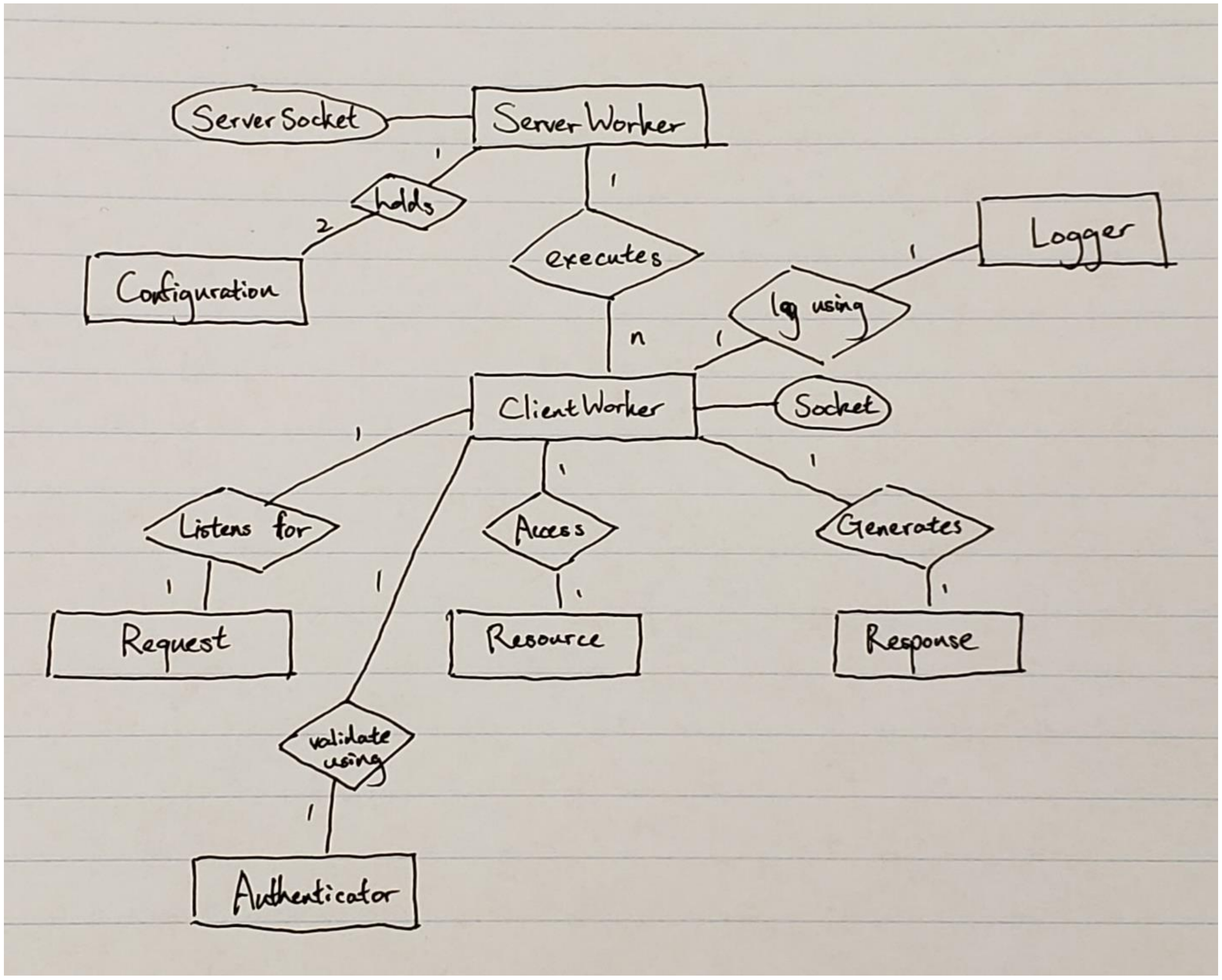
Eric Bristow 911754212
Tingfeng Wan 917912728
667-Server-Wan-Bristow

https://github.com/sfsu-csc-667-fall-2018/web-server-spring-2020-667-server-wan-bristow.git

We set up a socket for the server and then listened for incoming clients. When we found the client, we connect it to a temporary socket, have our server start to process requests and then free up the main socket. Our server gets the method, id and version info given by the client and any headers it sends us. It then processes the information given and then sends the object in the id, add or delete objects in the id. It will close the temporary socket when the user is done.



Grading Rubric

Category	Description		
Code Quality	Code is clean, well formatted (appropriate white space and indentation)	5	X
	Classes, methods, and variables are meaningfully named (no comments exist to explain functionality - the identifiers serve that purpose)	5	X
	Methods are small and serve a single purpose	3	X
	Code is well organized into a meaningful file structure	2	X
Documentation	A PDF is submitted that contains:	3	X
	Full names of team members	3	X
	A link to github repository	3	X
	A copy of this rubric with each item checked off that was completed (feel free to provide a suggested total you deserve based on completion)	1	X
	Brief description of architecture (pictures are handy here, but do not re-submit the pictures I provided)	5	X
	Problems you encountered during implementation, and how you solved them	5	X
	A discussion of what was difficult, and why	5	X
	A thorough description of your test plan (if you can't prove that it works, you shouldn't get 100%)	5	
Functionality - Server	Starts up and listens on correct port	3	X
	Logs in the common log format to stdout and log file	2	X
	Multithreading	5	X
Functionality - Responses	200	2	X
	201	2	X
	204	2	X
	400	2	X
	401	2	X
	403	2	X
Category	**Description**		
	404	2	X

	500	2	X
	Required headers present (Server, Date)	1	X
	Response specific headers present as needed (Content-Length, Content-Type)	2	X
	Simple caching (HEAD with If-Modified-Since results in 304 with Last-Modified header, Last-Modified header sent)	1	X
	Response body correctly sent	3	X
Functionality - Mime Types	Appropriate mime type returned based on file extension (defaults to text/text if not found in mime.types)	2	X
Functionality – Config	Correct index file used (defaults to index.html)	1	X
	Correct htaccess file used	1	X
	Correct document root used	1	X
	Aliases working (will be mutually exclusive)	3	X
	Script Aliases working (will be mutually exclusive)	3	X
	Correct port used (defaults to 8080)	1	X
	Correct log file used	1	X
CGI	Correctly executes and responds	4	X
	Receives correct environment variables	3	
	Connects request body to standard input of cgi process	2	

We had problems parsing the headers / collecting the necessary information. During parsing the server wouldn't ever stop so we had to add breaks in the while loop to stop after the first line to further parse the method, id and version. Then we parsed the headers and collected them in a string array.

Authentication was difficult because we kept getting an AccessDeniedException and it wouldn't ask the client for username/password. Getting the header to work as intended was difficult because we weren't sure how many headers and body lines get handled together.

We basically used the sample website given to test our code. Tested error 401/403, tested error 404 by asking for a file we knew didn’t exist. We pressed every button that came up in the sample website to test as many things as possible.