Software Requirements Specification
Presentation
Client Dr. Darren Lim

Friday October 30, 2009
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 and Use Case Diagram
- Data Flow Diagrams
- Activity Diagrams
- Requirements Inventory
- Prototype 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

Software Plan

Requirements Specification

Preliminary Design

Detailed Design

Acceptance Test

Development and Testing

Friday October 30, 2009
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives and Use Case Diagram
- Data Flow Diagrams
- Activity Diagrams
- Requirements Inventory
- Prototype 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
- View 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
UML Use Case Diagram Key
UML Use Case Diagram

Student
Submit Java Solutions
Register
Create Java Problems
View Problems

Faculty
View/Modify Grade Book
Login/Logout

Course Coordinator
Assign Faculty to Courses
Create/Manage Faculty Accounts

System Administrator
Create/Manage All Accounts
Manage Course Pool of Problems
Manage Global Pool of Problems

Java SDK
Database
Web Server
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives and Use Case Diagram
- Data Flow Diagrams
- Activity Diagrams
- Requirements Inventory
- Prototype Screens
- What’s Next
- Questions
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

D | Datastore

Data Flow
Context Diagram

- Student
- Faculty
- Course Coordinator
- Administrator
- Java Compiler

J.O.L.T.

D1. J.O.L.T. Database
D2. Temporary Directory
D3. Source File
Data Flow Diagram: Level 1
Student Solve Problem

Student

Input Response  Input Request

1.1
Process Source Code

Completed Class File  Compilation Result

Java Compiler

Source Information

Compilation Result  Test Results

D1. J.O.L.T. Database

Compiled Java Byte Code  Compiled Java Byte Code

D2. Temporary Directory

Test Results

1.2
Test Compiled Source Code

Source Code

D3. Source File

Friday October 30, 2009
Data Flow Diagram: Level 1
Faculty Create Problem

Friday October 30, 2009
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives and Use Case Diagram
- Data Flow Diagrams
- Activity Diagrams
- Requirements Inventory
- Prototype Screens
- What’s Next
- Questions
Activity Diagrams Provide:

- Visible Stepwise Progression
- Shows the overall flow of control
UML Activity Diagram Key

- Activity Description
- Activity
- Flow
- Join
- Initial Node
- Final Node
- Time Event
- Decision
UML Activity Diagram: Student Registration

1. Ask the system to register a new student
2. Student enters their information
3. Student verifies information
4. [Incorrect Information] \( \lor \) [Correct Information]
   - System rejects new user creation
   - Ask student to re-enter information
   - System creates new user
   - System sends email confirmation
UML Activity Diagram: Student Solve Problem

- Student selects problem
- Student views problem
- Student enters problem attempt
  - Student is presented with feedback
    - [Code does not compile]
    - [Code Compiles]
  - System compiles attempt
    - System tests attempt against faculty provided test cases
      - [Any test cases incorrect]
      - [All test cases correct]
        - Student is presented with feedback
        - Student is presented with feedback
          - [Student attempts problem again]
          - [Skip problem]
          - Problem score is added to Gradebook

Friday October 30, 2009
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives and Use Case Diagram
- Data Flow Diagrams
- Activity Diagrams
- Requirements Inventory
- Prototype Screens
- What’s Next
- Questions
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 and Use Case Diagram
- Data Flow Diagrams
- Activity Diagrams
- Requirements Inventory
- Prototype Screens
- What’s Next
- Questions
Prototypes

Problem Description
Write a method to return the number passed in as a parameter.

Java® Solution

```java
int function(int n)
{
    return n;
}
```

Correct Program Submission!
All Test Cases Passed!
Prototypes

Friday
October 30, 2009
Prototypes

Friday
October 30, 2009
Agenda

- Team Introduction
- Restatement of Problem
- Project Progression
- User Case Narratives and Use Case Diagram
- Data Flow Diagrams
- Activity Diagrams
- Requirements Inventory
- Prototype Screens
- What’s Next
- Questions
What's next

- Software Plan
- Requirements Specification
- Preliminary Design
- Detailed Design
- Acceptance Test
### Timeline (Gantt Chart)

<table>
<thead>
<tr>
<th>ID</th>
<th>Task Name</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Team Meetings</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Client Meetings</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>Software Plan</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>Software Plan Presentation</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>Requirements Specification</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>Requirements Specification Present</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>Preliminary Design</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>Preliminary Design Presentation</td>
<td></td>
</tr>
</tbody>
</table>

**Team:** 518 Interactive  
**Project Title:** JOLT  
**Date:** 10/28/2009  
**Page 1**
Questions?

Thank You For Coming

Friday October 30, 2009