
System Requirements Specification Index

For

E-Library Console Application

Version 1.0

TABLE OF CONTENTS

1	Project Abstract	3
2	Constraints	3
2.1	Book Constraints	3
2.2	Book Issue Constraints	3
2.3	Common Constraints	3
3	Template Code Structure	4
3.1	Package: com.iiht.training.elibrary.model	4
3.2	Package: com.iiht.training.elibrary.inventory	4
3.3	Package: com.iiht.training.elibrary.exception	5
3.4	Package: com.iiht.training.elibrary.controller	5
4	Execution Steps to Follow	6

E-Library Console Application

System Requirements Specification

1 PROJECT ABSTRACT

E-Library Console Application is a pure java application with Java collection, where it allows to manage the books and issue the books from the library.

2 CONSTRAINTS

2.1 BOOK CONSTRAINTS

- When adding a book, if the ISBN already exists in the inventory, the method should throw an **ISBNAlreadyExistsException** with the message:

"The ISBN <isbn> already exists"

2.2 BOOK ISSUE CONSTRAINTS

- When issuing a book, if the book is already issued to another user, the method should throw a **BookAlreadyIssuedException** with the message:

"The Book is already Issued"

- When issuing a book, if the ISBN does not exist in the inventory, the method should throw an **ISBNDoesNotExistsException** with the message:

"ISBN <isbn> does not exists"

2.3 COMMON CONSTRAINTS

- Take console input of number of books: (n)
- Take input of details of each book and store it in a collection.
- Take input of details of books to be issued (only 1 book at a time)
- Take input of details of book to issue and store in a collection
- Show the books stock remained after issuing books

3 TEMPLATE CODE STRUCTURE

3.1 PACKAGE: COM.IIHT.TRAINING.ELIBRARY.MODEL

Resources

Class/Interface	Description	Status
Book (class)	<ul style="list-style-type: none">This class contains all the properties of the Book class.	Already implemented.
BookIssue(class)	<ul style="list-style-type: none">This class contains all the properties of the BookIssue class	Already implemented.

3.2 PACKAGE: COM.IIHT.TRAINING.ELIBRARY.INVENTORY

Resources

Class/Interface	Description	Status
BookInventory (class)	<ul style="list-style-type: none">This class contains all the methods which are used to write the business logic for the applicationYou can create any number of private methods in the class	Partially implemented.

3.3 PACKAGE: COM.IIHT.TRAINING.ELIBRARY.EXCEPTION

Resources

Class/Interface	Description	Status
ISBNAlreadyExistsException (Class)	<ul style="list-style-type: none">Custom Exception to be thrown when trying to add a book for which ISBN is already exists	Already created.
ISBNDoesNotExistsException (Class)	<ul style="list-style-type: none">Custom Exception to be thrown when trying to issue a book for which the ISBN does not exists	Already created.
BookAlreadyIssuedException (Class)	<ul style="list-style-type: none">Custom Exception to be thrown when trying to issue a book which is already issued.	Already created.

3.4 PACKAGE: COM.IIHT.TRAINING.ELIBRARY.CONTROLLER

Resources

Class/Interface	Description	Status
BookInventoryController (Class)	<ul style="list-style-type: none">This is the class which has the main method. All the business logic methods of the BookInventory class will be called from this class.	To be implemented

4 EXECUTION STEPS TO FOLLOW

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers need to go to the Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal.
3. This editor Auto Saves the code.
4. To run your project use command:
`sudo JAVA_HOME=$JAVA_HOME /usr/share/maven/bin/mvn clean install exec:java -Dexec.mainClass="com.iiht.training.elibrary.controller.BookInventoryController"`

**If it asks for the password, provide password : pass@word1*

5. To test your project, use the command
`sudo JAVA_HOME=$JAVA_HOME /usr/share/maven/bin/mvn test`

**If it asks for the password, provide password : pass@word1*