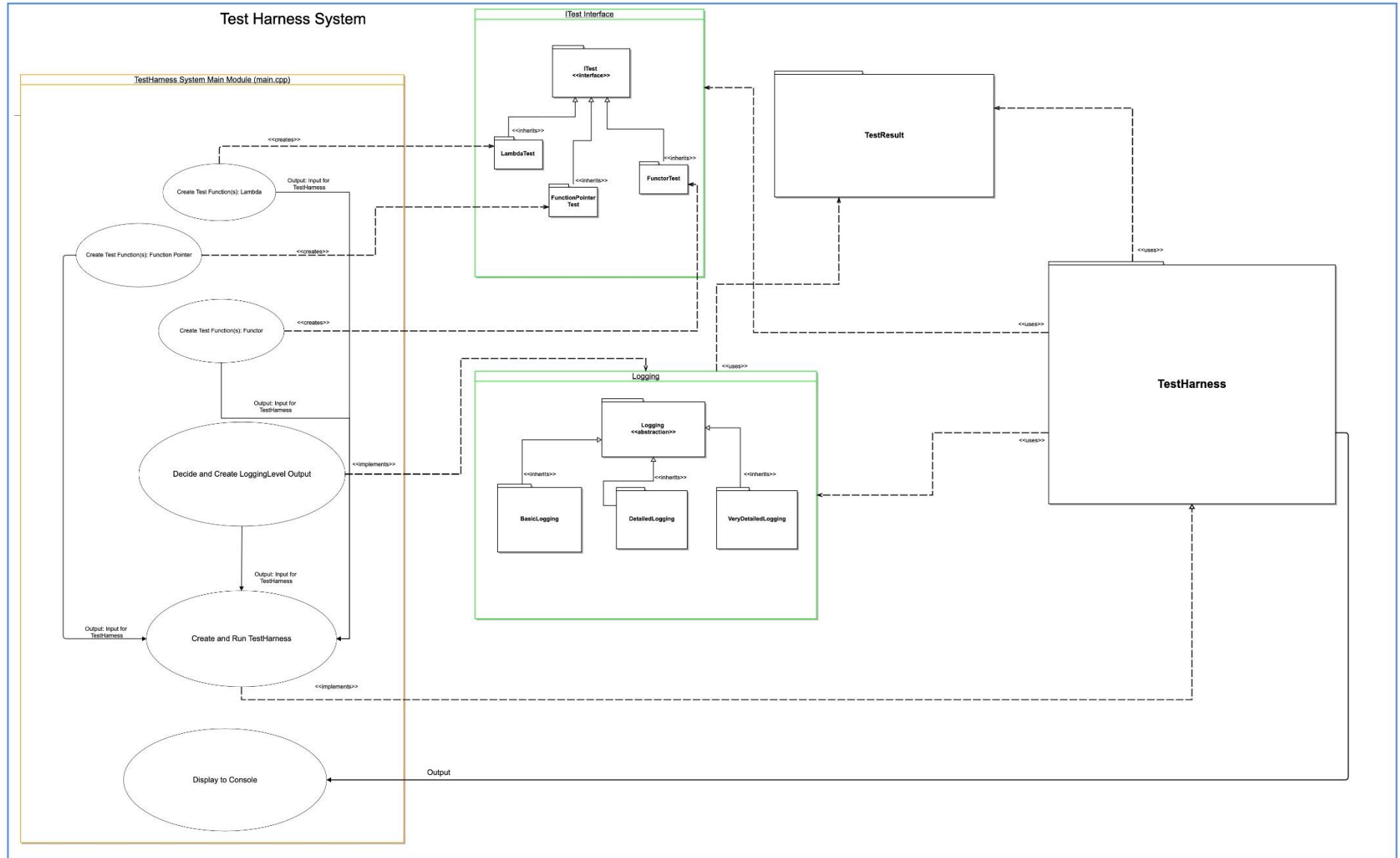


Architecture



Architecture Diagram Description

The architecture diagram provided, for Phase 2 of the Test Harness Project. This diagram will change as the project progress, but the overall goal of this diagram, is to produce a high-level diagram of what we think the final system will look like, once all phases are completed. Please see the information below on the purpose of each block of the diagram:

Test Harness System - This is the overall system that will be designed and created, for this project.

TestHarness - This is main part of the project, which consists of three different parts:

- Driver program (main.cpp) - This is the driver program that will execute the following:

1. Create different test functions using Lambdas, Functors, and Function Pointers.
2. Select or receive the appropriate logging level, based on user input.
3. Add all tests and logging level input into the TestHarness and run the TestHarness.

ITest - This is a C++ interface class, that also acts as the base class for the following classes:

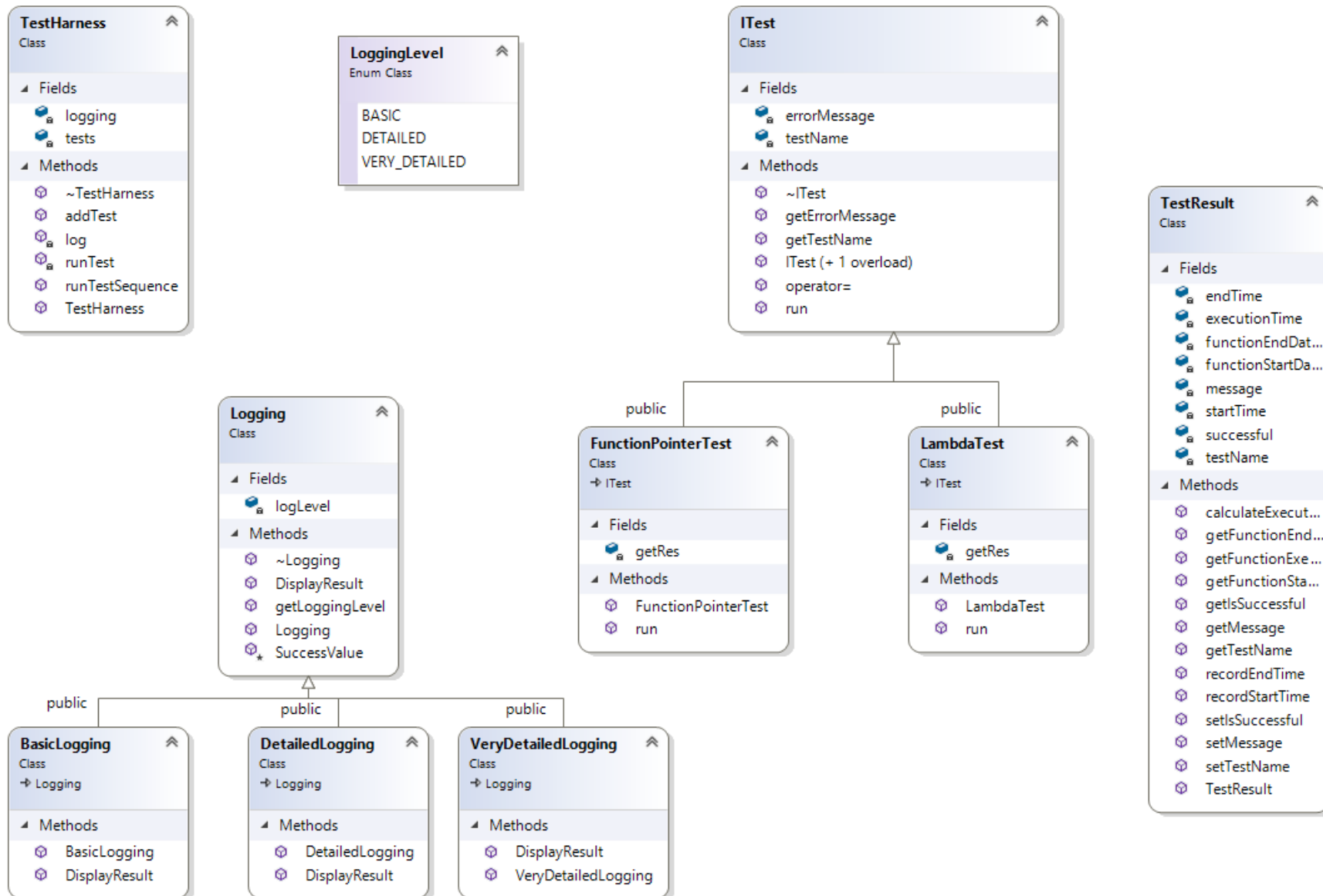
- LambdaTest
- FunctionPointerTest
- FunctorTest

- Logging - This abstract class will capture the results for the TestResult class and format the results of the tests, that will be run by printing to the console, based on user input. There are three levels of logging that will be supported by the system.

- BasicLogging (Level 1) - Pass or Fail Testing
- DetailedLogging (Level 2) - Pass or Fail Testing, Test-Specific Messages
- VeryDetailedLogging (Level 3) - Pass or Fail Testing, Test-Specific Messages, Time and Date Stamping, Values of Variables Tested

- TestHarnessClass - This will be the class that will execute the tests on the created functions and display the results to the console. This class uses the TestResult class, Logging classes and its child classes, the ITest interface (class) and its children classes, and receives information generated from the driver program.

Design



Teamwork

Everyone worked on Phase #2. Henry worked on re-designing the system block/architecture diagram to be in sync with the TestHarness classes, with feedback from Jack and Jonathan. The team worked together using Zoom and email, to collaborate on each portion of the Phase #2 development. The entire team was responsible for the overall review.