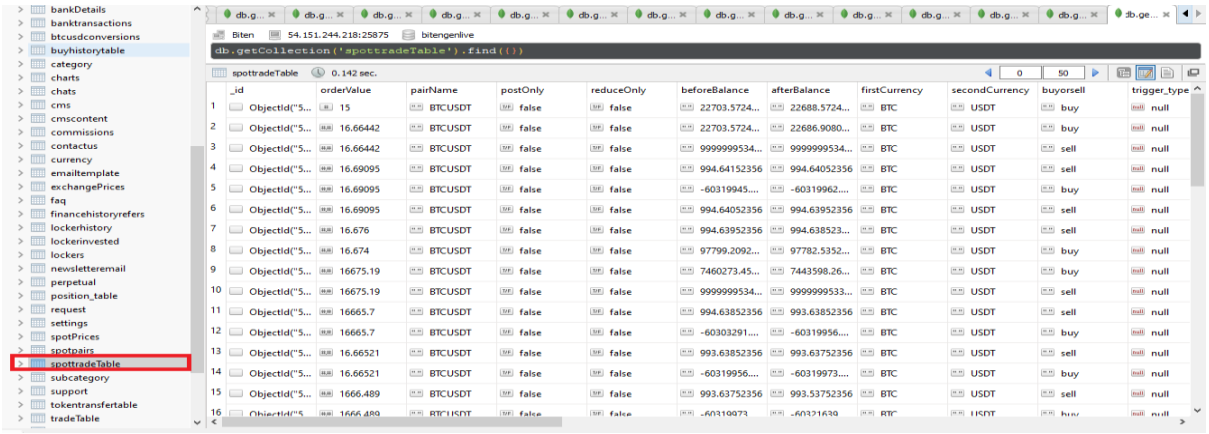
These are the open orders of the particular user For this we get the data from getspotuserTradeData service

History

It contains the all status of the trade like New, Filled, Partial, Cancel For this we get the data from getspotuserTradeData service For the ticker price we get the getspotPricevalue service

6.4.3 Database Overview:

Spot pairs: The spot pairs for for BTC, ETH, INR, USDT Markets are stored and Market price of each cryptos and market low ,high ,last price ,total volume trade ,first and second currency details are also stored in this table .



6.5 Calculation used for Derivative Trade & Spot Trade

6.5.1 Spot Trading

Pair = BTCUSD

first currency = BTC

second currency = USD

quantity - From user

price - From user

order value= quantity * price

if trade == buy

Balance deducted from second currency

order value is deducted for buy order

if trade == sell

Balance deducted from first currency

quantity is deducted for sell order

6.5.2 Derivative Trading

Pair = BTCUSD

first currency = BTC

second currency = USD

quantity - From user

price - From user

leverage - User set the Leverage

o**rdervalue** -(quantity * price) / current btc price

To calculate the Margin impact

1- order_value1=(quantity * price)

2- fee= order_value1 * takerfees /100

taker fees= fees for the trade

3- required margin = order_value1/ leverage;

4- margininbtc = required margin / btcprice;

btcprice =current btc price

5- feeinbtc = fee / btcprice;

6- margin impact = margininbtc + feeinbtc;

Margin impact is consider as initial margin

To calculate P and L

mainmarginwithleverage = mainmargin * leverage;

if quantity > 0 (long)

Difference = spot price - price;

liqprice = price * leverage / leverage + 1 - mainmarginwithleverage;

if quantity <0 (short)

difference = price - spotprice;

liqprice = price * leverage /leverage - 1 + mainmarginwithleverage;

profitnlossusd = difference * quantity;

profitnloss = profitnlossusd / price;

profitnlossbtc = profitnlossusd / btcprice;

profitnlossper = profitnloss * 100;

To calculate Liquidation Price

order is of BUY

(price * leverage) / (leverage + 1 -mainmargin*leverage)

order is of SELL

(price * leverage)/leverage - 1+ mainmargin * leverage

* mainmargin = maint_margin / 100;

maint_margin-set from the admin panel

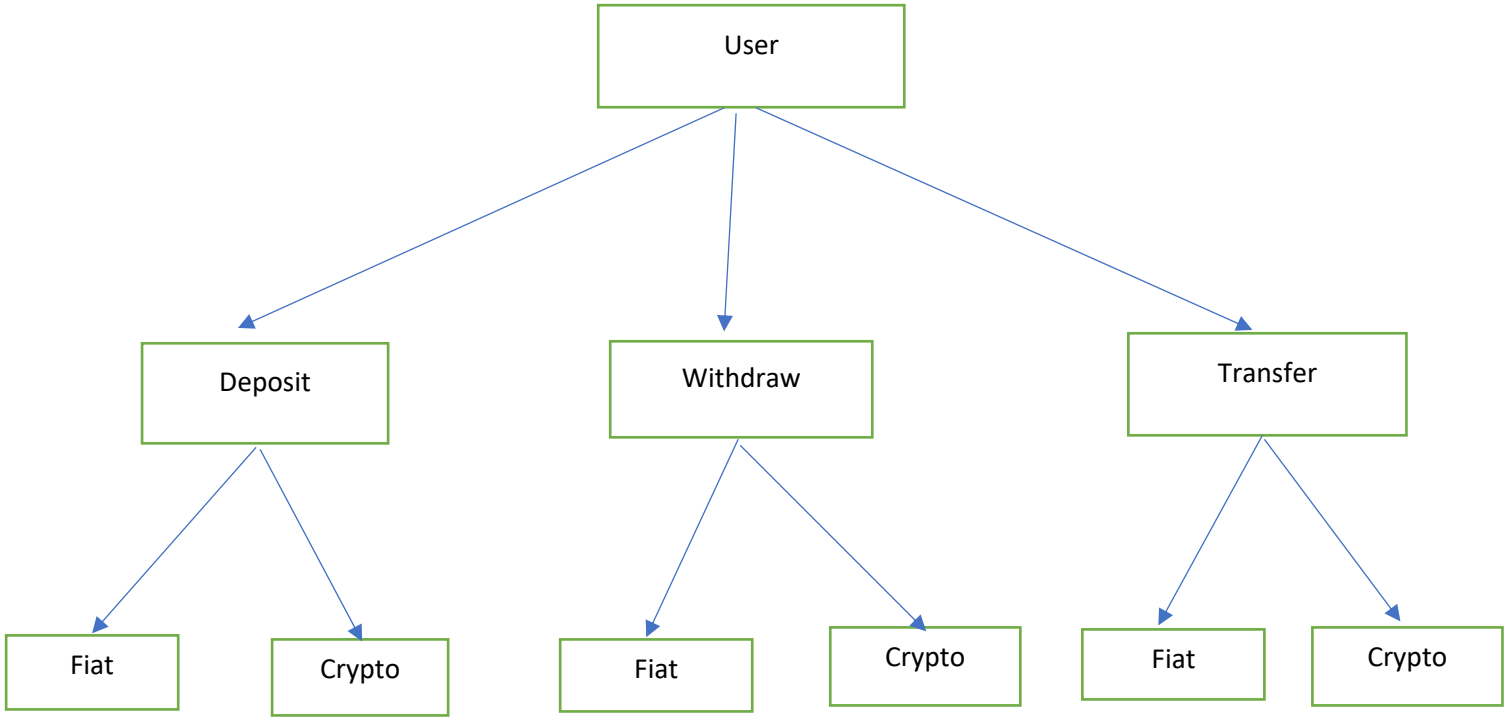
6.6.Account Modules:

The entire user wallet is maintained in this section and here two types of wallet are implemented:

6.5.1 Trading Wallet: The Trading Wallet is meant for Derivative trading and user can make deposit by selecting the corresponding cryptos from deposit section.

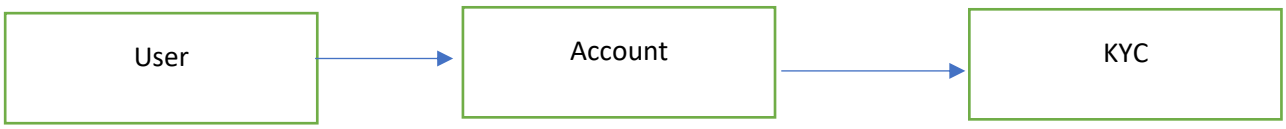
Exchange Wallet: The exchange is meant for Spot Trade and registered has provision to make direct crypto Deposit s.

In preference Section the user has provision to deposit, Withdraw and Transfer their funds and important thing is for transfer options the user can transfer token or Fiat within internal Wallets only.ie Transfer is allowed Between Trading Wallet and Exchange Wallet.



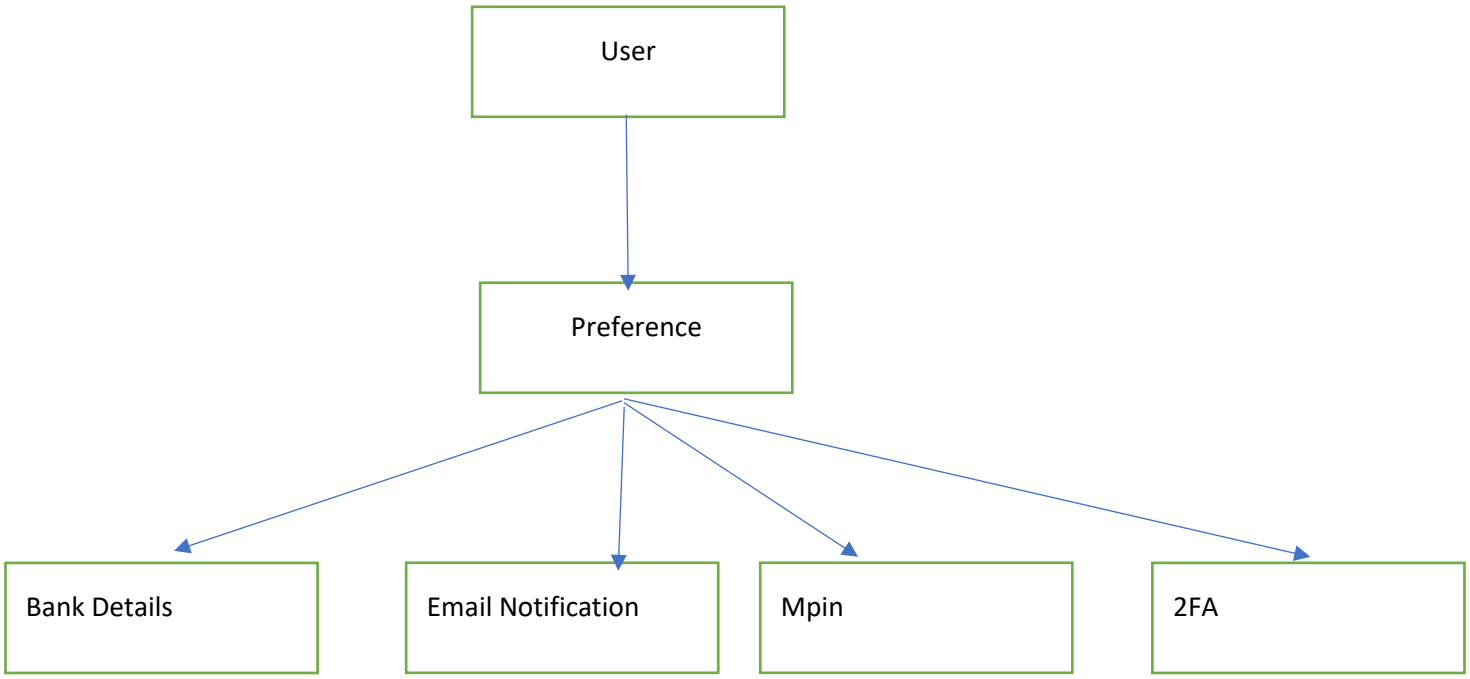
6.6.2 KYC:

The user has to complete the KYC to make Withdraws since the fiat deposit and withdraws are implemented so it is mandatory and all this function user can access in preference Section.



6.6.3 Preferences

User has provision to update the details in preference section in order to make the fiat withdraws and for all withdraws both fiat and crypto user has to complete the Mpin and 2FA activations and All fiat and crypto request are confirmed by email notifications and then approved by Admin.



Stateless authentication is implemented and, in Preference Section user can view all login details such as:

- Geolocation
- IP address
- Time of day
- The device being used

In additional to that user can view all closed profit and loss data and overall trading History in Account section.

7.Conclusion: We have break down the functionalities that has been implemented in Bitengen exchange platform with complete unit and functionality testing on each. Above Modules clarifies flows of each section and entire platforms is developed with meeting the highly standards.

----------END----------