CSE Design Competition 2018

Dear CSE Students:

Here is your *exclusive* chance to win attractive cash prizes, bragging rights, etc., by entering the inaugural (2018) CSE Design Competition!

To enter, you must be officially enrolled as a student (excluding audit students) for the Spring 2018 semester of the ISTD 50.005 CSE course, and your entry should be an extended version of your NS programming assignment. If you decide to enter, please take note of the details posted below.

Happy programming, and looking forward to receiving your submission!

Sincerely,

CSE Teaching Team

I) IMPORTANT DATES and TO-DO ITEMS

1) Register your team (which should be your original team for the NS programming assignment) by 11:59pm, April 5th (Thurs). The registration link will be announced by March 29th (Thurs) on eDimension.

2) Submit your entry by 12 noon, April 18th (Wed). The submission link will be posted on eDimension. Your submission must include:

- Full disclosure of any third-party code, libraries, software packages, servers, etc., used by your programs that were not implemented by you.
- Full source code of the part that was implemented by you. This submission is for evaluation purposes only; you will retain ownership of your own code.

- A succinct but clear description of up to three main features you would like to highlight for your program.
- A README file that provides clear instructions on how to use your program and invoke its key features.

3) Demo your submission to the judging panel on the afternoons of April 18th (Wed) and April 20th (Fri). The demo will be about 20 minutes, and at least one member of the team must be present. The demo schedule and venue will be decided and announced later.

II) PRIZES

First prize: $500

Second prize: $200

Third prize: $100

We reserve the right to share or not award any of the prizes based on the merits of the submissions as decided by the judging panel. All decisions by the panel will be final.

III) DESIGN IDEAS

- Implement an anycast design. For our purpose, anycast means the ability to reach any one among several available servers to perform a job. Presumably, you'll select the one that currently gives you the best performance. To do that, you might, for example, perform periodic pings to the different servers to dynamically track the RTT. You might then use *exponential averaging* to smooth out the RTT samples while giving more weight to more recent ones. Then you pick the server that has the current lowest (exponentially averaged) RTT.

- Implement a login system. To only allow authorized clients to upload files, the server should request a login username and password from the client and return a session cookie if the credentials are valid. Subsequent uploads would require the client to encrypt using the session cookie as a key. Think about how would you prevent sending passwords in plain-text and how to store them safely on the server.

- Implement peer-to-peer (p2p) file syncing. Suppose that the client needs to upload a single file to many different servers. It would be inefficient if the file is uploaded to each server one at a time. Assume that all the servers are in the same network and are already authorized to sync files with each other. Design a system to allow the server which the client uploaded to (can by any of the servers), to sync the uploaded file to all other servers in a p2p fashion. You may want to read up on how bit-torrent works. To demonstrate the feasibility of your p2p system, you may want to compare the total upload time with and without p2p syncing.

The above ideas, as well as any of the previously suggested ones, are for your reference only. Please feel free to create and implement your own cool ideas!