Welcome

Client:
Dr. Darren Lim
Assistant Professor
Siena College

Instructor:
Dr. Timoth C. Lederman
Our Professor
Siena College

Special Guests
The Team

- Lawrence Gregory - Team Leader
- Erik Stegmann - Lead Graphics and Interfaces Designer
- Christopher Hughto - Webmaster
- Jedidiah Turnbull - Systems Administrator
- Connor Vander Bogart - Organizational Information Manager
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives
- Data Flow Diagrams
- Requirements Inventory
- Data Design and Dictionary
- Testing
- Student Screens
- Faculty Screens
- Course Coordinator Screens
- Administrator Screens
- What’s Next?
- Questions
Restatement of Problem

- Dr. Darren Lim, an Assistant Professor at Siena College, has a busy schedule.

- Programming projects are time consuming to grade.

- Manual testing and grading is inefficient for both students and faculty.
Where we are in Development
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives
- Data Flow Diagrams
- Requirements Inventory
- Data Design and Dictionary
- Testing
- Student Screens
- Faculty Screens
- Course Coordinator Screens
- Administrator Screens
- What’s Next?
- Questions
User Case Narratives

User Case Narratives provide:

Description of user interaction with the system

Description of a specific user’s goals when interacting with the system
Student User Case Narrative

Student User:

- Register
- Authenticate
- View Problems
- Solve Problem Sets
- View Grades
Faculty User Case Narrative

Faculty User:

- Manage Problems
- Manage Problem Sets
- Manage Respective Student User Grades
Course Coordinator
User Case Narrative

Course Coordinator User:

- Create Faculty Accounts
- Manage Courses, Faculty, and Students
- Manage Course Pool
- Submit Problems to Global Pool
- Broadcast Messages
Administrator User Case Narrative

Administrator User:

- Manage all types of Users
- Manage Global Pool
- All abilities of a Course Coordinator
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives
- Data Flow Diagrams
- Requirements Inventory
- Data Design and Dictionary
- Testing
- Student Screens
- Faculty Screens
- Course Coordinator Screens
- Administrator Screens
- What’s Next?
- Questions

12/11/2009
Data Flow Diagrams

Data flow diagrams provide:

- System Decomposition
- Graphical representation of data “flow”
- Graphical representation of data manipulation
- Top down view of the system
Data Flow Diagram Key

Source/Sink  Process  Datastore

Data Flow
Context Diagram

- Student
- Faculty
- Course Coordinator
- Administrator
- Java Compiler
- D1. J.O.L.T. Database
- D2. Temporary Directory
- D3. Source File
Data Flow Diagram: Process 8 Level 1
Manage Users

Course Coordinator

Data Flow Diagram:
8.1 Create Faculty Account
Create Account Request
Create Account Response

8.2 Create Course Coordinator Account
Create Account Request
Create Account Response

8.3 Edit User Account Information
Edit Request
Edit Response

8.4 Unlock User Account
Unlock Request
Unlock Response

8.5 Disable Account
Disable Request
Disable Response
Enable Request
Enable Response

8.6 Enable Account
Unlock Request
Unlock Response
Disable Request
Disable Response
Enable Request
Enable Response

D1 J.O.L.T Database

12/11/2009
Data Flow Diagram: Process 8.1 Level 2
Create Faculty Account

12/11/2009
Create Course Coordinator Account
Data Flow Diagram: Process 8.3 Level 2
Edit User Account Information

12/11/2009
Data Flow Diagram: Process 8.3.1 Level 3
Edit Student Account Information
Data Flow Diagram: Process 8.3.3 Level 3
Edit Course Coordinator Account Information

12/11/2009
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives
- Data Flow Diagrams
- Requirements Inventory
- Data Design and Dictionary
- Testing
- Student Screens
- Faculty Screens
- Course Coordinator Screens
- Administrator Screens
- What’s Next?
- Questions

12/11/2009
General Functional Requirements
Inventory

- Web based
- All major web browsers supported
- Index page with common Authentication display
- All other pages have “Log Out” Functionality
- All users must authenticate
Student Requirements Inventory

- Only Self-Registering user
- Enroll in courses
- View Messages sent to the Student
- View problem sets from enrolled courses
- Solve problems
- Save problem progress
- View Grades and previously submitted solutions
Faculty Requirements Inventory

- Create individual problems and problem sets
- Submit problems to their course pool
- Search the global pool for problems
- View and Modify “Gradebooks” for their courses
- Ability to interact with J.O.L.T. as a “Student” user
Course Coordinator Requirements

Inventory

- Create Faculty accounts
- Assign Faculty to courses
- Create reports and statistics
- Manage their respective course pool
- Submit problems to the global pool
- Manage grades for students of the courses they coordinate
Administrator Requirements Inventory

- Manage all accounts
- Create course Coordinator and Faculty Accounts
- Manage the Global pool
- Broadcast Messages
- Assign courses to course coordinators
- Have all other abilities of a course coordinator
Agenda

• Team Introduction
• Restatement of Problem
• Project Progression
• User Case Narratives
• Data Flow Diagrams
• Requirements Inventory
• Data Design and Dictionary
• Testing
• Student Screens
• Faculty Screens
• Course Coordinator Screens
• Administrator Screens
• What’s Next?
• Questions

12/11/2009
Definition of a Data Dictionary: a "centralized repository of information about data such as meaning, relationships to other data, origin, usage, and format." - IBM Dictionary of Computing
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives
- Data Flow Diagrams
- Requirements Inventory
- Data Design and Dictionary
- Testing
- Student Screens
- Faculty Screens
- Course Coordinator Screens
- Administrator Screens
- What’s Next?
- Questions
Testing

- Overall Testing
- Unit Test
- Test Cases
- Acceptance Test
## Unit Test

<table>
<thead>
<tr>
<th>Test Result</th>
<th>Unit Number</th>
<th>Unit Test Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>F</td>
<td>1</td>
</tr>
<tr>
<td>P</td>
<td>F</td>
<td>2</td>
</tr>
<tr>
<td>P</td>
<td>F</td>
<td>3</td>
</tr>
</tbody>
</table>

- Registration
- Log on
- Answer Problem
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives
- Data Flow Diagrams
- Requirements Inventory
- Data Design and Dictionary
- Testing
- Student Screens
- Faculty Screens
- Course Coordinator Screens
- Administrator Screens
- What’s Next?
- Questions

12/11/2009
Student Screen: Registration

All Fields Required

First Name
Last Name
Email
Confirm Email
Username
Password
Confirm Password
Graduating Year
Security Question
Security Question Answer

Register Account
Welcome (Student Username)

List of Current Courses:
- CSIS-210  Data Structures  Dr. Flatland
- CSIS-225  Object Oriented  Dr. Lim

List of Previous Courses:
- CSIS-110  Data Structures  Dr. Flatland  Fall 2008
- CSIS-120  Object Oriented  Dr. Lim  Spring 2009

Enroll in Course Enter PIN  Submit
Student Screen: Enroll Error

---

Welcome (Student Username)

List of Courses:
- CSIS-210
- CSIS-225

List of Previous Courses:
- CSIS-110 Data Structures Dr. Flatland Fall 2008
- CSIS-120 Object Oriented Dr. Lim Spring 2009

Enroll in Course Enter PIN Submit

---

Incorrect PIN

---

Date: 12/11/2009

Powered By SIS Systems
Copyright 2009
Pending Active Assignments for CSIS-XXX-NN

Sort by: ✸ ✦ ✦ ✸

Assignment Name | Due Date | Grade
---|---|---
Binary Search | 11/17/2010 | –
Recursion | 11/18/2010 | –
String Practice | 11/19/2010 | –
Boolean Practice | 11/19/2010 | –
### Problem Set: String Practice

<table>
<thead>
<tr>
<th>Problem Name</th>
<th>Category</th>
<th>Point Value</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name1</td>
<td>Strings</td>
<td>20</td>
<td>--</td>
</tr>
<tr>
<td>Name2</td>
<td>Recursion</td>
<td>10</td>
<td>--</td>
</tr>
<tr>
<td>Name3</td>
<td>Arrays</td>
<td>10</td>
<td>--</td>
</tr>
<tr>
<td>Name4</td>
<td>Algorithms</td>
<td>20</td>
<td>--</td>
</tr>
<tr>
<td>Name5</td>
<td>Trees</td>
<td>40</td>
<td>--</td>
</tr>
</tbody>
</table>

**Total:** 100

*You can click on any problem's name to jump right to that problem*
Problem Set Name: Problem Set 1
Problem Name: Problem Name 1

Problem Description:
For problem 1 of this problem set you must code an iterative binary search algorithm. The method takes 2 parameters the first being a sorted array of integers containing the data, the second parameter is the number I want you to find.

Source Code:

```
public int findMe (int [] theData, int searchFor)
{
  // Insert Your Code Here
}
```

Compile, Save, Run
Problem Set Name: Problem Set 1
Problem Name: Problem Name 1
Problem Description:
For problem 1 of this problem set you must code an iterative binary search algorithm. The method takes 2 parameters the first being a sorted array of integers containing the data, the second parameter is the number I want you to find. If you can find the number return true, if you can't return false.

Source Code:

public int findMe(int[] theData, int searchFor)
{
    return n
}
Problem Set Name: Problem Set 1
Problem Name: Problem Name 1
Problem Description:
For problem 1 of this problem set you must code an iterative binary search algorithm. The method takes 2 parameters the first being a sorted array of integers containing the data, the second parameter is the number I want you to find. If you can find the number return true, if you can't return false.

Source Code:

```java
public int findMe (int [] theData, int searchFor)
{
    //Insert Your Code Here
}
```

Test Cases:

<table>
<thead>
<tr>
<th>Case</th>
<th>Expected</th>
<th>Results</th>
<th>Pass/Fail</th>
<th>Passed In</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>True</td>
<td>True</td>
<td>Pass</td>
<td>findMe (theData, 5)</td>
</tr>
<tr>
<td>#2</td>
<td>True</td>
<td>True</td>
<td>Pass</td>
<td>findMe (theData, 6)</td>
</tr>
<tr>
<td>#3</td>
<td>False</td>
<td>False</td>
<td>Pass</td>
<td>findMe (theData, 21)</td>
</tr>
<tr>
<td>#4</td>
<td>True</td>
<td>False</td>
<td>Fail</td>
<td>findMe (theData, 28)</td>
</tr>
<tr>
<td>#5</td>
<td>False</td>
<td>True</td>
<td>Fail</td>
<td>findMe (theData, 100)</td>
</tr>
</tbody>
</table>
Problem Set Name: Problem Set 1
Problem Name: Problem Name 1

Problem Description:
For problem 1 of this problem set you must code an iterative binary search algorithm. The method takes 2 parameters the first being a sorted array of integers containing the data, the second parameter is the number 1 want you to find. If you can find the number return true, if you can't return false.

Test Cases:

<table>
<thead>
<tr>
<th>Case</th>
<th>Expected</th>
<th>Results</th>
<th>Pass/Fail</th>
<th>Passed In</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>True</td>
<td>Timeout</td>
<td>Fail</td>
<td>findMe (theData, 12)</td>
</tr>
<tr>
<td>#2</td>
<td>True</td>
<td>Timeout</td>
<td>Fail</td>
<td>findMe (theData, 24)</td>
</tr>
<tr>
<td>#3</td>
<td>False</td>
<td>Timeout</td>
<td>Fail</td>
<td>findMe (theData, 167)</td>
</tr>
<tr>
<td>#4</td>
<td>True</td>
<td>Timeout</td>
<td>Fail</td>
<td>findMe (theData, 36)</td>
</tr>
<tr>
<td>#5</td>
<td>False</td>
<td>Timeout</td>
<td>Fail</td>
<td>findMe (theData, 100)</td>
</tr>
</tbody>
</table>
Problem Description:
For problem 1 of this problem set you must code an iterative binary search algorithm. The method takes 2 parameters the first being a sorted array of integers containing the data, the second parameter is the number 1 want you to find. If you can find the number return true, if you can’t return false.

You Have Passed All Test Cases!

Test Cases:

<table>
<thead>
<tr>
<th>Case</th>
<th>Expected</th>
<th>Results</th>
<th>Pass/Fail</th>
<th>Passed In</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>True</td>
<td>True</td>
<td>Pass</td>
<td>findMe (theData, 5)</td>
</tr>
<tr>
<td>#2</td>
<td>True</td>
<td>True</td>
<td>Pass</td>
<td>findMe (theData, 6)</td>
</tr>
<tr>
<td>#3</td>
<td>False</td>
<td>False</td>
<td>Pass</td>
<td>findMe (theData, 21)</td>
</tr>
<tr>
<td>#4</td>
<td>True</td>
<td>True</td>
<td>Pass</td>
<td>findMe (theData, 6)</td>
</tr>
<tr>
<td>#5</td>
<td>True</td>
<td>True</td>
<td>Pass</td>
<td>findMe (theData, 6)</td>
</tr>
<tr>
<td>#6</td>
<td>XXX</td>
<td>XXX</td>
<td>Pass</td>
<td>findMe (XXXXXXXX)</td>
</tr>
</tbody>
</table>

Italics Denote Hidden Test Cases

Return to Problem Set

Compile, Save, Run
**Student Screen: Section Gradebook**

### Student Gradebook for CSIS-XXX-NN

<table>
<thead>
<tr>
<th>Assignment Name</th>
<th>Due Date</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem Set 1</td>
<td>11/17/2010</td>
<td>100</td>
</tr>
<tr>
<td>Recursion</td>
<td>11/18/2010</td>
<td>--</td>
</tr>
<tr>
<td>String Practice</td>
<td>11/19/2010</td>
<td>--</td>
</tr>
<tr>
<td>Iteration</td>
<td>11/20/2010</td>
<td>--</td>
</tr>
</tbody>
</table>

Average: 100
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives
- Data Flow Diagrams
- Requirements Inventory
- Data Design and Dictionary
- Testing
- Student Screens
  - Faculty Screens
  - Course Coordinator Screens
  - Administrator Screens
- What’s Next?
- Questions

12/11/2009
Welcome (Faculty Username)

List of Current Courses:
- CSIS-210  Data Structures  Section-05
- CSIS-225  Object Oriented  Section-04

List of Previous Courses:
- CSIS-110  Data Structures  Section-03  Fall 2008
- CSIS-120  Object Oriented  Section-08  Spring 2009
Create/Edit Problem

Problem Information
Title: String Practice
Category: Practice
Method Name: stringLength
Description: Return the length of the string provided.

Parameters
Name: param1
Type: String
Result: int

Test Cases
Test Case 1: "Test"
Test Case 2: "Test2"
Test Case 3: 
Test Case 4: 
Test Case 5: 

Solution Code:
```java
public int stringLength(String param1)
{
    return param1.length();
}
```
Gradebook for CSIS-XXX

<table>
<thead>
<tr>
<th>Student Name</th>
<th>Assignment Name</th>
<th>Due Date</th>
<th>Grade</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jon Doe</td>
<td>Binary Search</td>
<td>11/17/2010</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>Jane Smith</td>
<td>Recursion</td>
<td>11/18/2010</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>Jesus Christ</td>
<td>String Practice</td>
<td>11/19/2010</td>
<td>90</td>
<td></td>
</tr>
</tbody>
</table>

High: 90
Average: 85
Low: 80
Edit Profile

All fields are required

First Name       Jon
Last Name        Smith
Email            ja28smit@siena.edu
Confirm Email    ja28smit@siena.edu
Password         ●●●●●●
Confirm Password ●●●●●●
Security Question First Pet Name
Security Question Answer ●●●●●●

Save Changes
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives
- Data Flow Diagrams
- Requirements Inventory
- Data Design and Dictionary
- Testing
- Student Screens
- Faculty Screens
  - Course Coordinator Screens
  - Administrator Screens
- What’s Next?
- Questions

12/11/2009
Course Coordinator Screen: Log In

J.O.L.T.
JAVA ONLINE LEARNING TUTOR

Log in
Username
Password
Log in
Reset Password

12/11/2009
Create Announcement

Title
New Problem Set Activated

Announcement
New Problem Set Activated for students of CSIS-120 (all sections)

Recipients
- CSIS-120-05
- CSIS-120-04
- CSIS-225-03
- Lim, Darren (Administrator)
- Flatland, Robin (Faculty)
- Egan, Mary Anne (CC)

List
- All Students
- CSIS-120-05
- CSIS-120-04
- CSIS-225-03
- Albert, Joseph
- Bernard, Andrew
- Bosc, Christina
- Brennan, Julie
- Colbert, Stephen
- Doe, John
- Sanders, Stephen
- Schrute, Dwight
- Smith, John
- Valenti, Angela
- Zabinski, Joe

Post Announcement
Your announcement has been successfully sent to:

CSIS-120-05
CSIS-120-04
CSIS-225-03
Darren Lim (Administrator)
Robin Flatland (Faculty)
Mary Anne Egan (CC)

Return to Course Home  Return to Course Select
Edit Course Section For CSIS-XXX

Semester: Fall 2009
Number: 09
Time: 8:15-9:10 AM
Days: MWF

Update Section
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives
- Data Flow Diagrams
- Requirements Inventory
- Data Design and Dictionary
- Testing
- Student Screens
- Faculty Screens
- Course Coordinator Screens
- Administrator Screens
- What’s Next?
- Questions

12/11/2009
## Administrator Screen: Manage Students

### Manage Students

<table>
<thead>
<tr>
<th>Name</th>
<th>Select</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ernesto Roldan</td>
<td></td>
</tr>
<tr>
<td>Stephen Sanders</td>
<td></td>
</tr>
<tr>
<td>Lucas Ferrara</td>
<td></td>
</tr>
<tr>
<td>Alex Wehren</td>
<td></td>
</tr>
<tr>
<td>Joe Rao</td>
<td></td>
</tr>
<tr>
<td>Lisa Cuddy</td>
<td></td>
</tr>
<tr>
<td>Gregory Haus</td>
<td></td>
</tr>
<tr>
<td>Stephen Colbert</td>
<td></td>
</tr>
<tr>
<td>Dwight Schrute</td>
<td></td>
</tr>
</tbody>
</table>

- **Edit Selected**
- **Add New Student**
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives
- Data Flow Diagrams
- Requirements Inventory
- Data Design and Dictionary
- Testing
- Student Screens
- Faculty Screens
- Course Coordinator Screens
- Administrator Screens
- What’s Next?
- Questions
What's Next

- Software Plan
- Requirements Specification
- Preliminary Design
- Detailed Design
- Acceptance Test
Timeline This Semester

Team: 518 Interactive
Project Title: J.O.L.T.
Date: 12/6/2009
# Timeline Next Semester

## Task List

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Start</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Team Meetings</td>
<td>Fri 1/22/10</td>
</tr>
<tr>
<td>26</td>
<td>Client Meetings</td>
<td>Fri 1/22/10</td>
</tr>
<tr>
<td>51</td>
<td>Detailed Design</td>
<td>Wed 1/20/10</td>
</tr>
<tr>
<td>52</td>
<td>Detailed Design Presentation</td>
<td>Wed 3/10/10</td>
</tr>
<tr>
<td>53</td>
<td>Acceptance Test</td>
<td>Thu 3/11/10</td>
</tr>
<tr>
<td>54</td>
<td>Acceptance Test Presentation</td>
<td>Fri 4/16/10</td>
</tr>
</tbody>
</table>

---

**Project:** Java Online Learning Toolkit  
**Team:** 518 Interactive  
**Date:** Sun 12/6/09  

---

**Page 1**
Questions?

Thank You For Coming