Grading for CMSC626 Project Instructions and Criteria

We will grade your project based on the following factors:

1. Your demo presentation.
    - We will randomly ask you 3 questions during your demo presentations as the grading points listed below.
2. Your design doc.
    - Your software design doc should detail your code design. Make sure your TA can understand your design.
3. Your code
    - Your code should be bug free.
    - Your code should reflect your design. If you changed your design during the implementation, update your design doc and let your TA know.
    - Your code should have comments and a consistent style. This means if you start out indenting three spaces and then go to six, this is not a good style.
4. Your project report
    - You should clearly address the problem and your solution in your report. Your report should include how you solve all the requirements in the project description.
    - Write a document of your project in all detail (benchmark should be put into the experiment part), and turn it in along with your project's code by the final deadline.
    - Take a look at the list of write ups from past years (2012): http://css.csail.mit.edu/6.858/2012/projects.html and 2013:

Grading points:

1. This is a P2P file system. - Show p2p via design doc and code implementation. Failure to do that will result in the failure of the project.
2. Users can create, delete, read, write, restore files. – Show a user create, delete, read, write, restore files on the primary server. Then show each replica (at least 3 replicas are needed) will maintain the same updates. For example, a student needs to connect to a server. Then show us the above process by asking the teammates to open another 3 servers. Show the updates that are processed in each replica.
    a. Bonus is you can do async read and write.
2. A client should always see the latest version of a file, or at least that a client should never see an older version of a file after there exists a newer one. - Two students need to connect to their primary server (different server). Then demonstrate what happens when after one client does a write on the primary server, the other client can see the update on the other server.
3. Users should be able to set permissions on files and directories. This requires that your file system is able to name users. -- show this case: one user creates a file. The other user accesses the file without/with permission.
4. File names (and directory names) should be treated as confidential and not visible except to the authorized person. --Show it
5. Users should not be able to modify files or directories without being detected unless they are authorized to do so. -- show: two users accessing the same directory. One modifies the file(update file name). The other user should not see any updates. We recommend that you demonstrate what would happen if an unauthorized user tries to access the file (update, delete)

6. The communication between Peer to Peer should be encrypted. -- Show a user case.
7. A malicious file server should not be able to create or delete files or directories without being detected. - one student works as a server and the other works as a client. Show if it is detected if the server tries to delete/create a file.

Late Policy

Each day late will incur 10 points per later day penalty. After three late days, the project will not be graded. Saturday and Sunday both count as days.

Important Deadline:
Design doc date:Final deadline is Oct.21st
demo date: We will post time weeks before the presentations.