EC500 J1 2017 Final Project Grading Criteria

Item	Grading	Score	Comments
Overall	Technical Depth and accomplishment of the project	10	What was the level of technical accomplishment for this project
Demo/video	Performance of the Demo	20	Started on time, went smooth, no hiccups, degree to which the project "worked", the video sufficiently shows the project.
Technical Report	Supporting figures and diagrams	10	Clarity, organization, and completeness of figures, diagrams, etc.
	Discussion of Failure	10	Discussion of failed / unimplemented portions of the project
	Technical report on success of the project	30	Refer to more detail below
Git/commits	Completeness and adherence to weekly commit schedule	20	Refer to more detail below

Total: 100

The following items are intended to capture the current state of your project, show how you got to this point, and what you learned along the way.

Overall:

This grades the overall difficulty of the project. An LED blinking project that works without any connectivity at 100% may not be considered the same as a DHT project that works at 90%.

Demo/Video (1-2 minutes):

This is comprised of two factors. One is the in-person demo (during demo office hours), and the metrics are on how well you understand the project, and how well the project works in-person. The demo is critical to the grade. Without the demo, none of the other grading metrics apply (i.e. the project will receive a failing grade even if a report is handed in, etc.).

The second part refers to a 1-2 minute video showcasing your project. In this video, you should show the totality of your project, and talk about your stack (I push this button, it publishes to such-and-such topic, which calls an API endpoint to do such-and-such). See "what to hand in" section at the end.

Technical Report (3-5 pages):

The final report is comprised of 3 parts. The first is the clarity and organization of the report, the second is a discussion of failure (anything that didn't work as you expected), and the third is the technical report of success.

Some items to consider:

- Aspects of the project which were not implemented and why
- Aspects which were implemented but not in original proposal
- Detail your stack, show how it relates to the course modules (crypto, authentication, API, M2M)

- Show a software/network architecture of how things are interconnected
- Talk about stack not directly covered in the course (such as DHT/XMPP/etc.)
- Significant data sheets, application notes, design templates, web resources
- Software is not limited to the code you write! Did you install a server? Are you using an API? Provide any libraries/packages that you used, including how to get and install them if necessary
- What did you learn from your project?

Individual reports can be 3 pages long, Team reports should be 5 pages. Also see the "what to hand in" section at the end.

Git/Commits:

This pertains to the quality and frequency of your commits. If software is committed on a weekly basis and shows progressive complexity in an organized fashion, then this is perhaps better than a sudden commit of 100% of the code on the last day. Weekly doesn't necessarily mean a commit every Monday night, rather it means a progressive development on the project.

Due date:

Everything is due at the beginning of our last class, Wednesday May 3rd 6:30 pm. All final project files/documents need to be committed to Git and/or uploaded to Blackboard Learn by that time.

What to Hand in:

- Please hand in the FinalVideo.mpg (or other video format) in two places: In your Git repo, and uploaded to the Blackboard Learn website.
- Please hand in the TechnicalReport.pdf in two places: In your Git repo, and uploaded to the Blackboard Learn website.
- Of course the last state of your code will also be reflected on the Git repo.