WEB SERVER

HOSTING

HOMEWORK TRACKER WEB SERVICE

By

Jeremy Stevens

PROG1090J Intermediate Java Programming

December 6, 2019

SUBMITTED BY: ACCEPTED BY:

_____ _____

AUTHOR, JEREMY STEVENS INSTRUCTOR, CHRIS CUSACK

Abstract

This report displays all related documentation to my Java project. I created a web server and locally hosted a web site on it to display it's uses.

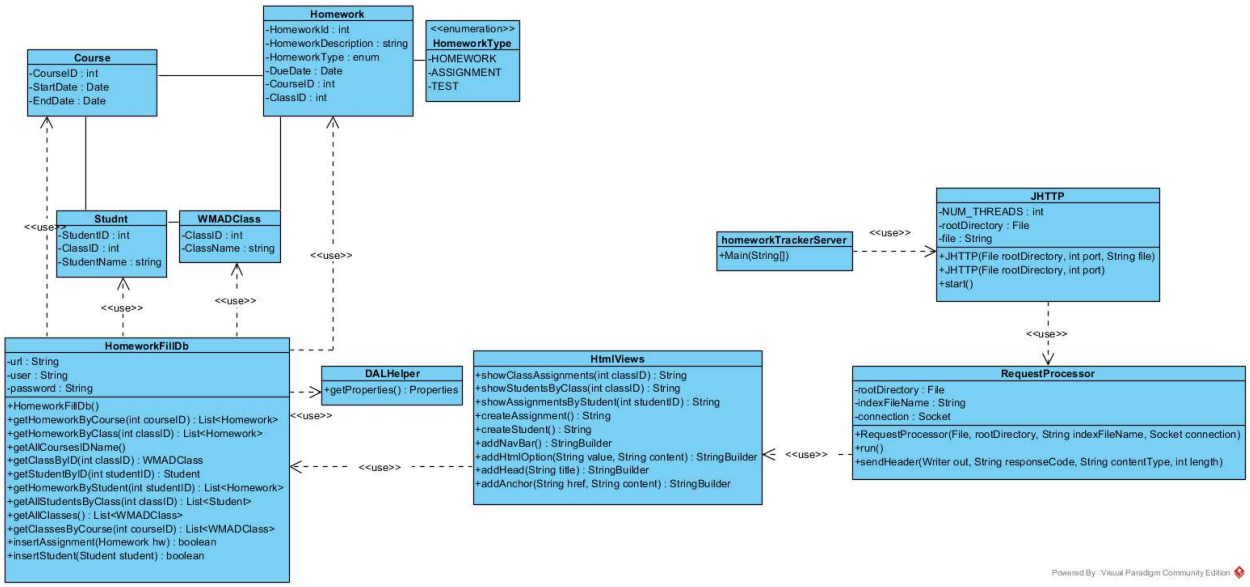
Contents

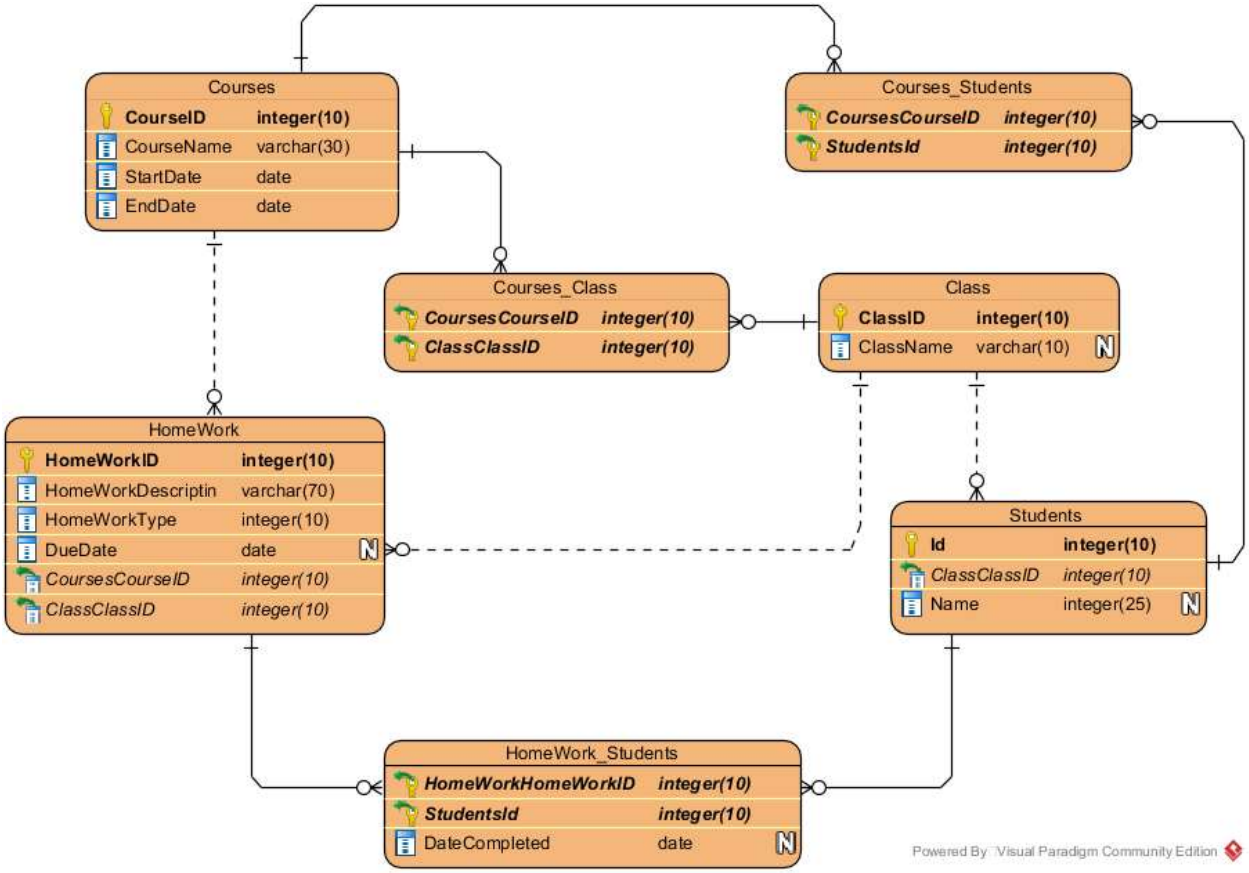
Introduction

The main purpose of this project is to explore the design of a home-made web server. I created a web site with some levels of business logic to display the uses of the web server. The website is a homework tracker. Each user can create a student account and assign themselves to a class. They can then either create a new assignment or check the current list of assignments for their class that their class mates previously created.

The web server is multi-threaded and logs exceptions and method calls. There Are three classes. Main which will spin a thread for the connecting client. JHTTP that handles the web socket and root directory. Request processor which will handle the routing.

Class Diagram and ERD





Research

The research for this project was most critical as I would not know how to do the core functionality without it. I entirely used *Java Network Programming* by Elliotte Rusty Harold for my research. I've learnt a lot reading it majority being sockets, thread pools, and logging.

Sockets

The books tutorial started with sending a string of time over port 13 and accessing it over telnet. That grew little by little until I had a working web server. The server socket listens for incoming TCP connections.

Threading

Threading as always been an interesting concept to me. This project is a great example for multithreading. You do not want your clients to have a slow connection because one client is slow. This project had a pool of 50 threads which was a new concept for me.

Logging

Logging can be used in any application. I've only dabbled with it in this project but I plan to research it more in future projects. Logging in a server is very important because there are many exceptions that can be caught. Exception handling and logging can help you build a better, more secure project.

Implementation

This project has two almost separate working parts. The web server, and the website the server is hosting. The web server is broken down into three basic components.

Main

Main is only a few lines of code, it sets the root file for where you would keep your web pages you would host. Then initialize the web server. The main method has this responsibility because you can use the following component multiple times with multiple servers.

JHTTP

JHTTP takes in the rootDirectory, port, and optional index file. With that starts an infinite loop with the sockets accept method. The accept method is a block method. Meaning the code will wait at this point until something happens, a client connecting in this example. When a client connects, we pull a runnable from our thread pool. This is processed in our next component.

Request Processor

This is where we handle request. Different GET or POST commands will do different actions. This is also where I handle my work around for the web site.

Web site – homework tracker

The index page is a .html file that gets sent. Everything else is done using what I called HtmlViews. Here I made a handful of methods that would stringbuilder append html text to be sent as the html body. I've created a few helper methods, but I could theoretically delve into this idea and create a rudimentary view engine.

I had to take this route to get generated html content. This was also the biggest challenge for me. Before I took this route I had no idea what I was going to do.

Testing

Testing for this project was very easy. As I mentioned before the book gives you bit by bit. So, as I developed the server, I could easily test these bits. The homework tracker homepage was also very easy to test. Because the server works so fast any changes I made I would near instantly see. I did not have the wait times I do with working with IIS. As I developed methods in the htmlViews class I would try to implement them in the server so I could test them. I did have many small issues that were easily found using breakpoints.

Possible Extensions

The next chapter in the book is about secure sockets. Over my winter break my plan is to go through that and upgrade this web server. Also I believe PHP would make a good complimentary language to go with this. Instead of dynastically building my webpages with my own albeit hacky view engine, I can send those PHP files as the html body. Granted I could be completely off the mark on this idea. I have not done through research.

Another possible extension is to work on the view model I have and try to take inspiration from MVC architecture.

Summary and Conclusions

The webserver is something that would easily be reusable. The classes have been mostly separated and it would not be hard to implement into a new project. The knowledge I have gained is valuable as understanding the lower-level technology will help understand applications and frameworks that use this technology.

References and Bibliography

Java Network Programming by Elliotte Rusty Harold

https://www.oreilly.com/library/view/java-network-programming/9781449365936/

User Manual

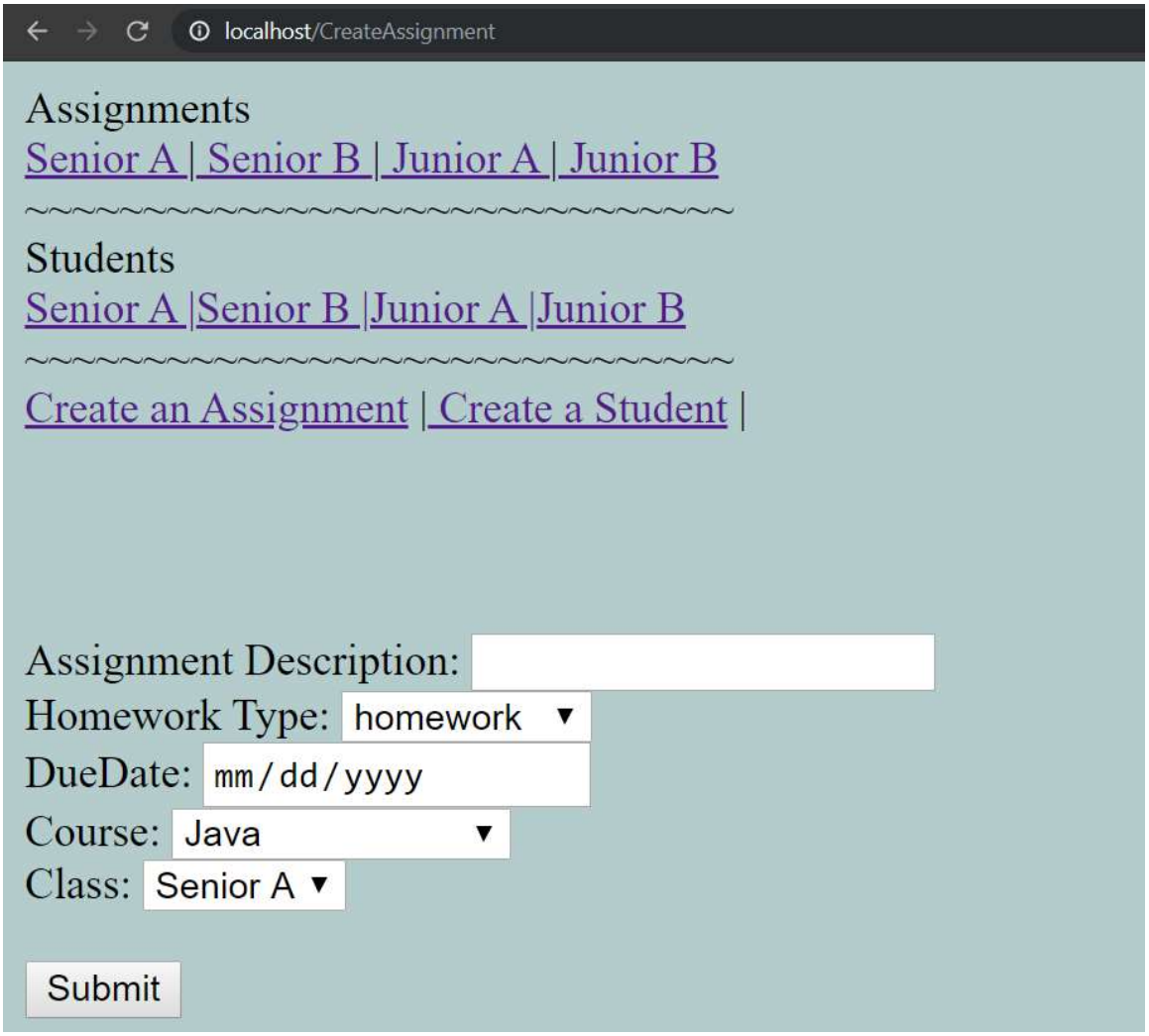
Begin at this beautiful web page

Assignments Senior A | Senior B | Junior A | Junior B
~~~~~~
Senior A
~~~~~~
Create an Assignment |

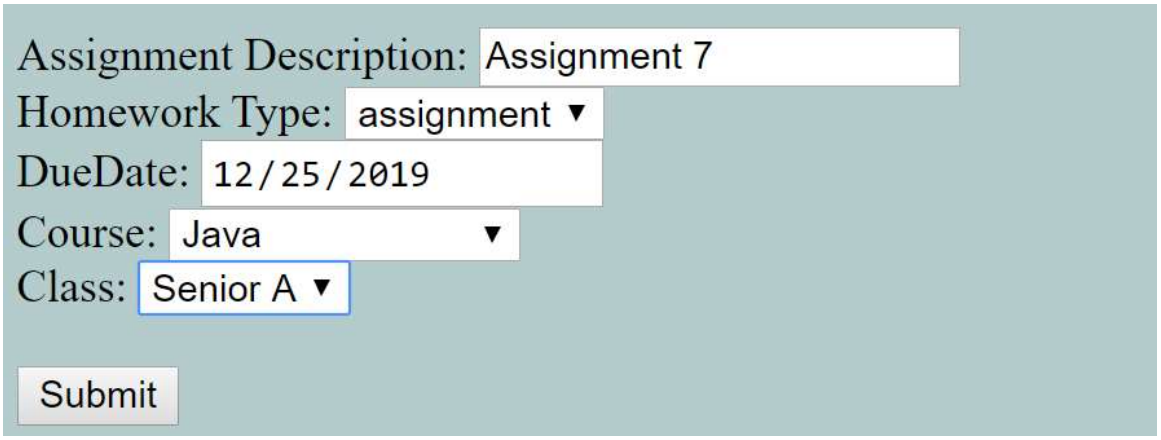
Welcome to WMAD Homework Tracker!

Click a link in the navigation to begin!

Let's begin by Creating an Assignment.

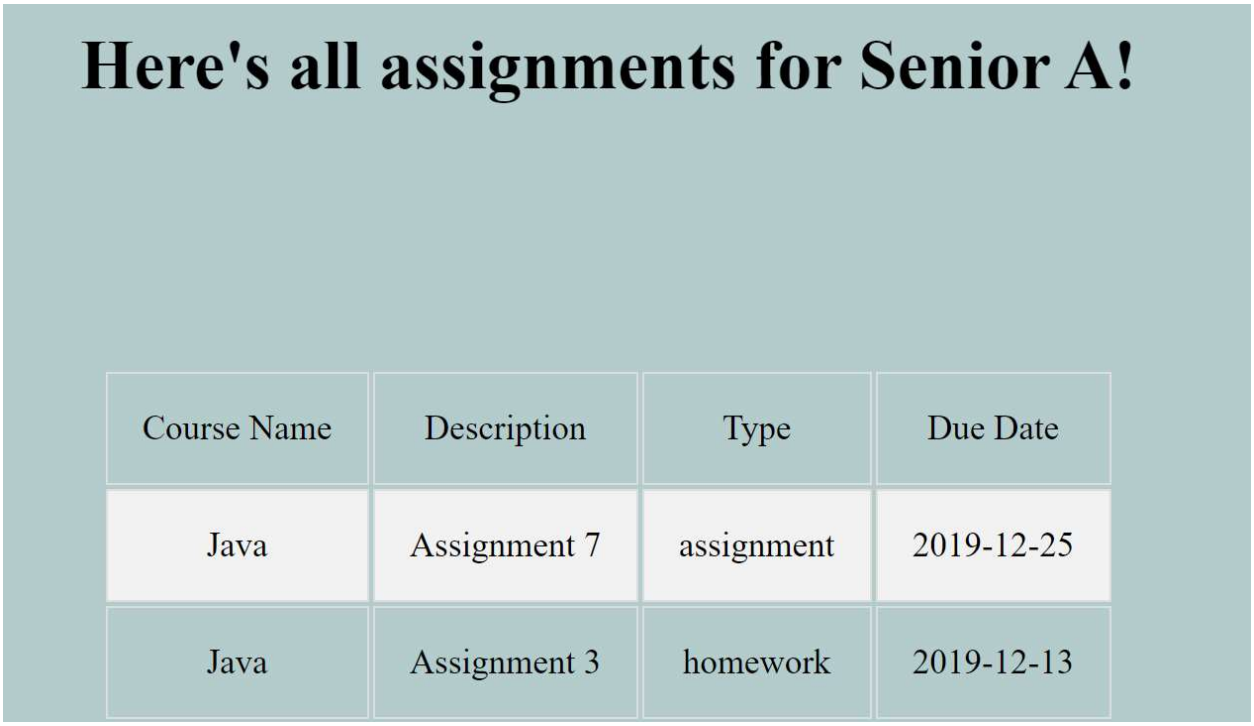


Enter in information. Due date is not required.



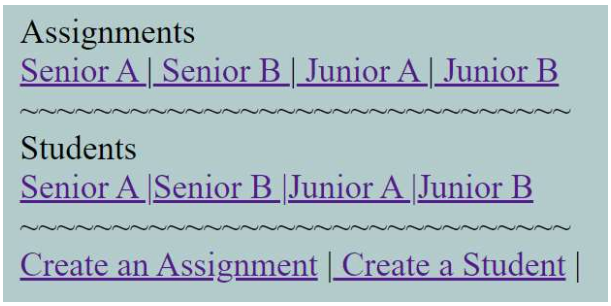
Submit

Click on Senior A under assignments

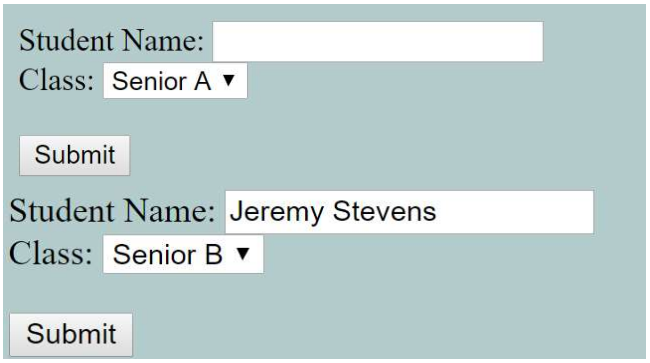


All assignments pre class that are not pass due are displayed.

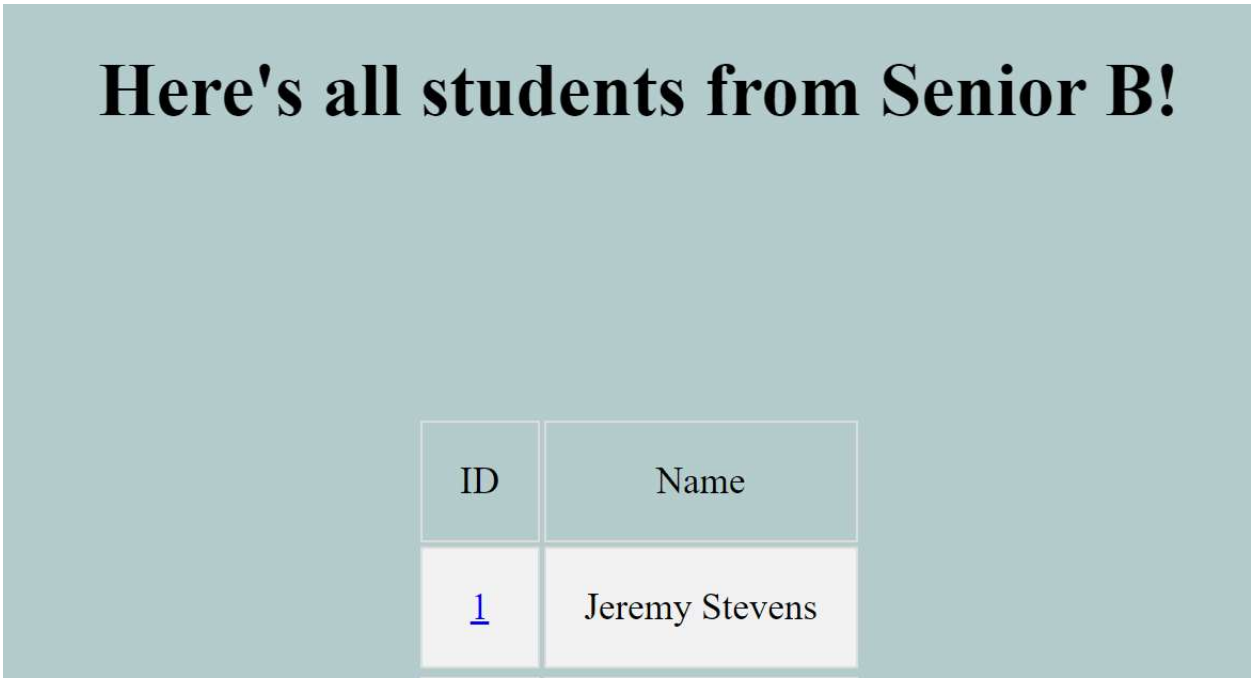
Now let's create and view students. Press the create a student link in the navigation.



Fill



Click on Senior B



Tada!