**EGRE526/CMSC506---Design Project Guidelines**

**Summary**

The purpose of this project is to provide a hands-on design and coding experience on computer networks.  For this project, you may work in teams of up to 2 students. Three project topics are presented in the blackboard for you to choose. As an alternative, each team may propose its own project (approved by the instructor) for the course. All proposed projects should focus on the computer network design.

**Schedule**

* March 27, 2016: Form Teams (team name, team members), email your team name and team members to the Instructor.
* April 14, 2016: System Design Report due
* May 10, 2016: Project Demo and Final Project Report due

**Requirements and Constraints**

* Work in groups of up to **two** students each
* Students select your group members
* Each member must design part of the project and must write his/her **OWN part of** the final report. Each member must **clearly** show his/her individual contribution to the project.
* Essential to help each other **WITHIN** teams!  No discussion **BETWEEN** teams.
* Design for proper functionality and efficiency.

**Phase 1 - System Design Report**

**Each** **team** will submit a project system design report with the following contents in order:

* Title Page with project title, VCU class number and title, date of submission, team name, and members.
* Project Specifications and Description, expanded and clarified from these given specs.
* Your overall program structure
* Team member responsibilities for the project.
* A general timeline to finish the project.

**Phase 2 - Final Report**

**Each team** will submit a final team report that includes the following contents in order:

* Title Page (team name, members, class number and name, submission date)
* Table of Contents (TOC)
* Project Specifications and Description.
* Detailed algorithms.
* Detail implementation.
* Analysis: functionality, and testing results.

**Each team** will email to the instructor and TA a **zipped file** (file name is your **EGRE526*team name*.zip**) which contains all electronic documents related to this project. This includes the source code of your implementation also.

In addition, each **team member** will submit a separate short individual report that includes the following contents in order:

* Title Page (team name, your name, class number and name, submission date)
* Your **individual contribution**, be specific.
* Evaluate your team member’s work.
* Any suggestions or comments on the project.

**Grade**

* Projects will be graded on the documentation, functionality, and teamwork.
* Grade = system design report (20%) + Project functionality (45%) + Final Report (35%)
* Deduct 5 points per day late on each phase

**Build an FTP service using TCP**

In this project you will design an FTP using TCP protocol.

1. Design an FTP service using TCP supporting single/master file server and multiple clients.
2. The file server is always running, waiting for clients to download/upload files.
3. Initially, the file server contains some files, but the file pool increases as clients upload files. The files on the server are distributed in a specified number (e.g., 8) of directories to emulate files being stored in a distributed storage system.
4. The file server implements a hash table with each entry mapping a file name to the directory ID where it is located. Choose your own hash function.
5. When a client requests to download or upload a file, the server first consults the hash table to find it directory location based on the file name.
6. The server should support multiple clients downloading/uploading files simultaneously.
7. When finishing uploading/downloading each file, the client should report file transfer rate.
8. The server should gracefully deal with different scenarios by responding appropriate CODE, for example, file not existent, too many files uploaded, out-of storage space, etc.

You are welcome to add more functionalities.

Bonus Points (up to 10%): Vary the number of multiple clients (i.e., concurrency) downloading/uploading files, and report the average transfer rates under different concurrencies.