

Test Plan  
Client: Dr. Darren Lim, Assistant Professor



Proposed Project: Java Online Learning Toolkit (J.O.L.T.)

**Delivered by: 518 Interactive**

**Team Members:**

Lawrence Gregory  
Christopher Hughto  
Erik Stegmann  
Connor Vander Bogart  
Jedidiah Turnbull

**Revision:** 1.0

**Date:** 12/08/2009

## **Section 1: System Test**

### **1.1 Testing Plan Identifier**

This is the first of many revised versions that will be used to describe the testing requirements, and general test plan, in detail, for Java Online Learning Tutor (J.O.L.T). These requirements need to be tested and met for J.O.L.T. to be considered ready to be used by regular users. As the project continues to approach completion, this document will also be edited to reflect any changes, or new requirements that are discovered. These series of documents will provide a record of any changes, as well as whether a test is passing or failing at that point in time. An updated version will be provided at the end of the Detailed Design phase, and a finalized version will be provided at the end of the Acceptance Test phase.

### **1.2 Introduction**

A test plan gives detailed information regarding upcoming testing, including but not limited to, the scope of the testing, scheduling, acceptance criteria, as well as test results. The formats of test plans can vary with the product and organization to which they apply. There are two major elements of a test strategy that will be described in this test plan: Item Pass/Fail Criteria, and approach to testing.

Item Pass/Fail Criteria in this test plan shows which requirements will be tested during which stages of J.O.L.T's product life. The Item Pass/Fail Criteria is derived from the Requirements Specification document where each requirement or specification will ideally have multiple tests applied, in order to verify the results.

Approach to testing in this test plan describes how the testing of J.O.L.T. will be conducted. Test methods may be determined by standards, regulatory agencies, contractual agreement, or by some other means. This will be discussed at a later time.

This document is the overview of the test plan. The goal of this test plan is to review the various functions and features of J.O.L.T. and test to make sure that each module is functioning properly as well as that each module is functioning correctly in regard to the whole system.

This particular document is incomplete in the individual modules of the test are concerned. Future versions of this document will be complete.

### **1.3 Approach to Testing**

Our approach to testing will start with the individual unit tests and proceed outwards, by testing module interaction. Then the modules will be integrated into the whole system, and the system will be tested as a whole. The goal of this is to catch all errors at a low level of complexity, in order to prevent small errors from creating larger ones.

Throughout the testing process our team may have to correct any problems in the systems and proceed to re-test any modules that interact with the module where the correction was made. As our team cannot predict how many of these problems may occur, we cannot predict how many times tests will have to be done. Problems will be discussed with the client in an effort to find a solution that suits the client's needs.

### **1.4 Exception Handling To Test**

Under certain circumstances J.O.L.T. may encounter an error that could not be predicted to happen. J.O.L.T will try to handle these errors by providing information to the user. Examples of errors that could occur are:

*The database could not complete a request for one of various reasons:*  
A message will appear saying the request could not be completed and to try again. The user should try to submit their request again.

*The user tried to submit incorrect java code:*  
J.O.L.T. will try to find a reason why the code could not run correctly and provide feedback to the user on what went wrong and how to fix it.

### **1.5 Functional Requirements Inventory**

The following list outlines the required functionality that will be tested as a result of this test plan. All of the tests are designed to test whether one or more of the requirements have been met.

Java Online Learning Tutor will be a web-based application viewable on the major browsers. Browsers included will be Internet Explorer 8, Mozilla Firefox, Safari, and Google Chrome.

All references to *Source Code* imply Java™ Source Code, made to work with Java™ Version 1.6

The requirements are listed according to User Case, as follows:

### 1.5.1 Functional Requirements: Student:

- Will be able to register online with the system
  - Will receive email confirmation following registration
- Will be able to log into system.
  - A failed log in will display an appropriate error message.
  - A link to an identity validation page will be provided if password is forgotten.
- Will be able to enroll into courses they are currently taking
  - PIN number provided by instructor required to enroll into course on the system
- Will be able to view announcements sent to them.
- Will be able to view problem sets for each course they are in enrolled in
  - Will be able to view each individual problem within the problem set
  - Will be able to view hints and solutions to individual problems, if provided by problem creator.
- Will be able to submit solutions to individual problems within active problem sets in the form of Java™ source code
  - Code will be compiled by the system online
  - Student will receive immediate, automatic feedback on compile errors, if any
  - Student will receive immediate, automatic feedback on how their solution compares to the test cases
- Will be able to complete problem sets
  - Will be able to navigate to individual problems in a problem set without completing them in a specific order
  - Will be able to save any progress made for a problem or problem set
- Will be able to view grades for each class
- Will be able to view all previously submitted solutions
- Will be able to log out

### **1.5.2 Functional Requirements: Faculty:**

- Will be able to log into the system
- Will be able to create individual problems
- Will be able to create problem sets
  - Will be able to import previously created problems to a problem set
  - Will be able to import problems from the course pool to a problem set
  - Will be able to import problems from the global pool to a problem set
  - Will be able to individually create each problem for a problem set
- Will be able to assign problems they create to a category
- Will be able to assign a grading scheme to problem sets
  - Will be able to assign a point value to specific problems within the problem set
- Will be able to assign problem sets to the students in the courses they teach
  - Will be able to set activation date and time of problem set
  - Will be able to set expiration date and time of problem set
- Will be able to submit problems to a Course Pool
- Will be able to search a Course Pool for problems
- Will be able to search the Global Pool for problems
- Will be able to view a grade book for each of the courses
- Will be able to modify grades for all students in each of the courses they are currently teaching
- Will be able to post announcements to students in their courses
- Will be able to view announcements sent to them.
- Will be able to interact with JOLT as a student user
- Will be able to log out

### **1.5.3 Functional Requirements: Course Coordinator:**

- Will be able to log into the system
- Will be able to create faculty accounts
- Will be able to assign faculty to a course
- Will be able to create content for courses they are in charge of
- Will have access to course tools which will provide statistics on problems and grades for a course
  - Will be able to create reports over multiple sections of a course involving all problems and problem sets or any subset thereof.
  - Reports may include general statistics such as number of participants, average score, median, low score, and high score.
- Will be able to manage the course pool for each course they are in charge of
  - Will be able to add, modify, or delete any problem in their course pool
  - Will be able to submit problems to the global pool
- Will be able to modify grades for all students currently enrolled in a course they currently manage
  - Will be able to keep track of all grades and any adjustments that are made
- Will be able to create announcement for all faculty and students of courses they manage or any subset thereof
- Will be able to log out

#### **1.5.4 Functional Requirements: Administrator:**

- Will be able log in
- Will be able to manage all accounts on the system
- Will be able to create course coordinator and faculty accounts
  - Will be able to assign courses to course coordinators
- Will have the same abilities as a course coordinator
- Will be able to manage the global pool of problems
- Will have access to tools for management of all accounts
  - Will be able to modify all account information for any user
  - Will be able to delete accounts
  - Will be able reset locked accounts
- Will be able to create announcements for all course coordinators, faculty and students, or any subset thereof
- Will be able to log out

#### **1.5.5 Java SDK:**

- Will accept and attempt to compile all Java™ source code submitted by students
  - Will output compile errors, when applicable
  - Will create Java™ Byte Code upon a successful compilation
- Will execute all successfully compiled Java™ solutions
  - Will monitor students' submissions while they are running for runtime errors
  - Will kill a student's submission if it takes too long to run (Timeout)

Will record the output generated from the students' submissions

