# SE 181: Intro to Software Engineering & Development

Summer 2020

## Term Project

Applying software engineering practices on the development of a software system is an important objective for this course. Unfortunately, the 10-week term constraints the types of software systems we can realistically work on, both in terms of size and scope. To make this a good learning experience for all, you should work on a project that does not require much domain expertise, it is fun, and can be completed by **a team of 5 people** with a reasonable amount of effort.

After analyzing the pros and cons of various alternatives, I concluded that a good project for our course would be to **design and implement a software application that allows two people to play a game of chess interactively, from remote locations.**

The objective of this project is NOT to develop a chess program. Rather, the objective is *to develop software that provides an environment for two people to play a game against each other*. Your application should attempt to make this interactive experience as close to the physical playing experience as possible. This should be your guiding principle when designing the capabilities of your system.

As I do not want to stifle your creativity, I will not place many specific requirements on the capabilities that your software should provide. I want you, as a group, to decide the features of your application. However, I can “see” that your software should provide a graphical representation of a chessboard, and support move validations. A graphical representation of the chessboard should allow the end user to “move/slide” a piece into a new position, and when the move is completed, refresh the board on the opponent’s screen to show the new configuration of the board. Because players from remote locations cannot challenge illegal moves, a move validation mechanism will ensure that the players observe the rules of the game.

You can find information about the game of chess (e.g., legal moves, special moves, etc.) at <http://en.wikipedia.org/wiki/Rules_of_chess>.

**Deliverables and Development Expectations:**

The primary objective of this term project is to provide you with the opportunity to develop software as a member of a team using accepted software engineering processes and tools. Towards this objective you will be expected to produce the following deliverables:

* Requirements Document
* Design Document
* Test Case Document
* Source Code
* Release Notes

In addition, as a team, you are expected to implement the following practices:

* All source code development will be done under version control.
* Use a static analysis tool, as part of your development environment, and address issues reported by the tool.
* Use a code coverage analysis tool to determine the adequacy of unit testing.
* Use a bug-tracking tool to keep a record of and manage the issues found during testing.
* Once testing begins, release candidate builds must have a unique identifier. This identifier should be displayed either in the application splash screen or the in the “About” menu item.
* Release candidate builds should be created using specific Change Requests (CRs) (or whatever corresponding term is used by your bug tracking tool).

You are free to choose the programming language and operating environment. The only constraint is that the TAs and I should be able to execute your application on ether a Mac or a Windows computer.

You must complete and submit you project via Bb Learn by **Monday, August 24, 2020,** **11:59AM**.

We would like to have an Acceptance Test of your project. This will be in the form of an online demo and will be performed some time during the week of 8/24/2020.  
**All members of the team must be present** during the Acceptance Test.

**Grading**

The grade breakdown for your project will be as follows:

|  |  |  |
| --- | --- | --- |
| **Requirements Document** | 10% | **Required** |
| Design Document | 10% |  |
| **Test Case Document (w/ execution results)** | 10% | **Required** |
| **Source Code** | 10% | **Required** |
| **Release Notes** | 5% | **Required** |
| **Use of Version Control** | 5% | **Required** |
| Use of Static Analysis Tool | 5% |  |
| Use of Code Coverage Tool | 5% |  |
| Use of Bug Tracking Tool | 5% |  |
| Use of Change Requests for Candidate Releases | 5% |  |
| **Acceptance Test** |  | **Required** |
| Final Product Quality | 30% |  |

**Please note that if any of the “Required” items receives a failing grade the project will receive a failing grade.**