

Statement of Verification Activity

Team 2 - CS374 - Fall 2014

This document describes the various kinds of verification activities that our team will conduct at various stages in the development process. We will be using Cucumber to perform these tests (see the `cucumber/` directory in the project root for feature files). Tests will be broken up into two main classes: small unit tests and large project tests. In each case, the testing framework will be devised by examining both the structure and function of the module/program under consideration. Obviously more tests will need to be added and old tests refined as the development process begins.

Unit Tests

Unit tests are performed to test the functionality of a single module. This will often be a PHP file or class. However we will also need to test the database connection for basic Banner-like functionality. We will implement each unit test as a Cucumber feature in Gerkin. Driver programs will most likely have to be created to perform the test given a set of test data. As previously mentioned, test data can be gathered from both the module's structure and function.

SQL Queries - We will require unit tests to test database functionality. These will be written in PHP and test querying to all tables in our database at least once. Along with a driver PHP program to perform the client-side querying, we will create test tables in the database with data that we expect. This will allow us to see if each database query is functioning correctly.

Data handlers - In our design (see `design/docs/specs-beta.pdf`) we proposed a mechanism for implementing section-change computations in our Section Conflict Calculator application called a *data handler*. A data handler is an object that executes a scenario given data obtained from the database. A scenario is one particular kind of section change operation. We will require unit tests to test the operation of each handler variety that we implement. To input to each handler will have to be simulated as input from a human user that was processed into a form the handler understands. This is simply an array of key-value pairs where the string key values match expected data values for the data handler kind.

Web server operation - We will require unit tests to make sure that Apache is working properly. This involves manually using a Web browser or similar tool to pull down Web content. Also we will write Cucumber features that implement the client-side of the HTTP. We will employ a testing server and a production server. Testing in the deployment environment will be essential for verifying the successful operation of the project.

Global Tests

At some point, we must verify that all modules function when combined. Thus we propose tests that apply to the global functioning of our Section Conflict Calculator application. Though we may implement Cucumber features in Gerkin for this process, ultimately this will require manual testing by human users. We will need to explore the domain of inputs that the application can accept and consider edge cases in the more complex data-handling operations.