

Integrative Exercise II Winter 2005

The purpose of this capstone integrative exercise is to give you an opportunity to develop a complete information systems solution, which draws upon the skills and knowledge that you have developed in both the Fall ISys Core and the Winter ISys Core. Since the INTEX I that was done last semester focused very heavily on the Systems Analysis and Design and Database classes, INTEX II will emphasize the skills and concepts you learned in the programming classes (ISys 403 & 413), the systems design class (ISys 411), and the networking and data communications class (ISys 412). Obviously, it is difficult to emphasize all aspects of the ISys Core equally. However, overall INTEX II should provide a comprehensive review and solidification of the most of the concepts that are included in the core.

INTEX II will be based on the case that you worked on last Fall, Two Tickets Plus. (However, we have simplified the system to include only a subset of the total use cases required for a complete system.) You will need to set up a complete “data center” for your company. This will include the complete operating environment for both development and production. You will also be required to complete the design of TT+, develop the test plan, write, test, and implement the Java system for enterprise level processing. The activity is expected to require the final six weeks of the Winter ISys Core. The workload in ISys 411 will be lightened substantially so that you can have sufficient time for INTEX II. ISys 412 will continue to cover critical topics in networking and data communication as well as a particular focus on skills necessary for your project. ISys 413 will focus primarily on the skills and topics you will need for the successful completion of your project.

Each week there are specific activities and deliverables that should be completed. The project will commence on Monday February 28, 2005, and should be completed by Saturday April 9, 2005. You should work on and finish the tasks as described in the following table. More detailed description of what is required is provided in subsequent pages. Each task (itemized below) should be signed off by a TA on the Monday (by 5:00 p.m) after the final week. For example task 1, Installation should be signed off by a TA on Monday Mar 14. We have tried to even out the workload, however, some weeks include deliverables that will require substantial work. You should plan your work by looking ahead at work to be completed in the future and schedule your time appropriately.

Task	Feb 28	Mar 7	Mar 14	Mar 21	Mar 28	Apr 4
1	Installation	Installation				
2	Systems Design	Systems Design				
3		Administrative System	Administrative System	Administrative System		
4			Customer System	Customer System	Customer System	
5				Sell Foreign Tickets	Sell Foreign Tickets	
6						Acceptance Test

The entire project will be graded at the end of the semester. Individual milestones will be checked off by the TAs or professors at the end of each phase. Failure to finish a phase by its milestone date will reduce your overall score by 1 percent per week (or fraction thereof).

Scoring for each phase is as follows:

Task	Activities	Weight
Installation Start: February 28 Due: March 12	<ul style="list-style-type: none"> • Set up company server. • Set up Remote Desktop for remote access • Configure your hardware router • Install IIS, Tomcat, and other required servers. 	10%
Systems Design Start: February 28 Due: March 12	<ul style="list-style-type: none"> • Database design and implementation • System Design of business layer • Comprehensive test plan 	20%
Administrative System Start: March 7 Due: March 26	<ul style="list-style-type: none"> • Implementation of business layer classes and DAO classes with methods for the specified use cases. Exported business layer objects via RMI. • Desktop system for administrative functions and testing of business layer and DAO layer. 	20%
Customer System Start: March 14 Due: April 2	<ul style="list-style-type: none"> • Implementation of the business layer classes and DAO classes with methods for the specified use cases. Exported business layer objects via RMI. • Web based system to allow users to connect to the system and carry out “customer functions” – simplified version 	20%
Sell Foreign Tickets Start: March 21 Due: April 2	<ul style="list-style-type: none"> • Credit card verification via Web Services. • Allow other venues to sell tickets. Sell tickets in behalf of other venues. • Implement data exporting subsystem 	20%
Acceptance Test Start: April 4 Due: April 9	<ul style="list-style-type: none"> • System Test and Acceptance Test 	10%

Task: Installation

Objective: Set up your company computing environment, including hardware, operating system, utilities, servers, network, etc.

Detail Requirements:

Your group will be given a computer in the networking lab for the next couple of months. You are being given an empty computer, so you need to install all required software. For conformity in grading and TA help, please use the following technologies. While some people may have experience with competing technologies, please do not install Linux or other software.

- Windows Server 2003
- Remote Desktop
- Hardware firewall and router (you will be given a router to use)
- Antivirus software
- Microsoft SQL Server
- Microsoft JDBC Driver
- Microsoft IIS web server
- Java SDK (full development kit, not runtime)
- Tomcat
- Tomcat connector for IIS
- Struts

Because of limited room in the lab, set up Remote Desktop so you can setup your server from offsite.

Test your configuration by creating a simple JSP that connects to your database via JDBC, increments a counter field using an UPDATE statement, and displays the current value of the counter. Note that this is a simple test of your configuration – you do not need a business layer! Simply open and close the connection directly within your JSP. You can hard code the connection parameters for this test.

Task: Systems Design

Objectives:

1. Design and implement the database. Using the conceptual data model provided in Appendix A, design the set of relational tables and create the database schema in SQLServer.
2. Design the detail processing for the identified use cases (two-layer sequence diagrams).
3. Develop a system test and acceptance test plan with test data.

Detail Requirements:

Appendix A gives a conceptual data model. Note that this is the guideline for your development of this project. As with any project, you may have to make minor adjustments to this design at implementation to meet needs that arise. You may find additional fields or tables that may be required. With any adjustments, please keep with the spirit and intent of the provided documentation.

Appendix B gives a use case diagram identifying the use cases to be included in the system. Detail descriptions for each use case are also included.

For this phase of the project, finish the design documents and schema for your database. Implement the database tables in SQL Server. The deliverable for this task are the finished documents and implemented database tables.

Task: Administrative System

Objective: Develop and execute the Administrative function use cases.

Detail Requirements:

The Administrative functions are those use cases identified in the Administrative Subsystem of Appendix B, Use Case Diagram. You are to program the business layer classes, the data access classes, and the view layer classes to support those use cases. These are desktop functions and should be supported with appropriate windows forms. Please use the Swing toolkit to create a client-side application for your view.

Program your business layer using the patterns studied in 413 this semester. Your factory, data access objects, and business objects will run on your server. Your administrative client will run on your laptops. Use Remote Method Invocation (RMI) so the client view can access the server-based business objects. This means your Factory, DAOs, and BOs will all be exported objects.

Create a group CVS repository for your project on mach.byu.edu. You should use CVS throughout this entire project.

Task: Customer System

Objective: Develop and execute the Sales Subsystem function use cases.

Detail Requirements:

The sales subsystem is a web based system used by Customers. You are to program the business layer classes, the data access layer classes, and the view layer classes (using Struts) to support those use cases.

The Tomcat engine (with Struts) will run the view layer in this task, similar to what we discussed in 413. Your business layer will run in a separate JVM on the same machine (your server). As with the administrative client task, use RMI to access the business layer in the separate JVM.

You should conduct as much programmer level testing as necessary and time allows. You may also include your test cases from your testing plan.

Note you should maintain change control procedures to ensure that you do not destroy previous work.

Task: Sell Foreign Tickets

Objective: Enhance your sales system as necessary to allow foreign venues to sell tickets. Be able to connect to foreign systems to sell their tickets. Develop and Execute Reporting Subsystem functions use cases.

Detail Requirements:

In this week you should be able to connect to foreign systems. First you should implement your credit card validation functions by connecting to a “bank” to validate credit cards. We will provide the bank that can be accessed via Web Services calls.

You should also be able to connect to a foreign venue to sell tickets from them. A foreign venue should be able to connect to your system to sell your tickets. We will provide a standard API that all venues in the coop will use. Foreign events and seats should show up in your system side by side with your local events. Sell them as you would any other event. The primary difference between foreign and local sales is the commission paid to the selling system. Your system should maintain sufficient detail to be able to produce the Venue Commission Report.

Your team should implement the Export Data Subsystem functions use cases. (Two XML data files.) Note that the data to be exported must be sufficient for a foreign system to analyze company performance and produce any necessary reports. Design the XML files so that they are comprehensive and provide sufficient detail.

Maintain change control procedures to ensure you do not destroy previous work.

Task Week 6: Acceptance Test

Objective: Final System Test and Acceptance Test

Detail Requirements:

This week is clean up week. You should complete testing and debugging that is necessary to ensure your system works well. You should conduct and document the necessary acceptance test as per your test plan.

Any “bells and whistles” that you would like to add can be inserted AND tested. Maintain CVS and change control procedures so that can back out any that are not working.