

# CS486C – Senior Capstone Design in Computer Science

## Project Description

<b>Project Title:</b> Evaluate2 – automated peer evaluation system	
<b>Sponsor Information:</b>	Dr. Eck Doerry, NAU-CS faculty

### Project Overview:

One of the most striking developments in the modern high-tech workplace in the last decade has been the move from larger monolithic divisions based on broad areas of expertise (e.g. "the database division") to small agile teams that work in semi-independent fashion on specific projects. This "small agile teams" model is mimicked precisely by the NAU Senior Capstone Design program: Teams of 4-6 students are asked to work semi-independently on real-world projects with real clients.

While learning effective teamwork is critical, it does pose a significant problem for the instructor: how to gain insight into specific internal team dynamics: how is the team functioning, who did what work on which parts, and so on. The most common mechanism used to address this challenge, both in real industrial practice and here in the Capstone sequence, is peer evaluations. In particular, we have developed an effective peer evaluation model in which all team members anonymously rate all other team members, and these ratings are then combined to arrive at overall rating for each team member.

While effective, this peer evaluation mechanism is very arduous for the instructor, who has to announce the evaluation, collect the email responses for all students, remind students to submit them multiple times, and then...the hardest part...arduously enter the data from the email into a spreadsheet just to perform basic statistics to get average peer evaluation scores for each student. Peer evaluation is most effective if applied early and often throughout a project ... yet this very high logistic cost limits both the frequency and quality of peer evaluations in Capstone. Having a simple, secure, and effective peer evaluation support system would improve our Capstone sequence delivery for decades!

The goal of this project is to build a simple, secure web-app that fully automates the peer evaluation process, making it simple and easy for both faculty and team members to complete peer evaluations regularly and often throughout the life of project. Specific requirements include:

- ✓ Allows for easy creation of faculty users by a system administrator. Faculty can then simply log in using their NAU ID using the standard NAUauth system.
- ✓ Faculty can see what courses they have created. New courses can easily be added by logged in faculty, e.g. by creating the course and uploading a standard course roster to populate the course.
- ✓ Within a course, the faculty member can create teams, and add students to teams.
- ✓ New peer evaluations can be easily added to an existing course. On "viewing" a course, the faculty member can see the peer evals for the course, whether they are open or closed, and the completion status for each. Individual peer evaluations can be given "due dates", and can be scheduled to "open" (become visible) on a particular date/time; and can be scheduled to "close" (become non-editable) on a particular date/time.
- ✓ Individual peer evaluations can be set to (a) announce to the class that they are available via email to members; and (b) can also be set to send reminders to any who have not filled out the eval at some interval.
- ✓ Students may log into the system using their NAU-ID, without creating any formal "account". They see which active (current semester) courses in the system they are part of, and can see all peer evaluations assigned for each course.

- ✓ Students may complete new peer evaluations for a course, as well as (if allowed by faculty in settings) potentially revise previous peer evals for that course.
- ✓ Faculty may view peer evals for any team in any of their courses. They can also download a spreadsheet for the course listing average peer eval score for every student on every team.
- ✓ The system must be implemented using AngularJS and must be hosted in the free tier of a cloud-based service like AWS.

### **Knowledge, skills, and expertise required for this project:**

All knowledge and skills required for this project should be quite simple to pick up at project startup. Beyond standard senior-level capabilities in programming and software design, helpful skills will include:

- ✓ Familiarity with creation of Web Apps, including specifically the AngularJS web app framework
- ✓ Familiarity with cloud-based hosting, using AWS or Google App Engine.

### **Equipment Requirements:**

- ✓ No special equipment requirements beyond access to workstations for software development.

### **Software and other Deliverables:**

Basic deliverables include:

- ✓ A fully-functioning and tested peer eval web app
- ✓ Complete users manual, written for low-level system administrators. Includes system implementation overview, step-by-step installation instructions, operator/administrator user guide, and troubleshooting section.

Additionally desirable deliverables:

- ✓ Installation of the software system in the free tier of AWS or Google App engine.
- ✓ Test of system for last peer evals for CS486