Acceptance Test
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
  Connor Vander Bogart
  Jedidiah Turnbull
  Christopher Hughto

Revision: 1.0
Date: 4/26/2010
Contents

1 Introduction .......................................................................................................................... 1
  1.1 Purpose .......................................................................................................................... 1
  1.2 Scope ............................................................................................................................. 2
  1.3 Audience ......................................................................................................................... 2
  1.4 Product Overview and Summary .................................................................................... 2

2 Requirements Specification .................................................................................................. 3
  2.1 Introduction ....................................................................................................................... 3
  2.2 Functional Requirements Inventory ................................................................................ 4
    2.2.1 Functional Requirements: Student: ...................................................................... 4
    2.2.2 Functional Requirements: Faculty: ........................................................................ 5
    2.2.3 Functional Requirements: Course Coordinator: ..................................................... 6
    2.2.4 Functional Requirements: Administrator: ............................................................... 7
    2.2.5 Java SDK: ................................................................................................................. 7
    2.2.6 Non-Functional Requirements .................................................................................. 8
    2.2.7 Future Enhancements ............................................................................................... 8

3 Architectural Design Specification ......................................................................................... 9
  3.1 Development, Operating, and Maintenance Environments ............................................. 9
  3.2 Production Environment ................................................................................................. 11
  3.3 UML Deployment Diagram ............................................................................................. 12
  3.4 Web Site Map .................................................................................................................. 13
    3.4.1 Context Web Site Map ........................................................................................... 14
    3.4.2 Student Web Site Map ............................................................................................ 15
    3.4.3 Faculty Web Site Map ............................................................................................. 16
    3.4.4 Course Coordinator Web Site Map ........................................................................... 17
    3.4.5 Administrator Web Site Map .................................................................................... 18

4 External & Internal Design Specification ............................................................................. 19
  4.1 User Screens with respective User Command Summaries .............................................. 19
  4.2 Student User ..................................................................................................................... 20
    4.2.1 Student User Narrative ............................................................................................ 20
    4.2.2 J.O.L.T Login Page .................................................................................................. 21
    4.2.3 Student Registration Page ....................................................................................... 22
    4.2.4 Student Registration Page (Error Message – Invalid Fields) ................................... 23
    4.2.5 Student Registration Page (Error Message – Invalid Email) ................................... 24
    4.2.6 Student Registration Page (Completed) .................................................................... 25
    4.2.7 Invalid Login Screen ............................................................................................... 26
    4.2.8 Invalid Login Screen ............................................................................................... 27
    4.2.9 Account Lockout Screen ........................................................................................... 28
    4.2.10 Common Header ..................................................................................................... 29
    4.2.11 Student Home Page ............................................................................................... 29
    4.2.12 Course Enrollment (Invalid PIN) .......................................................................... 30
    4.2.13 Course Enrollment (Completed) ............................................................................. 31
    4.2.14 Section Home Page ................................................................................................. 32
    4.2.15 Student Section Home (Populated) ....................................................................... 33
    4.2.16 Problem Set Overview ............................................................................................ 34
    4.2.17 Solve Problem ......................................................................................................... 35
    4.2.18 Student Solution Attempt (Compile Error) .............................................................. 36
    4.2.19 Student Solution Attempt (Incorrect Output) ......................................................... 37
    4.2.20 Student Solution Attempt (Timeout) ....................................................................... 38
    4.2.21 Updated Problem Set Overview Screen (Navigate Away) ...................................... 39
    4.2.22 Student Solution Attempt (Completed) ................................................................... 40
    4.2.23 Updated Problem Set Screen .................................................................................. 41
    4.2.24 View Gradebook .................................................................................................... 42
### Faculty User

4.3 Faculty User Case Narrative ................................................................. 47
4.3.2 Faculty Screens .............................................................................. 48
4.3.3 J.O.L.T Login .................................................................................. 48
4.3.4 Faculty Home Page .......................................................................... 49
4.3.5 Section Home Page .......................................................................... 50
4.3.6 Create Problem ................................................................................ 51
4.3.7 Create Problem (Invalid Field) ......................................................... 52
4.3.8 Create Problem Compiler Error ......................................................... 53
4.3.9 Faculty Sandbox (Unpopulated) ......................................................... 54
4.3.10 Faculty Sandbox (Populated) ........................................................... 54
4.3.11 Edit Problem ................................................................................... 55
4.3.12 Create Problem Set .......................................................................... 56
4.3.13 Create Problem Set (Error) ............................................................... 57
4.3.14 Create Problem Set (No Problems Added Error) ............................. 58
4.3.15 Create Problem Set (Completed) ...................................................... 59
4.3.16 Edit Problem Set (Completed) .......................................................... 60
4.3.17 Activate/Assign Problem Set ............................................................ 61
4.3.18 Create Announcement ...................................................................... 62
4.3.19 Create Announcement (Add Recipients) .......................................... 63
4.3.20 Create Announcement (Missing Field) ............................................. 64
4.3.21 Create Announcement (Complete) .................................................... 65
4.3.22 View Gradebook ............................................................................ 66
4.3.23 Edit Gradebook (Editing) ................................................................. 67
4.3.24 Manage Pools ................................................................................ 68
4.3.25 View Profile ................................................................................... 69
4.3.26 Edit Profile ..................................................................................... 70
4.3.27 Logout ............................................................................................ 71

### Course Coordinator User

4.4 Course Coordinator User Case Narrative ............................................ 72
4.4.1 Course Coordinator User Screens ..................................................... 73
4.4.2 J.O.L.T Login ................................................................................... 73
4.4.3 Course Coordinator Home Page ........................................................ 74
4.4.4 Create Faculty Account ................................................................... 75
4.4.5 Edit Faculty Account ....................................................................... 76
4.4.6 Create Course Section .................................................................... 77
4.4.7 Create Course Section (Error) .......................................................... 78
4.4.8 Edit Course Section .......................................................................... 79
4.4.1 View Gradebook ............................................................................. 80
4.4.2 Edit Gradebook (Editing) ................................................................. 81
4.4.3 Create Problem ................................................................................ 82
4.4.4 Create Problem (Invalid Field) .......................................................... 83
4.4.5 Create Problem Compiler Error ......................................................... 84
4.4.6 Create Coordinator Sandbox (Unpopulated) .................................... 85
4.4.7 Create Coordinator Sandbox (Populated) ........................................ 85
4.4.8 Edit Problem ................................................................................... 86
4.4.9 Create Problem Set ......................................................................... 87
4.4.10 Create Problem Set (Error) .............................................................. 88
4.4.11 Create Problem Set (No Problems Added Error) ............................ 89
4.4.12 Create Problem Set (Completed) ...................................................... 90
4.4.13 Edit Problem Set (Completed) ......................................................... 91
4.4.9 Activate/Assign Problem Set ................................................. 92
4.4.10 Create Announcement .......................................................... 93
4.4.11 Create Announcement (Add Recipients) ................................... 94
4.4.12 Create Announcement (Missing Field) ..................................... 95
4.4.13 Create Announcement (Complete) .......................................... 96
4.4.14 Manage Pools ........................................................................ 97
4.4.15 Accept Private Problems to Course Pool .................................... 98
4.4.16 Create Announcement ............................................................ 99
4.4.17 Create Announcement (Add Recipients) .................................... 100
4.4.18 Create Announcement (Missing Field) ...................................... 101
4.4.19 Create Announcement (Complete) .......................................... 102
4.4.1 View Profile ........................................................................... 103
4.4.2 Edit Profile ............................................................................. 104
4.4.3 Logout ................................................................................... 105
4.5 Administrator User .................................................................... 106
  4.5.1 Administrator User Case Narrative ......................................... 106
  4.5.2 Administrator User Screens .................................................... 106
  4.5.3 J.O.L.T Login ......................................................................... 107
  4.5.4 Administrator Home Page ....................................................... 108
  4.5.5 Manage Users ........................................................................ 109
  4.5.6 Create Course Coordinator ..................................................... 110
  4.5.7 Create Faculty Account .......................................................... 111
  4.5.8 Create Student Account .......................................................... 112
  4.5.9 Edit User Account .................................................................. 113
  4.5.10 Course Management .............................................................. 114
  4.5.11 Create Course ....................................................................... 115
  4.5.12 View Gradebook ................................................................... 116
  4.5.13 Edit Gradebook (Editing) ....................................................... 117
  4.5.14 Create Problem ..................................................................... 118
  4.5.15 Create Problem (Invalid Field) .............................................. 119
  4.5.16 Create Problem Compiler Error ............................................ 120
  4.5.17 Administrator Sandbox (Unpopulated) .................................... 121
  4.5.18 Administrator Sandbox (Populated) ....................................... 121
  4.5.19 Edit Problem ......................................................................... 122
  4.5.20 Create Problem Set ............................................................... 123
  4.5.21 Create Problem Set (Error) .................................................... 124
  4.5.22 Create Problem Set (No Problems Added Error) ...................... 125
  4.5.23 Create Problem Set (Completed) ............................................ 126
  4.5.24 Edit Problem Set (Completed) ............................................... 127
  4.5.25 Activate/Assign Problem Set ................................................ 128
  4.5.26 Create Announcement .......................................................... 129
  4.5.27 Create Announcement (Add Recipients) ................................. 130
  4.5.28 Create Announcement (Missing Field) .................................... 131
  4.5.29 Create Announcement (Complete) ......................................... 132
  4.5.30 Manage Pools ....................................................................... 133
  4.5.1 Accept Private/Course Problems to Global Pool ...................... 134
  4.5.2 View Profile .......................................................................... 135
  4.5.3 Edit Profile ............................................................................ 136
  4.5.4 Logout .................................................................................. 137
4.6 Logical Data Stores .................................................................. 138
  4.6.1 Announcement Table .............................................................. 138
  4.6.2 Assignment Table ................................................................. 139
  4.6.3 Course Table .......................................................................... 140
  4.6.4 Enrolled Students Table ......................................................... 141
  4.6.5 Grade Log Table .................................................................... 142
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6.6</td>
<td>Pool Table</td>
<td>143</td>
</tr>
<tr>
<td>4.6.7</td>
<td>Problem Table</td>
<td>144</td>
</tr>
<tr>
<td>4.6.8</td>
<td>Problem Location Table</td>
<td>145</td>
</tr>
<tr>
<td>4.6.9</td>
<td>Problem Set Table</td>
<td>146</td>
</tr>
<tr>
<td>4.6.10</td>
<td>Section Table</td>
<td>148</td>
</tr>
<tr>
<td>4.6.11</td>
<td>Set Location Table</td>
<td>148</td>
</tr>
<tr>
<td>4.6.12</td>
<td>Solution Attempt Table</td>
<td>149</td>
</tr>
<tr>
<td>4.6.13</td>
<td>Test Case Table</td>
<td>149</td>
</tr>
<tr>
<td>4.6.14</td>
<td>User Table</td>
<td>148</td>
</tr>
<tr>
<td>4.6.15</td>
<td>Graphical Representation Key</td>
<td>150</td>
</tr>
<tr>
<td>4.6.16</td>
<td>Graphical Representation (Entity Relationship Diagram)</td>
<td>151</td>
</tr>
<tr>
<td>4.7</td>
<td>Source Code</td>
<td>153</td>
</tr>
<tr>
<td>4.8</td>
<td>Deliverables</td>
<td>153</td>
</tr>
<tr>
<td>5</td>
<td>Appendices</td>
<td>152</td>
</tr>
<tr>
<td>5.1</td>
<td>Appendix 1: Sources of Information</td>
<td>152</td>
</tr>
<tr>
<td>5.2</td>
<td>Appendix 2: Glossary of Terms</td>
<td>153</td>
</tr>
<tr>
<td>5.3</td>
<td>Appendix 3: Current Timeline</td>
<td>155</td>
</tr>
<tr>
<td>5.4</td>
<td>Data Flow Diagram</td>
<td>156</td>
</tr>
<tr>
<td>5.5</td>
<td>Activity Diagram</td>
<td>156</td>
</tr>
<tr>
<td>5.6</td>
<td>Data Dictionary</td>
<td>156</td>
</tr>
<tr>
<td>5.7</td>
<td>Test Results</td>
<td>157</td>
</tr>
<tr>
<td>5.8</td>
<td>Source Code</td>
<td>157</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Purpose
This document shows the final user interface, requirements, source code, and test results for the Java Online Learning Tool (J.O.L.T) for the client, Dr. Darren Lim, Assistant Professor of the Computer Science Department of Siena College. The software requirements, user interface, and design specifications have been previously defined in the previous documents. It should be noted, however, that the Detailed Design document contains the most current information.

This document serves these main purposes:

- To provide a finalized interface design for all users of J.O.L.T
- To provide a finalized version of the functional requirements.
- To provide a complete listing of source code for J.O.L.T
- To provide the results of testing.

There are several supplementary documents to this design document, which are available under the “Documents” menu of the 518 Interactive Website. These documents include:

- Data Flow Diagram Document: Provides a graphical representation of how data is transmitted and manipulated within J.O.L.T
- Activity Diagram Document: Provides a graphical representation of how specific tasks are accomplished within J.O.L.T
- Data Dictionary: Provides a repository of all the data entities that J.O.L.T utilizes.
- Source Code Document: Provides all of the source code in J.O.L.T arranged alphabetically by file name.
- Test Results Document: The Test Results Document outlines the testing that has been performed.

NOTE: All references to Source Code imply Java™ Source Code, made to work with Java™ Version 1.6
1.2 Scope
The requirements, screen designs, source code, and test results in this document are final. This Acceptance Test document supersedes any prior documents as the descriptor of J.O.L.T’s requirements, interface design, and test plan/results.

1.3 Audience
This document is intended for 518 Interactive, Dr. Darren T. Lim, members of the Spring 2010 Software Engineering II class, and the other clients, Dr. Timoth C.Lederman and Dr. Eric A. Breimer.

1.4 Product Overview and Summary
The Java Online Learning Tool (J.O.L.T) is a comprehensive web application designed to enhance the experience of learning the Java programming language. J.O.L.T allows for students enrolled in programming courses to solve programming problems, which are created and entered by the Computer Science faculty at Siena College. J.O.L.T provides a personal gradebook for all students, as well as a course gradebook for Computer Science faculty members which is used to track progress.
2 Requirements Specification

2.1 Introduction
The following list outlines the required functionality included in the final solution.

Java Online Learning Tool is be a web-based application viewable on the major browsers. Browsers included will be Internet Explorer 8, Mozilla Firefox 3.6, Safari, Opera 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 Type, as follows:
2.2 Functional Requirements Inventory

The Functional Requirements Inventory contains additional information on which functional requirements have been met, unmet, or partially met. If a requirement is partially met, a description of how the requirement is partially met will be provided. Additionally, if a requirement has a set of sub-requirements that it needs to meet, in order for the whole requirement to be met and not all of the sub requirements are met, that particular requirement will be considered partially met.

2.2.1 Functional Requirements: Student:

- 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**
2.2.6 Non-Functional Requirements

A non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are typically used to describe the qualities of a system. Given this definition, there is no concrete way to measure whether or not a non-functional requirement has been met.

Non-Functional requirements have not been formally defined for J.O.L.T.

2.2.7 Future Enhancements

The following is a list of features that would enhance the functionality and overall usability of J.O.L.T. The features may be previous requirements that have not been met or new features that the members of 518 Interactive believe should be implemented in order to enhance the system.

- Detecting the use of recursion upon run time in order to allow recursion specific problems.
- Implement search of users, problems, problem sets, and assignments within the system
- Generate reports on multiple sections of a course involving all problems and problem sets or any subset thereof.
- Multi-select and drag and drop pool interface for the problem set screens and pool management screens.
- “Views” to change user type quickly
- “Linkable” accounts
3 Architectural Design Specification

3.1 Development, Operating, and Maintenance Environments

*J.O.L.T* was developed in Siena College’s Software Engineering Lab, located in Roger Bacon, third floor. The members of 518 Interactive used the following resources:

**Windows Machine:**

- **Operating System:** Microsoft Windows Vista Enterprise
  - Build: 6002
  - Revision: 18005
  - Service Pack 2
- **Processor:** Intel® Core™2 Duo CPU
  - Model: E7500
  - Speed: 2.93 GHz
- **Memory (RAM):** 4.00 GB
- **System Type:** 32-bit
- **Dual Monitor Setup**
- **Software Installed:**
  - Microsoft Office 2007 (Including Microsoft Project)
  - Macromedia Dreamweaver, Fireworks, Flash, Freehand, Studio (2004 Versions)
  - Internet Explorer, Mozilla Firefox, Google Chrome

**Macintosh Machine:**

- **Operating System:** Apple Mac OS X
  - Version 10.4.11
  - Model: iMac5
- **Processor:** Intel Core2 Duo
  - Speed: 2 GHz
- **Memory (RAM):** 1.00 GB
- **Dual Monitor Setup**
- **Software Installed:**
  - Microsoft Office 2004 for Mac
  - Macromedia Dreamweaver, Fireworks, Flash, Freehand, Studio (2004 Versions)
  - Safari, Mozilla Firefox
J.O.L.T was implemented, and designed to run on the following specifications:

- **Operating System**: CentOS (Linux) Release 5.2 (Final)
- **Server Name**: oraserv.cs.siena.edu
- **CPU Type**: x86_64
- **Web Server**: Apache Version 2.2.9
- **PHP Version**: 5.2.6
- **Database**: MySQL Version 5.0.45; Oracle Version 9i
- **Java™ Version**: 1.6.0_10
- **Java™ SE Runtime Environment**: Build 1.6.0_10-b28
- **Java HotSpot™ 64-Bit Server VM**: Build 11.0-b15, mixed mode)

Users of **J.O.L.T** are able to access the web application through an Internet connection, with any of the following browsers (of the latest version):

- Microsoft Internet Explorer
- Mozilla Firefox
- Apple Safari
- Google Chrome
- Opera Software’s Opera Browser
3.2 Production Environment

The production environment is the set of hardware, software, and tools that a system will run on. J.O.L.T is planned to be run on Siena’s oraserv server (oraserv.cs.siena.edu), which has the following characteristics.

- **Operating System**: CentOS (Linux) Release 5.2 (Final)
- **Server Name**: oraserv.cs.siena.edu
- **CPU Type**: x86_64
- **Web Server**: Apache Version 2.2.9
- **PHP Version**: 5.2.6
- **Database**: MySQL Version 5.0.45;
- **Java™ Version**: 1.6.0_10-rc
- **Java™ SE Runtime Environment**: Build 1.6.0_10-rc-b28
- **Java HotSpot™ 64-Bit Server VM**: Build 11.0-b15, mixed mode)
### 3.3 UML Deployment Diagram

Deployment Diagrams are a Unified Modeling Language (UML) based diagram used to show devices and execution environments for a system. It represents the physical layout of the System. The Deployment Diagram for J.O.L.T below shows the different browsers connecting via HTTP to J.O.L.T. It also shows the Development Environment connecting via SCP. Finally, J.O.L.T is using various components and devices such as a database, and temporary directory.

Dropbox is a file sharing utility that 518 Interactive uses as a source control mechanism during development. It stores the ‘Test’ version of the site, which automatically synchronizes all connected computers. This ensures that all developers have the most up-to-date source code at all times.

PHPMyAdmin is a web-based database manager. It allows the development team to easily view and alter the contents of the J.O.L.T database through a web browser using a graphical interface. There is currently a version of PHPMyAdmin installed on oraserv.cs.siena.edu under 518 Interactive’s account.
3.4 Web Site Map

A Web Site Map is a diagram that shows the structure of a website. It allows the user to see the functionality available at any part of the site. The following elements are used to build a Web Site Map:

**Home Page**: Denotes the main page for each user within *J.O.L.T*.

**Home**: Denotes the initial page that all users see when navigating to *J.O.L.T*.

**Overlay/Dialog**: Denotes a window that appears within the current page (Not blocked by popup blockers)

**Web Page**: Denotes a generic page within *J.O.L.T*.

**Page Redirect**: Denotes a forced change (by *J.O.L.T*) in where the user is within the system.

**Link**: Shows links on a given Web Page.

**Possible Page Result Link**: Denotes a page that a user *may* see, depending on conditions within the page they are currently on.

**Part of Common Header**: Denotes pages that are common to all users, regardless of position within the diagram.
3.4.1 Context Web Site Map
This shows the initial pages for J.O.L.T. All subsequent Web Site Map diagrams show the User Home Pages broken down.
3.4.2 Student Web Site Map
3.4.3 Faculty Web Site Map
3.4.4 Course Coordinator Web Site Map
3.4.5 Administrator Web Site Map
4 External & Internal Design Specification

4.1 User Screens with respective User Command Summaries

Unless otherwise specified, all screenshots which are depicted in browsers are taken with Mozilla Firefox Version 3.60

Note: User Command Summaries appear under each screenshot in this section.
4.2 Student User

4.2.1 Student User Narrative
Students can register an account with the System. Once registered, students can log into the system via their unique username and password. Once logged in, students can enroll in only the courses they are currently taking. Students can view problems in a categorized manner. Students can take exams and solve individual problems created by their instructor. Students can solve problems by submitting Java source code, which the system compiles and runs against provided test data. While solving a set of problems, students are able to navigate from problem to problem without completing them in a specific order. Students can save their progress for any individual problem and work on it again during a later session.

Students have a report card view which allows them to view their own grades and progress in all current and past courses. Students are able to browse all of their own solutions as well.
4.2.2 *J.O.L.T* Login Page

This is the initial screen that Students are presented with when they navigate to *J.O.L.T*. It prompts for a username and password, and provides a link for registration and forgotten password. As well as links to the “About” and “Support” pages.
4.2.3 Student Registration Page

This is the screen that Students are presented with when they click the “Register” link on the Login Page. Students are asked to provide information that is required to register with J.O.L.T. By clicking on “Register” with all of the correct fields the user’s account is created in the database. At the top, Home takes the non-logged in user back to the home Log in Page. The “About” button allows the user to view information pertaining to the objectives of J.O.L.T and its creators. Clicking the “Support” button provides the user with information if they are encountering any difficulties.
4.2.4 Student Registration Page (Error Message – Invalid Fields)

This is an example of an error message that Students would see if they fail to fill out all fields for registration. All functionality remains the same as the original registration page.
4.2.5 Student Registration Page (Error Message – Invalid Email)

This is an example of an error message that Students would see if they fail to provide a valid email address. All functionality remains the same as the original registration page.
4.2.6 Student Registration Page (Completed)

This is a confirmation overlay that is displayed to Students when they successfully provide all required fields for registration. A confirmation email is sent to the email address that they provided.
4.2.7 Invalid Login Screen

This error message appears when a Student user enters an incorrect username.
4.2.8 Invalid Login Screen

This is what the Student sees after entering a valid username and invalid password. This screen indicates that the Student has 1 more login attempts before their account is locked out.
4.2.9 Account Lockout Screen

This screen appears when a Student user has too many incorrect login attempts for a particular username, and is locked out of the system.
4.2.10 Common Header

This header appears on all pages that the user has. The “Home” button brings the user to their homepage, the “Profile” button brings them to their profile page, the “Support” button brings them to a support page where they can get assistance, and “Log Out” terminates the user’s session with J.O.L.T. This header functionality is common to all user types in J.O.L.T. The “(S)” on the right indicates user type. (S = student, F = Faculty, CC = Course Coordinator, and A = Administrator).

4.2.11 Student Home Page

This screen appears once a Student user successfully authenticates with J.O.L.T. This screen indicates that the Student is not enrolled in any courses. Using the inline form, the student can enter a course pin to register for a section.
4.2.12 Course Enrollment (Invalid PIN)

This overlay is displayed when a user attempts to enter an incorrect PIN to enroll in a course. Clicking “OK” brings the user back to the student homepage.
4.2.13 Course Enrollment (Completed)

This screen is shown when a Student user enters a correct PIN to register for a course. The newly added course is displayed on the screen. Clicking on the course name brings the student to the section homepage. Clicking on the “Home” button brings the user back to this page.
4.2.14 Section Home Page

This is an example Section Home Page that a Student user would see once selecting a Section on their Home Page. This screen indicates that there are no tasks pending, and no announcements. The “Gradebook” button brings the student to their grade book.
This is the section homepage with an active assignment as well as an announcement. Clicking the task brings the student to the Problem Set Overview screen. Clicking the announcement opens the announcement in a dialog.
4.2.16 Problem Set Overview

This is an example Problem Set Overview Screen. This screen shows each problem within the problem set, as well as each problem’s category, difficulty point value, and completion status.
4.2.17 Solve Problem

This is an example Solve Problem screen. The Student User is presented with the Problem Name, Description, and a text area for inputting Java™ Source Code. The text area is pre-populated with a method signature. Clicking on the “Previous” and “Next” buttons brings the user to a different problem in the problem set. The “Overview” button brings the user back to the problem set overview screen.
This is the Solve Problem Screen with an example Compile Error. This is shown when a Student enters Java™ Source Code that is not syntactically correct, and then pushes the “Compile, Save, Run” Button.
4.2.19 Student Solution Attempt (Incorrect Output)

This is the Solve Problem Screen with incorrect output. This is shown when a Student clicks the “Compile, Save, Run” button after entering Java™ Source Code that is syntactically correct, but produces the wrong output.
This is the Solve Problem Screen with a timeout error. This is shown when a Student clicks the “Compile, Save, Run” button after entering Java™ Source Code that is syntactically correct, but fails to complete execution within a reasonable amount of time.
4.2.21 Updated Problem Set Overview Screen (Navigate Away)

This is an example Problem Set Overview screen with an incomplete problem. This occurs when a student navigates away from a problem once attempting (unsuccessfully) to solve it.
4.2.22 Student Solution Attempt (Completed)

This screen appears when the Student enters a Java™ solution that passes all of the provided test cases.
4.2.23 Updated Problem Set Screen

This is an example Problem Set Overview Screen showing a completed problem.
4.2.24 View Gradebook

This is an example of the Student gradebook for a particular section. This screen is sortable by name, due date, and grade. The Assignment Name fields are clickable, with each link bringing you to the respective Problem Set Overview.
This screen appears when the Student clicks the “Profile” button in the header of any page. It shows the Student’s profile information.
This screen allows a Student user to edit their profile information. The Students gets here by clicking the “Edit Profile” button on the Student Profile Page. All error checking in the registration page applies to this page.
4.2.27 View Announcement

This overlay appears when a user clicks the announcement title on their homepage or section homepage. Clicking “Close” removes the overlay and keeps the announcement in the system. Clicking “Delete” remove the announcement from the system.
This screen shows a Student who has just logged out of *J.O.L.T* by clicking the Logout button in the header.
4.3 Faculty User

4.3.1 Faculty User Case Narrative
Faculty has the ability to log into the system via a unique username and password. Once logged in, Faculty are able to select a course and then create individual problems as well as problem sets for that course. Each problem is categorized based on type and difficulty. Faculty are able to assign a grading scheme to problems and problem sets. Each problem created has the ability to be modified; however, all changes are recorded into the System as a new problem. This allows users to view problems and problem sets in a temporal manner. Problems that are partially completed are saved to a “sandbox” where they may finish it at a later date.

Faculty have the ability to submit their problems and problem sets to a “Course Pool”, which allows other faculty members who teach the same course access to their problems and problem sets. Faculty have the ability to search a Course Pool or the Global Pool.

Faculty have a grade book view which allows them to see the progress of each student that they instruct or have previously instructed. Faculty have the ability to alter any grade that was assigned to a student in their course. Faculty have the ability to send a broadcast message to students that they instruct.
4.3.2 Faculty Screens
The Faculty User shall have the following screens available.

4.3.3 J.O.L.T Login

This is the initial screen that Faculty are presented with when they navigate to J.O.L.T. It prompts for a username and password, and provides a link for forgotten password. Note that faculty accounts are created by Course Coordinators and/or the Administrator. No self-registration is allowed.

Note: All validation that appears on the student screens applies to this page for Faculty.
This is the Faculty Home Page. This page is pre-populated with their courses by the Course Coordinator. From here, the Faculty can select a course, or create an announcement.
This is the Section home page for a Faculty member. From here, they can manage the gradebook, assignments, sandbox, students, and pools. They can also create a problem or problem set, and view the course information.
This is the Create Problem page for a faculty member. From here, the faculty enters all information about a problem.
4.3.7 Create Problem (Invalid Field)

This screen shows an error message indicating an invalid field is present on the Create Problem Screen.
4.3.8 Create Problem Compiler Error

This screen shows an error message pertaining to a compiler error in the Faculty member’s solution code.
4.3.9 Faculty Sandbox (Unpopulated)

This is the Faculty sandbox. The sandbox stores all problems that are not completed and ready for the pool.

4.3.10 Faculty Sandbox (Populated)

Clicking a problem name in the sandbox will bring up a window to edit the problem. The user can also select a problem to and click "Delete" to remove it from their sandbox. The user can also Create Problems from this page (The problems created from that create problem screen do not necessarily mean that they will be saved in the sandbox.)
4.3.11 Edit Problem

This is the edit problem screen that the Faculty user will see when they choose to edit a problem from their private pool, the course pool, the global pool, or the sandbox. It is simply the create problem screen populated with the information the user provided. All error checking on the Create Problem screen exists on the edit problem screen as well.
4.3.12 Create Problem Set

This is an example of the Create Problem Set Screen. Faculty can use problems from their own pool, the course pool, or the global pool when creating a Problem Set. Clicking on a problem in the problem set area will allow the faculty user to edit the problem. Point values can be assigned to each problem.
4.3.13 Create Problem Set (Error)

This is an example error message showing invalid fields on the Create Problem Set Screen.
4.3.14 Create Problem Set (No Problems Added Error)

This is an example error message showing a Problem Set with no problems added to it.
4.3.15 Create Problem Set (Completed)

This is a confirmation message showing a created Problem Set. This is an overlay dialog that will redirect to their pool management screen.
4.3.16 Edit Problem Set (Completed)

This is the screen for Editing an existing Problem Set. The format of the screen is identical to the Create Problem Set screen. All error screens for Edit Problem Set are identical to the ones for Create Problem Set.
4.3.17 Activate/Assign Problem Set

This is the screen to activate a Problem Set for a specific course. Problem Sets can either be activated instantly, or set to be active at a later date. Problem Set Deactivation is also accomplished on this screen. Note that this screen is only used for manual deactivation. *J.O.L.T* will automatically deactivate problem sets once the expiration date and time have been passed.
4.3.18 Create Announcement

This is the screen to create an announcement to be broadcast to users. Users select recipients of the announcement, and provide a title and announcement text.
This screen shows an announcement with populated data and selected recipients. Using the left and right arrow buttons add and remove users from the recipient list.
4.3.20 Create Announcement (Missing Field)

This shows an error message for a missing field when creating an announcement. This appears once the user selects the “Submit” button.
4.3.21 Create Announcement (Complete)

This shows a successfully created announcement. This appears once the user selects the “Submit” button. Clicking on “Close” removes the overlay, but keeps the user at the announcement screen in case they wish to send another announcement.
4.3.22 View Gradebook

This is the gradebook view for a specific course for a faculty user. Clicking on an Assignment Name brings the user to the Problem Set Overview screen. Clicking on the grade allows the user to change the grade. The absence of a grade indicates that the student has not yet started the assignment. The * next to some of the grades indicates that the grades have been changed.
4.3.23 Edit Gradebook (Editing)

This shows a user editing a student’s grade. Clicking the grade opens an overlay dialog where a new grade may be entered. A comment field is also provided, but not required.
4.3.24 Manage Pools

Clicking the “Manage Pools” button on the Section Home Page brings the user to this screen. Multiple problems can be selected at once. Problems in the private pool can be copied to the course pool, and problems in higher pools can be copied into the private pool. Problems in the private pool can also be removed from the pool. Note that deleting a problem from a pool does NOT remove the problem from the database.
4.3.25 View Profile

Clicking the “Profile” button in the header will bring faculty members to this screen, where they can view their profile information. There is a button to update their information as well.
4.3.26 Edit Profile

This screen allows a Faculty user to edit their profile information. The Faculty user gets here by clicking the “Edit Profile” button on the Faculty Profile Page. All error checking that the student user has also exists on this screen.
4.3.27 Logout

This screen shows a Faculty who has just logged out of *J.O.L.T* by clicking the Logout button in the header.
4.4 Course Coordinator User

4.4.1 Course Coordinator User Case Narrative
The Course Coordinator has the ability to log into the System via a unique username and password. Once logged in, the Course Coordinator is able to create Faculty accounts. The Course Coordinator can also create courses and assign the courses to specific Faculty members. The Course Coordinator has access to course-wide reporting tools, which allows for statistical analysis of problems and problem sets.

The Course Coordinator is able to manage the “Course Pool” for each course they manage. The Course Coordinator is responsible for adding, modifying, and deleting problems and problem sets from the pool. The Course Coordinator submits problems and problem sets to the “Global Pool” for use by all faculty members. Problems that are partially completed will be saved to a “sandbox” where they may finish it at a later date.

The Course Coordinator is able to send broadcast messages to faculty members and students that participate in the courses that the Course Coordinator manages, or a subset thereof.
4.4.2 Course Coordinator User Screens
The Course Coordinator has the following screens available.

4.4.1 J.O.L.T Login

This is the initial screen that Course Coordinator is presented with when they navigate to J.O.L.T. It prompts for a username and password, and provides a link for forgotten password. Note that Course Coordinator accounts are created by the Administrator. No self-registration is allowed.

**Note:** All validation that appears on the student screens applies to this page for Course Coordinator.
4.4.2 Course Coordinator Home Page

This is the screen that the Course Coordinator is presented with after they successfully log in. From here, they can Manage Pools, Manage Courses (that they coordinate), Manage their Sandbox, Create a Problem, Create a Problem Set, and Create an Announcement. They can also manage the faculty accounts that teach the courses that the course coordinator manages.
4.4.3 Create Faculty Account

This screen is used by the Course Coordinator to create a Faculty account. All error checking on the registration screen applies to this screen.
4.4.4 Edit Faculty Account

This screen allows a Course Coordinator to edit a faculty member’s account. Entering a value in the password fields will update the user’s password.
4.4.5 Create Course Section

From the Manage Courses button, the “Create Section” button will show this overlay dialog, which will allow the course coordinator to create a section for any course that they manage. Note the Course Enrollment Pin is automatically generated by the system. The Pin can be later changed, if necessary.
4.4.6 Create Course Section (Error)

This screen indicates an error in one or more fields while creating a new section. This appears after clicking the “Create Section” Button.
4.4.7 Edit Course Section

This overlay allows a Course Coordinator to edit an existing Course Section. Clicking on the course name under the “Manage Courses” screen will bring the user to this overlay. All error checking applies to the edit screen as the create course screen.
4.4.1 View Gradebook

This is the gradebook view for a specific course for a course coordinator user. Clicking on an Assignment Name brings the user to the Problem Set Overview screen. Clicking on the grade allows the user to change the grade. The absence of a grade indicates that the student has not yet started the assignment. The * next to some of the grades indicates that the grades have been changed.
4.4.2 Edit Gradebook (Editing)

This shows a user editing a student’s grade. Clicking the grade opens an overlay dialog where a new grade may be entered. A comment field is also provided, but not required.
4.4.1 Create Problem

This is the Create Problem page for a Course Coordinator member. From here, the user enters all information about a problem.
4.4.2 Create Problem (Invalid Field)

This screen shows an error message indicating an invalid field is present on the Create Problem Screen.
This screen shows an error message pertaining to a compiler error in the Course Coordinator member’s solution code.
4.4.1 Course Coordinator Sandbox (Unpopulated)

This is the Course Coordinator sandbox. The sandbox stores all problems that are not completed and ready for the pool.

4.4.2 Course Coordinator Sandbox (Populated)

Clicking a problem name in the sandbox will bring up a window to edit the problem. The user can also select a problem to and click “Delete” to remove it from their sandbox. The user can also Create Problems from this page (The problems created from that create problem screen do not necessarily mean that they will be saved in the sandbox.)
4.4.3 Edit Problem

This is the edit problem screen that the Course Coordinator user will see when they choose to edit a problem from either their private pool, the course pool, the global pool, or the sandbox. It is simply the create problem screen populated with the information the user provided. All error checking on the Create Problem screen exists on the edit problem screen as well.
4.4.4 Create Problem Set

This is an example of the Create Problem Set Screen. Course Coordinators can use problems from their own pool, the course pool, or the global pool when creating a Problem Set. Clicking on a problem in the problem set area will allow the user to edit the problem. Point values can be assigned to each problem. By selecting problems in the pools using the checkboxes and clicking the “Add to Problem Set” button the problems in the pool are added to the problem set area located underneath the “Total Points” and “Number of Problems” fields.
4.4.5 Create Problem Set (Error)

This is an example error message showing invalid fields on the Create Problem Set Screen.
4.4.6 Create Problem Set (No Problems Added Error)

This is an example error message showing a Problem Set with no problems added to it.
4.4.7 Create Problem Set (Completed)

This is a confirmation message showing a created Problem Set. This is an overlay dialog that will redirect to their pool management screen.
4.4.8 Edit Problem Set (Completed)

This is the screen for Editing an existing Problem Set. The format of the screen is identical to the Create Problem Set screen. All error screens for Edit Problem Set are identical to the ones for Create Problem Set.
4.4.9 Activate/Assign Problem Set

This is the screen to activate a Problem Set for a specific course. Problem Sets can either be activated instantly, or set to be active at a later date. Problem Set Deactivation is also accomplished on this screen. Note that this screen is only used for manual deactivation. J.O.L.T will automatically deactivate problem sets once the expiration date and time have been passed.
This is the screen to create an announcement to be broadcast to users. Users select recipients of the announcement, and provide a title and announcement text.
4.4.11 Create Announcement (Add Recipients)

This screen shows an announcement with populated data and selected recipients. Using the left and right arrow buttons add and remove users from the recipient list.
4.4.12 Create Announcement (Missing Field)

This shows an error message for a missing field when creating an announcement. This appears once the user selects the “Submit” button.
4.4.13 Create Announcement (Complete)

This shows a successfully created announcement. This appears once the user selects the “Submit” button. Clicking on “Close” removes the overlay, but keeps the user at the announcement screen in case they wish to send another announcement.
4.4.14 Manage Pools

This is the Pool Management screen for Course Coordinators. Clicking the “Manage Pools” button on the Section Home Page brings the user to this screen. Multiple problems can be selected at once. Problems in the private pool can be copied to the course pool. Problems in the private pool can also be removed from the pool. Note that deleting a problem from a pool does NOT remove the problem from the database.
4.4.15 Accept Private Problems to Course Pool

Clicking the “Pending Problems” button brings the Course Coordinator to this screen. From here, the Course Coordinator has the ability to approve or reject problems to the Course Pool. Clicking the problem name allows the Course Coordinator to view the problem details.
4.4.16 Create Announcement

This is the screen to create an announcement to be broadcast to users. Users select recipients of the announcement, and provide a title and announcement text.
4.4.17 Create Announcement (Add Recipients)

This screen shows an announcement with populated data and selected recipients. Using the left and right arrow buttons add and remove users from the recipient list.
4.4.18 Create Announcement (Missing Field)

This shows an error message for a missing field when creating an announcement. This appears once the user selects the “Submit” button.
4.4.19 Create Announcement (Complete)

This shows a successfully created announcement. This appears once the user selects the “Submit” button. Clicking on “Close” removes the overlay, but keeps the user at the announcement screen in case they wish to send another announcement.
4.4.1 View Profile

Clicking the “Profile” button in the header will bring course coordinators members to this screen, where they can view their profile information. There is a button to update their information as well.
This screen allows a Course Coordinator user to edit their profile information. The user gets here by clicking the “Edit Profile” button on the Course Coordinator Profile Page. All error checking that the student user has also exists on this screen.
4.4.3 Logout

This screen shows a Course Coordinator who has just logged out of J.O.L.T by clicking the Logout button in the header.
4.5 Administrator User

4.5.1 Administrator User Case Narrative
The Administrator is able to log into the System via a unique username and password. Once logged in, the Administrator can create and manage Course Coordinator, Faculty, and Student accounts. The Administrator has the same abilities as a Course Coordinator. The Administrator is able to send broadcast messages to all users, or a subset thereof. The Administrator manages the “Global Pool” of problems and problem sets.

4.5.2 Administrator User Screens
The Administrator has the following screens available.
4.5.3 J.O.L.T Login

This is the initial screen that the Administrator is presented with when they navigate to J.O.L.T. It prompts for a username and password, and provides a link for forgotten password.

Note: All validation that appears on the student screens applies to this page for the Administrator. Note that the administrator account cannot get locked out.
This is the home page for the Administrator. From here, they can Manage Pools, Manage Courses, Manage their Sandbox, Create a Problem, Create a Problem Set, and Create an Announcement. They can also manage all user accounts, and Create Courses and Sections.
4.5.5 Manage Users

This screen is reached from the “Manage Users” button on the Administrator home page. This screen allows the Administrator to edit and add user accounts.
4.5.6 Create Course Coordinator

This is the screen the Administrator uses to create a Course Coordinator account. All fields are verified for correct values.
4.5.7 Create Faculty Account

This is the screen the Administrator uses to create a Faculty account. All fields are verified for correct values.
4.5.8 Create Student Account

This is the screen the Administrator uses to create a Student account. All fields are verified for correct values.
4.5.9 Edit User Account

This screen is the same for all user types. The administrator has the ability to update all fields for user account management.
4.5.10 Course Management

Clicking on the “Manage Courses” button brings the administrator to this screen, where they can create sections, create courses, and manage existing sections and courses (by clicking on them).
4.5.11 Create Course

This overlay allows the Administrator to create a course from the “Manage Courses” screen. The Administrator can also create sections as seen in the course coordinator screen “Create Course Section/” All error checking applies.
4.5.12 View Gradebook

This is the gradebook view for a specific course for an administrator user. Clicking on an Assignment Name brings the user to the Problem Set Overview screen. Clicking on the grade allows the user to change the grade. The absence of a grade indicates that the student has not yet started the assignment. The * next to some of the grades indicates that the grades have been changed.
4.5.13 Edit Gradebook (Editing)

This shows a user editing a student’s grade. Clicking the grade opens an overlay dialog where a new grade may be entered. A comment field is also provided, but not required.
4.5.14 Create Problem

This is the Create Problem page for the Administrator. From here, the user enters all information about a problem.
This screen shows an error message indicating an invalid field is present on the Create Problem Screen.
4.5.16 Create Problem Compiler Error

This screen shows an error message pertaining to a compiler error in the Administrator member’s solution code.
4.5.17 Administrator Sandbox (Unpopulated)

This is the Administrator sandbox. The sandbox stores all problems that are not completed and ready for the pool.

4.5.18 Administrator Sandbox (Populated)

Clicking a problem name in the sandbox will bring up a window to edit the problem. The user can also select a problem to and click “Delete” to remove it from their sandbox. The user can also Create Problems from this page (The problems created from that create problem screen do not necessarily mean that they will be saved in the sandbox.)
4.5.19 Edit Problem

This is the edit problem screen that the Administrator user will see when they choose to edit a problem from their private pool, the course pool, the global pool, or the sandbox. It is simply the create problem screen populated with the information the user provided. All error checking on the Create Problem screen exists on the edit problem screen as well.
4.5.20 Create Problem Set

This is an example of the Create Problem Set Screen. Administrators can use problems from their own pool, the course pool, or the global pool when creating a Problem Set. Clicking on a problem in the problem set area will allow the faculty user to edit the problem. Point values can be assigned to each problem.
4.5.21 Create Problem Set (Error)

This is an example error message showing invalid fields on the Create Problem Set Screen.
4.5.22 Create Problem Set (No Problems Added Error)

This is an example error message showing a Problem Set with no problems added to it.
4.5.23 Create Problem Set (Completed)

This is a confirmation message showing a created Problem Set. This is an overlay dialog that will redirect to their pool management screen.
This is the screen for editing an existing Problem Set. The format of the screen is identical to the Create Problem Set screen. All error screens for Edit Problem Set are identical to the ones for Create Problem Set.
4.5.25 Activate/Assign Problem Set

This is the screen to activate a Problem Set for a specific course. Problem Sets can either be activated instantly, or set to be active at a later date. Problem Set deactivation is also accomplished on this screen. Note that this screen is only used for manual deactivation. J.O.L.T will automatically deactivate problem sets once the expiration date and time have been passed.
This is the screen to create an announcement to be broadcast to users. Users select recipients of the announcement, and provide a title and announcement text.
4.5.27 Create Announcement (Add Recipients)

This screen shows an announcement with populated data and selected recipients. Using the left and right arrow buttons add and remove users from the recipient list.
4.5.28 Create Announcement (Missing Field)

This shows an error message for a missing field when creating an announcement. This appears once the user selects the “Submit” button.
4.5.29 Create Announcement (Complete)

This shows a successfully created announcement. This appears once the user selects the “Submit” button. Clicking on “Close” removes the overlay, but keeps the user at the announcement screen in case they wish to send another announcement.
4.5.30 Manage Pools

This is the pool management screen for the administrator user. Clicking the “Manage Pools” button on the Section Home Page brings the user to this screen. Multiple problems can be selected at once. Problems in the private pool can be copied to the global pool. Problems in the private pool can also be removed from the pool. Note that deleting a problem from a pool does NOT remove the problem from the database.
4.5.1 Accept Private/Course Problems to Global Pool

Clicking the “Pending Problems” button brings the Administrator to this screen. From here, the Administrator has the ability to approve or reject problems to the Global Pool. Clicking the problem name allows the Administrator to view the problem details.
### 4.5.2 View Profile

Clicking the “Profile” button in the header will bring the Administrator member to this screen, where they can view their profile information. There is a button to update their information as well.
4.5.3 Edit Profile

This screen allows the Administrator user to edit their profile information. The Administrator user gets here by clicking the “Edit Profile” button on the Administrator Profile Page.

All error checking for previous users for this screen applies to the Administrator.
4.5.4 Logout

This screen shows the Administrator who has just logged out of J.O.L.T by clicking the Logout button in the header.
4.6 Logical Data Stores
The logical data stores are a set of tables that represent the different fields for the MySQL database. The first part of this is a tabular, text-based description, followed by a graphical representation of the tables.

4.6.1 Announcement Table
This table stores all information regarding announcements within the system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>announcementId</td>
<td>int(5)</td>
<td>No</td>
<td></td>
<td>unique ID for each announcement</td>
</tr>
<tr>
<td>fromUser</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>username of announcement sender</td>
</tr>
<tr>
<td>toUser</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>username of announcement receiver</td>
</tr>
<tr>
<td>sendDate</td>
<td>int(20)</td>
<td>Yes</td>
<td>NULL</td>
<td>Timestamp of announcement</td>
</tr>
<tr>
<td>subject</td>
<td>varchar(50)</td>
<td>Yes</td>
<td>NULL</td>
<td>Subject of announcement</td>
</tr>
<tr>
<td>message</td>
<td>varchar(1000)</td>
<td>Yes</td>
<td>NULL</td>
<td>Announcement Text</td>
</tr>
</tbody>
</table>
### 4.6.2 Assignment Table

This table lists all assignments (past and current) for all courses in the system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>assignmentId</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>Unique id for each assignment</td>
</tr>
<tr>
<td>assignmentName</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>Name of Assignment</td>
</tr>
<tr>
<td>beginOn</td>
<td>int(20)</td>
<td>Yes</td>
<td>NULL</td>
<td>timestamp of when problem should be activated</td>
</tr>
<tr>
<td>endOn</td>
<td>int(20)</td>
<td>Yes</td>
<td>NULL</td>
<td>timestamp of when problem should be deactivated</td>
</tr>
<tr>
<td>problemSet</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>id of problem set that the assignment refers to</td>
</tr>
<tr>
<td>assignedTo</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Section id that assignment is assigned to</td>
</tr>
</tbody>
</table>
### 4.6.3 Course Table
This table stores course information for the system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>courseId</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>Unique course id for each course</td>
</tr>
<tr>
<td>courseName</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>Course Name (i.e. Intro To Programming)</td>
</tr>
<tr>
<td>courseNumber</td>
<td>varchar(10)</td>
<td>Yes</td>
<td>NULL</td>
<td>Course Number (i.e. CSIS-010)</td>
</tr>
<tr>
<td>createdBy</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>Username that created the course</td>
</tr>
<tr>
<td>managedBy</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>The Course Coordinator that manages the course</td>
</tr>
<tr>
<td>poolId</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of the pool that belongs to the course</td>
</tr>
</tbody>
</table>
4.6.4 Enrolled Students Table

This table shows which students are enrolled in which section it also shows past and current enrollment information.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>varchar(30)</td>
<td>No</td>
<td>username of student enrolled in section</td>
<td></td>
</tr>
<tr>
<td>sectionId</td>
<td>int(9)</td>
<td>No</td>
<td>0</td>
<td>section id of section that student is enrolled in</td>
</tr>
</tbody>
</table>
4.6.5 Grade Log Table
This table shows all grade adjustments made by Faculty, Course Coordinators, and Administrators.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>assignment</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>Assignment Id of changed grade</td>
</tr>
<tr>
<td>faculty</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td>Username of individual who modified grade</td>
</tr>
<tr>
<td>student</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td>Student whose grade was changed</td>
</tr>
<tr>
<td>timestamp</td>
<td>int(10)</td>
<td>No</td>
<td></td>
<td>Date/Time of Grade Modification</td>
</tr>
<tr>
<td>oldGrade</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>Original grade for student.</td>
</tr>
<tr>
<td>newGrade</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>New grade for student.</td>
</tr>
<tr>
<td>comment</td>
<td>varchar(100)</td>
<td>Yes</td>
<td>NULL</td>
<td>Comment of grade change (Optional, viewable by student if set)</td>
</tr>
</tbody>
</table>
### 4.6.6 Pool Table
This table contains the information for all pools in the system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>poolId</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>unique id for each pool</td>
</tr>
<tr>
<td>poolType</td>
<td>int(1)</td>
<td>Yes</td>
<td>NULL</td>
<td>type of pool (0=sandbox, 1=private, 2=course, 3=global)</td>
</tr>
<tr>
<td>poolOwner</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>userid of pool owner</td>
</tr>
</tbody>
</table>
### 4.6.7 Problem Table

This table stores all information regarding each problem of the system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>problemId</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>unique id for each problem</td>
</tr>
<tr>
<td>problemName</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td>Name of Problem</td>
</tr>
<tr>
<td>problemCategory</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td>Category of problem</td>
</tr>
<tr>
<td>problemDescription</td>
<td>varchar(500)</td>
<td>No</td>
<td></td>
<td>description of problem</td>
</tr>
<tr>
<td>problemActive</td>
<td>int(1)</td>
<td>No</td>
<td>0</td>
<td>field indicating that problem is active (i.e., that it passes all checks, a valid solution is specified, etc) and can be used in a problem set.</td>
</tr>
<tr>
<td>totalAttempts</td>
<td>int(9)</td>
<td>No</td>
<td>0</td>
<td>number of times this problem has been attempted to be solved.</td>
</tr>
<tr>
<td>correctAttempts</td>
<td>int(9)</td>
<td>No</td>
<td>0</td>
<td>number of times this problem has been successfully solved.</td>
</tr>
<tr>
<td>createdOn</td>
<td>int(10)</td>
<td>No</td>
<td></td>
<td>timestamp of when the problem was created.</td>
</tr>
<tr>
<td>createdBy</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td>username if problem creator</td>
</tr>
<tr>
<td>methodSignature</td>
<td>varchar(100)</td>
<td>No</td>
<td></td>
<td>method signature of problem</td>
</tr>
<tr>
<td>methodName</td>
<td>varchar(40)</td>
<td>No</td>
<td></td>
<td>name of method of problem</td>
</tr>
<tr>
<td>numParameters</td>
<td>int(1)</td>
<td>No</td>
<td>1</td>
<td>number of parameters the method contain</td>
</tr>
<tr>
<td>parameters</td>
<td>varchar(60)</td>
<td>No</td>
<td></td>
<td>parameter types, in object format (Integer, Boolean, Character, etc)</td>
</tr>
<tr>
<td>numTestCases</td>
<td>int(2)</td>
<td>No</td>
<td></td>
<td>number of test cases for this problem</td>
</tr>
<tr>
<td>parm1Name</td>
<td>varchar(20)</td>
<td>Yes</td>
<td>NULL</td>
<td>Name of first parameter of problem</td>
</tr>
<tr>
<td>parm1Type</td>
<td>varchar(10)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data type of first parameter of problem</td>
</tr>
<tr>
<td>parm2Name</td>
<td>varchar(20)</td>
<td>Yes</td>
<td>NULL</td>
<td>Name of second parameter of problem</td>
</tr>
<tr>
<td>parm2Type</td>
<td>varchar(10)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data type of second parameter of problem</td>
</tr>
<tr>
<td>parm3Name</td>
<td>varchar(20)</td>
<td>Yes</td>
<td>NULL</td>
<td>Name of third parameter of problem</td>
</tr>
<tr>
<td>parm3Type</td>
<td>varchar(10)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data type of third parameter of problem</td>
</tr>
<tr>
<td>parm4Name</td>
<td>varchar(20)</td>
<td>Yes</td>
<td>NULL</td>
<td>Name of fourth parameter of problem</td>
</tr>
<tr>
<td>parm4Type</td>
<td>varchar(10)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data type of fourth parameter of problem</td>
</tr>
<tr>
<td>parm5Name</td>
<td>varchar(20)</td>
<td>Yes</td>
<td>NULL</td>
<td>Name of fifth parameter of problem</td>
</tr>
<tr>
<td>parm5Type</td>
<td>varchar(10)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data type of fifth parameter of problem</td>
</tr>
<tr>
<td>resultType</td>
<td>varchar(10)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data type of result of problem</td>
</tr>
<tr>
<td>solution</td>
<td>varchar(1000)</td>
<td>No</td>
<td></td>
<td>Faculty provided solution</td>
</tr>
<tr>
<td>publishSolution</td>
<td>int(1)</td>
<td>No</td>
<td>0</td>
<td>Field indicating that the solution should be published for the students to see.</td>
</tr>
</tbody>
</table>
## 4.6.8 Problem Location Table

This table links problems to pools in the system. Problems may exist in more than one pool.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>problemId</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>Problem ID identifying a unique problem in the Problem table</td>
</tr>
<tr>
<td>poolId</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>Pool ID identifying the pool that the problem resides in</td>
</tr>
<tr>
<td>status</td>
<td>int(1)</td>
<td>No</td>
<td>0</td>
<td>Status of problem (0=pending approval, 1=approved)</td>
</tr>
</tbody>
</table>
### 4.6.9 Problem Set Table

This table shows data regarding the problem sets.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>setId</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>Unique id for each problem set</td>
</tr>
<tr>
<td>setName</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>Name of Problem Set</td>
</tr>
<tr>
<td>setCategory</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>Category of Problem Set</td>
</tr>
<tr>
<td>setDescription</td>
<td>varchar(500)</td>
<td>Yes</td>
<td>NULL</td>
<td>Description of Problem Set</td>
</tr>
<tr>
<td>numProblems</td>
<td>int(2)</td>
<td>Yes</td>
<td>NULL</td>
<td>Number of Problems in problem set</td>
</tr>
<tr>
<td>problem1Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #1 in Problem set</td>
</tr>
<tr>
<td>problem1Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem2Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #2 in Problem set</td>
</tr>
<tr>
<td>problem2Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem3Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #3 in Problem set</td>
</tr>
<tr>
<td>problem3Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem4Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #4 in Problem set</td>
</tr>
<tr>
<td>problem4Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem5Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #5 in Problem set</td>
</tr>
<tr>
<td>problem5Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem6Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #6 in Problem set</td>
</tr>
<tr>
<td>problem6Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem7Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #7 in Problem set</td>
</tr>
<tr>
<td>problem7Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem8Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #8 in Problem set</td>
</tr>
<tr>
<td>problem8Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem9Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #9 in Problem set</td>
</tr>
<tr>
<td>problem9Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem10Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #10 in Problem set</td>
</tr>
<tr>
<td>problem10Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem11Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #11 in Problem set</td>
</tr>
<tr>
<td>problem11Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem12Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #12 in Problem set</td>
</tr>
<tr>
<td>problem12Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem13Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #13 in Problem set</td>
</tr>
<tr>
<td>problem13Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem14Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #14 in Problem set</td>
</tr>
<tr>
<td>problem14Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem15Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #15 in Problem set</td>
</tr>
<tr>
<td>problem15Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem16Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #16 in Problem set</td>
</tr>
<tr>
<td>problem16Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem17Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #17 in Problem set</td>
</tr>
<tr>
<td>problem17Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem18Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #18 in Problem set</td>
</tr>
<tr>
<td>problem18Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem19Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #19 in Problem set</td>
</tr>
<tr>
<td>Problem ID</td>
<td>Type</td>
<td>Required</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>problem19Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem20Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #20 in Problem set</td>
</tr>
<tr>
<td>problem20Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem21Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #21 in Problem set</td>
</tr>
<tr>
<td>problem21Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem22Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #22 in Problem set</td>
</tr>
<tr>
<td>problem22Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem23Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #23 in Problem set</td>
</tr>
<tr>
<td>problem23Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem24Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #24 in Problem set</td>
</tr>
<tr>
<td>problem24Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
<tr>
<td>problem25Id</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Id of Problem #25 in Problem set</td>
</tr>
<tr>
<td>problem25Point</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Point value for this problem</td>
</tr>
</tbody>
</table>
### 4.6.10 Section Table

This table shows information about each section in the system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>sectionId</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>Unique id for each section</td>
</tr>
<tr>
<td>semester</td>
<td>varchar(6)</td>
<td>Yes</td>
<td>NULL</td>
<td>Semester that section is active for</td>
</tr>
<tr>
<td>year</td>
<td>int(4)</td>
<td>Yes</td>
<td>NULL</td>
<td>Year that section is active for</td>
</tr>
<tr>
<td>courseId</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>ID of course section belongs to</td>
</tr>
<tr>
<td>sectionName</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>Name of section</td>
</tr>
<tr>
<td>sectionNumber</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>Number of Section</td>
</tr>
<tr>
<td>faculty</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>Faculty username in charge of section</td>
</tr>
<tr>
<td>coordinator</td>
<td>varchar(30)</td>
<td>Yes</td>
<td>NULL</td>
<td>Coordinator for course that section lies in.</td>
</tr>
<tr>
<td>enrollPin</td>
<td>varchar(10)</td>
<td>No</td>
<td></td>
<td>PIN for section that students use to enroll in course</td>
</tr>
</tbody>
</table>

### 4.6.11 Set Location Table

This table links problems sets to pools in the System. Problems sets may exist in more than one pool.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>setId</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>Problem Set ID identifying a unique problem in the Problem Set table</td>
</tr>
<tr>
<td>poolId</td>
<td>int(9)</td>
<td>No</td>
<td></td>
<td>Pool ID identifying the pool that the problem set resides in</td>
</tr>
<tr>
<td>status</td>
<td>int(1)</td>
<td>No</td>
<td>0</td>
<td>Status of problem (0=pending approval, 1=approved)</td>
</tr>
</tbody>
</table>
### 4.6.12 Solution Attempt Table

This table stores the most recent attempt for each student at solving a problem in the system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>assignmentId</td>
<td>int(9)</td>
<td>No</td>
<td>0</td>
<td>Assignment ID that solution attempt belongs to.</td>
</tr>
<tr>
<td>problemId</td>
<td>int(9)</td>
<td>No</td>
<td>0</td>
<td>Problem ID that is being solved</td>
</tr>
<tr>
<td>student</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td>Username of student attempting a solution</td>
</tr>
<tr>
<td>timestamp</td>
<td>int(20)</td>
<td>Yes</td>
<td>NULL</td>
<td>Timestamp of last submission</td>
</tr>
<tr>
<td>code</td>
<td>varchar(100)</td>
<td>Yes</td>
<td>NULL</td>
<td>Java Source that student used</td>
</tr>
<tr>
<td>numAttempts</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Number of attempts student has used for this particular solution</td>
</tr>
<tr>
<td>score</td>
<td>int(9)</td>
<td>Yes</td>
<td>NULL</td>
<td>Grade that student received (NULL indicates unsuccessful attempt)</td>
</tr>
<tr>
<td>comment</td>
<td>varchar(100)</td>
<td>Yes</td>
<td>NULL</td>
<td>Comment field for faculty to explain a grade change.</td>
</tr>
</tbody>
</table>

### 4.6.13 Test Case Table

This table stores a single test case for a problem in the system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>problemId</td>
<td>int(9)</td>
<td>No</td>
<td>0</td>
<td>Id of problem that test case belongs to</td>
</tr>
<tr>
<td>testCaseNumber</td>
<td>int(2)</td>
<td>No</td>
<td>0</td>
<td>The test case number (1 – 25)</td>
</tr>
<tr>
<td>param1</td>
<td>varchar(100)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data for first parameter of test case.</td>
</tr>
<tr>
<td>param2</td>
<td>varchar(100)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data for second parameter of test case.</td>
</tr>
<tr>
<td>param3</td>
<td>varchar(100)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data for third parameter of test case.</td>
</tr>
<tr>
<td>param4</td>
<td>varchar(100)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data for fourth parameter of test case.</td>
</tr>
<tr>
<td>param5</td>
<td>varchar(100)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data for fifth parameter of test case.</td>
</tr>
<tr>
<td>result</td>
<td>varchar(100)</td>
<td>Yes</td>
<td>NULL</td>
<td>Data for result of test case.</td>
</tr>
<tr>
<td>hidden</td>
<td>int(1)</td>
<td>Yes</td>
<td>NULL</td>
<td>Indicates that this is a hidden test case.</td>
</tr>
</tbody>
</table>
### 4.6.14 User Table

This table stores the information for all users in the system.

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Null</th>
<th>Default</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td>username of user</td>
</tr>
<tr>
<td>password</td>
<td>varchar(32)</td>
<td>No</td>
<td></td>
<td>md5 hash of user password</td>
</tr>
<tr>
<td>firstname</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td>user first name</td>
</tr>
<tr>
<td>lastname</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td>user last name</td>
</tr>
<tr>
<td>email</td>
<td>varchar(30)</td>
<td>No</td>
<td></td>
<td>user email address</td>
</tr>
<tr>
<td>squestion</td>
<td>varchar(50)</td>
<td>No</td>
<td></td>
<td>security question for password reset</td>
</tr>
<tr>
<td>answer</td>
<td>varchar(100)</td>
<td>No</td>
<td></td>
<td>security answer for password reset</td>
</tr>
<tr>
<td>gradSemester</td>
<td>varchar(6)</td>
<td>Yes</td>
<td>NULL</td>
<td>Graduation Semester (Spring/Fall) - Only used for students</td>
</tr>
<tr>
<td>gradYear</td>
<td>int(4)</td>
<td>Yes</td>
<td>NULL</td>
<td>Graduation year (20XX) - Only used for students</td>
</tr>
<tr>
<td>loginAttempts</td>
<td>int(1)</td>
<td>No</td>
<td>0</td>
<td>number of incorrect login attempts</td>
</tr>
<tr>
<td>userType</td>
<td>int(1)</td>
<td>No</td>
<td>0</td>
<td>type of user (0=student, 1=faculty, 2=course coordinator, 3=administrator)</td>
</tr>
<tr>
<td>confirmed</td>
<td>varchar(32)</td>
<td>No</td>
<td></td>
<td>This is a md5 hash of the username field concatenated with the email field. When user successfully registers, the field changes to &quot;true&quot;</td>
</tr>
<tr>
<td>Field</td>
<td>Type</td>
<td>Nullable</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>bornOn</td>
<td>varchar(20)</td>
<td>No</td>
<td>Timestamp of when the account was registered.</td>
<td></td>
</tr>
<tr>
<td>active</td>
<td>int(1)</td>
<td>No</td>
<td>Field indicating status of account (1=active, 0=inactive).</td>
<td></td>
</tr>
</tbody>
</table>

### 4.6.15 Graphical Representation Key

- **Could Be Null** 🔄
- **Not Null** 🔴
- **Primary Key or Foreign Key** 🎉
- **Table Name** 📚 *problem*
- **Links Primary Key to Primary Key** 💻
- **Link meaning at least one of the fields is not a Primary Key** ————
4.6.16 Graphical Representation (Entity Relationship Diagram)
4.7 Source Code

The Source Code document contains the source code that comprises the system.

The Source Code document is in an accompanied document titled “Source Code”, which can be found under the “Documents” link of the 518 Interactive Website.

4.8 Deliverables

518 Interactive will provide all required source files and setup scripts (such as database create table statements) to the client, Dr. Lim, on a portable media device, such as a CD or DVD. Instructions on using the system, installation, setup, and configuration will also be provided. It will include the following:

* Test Plan & Test Results.

* A copy of the Acceptance Test PowerPoint Presentation.

* A full copy of the completed team files from the 518 Interactive team directory, including, all website files (all folders, files, images, etc.)

* A README.TXT file that explains what files are where.
5 Appendices

- Appendix 1: Sources of Information
- Appendix 2: Glossary of Terms
- Appendix 3: Current Timeline
- Appendix 4: Data Flow Diagrams
- Appendix 5: Activity Diagrams
- Appendix 6: Data Dictionary
- Appendix 7: Test Plan
- Appendix 8: Source Code

5.1 Appendix 1: Sources of Information

Information found within this Requirement Specification document has been obtained through meetings with our client, Dr. Darren Lim. Information was also obtained through Dr. Lederman’s Software Engineering lectures. Information has also been collected from various internet resources, as well as requirement specification documents from previous years.
5.2 Appendix 2: Glossary of Terms

The following are a list of technical terms used within the document. This section is provided to clarify their meaning.

**Actor:** An entity in UML Use Case Diagrams and UML Activity Diagrams. It represents the human and non-human external entities (outside the system boundary) that interact with the system.

**Activity Diagram:** A diagram based on the Unified Model Language (UML). This represents the processes that comprise a certain activity within the system. These diagrams are generally created with the perspective of an actor in mind.

**Client:** Used to refer to Dr. Darren Lim, the client of 518 Interactive who requested J.O.L.T.

**Compiler:** A program that reads in source code and generates an executable.

**CSS:** Cascading Style Sheets – Used within HTML documents in order to control the presentation of web pages.

**DFD:** Data Flow Diagrams are used to show how data moves and is processed within a system. There are various levels to DFDs, with each subsequent level providing more detail than the previous.

**Hardware:** The tangible components of a computer and server. Examples include monitors, disk drives, printers, keyboard, processor, and memory.

**HTML:** Hypertext Markup Language is the scripting language used to describe the information contained on a website. HTML utilizes Cascading Style Sheets (CSS) to generate the style of the page. HTML and CSS are parsed by web browsers, such as Internet Explorer and Firefox, to render the websites for users.

**Java:** A programming language which the System will be able to compile and execute. This language will be used by the students to solve the assigned problems.

**Java Byte Code:** The output of the Java™ compiler upon successful compilation of Java™ source code. Java Byte Code is read by the Java™ runtime environment, which in turn executes the proper machine-level commands.

**Java SDK:** Software Development Kit for Java – a collection of tools used by developers to aid in the creation of programs. The Java SDK includes the Java™ (V. 1.6) compiler. The Java™ SDK also includes the Java™ (V. 1.6) runtime environment, which allows for Java™ Byte Code to be executed.

**J.O.L.T:** Java Online Learning Tool is the name of the system being developed for Dr. Lim, the client of 518 Interactive.

**MySQL:** A free implementation of a Relational Database Management System. Used to store and retrieve information relevant to the website, such as usernames, passwords, problems, solutions, and scores. Accessing information within the database is achieved by submitting a “query” in the Structured Query Language (SQL) form.
**PHP**: PHP Hypertext Processor is a programming language used to create dynamic web sites. Has the ability to interact with a database.

**Software**: The intangible components of a computer and server. It is a set of machine-level instructions that is run from within the memory, and is used to perform a specific set of functions. Examples include Microsoft Word, Adobe Photoshop, and Mozilla Firefox.

**Source Code**: A document that a compiler parses to generate machine code (which the computer can run directly), or code that gets interpreted by a third-party application, which then gets executed.

**Source/Sink**: This is a term used within Data Flow Diagrams to represent an entity that either provides (source) or receives (sink) data.

**System**: Used within this document to describe the Java Online Learning Tool (J.O.L.T).

**UML**: Unified Modeling Language is the industry-standard language for the specification, visualization, construction, and documentation of the components of software systems.

**Use Case Diagram**: Represents the high-level functions of the system. It also depicts how actors interact with each of those functions.
5.3 Appendix 3: Current Timeline
5.4 Data Flow Diagrams
The Data Flow Diagrams (DFDs) are used for structure analysis and design. DFDs show the flow of data from external entities into the system. DFDs also show how the data moves and is transformed from one process to another, as well as its logical storage.

Data Flow Diagrams are in an accompanied document titled “Data Flow Diagrams”, which can be found under the “Documents” link of the 518 Interactive Website.

5.5 Activity Diagrams
Activity Diagrams are a UML (Unified Modeling Language) specified diagram which shows workflows of stepwise activities and actions, with support for choice, iteration, and concurrency. It outlines the process that Actors (both human and non-human) go through while interacting with the System to accomplish a specific task.

Activity Diagrams are in an accompanied document titled “Activity Diagrams”, which can be found under the “Documents” link of the 518 Interactive Website.

5.6 Data Dictionary
The Data Dictionary lists all data entities within J.O.L.T.

The Data Dictionary is in an accompanied document titled “Data Dictionary”, which can be found under the “Documents” link of the 518 Interactive Website.
5.7 Test Results
The Test Results document contains the overview of the testing that took place as well as the results of every unit test.

The Test Results Document is in an accompanied document titled “Test Results”, which can be found under the “Documents” link of the 518 Interactive Website.

5.8 Source Code
The Source Code document contains the source code that comprises the system.

The Source Code document is in an accompanied document titled “Source Code”, which can be found under the “Documents” link of the 518 Interactive Website.