Software Requirements Specification

For

Crypt-O-Gear

(Crypto Currency Value Provider with Market Analysis)

Rithvik S (2018135)

Shameer K (2018L18)

Department of Information Technology

Government College of Technology – Coimbatore

Date:13.10.2022

1. Introduction

1.1 Purpose

This Documentation provides detailed information about the software functional and nonfunctional requirements used for Crypt-O-Gear. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system.

1.2 Project Scope

- This Project have a great role in the field of Cryptography where people Monitor the value of Crypto Currency throughout their trade Time.
- The Crypt-O-Gear Helps traders to take wise decision based on the value of Crypto Currency.
- This is completely made by Python and the details of Crypto Currency are provided by the API

1.3 Product Features

The major features of product are:

i) Easy and Flexible to retrieve the values of particular bit coin. Eg:Ethereum

ii) Provides The marketing moves to make decision based on the values and analytical data

iii) This application has the major feature i.e. To Provide the Complete information about the cryptocurrency and as well as the market study of that particular coin

1.4 Operating Environment

This Crypt-O-Gear is deigned in the way that it can be only operated by terminal like in Python IDLE and Visual Studio Code and Spyder IDE.

1.5 Design and Implementation Constraints

Major Constraints in Crypt-O-Gear:

i) Only Single Crypto Currency value could be acquired at a time

ii) The value is fetched once in a session.

iii) Internet is required throughout the usage in order to get information from API and for API too.

iv) User have to wait for few seconds for results to be fetched from API.

1.5.1 Stimulus & Response Sequences:

Stimuli:

- Run the Program
- Enter the required name of the bitcoin of which you need the Market value.
- Enter “Stop” to terminate the program.

Response:

* As the user enters this data fields, server checks the data received from the client side whether it is present aren’t it.

* If the value found then the server will display the value of the requested crypto currency and additionally it also displays the last modified data of that value and market advisory about that particular coin or else it will show that the data not found in the server side.

* The user can simply Stop the program just by providing the keyword STOP instead of Entering the Bitcoin value.

1.6 Packages used:

socket : for communication between client and server , where cliets's request is satisfied by server

requests : to fetch the API and decode it for our purpose (pip install requests)

json : used for proper usage of API's data in our program for necessary output and program readable language

termcolor : for customize the output and for better user interface (pip install termcolor)

2. User Interface [SNAPSHOTS]:

2.1 CLIENT SIDE:

2.1.1 PROVIDING BITCOIN NAME:

```
PS C:\Users\ksham\OneDrive\Desktop\project dcn> py client.
py
Enter Coin name for info : bitcoin
```

2.1.2 CLIENT-SIDE RESPONSE (AFTER FETCHING INFO FROM API):

```
Enter Coin name for info : bitcoin

Percentage change :  -4.17438 % ( Bitcoin )

Data found successfully on server You can buy this coin If
 you already hold this coin you can sell this after prices
 increase in its value

Enter Coin name for info :
```

2.2 SERVER SIDE:

2.2.1 INITIALIZING THE PROGRAM:

```
PS C:\Users\ksham\OneDrive\Desktop\project dcn> python serv
er.py
Socket created and waiting for request from client
```

2.2.2 DISPLAYING FETCHED DATA FROM API TO THE CLIENT:

```
Socket created and waiting for request from client

API status code :  200

Requested data from the user :

Data found
Bitcoin : -4.17438 %
Last modified :  2022-10-13T13:36:51.093Z
Total volume : 27237003416

Send response : Send a request for another coin
```

3. Hardware Interfaces

No Hardware Interface have been identified

3.1 Software Interfaces

No Software Interface have been identified

3.2 Communication Interfaces

The API and Sockets are used for communication.

4. Technology Stack

- Socket
- API (Application Programming Interface)

5. Program:

5.1 Server side:

```
import socket
import requests
import json
from termcolor import colored

s = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.bind(("127.0.0.1",8000))
print("Socket created and waiting for request from client ")
while True :
    data,addr = s.recvfrom(1024)
    data = data.decode()
    if data.lower() == "stop":
        break
    response =
requests.get('https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&order=mark
et_cap_desc&per_page=2000&page=1&sparkline=false')
    print("\nAPI status code : ",response.status_code,"\n")
    print("Requested data from the user : \n")
    obj = response.text
    parse_json = json.loads(obj)
    length = int(len(parse_json))
    flag = 0
    for i in range(0,length):
        id = parse_json[i]['id']
        name = parse_json[i]['name']
        col = parse_json[i]['price_change_percentage_24h']
        last = parse_json[i]['last_updated']
        vol = parse_json[i]['total_volume']
        # print(id,name)
        if id.lower() == data.lower() or name == data:
            print("Data found  ")
            if col < 0:
                text = colored(col,'red')
                print(name,':',text,'%')
                print("Last modified : ",last,"\nTotal volume :",vol)
                flag=1
                break
            else:
                text = colored(col,'green')
                print(name,':',text,'%')
                print("Last modified : ",last,"\nTotal volume :",vol)
```

```
            flag=1
            break
    change = str(col)
    if flag == 1:
        print("\nSend response : Send a request for another coin ")
        s.sendto(str.encode(change),addr)
    else:
        print("No data found as given coin name is not a specified one , send correct request ")
        s.sendto(str.encode("no-match"),addr)
```

5.2 Client side:

```
import socket
from termcolor import colored

s = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
s.connect(("127.0.0.1",8000))
while True :
    data = input("Enter Coin name for info : ")
    data2 = data.capitalize()
    # print(data)
    s.sendto(str.encode(data2),("127.0.0.1",8000))
    if data.lower() == "stop":
        break
    dat,addr = s.recvfrom(1024)
    dat1 = dat.decode()
    flag=0
    if dat1 >= "0" and flag == 0 and dat1 != 'no-match':
        flag=1
        dat1 = colored(dat1,'green')
        print("\nPercentage change : ",dat1,"% (",data2,")\n")
        print("Data found successfully on server You can buy this coin after decrease of it
value If you already hold this coin you can sell this now\n")
    if dat1 != "no-match" and flag == 0:
        dat1 = colored(dat1,'red')
        print("\nPercentage change : ",dat1,"% (",data2,")\n")
        print("Data found successfully on server You can buy this coin If you already hold this
coin you can sell this after prices increase in its value\n")
    if dat1 == "no-match":
        dat1 = colored(dat1,'magenta')
        data = colored(data,'cyan')
        print("",dat1,"' says that '",data,"' is not a coin enter correct request for coin ")
```