**Project 2**

**Document 2 – Development Requirements**

**These requirements are the same as for Project 1**

1. **Project Management**
2. You will use a GitHub *Project* to manage your development. Name the project, *Unit Testing*. You will have these columns: *To Do, In Progress, Complete.*
3. TLDR: Tasks will be entered as *Issues* in GitHub and added to the *To Do* column. They should be assigned to a group member, there, or when they are moved to the *In Progress* column. Upon completion, and after a pull request and review, they are moved to the *Done* column.

[You should read this, but some of it may only come into play on Project 2. For your project you are required to break work down into small tasks that 1-2 people will work on (some tasks may involve the whole group). These tasks should be estimated to take between 0.5 hours and 4 hours in length. Think about this carefully keeping in mind that you must work as a team, and that all team members should contribute to the development tasks. Tasks will be entered as *Issues* in GitHub and added to the *To Do* column. From there, they will be moved to an *In Progress* column and finally to the *Done* column.

This will be new activity and it might be a bit of a struggle to do this. Almost all software companies use this approach. Many times a project manager will develop these, sometimes in conjunction with developers. In some companies, a developer will simply go to the *To Do* list and “pull a ticket.” When complete, they pull another one.

Task/Issue breakdown is iterative. Your group will identify some initial tasks to get started on development. As issues are in progress or completed you will learn more about what you need to do. If a task turns out to be much bigger, or involve things you didn’t initially think of, then simply add them as new tasks, possibly closing out the initial task. Thus, Task/Issue breakdown is ongoing.

Sometimes a task may be to figure out what needs to be done, a planning session, which will result in concrete tasks that contribute to the development.

1. **Version Control**
2. Each person is required to work in their own branch with name: *lastName*.
3. When you have code on a branch ready to merge with master, you should use the rebase workflow and then open a pull request. Pull requests are required to be reviewed by at least one other member before merging.
4. **Individual Team Members**
5. All time spent on the project is entered into the Time Log as soon as you end working.
6. You should work consistently on the project. In other words, you should not do a minimal amount of work one week, and then double the next week. This ensures that project is not held back.