**COMP 4621 Project**

**Hans Krishandi**

**20373403**

**MultiThreading**

The multithreading is simply just using the Python’s threading package. For every acceptance of a client socket, a new thread is created.

**HTTP Request Forwarding**

We first just take all the contents of the client request and extract information like the host, method, full URL, and relative URL. Next, if the requested file is not found in the cache, the we start performing the forwarding. We convert the original request URL into a relative URL and change the header Proxy-Connection to connection. Finally, we just initiate a client which makes request from the converted request. The result is later stored into the cache

**HTTPS Request Forwarding**

Same first step as the HTTP request, we extracted the information. If the method is detected to be “CONNECT”, then it’s an HTTPS request. Hence, we first initiate a proxy client. And then we send a ‘HTTP/1.1 200 Connection Established\r\n\r\n’ to the original client.

Next, we create a loop that takes the request from the original client and forward it using the proxy client, followed by proxy client receiving the response and immediately sending it back to the original client.

**Access Control**

During the extraction of the information mention above, we also get the host. We then check whether the host is inside the block list. If it does, then we sent a 404 Not found to the client and close the connection socket.

**Caching**

For each requested file we check the cache first, whether its stored in a Python dictionary. Each cache file is stored in a with key name corresponding to the host name and its relative path. If file exists, then we read from the file and send back to the client socket. If not then we do the HTTP forwarding and store the response to the cache.