Software Engineering Project: an Overview

Welcome to the Software Engineering Project component of the course. You will perform this component as a team.

Table of Contents

What will you do?

- Your team will develop a social network application, which will be deployed in the Cloud.

What is this application about?

- We call it **Emergency Social Network**, or **ESN**. The goal is to provide civilians with a social network that they can use during emergency situations. The system is different from other existing social networks because it is specifically designed to effectively support small communities of civilians seriously affected in case of natural disasters like earthquake, tsunami, tornado, wildfire, etc.

Where are the application's requirements?

- We have defined a set of use cases that describe the requirements. You'll find them in the Use Case Survey.

How will the SE Project work?

- Your team will develop the application incrementally in several iterations.
- Each iteration will have a different focus, emphasizing certain SE disciplines, concepts, and practices.
- Each iteration will have a different scope: we'll let you know before each iteration what that scope is and give you the details of the requirements for that scope.
- Each iteration will have specific deliverables.

Which technologies will we use?

- You'll use the same software stack you used in your Chat Room assignment: **HTML**, **CSS**, and **JavaScript** for the UI and front-end functionality; **Node.js** with **express.js** and any other supporting JS-based frameworks for the back-end; plus you'll need a database (e.g., **PostgreSQL**) for persisting data. You'll deploy your solution on a Cloud platform. You will be using **Heroku** for this purpose. Your choice of server-side technologies should be selected by the Cloud platform you're using.

What software engineering method will you follow?

You will follow an hybrid development method based on the following practices:

- Estimation
- Reflection (retrospective)
- Architecture Definition
- OOAD
- Pair programming
- Continuous integration
- Unit testing
- UI / Acceptance testing
- Code coverage
- Code review
- Static analysis
- Use case definition
- User story definition
- UI mockup creation
- Refactoring

What is the final-final deliverable?

- The final deliverable is a **presentation** in which your team will present your solution and development method. Of course, you'll have a fully functioning application as well. All your documentation should be complete and uploaded to your the github repo in the docs subfolder of your project.

How you will be graded?

- **SE Project Iterations.** At the end of each iteration, we will give you feedback about your progress. At the end of the project, we will give you a grade which will be a holistic

assessment of your work (related to your team method and use of practices, the functionality delivered, the quality of the work, and the maturity of your team).

- **SE Project Presentation.** Iteration 6 is the presentation.

About your contribution to the SE project

- Each team member should **organizationally and technically** contribute to the project. From a technical point of view, this includes applying the technical practices covered in class and **committing code to github** (if you use pair programming, take turns committing, or indicate who worked on the committed code in commit comments).
- Each team member should **pull his/her own weight** by contributing an equivalent amount of work. Make sure to sign up for tasks on the team Trello board. Enter your effort in the board for the tasks you have completed, independently or with a team member. Be honest. Provide ballpark effort: you don't have to measure minutes or be precise. Half a day of work with occasional interruptions = about 4 hours. Full day with occasional interruptions = about 8 hours. Adjust for more frequent or prolonged interruptions. Go from there.