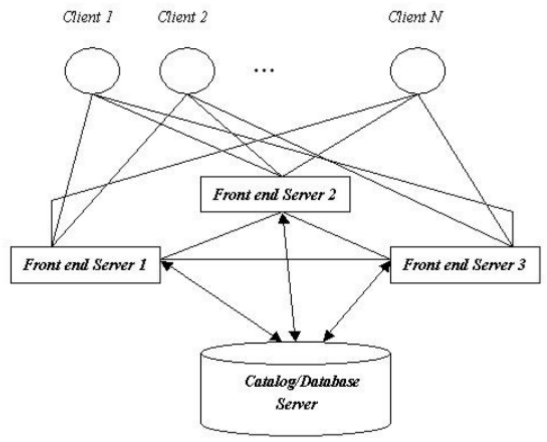
Saeef Ahmad
ID: gij817

Abstract

Here in this project we have developed Mybooks project. This is a small book store. Right now it has the following four books for sale. We can add and remove any item from this.
• How to be good at CS5523.
• RPCs and RMI in distributed systems.
• Why go to the graduate school.
• How to survive the graduate school.
We have implemented a client-server program using RMIs in distributed system in java Programming. There are three front end servers, one backend server and multiple clients.

Descriptions:
(1) Two-Tier design:
We emploied a two-tier design: a front-end and a back-end. The front-end tier will accept user requests and perform initial processing. The number of servers of front-end servers is 3 now, but can be configured to a larger number.



(2) Clock Synchronization:
Since there are 3 front-end servers, any customer can contact any of three front-end replicas with their requests. In reality, we may chose a timer daemon by a leader election, which will periodically send out synchronization information among all front-end servers. We implemented the Berkeley clock synchronization algorithm to synchronize their clocks. Each node then maintains a clock-offset variable that is the amount of time by which the clock must be adjusted, as per the Berkeley algorithm. We will not actually adjust the system clock by this value. Rather to timestamp any request, we simply read the current system time, add this offset to the time, and use this adjusted time to timestamp request. Every incoming order request at each front-end replica is time-stamped with the synchronized clock value; this time stamp is relayed to the database server for buy requests. The time stamps are used to determine an ordering of buy requests and every 100th buy request is automatically given a 10% discount for the requested book.

Functionality
For each book, it maintains the number of available items in stock, the cost and title of each book. The server also maintains how many order of books it have received, and how many items are still available. The server supports the following services:
• Search: allows the user to specify a topic/title and returns all books belonging to this topic/title.
• Lookup(item number) : allows an item number to be specified and returns details of all items, such as number of available items in stock, cost and title.
• Order(item number) : allows the user to buy/order a book with the specified item number.
The server also supports some report functionalities so that we can use it to improve our service or

performance in the future. Also, it can report the performance to serve each request, thus this can provide some information of how to improve the performance in the future.
• reportRequestsNumber (service): allows the user to query that how many requests on each service have been received.
• reportGoodOrders(): tell users how many books have been sold successfully starting from the beginning.
• reportFailedOrders(): report how many orders are failed in total.
• reportServicePerformance(service): report the average performance for serving a request for the specific service.
The server has a counter for each service -- search, lookup and buy (total 3 counters), which records the number of requests for each service. The server also includes 3 timers to track the accumulated processing time for each service with synchronizations.

Status of the Project

The project is fully completed. It has the full functionality. We have worked and tested on windows operating system.

<u>The Instruction to run this program as given below:</u>

The Instruction to run this program as given below:
For backend server:- (for windows)
1) Open comand prompt
2) Go to the src directory of the project
2) Run the following command:-

```
javac com\mybook\server\*.java
rmic com.mybook.server.MyBook
start rmiregistry
java com.mybook.server.MyBookServer
```

For backend server:- (for linux)
1) Open terminal
2) Go to the src directory of the project
2) Run the following command:-

```
javac com/mybook/server/*.java
rmic com.mybook.server.MyBook
rmiregistry &
java com.mybook.server.MyBookServer
```

For frontend server 2:- (for windows)
1) Open comand prompt
2) Go to the src directory of the project
2) Run the following command:-

```
javac com\mybook\server\*.java
rmic com.mybook.server.MyBookFront
start rmiregistry
java com.mybook.server.MyBookServersFront 2 0
```

For frontend server 2:- (for linux)
1) Open terminal
2) Go to the src directory of the project
2) Run the following command:-

```
javac com/mybook/server/*.java
rmic com.mybook.server.MyBookFront
rmiregistry &
```

Saeef Ahmad
ID: gij817

```
            java com.mybook.server.MyBookServersFront 2 0
```

For frontend server 3:- (for windows)
1) Open comand prompt
2) Go to the src directory of the project
2) Run the following command:-

```
            javac com\mybook\server\*.java
            rmic com.mybook.server.MyBookFront
            start rmiregistry
            java com.mybook.server.MyBookServersFront 3 0
```

For frontend server 3:- (for linux)
1) Open terminal
2) Go to the src directory of the project
2) Run the following command:-

```
            javac com/mybook/server/*.java
            rmic com.mybook.server.MyBookFront
            rmiregistry &
            java com.mybook.server.MyBookServersFront 3 0
```

For frontend server 1 (LEADER SERVER):- (for windows) // Start this server at the last
1) Open comand prompt
2) Go to the src directory of the project
2) Run the following command:-

```
            javac com\mybook\server\*.java
            rmic com.mybook.server.MyBookFront
            start rmiregistry
            java com.mybook.server.MyBookServersFront 1 1
```

For frontend server 1 (LEADER SERVER):- (for linux)
1) Open terminal
2) Go to the src directory of the project
2) Run the following command:-

```
            javac com/mybook/server/*.java
            rmic com.mybook.server.MyBookFront
            rmiregistry &
            java com.mybook.server.MyBookServersFront 1 1
```

For Client:-(for windows)
1) Open command prompt
2) Go to the src directory of the project
3) Run the following command:

```
            javac com\mybook\client\*.java
            java com.mybook.client.MyBookClient
```

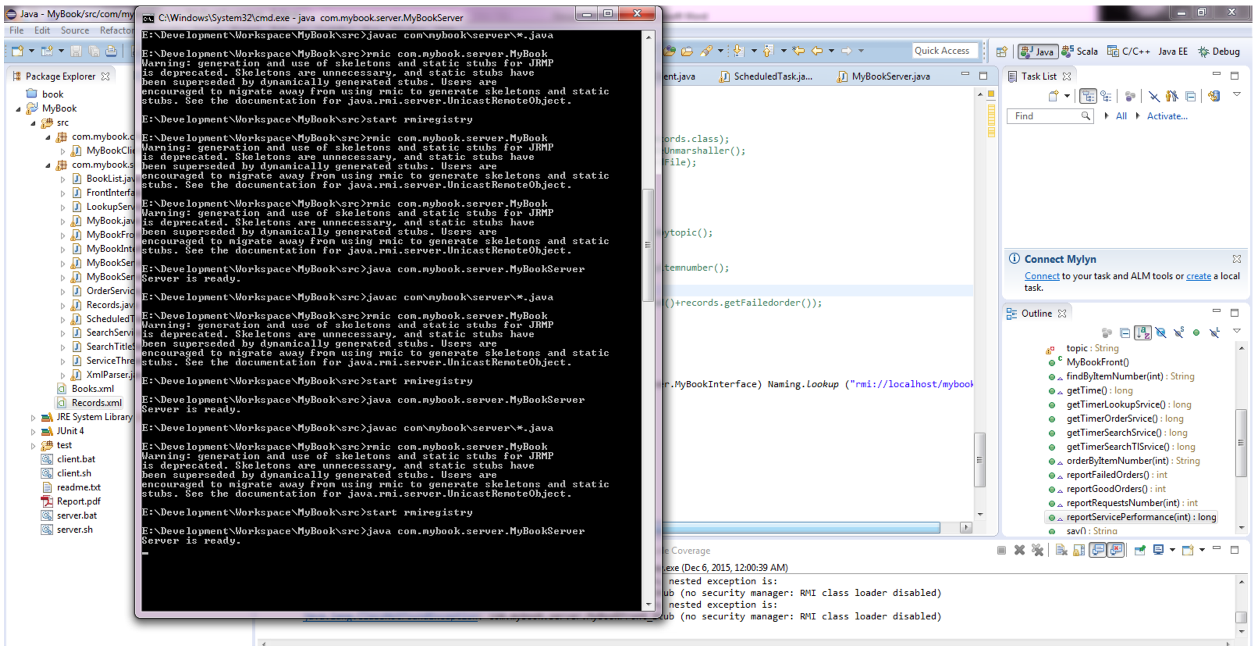
For Client:-(for linux)
1) Open terminal
2) Go to the src directory of the project
3) Run the following command:

```
            javac com/mybook/client/*.java
            java com.mybook.client.MyBookClient
```

Saeef Ahmad
ID: gij817

Maintaining books information in server:
The book information and the records of different servives are maintained in xml files. It is scalable.
Client-Server Communication
Client communicate with server using rmi interface. We have used multithreading for operations requested from client side in server. Each client request is processed in a seperate thread. Each threading has full synchronization applied to it.
Measurement of performance of service
We measure performance of service using avarage processing time
Difficulties
Maintaining information in server was challenging. We tried to overcome it by using xml file to maintain information.

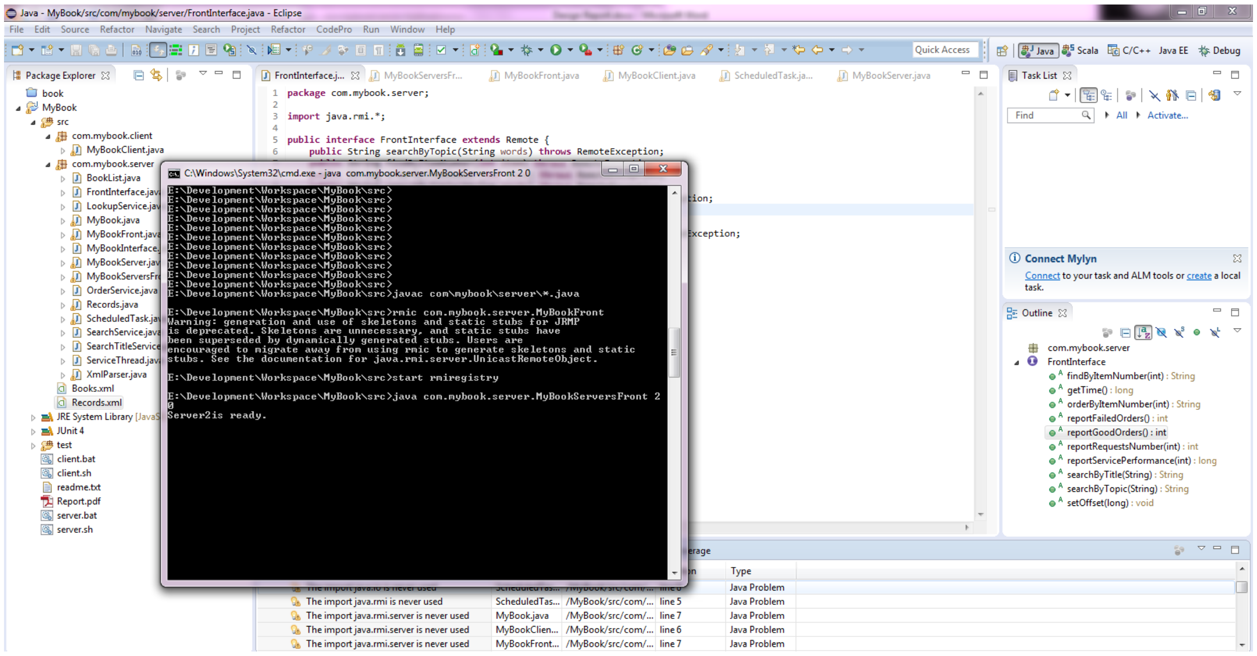
Testing:



C:\Windows\System32\cmd.exe - java com.mybook.server.MyBookServer

```
E:\Development\Workspace\MyBook\src>javac com\mybook\server\*.java

E:\Development\Workspace\MyBook\src>rmic com.mybook.server.MyBook
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

E:\Development\Workspace\MyBook\src>start rmiregistry

E:\Development\Workspace\MyBook\src>rmic com.mybook.server.MyBook
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

E:\Development\Workspace\MyBook\src>rmic com.mybook.server.MyBook
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

E:\Development\Workspace\MyBook\src>java com.mybook.server.MyBookServer
Server is ready.

E:\Development\Workspace\MyBook\src>javac com\mybook\server\*.java

E:\Development\Workspace\MyBook\src>rmic com.mybook.server.MyBook
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

E:\Development\Workspace\MyBook\src>start rmiregistry

E:\Development\Workspace\MyBook\src>java com.mybook.server.MyBookServer
Server is ready.

E:\Development\Workspace\MyBook\src>javac com\mybook\server\*.java

E:\Development\Workspace\MyBook\src>rmic com.mybook.server.MyBook
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

E:\Development\Workspace\MyBook\src>start rmiregistry

E:\Development\Workspace\MyBook\src>java com.mybook.server.MyBookServer
Server is ready.
_
```

Saeef Ahmad
ID: gij817

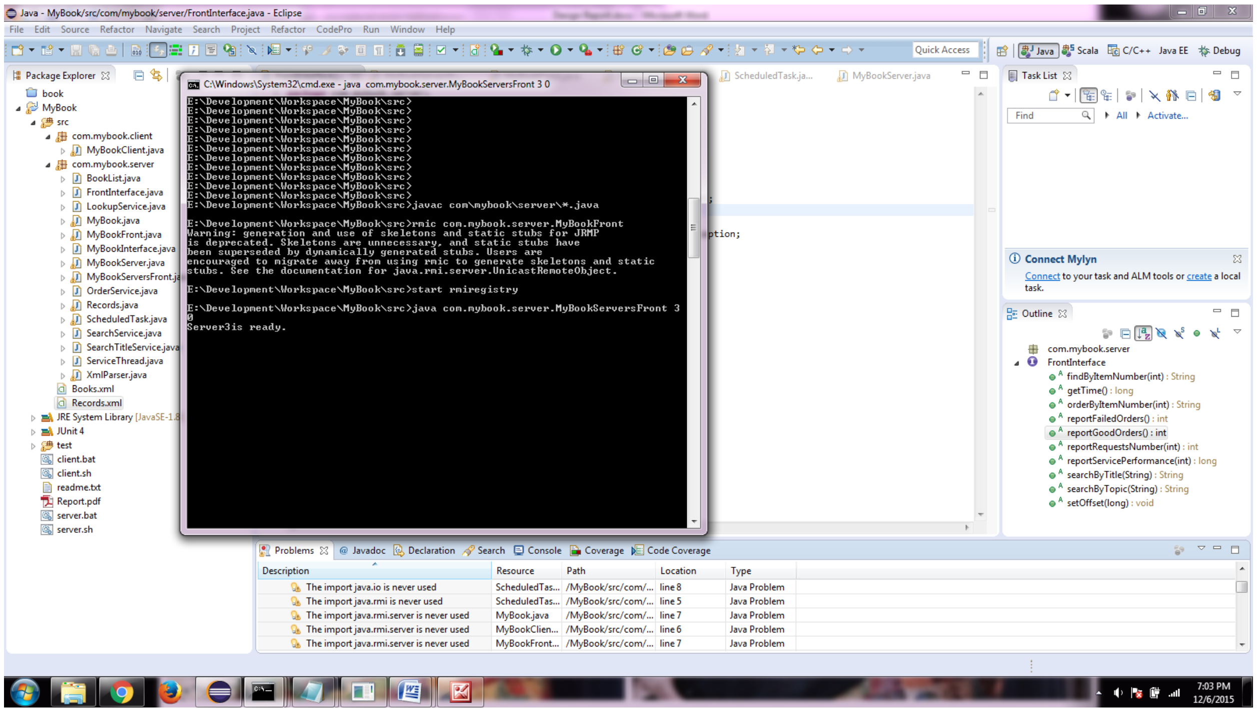


Java - MyBook/src/com/mybook/server/FrontInterface.java - Eclipse

```
package com.mybook.server;

import java.rmi.*;

public interface FrontInterface extends Remote {
    public String searchByTopic(String words) throws RemoteException;
```

C:\Windows\System32\cmd.exe - java com.mybook.server.MyBookServersFront 2 0

```
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>javac com\mybook\server\*.java

E:\Development\Workspace\MyBook\src>rmic com.mybook.server.MyBookFront
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

E:\Development\Workspace\MyBook\src>start rmiregistry

E:\Development\Workspace\MyBook\src>java com.mybook.server.MyBookServersFront 2
0
Server2is ready.
```



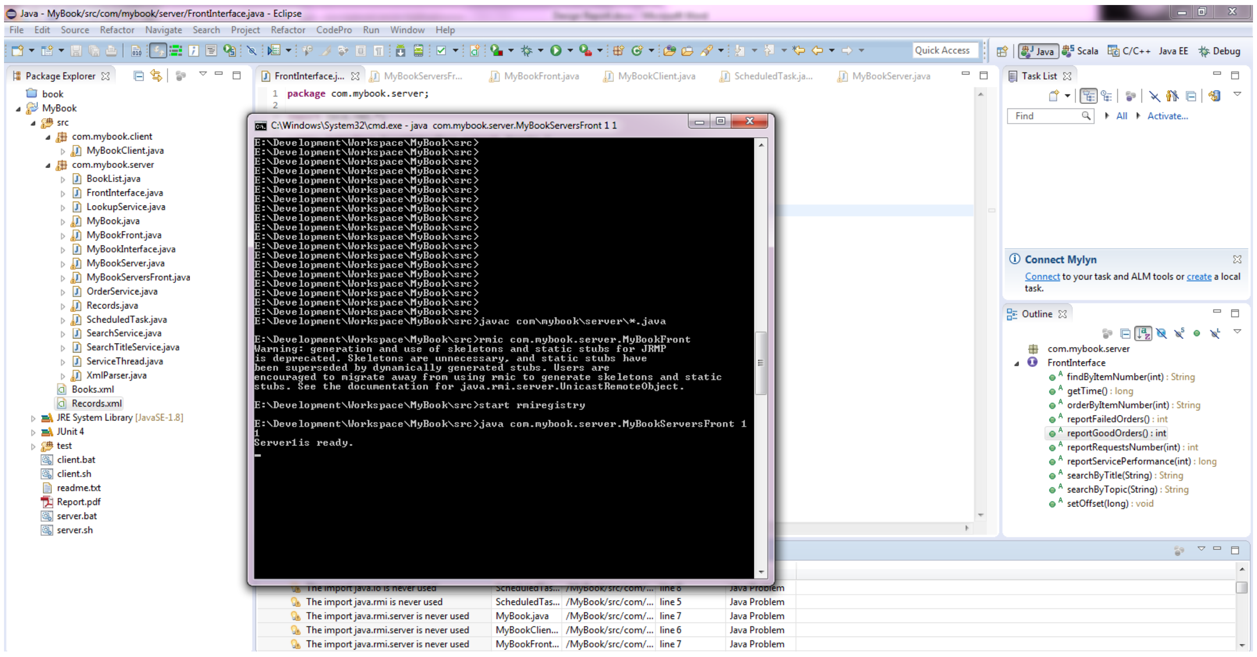
Java - MyBook/src/com/mybook/server/FrontInterface.java - Eclipse

C:\Windows\System32\cmd.exe - java com.mybook.server.MyBookServersFront 3 0

```
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>javac com\mybook\server\*.java

E:\Development\Workspace\MyBook\src>rmic com.mybook.server.MyBookFront
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

E:\Development\Workspace\MyBook\src>start rmiregistry

E:\Development\Workspace\MyBook\src>java com.mybook.server.MyBookServersFront 3
0
Server3is ready.
```

Description	Resource	Path	Location	Type
The import java.io is never used	ScheduledTas...	/MyBook/src/com/...	line 8	Java Problem
The import java.rmi is never used	ScheduledTas...	/MyBook/src/com/...	line 5	Java Problem
The import java.rmi.server is never used	MyBook.java	/MyBook/src/com/...	line 7	Java Problem
The import java.rmi.server is never used	MyBookClien...	/MyBook/src/com/...	line 6	Java Problem
The import java.rmi.server is never used	MyBookFront...	/MyBook/src/com/...	line 7	Java Problem

Saeef Ahmad
ID: gij817

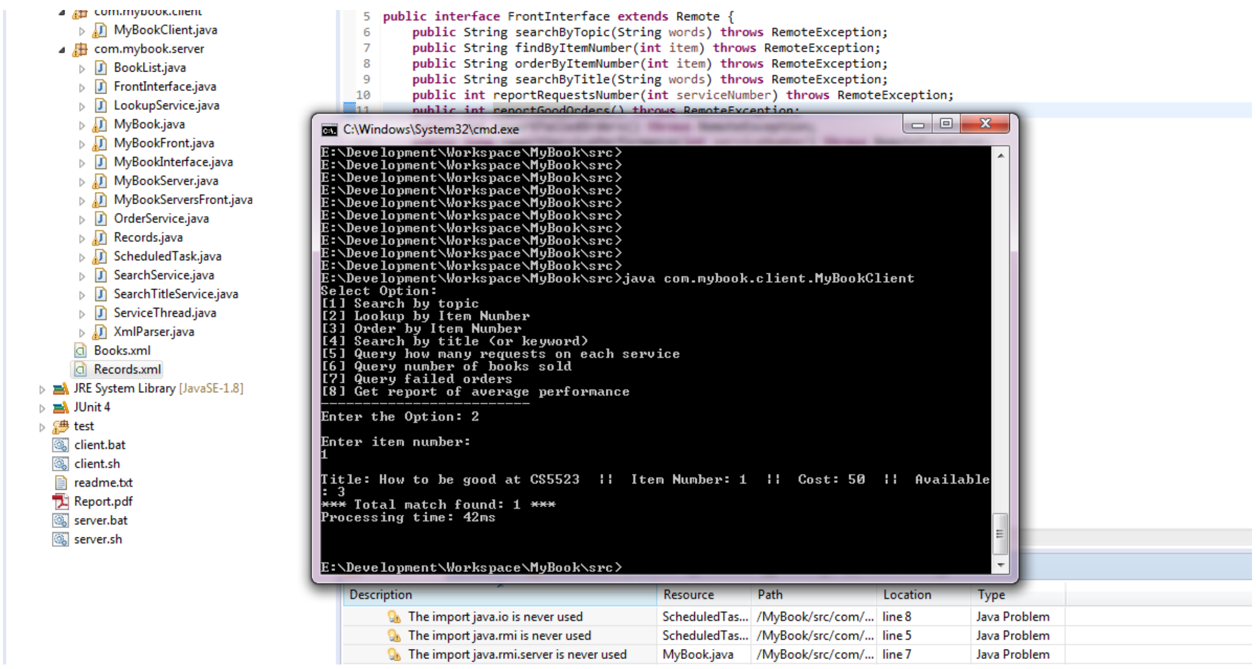


```
package com.mybook.server;
```

C:\Windows\System32\cmd.exe - java com.mybook.server.MyBookServersFront 1 1

```
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>javac com\mybook\server\*.java

E:\Development\Workspace\MyBook\src>rmic com.mybook.server.MyBookFront
Warning: generation and use of skeletons and static stubs for JRMP
is deprecated. Skeletons are unnecessary, and static stubs have
been superseded by dynamically generated stubs. Users are
encouraged to migrate away from using rmic to generate skeletons and static
stubs. See the documentation for java.rmi.server.UnicastRemoteObject.

E:\Development\Workspace\MyBook\src>start rmiregistry

E:\Development\Workspace\MyBook\src>java com.mybook.server.MyBookServersFront 1
1
Server1is ready.
_
```

Description	Resource	Path	Location	Type
The import java.io is never used	ScheduledTas...	/MyBook/src/com/...	line 8	Java Problem
The import java.rmi is never used	ScheduledTas...	/MyBook/src/com/...	line 5	Java Problem
The import java.rmi.server is never used	MyBook.java	/MyBook/src/com/...	line 7	Java Problem
The import java.rmi.server is never used	MyBookClien...	/MyBook/src/com/...	line 6	Java Problem
The import java.rmi.server is never used	MyBookFront...	/MyBook/src/com/...	line 7	Java Problem



```
public interface FrontInterface extends Remote {
    public String searchByTopic(String words) throws RemoteException;
    public String findByItemNumber(int item) throws RemoteException;
    public String orderByItemNumber(int item) throws RemoteException;
    public String searchByTitle(String words) throws RemoteException;
    public int reportRequestsNumber(int serviceNumber) throws RemoteException;
    public int reportGoodOrders() throws RemoteException;
```

C:\Windows\System32\cmd.exe

```
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>
E:\Development\Workspace\MyBook\src>java com.mybook.client.MyBookClient
Select Option:
[1] Search by topic
[2] Lookup by Item Number
[3] Order by Item Number
[4] Search by title (or keyword)
[5] Query how many requests on each service
[6] Query number of books sold
[7] Query failed orders
[8] Get report of average performance
-----------------------------------

Enter the Option: 2

Enter item number:
1

Title: How to be good at CS5523   ||  Item Number: 1   ||  Cost: 50   ||  Available
: 3
*** Total match found: 1 ***
Processing time: 42ms



E:\Development\Workspace\MyBook\src>
```

Description	Resource	Path	Location	Type
The import java.io is never used	ScheduledTas...	/MyBook/src/com/...	line 8	Java Problem
The import java.rmi is never used	ScheduledTas...	/MyBook/src/com/...	line 5	Java Problem
The import java.rmi.server is never used	MyBook.java	/MyBook/src/com/...	line 7	Java Problem