

CS614B – Advanced Topics in Software Quality
Assignment 3 – Group Project - Winter 2006
Due April 6th

This project requires the formation of two teams of four, development model selection, program design, program testing, a paper submission and a class presentation on April 6th.

Your paper submission should be **fifteen to twenty pages (including tables, figures and references)**, Times fonts (11 or 12 point), one inch margins, single-spaced, single column, and printed both sides. Include the team members' names, universities and ID numbers at the top of the first page. You should also submit the code for your program, the test cases and their output.

Read the entire assignment before your team starts. Submit in paper copy, April 6th, in class. Email electronic copies to mgittens@ca.ibm.com by 12noon April 6th.

Learning Objectives

At the end of this project, you will have gained experience in the following:

1. In choosing, justifying, implementing, and evaluating a development model based on the characteristics of the project.
2. In testing software at the three primary levels with three specific approaches, with a focus on creating an environment that is representative of the customers for reliability testing.

Requirements

Your team has been commissioned to create a prototype for a new “bricks and mortar” bookstore. The company wants to manage information for their customers, employees, books, book orders, publishers and invoices. This prototype is a custom database system made up of arrays. Each entity and relationship is represented by a separate array. This database system has a number of functions to manage the data and relationships between the entities.

Although this bookstore is new, they have previously been operating via a paper-based “book-mobile” model, so they have a large customer base, and expect it to grow rapidly once their bookstore sets up a bricks and mortar base. This means that the custom database system must handle a large volume of data for at least 10,000 customers.

This is a case where the prototype will be used by the company initially, before the final system is developed. The client wants the system to observe current company workflows before committing to a larger more expensive system. This means that your team must thoroughly test the prototype system.

The arrays in the database system must hold the following information for all entities and relationships for the bookstore:

- **Employee**
 - EmployeeID – string: This uniquely identifies each employee.
 - Range: E0001 to E9999 (cannot be empty)
 - Name – string: This is a combination of the first and last name of the employee.

- NumberAvailable – integer: This is the quantity of the book available in-store
Range: 0 to 20
- PublisherID – integer: This uniquely identifies the publisher.
 - Range: “P0001” to “P9999” (cannot be empty)
- **Invoice**
 - InvoiceID – integer: This uniquely identifies each invoice.
 - Range: “I0001” to “I9999” (cannot be empty)
 - EmployeeID – integer: This identifies the employee entering the order
 - Range: E0001 to E9999
 - CustomerID – integer: This identifies the customer placing the order
 - Range: C0001 to C9999
 - Date – string: This maintains the date when the order was made.
 - No dates earlier than 2000/01/31 allowed
 - Sum – real: This is the total cost of the order that is recorded on the invoice.
 - Range: \$1.00 to \$1,000,000.00
 - IsRefund – boolean: Is this an invoice for a refunded order? Values are ‘1’ or ‘0’
- **Customer-Invoice-Book Relationship**
 - InvoiceID – integer: This identifies the invoice involved in the relationship.
 - Range: “I0001” to “I9999” (cannot be empty)
 - BookID – string: This identifies the book recorded by the invoice.
 - Range: “B0001” to “B9999” (cannot be empty)
 - CustomerID – string: This customer triggering the invoice.
 - Range: C0001 to C9999
 - NumberRefund – integer: This is the quantity of the given book refunded.
 - Range: 0 to 20
 - OrderNumber – integer: This is the quantity of the book ordered.
 - Range: 0 to 20
- **Publisher**
 - PublisherID – integer: This uniquely identifies the publisher.
 - Range: “P0001” to “P9999” (cannot be empty)
 - Name – string: This is the name of the publisher.
 - Address – string: This is the address of the publisher.
 - Phone – string: This is the telephone number of the publisher.
 - Format is 000-000-0000

There are a number of interactions between these entities. There are however only a few major interactions that your team must focus on. (Do not focus on building extensive interfaces for populating the database or other management issues. Use scripts for array population. Provide minimal interfaces to the end user for placing orders and performing the following scenarios. Text interfaces are satisfactory.)

1. An employee orders a book for a customer

- If the customer is not already in the array, the customer is placed into the customer array.
- The program checks the Book array to see how many books are in the store.
- If there are no books in the store, an order is made for the book.

- An entry is made into the BookOrder array with the customer, employee, book information, and the date of the order.
- A BookOrderID is assigned to the order. This is entered into the BookOrder array, with the EmployeeID, CustomerID, PublisherID, the OrderNumber and Date for the order.
- With the entry of the PublisherID, the publisher is contacted with the standard Order for the book.
- While the order is outstanding, it is marked as pending.
- When the order is received, pending is marked as false and this triggers a purchase.

2. A customer purchases/returns a book

- If it is a purchase and if the customer is not already in the array, the customer is placed into the customer array. (A customer who is not in the array cannot return books.)
- An invoice is created and entries are made into the Invoice array for InvoiceID, the employee (ID) selling the book, the customer (ID) buying the book, the total purchase/return price of the book, and a record of whether the invoice is for a refund/return.
- A record is kept of the transactions by updating the Customer-Invoice-Book array. Entries must be made in the array for the InvoiceID, BookID, CustomerID, NumberRefund, and Number.

3. An employee does a search for information on a book

- The number of a given book in stock.
- The number of a given book on order.
- The number of a given book sold and the invoice numbers for those sales.

Your team has five main tasks before you. You may use any common programming/scripting language to implement your solution. These solutions (including, algorithms, discussions, and code extracts) to these tasks are to be presented in a paper. Submit program code separately.

2. There are three main levels/targets of testing for this project. They are unit testing, integration testing, and system testing [1].
 - i. Unit testing verifies the functioning in isolation of software pieces that are separately testable. This level of testing will reveal coding and functional errors. You are to conduct Boundary Value Analysis (BVA) [2] to test your code.
 - Generate your own goals for BVA testing.
 - Determine when your goals have been met.
 - Assess the adequacy of your test cases for meeting these goals.
 - ii. Integration testing is the process of verifying the interaction between software components. Classical integration testing strategies are top-down or bottom-up. This level of testing will reveal functional errors. You are to conduct functional testing, that is, ensure that each function performs its assigned task.
 - Your goals for testing are to test each function of the program.
 - Determine when your goals have been met.
 - Assess the adequacy of your test cases for meeting these goals.

- iii. System testing is concerned with the behavior of a whole system. System testing is appropriate for comparing the system to the non-functional system requirements, such as security, speed, accuracy, and reliability. You are to focus on reliability.
 - o Your goals for testing are to run the program without having the system terminate unexpectedly.
 - i. Purchase books for five new and five existing customers
 - ii. Order books for five new customers.
 - iii. Conduct an employee search to find how many of ten distinct books are in stock.
 - iv. Conduct an employee search to find how many of ten distinct books are on order.
 - v. Conduct an employee search to find the number of ten distinct books that were sold and the invoice numbers for those sales.
 - o Determine when your goals have been met.
 - o Assess the adequacy of your test cases for meeting these goals.
2. This is an important step. Before conducting your system testing, you must create an environment similar to your client's environment.
- a. They will run the program on a Windows 9x/XP or UNIX environment. They would prefer a UNIX system, but will purchase a system according to your advice. Therefore, machines accessible to you will suffice for testing.
 - b. The records from the paper-based "book-mobile" business were previously held in a "shoe-box repository", and are therefore difficult to parse, so the only information that we have on the nature of the data is given in the entity specifications. *In essence, the data that is to be used to test the database is missing.*

You must therefore generate your own data. This data must be based on the data specifications. You must then pre-populate the arrays with sample data for 1000 customers, 100 employees, 5000 books, 100 publishers, 1000 invoices, and 100 book orders. This pre-population must also populate the arrays that hold the relationships between the primary arrays, and use the specified data ranges.
 - c. The customers historically conduct the following ratio of operations.
 - i. Purchase books – 50% of operations
 - ii. Order books – 10% of operations
 - iii. Conduct an employee search to find how many of a book are in stock – 15% of operations
 - iv. Conduct an employee search to find how many of a book are on order – 15% of operations
 - v. Conduct an employee search to find how many of a book were sold and the invoice numbers for those sales – 10% of operations
3. Identify the software development process model that is most suitable to your team and this task, and discuss the relevance of this process model to your project, based on experiences of others, as documented in two or more papers.

4. Develop the program described while following the chosen process model. Discuss the steps taken throughout the development phases, and explain how the process helped or hindered your efforts.
5. Discuss how the process followed will facilitate adaptive, corrective, perfective or preventive maintenance of your solution. You should consider the intention to enhance this prototype to create a permanent bookstore management facility for the client.

Properly cite any references in ACM conference format and include a list of references at the end of your paper. The assignment references can be used as an example, and are given below.

REFERENCES

- [1] The Industrial Advisory Board of the Guide to the Software Engineering Body of Knowledge (SWEBOK) - Institute of Electrical and Electronics Engineers, Inc. *Software Engineering Body of Knowledge (SWEBOK)*, 2001. www.swebok.org - current February 2006.
- [2] Adrion, W.R. , Branstad, M.A. , Cherniavsky J.C. Validation, Verification, and Testing of Computer Software. *ACM Computing Surveys (CSUR)*, Volume 14 Issue 2 (June 1982) . ACM Press.

Marking Scheme

The paper: 35%

- 30% Content
 - Introduction – 2%
 - Discussion: process model, testing goals, and testing assessment – 25%
 - Conclusions – 3%
- 5% Format
 - Grammar, spelling, typing, overall layout of the paper – 2%
 - Extra effort (additional research, graphics, etc. if appropriate) - 3%

The presentation: 25%

You are also required to give a 60-minute presentation about the material presented in the paper. All members of the group must participate in the presentation. Discuss each of the five tasks that are presented in the paper and the program.

1. Clarity – 5%
2. Quality of Presentation Materials – 5%
3. Professionalism, Communication and Group Interaction – 5%
4. Coverage of paper and the program materials – 5%
5. Questions – 5%

The program: 35%

Submit your program, and the underlying algorithm, which can be written in natural language.

1. Documentation - 5%
2. Implementation of the functionality – 15%
3. Test Output – 15%

Team peer evaluation: 5%