

## TL;DR

- What: Each team must submit a document on behalf of all members of the team documenting the final design for their project including algorithms, data structures, and components
- Due: See Piazza/the calendar for deadline.
- Where: Google folder and GitHub

## Background

As we have discussed, one of the fundamental characteristics of software is that it will change. As it changes, the team that works on the software itself changes. While teams should strive to avoid too much up front design, a decent team will ultimately document their design in several ways so that others may be more easily able to modify and update the software. The first is the code itself, the ultimate form of documentation. However, as also discussed in class, human have limited cognitive ability and so this source code, designed for consumption by a machine, should also be accompanied by documentation intended for human consumption.

Moreover, as students at a university, you are subject to the philosophy of the university and its instructors. For students who graduate, the university is certifying that those students have demonstrated sufficient competence and/or critical thinking in their fields of study. Therefore, in addition to being a simulation of a commercial-world scenario, this assignment is also intended to help the instructors determine that you have sufficient software design competence and have demonstrated critical thinking as it pertains to software design. As the cartoonist Richard Guindon is purported to have said:

*“Writing is nature’s way of letting you know how sloppy your thinking is.”*

## Assignment

Each team must provide a joint software design document for your project. This document should extend and update your previous design assignment, and therefore can include what you had previously written, although you should take advantage of any feedback that was provided. The purpose is to convince the Instructor that you have given deep thought to the design of your software and understand basic principles of software design. This document should include:

- An overall description of the purpose and objective of your finished product, including a description, when appropriate, of the problem or gap it is trying to fill;
- A description of the users and the devices each will use to interact with your system
- A description of the use cases;
- Identification of the major components of your software, and the container (e.g. mobile application, web server) in which each component will execute, including data storage systems and technology (e.g. relational database? NoSQL database?):
  - For each major component, the function and non-functional requirements of that component;
  - Be sure to include how the components interact with one another – do they hold a reference to an instance of another object? Communicate via message passing?

- Identification of the classes, interfaces, and similar software modules you will use to implement the components, to include network interfaces (e.g. RESTful interfaces) where appropriate:
  - Where helpful, control flow graphs and data/information flow diagrams to highlight interactions between components or between users and components
- A description of the core algorithm your team is implementing, associated with the component in which it will be implemented
- Identification and description of the primary data structures used by your algorithm and used to encapsulate the primary data processed by your software

Finally, for each logical software module you identify, you must provide a time line indicating when you expect to begin (or have already begun) and when you expect to complete development, complete unit testing, and complete integration testing, as appropriate.

### **Grading**

This assignment will be evaluated for the following:

- Design: does the design represent good engineering practices; how well does it adhere to principles of loose coupling and incorporate practices that limit the probability and impact of failure; how easily can components be modified and updates as new features are required?
- Completeness: does the documentation completely describe the design and algorithms of the final system?
- Clarity: is the documentation well-written, precise, and easy to understand?
- Structure: Is the document structured and organized in a logical manner?

### **Form & Submission**

Create a document with the content described herein in your Google folder. Post PDF version of the document to your web page in the gh-pages branch of your repository. The document should use 11pt. font, single-spaced. See Piazza/the calendar for deadline.