CS486C – Senior Capstone Design in Computer Science

Project Description

Insert Logo here

Project Title: Team Wrangler – automated peer evaluation system	
Sponsor Information:	**Dr. Eck Doerry, NAU-CS faculty**

Project Overview:

Over the last two decades, the organization of work groups within modern companies has moved increasingly from large cumbersome "divisions" to smaller, lightweight teams. To match this trend, higher education has also shifted increasingly towards focusing learning towards a team-based model, with students working in small teams to complete small or large projects; the skills learned in the areas of team communication, task distribution and re-integration, and other aspects of teamwork are at least as valuable as any specific technical skills learned. Finally, teams exist ubiquitously in our everyday world, bringing together complementary (we hope!) skillsets to work together on some goal or project. Although the concrete application scenario for this sponsor is student teams working on classroom projects, a broad range of other usage scenarios exist, including:

- Industry project teams
- Boy scout or Outward Bound activity teams
- Teams of medical personnel, either in training or actual practice
- Any other scenario is which it is useful to gather team efficacy information across the milestones of an evolving project.

Although these team-based work is the core metaphor for collaboration in almost every walk of life, they have also brought new challenges in terms of understanding team efficacy and, ultimately, the efficacy of individuals as elements of various project teams over some period of time. The key challenge here is how to gain insight into internal team dynamics --- who is doing what on the project, the quality of individual contributions, and the overall efficacy of each team member. Without effective mechanisms for gaining this insight, it becomes very difficult to assign appropriate credit to each individual team member, i.e., to make each team member's individual score on the project reflect their true contribution to the team's combined output. Ultimately, a lack of insight into teaming performance makes it extremely difficult to effectively evaluate organizational members in team-based scenarios over the course of projects and years.

One proven mechanism for monitoring team performance is an effective **peer evaluation system**, meaning some way for team members to regularly give feedback on each other's performance and contribution to overall project deliverables. Team supervisors can use this information in a variety of ways: as one piece of evidence in determining who is doing their work, as an "early warning system" for team dysfunction, and – especially as data accumulates over time and projects – to view each organizational member's evolution of efficacy over time.

Unfortunately, an effective peer evaluation system can be very difficult to manage practically, requiring great effort to remind team member to submit evaluations, gathering that input in some way, monitoring incoming input for signs of trouble in the team, and calculating evolving scores for all team members based on all input. Peer evaluation is most effective if applied early and often throughout a project … yet every increase in frequency aggravates the logistics and effort involved by all parties.

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

- ✓ Allows easy creation (e.g. via a sign-up mechanism) for new organization accounts within the system. That is, various universities or departments within universities could create an account, and then run all of their project-based courses within this account. This allows one site to support an infinite number of clients, each with many teams.
- ✓ Allows easy setup of new "companies". These serve as containers for available personnel. For the acdemic scenario, the company would correspond to a course. Must be easy to import text or spreadsheet lists of personnel for fast launch.
- ✓ Allows easy setup of a new project within a company. Projects are the containers for one or more specific teams working on that project.
- ✓ Allows easy creation of teams, including quickly populating teams from personnel in the company. A single person may be on multiple teams on different projects, as is often the case in real-life scenarios.
- ✓ Allows creation of milestones at which peer evaluations will be required by the supervisor, e.g., to match key project deliverables or events.
- ✓ Provides automated email reminders to team members about upcoming peer eval deadlines, including a link to follow to submit the eval.
- ✓ Provides automated mechanisms for detecting potential problems within a team (i.e., bad peer ratings) and notifying the team supervisor(s) for further attention.
- ✓ Provides one or more models for computing peer eval scores based on team member input; formulas to be included will be provided.
- ✓ Provides easy overview of current peer evals and combined total scores for all teams and participants. Should be able to drill in to view peer score summaries by company, project, team, or individual (across all teams the person in/has been on).
- ✓

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 web app frameworks and AJAX frameworks.

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, if possible:

- ✓ Installation of the software system on a server provided by the client.