# Milestone 1 – Web Server Report Team RAJ – Angelica Rodriguez, John Krasich CSSE 477

## Architecture Diagram

**Protocol**

**Server**

**Client**

Protocol

Server

Web browser

Connection Handler

HTTP Response

Web Server GUI

HTTP Request

400

200

404

304

Request

Response

PUT

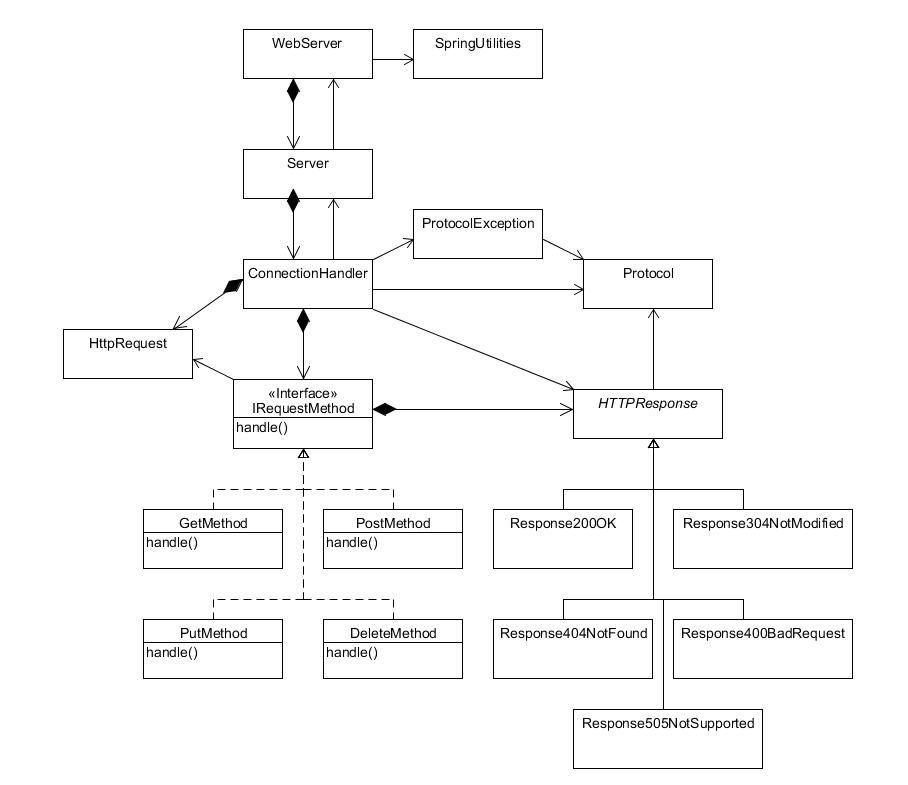
POST

505

GET

DELETE

## Detailed Design



Our refactoring of the web server utilized the following design patterns:

**Strategy Pattern** – The IRequestMethod interface allows for the various implementations of request handling to be completed in unique classes. This way, additional request handling can implanted with minimal changes to the ConnectionHandler class – simply add the new request to the ConnectionHandler’s map of request methods.

**Bridge Pattern** – The abstract HTTPResponse class is used by the ConnectionHandler to write the generated response back to the client. However, the responses vary depending on the response code. Using a bridge pattern, each different response’s implementation can be handled in separate classes without the ConnectionHandler needing to have any knowledge of how it is implemented.

## Further Improvements

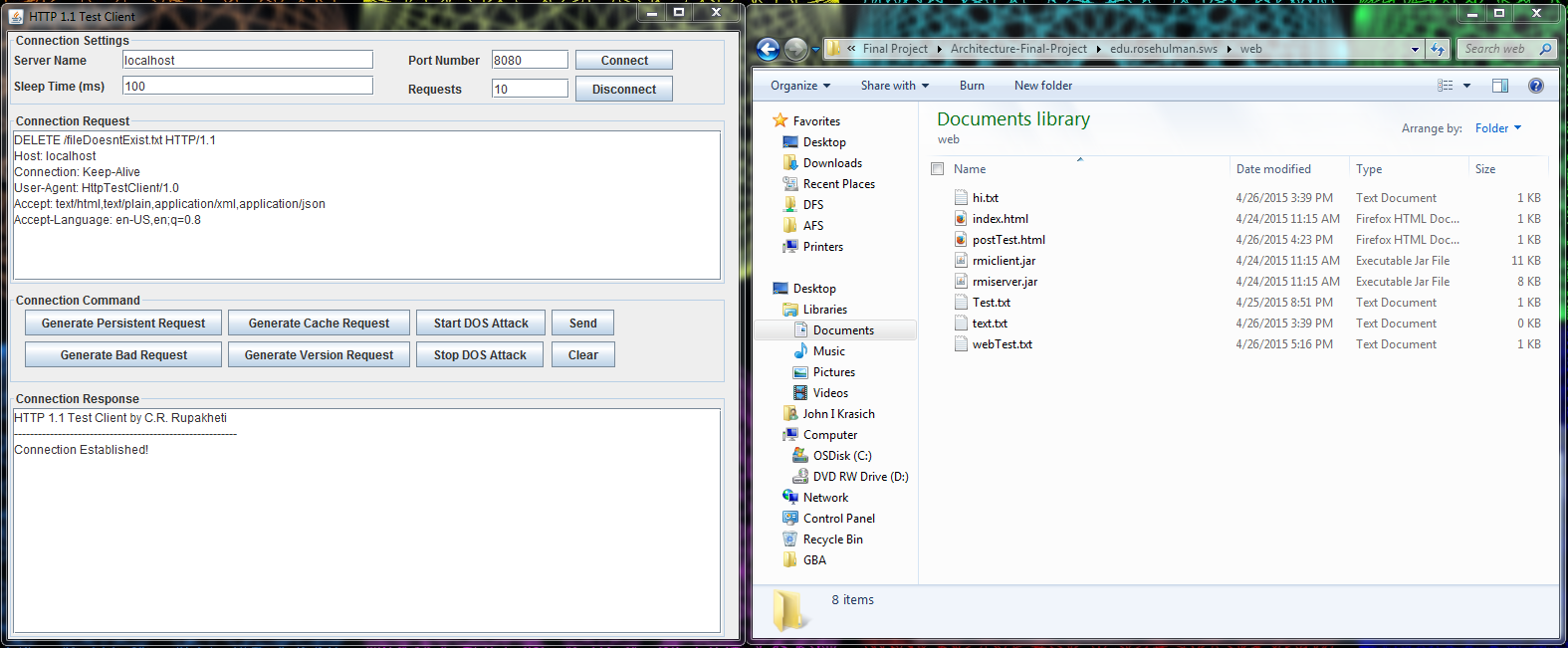
One area that can be further refactored would involve dividing the responsibilities of the ConnectionHandler class separately between requests and responses. The “run” method is rather long – breaking this up into different methods (or different classes) would make the code much more organized and easier to understand.

## Test Report

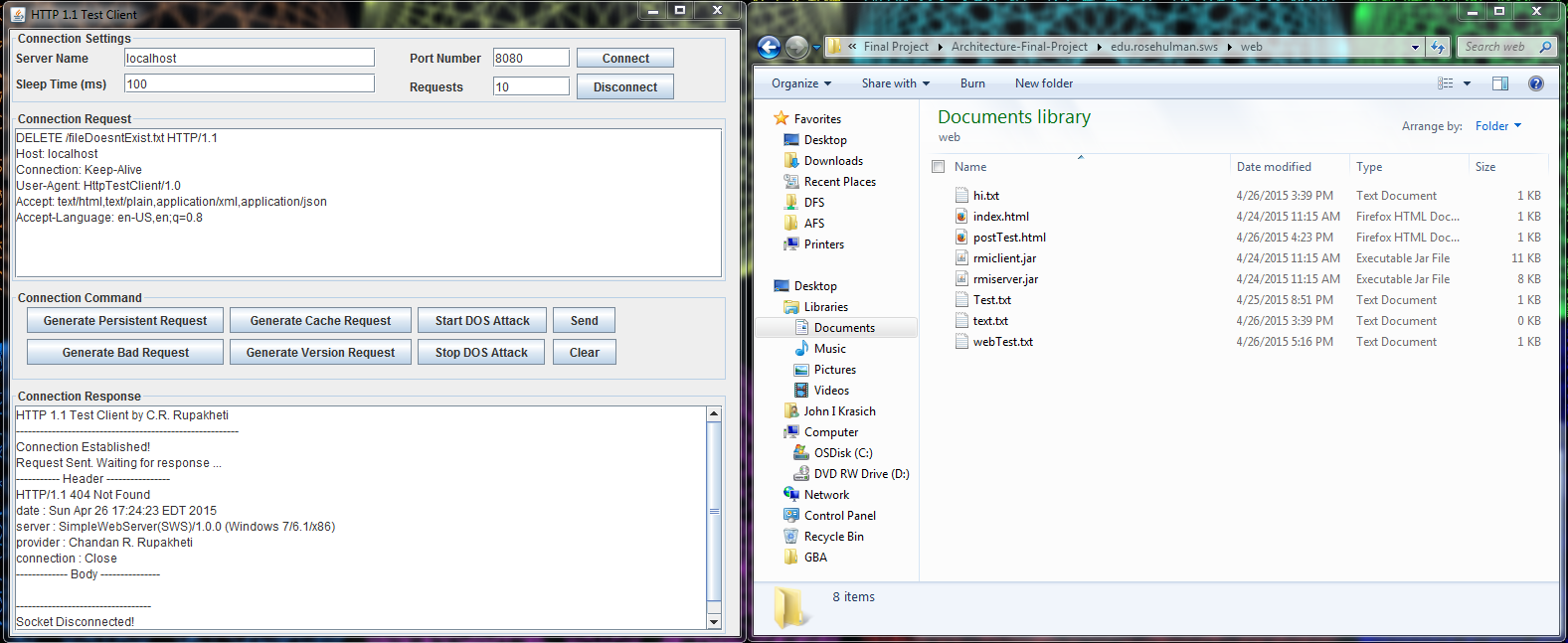
### Testing Utility

#### DELETE of Non-Existent File

Before:

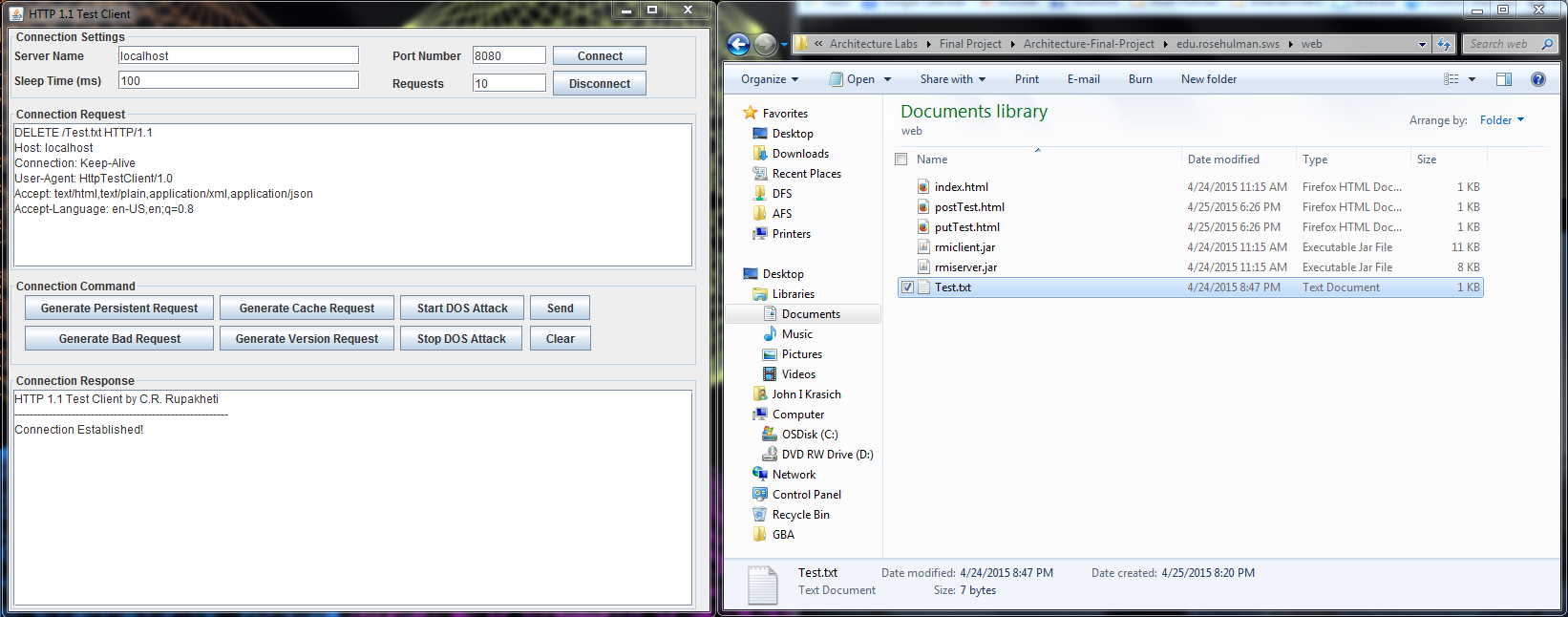


After: Response 404 Not Found

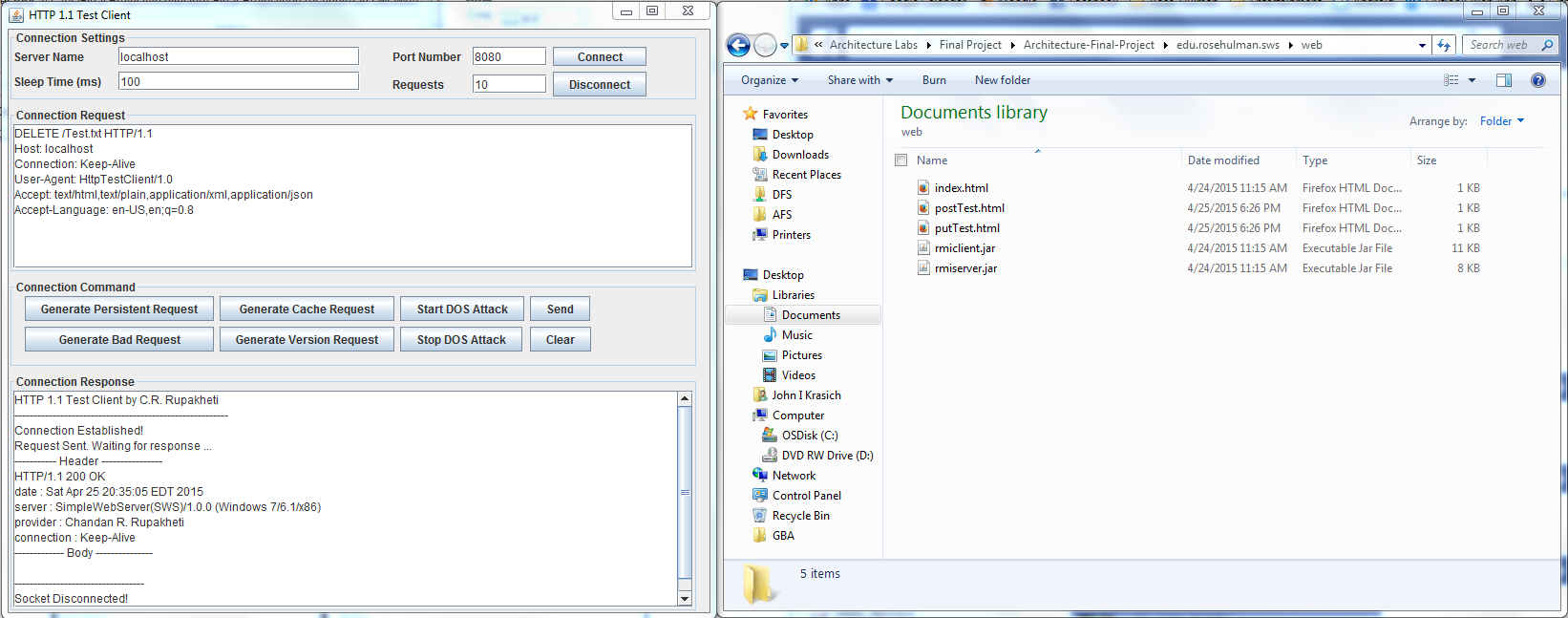


#### DELETE

Before:

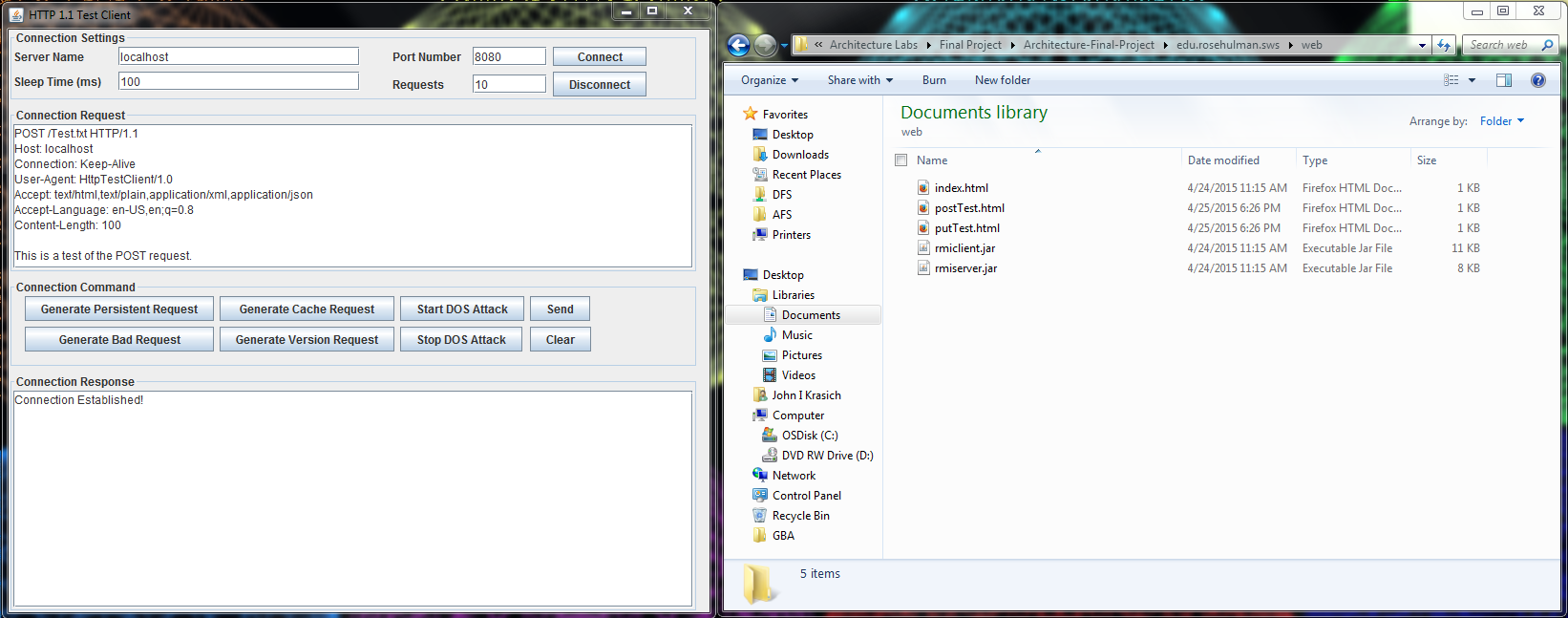


After: Response – 200 OK. File Test.txt has been deleted successfully

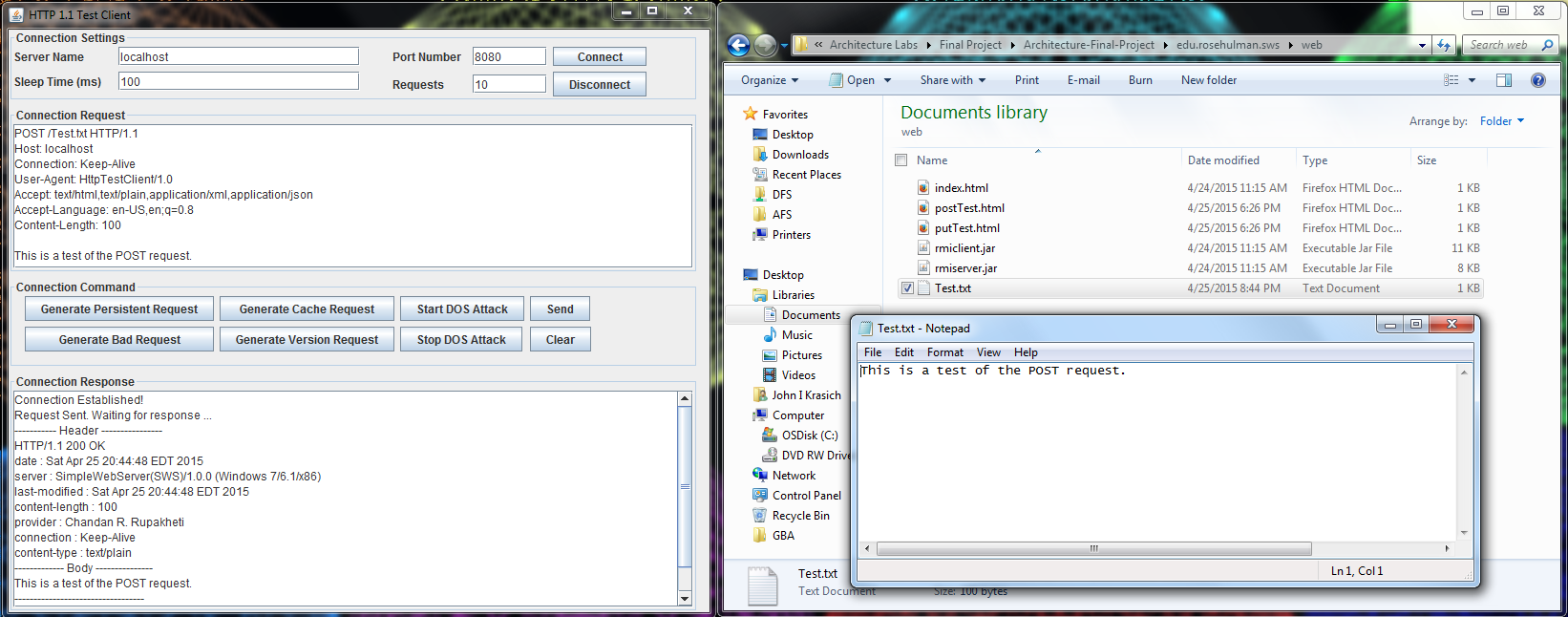


#### POST

Before:

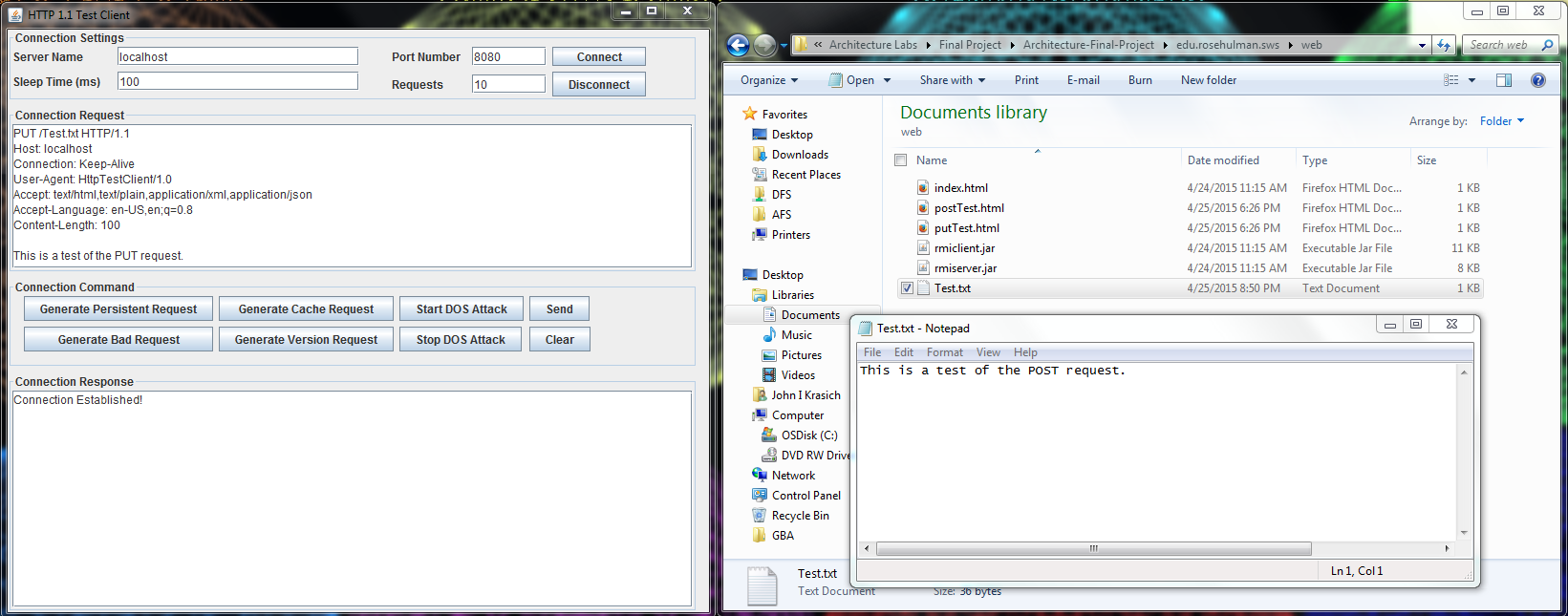


After: Response 200 OK. The file Test.txt has been created and filled with the body of the request.

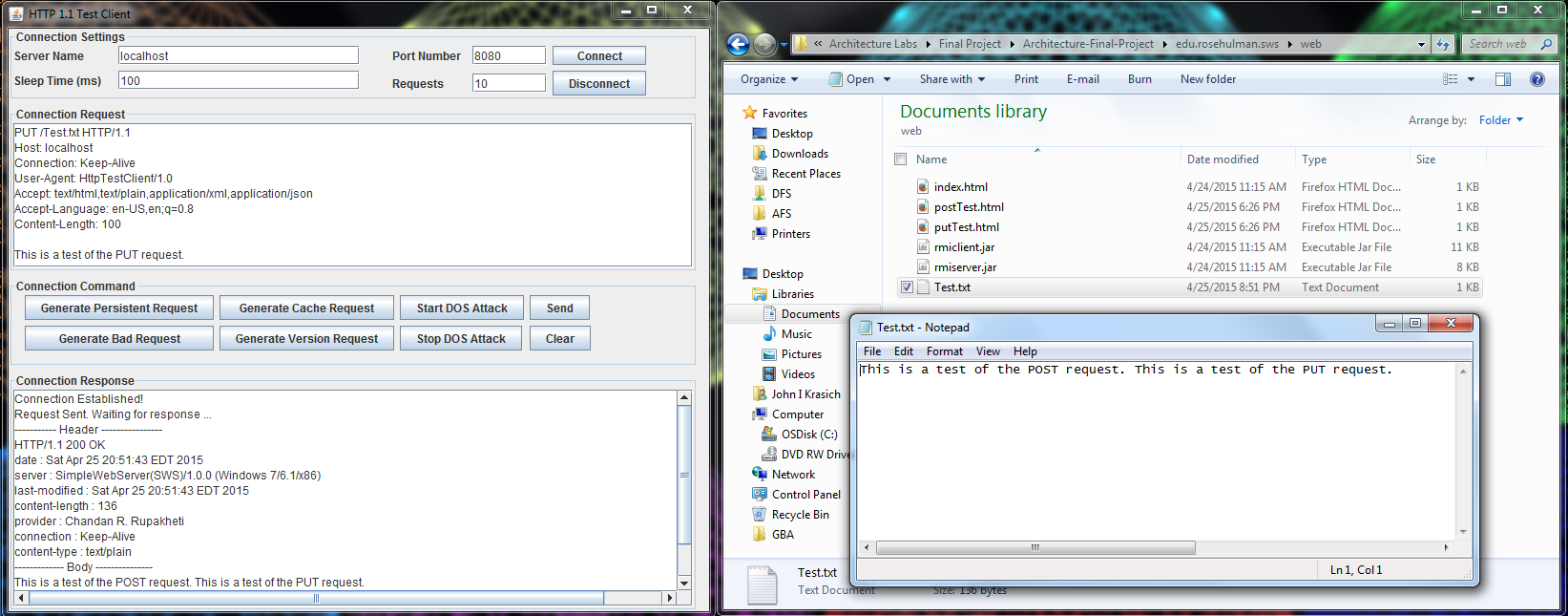


#### PUT

Before:



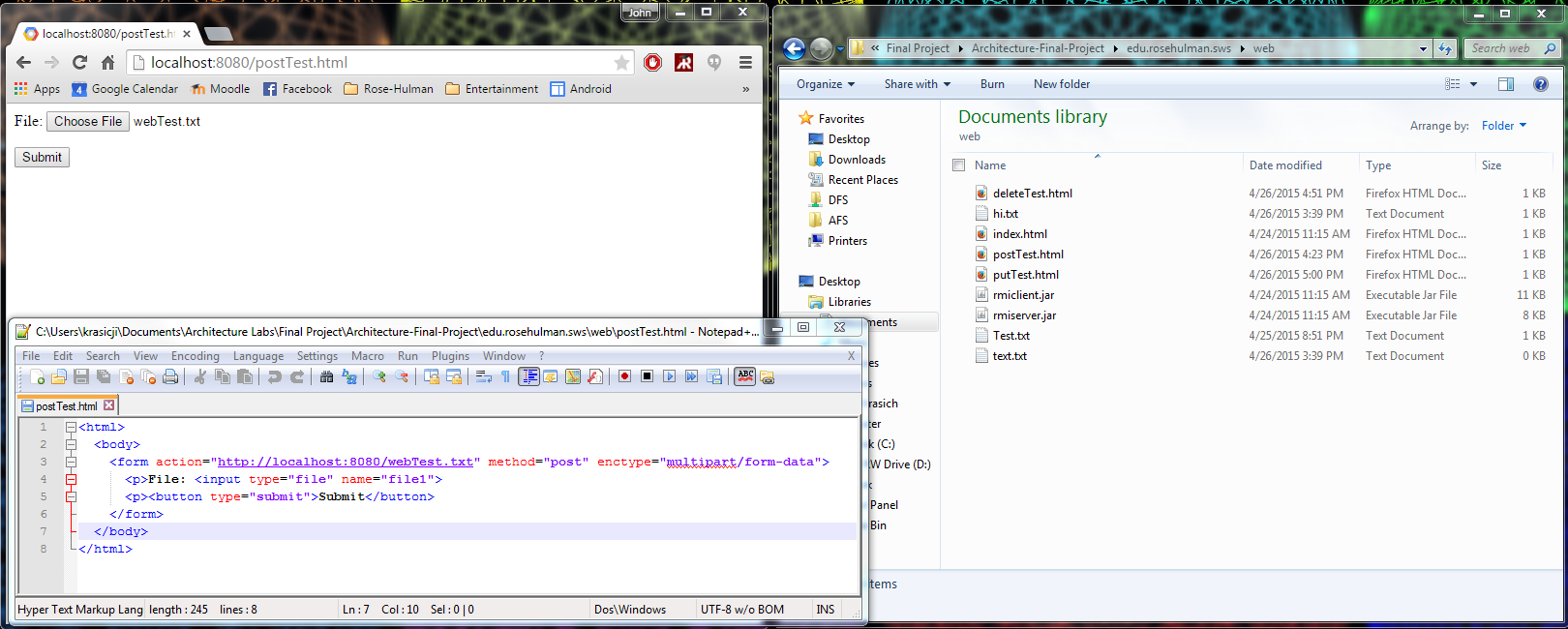
After: Response 200 OK. The body of the request was appended to the Test.txt file.

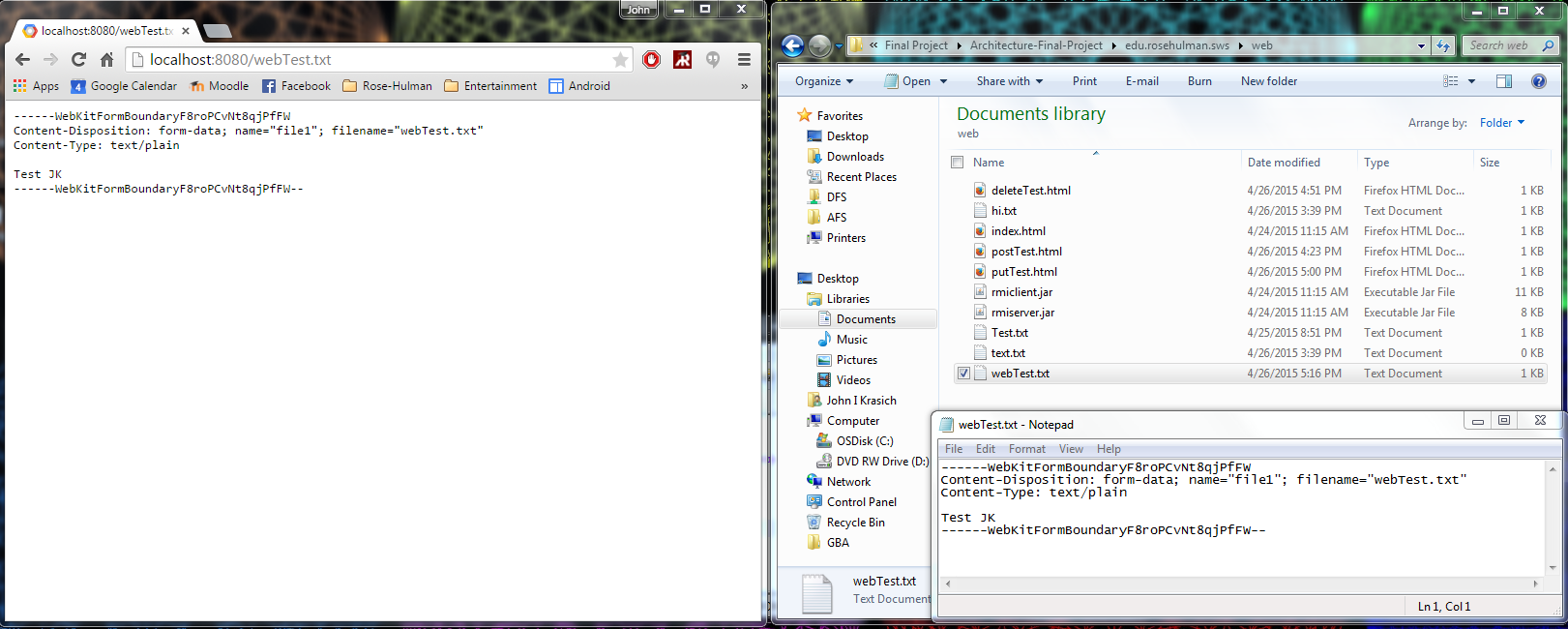


### Web Browser

#### POST

Before:



After: Response 200 OK – the text of the file was written into webTest.txt and returned as the body in the response.

# Change History – MS2

## Updated Architecture Diagram

**Server**

## 

**Protocol**

**Client**

Connection Handler

Server

Protocol

Web browser

HTTP Response

Plugin Handler

304

200

400

Servlet

Web Server GUI

404

StaticPUT

StaticPOST

Request

Response

505

StaticDELETE

StaticGET

UserServlet

## Updated Detailed Design

### Brief Description:

**Put into better wording please.**

The biggest changes made for this milestone were the addition of the PluginHandler and Servlet Interface. The ConnectionHandler now talks to the PluginHandler that is watching the Plugin directory for changes in order to maintain its Hashmap of plugins and recognize which one is the appropriate one to run based on the request it is given. The Servlet interface is what any user servlet must implement that contains the methods necessary to let the PluginHandler know what kind of request it is and the method needed to process it. We turned the basic GET, POST, PUT, and DELETE methods from MS1 into “static servlets” that will be run if no plugin is found for that kind of request.

## Feature Listing & Assignment

**Angelica Rodriguez**

* W-1: GET Requests
* W-2: POST Requests
* W-3: PUT Requests
* W-4: DELETE Requests

**John Krasich**

* P-1: Dynamic Loading
* E-1: Root Context and Configurable Route

## Test Report

## Future Improvements

**Correct me if I’m wrong but here’s what I thought from our conversation with Chandan:**

One thing we didn’t get to try but wanted to was to have the users supply a configuration file for the servlets with the information necessary, rather than have them hardcode the request code and create the JAR file. This would be a significant improvement because it would allow it to be easily modifiable, since they could make changes dynamically.