King Saud University
The College of Computer & Information Science
Computer Science Department (Master)
Software Engineering (CSC 541)

EasyLib

Library System

Technical Documentation

Presented To:

Dr. Ghazy Assassa

Prepared By:

Amani AL-Ajlan
(427220111)
Eidah Assidan
(427220120)
Nuha Al-Ojayan
(427220109)
Asma Al-Saleh
(427220094)
Manal Al-Bahlal
(427220097)
Sumayah Al-Rwais
(427220113)

Table of Contents

Table of Figures

Table of Tables

Introduction

A library automation system is a computer system designed to automate tasks in a library.

Such systems primarily host a catalog of the library's collection, replacing the old card catalog once used. Under the card catalog system, several cards may be prepared for each book, to be used in a sorted series of cards by Title, Author and Subject.

Automation of the catalog saves the labor involved in resorting the card catalog, keeping it up-to-date with respect to the collection, etc.

Other tasks automated include checking out and checking in books, generating statistics and reports, acquisitions and subscriptions, indexing journal articles and linking to them, as well as tracking interlibrary loans[10].

EasyLib is a web based library system that will be used by Librarians in order to perform an overall management of books and their copies. Using the EasyLib library system, librarians can, also, perform books borrowing and returning for members.

Chapter 1

Planning

1.1 XP Planning Table:

Actors	#	Story name	Priority	Risk	Est Man-hr (For 1 prog)	Prerequisite Stories	Essential	Included
Any user	1	Log in	LOW	LOW	27	3	Y	N
Any user	2	Log out	LOW	LOW	14	1	N	N
Librarian member administrator	3	Add user Note: user role can be member or librarian	Medium	LOW	40	1	Y	N
Librarian, member administrator	4	Delete user	LOW	LOW	27	1,3	N	N
Librarian member administrator	5	Update user	LOW	LOW	40	1,3		N
Librarian	6	Add Book	High	LOW	40	1	Y	Y
Librarian	7	Delete Book	LOW	LOW	27	1,6	N	Y
Librarian	8	Update Book	LOW	LOW	27	1,6	N	Y
Any user	9	Show a certain book's detailed information	High	LOW	40	6,10	N	N
Any user	10	Search books Note : search criteria can be one or a combination of the following: Author Name, Publisher Name, ISDN, Book Title, Book Subject, Book Year	Medium	LOW	40	6	Y	N
Librarian member	11	Borrow book	High	LOW	40	1,6,3,14	Y	Y
Librarian member	12	Return book	High	LOW	40	1,11	N	Y
Librarian	13	Display a certain member's borrowing history	LOW	LOW	40	1,11	N	N
Librarian	14	Get a member’s fine	Medium	LOW	27	1,11	Y	Y

Table 1: XP Planning

1.2Release Planning Table:

One week (one developer) = Man-Hr "12"				
Release #	Release Time Frame ~ 100 hours	Use case/ XP Story	Est Story Man-hr	Pair#
R1	3 weeks	6 11 14	40 40 27	X Y Z
R2	3 weeks	7 8 12	27 27 40	Y X Z

Table 2: Release Planning

Chapter 2

Analysis

2.1 Use Cases Model:

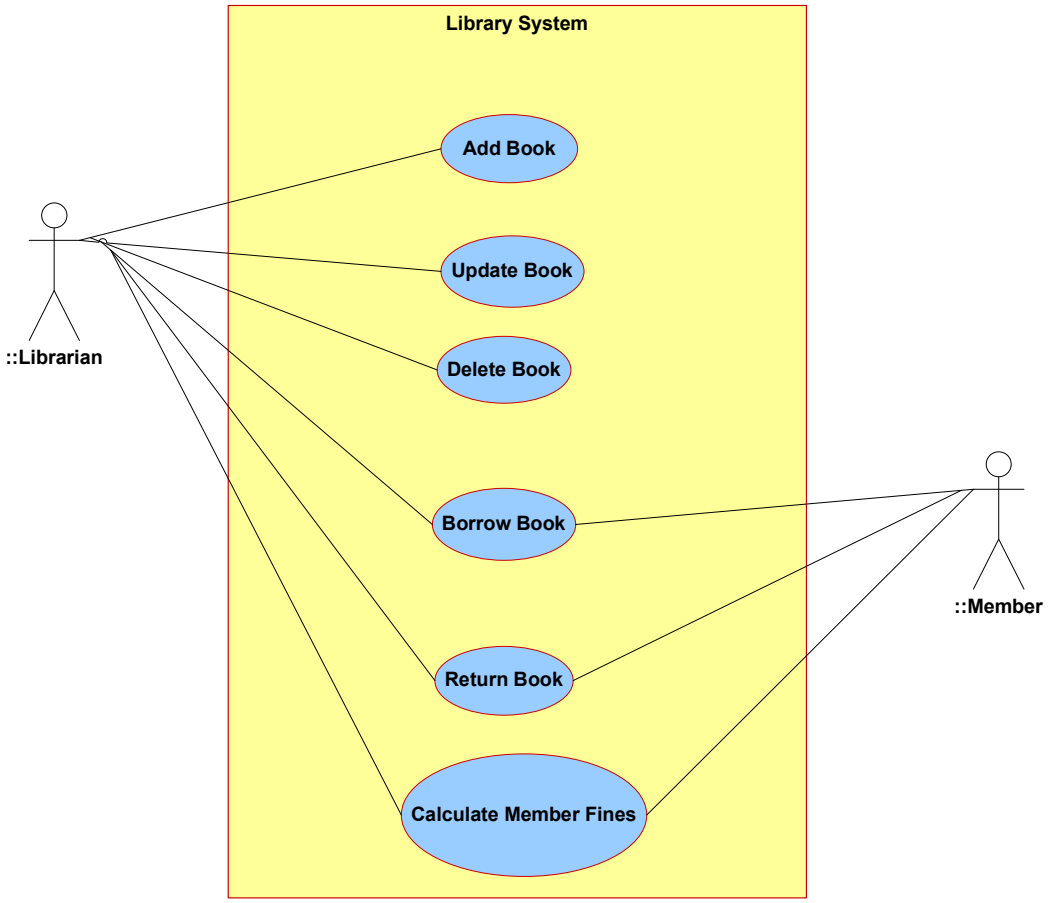


Figure1: Use Case Model

2.2Use Cases:

2.2.1Add Book:

Project Name:	Library System.
Release Number:	1
Use Case Name:	Add book
Unique Identifier:	UC.1

1. Goal

Add book to the library system. If the book is new, information of that book will be added into the library system or update the information of book if it was available.

2. Actors

1. Librarian.

3. Assumptions

One copy for each book will not be allowed to borrow.

4. Constraints

None.

5. Priority:

High.

6. Pre-Conditions

1. The book has been received by the librarian.
2. Librarian Logged in (not applicable in our system).

7. Basic Flow

Flow Identifier	F1: <Add new book information>	
Step	**User Action**	**System Response**
1	Librarian enters book's ISBN.	

7. Basic Flow		
2		The system checks that the book is new one.
3	The librarian enters the information of book (title, author's names, publication year, edition, binding type, and publisher).	
4	The librarian enters the number of copies and if they allowed them to be borrowed.	
5		Add the book information.
6	Display a confirmation message.	

8. Alternative Flows		
Flow Identifier	AF1: <The book exists>	
Step	**User Action**	**System Response**
2.1		Book info already exist go to 4.
Flow Identifier	AF2: <Librarian cancels the add operation>	
Step	**User Action**	**System Response**
4.1	The librarian cancels add the books.	
		The system cancels the operation.

9. Exception Flows		
Flow Identifier	EF1 : <Any Book information is missed or not valid>	
Step	**User Action**	**System Response**
3.1		Display an error message indicates that error.
3.2	Resume 3.	
Flow Identifier	EF2: <Any copy information is missed or not valid>	
Step	**User Action**	**System Response**
4.1		Display an error message indicates that error.
4.2	Resume 4.	

10. "Includes" Use-Cases
None.

11. Post-Condition
1. Library system has been updated with the new information of the book.

12. Special Requirements
None.

13. Artifacts (Outputs)
1. Confirmation message is displayed to the librarian.

14. Test Cases
Unit testing: UT.1 / UT.2
Functional testing: FT.1 / FT.2 / FT.3 / FT.4 / FT.5 / FT.6

15. Identification:	
Written by:	Amani Al-Ajlan & Asma Al-Saleh.

2.2.2 Borrow Book:

Project Name:	Library System
Release Number:	1
Use Case Name:	Borrow book
Unique Identifier:	UC.2

1. Goal
This use case provides the functionality scenario of borrowing a book.

2. Actors
1. Librarian. 2. Member.

3. Assumptions

1. The ISBN is unique for each edition.
2. The copy ID is unique for each book copy in one edition.
3. For each book there is at least one copy is not allowed to be borrowed.
4. The book should be returned after two weeks.

4. Constraints

None.

5. Priority

High.

6. Pre-Conditions

1. Member is registered.
2. Book is existed.

7. Basic Flow

Flow Identifier	F1: <Borrow book>

Step	User Action	System Response
1	This use case begins when a member arrives at librarian desk with the book(s) to borrow.	
2	Librarian enters member ID.	
3	Librarian enters the Book ISBN number.	
4		The system displays all book copies.
5	Librarian selects the copy ID.	
6		The system checks the book state*.
7		The system checks the book availability**.
8		The system records Librarian ID, Member ID, Book ISBN, copy ID, Date, and Time.
9		The system changes the availability of the book copy.
10		The system displays the confirmation message and the return date.
11	Repeat (3-11) until no more books.	

7. Basic Flow

*book state: indicate if the book is allowed to be borrowed.

**book availability: indicate if the book is available.

8. Alternative Flows

Flow Identifier	AF1: <Borrow Not allowed book: prevent borrowing not allowed book>	
Step	**User Action**	**System Response**
6.1	The member asks to borrow a book that is not allowed to be borrowed.	The system indicates that the book is not allowed to be borrowed.
6.2		Go to step 11.
Flow Identifier	AF2: <Borrow not available book: prevent borrowing not available book>	
Step	**User Action**	**System Response**
7.1	The member asks to borrow a book that is not available.	The system indicates that the book is not available.
7.2		Go to step 11.

9. Exception Flows

Flow Identifier	EF1: <Validate member ID: make sure the member ID is a valid ID>	
Step	**User Action**	**System Response**
2.1	The Librarian enters invalid member ID.	Indicate that it was an error.
2.2		Go to step 2 in Basic Flow.
Flow Identifier	EF2: <Validate ISBN: make sure the ISBN is a valid ID>	
Step	**User Action**	**System Response**
3.1	The Librarian enters invalid ISBN.	Indicate that it was an error.
3.2		Go to step 3 in Basic Flow.

10. "Includes" Use-Cases

None.

11. Post-Condition

This use case can end with the following post-conditions:

1. New Borrowing record is added to the system.
2. The copy availability is changed to unavailable.

12. Special Requirements
None.

13. Artifacts
1. Confirmation message. 2. Return date.

14. Test Cases
Unit testing: UT.3 / UT.4 / UT.5 / UT.6 / UT.7 / UT.8
Functional testing: FT.7 / FT.8 / FT.9

14. Identification:	
Written by:	Manal AlBahlal, Nuha Alojian.

2.2.3Get Member Fines:

Project Name:	Library system
Release Number:	1
Use Case Name:	Get member fines
Unique Identifier:	UC.3

1. Goal
Get the total amount of fines for all overdue books for a certain member.

2. Actors
1. Librarian. 2. Member.

3. Assumptions
1. Only Returned books are taken into account. 2. Only members are allowed to borrow books and thus earn fines.

4. Constraints

None.

5. Priority

Medium.

6. Pre-Conditions

1. Librarian logged in (not applicable in our system).
2. Member is registered

7. Basic Flow

Flow Identifier	F1: <Get member's fines>	
Step	**User Action**	**System Response**
1	This use case starts when the Librarian enter a member ID and requests to calculate the total amount of Fines.	
2		Get the fine on an overdue book and add it to the total amount of fines.
3		Repeat step 2 for all overdue books.
4		Displays the Total amount of Fines on all over due books.

9. Exception Flows

Flow Identifier	EF1: <Validate member ID>	
Step	**User Action**	**System Response**
1.1	Invalid Member ID entered.	Displays an error message and requests re-entry of the memberID.

10. "Includes" Use-Cases

None.

11. Post-Condition

None.

12. Special Requirements

None.

13. Artifacts

1. Total amount of fines is displayed.

14. Test Cases

Functional testing: FT.10 / FT.11

14. Identification:

Written by:	Eidah Assidan, Sumayah ALRwais.

2.2.4Return Book:

Project Name:	Library system
Release Number:	2
Use Case Name:	Return Book
Unique Identifier:	UC.4

1. Goal

Returns an already borrowed book to library system and registers a fine for the returned book if applicable.

2. Actors

1. Librarian.
2. Member.

3. Assumptions

3. Allowed borrowing period is two weeks (14 days).
4. Fine for each late day is 0.5 SR.
5. Late days less than 1 day (e.g. half a day (12 Hours)) don't earn any fines
6. The Book copy ID is unique in a group of copies with an identical ISBN.

4. Constraints

None.

5. Priority

High.

6. Pre-Conditions

1. Librarian logged in (not applicable in our system).
2. The book to be returned is received by the Librarian.
3. The Book has been borrowed.

7. Basic Flow

Flow Identifier F1: <Return Book>

Step	User Action	System Response
1.	This use case begins when the librarian receives a book to be returned.	
2.	Librarian enters the Book ISBN number.	
3.		The system displays all the book copies.
4.	Librarian selects the book copy ID to be returned.	
5.		The system checks that the book is borrowed.
6.		The system calculates the fine earned for the returned book copy.
7.		The system records the book return date, fines earned and the book copy availability.
8.		The system displays a confirmation message with the fine earned for the returned book

8. Alternative Flows

Flow Identifier AF1: <The book copy is not borrowed>

Step	User Action	System Response
5.1	The librarian selects to return a book that is not borrowed.	The system indicates that the book is not borrowed currently.
5.2	Resume at step 4 of the basic flow.	

9. Exception Flows

Flow Identifier	EF1: <Validate ISBN>	
Step	**User Action**	**System Response**
2.1	The Librarian enters invalid ISBN	The system indicates that the ISBN is invalid.
2.2		Resume at step 2 of the basic flow.

10. "Includes" Use-Cases

None.

11. Post-Condition

1. Book copy return date is recorded in the system.
2. Fine is recorded for the returned book if applicable.
3. Book copy availability is set.

12. Special Requirements

None.

13. Artifacts

1. Confirmation message.
2. Fine amount if applicable.

14. Test Cases

Unit testing:UT.9 / UT.10 / UT.11.

Functional testing:FT.8 / FT.12 / FT.13.

14. Identification:

Written by:	Eidah Assidan, Sumayah ALRwais.

2.2.5Delete Book:

Project Name:	Library System
Release Number:	2
Use Case Name:	Delete book
Unique Identifier:	UC.5

1. Goal

This use case provides the functionality scenario of deleting a book.

2. Actors

3. Librarian.

3. Assumptions

5. The ISBN is unique for each edition.
6. The copy ID is unique for each book copy in one edition.
7. For each book there is at least one copy is not allowed to be borrowed.

4. Constraints

None.

5. Priority

Low.

6. Pre-Conditions

1. Book is existed.

7. Basic Flow

Flow Identifier	F1: <Delete book>	
Step	**User Action**	**System Response**
1	This use case begins when a librarian should delete a book copy.	
2	Librarian enters the Book ISBN number.	

7. Basic Flow		
3		The system displays all book copies.
4	Librarian selects the copy ID.	
5	Librarian enters a comment then he/she presses delete button.	
6		The system checks the book is allowed to be borrowed.
7		The system checks the book availability*.
8		The system sets the delete flag** to "Deleted".
9		The system records the entered comment.
10		The system displays the confirmation message.
*book availability: indicate if the book is available (i.e. not borrowed). ** Delete flag is a flag attribute indicates the deletion of entire book.		

8. Alternative Flows		
8.1 Alternative Flows (level 1)		
Flow Identifier	AF1: < Delete Not allowed book: Set another book copy to not allowed state>	
Step	**User Action**	**System Response**
6.1	The librarian shall delete a book that is not allowed to be borrowed.	
6.2		Check there are no more not allowed copies.
6.3		Retrieve any available book copy.
6.4		Change the retrieved copy state to not allowed.
6.5		Go to step 8 in Basic Flow.
Flow Identifier	AF2: < Delete not available book: prevent deleting not available book.>	
Step	**User Action**	**System Response**
7.1	The librarian shall delete a book that is not available.	The system indicates that the book is not available.
7.2		Go to step 10 in Basic Flow.
8.1.1 Alternative Flows (level 2)		
Flow Identifier	AF1.1: < Delete additional not allowed book: delete not allowed book where another not allowed book is available.>	
Step	**User Action**	**System Response**
6.2.1	It begins when the librarian shall delete a not allowed book copy and there is another allowed copy.	

8. Alternative Flows

6.2.2		Go to step 8 in Basic Flow.

Flow Identifier	AF1.2: < There are no available copies: delete the last not allowed book where all other copies are borrowed.>	
Step	**User Action**	**System Response**
6.3.1	It begins when the librarian shall delete the last not allowed book while there is no available book.	
6.3.2		Check if all copies are borrowed.
6.3.3		The system indicates an error.
6.3.4		Go to step 10 in Basic Flow.

8.1.1.1 Alternative Flows (level 3)

Flow Identifier	AF1.2.1: < There is no copy: delete the last not allowed book where there is no another copy>	
Step	**User Action**	**System Response**
6.3.2.1	It begins when the librarian should delete a not allowed book where there are no available copies.	
6.3.2.2		Go to step 8 in Basic Flow.

9. Exception Flows

Flow Identifier	EF1: < Validate ISBN: make sure the ISBN is a valid ID >	
Step	**User Action**	**System Response**
2.1	The Librarian enters invalid ISBN.	Indicate that it was an error.
2.2		Return to step 3 in Basic Flow.

10. "Includes" Use-Cases

None.

11. Post-Condition

This use case can end with the following post-conditions:

1. The delete flag attribute changed to indicate the deletions of the book.
2. A comment added to describe the deletion purpose.

12. Special Requirements

None.

13. Artifacts
1. Confirmation message.

14. Test Cases
Unit testing: UT.12 / UT.13 / UT.14 / UT.15 / UT.16 / UT.17.
Functional testing: FT.8 / FT.15 / FT.16

14. Identification:	
Written by:	Manal AlBahlal, Nuha Alojian.

2.2.6Update Book:

Project Name:	Library System.
Release Number:	2
Use Case Name:	Update book
Unique Identifier:	UC.6

1. Goal
Update the information of book that it is available in the library system.

2. Actors
1. Librarian.

3. Assumptions
None.

4. Constraints
None.

5. Priority:
Low.

6. Pre-Conditions

1. Librarian logged in (not applicable in our system).

7. Basic Flow

Flow Identifier	F1: <Update exist book information>	
Step	**User Action**	**System Response**
1	Librarian enters book's ISBN.	
2		The system checks that the book has been exist.
3		The system displays all book information.
4	The librarian enters updated information of the book (title, author's names, publication year, edition, binding type, and publisher).	
5		Update the book information.
6	Display a confirmation message.	

8. Alternative Flows

Flow Identifier	AF1: <Librarian cancels the update operation>	
Step	**User Action**	**System Response**
4.1	The librarian cancels update operation.	
4.2		The system cancels the operation.

9. Exception Flows

Flow Identifier	EF1 : <ISBN is not valid (not available)>	
Step	**User Action**	**System Response**
2.1		Display an error message indicates that error.
2.2	Resume 1.	
Flow Identifier	EF2 : <Any Book information is missed or not valid>	
Step	**User Action**	**System Response**
4.1		Display an error message indicates that error.
4.2	Resume 3.	

10. "Includes" Use-Cases

None.

11. Post-Condition

1. Library system has been updated with the new information of the book.

12. Special Requirements

None.

13. Artifacts (Outputs)

1. Confirmation message is displayed to the librarian.

14. Test Cases

Functional testing: FT.17 / FT.18.

15. Identification:

Written by:	Amani Al-Ajlan & Asma Al-Saleh.

Chapter 3

Design

3.1Three Tiers Architecture:

The Library system is based on a website representing the front end of the system which integrates with a BookManagement component to perform book management functionalities such as a new book insertion, book borrowing…etc. The book management component includes a Data Access Layer (DAL) which is responsible for handling all database related issues by providing all data services needed for the Book Management component.

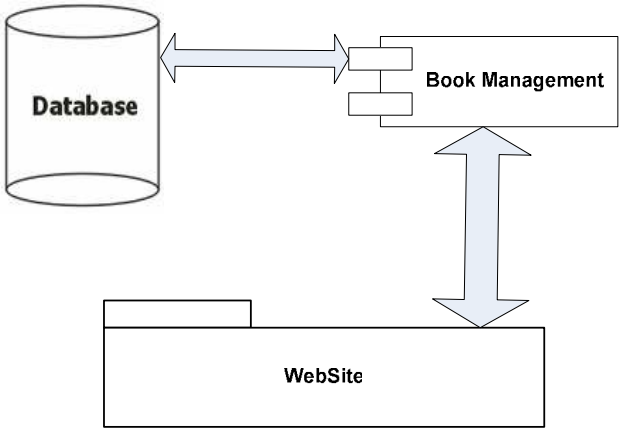


Figure 2: 3 tiers Architecture

3.2 Entity Relationship Diagram :

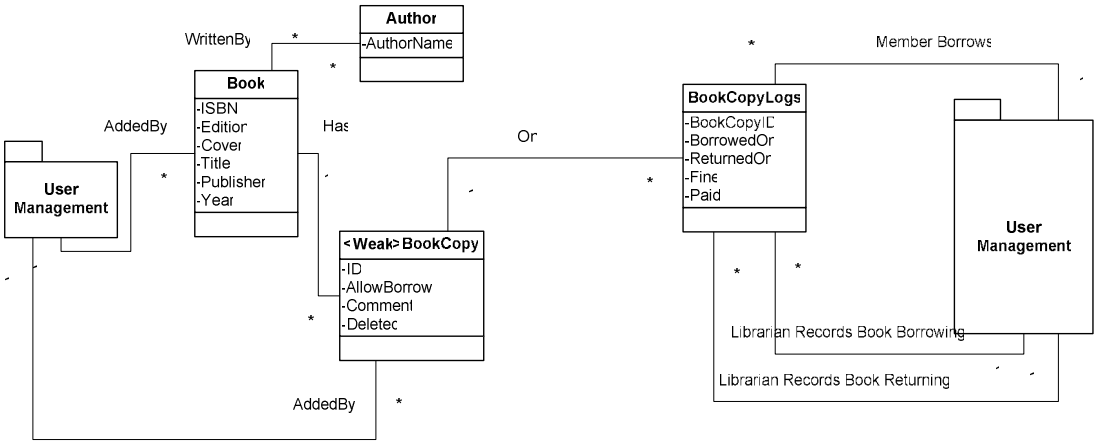


Figure 3: Entity Relationship Diagram

3.3Class Diagram:

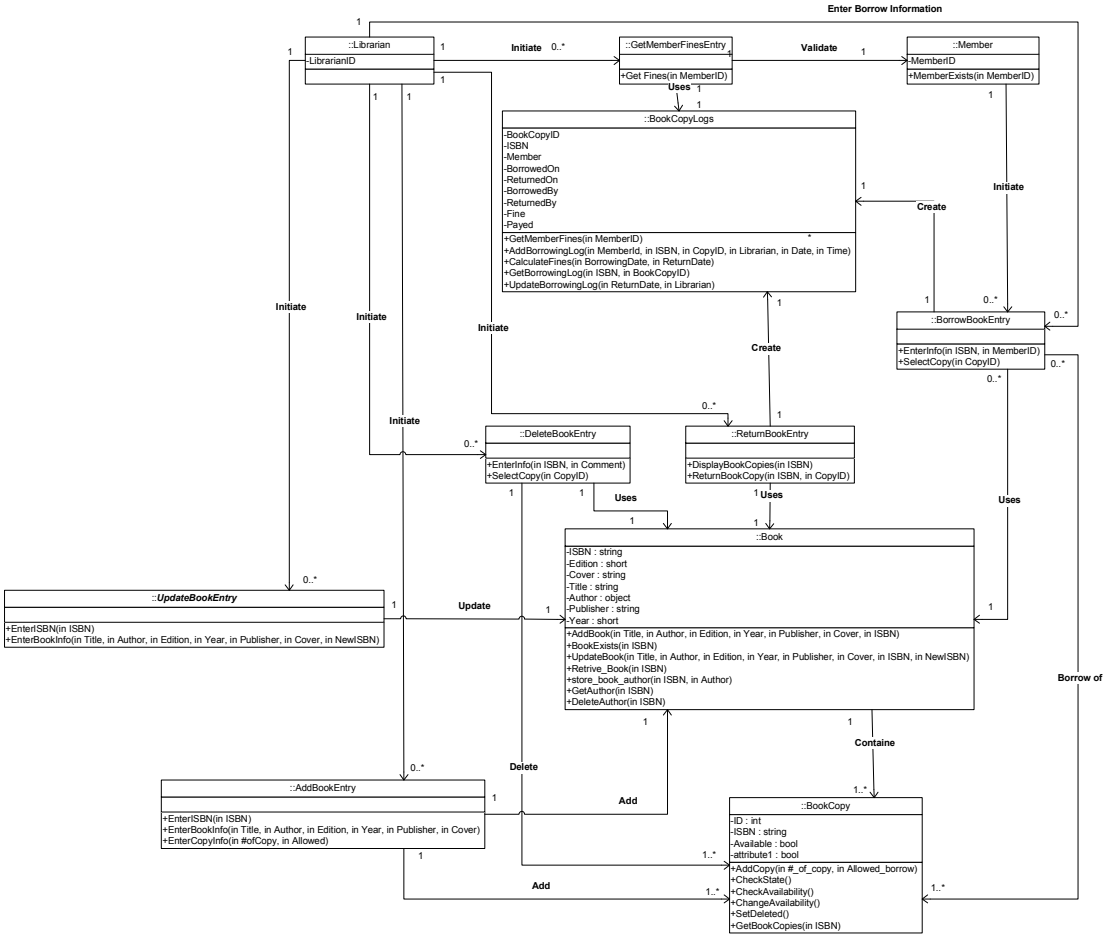


Figure 4: Class Diagram

3.4Sequence Diagrams:

3.4.1 Add Book (Release#1):

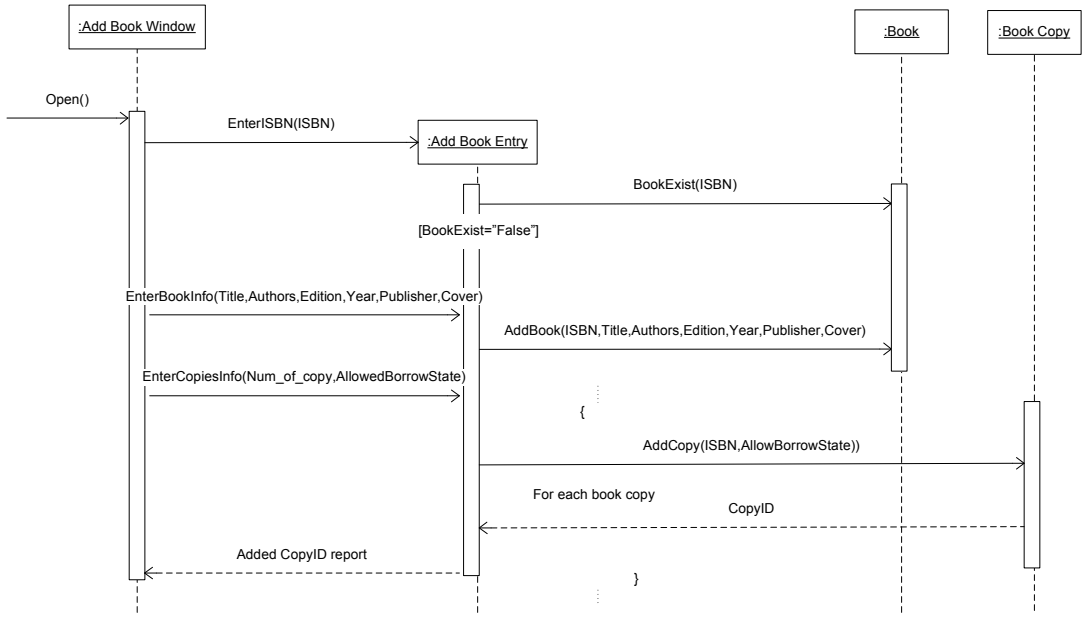


Figure 5: Add Book Sequence Diagram

3.4.2Borrow Book (Release#1):

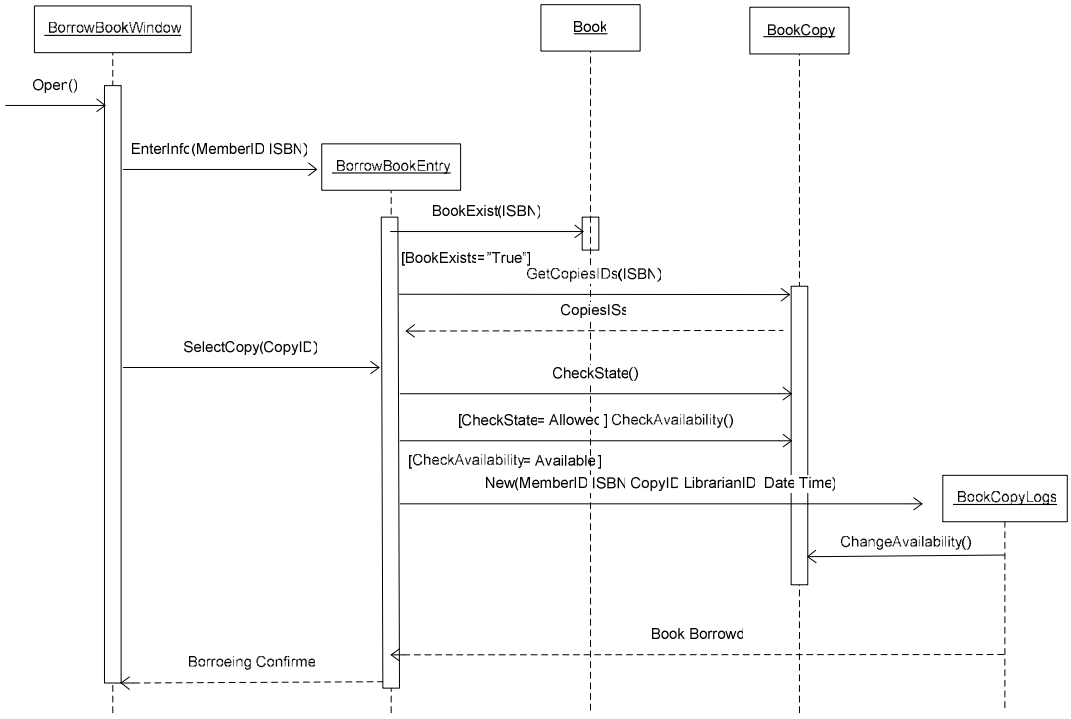


Figure 6: Borrow Book Sequence Diagram

3.4.3 Get Member's Fines (Release#1):

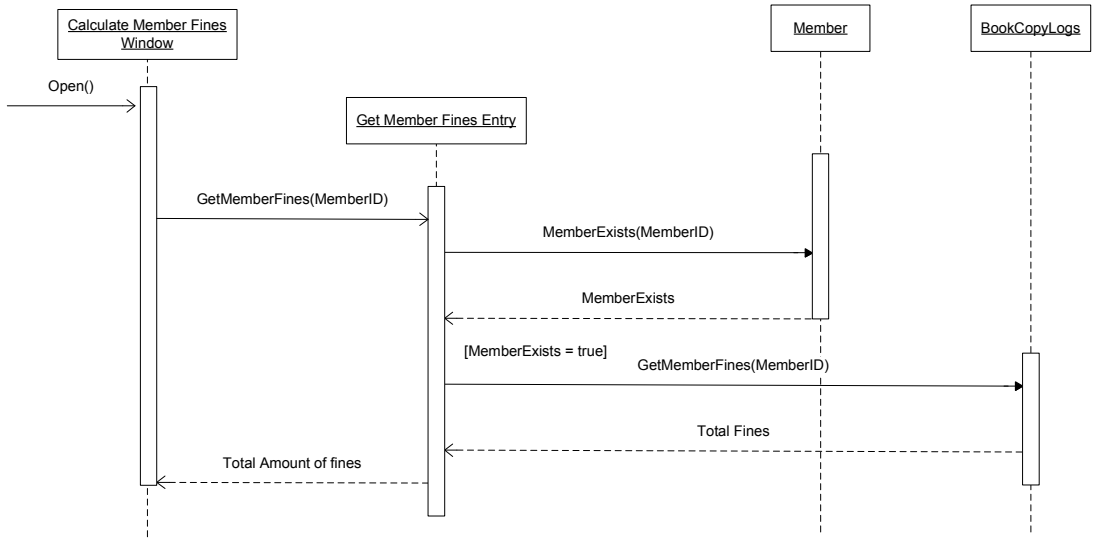


Figure 7: Get Member Fines Sequence Diagram

3.4.4Return Book (Release#2):

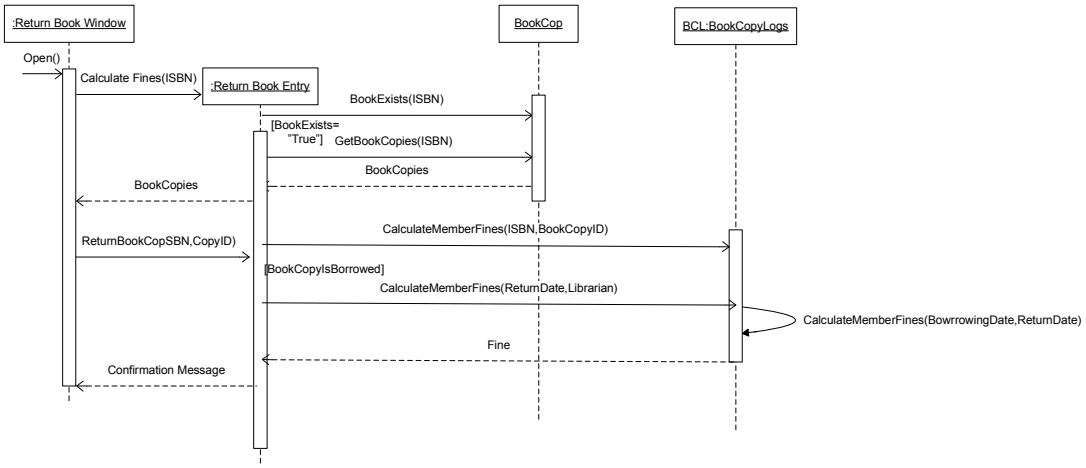


Figure 8: Return Book Sequence Diagram

3.4.5Delete Book (Release#2):

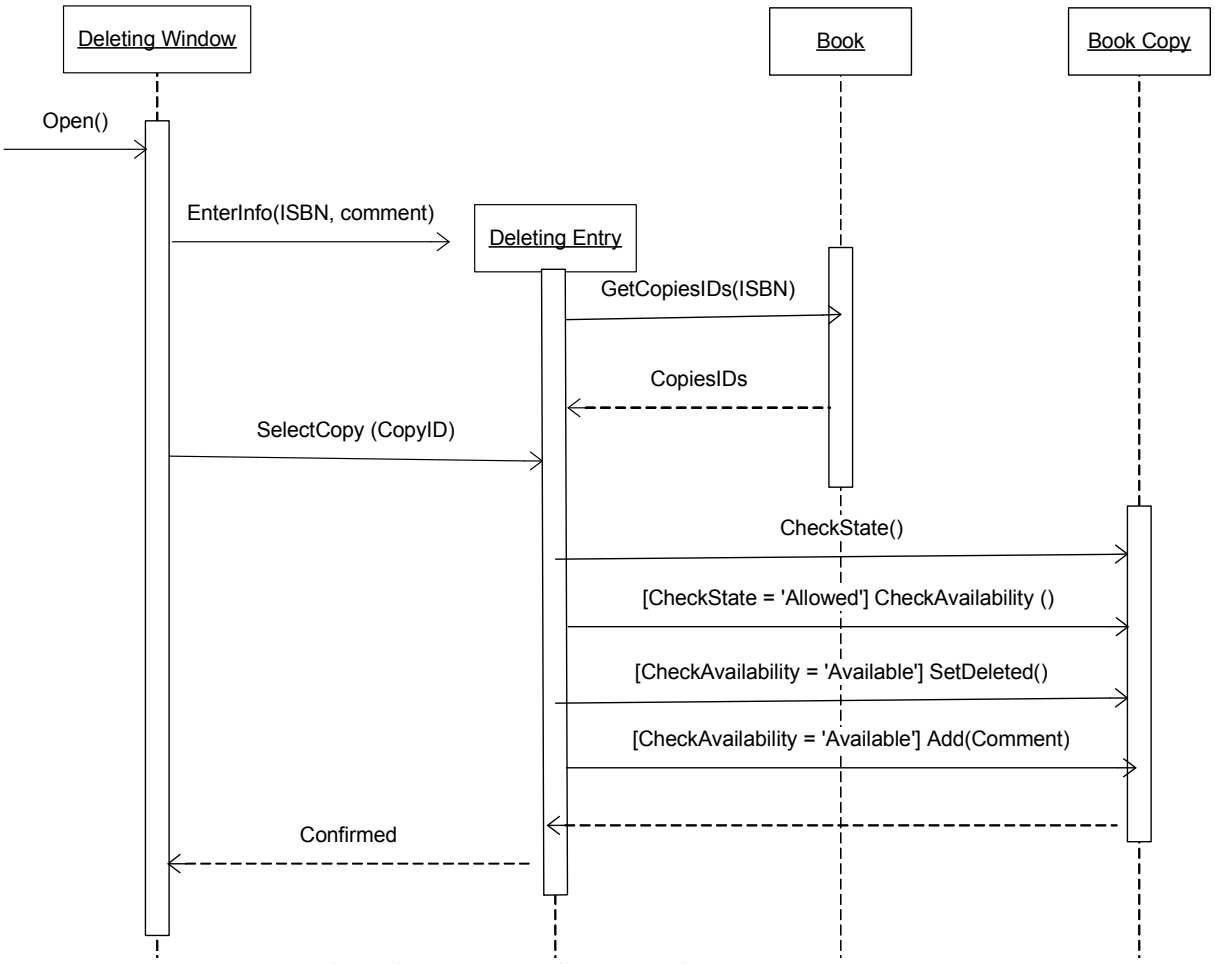


Figure 9: Delete Book Sequence Diagram

3.4.6Update Book (Release#2):

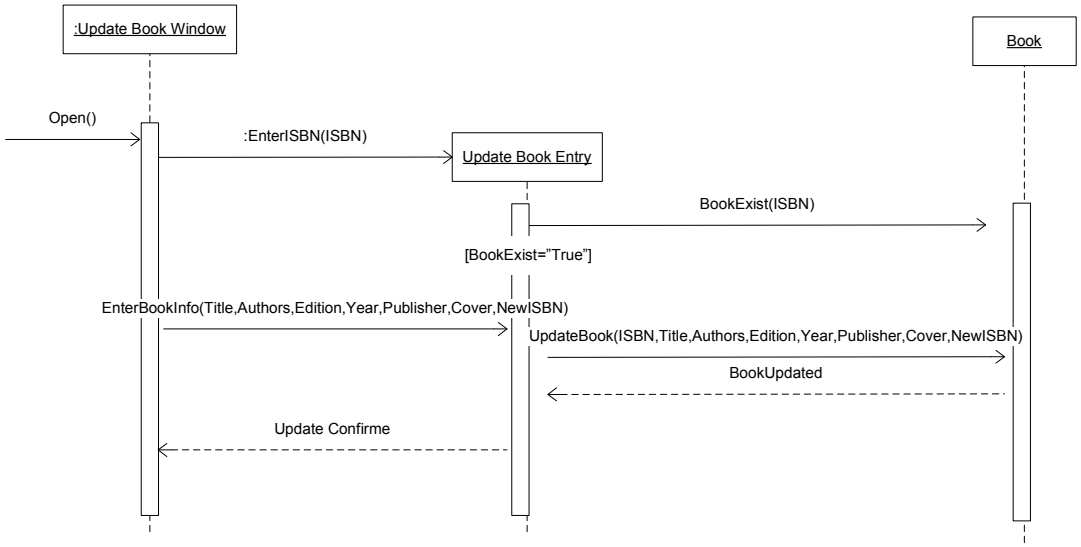


Figure 10: Update Book Sequence Diagram

3.5State Diagrams:

3.5.1Borrow Book (Release#1):

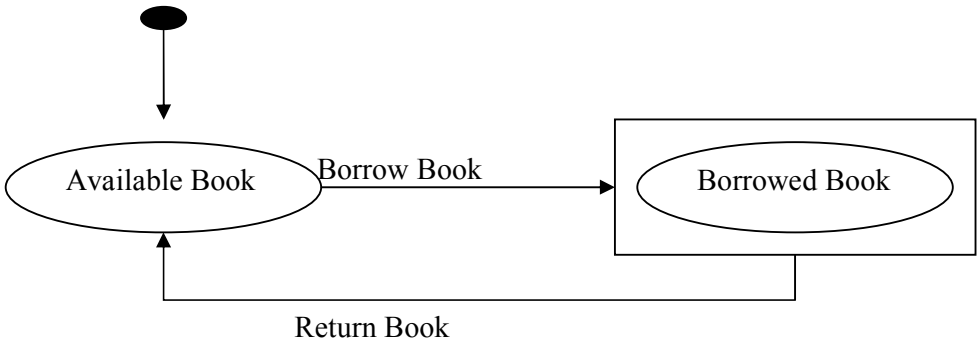


Figure 11: Borrow Book State Diagram

3.5.2Delete Book (Release#2):

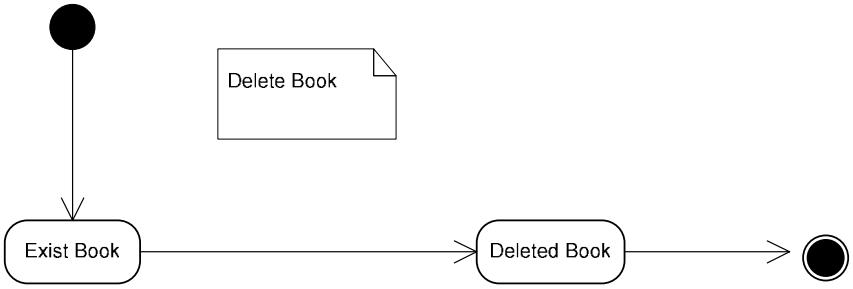


Figure 12: Delete Book State Diagram

Chapter 4

Testing

4.1 Functional Tests:

4.1.1 Add Book:

Test Case #: FT.1 **Test Case Name:** Check ISBN **Page:** 1 of 2

System: Library System **Subsystem**: Add book **Release #**: 1

Designed by: Amani & Asma **Design Date:** 27/11/2006

Executed by: Amani **Execution Date:** 28/11/2006

Short Description: Test the Library add book

Pre-conditions:
1. The Librarian has accessed the library system.
2. The book is new book (not available before)
3. The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click add book button	The system displays a message asking the librarian to enter ISBN	**Pass**	
2	Enter ISBN= '0764579072' and click 'Continue' button.	The system displays a message asking the librarian to enter book information	**Pass**	
3	Enter book's title 'introduction to algorithm'	-------	**Pass**	
4	Enter book's author 'Thomas H. Cormen'	-------	**Pass**	
5	Enter book's publisher 'McGraw-Hill'	-------	**Pass**	
6	Enter book's year '2001'	-------	**Pass**	
7	Enter book's edition'2'	-------	**Pass**	
8	Select book's binding type 'hard cover'	-------	**Pass**	
9	Then click 'Continue' button.	The system displays a message asking the librarian to enter number of book's copy and allowed to borrow or not.	**Pass**	
10	Enter number of book's copy '5'.	-------	**Pass**	
11	Select copies allowed to borrow.	-------	**Pass**	
12	Then click 'Add' button	The system displays a message of successful operation and number of copies that allowed borrowing and number of not allowed. Then system displays main menu.	**Pass**	

Test Case #: FT.1 **Test Case Name:** Check ISBN **Page:** 2 of 2

System: Library System **Subsystem**: Add book **Release #**: 1

Designed by: Amani & Asma **Design Date:** 27/11/2006

Executed by: Amani **Execution Date:** 28/11/2006

Short Description: Test the Library add book

13	**Check post-condition 1**			
14	Repeat step 1, then enter ISBN= ' '	The system displays a message of unsuccessful operation and asks the librarian to enter valid ISBN.	**Pass**	
15	Enter '0071238409' and click 'Continue' button.	The system displays a message asking the librarian to enter book information	**Pass**	
16	Enter book's title 'Software Engineering'	-------	**Pass**	
17	Enter book's author 'Roger s. Pressman'	-------	**Pass**	
18	Enter book's publisher 'McGraw-Hill'	-------	**Pass**	
19	Enter book's year '2005'	-------	**Pass**	
20	Enter book's edition'6'	-------	**Pass**	
21	Select book's binding type 'hard cover'	-------	**Pass**	
22	Then click 'Continue' button.	The system displays a message asking the librarian to enter number of book's copy and allowed to borrow or not.	**Pass**	
23	Enter number of book's copy '10'.	-------	**Pass**	
24	Select copies allowed to borrow.	-------	**Pass**	
25	Then click 'Add' button	The system displays a message of successful operation and number of copies that allowed borrowing and number of not allowed. Then system displays main menu.	**Pass**	
26	**Check post-condition 2**			

Post-conditions:

1. Book full information saved in database.
2. Book full information saved in database.

Test Case #: FT.2 **Test Case Name:** Check ISBN **Page:** 1 of 1

System: Library System **Subsystem:** Add book **Release #:** 1

Designed by: Amani & Asma **Design Date:** 27/11/2006

Executed by: Amani **Execution Date:** 28/11/2006

Short Description: Test the Library add book

Pre-conditions:

1. The Librarian has accessed the library system.
2. The book is available before, add new copies.
3. The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click add book button	The system displays a message asking the librarian to enter ISBN	**Pass**	
2	Enter '0-7645-1111-x' and click 'Cancel' button.	The subsystem exits and displays main menu.	**Pass**	
3	Repeat step 1, Enter ISBN= '0764579072' and click 'Continue' button.	The system displays a message asking the librarian to enter number of book's copy and allowed to borrow or not.	**Pass**	
4	Enter number of book's copy '100'.	-------	**Pass**	
5	Select copies allowed to borrow.	-------	**Pass**	
6	Then click 'Add' button	The system displays a message of successful operation and number of copies that allowed borrowing and number of not allowed. Then system displays main menu.]	**Pass**	
7	Check post-condition 1			

Post-conditions:

1. Book copies information saved in database.

Test Case #: FT.3 **Test Case Name:** Check book information **Page:** 1 of 2

System: Library System **Subsystem:** Add Book **Release #**: 1

Designed by: Amani & Asma **Design Date:** 27/11/2006

Executed by: Amani **Execution Date:** 28/11/2006

Short Description: Test the Library add book

Pre-conditions:

1. The Librarian has accessed the library system.
2. The book is new book (not available before)
3. The system displays the main menu
4.

Step	Action	Expected System Response	Pass/Fail	Comment
1	Click add book button	The system displays a message asking the librarian to enter ISBN	**Pass**	
2	Enter ISBN= '0769923412' and click 'Continue' button.	The system displays a message asking the librarian to enter book information	**Pass**	
3	Enter " " in title.	The system displays a message of unsuccessful operation and asks the librarian to enter valid information.	**Pass**	
4	Re-enter title ' Data Structures: A Pseudocode Approach Using C++'	------	**Pass**	
5	Enter book's author ' Forouzan'	-------	**Pass**	
6	Enter book's publisher 'McGraw-Hill'	-------	**Pass**	
7	Enter book's year '2001'	-------	**Pass**	
8	Enter book's edition'3'	-------	**Pass**	
9	Select book's binding type 'Normal cover'	-------	**Pass**	
10	Then click 'Continue' button.	The system displays a message asking the librarian to enter number of book's copy and allowed to borrow or not.	**Pass**	
11	Enter number of book's copy '7'.	-------	**Pass**	
12	Select copies allowed to borrow.	-------	**Pass**	

Test Case #: FT.3 **Test Case Name:** Check book information **Page:** 2 of 2

System: Library System **Subsystem:** Add Book **Release #**: 1

Designed by: Amani & Asma **Design Date:** 27/11/2006

Executed by: Amani **Execution Date:** 28/11/2006

Short Description: Test the Library add book

13	Then click 'Add' button	The system displays a message of successful operation and number of copies that allowed borrowing and number of not allowed. Then system displays main menu.	**Pass**	
14	Check post-condition 1			

Post-conditions:

1. Book full information saved in database.

Test Case #: FT.4 **Test Case Name:** Check book information **Page:** 1 of 2

System: Library System **Subsystem:** Add Book **Release #**: 1

Designed by: Amani & Asma **Design Date**: 27/11/2006

Executed by: Amani **Execution Date:** 28/11/2006

Short Description: Test the Library add book

Pre-conditions:

1. The Librarian has accessed the library system..
2. The book is new book (not available before).
3. The system displays the main menu.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click add book button	The system displays a message asking the librarian to enter ISBN	**Pass**	
2	Enter ISBN= 'P341123412' and click 'Continue' button.	The system displays a message asking the librarian to enter book information	**Pass**	
3	Enter title 'Data communication and networking'	------	**Pass**	
4	Enter book's author ' Forouzan'	-------	**Pass**	
5	Click 'Cancel' button.	The subsystem exits and displays main menu.	**Pass**	
6	Repeat 1,2,3,4 enter book's publisher 'McGraw-Hill'	-------	**Pass**	
7	Enter book's year '2010'	The system displays a message of unsuccessful operation and asks the librarian to enter valid year.	**Pass**	'2010' greater than current
8	Re-enter valid book's year '2003'	-------	**Pass**	
9	Enter book's edition '0'	The system displays a message of unsuccessful operation and asks the librarian to enter valid edition.	**Pass**	
10	Re-enter book's edition '4'	--------	**Pass**	
11	Select book's binding type 'hard cover'	--------	**Pass**	
12	Then click 'Continue' button.	The system displays a message asking the librarian to enter number of book's copy and allowed to borrow or not.	**Pass**	
13	Enter number of book's copy '1'.	-------	**Pass**	

Test Case #: FT.4	**Test Case Name:** Check book information	**Page:** 2 of 2
System: Library System	**Subsystem:** Add Book	**Release #**: 1
Designed by: Amani & Asma	**Design Date**: 27/11/2006	
Executed by: Amani	**Execution Date:** 28/11/2006	
Short Description: Test the Library add book		

14	Select copies allowed to borrow.	-------	**Pass**	
15	Check post-condition 1			

Post-conditions:
1. Book full information saved in database.

Test Case #: FT.5 **Test Case Name:** Add book copies **Page**: 1 of 1

System: Library System **Subsystem:** Add book **Release #**: 1

Designed by: Amani & Asma **Design Date:** 27/11/2006

Executed by: Amani **Execution Date:** 28/11/2006

Short Description: Test the Library adds book's copies information.

Pre-conditions:
1. The Librarian has accessed the library system.
2. The book is available before, add more copies.
3. The system displays the main menu.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click add book button	The system displays a message asking the librarian to enter ISBN	**Pass**	
2	Enter 'P-3411-2341-x' and click 'Continue' button.	The system displays a message asking the librarian to enter number of book's copy and allowed to borrow or not.	**Pass**	
3	Enter number of book's copy '-10'.	The system displays a message of unsuccessful operation and asks the librarian to enter valid number.	**Pass**	
4	Re-enter number of book's copy '32'.	------	**Pass**	
5	Select copies allowed to borrow.	The system displays a message of successful operation and number of copies that allowed borrowing and number of not allowed.	**Pass**	
6	Check post-condition 1			

Post-conditions:
1. Book copies information saved in database.

Test Case #: FT.6 **Test Case Name:** Add book copies **Page:** 1 of 2

System: Library System **Subsystem:** Add Book **Release #:** 1

Designed by: Amani & Asma **Design Date:** 27/11/2006

Executed by: Amani **Execution Date:** 28/11/2006

Short Description: Test the Library add book

Pre-conditions

1. The Librarian has accessed the library system.
2. The book is new book (not available before)
3. The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click add book button	The system displays a message asking the librarian to enter ISBN	**Pass**	
2	Enter ISBN= '1-7645-5492-x' and click 'Continue' button.	The system displays a message asking the librarian to enter book information	**Pass**	
3	Enter book's title 'C++ how to program'	-------	**Pass**	
4	Enter book's author 'Datail'	-------	**Pass**	
5	Enter book's publisher 'McGraw-Hill'	-------	**Pass**	
6	Enter book's year '2004'	-------	**Pass**	
7	Enter book's edition'3'	-------	**Pass**	
8	Select book's binding type 'Normal cover'	-------	**Pass**	
9	Then click 'Continue' button.	The system displays a message asking the librarian to enter number of book's copy and allowed to borrow or not.	**Pass**	
10	Enter number of book's copy '5'.	-------	**Pass**	
11	Select copies allowed to borrow.	-------	**Pass**	
12	Then click 'Cancel' button	The subsystem exits and displays main menu.	**Pass**	

Test Case #: FT.6	**Test Case Name:** Add book copies	**Page:** 2 of 2
System: Library System	**Subsystem:** Add Book	**Release #**: 1
Designed by: Amani & Asma	**Design Date:** 27/11/2006	
Executed by: Amani	**Execution Date:** 28/11/2006	
Short Description: Test the Library add book		

13	Repeat 1 to 11 Then click 'Add' button	The system displays a message of successful operation and number of copies that allowed borrowing and number of not allowed. Then system displays main menu.	**Pass**	
14	Check post-condition 1			

Post-conditions:
1. Book full information saved in database.

4.1.2 Borrow Book:

Test Case #: FT.7 **Test Case Name:** Validate Inputs **Page:** 1 of 2

System: Library System **Subsystem:** Borrow Book **Release #:** 1

Designed by: Manal, Nuha **Design Date:** 23/11/2006

Executed by: Manal, Nuha **Execution Date:** 27/11/2006

Short Description: Test user inputs.

Pre-conditions:
1. The system display the borrow page.
2. The Member ID 5 is not registered in the system.
3. The Member has a valid ID number (ID = 2).
4. The book with ISBN = 0071238409 has 2 available copies, the copy 5 is allowed to be borrowed but copy 1 are not.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Enter the Member ID = 5		**pass**	
2	Enter the Book ISBN = 0071238409		**pass**	
3	Click retrieve button.	The system fills the copies list (1, 5).	**pass**	
4	Select the copy ID = 5.		**pass**	
5	Click Borrow button.	The system indicates an error and a message are displayed (Message: This member is not registered in the system).	**pass**	
6	Repeat step 1 with Member ID = #4!b		**pass**	
7	Enter the Book ISBN = 0071238409		**pass**	
8	Click retrieve button.	The system fills the copies list (1, 5).	**pass**	
9	Select the copy ID = 5.		**pass**	
10	Click Borrow button.	The system indicates an error and a message are displayed (Message: Incorrect Format).	**pass**	
11	Repeat step 1 with Member ID = 2		**pass**	
12	Enter the Book ISBN = 0071238409		**pass**	

Test Case #: FT.7 **Test Case Name:** Validate Inputs **Page:** 2 of 2

System: Library System **Subsystem:** Borrow Book **Release #**: 1

Designed by: Manal, Nuha **Design Date:** 23/11/2006

Executed by: Manal, Nuha **Execution Date:** 27/11/2006

Short Description: Test user inputs.

13	Click retrieve button.	The system fills the copies list (1,5).	**pass**	
14	Select the copy ID = 5.		**pass**	
15	Click Borrow button.	The system displays a success message, and asks if there are more books to borrow.	**pass**	
16	Check post-condition 1 and 2.		**pass**	
17	Click No.	The systems back to the Borrow page.	**pass**	

Post-conditions:
1- A new borrowing record (Member ID = 2, ISBN = 0071238409, Copy ID = 5, Current date and time) is saved in the database.
2- The availability of copy 5 is changed to unavailable.

Test Case #: FT.8 **Test Case Name:** Fill Copies IDs list **Page:** 1 of 1

System: Library System **Subsystem:** Borrow Book **Release #**: 1

Designed by: Manal, Nuha **Design Date:** 23/11/2006

Executed by: Manal, Nuha **Execution Date:** 27/11/2006

Short Description: Test the retrieving copies IDs service.

Pre-conditions:

1. The system display the book borrow page.
2. There is no book with ISBN = 1203450066.
3. The book with ISBN = 1234123400 has 4 copies with IDs (1, 2, 3, 4).

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Enter the book ISBN = 1203450066		**pass**	
2	Click retrieve button.	The system indicates an error and a message displayed(Message: There is no book with this ISBN)	**pass**	
3	Re-enter another book ISBN = 1234123400		**pass**	
3	Click retrieve button.	The system fills the Copies IDs list (1,2,3,4).	**pass**	
4	Check post-condition 1.		**pass**	

Post-conditions:

1- All copies IDs with ISBN = 1234123400 are retrieved and displayed in the list.

Test Case #: FT.9 **Test Case Name:** Borrow **Page:** 1 of 2

System: Library System **Subsystem:** Borrow Book **Release #**: 1

Designed by: Manal, Nuha **Design Date:** 23/11/2006

Executed by: Manal, Nuha **Execution Date:** 27/11/2006

Short Description: Test the borrow book service.

Pre-conditions:
1. The system display the borrow page.
2. The Member has a valid ID number (ID = 2).
3. The book with ISBN = 2211123809 has 2 available copies, the copy 2 is allowed to be borrowed but copy 1 is not.
4. The book with ISBN = 3256236699 has 3 available copies, the copies 6 and 5 are allowed to be borrowed but copy 4 are not.
5. The book with ISBN = 4533330099 has 3 copies, the copies 8 and 10 are allowed to be borrowed but copy 7 are not (Copy 10 is not available).

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Enter the Member ID = 2.		**pass**	
2	Enter the Book ISBN = 2211123809		**pass**	
3	Click retrieve button.	The system fills the copies list (1, 2).	**pass**	
4	Select the copy ID = 1.		**pass**	
5	Click Borrow button.	The system indicates an error and a message are displayed (Message: This book is not allowed to be borrowed)	**pass**	
6	Select another book copy (Copy ID = 2).		**pass**	
7	Click Borrow button.	The system displays a success message, and asks if there are more books to borrow.	**pass**	
8	Check post-condition 1 and 2.		**pass**	
9	Click Yes.	The system reinitializes the ISBN fields and the copies list.	**pass**	
10	Repeat steps 2 (ISBN = 3256236699)		**pass**	
11	Click retrieve button.	The system fills the copies list (4,5,6).	**pass**	

Test Case #: FT.9 **Test Case Name:** Borrow **Page:** 2 of 2

System: Library System **Subsystem:** Borrow Book **Release #:** 1

Designed by: Manal, Nuha **Design Date:** 23/11/2006

Executed by: Manal, Nuha **Execution Date:** 27/11/2006

Short Description: Test the borrow book service.

12	Select the copy ID = 5.		**pass**	
13	Click Borrow button.	The system displays a success message, and asks if there are more books to borrow.	**pass**	
14	Check post-condition 1 and 2.		**pass**	
15	Click Yes		**pass**	
16	Repeat steps 2 (ISBN = 4533330099)		**pass**	
17	Click retrieve button.	The system fills the copies list (7,8,10).	**pass**	
18	Select the copy ID = 10.		**pass**	
19	Click Borrow button.	The system indicates an error and a message are displayed (Message: This book is borrowed)	**pass**	
20	Select another book copy (Copy ID = 8).		**pass**	
21	Click Borrow button.	The system displays a success message, and asks if there are more books to borrow.	**pass**	
22	Check post-condition 5 and 6.		**pass**	
23	Click No.	The systems back to the Borrow page.	**pass**	

Post-conditions:
1- A new borrowing record (Member ID = 2, ISBN = 2211123809, Copy ID = 2, Current date and time) is saved in the database.
2- The availability of copy 2 is changed to unavailable.
3- A new borrowing record (Member ID = 2, ISBN = 3256236699, Copy ID = 5, Current date and time) is saved in the database.
4- The availability of copy 5 is changed to unavailable.
5- A new borrowing record (Member ID = 2, ISBN = 4533330099, Copy ID = 8, Current date and time) is saved in the database.
6- The availability of copy 8 is changed to unavailable.

4.1.3Get Member Fines:

Test Case #: FT.10 **Test Case Name:** Get fines **Page:** 1 of 1

System: library system **Subsystem:** Get member's fines **Release #**: 1

Designed by: Eidah, Sumayah **Design Date:** 25/11/2006

Executed by: Eidah, Sumayah **Execution Date:** 26/11/2006

Short Description: Test the library system calculate member's fine

Pre-conditions
For the member with ID = '2', there are three borrowed books:
1. First book: returned late, fine is 5.5, not paid yet.
2. Second book: returned late, fine is 4.25, not paid yet.
3. Third book: returned late, fine is 1, not paid yet.
4. Fourth book: returned on time, fine is 0
5. Fifth book: returned late, fine is 9, already paid.

For the member with ID = '3', one book is borrowed and returned late with fine = 20
The system displays the Calculate member fines screen

Step	Action	Expected System Response	Pass/Fail	Comment
1	Enter Member ID '2', then click on Get member fines	The System displays the total amount of unpaid fines for the entered member ID	Pass	
	Check post-condition 1		pass	

Post-conditions:
1. displays the total fine 10.75 SR

Test Case #: FT.11 **Test Case Name:** Check if the member exists **Page:** 1 of 1

System: Library System **Subsystem:** Get member's fines **Release #**: 1

Designed by: Eidah, Sumayah **Design Date:** 25/11/2006

Executed by: Eidah, Sumayah **Execution Date:** 26/11/2006

Short Description: Test the library system calculate member's fine

Pre-conditions

1. Librarian exists with ID '1'
2. No member or librarian with ID '12' exists
3. The system displays the Calculate member fines screen

Step	Action	Expected System Response	Pass /Fail	Comment
1	Click on Calculate member fines button	The system requests entry of the member ID	Pass	
2	Enter Member ID '1', then click on Get member fines	The system displays an appropriate message explaining that the entered ID is not registered in the system as a member.	Pass	
3	Check post-condition 1		Pass	
4	Repeat step 2 with member ID '12'	The system displays an appropriate message explaining that the entered ID is not registered in the system as a member.	Pass	
5	Check post-condition 2		Pass	

Post-condition 1

Appropriate message is displayed.

Post-condition 2

Appropriate message is displayed.

4.1.4Return Book:

Test Case #: FT.12	**Test Case Name:** validate inputs	**Page:** 1 of 1
System: Library System	**Subsystem**: Return book	**Release #**: 2
Designed by: Eidah, Sumayah	**Design Date:** 25/12/2006	
Executed by: Eidah, Sumayah	**Execution Date:** 27/12/2006	
Short Description: Test user inputs.		

Pre-conditions:

1. The system displays the return book page.
2. The book ISBN "1010" exists and has copies (153,154). Book copy with ID 154 was borrowed on Dec 26,2006 (i.e. can be returned) and book copy with ID 153 is not borrowed (i.e. available to be borrowed)
3. Current date is Dec 27,2006

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Enter ISBN "1010" and click "Retrieve Book copies" link.	The system fills the copy IDs list with copies (153,154)	**Pass**	
2	Select Copy ID "153" and click "Return Book"	The system displays a message indicating the Book copy 153 is not borrowed currently.	**Pass**	
3	**Check post-condition 1**			
4	Select Copy ID "154" and click "Return Book"	The system displays a confirmation message confirming the return operation success.	**Pass**	
5	**Check post-condition 2**			

Post-conditions 1:

1. Error message is displayed.

Post-conditions 2:

1. Book copy return date is recorded.
2. Confirmation message is displayed.

Test Case #: FT.13	**Test Case Name:** Return book	**Page:** 1 of 1
System: Library System	**Subsystem**: Return book	**Release #**: 2
Designed by: Eidah & Sumayah	**Design Date:** 25/12/2006	
Executed by: Eidah & Sumayah	**Execution Date:** 27/12/2006	
Short Description: Test the return book functionality		

Pre-conditions:

1. The system displays the return book page.
2. The book ISBN "6789" exists and has copies (162,165,166).
3. Book copy with ID 162 was borrowed on Nov 27,2006 (i.e. can be returned) (late by 16 days)
4. Book copy with ID 165 was borrowed on Dec 13,2006 (i.e. can be returned) (returned on the last day)
5. Book copy with ID 166 was borrowed on Dec 26,2006 (i.e. can be returned) (returned on the next day)
6. Current date is Dec 27, 2006.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Enter ISBN "6789" and click "Retrieve Book copies" link.	The system fills the copy IDs list with copies (162,165,166)	**Pass**	
2	Select Copy ID "162" and click "Return Book"	The system displays a confirmation message confirming the return operation success with the fine of 8 SR.	**Pass**	
3	**Check post-condition 1**			
5	Select Copy ID "165" and click "Return Book"	The system displays a confirmation message confirming the return operation success.	**Pass**	
6	**Check post-condition 2**			
7	Select Copy ID "166" and click "Return Book"	The system displays a confirmation message confirming the return operation success.	**Pass**	
8	**Check post-condition 2**			

Post-conditions 1:

1. Book copy return date is recorded.
2. Fine of 8 SR is recorded.
3. Confirmation message is displayed.

Post-conditions 2:

1. Book copy return date is recorded.
2. Confirmation message is displayed.

4.1.5 Delete Book:

Page: 1 of 2

Test Case #: FT.14 **Test Case Name:** Validate confirmation window

System: Library System **Subsystem:** Delete Book **Release #:** 2

Designed by: Manal, Nuha **Design Date:** 26/12/2006

Executed by: Manal, Nuha **Execution Date:** 27/12/2006

Short Description: Test the book copy is deleted or not deleted according to the user response to the confirmation window.

Pre-conditions:

1. The system displays the delete book page.
2. The book with ISBN = 0071238408 has 4 copies with IDs(30) not allowed to be borrowed while IDs(31,32,33) not borrowed books.
3. The book with ISBN = 0071238409 has 4 copies with IDs(34) while IDs(35,36) not borrowed.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Enter the book ISBN = 0071238408		**pass**	
2	Click retrieve button.	The system fills the Copies IDs list(30,32,33)	**pass**	
3	Select copy ID 32, insert comment then click Delete button	A confirmation message is displayed to make sure that the librarian want to delete Copy with ID=32	**pass**	
4	Click Yes button	The system displays a success message.	**pass**	
5	Check post-condition 1		**pass**	
6	Click on the Delete book link to delete another book.	The system reinitializes the ISBN fields and the copies list.	**pass**	
7	Enter the book ISBN = 0071238408		**pass**	
8	Click retrieve button.	The system fills the Copies IDs list(30,33)	**pass**	
9	Select copy ID 33, insert comment then click Delete button	A confirmation message is displayed to make sure that the librarian want to delete Copy with ID=33	**pass**	
10	Click No button	remain on the same page	**pass**	
11	Check post-condition 2		**pass**	

Test Case #: FT.14 **Test Case Name:** Validate deletion with confirmation message

System: Library System **Subsystem:** Delete Book **Release #**: 2

Designed by: Manal, Nuha **Design Date:** 26/12/2006

Executed by: Manal, Nuha **Execution Date:** 27/12/2006

Short Description: Test the book copy is deleted or not deleted according to the user response to the confirmation message.

12	Enter the book ISBN = 0071238409		**pass**	
13	Click retrieve button.	The system fills the Copies IDs list(34,35)	**pass**	
14	Select copy ID 35, insert comment then click Delete button.	A confirmation message is displayed to make sure that the librarian want to delete Copy with ID=35	**pass**	
15	Click Cancel button	remain on the same page	**pass**	
16	Check post-condition 3		**pass**	
17	Select copy ID 35, insert comment then click Delete button	A confirmation message is displayed to make sure that the librarian want to delete Copy with ID=35	**pass**	
18	Click YES button	The system displays a success message, and asks if the user wants to delete another book.	**pass**	
19	Check post-condition 4.		**pass**	

Post-conditions:

1- The Deleted flag of the book with ISBN=0071238408 Copy ID= 32 is changed to true and the comment is inserted.
2- No changes are made on Deleted flags of any copy of book with with ISBN= 0071238408.
3- No changes are made on Deleted flags of any copy of book with with ISBN= 0071238409.
4- The Deleted flag of the book with ISBN=0071238409 Copy ID= 35 is changed to true and the comment is inserted.

Test Case #: FT.15 **Test Case Name:** Delete not allowed to be borrowed copy

System: Library System **Subsystem:** Delete Book **Release #**: 2

Designed by: Manal, Nuha **Design Date:** 26/12/2006

Executed by: Manal, Nuha **Execution Date:** 27/12/2006

Short Description: Test the not allowed to be borrowed copy book is deleted if one of the following cases is true.

1. There is at least another not allowed copy.
2. There is at least one available copy if it is the last not allowed copy.
3. It is the last copy.

Pre-conditions:

5. The system displays the Delete Book page.
6. The book with ISBN = 0071238401 has 5 copies, IDs(1,2,3,4,5) are not allowed to be borrowed.
7. The book with ISBN = 0071238402 has 5 copies, IDs(6,7,10) are not allowed to be borrowed while IDs(8,9) not borrowed.
8. The book with ISBN = 0071238403 has 8 copies, IDs(11,12) are not allowed to be borrowed, IDs(13,15,17) are borrowed and IDs(14,16,18) are not borrowed.
9. The book with ISBN = 0071238404 has 5 copies, IDs(19,20) are not allowed to be borrowed and IDs(21,22,23) are borrowed.
10. The book with ISBN = 0071238405 has 5 copies, ID(24) is not allowed to be borrowed, ID(25) borrowed and ID(26) not borrowed.
11. The book with ISBN = 0071238406 has 5 copies, ID(27) is not allowed to be borrowed, and ID(28,29) is not borrowed.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Enter the Book ISBN = 0071238401		**pass**	
2	Click retrieve button.	The system fills the copies list (1,2,3,4,5).	**pass**	
3	Select the copy ID = 3 and insert comment.		**pass**	
4	Click Delete button.	A confirmation message is displayed to make sure that the librarian want to delete Copy with ID = 3.	**pass**	
5	Click Yes button.	The system displays a success message, and asks if the user wants to delete another book.	**pass**	
6	Check post-condition 1.		**pass**	

Test Case #: FT.15 **Test Case Name:** Delete not allowed to be borrowed copy

System: Library System **Subsystem:** Delete Book **Release #**: 2

Designed by: Manal, Nuha **Design Date:** 26/12/2006

Executed by: Manal, Nuha **Execution Date:** 27/12/2006

Short Description: Test the not allowed to be borrowed copy book is deleted if one of the following cases is true.

1. There is at least another not allowed copy.
2. There is at least one available copy if it is the last not allowed copy.
3. It is the last copy.

7	Repeat steps from 1 to 5 with ISBN=0071238402 and copy ID=7		**pass**	
8	Check post-condition 2		**pass**	
9	Repeat steps from 1 to 5 with ISBN=0071238402 and copy ID=9			
10	Check post-condition 3		**pass**	
11	Repeat steps from 1 to 5 with ISBN=0071238402 and copy ID=10		**pass**	
12	Check post-condition 4		**pass**	
13	Repeat steps from 1 to 5 with ISBN=0071238404 and copy ID=19		**pass**	
14	Check post-condition 5		**pass**	
15	Repeat steps from 1 to 5 with ISBN=0071238403 and copy ID=12		**pass**	
16	Check post-condition 6			
17	Repeat steps from 1 to 3 with ISBN=0071238404 and copy ID=20		**pass**	
18	Click Delete button	An error message is displayed indicate this is the last not allowed to be borrowed book and all other book copies are borrowed.	**pass**	
19	Repeat steps from 1 to 5 with ISBN=0071238403 and copy ID=11		**pass**	
20	Check post-condition 7		**pass**	
21	Repeat steps from 1 to 5 with ISBN=0071238406 and copy ID=24		**pass**	
22	Check post-condition 8		**pass**	

Test Case #: FT.15 **Test Case Name:** Delete not allowed to be borrowed copy

System: Library System **Subsystem:** Delete Book **Release #**: 2

Designed by: Manal, Nuha **Design Date:** 26/12/2006

Executed by: Manal, Nuha **Execution Date:** 27/12/2006

Short Description: Test the not allowed to be borrowed copy book is deleted if one of the following cases is true.

1. There is at least another not allowed copy.
2. There is at least one available copy if it is the last not allowed copy.
3. It is the last copy.

23	Repeat steps from 1 to 5 with ISBN=0071238406 and copy ID=27		**pass**	
24	Check post-condition 9		**pass**	
25	Repeat steps from 1 to 5 with ISBN=0071238406 and copy ID=28		**pass**	
26	Check post-condition 10		**pass**	
27	Repeat steps from 1 to 5 with ISBN=0071238406 and copy ID=29		**pass**	
28	Check post-condition 10		**pass**	

Post-conditions:

1- The Deleted flag of the book with ISBN=0071238401 Copy ID= 3 is changed to true and the comment is inserted.
2- The Deleted flag of the book with ISBN=0071238402 Copy ID= 7 is changed to true and the comment is inserted.
3- The Deleted flag of the book with ISBN=0071238402 Copy ID= 9 is changed to true and the comment is inserted.
4- The Deleted flag of the book with ISBN=0071238402 Copy ID= 10 is changed to true and the comment is inserted.
5- The Deleted flag of the book with ISBN=0071238404 Copy ID= 19 is changed to true and the comment is inserted.
6- The Deleted flag of the book with ISBN=0071238403 Copy ID= 12 is changed to true and the comment is inserted.
7- The Deleted flag of the book with ISBN=0071238403 Copy ID= 11 is changed to true, the comment is inserted and the sate of Copy ID=14 is changes to not allowed.
8- The Deleted flag of the book with ISBN=0071238405 Copy ID= 24 is changed to true, the comment is inserted and the sate of Copy ID=26 is changes to not allowed.
9- The Deleted flag of the book with ISBN=0071238406 Copy ID= 27 is changed to true, the comment is inserted and the sate of Copy ID=28 is changes to not allowed.
10- The Deleted flag of the book with ISBN=0071238406 Copy ID= 28 is changed to true, the comment is inserted and the sate of Copy ID=29 is changes to not allowed.
11- The Deleted flag of the book with ISBN=0071238406 Copy ID= 29 is changed to true and the comment is inserted.

Test Case #: FT.16 **Test Case Name:** Delete allowed to be borrowed copy

System: Library System **Subsystem:** Delete Book **Release #:** 2

Designed by: Manal, Nuha **Design Date:** 26/12/2006

Executed by: Manal, Nuha **Execution Date:** 27/12/2006

Short Description: Test the allowed copy is deleted if it's not borrowed at the deletion time.

Pre-conditions:
1. The system displays the Delete Book page.
2. The book with ISBN = 0071238501 has 5 copies, ID(36) is not allowed to be borrowed while IDs(37,39) are borrowed IDs(38,40) are not borrowed.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Enter the Book ISBN = 0071238501		**pass**	
2	Click retrieve button.	The system fills the copies list (36,37,38,39,40).	**pass**	
3	Select the copy ID = 40 and insert a comment.		**pass**	
4	Click Delete button.	A confirmation message is displayed to make sure that the librarian want to delete Copy with ID = 40.	**pass**	
5	Click Yes button.	The system displays a success message, and asks if the user wants to delete another book.	**pass**	
	Check post-condition 1.		**pass**	
6	Repeat steps from 1 to 4 with ISBN=0071238501 and copy ID=39		**pass**	
7	Click Yes button.	An error message is displayed indicate that the book copy is borrowed	**pass**	
9	Repeat steps from 1 to 5 with ISBN=0071238501 and copy ID=38		**pass**	
11	Check post-condition 2.		**pass**	

Post-conditions:
1- The Deleted flag of the book with ISBN=0071238501 Copy ID= 40 is changed to true and the comment is inserted.
2- The Deleted flag of the book with ISBN=0071238501 Copy ID= 39 is changed to true and the comment is inserted.

4.1.6 Update Book:

Test Case #: FT.17	**Test Case Name:** Update book information	**Page:** 1 of 1
System: Library System	**Subsystem**: update book	**Release #:** 2
Designed by: Amani & Asma	**Design Date:** 26/12/2006	
Executed by: Amani	**Execution Date:** 28/12/2006	
Short Description: Test the Library update book		

Pre-conditions:
1. The librarian has accessed the library system.
2. The book with ISBN= '0764579072' is one of the library books.
3. The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click 'Update' book button	The system displays a message asking the librarian to enter ISBN for the updated book.	**Pass**	
2	Enter ISBN= '0764579072' and click 'Update' button.	The system displays all book information and allows librarian to edit this information.	**Pass**	
3	Edit book's title 'introduction to algorithm and Analysis'	-------	**Pass**	
4	Edit book's year '2005'	-------	**Pass**	
5	Then click 'Update' button	The system displays a message of successful operation.Then system displays main menu.	**Pass**	
6	**Check post-condition 1**			

Post-conditions:
1. Book full information updated in database.

Test Case #: FT.18 **Test Case Name:** Update book information **Page:** 1 of 1

System: Library System **Subsystem**: update book **Release #:** 2

Designed by: Amani & Asma **Design Date:** 16/12/2006

Executed by: Amani **Execution Date:** 18/12/2006

Short Description: Test the Library update book

Pre-conditions:

1. The librarian has accessed the library system.
2. The book with ISBN= '0071238409' is one of the library books.
3. No book in the library has ISBN= 'E71238429'.
4. The system displays the main menu

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click update book button	The system displays a message asking the librarian to enter ISBN for the updated book.	**Pass**	
2	Enter ISBN= ' ' Then click 'Update' button	The system displays a message of unsuccessful operation and asks the librarian to enter valid ISBN.	**Pass**	
3	Re-enter ISBN='E71238429' Then click 'Update' button	The system displays a message of unsuccessful operation and asks the librarian to enter valid ISBN.	**Pass**	
4	Re-enter '0071238409' and click 'Update' button.	The system displays all book information and allows librarian to edit this information.	**Pass**	
5	Edit book's edition ' '	The system displays a message of unsuccessful operation and asks the librarian to enter valid edition.	**Pass**	
6	Re-enter valid book's edition '5'	-------	**Pass**	
7	Then click 'Update' button	The system displays a message of successful operation.Then system displays main menu.		
8	**Check post-condition 1**			

Post-conditions:

1. Book full information updated in database.

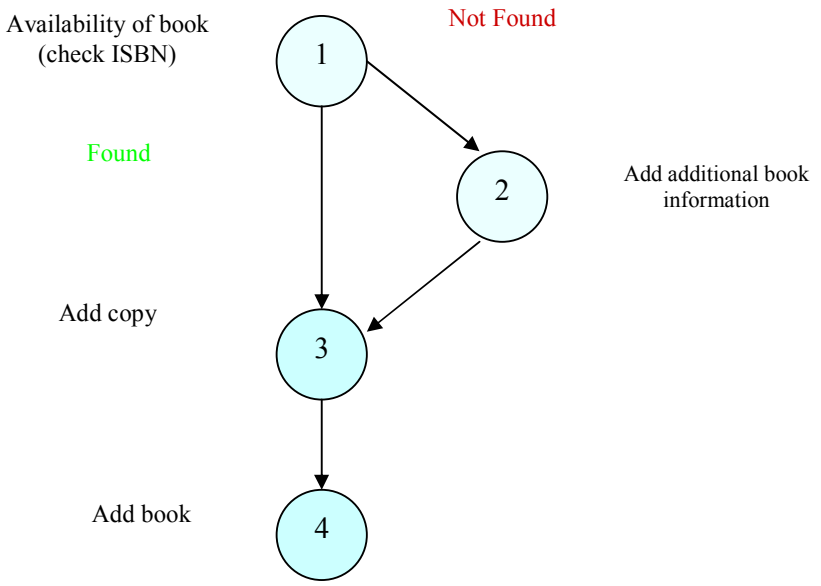
4.2Unit Tests:

We will use the program flow graphs approach in writing unit test cases:

•Flow Graph:

- nodes representing program decisions
- arcs representing the flow of control
- Ignore sequential statements (assignments, procedures calls, I/O)

4.2.1 Add Book:



Cyclomatic complexity = Number of edges -Number of nodes + 2
= 4 – 4 +2= 2 paths

Thus 2 independent paths:

- Path1: 1,2,3,4
- Path2: 1,3,4

Unit Test Case #: 1	Path #: 1	Release #: 1

Path sequence	Data	result	output	change on DB
1,2,3,4	Book ISBN = '220170857 5' (book not available before) Book's title 'introduction to algorithm' Book's author 'Thomas H. Cormen' Book's edition'2' Book's year '2001' Book's publisher 'McGraw-Hill' Book's binding type 'hard cover' Number of copies=4 Allowed to borrow	**pass**	confirmation message (Borrowing Process is successful) “see screen shots”	a new record with (Book ISBN = '220170857 5' Book's title 'introduction to algorithm' Book's author 'Thomas H. Cormen' Book's edition'2' Book's year '2001' Book's publisher 'McGraw-Hill' Book's binding type 'hard cover' Number of copies=4 Allowed to borrow) is saved

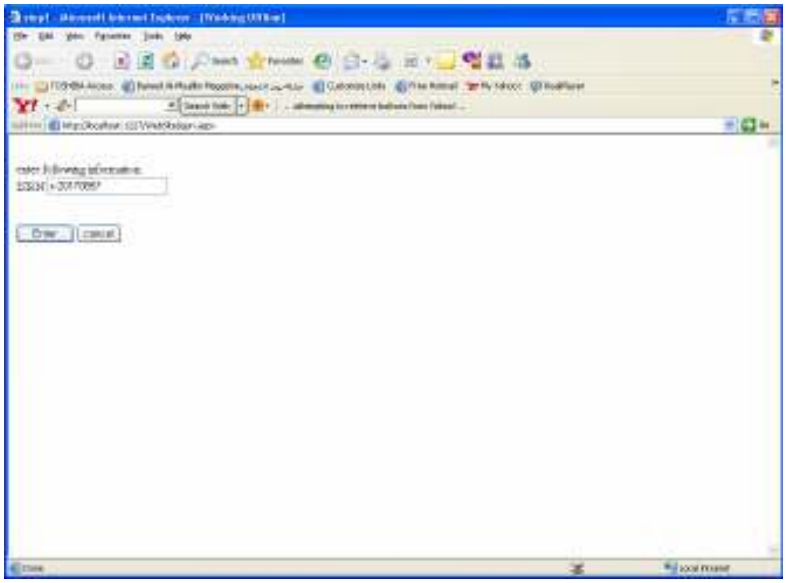


Figure 13: unit test screen shot

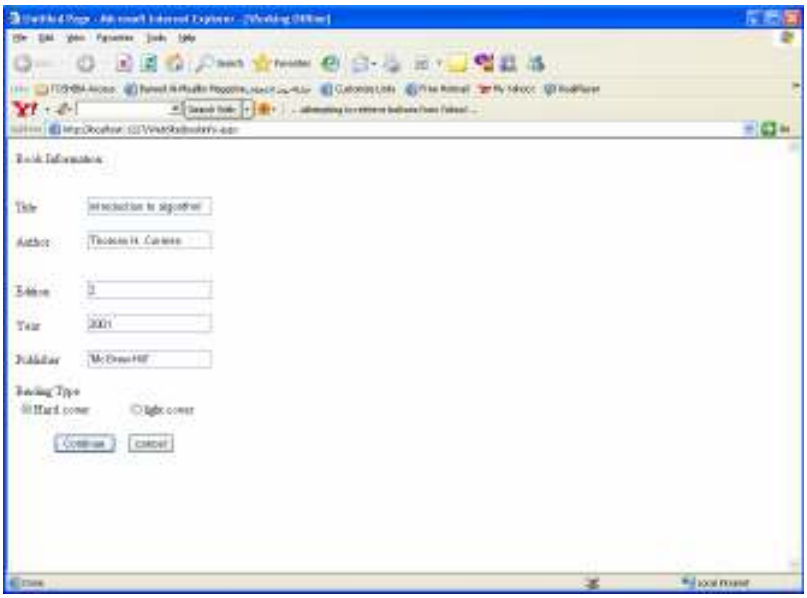


Figure 14: unit test screen shot

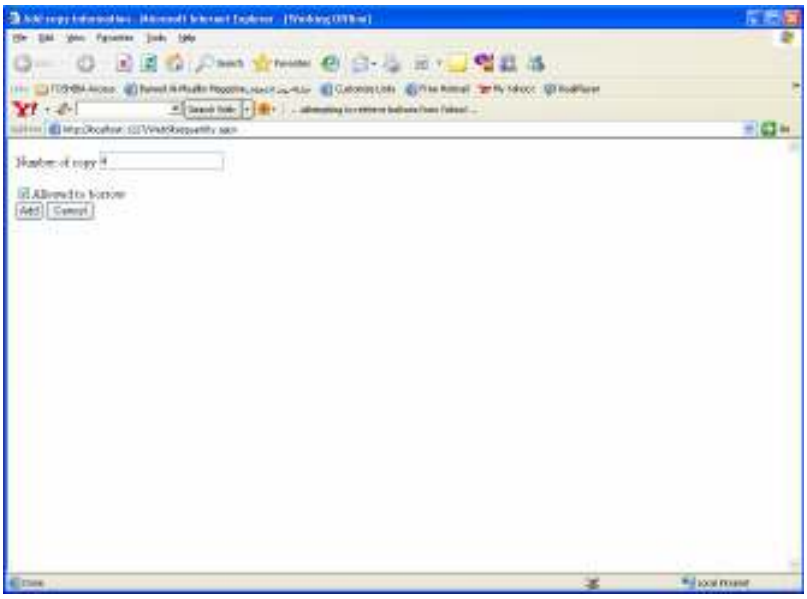


Figure 15: unit test screen shot

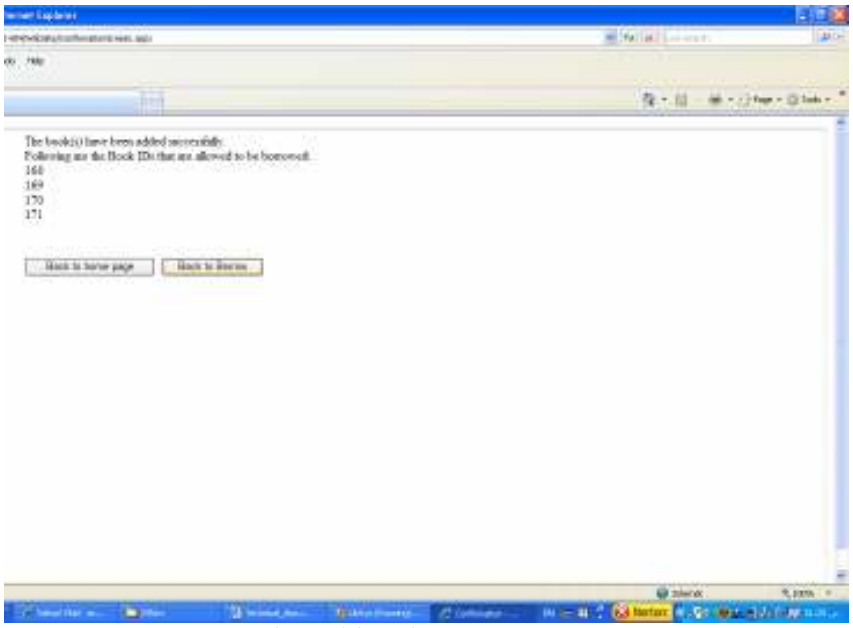


Figure 16: unit test screen shot

Unit Test Case #: 2	**Path #: 2**	**Release #:** 1

Path sequence	Data	result	output	change on DB
1,3,4	Book ISBN = '2201708575' (book available before) Number of copies=6 Allowed to borrow	**pass**	confirmation message (add book is successful) "see screen shots"	a new record with (Book ISBN = '2201708575' Number of copies=6 Allowed to borrow) is saved

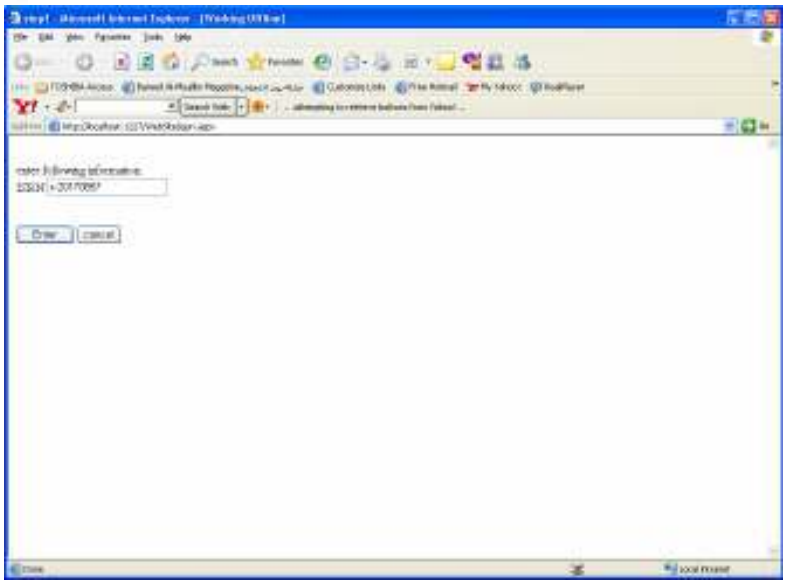


Figure 17: unit test screen shot

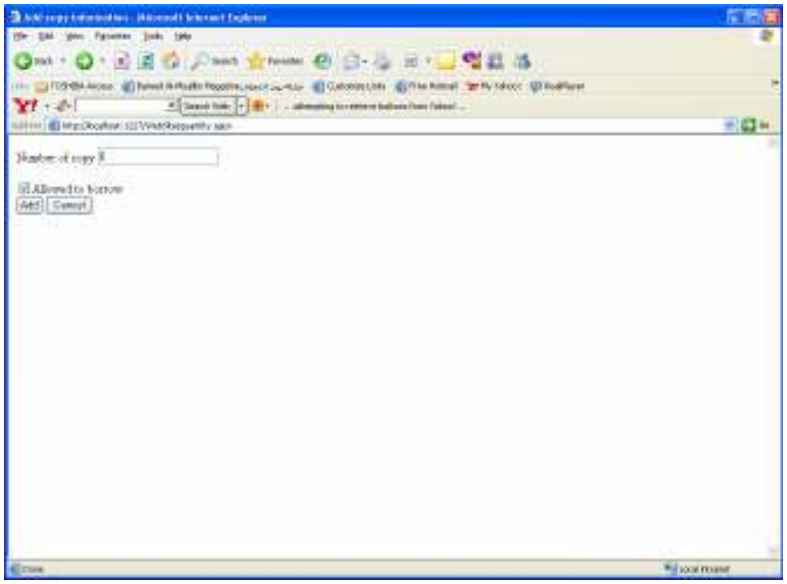


Figure 18: unit test screen shot

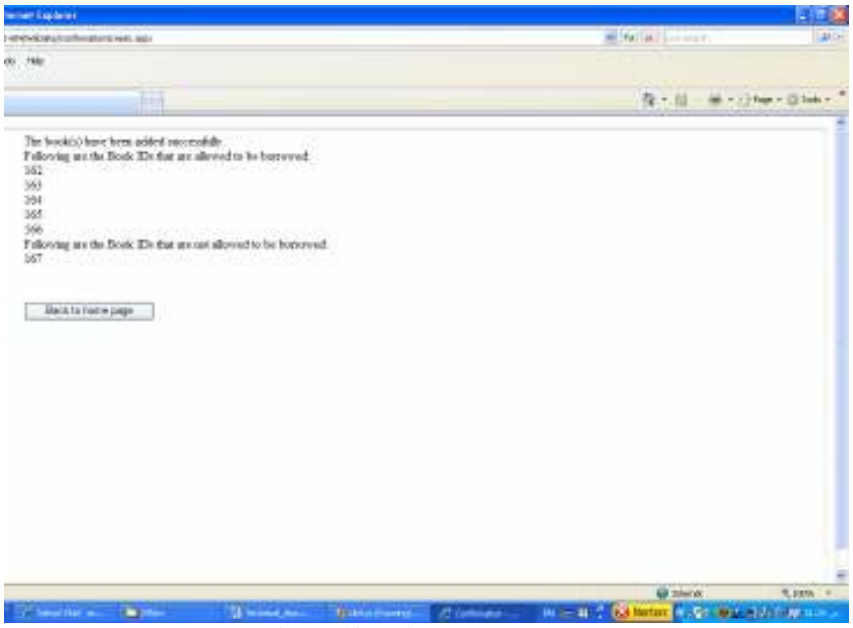
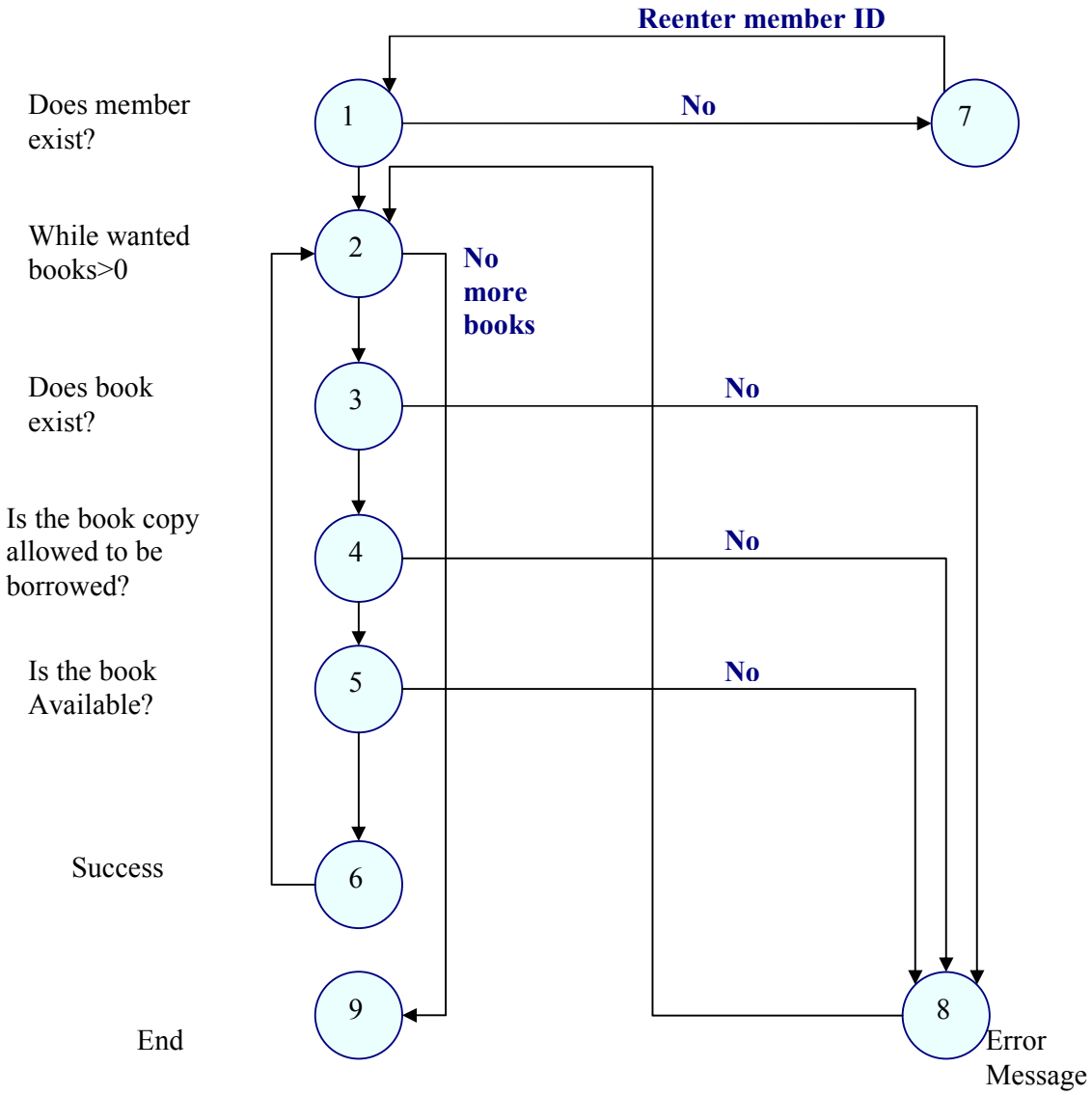


Figure 19: unit test screen shot

4.2.2 Borrow Book:



Cyclomatic complexity = Number of edges -Number of nodes + 2
= 13 – 9 + 2 = 6

Thus 6 independent paths:

Path 1: 1,7,1
Path 2: 1,2,3,4,5,6,2,9
Path 3: 1,2,3,4,5,8,2
Path 4: 1,2,3,4,8,2
Path 5: 1,2,3,8,2
Path 6: 1,2,8

Unit Test Case #: 3	Path #: 1	Release #: 1

Path sequence	Data	result	output	change on DB
1,2,3,4,5,6,2,9	Member ID =2 Book ISBN = 0201708574 Book Copy ID= 70	**pass**	confirmation message (Borrowing Process is successful) "see screen shots"	a new record with (Member ID = 2 , Book ISBN = 0201708574, Copy ID = 70 , today date and time) is saved

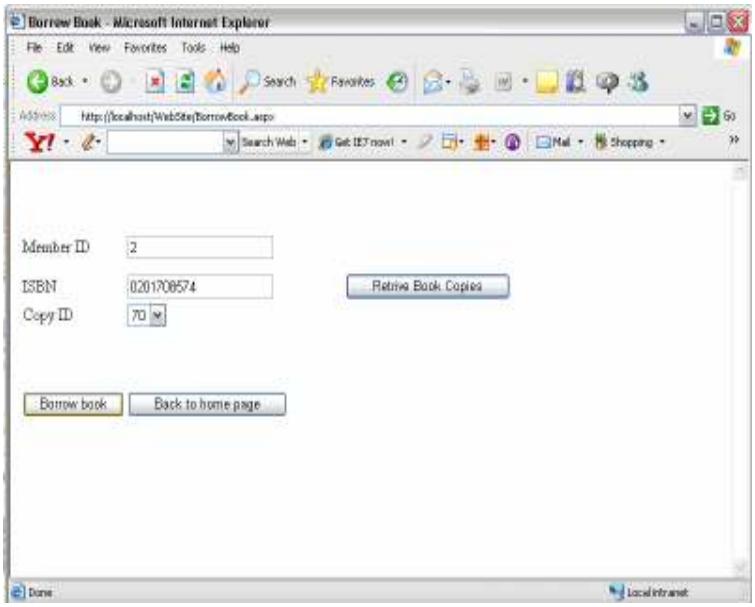


Figure 20: unit test screen shot

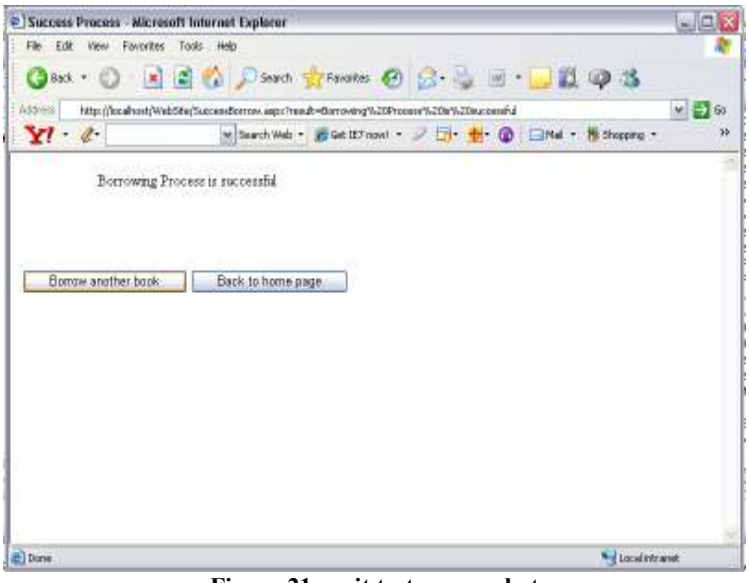


Figure 21: unit test screen shot

Unit Test Case #: 4	**Path #:** 2	**Release #:** 1

Path sequence	Data	result	output	change in DB
1,2,3,4,5,8,2	Member ID =2 Book ISBN = 0201708574 Book Copy ID= 70	**pass**	Error message (The book is borrowed) "see screen shots"	nothing

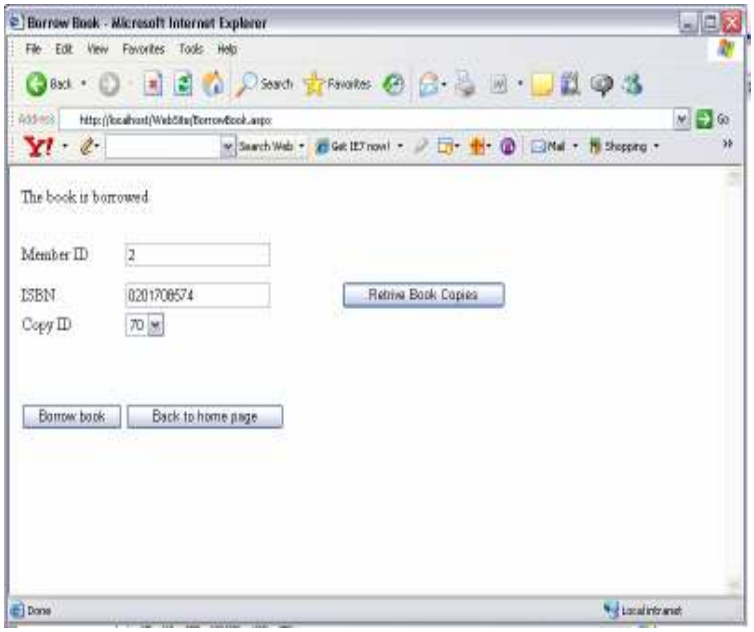


Figure 22: unit test screen shot

Unit Test Case #: 5	Path #: 3	Release #: 1

Path sequence	Data	result	output	change in DB
1,2,3,4,8,2	Member ID =2 Book ISBN = 0201708574 Book Copy ID= 69	**pass**	Error message (The book is not allowed to be borrowed) "see screen shot"	nothing

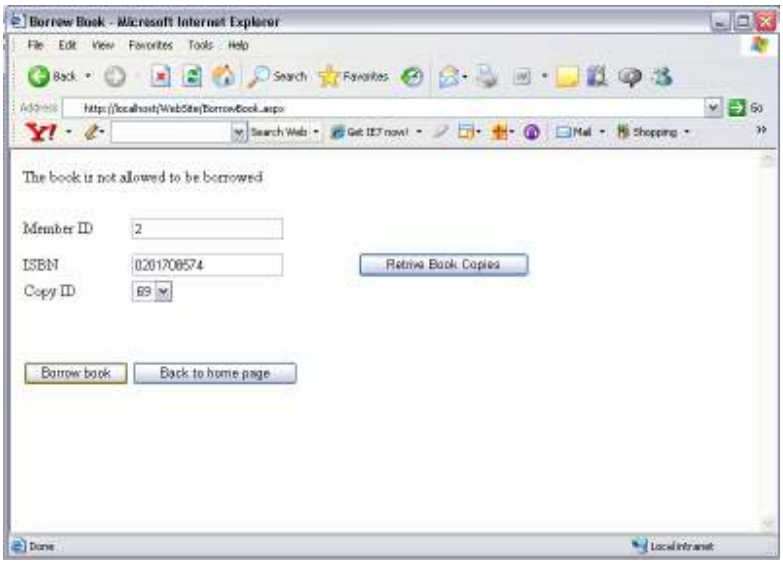
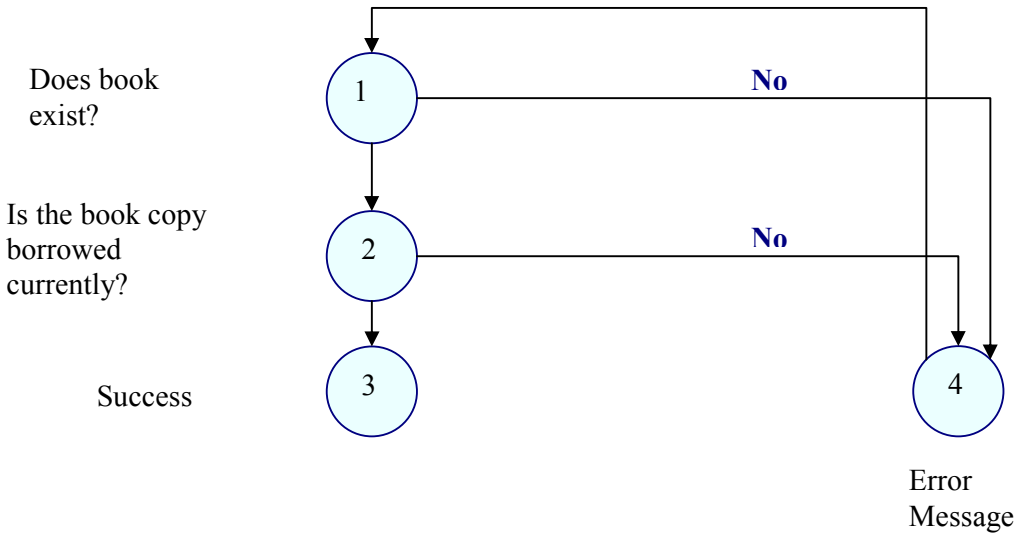


Figure 23: unit test screen shot

Release #	UTC #	Path #	Path sequence	Data	result	output	change in DB
1	**6**	**4**	**1,2,3,8,2**	Member ID =2 Book ISBN = 020170857	**pass**	Error message (Invalid book ISBN)	nothing
1	**7**	**5**	**1,2,8**	press back to home page	**pass**	display home page	nothing
1	**8**	**6**	**1,7,1**	Member ID =33	**pass**	Error message (This member is not registered in the system)	nothing

4.2.3Return Book:



Cyclomatic complexity = Number of edges -Number of nodes + 2
= 5 – 4 + 2 = 3 paths

Thus 4 independent paths:

Path 1: 1, 4
Path 2: 1, 2, 4
Path 3: 1, 2, 3, 4

Unit Test Case #: 9 **Path #:** 1 **Release #:** 1

Path sequance	Data	result	output	change on DB
1, 4	Book ISBN = 123123	**pass**	Error Message Displayed “see screen shots”	none

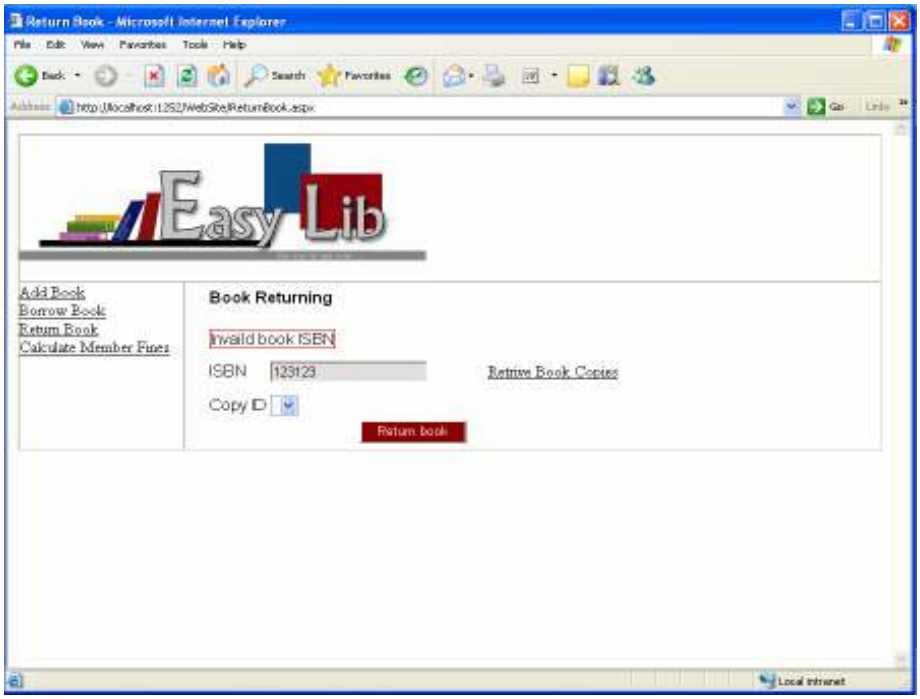


Figure 24: unit test screen shot

Unit Test Case #: 10 **Path #:** 2 **Release #:** 1

Path sequence	Data	result	output	change on DB
1, 2, 4	Book ISBN = 6789 Book Copy ID = 165	**pass**	Error Message Displayed "see screen shots"	none

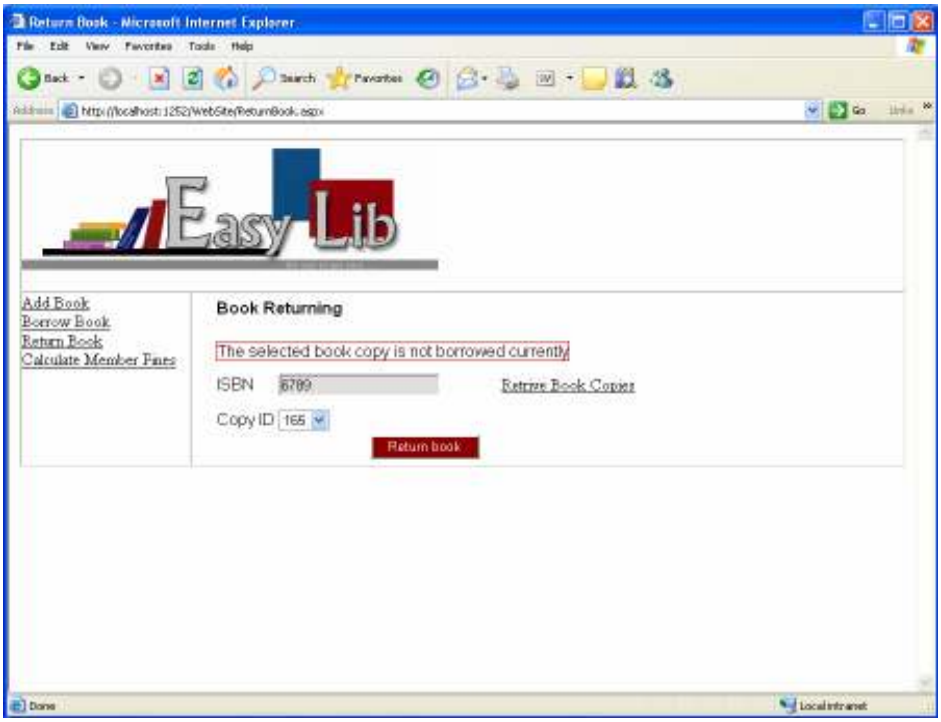


Figure 25: unit test screen shot

Unit Test Case #: 11	**Path #:** 3	**Release #:** 1

Path sequence	Data	result	output	change on DB
1, 2, 3, 4	Book ISBN = 6789 Book Copy ID = 162 Borrowed On = Nov 27,2006	**pass**	Confirmation message with the fine of 8 SR "see screen shots"	Return Date set to Current date (Dec 27, 2006) Fine set to 8 SR

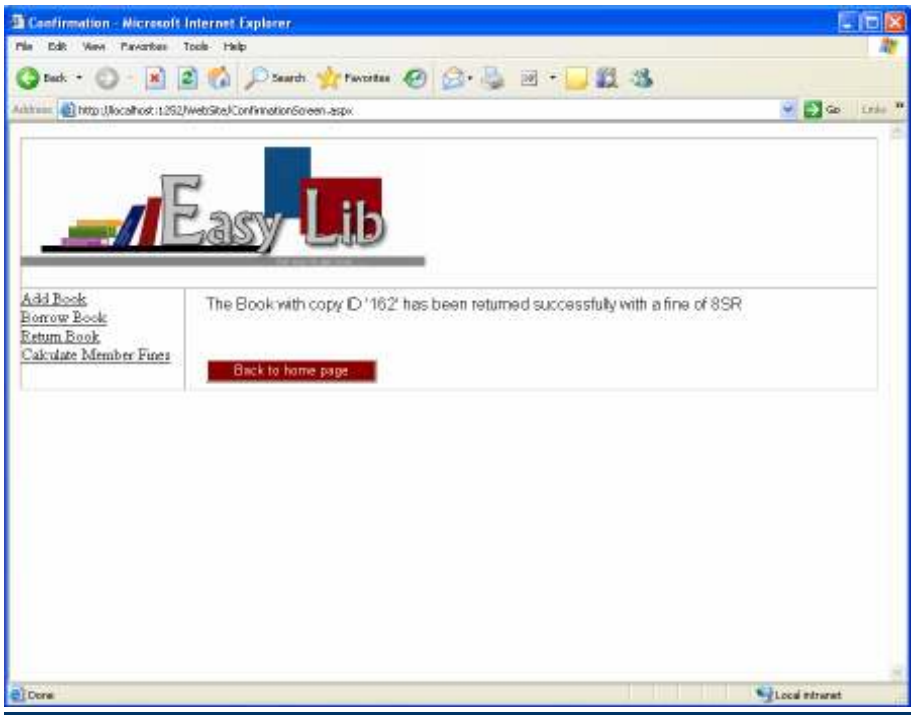
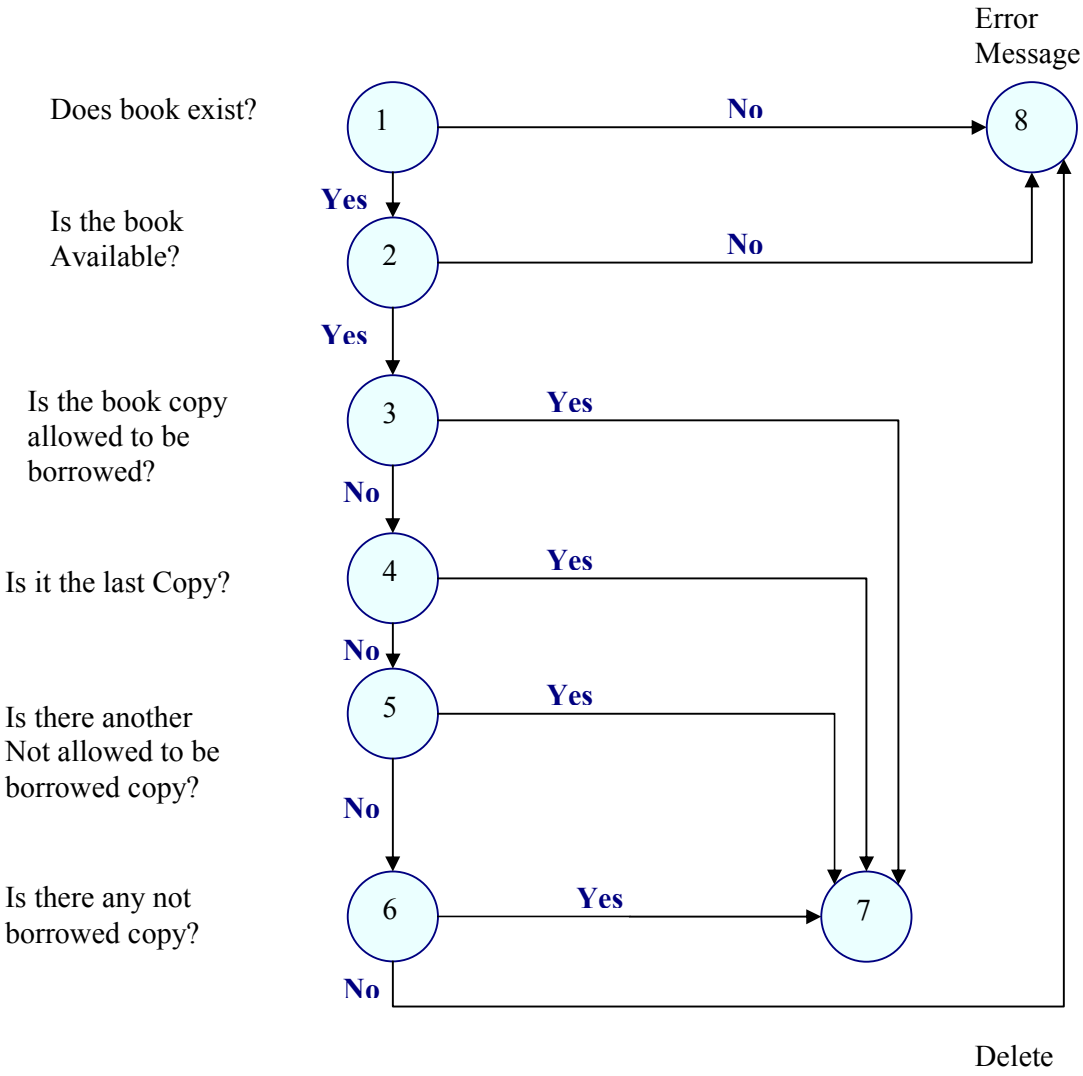


Figure 26: unit test screen shot

4.2.4Delete Book:



Cyclomatic complexity = Number of edges -Number of nodes + 2
= 12 – 8 + 2 = 6

Thus 6 independent paths:

Path 1: 1,2,3,7
Path 2: 1,2,3,4,7
Path 3: 1,2,3,4,5,7
Path 4: 1,2,3,4,5,6,7
Path 5: 1,2,3, 4,5,6,8
Path 6: 1,2,8

Unit Test Case #: 12	**Path #:** 1	**Release #:** 2

Path sequence	Data	result	output	change on DB
1,2,3,7	Book ISBN = 0201708574 Book Copy ID= 206 Comment="It wasn't possible to be borrowed any more"	**pass**	confirmation message (The Book has been deleted successfully) "see screen shots"	The Deleted flag of the book with ISBN=0201708574 Copy ID= 206 is changed to true and the comment is inserted.

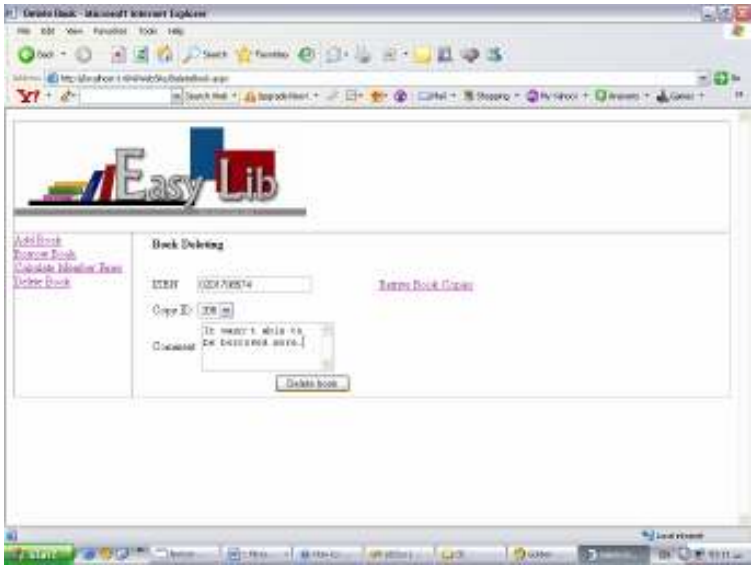


Figure 27: unit test screen shot

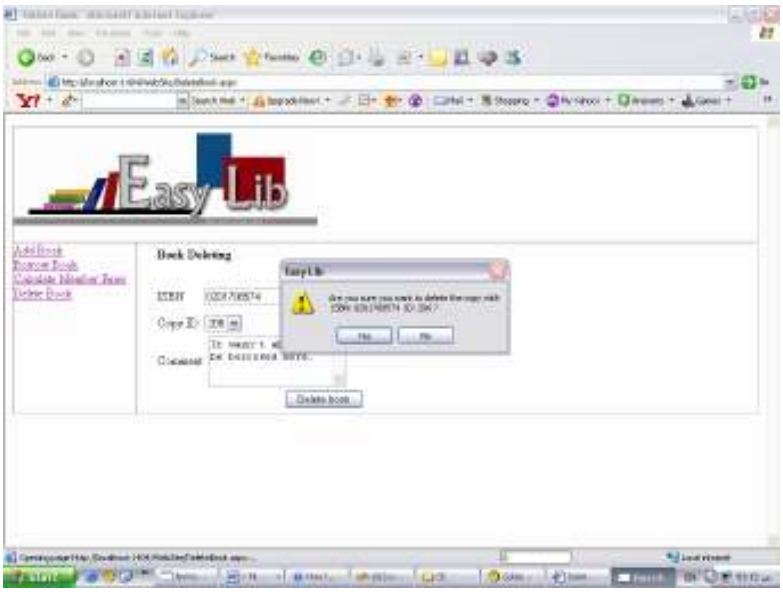


Figure 28: unit test screen shot

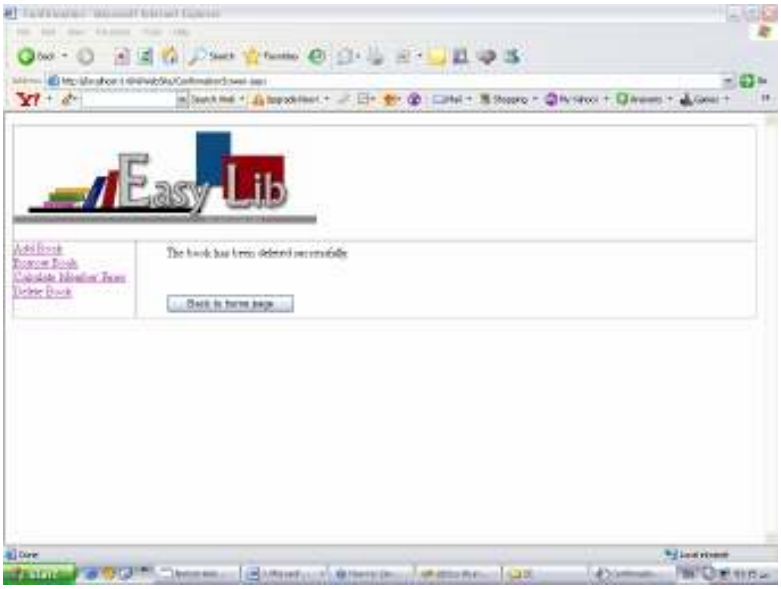


Figure 29: unit test screen shot

Unit Test Case #: 13	**Path #:** 2	**Release #:** 2

Path sequence	Data	result	output	change in DB
1,2,3,4,7	Book ISBN = 1230012401 Book Copy ID= 221 Comment = “Last Copy can’t survive more”	**Pass**	confirmation message (The Book has been deleted successfully) “see screen shots”	The Deleted flag of the book with ISBN=1230012401 Copy ID= 221 is changed to true and the comment is inserted.

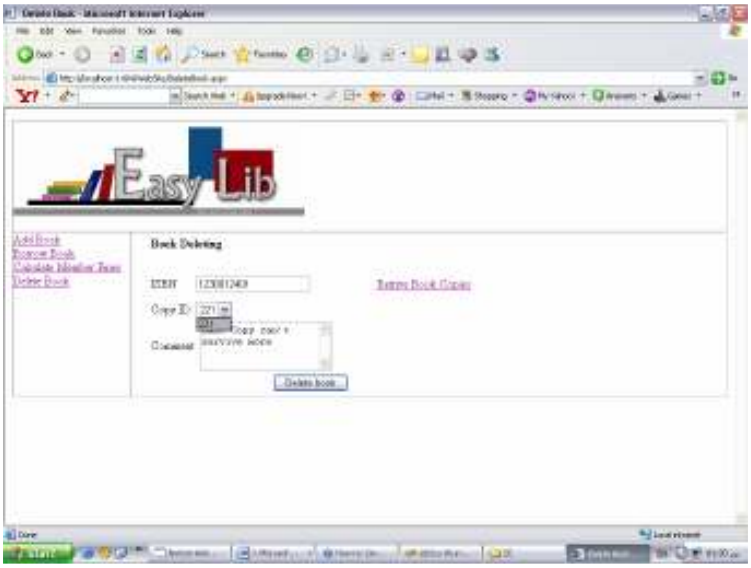


Figure 30: unit test screen shot

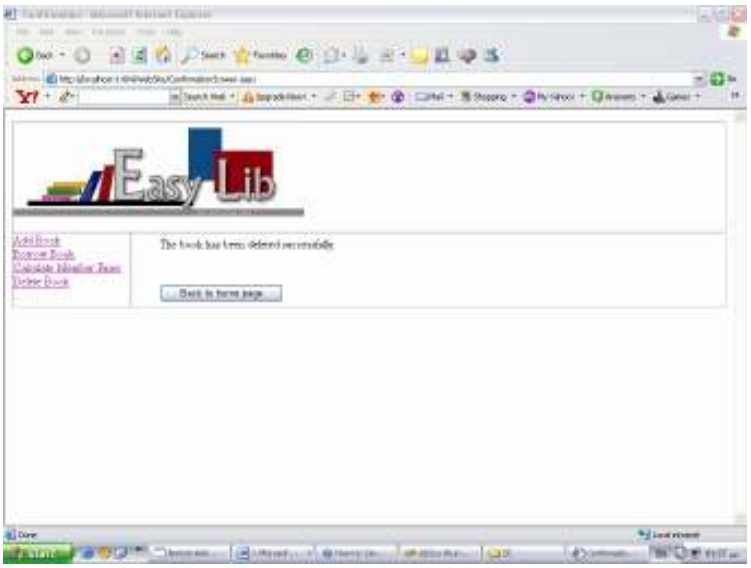


Figure 31: unit test screen shot

Unit Test Case #: 15 **Path #:** 4 **Release #:** 2

Path sequence	Data	result	output	change in DB
1,2,3,4,5,6,7	Book ISBN = 0201708574 Book Copy ID= 203 Comment = "Last Not Allowed, find another copy"	**pass**	confirmation message (The Book has been deleted successfully) "see screen shot"	The Deleted flag of the book with ISBN=0201708574 Copy ID= 203 is changed to true and the comment is inserted.

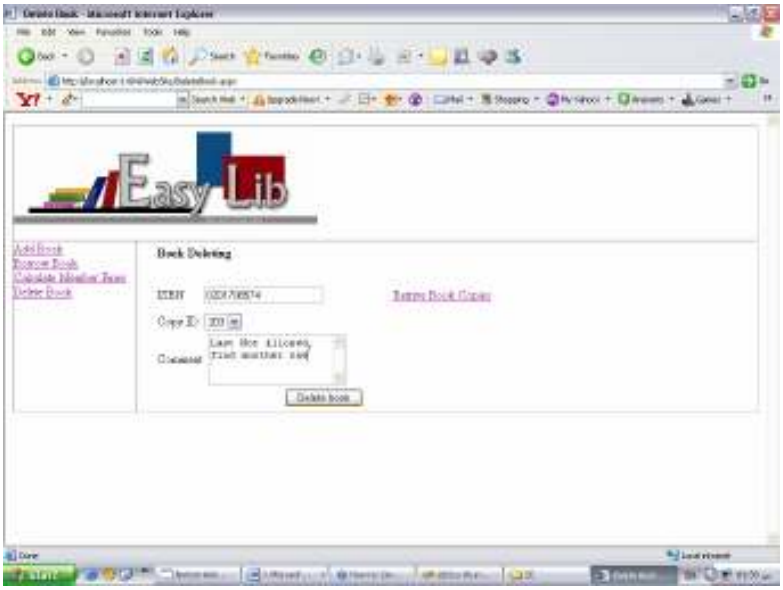


Figure 32: unit test screen shot

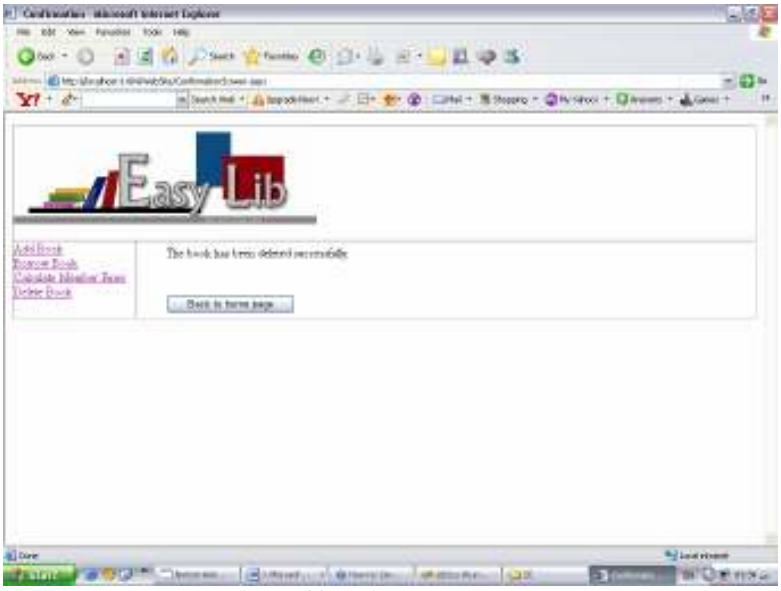


Figure 33: unit test screen shot

Unit Test Case #: 16	Path #: 5	Release #: 2

Path sequence	Data	result	output	change in DB
1,2,3,4,5,6,8	Book ISBN = 1122002200 Book Copy ID= 235 Comment = "Delete"	**pass**	Error message (You can't delete this copy at this time, because all other copies are borrowed) "see screen shot"	No changes

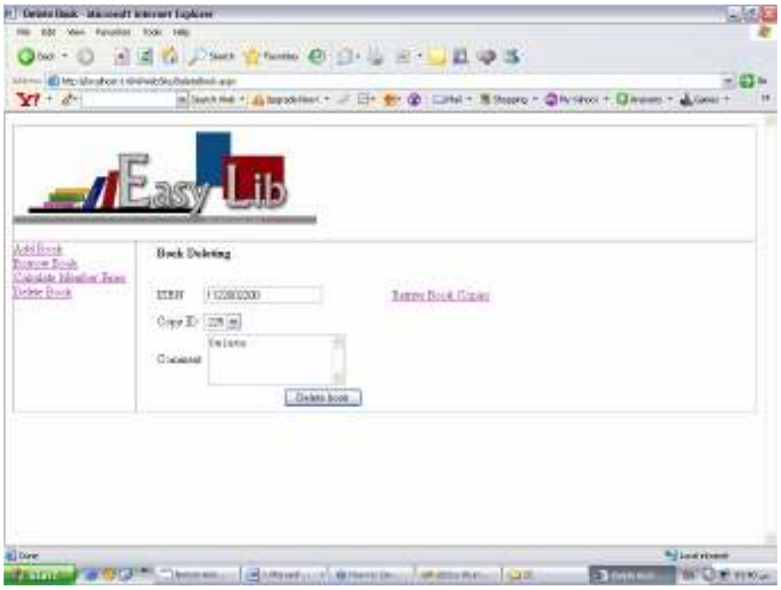


Figure 34: unit test screen shot

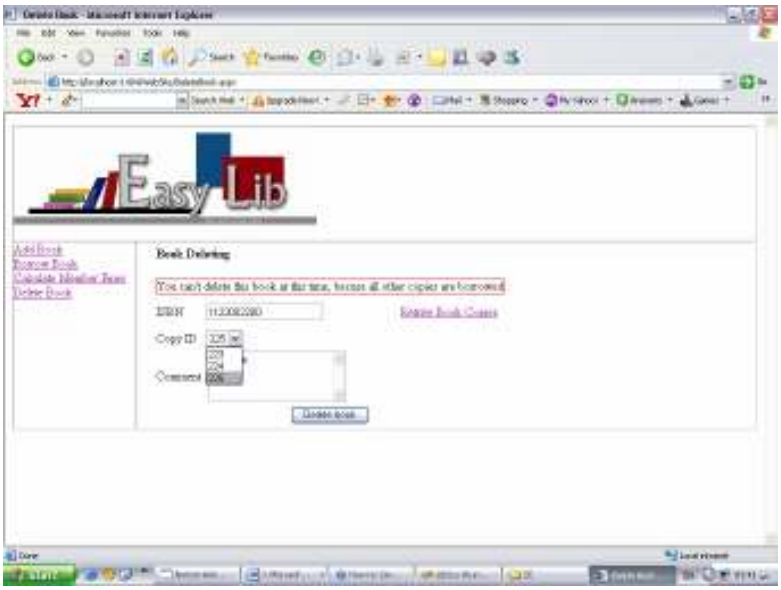


Figure 35: unit test screen shot

Release #	UCT #	Path #	Path Sequence	Data	result	output	change in DB
2	**14**	**3**	**1,2,3,4,5,7**	Book ISBN = 11220344301 Book Copy ID= 274 Comment = "Delete"	**pass**	confirmation message (The Book has been deleted successfully) "see screen shot"	The Deleted flag of the book with ISBN=11220344301 Copy ID= 274 is changed to true and the comment is inserted.
2	**17**	**6**	**1,2,8**	Book ISBN = 1122002200 Book Copy ID= 234 Comment = "Delete"	**pass**	Error message (The Book is borrowed)	nothing

4.3Integration Testing (black-box)

Test plan

Integration testing will be applied on the over all system, by running the following scenario:

1. Add a book with 4 copies
2. Borrow the Not Allowed to be borrowed book copy.
3. Borrow one of the allowed to be borrowed book copies.
4. Delete the borrowed book copy.
5. Return the borrowed book copy late.
6. Delete the returned book copy
7. Delete the book copy that is not allowed to be borrowed.
8. Borrow the alternate book copy that is not allowed to be borrowed.
9. Borrow and return another book copy late.
10. Get the total amount of fines on the borrowing member.
11. Update the book information for the added book.
12. Update the book information for a non-existening book.
13. Delete all copies of the added book.
14. Update the book information of the added book

Test Case #: IT.13 **Test Case Name:** Scenario1 **Page:** 1 of 2

System: Library System **Subsystem**: Add book, Update book, Delete Book Borrow book, Return book, Get Member fines

Designed by: Eidah & Sumayah **Design Date:** 26/12/2006

Executed by: Eidah & Sumayah **Execution Date:** 28/12/2006

Short Description: Test the integrated system.

Pre-conditions:
1. Member with ID ”6” exists and the amount of fines on that member is 0
2. Book with ISBN does not exist in the system.

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Add a book with the following information: ISBN:10101010 Title: Calculus Authors: Olinick& Pence Edition:6 Year:2000 Publisher: Brooks #Copies: 4	The system displays a confirmation message with the book’s copy IDs and which copies are not to be borrowed.	**Pass**	See Figure 36 Figure 37
2	Borrow the book copy with ID “185”	The system displays a message stating that the book is not allowed to be borrowed	**Pass**	See Figure 38
3	Borrow the book copy with ID “182”	The system displays a message stating that the borrowing operation has completed successfully	**Pass**	See Figure 39
4	Delete Book Copy with ID “182”	The system displays a message stating that the book can’t be deleted because it’s borrowed currently.	**pass**	See Figure 40
5	Return the book Copy with “182” on 16 Jan,2007	The system displays a message stating that the book has been returned successfully and the amount of fine earned is “1.5 SR”.	**pass**	See Figure 41
6	Delete Book Copy with ID “182”	The system displays a message stating that the book has been deleted successfully.	**pass**	See Figure 42
7	Delete the book copy with ID “185”	The system displays a message stating that the book has been deleted successfully and states the alternate book copy that is not to be borrowed anymore.	**pass**	See Figure 43
8	Borrow the book copy with ID “183”	The system displays a message stating that the book is not allowed to be borrowed	**pass**	See Figure 44
9	Borrow the book copy with ID “184”	The system displays a message stating that the borrowing operation has completed	**pass**	

		successfully.		
10	Return the book copy with ID “184” on 27 Jan, 2007	The system displays a message stating that the book has been returned successfully and the amount of fine earned is “7 SR”.	**pass**	
11	Get the total amount of fines on Member with ID “6”	The system should displays a message stating the amount of fines is “8.5 SR”	**pass**	See Figure 45
12	Update book information for the ISBN “101010” By changing the cover to normal cover and the year to 2003	The system displays a message stating that the update has been completed successfully.	**pass**	See Figure 46 Figure 47
13	Open the Update screen for the book with ISBN “101010”	The system displays a window containing the current book information which are the information updated at step12	**pass**	See Figure 48
14	Update book information for the ISBN “654321”	The system displays a message stating that the book does not exist in the system.	**pass**	See Figure 49
15	Delete all Book copies for the ISBN “101010”	The system displays a message stating that the book deletions have been performed successfully.	**pass**	
16	Update book information for the ISBN “101010”	The system displays a message stating that the book does not exist in the system.	**pass**	See Figure 50
17	Add 3 Book copies for the ISBN “101010”	The system displays a message stating that 3 book copies have been added and also, states which book copy is set not to be allowed to be borrowed.	**Pass**	See Figure 51 Figure 52
18	Update book information for the ISBN “101010”	The system displays a form containing the current book information.	**pass**	See Figure 48
19	**Check post-condition 1**			

Post-conditions 1:
All messages stated in the Expected system response are displayed.

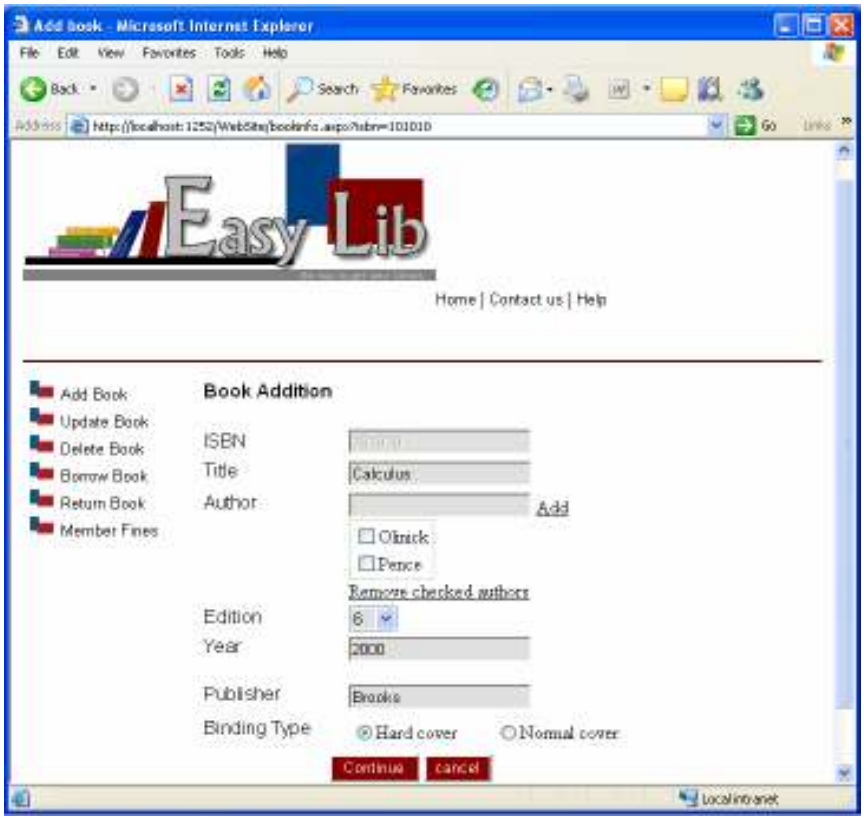


Figure 36: Integration test screen shot

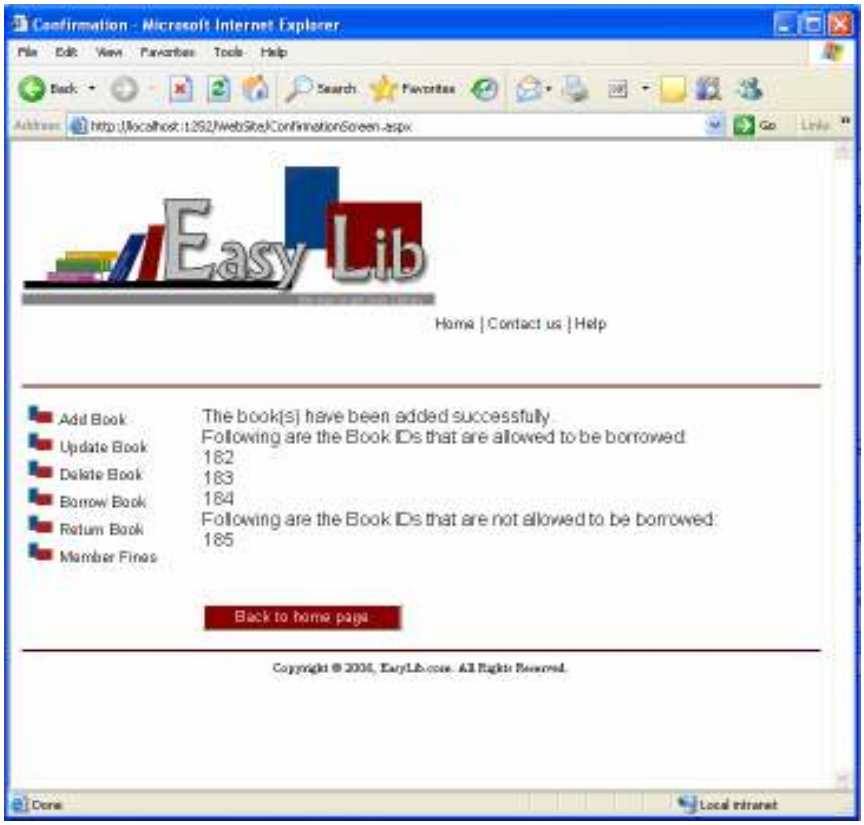


Figure 37: Integration test screen shot

Figure 38: Integration test screen shot

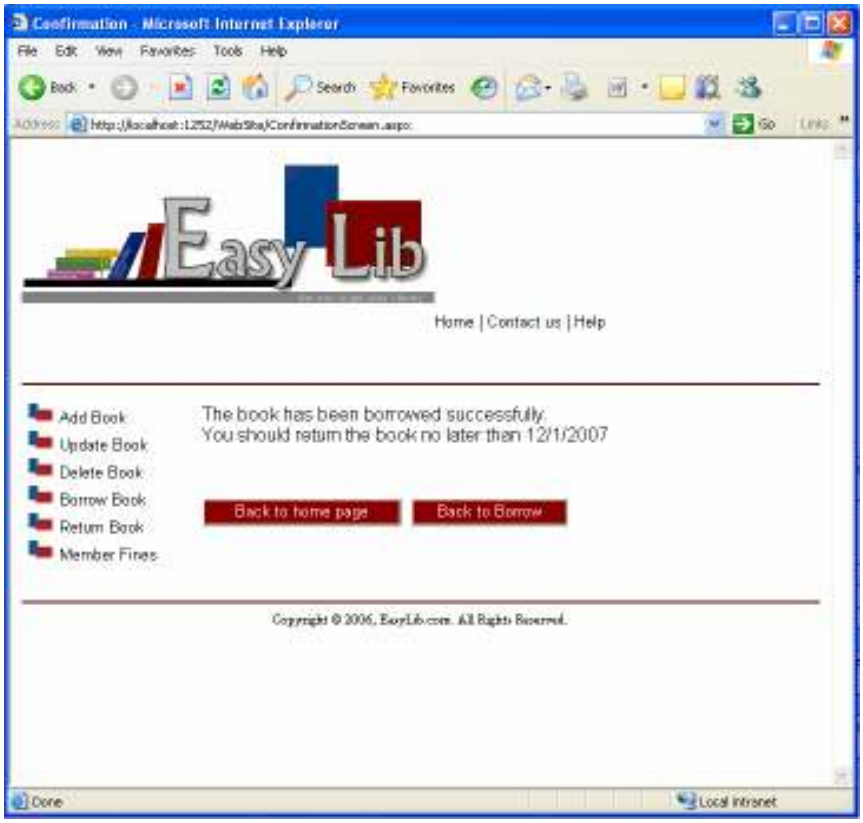


Figure 39: Integration test screen shot

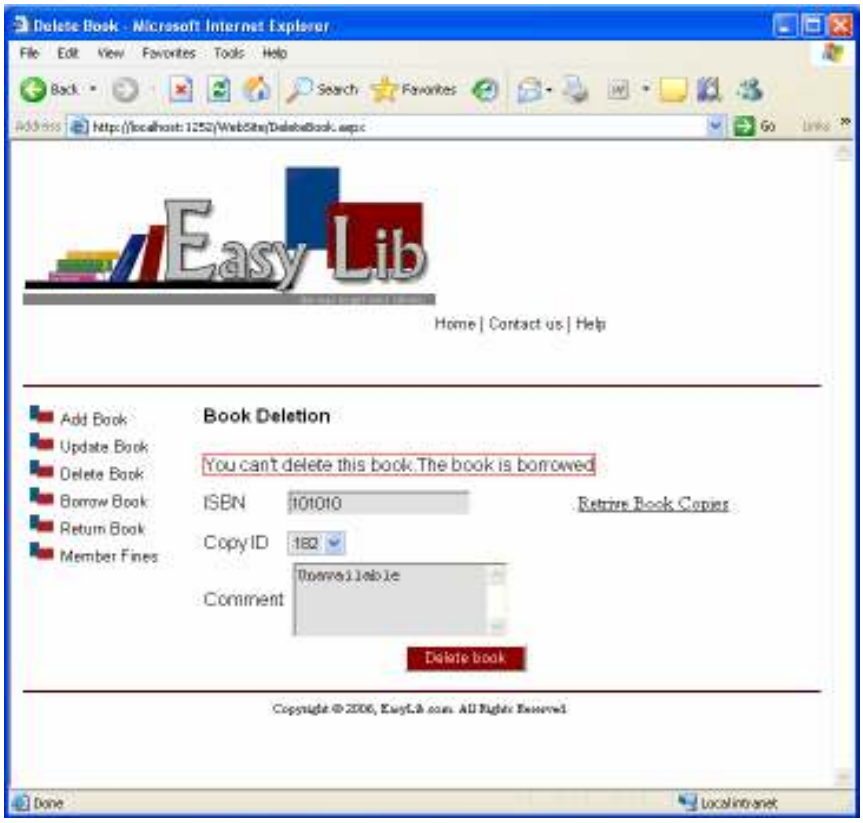


Figure 40: Integration test screen shot

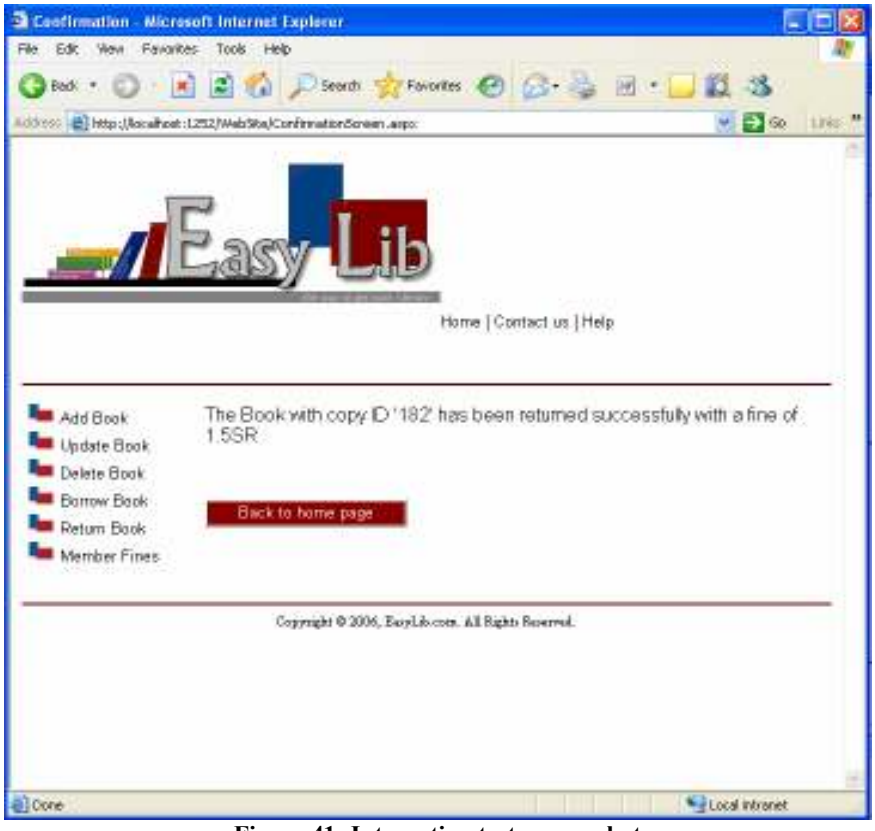


Figure 41: Integration test screen shot

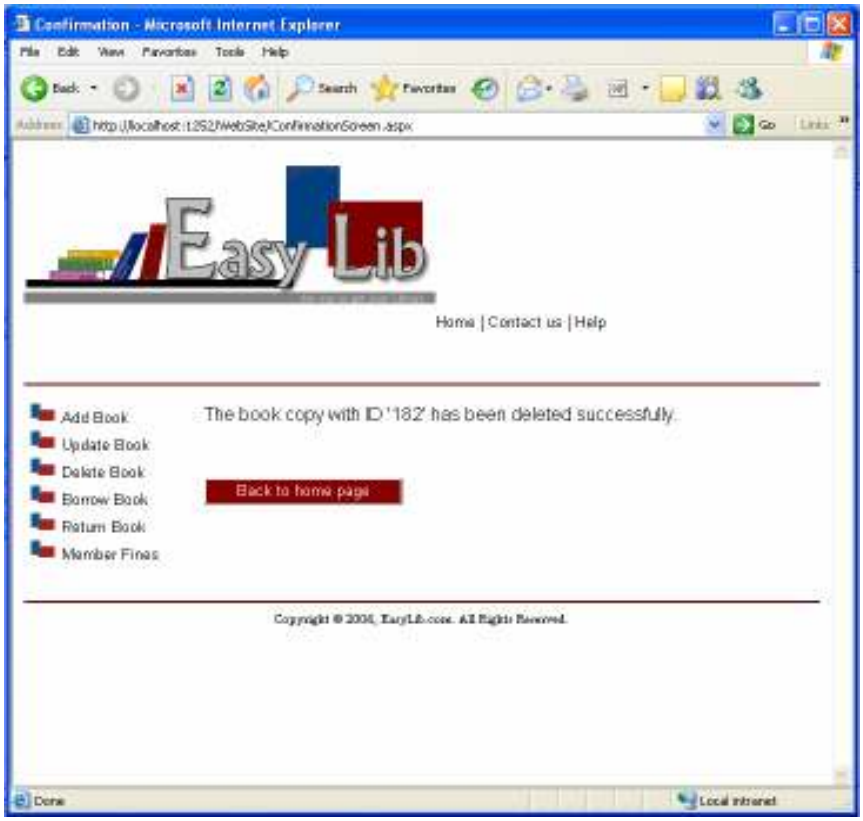


Figure 42: Integration test screen shot

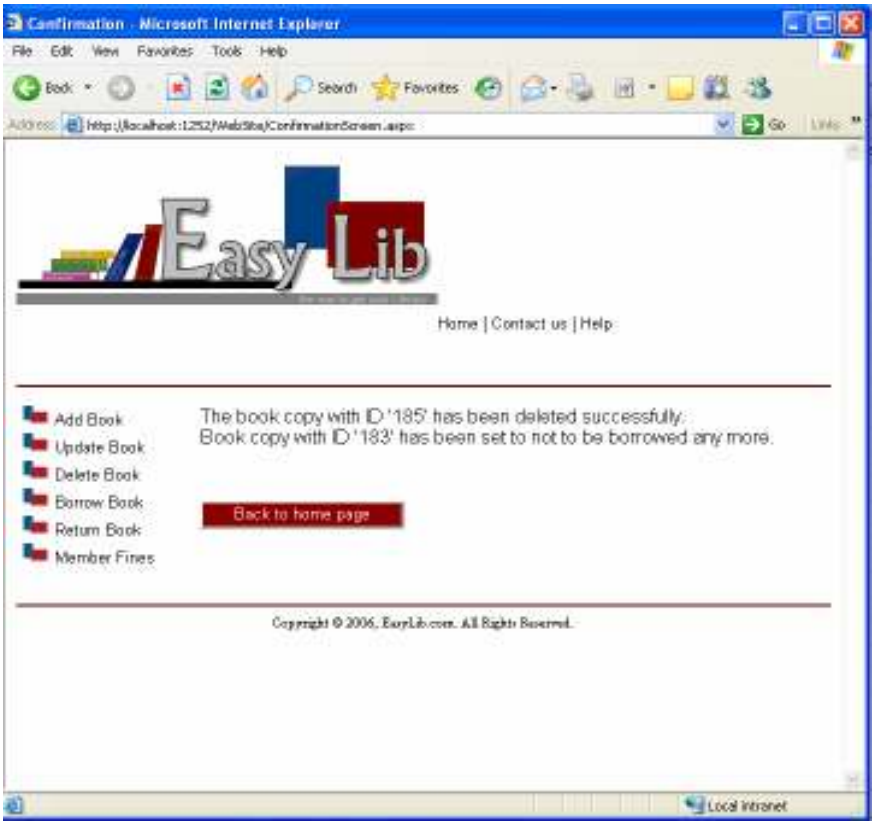


Figure 43: Integration test screen shot

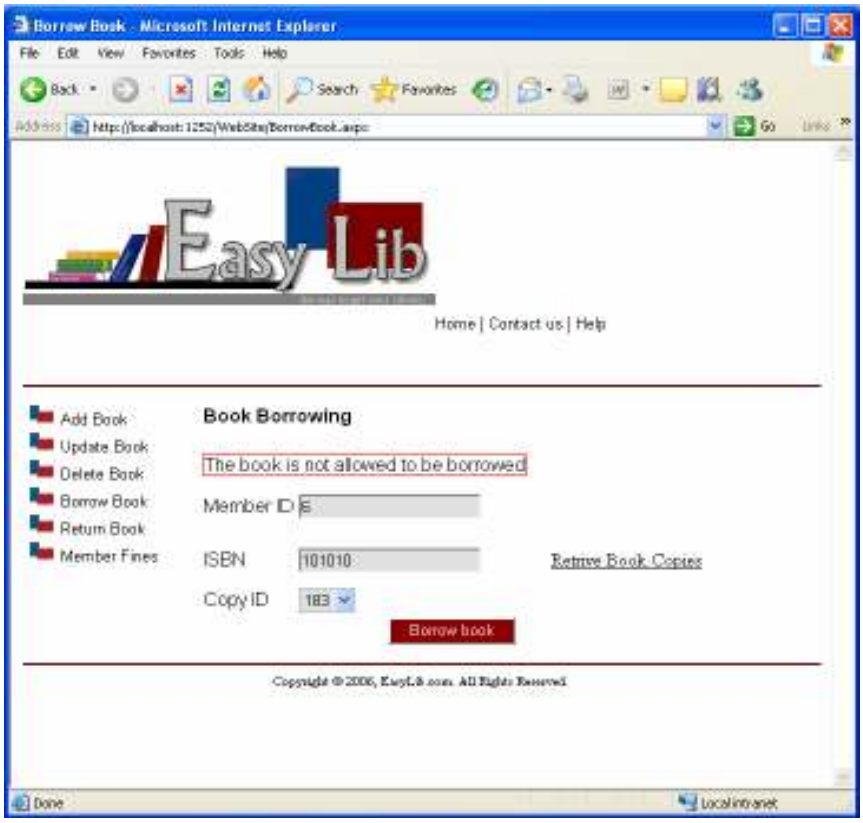


Figure 44: Integration test screen shot

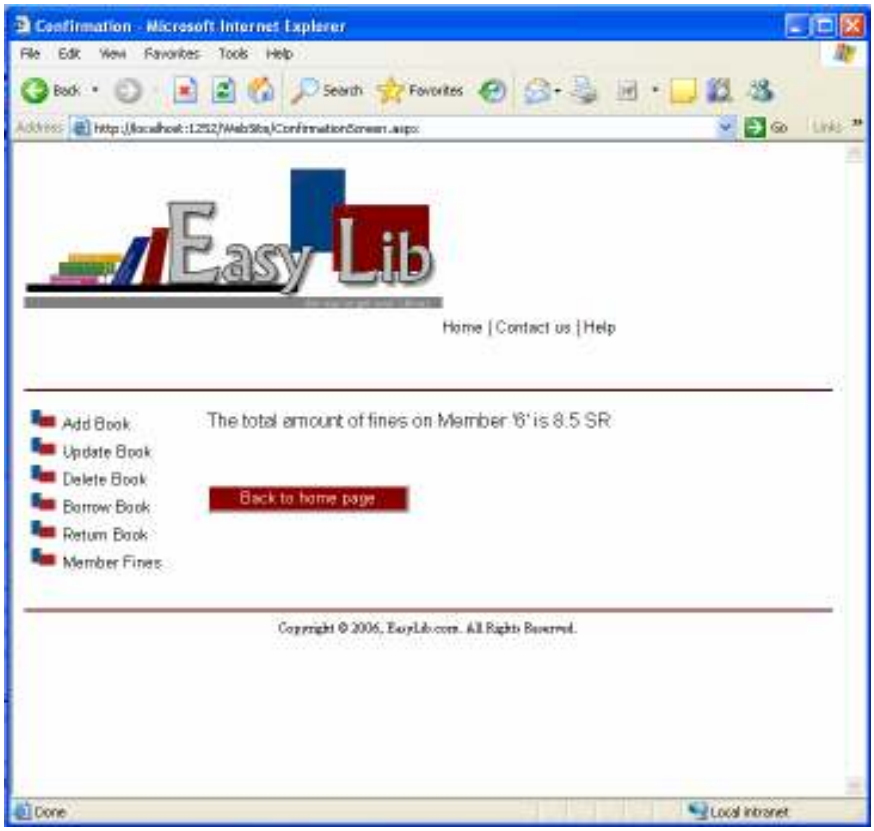


Figure 45: Integration test screen shot

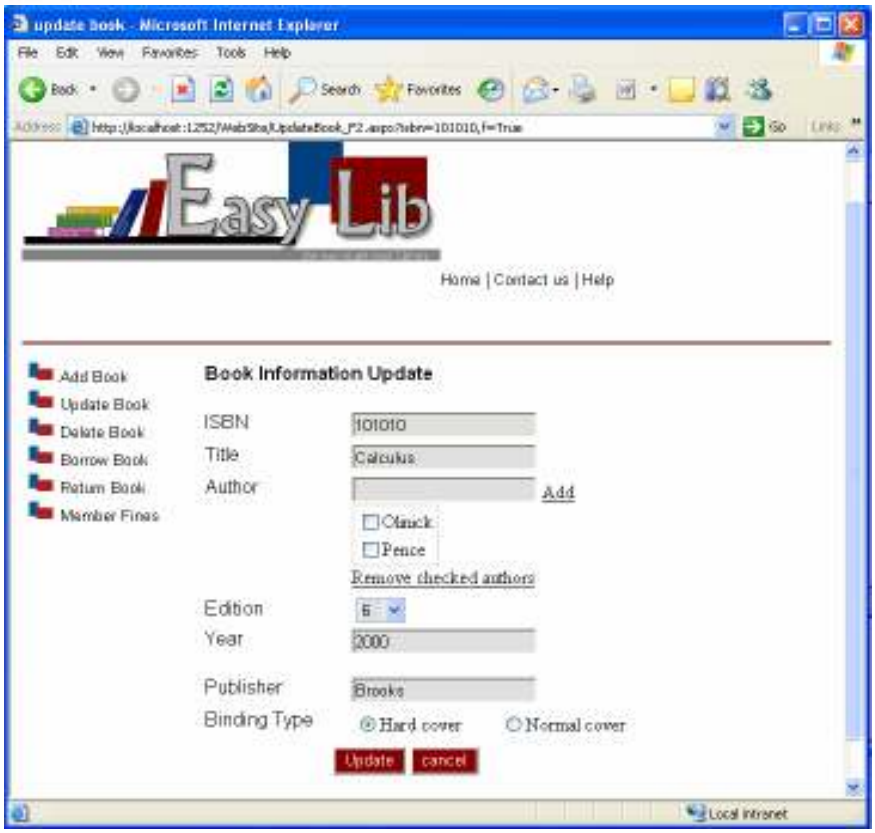


Figure 46: Integration test screen shot

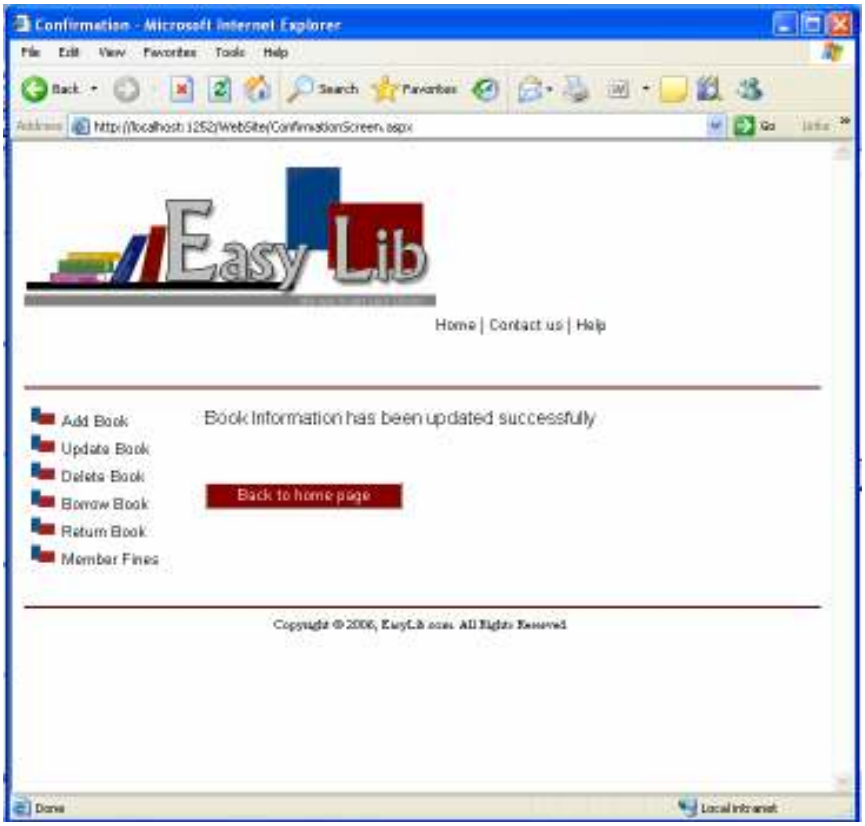


Figure 47: Integration test screen shot

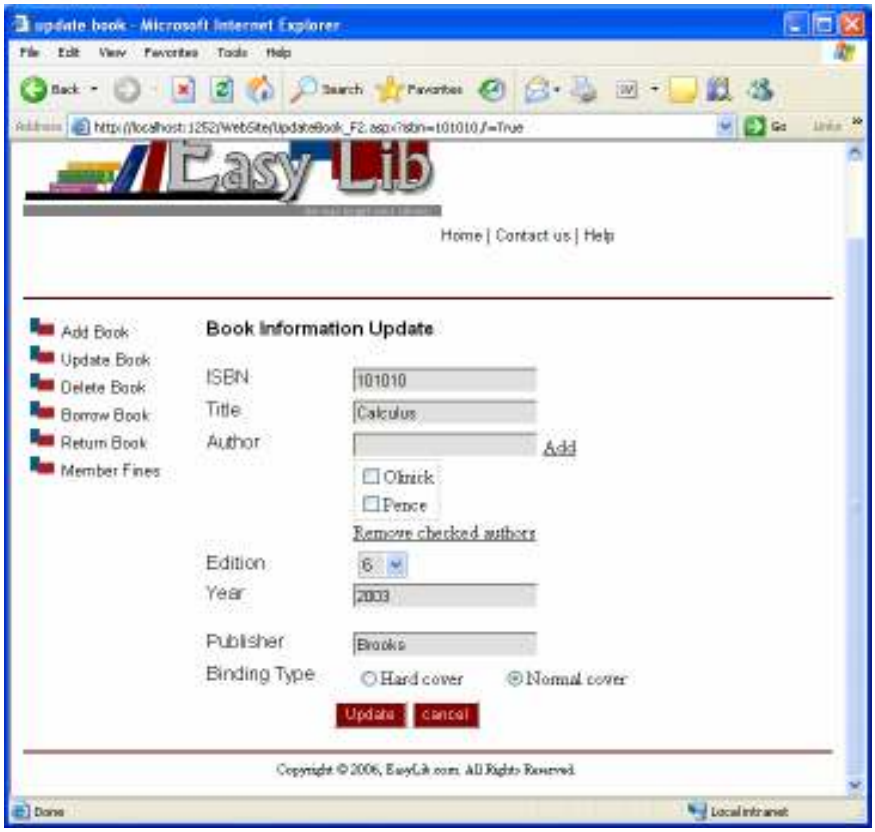


Figure 48: Integration test screen shot

Figure 49: Integration test screen shot

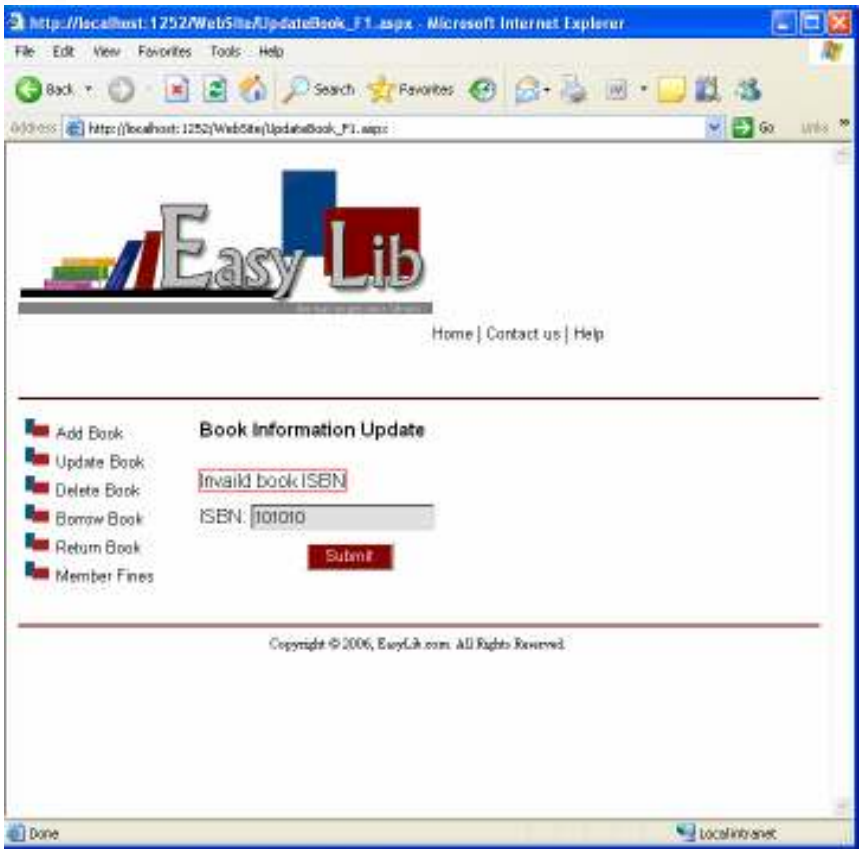


Figure 50: Integration test screen shot

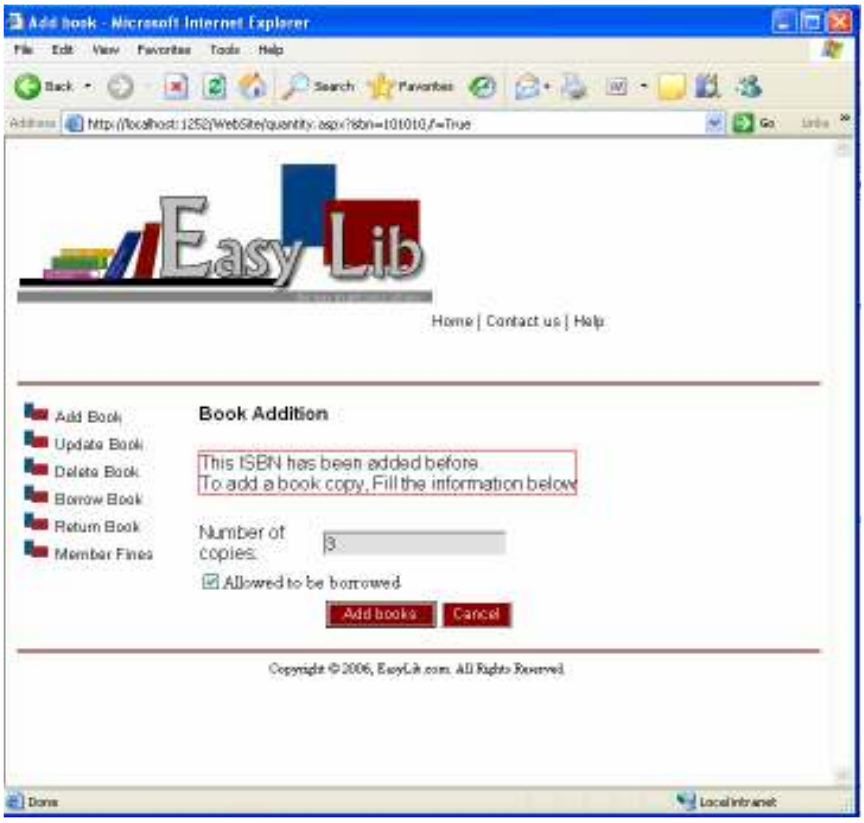


Figure 51: Integration test screen shot

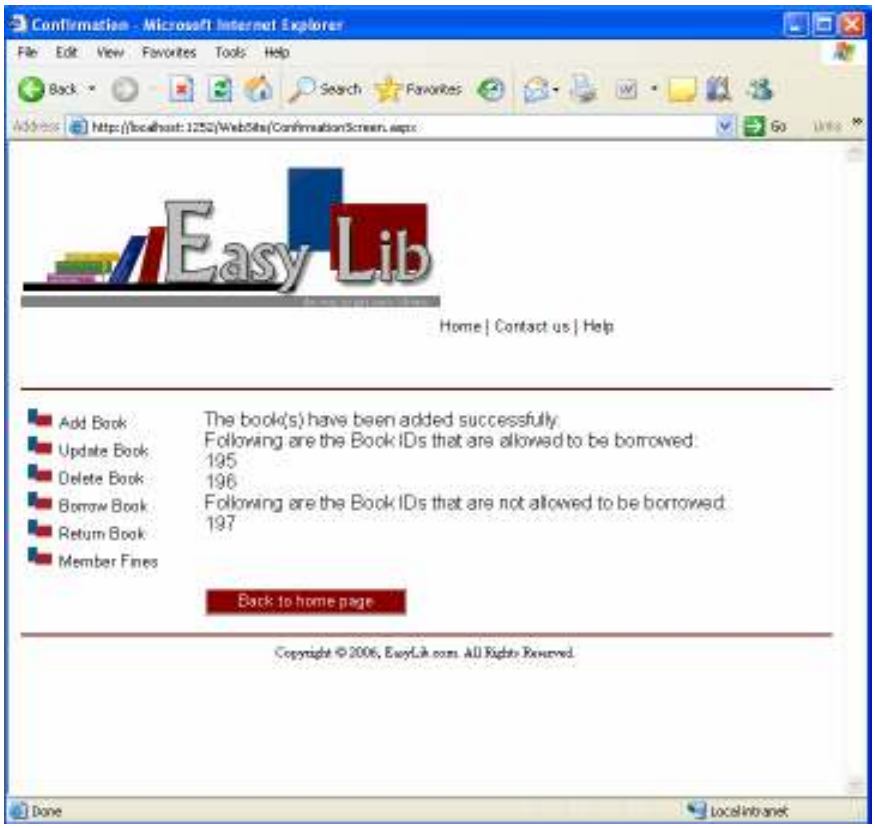


Figure 52: Integration test screen shot

Chapter 5

Release Measurement & Metrics

5.1 Matrices:

5.1.1 Functional Point:

5.1.1.1 Calculate system: adjustment complexity factor:

14 Fi Factors:

#	Question	Scale
1	Does the system require reliable backup and recovery?	4
2	Are specialized data communications required to transfer information to or from the application?	0
3	Are there distributed processing function?	0
4	Is performance critical?	3
5	Will the system run in an existing, heavily utilized operational environment?	3
6	Does the system require on-line data entry?	5
7	Does the on-line data entry the input transaction to be built over multiple screens or operation?	5
8	Are the internal files updated online?	5
9	Are the input, outputs, files, inquires complex?	3
10	Is the internal processing complex?	3
11	Is the code designed to reusable?	5
12	Are the conversion and installation included in the design?	1
13	Is the system designed for multiple installations in different organization?	4
14	Is the application designed to facilitate change and for ease of use by the user?	4
	∑ Fi	**45**

Table 3: 14 Fi Factors For Add book

Adjustment complexity factor = $[0.65+ (0.01 \times \sum (Fi))]$

$= [0.65+ (0.01 \times 45)]$

$= 1.1$

5.1.1.2 Weighting Factor Complexity:

Faction Points based on a combination of program characteristics:

1. Internal logical File (ILF): Each identified ILF must be assigned a functional complexity of Simple, Average or High based upon the number of Data Element Types (DETs) and Record Element Types (RETs) associated with that ILF.

- Data element types (DETs) are unique user recognizable, non-repeating fields/attributes, including foreign key attributes, maintained on the ILF.
- Record element types (RETs) are user recognizable subgroups (optional or mandatory) of data elements contained within an ILF. Subgroups are typically represented in an entity relationship diagram as entity subtypes or attributive entities, commonly called parent-child relationships.
- DETs and RETs are applied to the following matrix in order to determine the functional complexity of each ILF:

ILF and EIF Complexity Matrix			
RETs	1-19 DETs	20-50 DETs	51+ DETs
0-1	Simple	Simple	Avg
2-5	Simple	Avg	High
6+	Avg	High	High

2. External interface File (EIF): Record Element Types and Fields/Attributes read/referenced are counted in the same way as for ILFs in order to determine the complexity of each EIF. The same matrix used for sizing ILFs is used to size EIFs.

3. External Input (EI): Each identified EI must be assigned a functional complexity a functional complexity of Simple, Average or High based upon the number of Data Element Types (DETs) and File Types Referenced (FTRs) associated with the EI**.**

- Data element types (DETs) are usually unique user recognizable, non-repeating fields/attributes, including foreign key attributes.
- File types referenced (FTRs), or more simply files referenced, totals the number of internal logical files (ILFs) maintained, read or referenced and the external interface files read or referenced by the EI transaction.
- DETs and FTRs are applied to the following matrix in order to determine the functional:

EI Complexity Matrix			
FTRs	1-4 DETs	5-15 DETs	16+ DETs
0-1	Simple	Simple	Avg
2	Simple	Avg	High
3+	Avg	High	High

4. External Output (EO): Each identified EO must be assigned a functional complexity of Simple, Average or High based upon the number of Data Element Types (DETs) and File Types Referenced (FTRs) associated with the EO**.**

- Data element types (DETs) are usually unique user recognizable, non-repeating fields/attributes, including foreign key attributes that exit the boundary of the subsystem/application.
- File types referenced (FTRs), or more simply files referenced, totals the number of internal logical files (ILFs) read or referenced and the external interface files read or referenced by the EO transaction.
- DETs and FTRs are applied to the following matrix in order to determine the functional complexity of each EO:

EO and EQ Complexity Matrix			
FTRs	1-5 DETs	6-19 DETs	20+ DETs
0-1	Simple	Simple	Avg
2-3	Simple	Avg	High
4+	Avg	High	High

5. External Inquiry (EQ): Data Element Types (DETs) and File Types Referenced (FTRs) are counted in the same way as for EO in order to determine the complexity of each EQ. The same matrix used for sizing EOs is used to size EQs [9].

- *Note*
  -DETs are equivalent to non-repeated fields or attributes.
  -RETs are equivalent to mandatory or optional sub-groups contained within an ILF. Subgroups are typically represented in an entity relationship diagram as entity subtypes or attributive entities, commonly called parent-child relationships.
  -FTRs are equivalent to ILFs referenced by that transaction[9].

Example "Borrow Book":

Information Domain Value	Count		Weighting Factor				
			Simple	Average	complex		
Number of user inputs (EI)	3	×	3	4	6	=	9
Number of user outputs (EO)	4	×	4	5	7	=	16
	1	×	4	5	7	=	5
Number of user inquires(EQ)	1	×	3	4	6	=	3
Internal logical File(ILF)	3	×	7	10	15	=	21
External interface File(EIF)	0	×	5	7	10	=	0
					Count total	**=**	**54**

-One of the external outputs has weighting factor average because it produces the output by referencing two tables and the number of referenced fields is six.
-External file are three:

1. Book file: has 2 RETs and 7 DETs, so it is simple.
2. Member file: has 0 RETs and 2 DETs, so it is simple.
3. Book copy log file: has 0 RETs and 9 DETs, so it is simple.

5.1.1.3 Calculate FP for Use Cases:

FP for use case: Add Book **Release #:** 1

Information Domain Value	Count		Weighting Factor				
			Simple	Average	complex		
Number of user inputs (EI)	6	×	3	4	6	=	18
Number of user outputs (EO)	7	×	4	5	7	=	28
Number of user inquires(EQ)	0	×	3	4	6	=	0
Internal logical File(ILF)	1	×	7	10	15	=	7
External interface File(EIF)	0	×	5	7	10	=	0
					Count total	**=**	**53**

Table 4: FP for use case Add Book

FB = count total × Adjustment complexity factor
= 53 × 1.1
= 58

FP for use case: Borrow Book **Release #:** 1

Information Domain Value	Count		Weighting Factor				
			Simple	Average	complex		
Number of user inputs (EI)	3	×	3	4	6	=	9
Number of user outputs (EO)	4	×	4	5	7	=	16
	1	×	4	5	7	=	5
Number of user inquires(EQ)	1	×	3	4	6	=	3
Internal logical File(ILF)	3	×	7	10	15	=	21
External interface File(EIF)	0	×	5	7	10	=	0
					Count total	**=**	**54**

Table 5: FP for use case Borrow book

FB = count total × Adjustment complexity factor
= 54 × 1.1
= 59

FP for use case: Get Member's Fines **Release #:** 1

Information Domain Value	Count		Weighting Factor				
			Simple	Average	complex		
Number of user inputs (EI)	2	×	3	4	6	=	6
Number of user outputs(EO)	3	×	4	5	7	=	12
Number of user inquires(EQ)	0	×	3	4	6	=	0
Internal logical File(ILF)	1	×	7	10	15	=	7
External interface File(EIF)	0	×	5	7	10	=	0
					Count total	**=**	**25**

Table 6: FP for use case Get Member's Fines

FB = FB = count total × Adjustment complexity factor
= 25 × 1.1
= 28

FP for use case: Return Book **Release #:** 2

Information Domain Value	Count		Weighting Factor				
			Simple	Average	complex		
Number of user inputs (EI)	4	×	3	4	6	=	12
Number of user outputs(EO)	6	×	4	5	7	=	24
Number of user inquires(EQ)	2	×	3	4	6	=	6
Internal logical File(ILF)	2	×	7	10	15	=	14
External interface File(EIF)	0	×	5	7	10	=	0
					Count total	**=**	**56**

Table 7: FP for use case Return Book

FB = count total × Adjustment complexity factor
= 56 × 1.1
= 62

FP for use case: Delete Book **Release #:** 2

Information Domain Value	Count		Weighting Factor				
			Simple	Average	complex		
Number of user inputs (EI)	4	×	3	4	6	=	12
Number of user outputs (EO)	4	×	4	5	7	=	16
Number of user inquires(EQ)	1	×	3	4	6	=	3
Internal logical File(ILF)	1	×	7	10	15	=	7
External interface File(EIF)	0	×	5	7	10	=	0
					Count total	**=**	**38**

Table 8: FP for use case Delete Book

FB = count total × Adjustment complexity factor
= 38 × 1.1
= 42

FP for use case: Update Book **Release #:** 2

Information Domain Value	Count		Weighting Factor				
			Simple	Average	complex		
Number of user inputs(EI)	4	×	3	4	6	=	12
Number of user outputs (EO)	5	×	4	5	7	=	20
Number of user inquires(EQ)	1	×	3	4	6	=	3
Internal logical File(ILF)	1	×	7	10	15	=	7
External interface File(EIF)	0	×	5	7	10	=	0
					Count total	**=**	**42**

Table 9: FP for use case Update Book

FB = FB = count total × Adjustment complexity factor
= 42 × 1.1
= 46

5.1.2Object Point:

Release #	Function	Number of Screens	Screen Complexity (Using Boehm criteria)	Weight	Screen	Number of Report	Number of 3GL Module	Weight	3GL Module	**NOP**
1	Main Page	1	Simple	1	1	0	1	10	10	11
1	Add Book	4	Simple	1	4	0	39	10	390	394
1	Borrow Book	2	Simple	1	2	0	15	10	150	152
1	Get Fee	2	Simple	1	2	0	8	10	80	82
Total										NOP = 639
2	Main Page	1	Simple	1	1	0	7	10	70	71
2	Return Book	2	Simple	1	2	0	9	10	90	92
2	Delete Book	2	Simple	1	2	0	15	10	150	152
2	Update Book	3	Simple	1	3	0	33	10	330	333
Total										NOP = 648

Table 10: Object Point

Boehm Criteria:

Screen:

	Number and source of data tables		
Number of views contained	**Total <4**	**Total <8**	**Total 8+**
<3	simple	simple	medium
3-7	simple	medium	difficult
8+	medium	difficult	difficult

Table 11: Screen Object Point

Report:

	Number and source of data tables		
Number of views contained	**Total <4**	**Total <8**	**Total 8+**
<3	simple	simple	medium
3-7	simple	medium	difficult
8+	medium	difficult	difficult

Table 12: Report Object Points

Weight:

Object Type	Simple	Medium	Difficult
Screen	1	2	3
Report	2	5	8
3 GL Module	10	10	10

Table 13: Object Points Weight

5.1.3Use Case Point:

Calculate UCP:

1. Determine the UAW (Unadjusted Actor weight):
2. Determine number of UUCW (Unadjusted Use case Weight):
3. Determine Total UUCP (Unadjusted Use Case Point): Total UUCP = Total UAW + Total UUCW.
4. Computing technical and environmental factor:
    a. Equation for Tfactor = sum(T1....T13)
    b. TCF (Technical Complexity Factor): TCF = 0.6 + (0.01 * Tfactor).
    c. EF (Environmental Factor):
    d. Efactor = SUM(e1...e8).
    e. Calculating Environmental Factor = EF = 1.4 + (-0.03 * Efactor).
5. Calculating the Adjusted Use case points: AUCP = UUCP * TCF * EF
6. Multiplying by Man/Hours Factor: AUCP * Person/Hours/AUCP.[5][8]

Release #	Use case	UAW	UUCW	UUCP	TCF	EF	AUCP	UCP
1	Borrow book	3	10	13	0.78	0.74	7.5	90
1	Add book	3	10	13	0.78	0.74	7.5	90
1	Get Fee	3	5	8	0.78	0.74	4.6	55.2
Total UCP = 235.2								
2	Return book	3	10	13	0.78	0.69	7	84
2	Delete book	3	10	13	0.78	0.69	7	84
2	Update Book	3	10	13	0.78	0.69	7	84
Total UCP = 252								

Table 14: Total UCP

Computing technical factor			**Release #:** 1	
	Technical factor	**Weight**	**Value**	**Weighted Value**
t1	Distributed System	2	2	4
t2	Response time	1	4	4
t3	End user efficiency	1	3	3
t4	Complex Internal Processing	1	2	2
t5	Reusable Code	1	1	1
t6	Installation Ease	0.5	0	0
t7	Easy use	0.5	4	2
t8	Portable	2	0.5	1
t9	Easy to change	1	0	0
t10	Concurrent	1	0	0
t11	Security	1	0	0

	objectives			
t12	Direct access to third parties	1	0	0
t13	User training facilities	1	1	1
	Total			18

Table 15: Computing technical factor (release #1)

Computing environmental factor			**Release #:** 1	
	Environmental Factor	**Value**	**Weight**	**Weighted Columns**
e1	Familiarity with project	4	1.5	6
e2	Application experience	4	0.5	2
e3	Object-oriented programming experience	5	1	5
e4	Lead analyst capability	5	0.5	2.5
e5	Motivation	0.5	1	0.5
e6	Stable requirements	5	2	10
e7	Part-time Staff	0	-1	0
e8	Difficult programming language.	4	-1	-4
	total			22

Table 16: Computing environmental factor (release #1)

Computing technical factor				**Release #:** 2
	Technical factor	**Weight**	**Value**	**Weighted Value**
t1	Distributed System	2	2	4
t2	Response time	1	4	4
t3	End user efficiency	1	3	3
t4	Complex Internal Processing	1	2	2
t5	Reusable Code	1	1	1
t6	Installation Ease	0.5	0	0
t7	Easy use	0.5	4	2
t8	Portable	2	0.5	1
t9	Easy to change	1	0	0

t10	Concurrent	1	0	0
t11	Security objectives	1	0	0
t12	Direct access to third parties	1	0	0
t13	User training facilities	1	1	1
	Total			18

Table 17: Computing technical factor (release #2)

Computing environmental factor			**Release #:** 2	
	Environmental Factor	**Value**	**Weight**	**Weighted Columns**
e1	Familiarity with project	5	1.5	7.5
e2	Application experience	4	0.5	2
e3	Object-oriented programming experience	5	1	5
e4	Lead analyst capability	5	0.5	2.5
e5	Motivation	0.5	1	0.5
e6	Stable requirements	5	2	10
e7	Part-time Staff	0	-1	0
e8	Difficult programming language.	4	-1	-4
	total			23.5

Table 18: Computing environmental factor (release #2)

5.2Shodan Input Metric Survey:

10 -	Fanatic (100%)	5 -	Half & Half (50%)
9 -	Always (90%)	4 -	Common (40%)
8 -	Regular (80%)	3 -	Sometimes (30%)
7 -	Often (70%)	2 -	Rarely (20%)
6 -	Usually (60%)	1 -	Hardly ever (10%)
		0 -	Never

Pair #1	Eidah & Sumayah
Pair #2	Amani & Asma
Pair #3	Manal & Nuha

Release #:	**1**			**2**		
XP Practice	**Pair #1**	**Pair #2**	**Pair #3**	**Pair #1**	**Pair #2**	**Pair #3**
Foundations						
Automated Unit Tests (40)	0: Never (0%)	0: Never (0%)	0: Never (0%)	0: Never (0%)	0: Never (0%)	0: Never (0%)
Customer Acceptance Tests (20)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)
Test-First Design (20)	3: Sometimes (30%)	3: Sometimes (30%)	3: Sometimes (30%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)
Pair Programming (80)	9: Always (80%)	9: Always (90%)	9: Always (90%)	8: Regular (80%)	8: Regular (80%)	8: Regular (80%)
Refactoring (70)	9: Always (90%)	7: Often (70%)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)
Customer Planning						
Release Planning/Planning Game (38)	9: Always (90%)	9: Always (90%)	9: Always (90%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)
Customer Access (32)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)
Short Releases (40)	10: Fanatic (100%)	9: Always (90%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)
Stand Up Meeting (5)	10: Fanatic (100%)	10: Fanatic (100%)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)
Teaming						
Continuous Integration (60)	9: Always (90%)	8: Regular (80%)	8: Regular (80%)	8: Regular (80%)	8: Regular (80%)	8: Regular (80%)
Coding Standards (30)	1: Hardly ever (10%)	1: Hardly ever (10%)	1: Hardly ever (10%)	3: Sometimes (30%)	3: Sometimes (30%)	3: Sometimes (30%)

Collective Code Ownership (50)	7: Often (70%)	10: Fanatic (100%)	8: Regular (80%)	6: Usually (60%)	7: Often (70%)	6: Usually (60%)
Craftsmanship						
Sustainable Pace (30)	5: Half n Half (50%)	5: Half n Half (50%)	6: Usually (60%)	5: Half n Half(50%)	5: Half n Half(50%)	5: Half n Half(50%)
Simple Design (55)	10: Fanatic (100%)	9: Always (80%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)
Metaphor (35)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)
Introspection						
Lessons Learned (6)	9: Always (90%)	9: Always (90%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)	10: Fanatic (100%)
Growth (6)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)	9: Always (90%)
Morale (6)	7: Often (70%)	9: Always (90%)	9: Always (90%)	8: Regular (80%)	8: Regular (80%)	8: Regular (80%)
Artifact Reduction (6)	5: Half n Half (50%)	5: Half n Half (50%)	6 - Usually (60%)	6 - Usually (60%)	6 - Usually (60%)	6 - Usually (60%)
weighted average (Shodan Score)	76%			77.8%		

5.3 Improvement & estimate accuracy:

Table A3
Improvements of estimate accuracy with time (Release and Project level)

Release #	# of updates of estimated effort	Initial estimated effort (Man-hr)	Final estimated effort (Man-hr)	Actual effort (Man-hr)	(Final – Initial) / Actual
1	0	131.5	131.5	151	0
2	1	168	129	108	-0.36
Project Level (Total) Time slots of **0.25 hour**		299.5	260.5	259	-0.15

5.4 Estimation & Actual Effort:

Table A2

UC / Story Effort
Estimation & Actual

System: Library System **Date:** 28-11-2006
UC / Story Card # & Name: 6-Add Book **Release #:** 1
Developers Team (Pair): Asma, Amani

Task			Effort Man-hrs Slots of **0.25 hour**					Remark
#	Name	Type	Estimated				Actual	
			1st	2nd	3rd	4th		
1	Understand requirements	P	4				5	
2	Use case writing	A	6				5	
3	Class diagram design	D	4				3	
4	Sequence diagrams design	D	6				5	
5	Learning New Technology (.Net)	L	7				8	
6	Interface design	D	2				1	
7	Implementation	I	11				7	
8	Functional test cases writing	OTH	4				5	
9	Writing unit testing	UTW	3				1.25	
10	Test cases running	UTR	3				4	
11	Documents writing	DOC	2				2	
12	Drawing diagrams	DOC	2				1	
Story estimated effort			54				47.25	

Task type: P (Planning), A (Analysis), D (design), I (Impl), UTW (Unit Test writing), UTR (Unit Test run), DOC (Doc writing, Drawing Diagrams), L (learning), OTH (Others: specify).
Time slots of **0.25 hour**

Table A2

UC / Story Effort Estimation & Actual

System: Library System **Date:** 27-11-2006
UC / Story Card # & Name: 11- Borrow Book **Release #:** 1
Developers Team (Pair): Manal, Nuha

Task			Effort Man-hrs Slots of **0.25 hour**					Remark
#	Name	Type	Estimated				Actual	
			1st	2nd	3rd	4th		
1	Understand requirements	P	3				4	
2	Use case writing	A	6				5	
3	Class diagram design	D	7				2.5	
4	Sequence diagrams design	D	14				2.5	
5	State diagram design	D	3				0.5	
6	Learning New Technology (.Net)	L	4				6.5	
7	Interface design	D	2				1	
8	Implementation	I	4				3	
9	Functional test cases writing	OTH	1.5				2.5	
10	Writing unit testing	UTW	1.5				1.25	
11	Test cases running	UTR	5				6	
12	Documents writing	DOC	2				3	
13	Drawing diagrams	DOC	2				1.25	
Story estimated effort			55				39	

Task type: P (Planning), A (Analysis), D (design), I (Impl), UTW (Unit Test writing), UTR (Unit Test run), DOC (Doc writing, Drawing Diagrams), L (learning), OTH (Others: specify).
Time slots of **0.25 hour**

Table A2

UC / Story Effort Estimation & Actual

System: Library System **Date:** 27-11-2006
UC / Story Card # & Name: 14- Get member's fines **Release #:** 1
Developers Team (Pair): Eidah, sumayah

Task			Effort Man-hrs Slots of **0.25 hour**					Remark
#	Name	Type	Estimated				Actual	
			1st	2nd	3rd	4th		
1	Understanding requirements	P	1				2	
2	Use case text writing	DOC	4				2.5	
3	Sequence diagrams design	D	8				4	
4	Class diagram design	D	5				1	
5	Learning New Technology (.Net)	L	3				5	
6	Interface design	D	0.25				0.75	
7	Implementation	I	3				2	
8	Functional test cases writing	UTW	1				1	
9	Test cases running	UTR	1				0.5	
10	Drawing diagrams	DOC	1				1	
Story estimated effort			27.25				19.75	

Task type: P (Planning), A (Analysis), D (design), I (Impl), UTW (Unit Test writing), UTR (Unit Test run), DOC (Doc writing, Drawing Diagrams), L (learning), OTH (Others: specify).
Time slots of **0.25 hour**

Table A2

UC / Story Effort Estimation & Actual

System: Library System **Date:** 26-12-2006
UC / Story Card # & Name: 12- Return Book **Release #:** 2
Developers Team (Pair): Eidah, sumayah

Task			**Effort Man-hrs** Slots of **0.25 hour**					**Remark**
#	**Name**	**Type**	**Estimated**				**Actual**	
			1st	2nd	3rd	4th		
1	Understanding requirements	P	3	3			3	
2	Use case text writing	A	6	2			2	
3	Sequence diagrams design	D	14	4			4	
4	Class diagram design	D	7	2			2	
5	Learning New Technology (.Net)	L	3	1			0	
6	Interface design	D	2	1			0.25	
7	Implementation	I	4	3			4	
8	Functional test cases writing	OTH	3	3			3	
9	Writing unit testing	UTW	5	1			1	
10	Test cases running	UTR	2	1			1	
11	Drawing diagrams	DOC	2	1			1	
Story estimated effort			51	22			21.25	

Task type: P (Planning), A (Analysis), D (design), I (Impl), UTW (Unit Test writing), UTR (Unit Test run), DOC (Doc writing, Drawing Diagrams), L (learning), OTH (Others: specify).
Time slots of **0.25 hour**

Table A2

UC / Story Effort Estimation & Actual

System: Library System **Date:** 26-12-2006
UC / Story Card # & Name:7- Delete Book **Release #:** 2
Developers Team (Pair): Manal, Nuha

Task			Effort Man-hrs Slots of **0.25 hour**					Remark
#	Name	Type	Estimated				Actual	
			1st	2nd	3rd	4th		
1	Understand requirements	P	3	3			3	
2	Use case writing	A	6	6			6	
3	Class diagram design	D	7	3			1.25	
4	Sequence diagrams design	D	14	4			1.5	
5	State diagram design	D	3	0.5			1	
6	Learning New Technology (.Net)	L	4	7			.75	
7	Interface design	D	2	1			.25	
8	Implementation	I	4	3			3.5	
9	Functional test cases writing	OTH	1.5	3			.75	
10	Writing unit testing	UTW	1.5	1.25			.25	
11	Test cases running	UTR	5	6			1	
12	Documents writing	DOC	2	3			4.25	
13	Drawing diagrams	DOC	2	1.25			2	
Story estimated effort			55	42			25.25	

Task type: P (Planning), A (Analysis), D (design), I (Impl), UTW (Unit Test writing), UTR (Unit Test run), DOC (Doc writing, Drawing Diagrams), L (learning), OTH (Others: specify).
Time slots of **0.25 hour**

Table A2

UC / Story Effort Estimation & Actual

System: Library System **Date** 28-12-2006
UC / Story Card # & Name: 8-Update Book **Release #:** 2
Developers Team (Pair): Amani, Asma

Task			**Effort Man-hrs** Slots of **0.25 hour**					**Remark**
#	**Name**	**Type**	**Estimated**				**Actual**	
			1st	2nd	3rd	4th		
1	Understanding requirements	P	3	1			1	
2	Use case text writing	A	4	3			2	
3	Sequence diagrams design	D	8	6			3	
4	Class diagram design	D	5	3			1	
5	Learning New Technology (.Net)	L	3	4			5	
6	Interface design	D	2	1			1	
7	Implementation	I	3	5			8	
8	Functional test cases writing	UTW	2	2			1	
9	Test cases running	UTR	5	1			0.5	
10	Drawing diagrams	DOC	2	2			1	
Story estimated effort			37	28			23.5	

Task type: P (Planning), A (Analysis), D (design), I (Impl), UTW (Unit Test writing), UTR (Unit Test run), DOC (Doc writing, Drawing Diagrams), L (learning), OTH (Others: specify).
Time slots of **0.25 hour**

5.5Analysis of Actual Effort :

Table A4 Analysis of Actual Effort Man-hr For UC, Rel, Proj Time slots of 0.25 hour											
Rel #	UC Name	Task									
		Planning	Analysis	Design	Impl	Unit Test writing	Unit Test run	DOC Text + Diagrams	learning	Others	UC Total Actual Effort
1	Add Book	5	5	9	7	1.25	4	3	8	5	47.25
1	Borrow Book	4	5	6.5	3	1.25	6	4.25	6.5	2.5	39
1	Get Member's fines	2	0	5.75	2	1	0.5	3.5	5	0	19.75
Rel 1 Total		11	10	21.25	12	3.5	10.5	10.75	19.5	7.5	106
Rel i% = Task(i)/Total		0.10	0.09	0.2	0.11	0.033	0.01	0.1	0.18	0.07	1
2	Return Book	3	2	6.25	4	1	1	1	0	3	21.25
2	Delete Book	3	6	3.75	3.5	0.25	1	6.25	0.75	0.75	25.25
2	Update Book	1	2	5	8	1	0.5	1	5	0	23.5
Rel 2 Total		7	10	15	15.5	2.25	2.5	8.25	5.75	3.75	70
Rel 2% = Task(i)/Total		0.10	0.143	0.214	0.22	0.032	0.036	0.118	0.082	0.054	1
Proj Total		18	20	36.25	27,5	5,75	13	19	25,25	11,25	176
Proj % = Task(i)/Total		0.102	0.114	0,206	0,156	0.033	0.074	0.108	0.143	0.064	1

5.6 Productivity Metrics Table:

Able A5 Productivity Metrics										
			Actual Size Measurements				Productivity Metrics			
Rel #	UC Name	Actual Man-Hr	LOC	FP	NOP	UCP	LOC/ Man-Hr	FP / Man-Hr	NOP / Man-Hr	UCP/ Man-Hr
1	Add book	12	400	58	394	90	33.33	4.83	32.83	7.50
1	Borrow book	12	325	59	152	90	27.08	4.91	12.66	7.50
1	Get member fines	12	111	28	82	55.2	9.25	2.33	6.83	4.60
Rel 1 Total		36	836	145	628	235.2	69.66	12.07	52.32	19.60
2	Return Book	12	240	62	92	84	20	5.17	7.67	7
2	Delete Book	12	343	42	152	84	28.58	3.50	12.67	7
2	Update Book	12	479	46	333	84	39.92	3.83	27.75	7
Rel 2 Total		36	1062	150	577	252	88.5	12.50	48.09	21
Proj Total		72	1898	295	1205	7.2	158.16	24.75	100.41	40.60

5.7 Quality Measurements and Metrics:

Able A6 Quality Measurements and Metrics								
Rel #	**UC Name**	**Actual Size LOC**	**# of test cases (Functional 'Acceptance' + Unit)**	**Effort (M-HR) Designing test cases (Functional + Unit)**	**# of Automated tests**	**Test density: Number of test cases (Functional + Unit) / Story LOC**	**Number of defects (Functional + Unit)**	**D de Nu d (Fu + Sto**
1	Add book	400	8	6.25	0	0.020	0	
1	Borrow book	325	10	3.75	0	0.031	0	
1	Get Member's fines	111	2	1	0	0.018	0	
Rel 1 Total		836	20	11	0	0.069	0	
2	Return Book	240	6	4	0	0.025	0	
2	Delete Book	343	10	1	0	0.029	0	
2	Update Book	497	2	1	0	0.004	0	
Rel 2 Total		1080	18	6	0	0.058	0	
Proj Total		1916	38	17	0	0.127	0	

5.8Other Estimation & Actual:

Table A7 Other Estimation & Actual									
System: Library System				**Date:** 28/12/2006				**Release #:** 1	
Task				**Effort Man-hrs** Slots of **0.25 hour**					**Remark**
#	**Task Name**	**Developer Name**	**Type**	**Estimated**				**Actual**	
				1st	2nd	3rd	4th		
1	Setting up the system architecture	Sumayah	OTH	0.5				1	3 tiers. Presentation Layer Business layer Data Layer
2	ER	Sumayah	D	1				2	
3	DAL Implementation(Code + DB)	Sumayah	I	None				5	This is incremental work as each story is developed; the corresponding data services are implemented.
4	Technical documentation	Asma, amani	DOC	5				10	
5	Metrics Measurements	Asma,Amani,Manal,Nuha	OTH	7				10	
6	Shodan metric	Eidah, sumayah	OTH	2				4	
7	SRS document	Eidah,Sumayah	DOC	5				6	
8	System Integration	Sumayah,Asma,Manal	OTH	4				7	
Story estimated effort				24.5				45	
Task type: P (Planning), A (Analysis), D (design), I (Impl), UTW (Unit Test writing), UTR (Unit Test run), DOC (Doc writing, Drawing Diagrams), L (learning), OTH (Others: specify). Time slots of **0.25 hour**									

Table A7
Other
Estimation & Actual

System: Library System **Date:** 28/12/2006 **Release #:** 2

#	Task Name	Developer Name	Type	Effort Man-hrs Slots of **0.25 hour**					Remark
				Estimated				Actual	
				1st	2nd	3rd	4th		
1	DAL Implementation(Code + DB)	Sumayah	I	1	2			4	
2	Technical documentation	Manal,Eidah	DOC	10	15			11	
3	Metrics Measurements	Asma,Amani,Manal,Nuha	OTH	7	9			10	
4	Shodan metric	Eidah, sumayah	OTH	1	1			1	
5	SRS document	Nuha	DOC	2	4			3	
6	Integration Testing	Eidah, sumayah	OTH	4	6			9	
Story estimated effort				25	37			38	

Task type: P (Planning), A (Analysis), D (design), I (Impl), UTW (Unit Test writing), UTR (Unit Test run), DOC (Doc writing, Drawing Diagrams), L (learning), OTH (Others: specify).
Time slots of **0.25 hour**

Chapter 6

Implementation & Tools

6.1 Framework:

.NET Technology :

.NET is the Microsoft Web services strategy to connect information, people, systems, and devices through software. Integrated across the Microsoft platform, .NET technology provides the ability to quickly build, deploy, manage, and use connected, security-enhanced solutions with Web services. .NET-connected solutions enable businesses to integrate their systems more rapidly and in a more agile manner and help them realize the promise of information anytime, anywhere, on any device [4].

6.2 Programming Language:

ASP.NET with C#:

ASP.NET is Microsoft's new programming technology which is now gaining acceptance and momentum. Soon, it will completely replace standard ASP. Glaserweb.com is already up to speed with .NET technology, and is fully ready for this switch. While .NET development is not as rapid (leading to higher production costs), it is significantly more stable, and runs much faster than older programming technologies, opening up new possibilities for web development.[7]

6.3 Tools Used:

- **Visual Studio 2005:** Visual Studio is a suite of applications created by Microsoft to give developers a compelling development environment for the Windows and .NET platforms. Visual Studio can be used to write console applications, Windows applications, Windows services, Windows Mobile applications, ASP.NET applications, and ASP.NET web services, in your choice of C++, C#, VB.NET, J#, and more. Visual Studio also includes various additional development tools, such as Visual SourceSafe; which tools are included depends greatly on the edition of Visual Studio that you are using.[3]
- Microsoft Word
- Microsoft Visio.
- MySQL Express.
- Adobe Photoshop 8.

References

Books:

[1] Craig Laman. Applying UML and Patterns: An Introduction to Object-Oriented and Design

[2] Jeffery A. Hoffer, Joey F. George, Joseph S. Valacich. 2002. Modern System Analysis & Design. 3rd edition.

Web Sites:

[3] Avery, James. "What is Visual Studio." Windows Devcenter. 22 Aug. 2005. 1 Dec. 2006 <http://www.windowsdevcenter.com>. /pub/a/windows/2005/08/22/whatisVisualStudio.html>.

[4] "Basics of .NET." Microsoft. 1 Feb. 2005. 1 Dec. 2006 <http://www.microsoft.com/net/basics.mspx>.

[5] Clemmons, Roy K. "Project Estimation with Use Case Points." STSC CrossTalk. 1 Dec. 2006 <http://www.stsc.hill.af.mil/ CrossTalk/2006/02/0602Clemmons.html>

[6] David Garmus, An Introduction to Function Point Counting. < http://www.davidconsultinggroup.com/articles/pbfpart.htm>.

[7] "Glossary of Terms." Glaserweb. 1 Dec. 2006 <http://www.glaserweb.com/glossary.php>.

[8] Koirala, Shivprasad. "How to Prepare Quotation Using Use Case Points." The Code Project. 14 Dec. 2004. 1 Dec. 2006 <http://www.codeproject.com/gen/design/usecasepoints.asp>.

[9] Garmus, David. "Function Point Counting." David Consulting Group. 1 Jan. 2007 <http://www.davidconsultinggroup.com/articles/pbfpart.htm>.

[10] "Library System." Wikipedia. 27 Dec. 2006 <http://en.wikipedia.org/wiki/Main_Page>.

[11] "ieee-829." answers.com. 1 Jan. 2007 <http://www.answers.com/topic/ieee-829>.