HTTP SERVER

DESIGN DOCUMENTATION

### Team 3

### Robbie

### Caleb Post

### Grant Smith

# Chosen Language

Our team chose Java as our programming language. It provides us with a simple language that all three members of our team have worked with before and have some familiarity with. This enables us to focus less on details in the language and more on the implementation of the ideas behind a web server itself.

# Socket Infrastructure

Java provides us with two different sockets. It provides us with the Socket class, an implementation of the socket architecture which supports reading and writing from data buffers which the socket automatically handles transfer of packets from, and the ServerSocket class which allows us to bind new instances of Socket once an incoming connection to the ServerSocket is detected. This means that we don’t have to create one socket which when it receives a connection request we then pass connection to a different socket, instead we have to only accept the incoming socket that ServerSocket generates for us. This greatly simplifies the code.

# Stateful Connections

HTTP by design is stateless, however, if you wish to implement state you must use cookies. The server will set and retrieve the cookies, while the website has the responsibility to use the information in them.

For our server, each time a user makes a new connection and does not send a cookie, the HTTP server will send a random cookie to that user. As the user interacts with the website, the cookie will be updated by the server. For each item requested, the cookie value will be updated to reflect the number of requests made by the user.

## Example:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GET /index.html HTTP/1.1 Host: localhost:8000 | | | |
| **browser** | | -------→ | **server** |

HTTP/1.0 200 OK  
Content-type: text/html  
Set-Cookie: chocolateChip=1; Expires=Wed, 09 Jun 2021 10:18:14 GMT   
(content of page)

|  |  |  |
| --- | --- | --- |
| **browser** |  | **server** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | GET /anotherPage.html HTTP/1.1 Host: localhost:8000  Cookie: chocolateChip=1 | | | |
| **browser** | | -------→ | **server** |

HTTP/1.0 200 OK  
Content-type: text/html  
Set-Cookie: chocolateChip=2; Expires=Wed, 09 Jun 2021 10:18:14 GMT   
(content of page)

|  |  |  |
| --- | --- | --- |
| **browser** |  | **server** |

# Security Component

Create a whitelist of allowed IP addresses, similar to an “.htaccess” file.

When a client requests access to the server, their IP is cross-referenced against the whitelisted IPs and, if those IPs do not include the client’s IP, the connection is refused. Otherwise, the connection is accepted as usual. Also, the server will deny any ‘GET’ requests for the whitelist file itself.