Code Review Checklist – Group 4

**General**

* Does the code work? Does it perform its intended function, the logic is correct etc.
* Is the code readable by other members of the team?
* Does the code follow planned architecture, code style, and commenting conventions?
* Is there any redundant code that would lead to bug fixes to be applied to additional area of the code?
* Is there any commented out code? Is so, Identify the creator and reasoning.
* Do all fields, key presses, and listeners work as intended?
* Are drop downs order in a way that makes sense?
* Are input fields flexible on input syntax?
* Can any of the code be replaced with library functions?
* Can any logging such as console prints be removed?Where third-party utilities are used, are returning errors being caught?
* Are invalid parameter values handled properly?

**Documentation**

* Do comments exist and describe the intent of the code?
* Are all functions commented including creator/date/parameters/return value/purpose etc?
* Is any documentation seem unusual, incomplete, or in an inappropriate format?
* Is the use and function of third-party libraries documented?
* Are all planned methods present?

**Testing**

* Is the code testable?
* Has this code been tested before, if so, are test up to date? Do they need to be updated, added to, or removed? Does the code pass previous test versions?
* Does test code have any errors? Does it encompass the full range of functionality of the code it is testing?
* Are arrays checked for ‘out-of-bound’ errors?
* Are exceptions correctly implemented?
* Would manual testing be more efficient than created test code?

Resource: <http://blog.fogcreek.com/increase-defect-detection-with-our-code-review-checklist-example/>