

Test Results  
Client: Dr. Darren Lim, Assistant Professor



Project: Java Online Learning Tool (*J.O.L.T*)

**Delivered by: 518 Interactive**

**Team Members:**

Erik Stegmann  
Lawrence Gregory  
Christopher Hughto  
Connor Vander Bogart  
Jedidiah Turnbull

**Revision: 1.0**

**Date: 4/26/2010**

## Contents

Section 1 Introduction.....	3
Section 2 Functional Requirements Inventory.....	4
2.1 Functional Requirements Inventory Introduction.....	4
2.2 Functional Requirements Inventory.....	5
2.2.1 Functional Requirements: Students: .....	5
2.2.2 Functional Requirements: Faculty: .....	6
2.2.3 Functional Requirements: Course Coordinator: .....	7
2.2.4 Functional Requirements: Administrator:.....	8
2.2.5 Java SDK: .....	8
Section 3 Approach to Testing.....	9
3.1 Approach to Testing Introduction.....	9
3.2 Exception Handling To Test .....	9
3.3 Item Pass/Fail Criteria.....	9
3.4 Staffing.....	21
3.5 Responsibilities .....	21
Section 4 Types of Testing .....	22
4.1 Test Results.....	22
4.2 System Testing.....	22
4.3 Regression Testing.....	22
4.4 Integration Testing.....	22
4.5 Unit Testing .....	22
Section 5 Deliverables and Approval .....	23

## Section 1 Introduction

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

Item Pass/Fail Criteria in the test results shows which requirements have been 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 has multiple tests applied, in order to verify the results.

Approach to testing describes how the testing of *J.O.L.T* was conducted. Test methods have been determined by standards, regulatory agencies, contractual agreement.

This document is the overview of the test results. The goal of this document 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.

## Section 2 Functional Requirements Inventory

### *2.1 Functional Requirements Inventory Introduction*

The following list outlines the required functionality that was tested. All of the tests are designed to test whether one or more of the requirements have been met.

Java Online Learning Tool is a web-based application viewable on the major browsers. Browsers included are Internet Explorer 8, Mozilla Firefox 3.6, Opera, Safari, and Google Chrome.

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

## ***2.2 Functional Requirements Inventory***

### **2.2.1 Functional Requirements: Students:**

- Are able to register online with the system. **Met**
  - Receive email confirmation following registration. **Met**
- Are able to log into system. **Partially Met**
  - A failed log in displays an appropriate error message. **Met**
  - A link to an identity validation page is provided if password is forgotten. **Not Met**
  - 3 Failed login attempts leads to system lockout. **Met**
- Are able to enroll into courses they are currently taking. **Met**
  - PIN number provided by instructor required to enroll into course on the system. **Met**
- Are able to view announcements sent to them. **Met**
  - Are able to delete their announcements. **Met**
- Are able to view problem sets for each course they are in enrolled in. **Met**
  - Are able to view each individual problem within the problem set. **Met**
  - Are able to view hints and solutions to individual problems, if provided by problem creator. **Met**
- Are able to submit solutions to individual problems within active problem sets in the form of Java™ source code. **Met**
  - Code is compiled by the system online. **Met**
  - Student receives immediate, automatic feedback on compile errors, if any. **Met**
  - Student receives immediate, automatic feedback on how their solution compares to the test cases. **Met**
- Are able to complete problem sets. **Met**
  - Are able to navigate to individual problems in a problem set without completing them in a specific order. **Met**
  - Are able to save any progress made for a problem or problem set. **Met**
- Are able to view grades for each assignment in each class they are enrolled in. **Met**
- Are able to view all previously submitted solutions. **Met**
  - Have access to their solutions and grades for all prior classes they were enrolled in. **Met**
- Are able to log out. **Met**

### 2.2.2 Functional Requirements: Faculty:

- Are able to log into system. **Partially Met**
  - A failed log in displays an appropriate error message. **Met**
  - A link to an identity validation page is provided if password is forgotten. **Not Met**
  - 3 Failed login attempts leads to system lockout. **Met**
- Are able to create individual problems. **Met**
  - Individual problems that are partially completed by faculty are saved to a sandbox area, until they are complete. **Met**
    - Once complete, problem gets transferred to private pool. **Met**
- Are able to create problem sets. **Met**
  - Are able to import previously created problems to a problem set. **Met**
  - Are able to import problems from the course pool to a problem set. **Met**
  - Are able to import problems from the global pool to a problem set. **Met**
  - Are able to individually create each problem for a problem set. **Met**
- Are able to assign problems they create to a category. **Met**
- Are able to assign a grading scheme to problem sets. **Met**
  - Are able to assign a point value to specific problems within the problem set. **Met**
- Are able to assign problem sets to the sections they teach. **Met**
  - Are able to set activation date and time of problem set. **Met**
- Are able to submit problems to a Course Pool. **Met**
- Are able to search a Course Pool for problems. **Partially Met** (No search implemented, browse through list is available)
- Are able to search the Global Pool for problems. **Partially Met** (No search implemented, browse through list is available)
- Are able to view a grade book for each of the courses. **Met**
- Are able to modify grades for all students in each of the courses they are currently teaching. **Partially Met** (Grade-by-grade manipulation only – curving of an assignment automatically has not been implemented)
- Are able to post announcements to students in their courses. **Met**
- Are able to view announcements sent to them. **Met**
- Are able to remove students from their sections. **Met**
- Are able to log out. **Met**

### 2.2.3 Functional Requirements: Course Coordinator:

- Are able to log into system. **Partially Met**
  - A failed log in displays an appropriate error message. **Met**
  - A link to an identity validation page is provided if password is forgotten. **Not Met**
  - 3 Failed login attempts leads to system lockout. **Met**
- Are able to create faculty accounts. **Met**
- Are able to assign faculty to a section. **Met**
- Are able to create problems and problem sets for courses they are in charge of. **Met**
  - Individual problems that are partially completed by Course Coordinators are saved to a sandbox area, until they are complete. **Met**
    - Once complete, problem gets transferred to private pool. **Met**
- Have access to course tools which will provide statistics on problems and grades for a course **Not Met**
  - Are able to create reports over multiple sections of a course involving all problems and problem sets or any subset thereof. **Not Met**
  - Reports may include general statistics such as number of participants, average score, median, low score, and high score. **Not Met**
- Are able to manage the course pool for each course they are in charge of. **Met**
  - Are able to add, modify, or delete any problem in their course pool. **Met**
  - Are able to submit problems to the global pool. **Met**
- Are able to modify grades for all students currently enrolled in a course they currently manage **Partially Met** (**Grade-by-grade manipulation only – curving of an assignment automatically has not been implemented**)
  - Are able to keep track of all grades and any adjustments that are made **Not Met**
- Are able to create announcement for all faculty and students of courses they manage or any subset thereof. **Met**
- Are able to log out. **Met**

### 2.2.4 Functional Requirements: Administrator:

- Are able to log into system. **Partially Met**
  - A failed log in will display an appropriate error message. **Met**
  - A link to an identity validation page is provided if password is forgotten. **Not Met**
- Are able to manage all accounts on the system. **Met**
- Are able to create course coordinator and faculty accounts. **Met**
  - Are able to assign courses to course coordinators. **Met**
- Will have the same abilities as a course coordinator **Partially Met** (See Section 2.2.3 “Functional Requirements: Course Coordinator” for details)
- Are able to manage the global pool of problems. **Met**
- Will have access to tools for management of all accounts. **Met**
  - Are able to modify all account information for any user. **Met**
  - Are able to delete accounts. **Met**
  - Are able reset locked accounts. **Met**
- Are able to create announcements for all course coordinators, faculty and students, or any subset thereof. **Met**
- Are able to log out. **Met**

### 2.2.5 Java SDK:

- Accepts and attempts to compile all Java™ source code submitted by students. **Met**
  - Outputs compile errors, when applicable. **Met**
  - Creates Java™ Byte Code upon a successful compilation. **Met**
- Executes all successfully compiled Java™ solutions. **Met**
  - Monitors students’ submissions while they are running for runtime errors. **Met**
  - Kills a student’s submission if it takes too long to run (Timeout). **Met**
- Records the output generated from the students’ submissions. **Met**



## Section 3 Approach to Testing

### *3.1 Approach to Testing Introduction*

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

### *3.2 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 appears 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* tries 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.

### *3.3 Item Pass/Fail Criteria*

The following is a checklist to ensure that individual parts of the system are working properly. This outlines the major requirements that are being tested based on individual modules' unit tests. These are arranged by user type as they appear in the functional requirements inventory.

### *3.3.1 Student*

-The student user can register with the system

Yes

-Once completing the registration form the Student receives an email message with a link to activate their account

Yes

-The Student can Log in to the System Once Registered

Yes

-The system displays correct error messages when a log in attempt fails

Yes

-A link to an identity validation page is if password is forgotten.

No

-3 failed login attempts with a given username locks that specific username out of the system

Yes

-Once Logged in the student can enroll in a course section

Yes

-Students can view announcements sent to them

Yes

-Students can delete announcements sent to them

Yes

-Students can view problem sets assigned to them

Yes

-Students can view each individual problem within the problem set

Yes

-Students can view solutions to individual problems, if provided by problem creator.

Yes

-Students can submit solutions to individual problems within active problem sets in the form of Java™ source code

Yes

-Student receives immediate, automatic feedback on compile errors, if any

Yes

-Student receives immediate, automatic feedback on how their solution compares to the test cases

Yes

-Student can complete problem sets

Yes

-Student can navigate to individual problems in a problem set without completing them in a specific order

Yes

-Any incomplete progress made for a problem or problem set is saved for the student

Yes

-Student can view grades for each assignment in each class they are enrolled in.

Yes

- Students can view all previously submitted solutions

Yes

-Student has access to their solutions and grades for all prior classes they were enrolled in.

Yes

-Student can log out

Yes

### 3.3.2 Faculty

-Can log into system.

Yes

-A failed log in displays an appropriate error message.

Yes

-A link to an identity validation page is provided if password is forgotten.

No

-3 failed login attempts locks the faculty member out of the system.

Yes

-Faculty user can create individual problems

Yes

-Individual problems that are partially completed by faculty are saved to a sandbox area, until they are complete.

Yes

-Once complete, problem gets transferred to private pool.

Yes

-Faculty user can create problem sets

Yes

-Faculty user can import previously created problems to a problem set

Yes

-Faculty user can import problems from the course pool to a problem set

Yes

-Faculty user can import problems from the global pool to a problem set

Yes

-Faculty user can individually create each problem for a problem set

Yes

-Faculty user can assign problems they create to a category

Yes

-Faculty user can assign a grading scheme to problem sets

Yes

-Faculty user can assign a point value to specific problems within the problem set

Yes

-Faculty user can assign problem sets to the sections they teach

Yes

-Faculty user can set activation date and time of problem set

Yes

-Faculty user can set expiration date and time of problem set

Yes

-Faculty user can submit problems to a Course Pool

Yes

-Faculty user can search a Course Pool for problems

No

-Faculty user can search the Global Pool for problems

No

- Faculty user can view a grade book for each of the courses

Yes

-Faculty user can modify grades for all students in each of the courses they are currently teaching

Yes

-Faculty user can post announcements to students in their courses

Yes

-Faculty user can view announcements sent to them.

Yes

-Faculty user can log out

Yes

### 3.3.3 *Course Coordinator*

-Course Coordinator can log into system.

Yes

-A failed log in displays an appropriate error message.

Yes

-A link to an identity validation page is provided if password is forgotten.

No

-3 failed login attempts will lead to system lockout.

Yes

-Course Coordinator can create faculty accounts

Yes

-Course Coordinator can assign faculty to a section

Yes

-Course Coordinator can create problems and problem sets for courses they are in charge of

Yes

-Individual problems that are partially completed by Course Coordinators are saved to a sandbox area, until they are complete.

Yes

-Once complete, problem gets transferred to private pool.

Yes

-Course Coordinator can access course tools which will provide statistics on problems and grades for a course

No



-Course Coordinator can create reports over multiple sections of a course involving all problems and problem sets or any subset thereof.

No

-Course Coordinator can manage the course pool for each course they are in charge of

Yes

-Course Coordinator can add, modify, or delete any problem in their course pool

Yes

-Course Coordinator can submit problems to the global pool

Yes

-Course Coordinator can modify grades for all students currently enrolled in a course they currently manage

No

-Course Coordinator can keep track of all grades and any adjustments that are made

No

-Course Coordinator can create announcements for all faculty and students of courses they manage or any subset thereof

Yes

-Course Coordinator can log out

Yes

### 3.3.4 Administrator

-The Administrator can log into system.

Yes

-A failed log in displays an appropriate error message.

Yes

-A link to an identity validation page is provided if password is forgotten.

No

-The Administrator can manage all accounts on the system

Yes

-The Administrator can create course coordinator and faculty accounts

Yes

-The Administrator can assign courses to course coordinators

Yes

-The Administrator has all the same abilities as a course coordinator

Yes [those that pass for course coordinator only]

-The Administrator can manage the global pool of problems

Yes

-Has access to tools for management of all accounts

Yes

-The Administrator can modify all account information for any user

Yes

-The Administrator can disable accounts

Yes

-The Administrator can reset locked accounts

Yes

-The Administrator can create announcements for all course coordinators, faculty and students, or any subset thereof

Yes

-The Administrator can log out

Yes

### *3.3.5 Java SDK*

-Accepts and attempts to compile all Java™ source code submitted by students

Yes

-Outputs compile errors, when applicable

Yes

-Creates Java™ Byte Code upon a successful compilation

Yes

-Executes all successfully compiled Java™ solutions

Yes

-Monitors students' submissions while they are running for runtime errors

Yes

-"Kills" a student's submission if it takes too long to run (Timeout)

Yes

-Records the output generated from the students' submissions

Yes

### *3.4 Staffing*

Members of the 518 Interactive Development team were responsible for the testing of *J.O.L.T*, in order to ensure that all of the functional requirements were met.

### *3.5 Responsibilities*

The responsibilities of those who are testing *J.O.L.T* are to execute the tests accurately in order to ensure that all of the functional requirements are met.

## Section 4 Types of Testing

### ***4.1 Test Results***

The Test Results document is the final test that 518 Interactive will perform on *J.O.L.T* before it is delivered to the client, Dr. Lim. The Acceptance Test is a type of System Test which validates that all Functional Requirements have been met.

### ***4.2 System Testing***

The System Test is the act of testing the entire (complete, integrated) system. It is used to evaluate whether or not the System meets all functional requirements. The System Test for *J.O.L.T* is performed after all Unit and Integration Testing has been completed.

### ***4.3 Regression Testing***

Regression Testing is used to make sure that the system still works as expected after changes to a unit, or related unit are made. If one unit is changed, and this unit which works with two other units to accomplish a task, then all three units must be tested to ensure that nothing was broken in the process.

### ***4.4 Integration Testing***

Integration Testing is the process of testing multiple units at once to determine if they function together correctly. This is performed after all Unit Test are performed. The Test Results for *J.O.L.T* includes a listing for each unit which shows the other units that it is associated with.

### ***4.5 Unit Testing***

Unit Testing is the process of testing an individual component of the system to ensure that it functions as expected. 518 Interactive has provided a Unit Test for each unit in *J.O.L.T*, which is enclosed.

## Section 5 Deliverables and Approval

### 5.1 Deliverables

The following documents shall be delivered to the client as part of the approval process:

- Unit Test Catalog
- Results of all unit tests (completed, and documented)
- Results of Pass/Fail Criteria
- Results of Acceptance Test

### 5.2 Approval

*J.O.L.T* shall be approved as complete and ready for delivery once it is shown that all functional requirements have been met. This is shown through the Acceptance Test, which tests all individual units of the system, as well as the complete, integrated system. This approval would be granted by Dr. Lim, the client of 518 Interactive.