Software Design Document
HTTP Server

28/03/2017

Version 1.0

Group Number : 11
Group Members: Mohit Jindal[15114046]
Chirag Maheshwari[15114020]
Nitish Bansal[15114048]

1.Product Overview

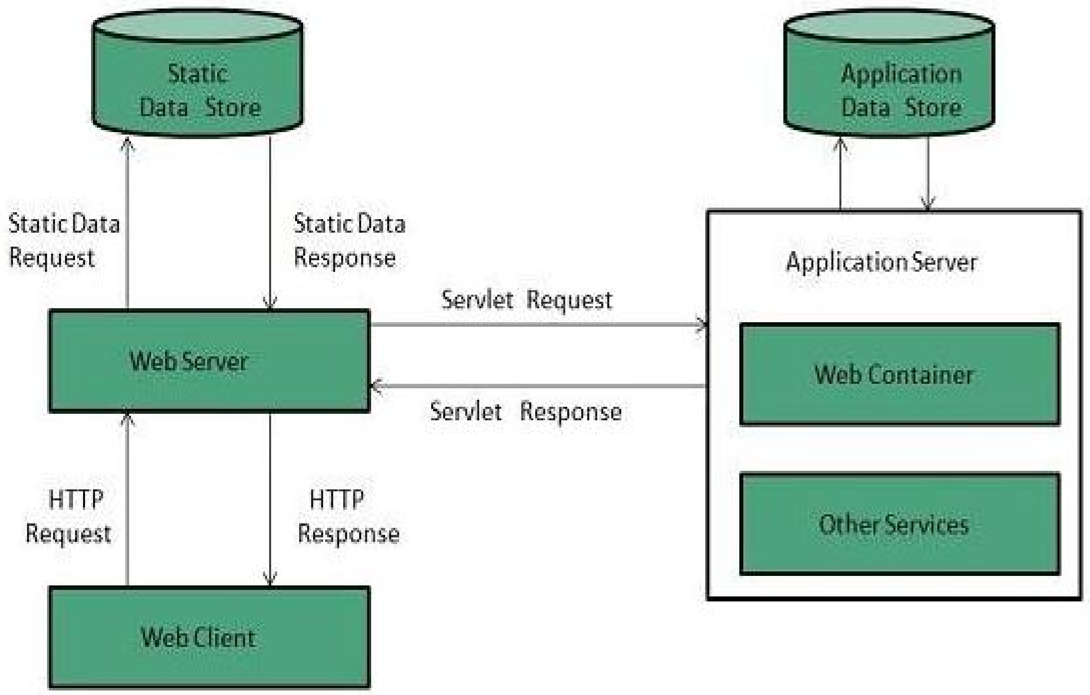
HTTP Server is designed keeping in mind the requirements of learner.It mainly focuses on basic server functions.

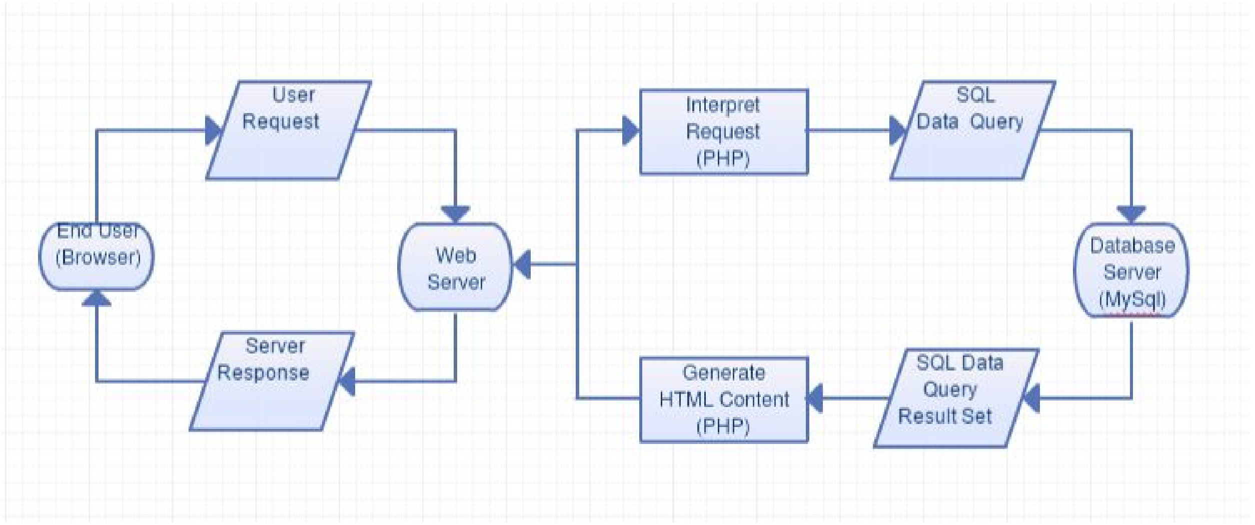
This server performs several basic functions that are avaliable on online learning servers ,in addition it has many more benefits.It provides an easy way to store and edit database and provides access to web pages stored in the database.It can perform GET,POST ,PATCH ,DELETE and HEAD requests efficiently.It follows KIS(keep it simple) principle for the sake of understanding of the learner.
It has several benefits:

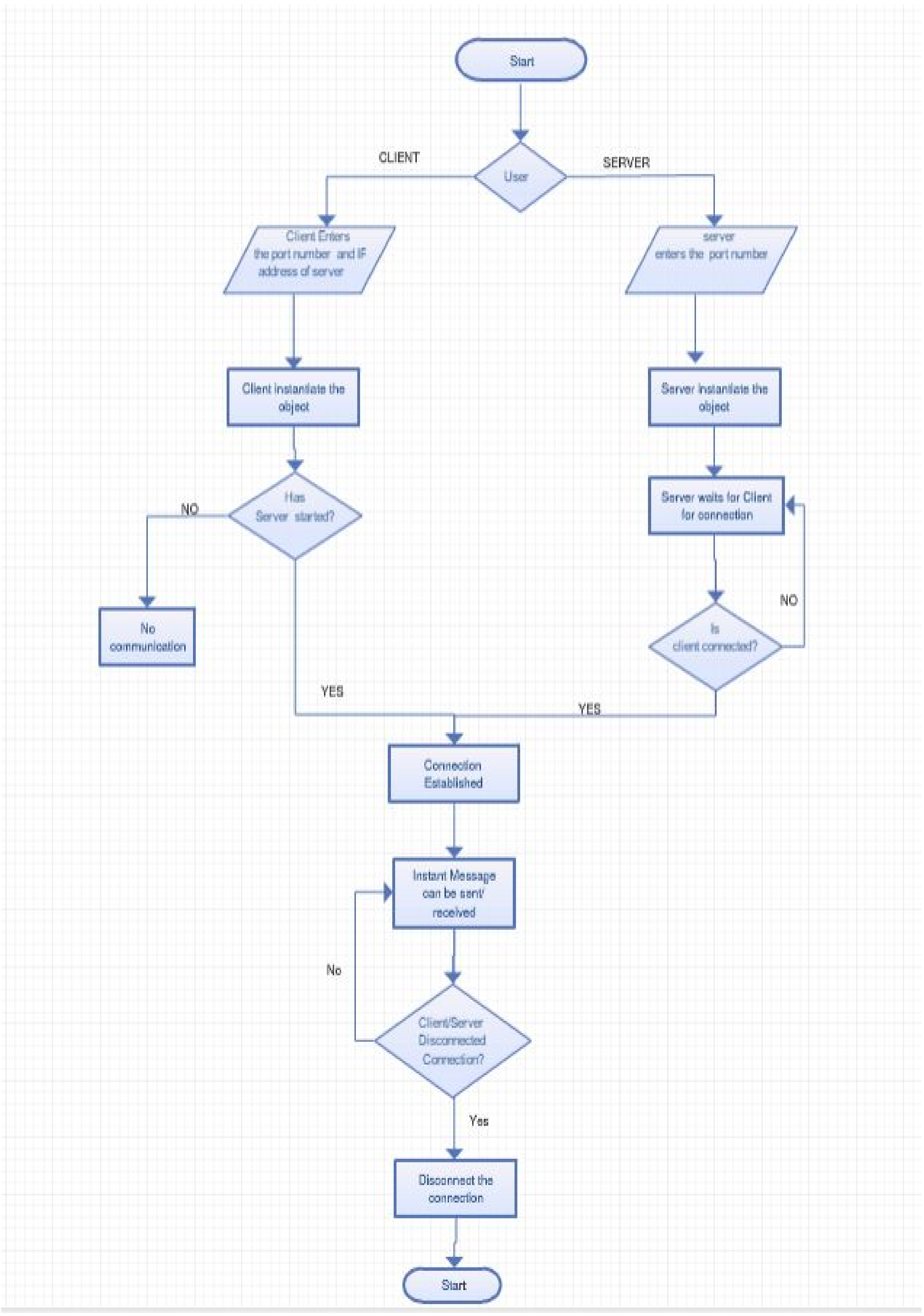
- Easy to understand
- Additional request handling have been included
- Easily manageable database

High Level design

Flow Chart Diagram






Start
User
CLIENT
SERVER
Client Enters the port number and IP address of server
server enters the port number
Client instantiate the object
Server instantiate the object
Has Server started?
NO
No communication
Server waits for Client for connection
Is client connected?
NO
YES
YES
Connection Established
Instant Message can be sent/ received
Client/Server Disconnected Connection?
No
Yes
Disconnect the connection
Start

Low Level Design

Algorithm

```
import socket
import signal
import sys
import threading
from threading import Thread

class Server(object):
    """doc string for the server"""
    def __init__(self):
        self.host = '172.25.14.62'
        self.port = 8888

    def activate_server(self):
        self.socket = socket.socket(socket.AF_INET,
socket.SOCK_STREAM, 0)
        self.socket.setsockopt(socket.SOL_SOCKET,
socket.SO_REUSEADDR, 1)
        # assign a port to the socket
        try:
            self.socket.bind((self.host, self.port))
            print("Launching HTTP server at ", self.host, ":", self.port)
        except Exception as e:
            print (e)
            sys.exit()
        print("Server successfully acquired the socket with port:",
self.port)
        print("Press ctrl+c to close the server.\n")
        # handle this keyboard_interrupt
        self.listen_client()

    def listen_client(self):
        # start TCP listen
        n = 5
        self.socket.listen(n)
        while True:
            # now a request from a client for a connection has arrived
            c_socket, c_addr = self.socket.accept()
            print("Got a connection from ", c_addr)
            # extract the information data from the client socket
            data = c_socket.recv(1024)
            # 1024 byte size data is received
            self.handle_request(c_socket, data)
            print("")
            c_socket.close()

    def handle_request(self, client_socket, client_data):
        msg = ""
        print("Following data received from client:")
        print(client_data.decode('ascii'))
        data = client_data.split( )
        request_method = ""
        request_path = "/"
        http_version = ""
        try:
            request_method = data[0]
            request_path = data[1]
            http_version = data[2]
        except Exception as e:
            print (e)
        # handle GET request
        if(request_method == 'GET'):
            if(request_path == '/'):
                request_path = '/index.html'
            path = 'Resources' + request_path
            file_type = request_path.split('.')[1]
            if(file_type == 'php'):
                path = '/var/www/html' + request_path
            if(request_path == '/favicon.ico'):
                print("favicon.ico request")
                client_socket.send('HTTP/1.1 200 OK\r\n')
                client_socket.send('Content-Length: 318\r\n')
                client_socket.send('Connection: close\r\n')
```

```
                    client_socket.send('Content-Type:
image/x-icon\r\n\r\n')
                    File = open(path, 'rb')
                    msg = File.read()
                    client_socket.send(msg)
                else:
                    print('Request path: ',request_path)
                    msg = ""
                    try:
                        File = open(path, 'rb')
                        msg += 'HTTP/1.1 200 OK\r\n'
                    except Exception as e:
                        path = 'Resources/error.html'
                        File = open(path, 'rb')
                        msg += 'HTTP/1.1 404 ERROR\r\n'
                        msg += File.read()
                        File.close()
                    if(file_type == 'jpg'):
                        msg += 'Content-Type: image/x-icon\r\n\r\n'
                    msg += File.read()
                    File.close()
                    client_socket.send(msg)
        # handle POST request
        if(request_method == 'POST'):
            #print(client_data.decode('ascii'))
            data = ""
            try:
                data = client_data.split('\r\n\r\n')[1]
            except Exception as e:
                print e
                data = ""
            print("Data received: \n", data)
            msg = "HTTP/1.1 200 OK\r\n\nThanks for connecting."
            client_socket.send(msg)

s = Server()
 s.activate_server()
```

2. FAQs

What is HTTP Server?
HTTP server is a computer where the web content is stored. Basically HTTP server is used to host the web sites but there exists other web servers also such as gaming, storage, FTP, email etc.

How to use HTTP server?
Client makes a request at the IP of the server .The server tries to understand this request and then responds accordingly.

What Programming languages are used ?
Python

Visibility of the code - what's that?
Code is available on the github i.e its public. If anyone wants to change or has any suggestion is welcome to do so.

Which requests are handled by this server?
GET,POST,PATCH, HEAD and DELETE are handled till time further more may be added .

What will happen in case of there is an issue with my computer or Internet connection?
No connection will be established ,hence the server wont work .

On my machine the same code gives an error .Why ?
Try compiling the code using python 2.7. Hope it helps !

How to i send various HTTP requests like PUT, PATCH etc. ?
Install chrome add-on “Postman” on your machine. It provides user interface to send various http requests.

3. Comments

We are planning to add further more requests in future updates.Email services or file transfer services may be added in later versions . Security issues will be taken into account.We will try to simply user interface as much as we can.