

Library Management System

Project Report

Rajankumar Patel(1115905)

Nachiket Patel(1058527)

Subject : DataBase System

CSCI-760

Fall 2016

Professor - Dr Huanying Gu

**ACKNOWLEDGMENT**

We are very thankful to all the professors of New York Institute of Technology for their support and motivation. Prof **Dr. Huanying** **Gu** is one of them, she not only teach us Database subject, but give us chance to represent our knowledge in the subject by giving course oriented project. We also want to thank to our friends and family for the support and encourage us.

Rajankumar Patel

1119505

Nachiket Patel

1058527

NYIT

**ABSTRACT**

*The library management system made less work to student, librarian and administrators. Primary element of the system is maintaining and organizing library stuff. This system may overcome many general problems which might occur in library services, the system adds reader, publisher and authors, it also keeps track of more frequent borrower, top most books and fine for the late returning book. The given search function could give user more attention.*

**Contents**

|  |  |  |
| --- | --- | --- |
|  | Acknowledgement |  |
|  | Abstract |  |
| Chapter 1 | Introduction |  |
| Chapter 2 | ER Diagram |  |
| Chapter 3 | Logical Design |  |
| Chapter 4 | Relation Database Model |  |
| Chapter 5 | Implementation of database and SQL Queries |  |
| Chapter 6 | Application design |  |
| Chapter 7 | User guide |  |
| Chapter 8 | Windows snapshots |  |
| Chapter 9 | Contribution of a member |  |

**Ch 1. Introduction**

The given application is about the library system, which contains the information about different branches and their students. Moreover, the application has two role models , first one is reader role, who can maintain the his library account such as add book in the account , knowing about fine for late submission, other side is admin, who can add more copy of books, add publisher and branch. The system is made by HTML5, CSS3 as FrontEnd and ASP.net C# as BackEnd. Microsoft SQL Server Management tools for database.

**Ch 2. ER Diagram**

**ER Data Model Design:**

1. **Book**: This entity contains information pertaining to books. Attributes of this entity are

ISBN

Bookid

bookTitle

Author

ISBN uniquely identifies different books.

2. **Publisher**: This entity contains information related to Publisher of the book. Each book as

single Publisher. Attributes of this entity are

Publisher ID

Publisher Location Publisher Name

Publication Date

**Publisher Id** uniquely identifies different instances of publisher.

3. **Author**: This entity contains information related to Author of the book. Each

book can have more than one author. Attributes of this entity are

Author Id

Author Name

**Author Id** uniquely identifies different instances of Author

4. **Branch**: This entity contains information related to Branch.

Branchid

Name

Location

Branch Id uniquely identifies different instances of Branch.

5. **Book Info**: This entity contains all the information about book like its ISBN, Author,

and Publisher etc.

ISBN

Tilte

Publisherid

Publication date

Authored

6. **Reader**: This entity contains all information about all readers.

Readerid

Name

Address

Phone

ReaderID uniquely identifies different users.

7. **Admin**: This entity contains information regarding the ID and Password of Administrator

which is used for Administrator login purpose. This entity is not related to any other

entity. Attributes of this entity are

Id

Password

8. **Borrow**: this entity has data about borrow date and associated book to a reader

borrowId

BorrowDate

ReturnDate

BookId

ReaderId

BorrowId is unique

9. **Reserved** : this entity contain the data about the all borrowers have fine to particular book

ReservedId ← unique

BorrowId

ReaderId

Fine in numerical

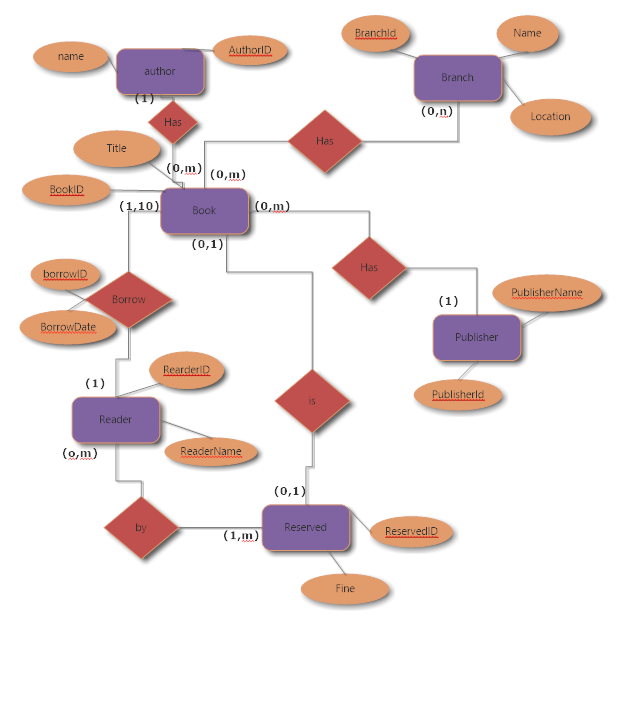
10. Branchbook: this entity has information about books in a branch

BranchbookId ← unique

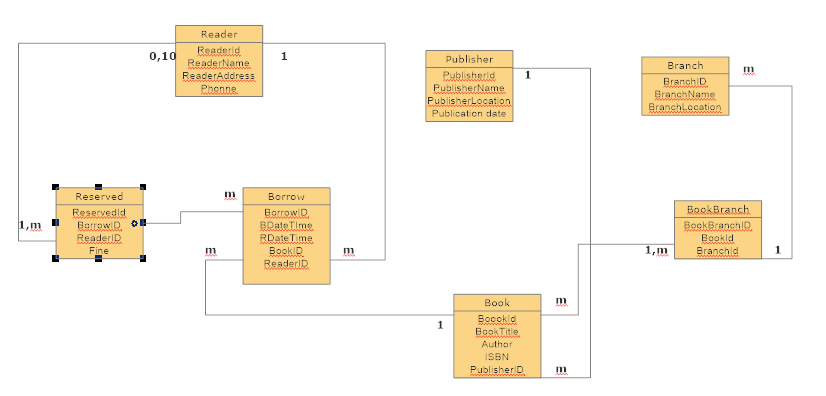
Branch ID

BookID

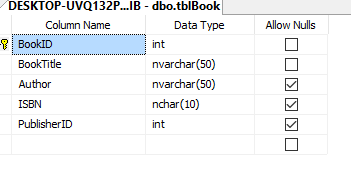
ER Diagram :



**Ch 3 Logical Design of the Database**



Book:



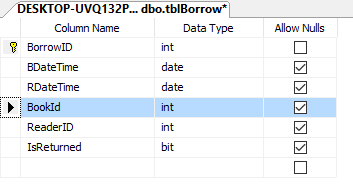
Primary Key: BookId

Auto\_increment +1

Foreign Key : 1 to many relationship with publisher.PublisherID

Integrity constraints :

Borrow:



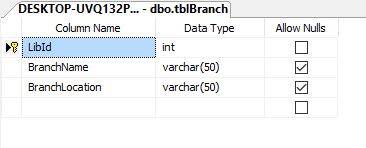
Primary Key: BorrowID

Foreign Key: m to m realtioship with Book.BookId

M to m realtionship with reader.ReaderID

Integrity constraints : referential integrity, entity integrity

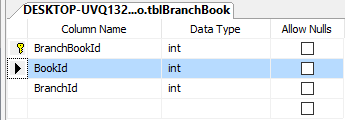
Branch:



Primary key = LidId

Integrity constraints : entity integrity

BranchBook



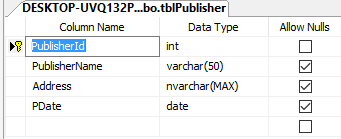
Primary key = BranchBookId

Foreign Key = BookId

BranchId

Integrity constraints : referential integrity, entity integrity

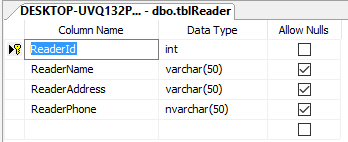
Publisher:



Primary key : publisherId

Integrity constraints : entity integrity

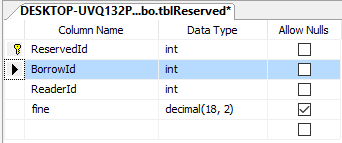
Reader:



Primary Key : Reader

Integrity constraints : entity integrity

Reserved :



Primary Key : ReservedId

Foreign key :1 to m relationship with Borrow. BorrowId

1 to 10 Relationship with Reader.ReaderId

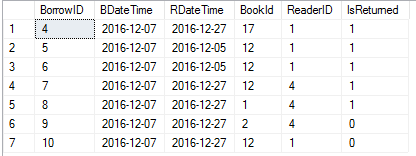
Integrity constraints : referential integrity, entity integrity

**Ch 4 Relational Database Design**

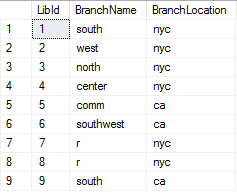
Book



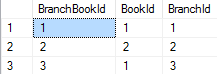
Borrow



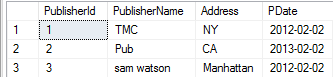
Branch



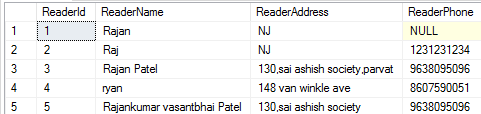
BranchBook



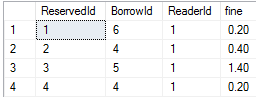
Publisher



Reader



Reservedw



**Ch 5. Implementation of Database and SQL Query**

**Create Query**

**Book**

USE [NYCLIB]

GO

CREATE TABLE [dbo].[tblBook](

[BookID] [int] IDENTITY(1,1) NOT NULL,

[BookTitle] [nvarchar](50) NOT NULL,

[Author] [nvarchar](50) NULL,

[ISBN] [nchar](10) NULL,

[PublisherID] [int] NULL,

CONSTRAINT [PK\_tblTest] PRIMARY KEY CLUSTERED

(

[BookID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[tblBook] WITH CHECK ADD CONSTRAINT [FK\_tblBook\_tblPublish] FOREIGN KEY([PublisherID])

REFERENCES [dbo].[tblPublisher] ([PublisherId])

GO

ALTER TABLE [dbo].[tblBook] CHECK CONSTRAINT [FK\_tblBook\_tblPublish]

GO

**Borrow:**

USE [NYCLIB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[tblBorrow] Script Date: 12/07/2016 11:24:22 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[tblBorrow](

[BorrowID] [int] IDENTITY(1,1) NOT NULL,

[BDateTime] [date] NULL,

[RDateTime] [date] NULL,

[BookId] [int] NULL,

[ReaderID] [int] NULL,

[IsReturned] [bit] NULL,

CONSTRAINT [PK\_Borrow] PRIMARY KEY CLUSTERED

(

[BorrowID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[tblBorrow] WITH CHECK ADD CONSTRAINT [FK\_tblBorrow\_tblBook] FOREIGN KEY([BookId])

REFERENCES [dbo].[tblBook] ([BookID])

GO

ALTER TABLE [dbo].[tblBorrow] CHECK CONSTRAINT [FK\_tblBorrow\_tblBook]

GO

ALTER TABLE [dbo].[tblBorrow] WITH CHECK ADD CONSTRAINT [FK\_tblBorrow\_tblReader] FOREIGN KEY([ReaderID])

REFERENCES [dbo].[tblReader] ([ReaderId])

GO

ALTER TABLE [dbo].[tblBorrow] CHECK CONSTRAINT [FK\_tblBorrow\_tblReader]

GO

ALTER TABLE [dbo].[tblBorrow] ADD CONSTRAINT [DF\_tblBorrow\_IsReturned] DEFAULT ((0)) FOR [IsReturned]

GO

**Branch:**

USE [NYCLIB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[tblBranch] Script Date: 12/07/2016 11:26:47 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[tblBranch](

[LibId] [int] IDENTITY(1,1) NOT NULL,

[BranchName] [varchar](50) NULL,

[BranchLocation] [varchar](50) NULL,

CONSTRAINT [PK\_Branch] PRIMARY KEY CLUSTERED

(

[LibId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

**BranchBook:**

USE [NYCLIB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[tblBranchBook] Script Date: 12/07/2016 11:27:15 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[tblBranchBook](

[BranchBookId] [int] IDENTITY(1,1) NOT NULL,

[BookId] [int] NOT NULL,

[BranchId] [int] NOT NULL,

CONSTRAINT [PK\_BranchBook] PRIMARY KEY CLUSTERED

(

[BranchBookId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[tblBranchBook] WITH CHECK ADD CONSTRAINT [FK\_tblBranchBook\_tblBook] FOREIGN KEY([BookId])

REFERENCES [dbo].[tblBook] ([BookID])

GO

ALTER TABLE [dbo].[tblBranchBook] CHECK CONSTRAINT [FK\_tblBranchBook\_tblBook]

GO

ALTER TABLE [dbo].[tblBranchBook] WITH CHECK ADD CONSTRAINT [FK\_tblBranchBook\_tblBranch] FOREIGN KEY([BranchId])

REFERENCES [dbo].[tblBranch] ([LibId])

GO

ALTER TABLE [dbo].[tblBranchBook] CHECK CONSTRAINT [FK\_tblBranchBook\_tblBranch]

GO

**Publisher:**

USE [NYCLIB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[tblPublisher] Script Date: 12/07/2016 11:27:48 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[tblPublisher](

[PublisherId] [int] IDENTITY(1,1) NOT NULL,

[PublisherName] [varchar](50) NULL,

[Address] [nvarchar](max) NULL,

[PDate] [date] NULL,

CONSTRAINT [PK\_Publisher] PRIMARY KEY CLUSTERED

(

[PublisherId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

**Reader:**

USE [NYCLIB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[tblReader] Script Date: 12/07/2016 11:28:33 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

SET ANSI\_PADDING ON

GO

CREATE TABLE [dbo].[tblReader](

[ReaderId] [int] IDENTITY(1,1) NOT NULL,

[ReaderName] [varchar](50) NULL,

[ReaderAddress] [varchar](50) NULL,

[ReaderPhone] [nvarchar](50) NULL,

CONSTRAINT [PK\_Reader] PRIMARY KEY CLUSTERED

(

[ReaderId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

SET ANSI\_PADDING OFF

GO

**Reserved:**

USE [NYCLIB]

GO

/\*\*\*\*\*\* Object: Table [dbo].[tblReserved] Script Date: 12/07/2016 11:28:46 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[tblReserved](

[ReservedId] [int] IDENTITY(1,1) NOT NULL,

[BorrowId] [int] NOT NULL,

[ReaderId] [int] NOT NULL,

[fine] [decimal](18, 2) NULL,

CONSTRAINT [PK\_Reserved] PRIMARY KEY CLUSTERED

(

[ReservedId] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[tblReserved] WITH CHECK ADD CONSTRAINT [FK\_tblReserved\_tblReader] FOREIGN KEY([ReaderId])

REFERENCES [dbo].[tblReader] ([ReaderId])

GO

ALTER TABLE [dbo].[tblReserved] CHECK CONSTRAINT [FK\_tblReserved\_tblReader]

GO

• Provide the SQL statements that query the database

INSERT INTO [NYCLIB].[dbo].[tblBook]

([BookTitle]

,[Author]

,[ISBN]

,[PublisherID])

VALUES

(<BookTitle, nvarchar(50),>

,<Author, nvarchar(50),>

,<ISBN, nchar(10),>

,<PublisherID, int,>)

GO

INSERT INTO [NYCLIB].[dbo].[tblBorrow]

([BDateTime]

,[RDateTime]

,[BookId]

,[ReaderID]

,[IsReturned])

VALUES

(<BDateTime, date,>

,<RDateTime, date,>

,<BookId, int,>

,<ReaderID, int,>

,<IsReturned, bit,>)

GO

INSERT INTO [NYCLIB].[dbo].[tblBranch]

([BranchName]

,[BranchLocation])

VALUES

(<BranchName, varchar(50),>

,<BranchLocation, varchar(50),>)

GO

INSERT INTO [NYCLIB].[dbo].[tblBranchBook]

([BookId]

,[BranchId])

VALUES

(<BookId, int,>

,<BranchId, int,>)

GO

INSERT INTO [NYCLIB].[dbo].[tblPublisher]

([PublisherName]

,[Address]

,[PDate])

VALUES

(<PublisherName, varchar(50),>

,<Address, nvarchar(max),>

,<PDate, date,>)

GO

INSERT INTO [NYCLIB].[dbo].[tblReader]

([ReaderName]

,[ReaderAddress]

,[ReaderPhone])

VALUES

(<ReaderName, varchar(50),>

,<ReaderAddress, varchar(50),>

,<ReaderPhone, nvarchar(50),>)

GO

INSERT INTO [NYCLIB].[dbo].[tblReserved]

([BorrowId]

,[ReaderId]

,[fine])

VALUES

(<BorrowId, int,>

,<ReaderId, int,>

,<fine, decimal(18,2),>)

GO

**Ch 6. Application design  
Admin Function :**

**Add Book:**

insert into tblBook values ('"+BookTitle+"','"+Author+"','"+ISBN+"',"+PublisherID+”;

**Search Book:**

SELECT \* FROM tblBook as bo inner join tblBorrow as b on bo.BookID=b.BookId inner join tblReader as r on b.ReaderID=r.ReaderId where BookTitle like “+ bookTitle +” or " author like " + Author + '";

**Add New Reader:**

insert into tblReader values ('"+name+"','"+address+"','"+phone+"');

**Add Branch:**

insert into tblBranch (BranchName,BranchLocation) values ('"+name+"','"+location+"');

**Add Publisher:**

insert into tblPublisher values ('" + PublisherName + "','" + PublisherLocation + "','"+PublishDate+"')";

**Branch Information:**

"select \* from tblBranch";

Frequent Borrowers :

**Top Borrow**

“select r.ReaderId,r.ReaderName,COUNT(r.ReaderID) AS read\_Count from tblBorrow as b INNER JOIN tblReader AS R on b.ReaderID = R.ReaderId GROUP BY r.ReaderId,r.ReaderName ORDER BY read\_Count desc";

**Top Bookes**

“ select bo.BookID,bo.BookTitle,count(b.BookId) AS Book\_Count from tblBorrow as b INNER JOIN tblBook AS bo on b.BookId = bo.BookID GROUP BY bo.BookID,bo.BookTitle ORDER

BY Book\_Count desc ";

**Average Fine:**

select r.ReaderId,r.ReaderName, SUM(fine)/COUNT(\*) as avgFine from tblReserved as res inner join tblReader as r on res.ReaderId = r.ReaderId group by r.ReaderId,r.ReaderName";

**Reader Function:**

**SearchBook:**

select \* from tblBook as b inner join tblPublisher as p on b.PublisherID = p.PublisherId" where p.PublisherName like '%" + publisher + "%';

**Book CheckOut:**

"insert into tblBorrow values('"+today+ "','" + release + "'," + BookID + "," + readerID+ ",0)";

**Book Return:**

update tblBorrow set IsReturned = 1 where BorrowID=" + BorrowID ;

**Book Reserve:**

"SELECT \* from tblBorrow as b inner join tblBook as bo on bo.BookID=b.BookId inner join tblPublisher as p on p.PublisherId=bo.PublisherID Where b.ReaderID =" + ReaderID + "";

**Fine:**

select bk.BookID,bk.BookTitle,res.BorrowId,res.ReservedId,res.fine from tblReserved as res inner join NYCLIB.dbo.tblBorrow as bor on res.BorrowId=bor.BorrowID inner join tblBook as bk on bor.BookId = bk.BookID where res.ReaderId="+ReaderID+"";

**Log Out**

Session["admin"] = "admin";

**Ch 7. User guide**

**admin:**

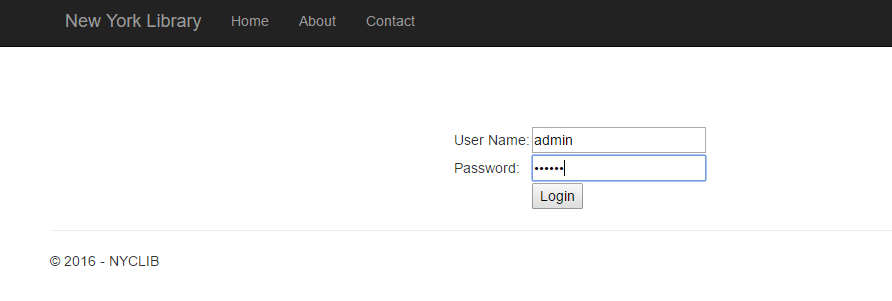
1. Open web application
2. Login into system as admin
3. Search book then submit
4. Add book then submit
5. Add publisher than submit
6. Add branch then submit
7. Branch information
8. Frequant borrower and books borrower
9. Aveage fine
10. Log out

**Reader**:

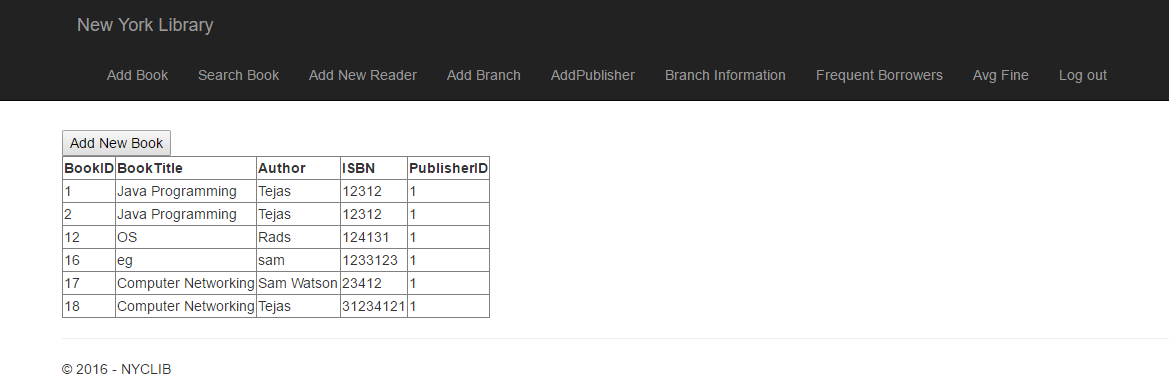
1. Open web applciation
2. Login into system as reader
3. Search book submit
4. Book checkout
5. Book return
6. Book reserved
7. Fine
8. Log out

**Ch 8. Window snapshots**

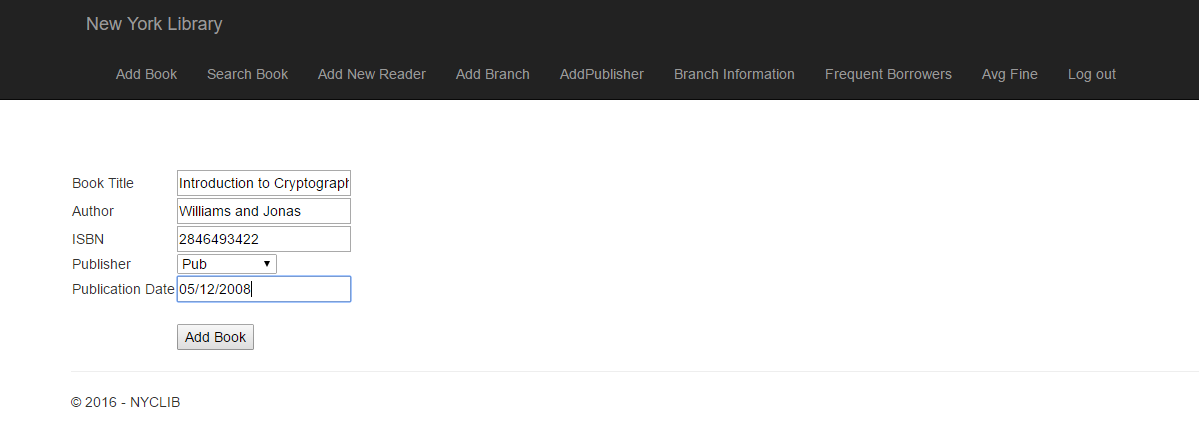
1. **Home Page**



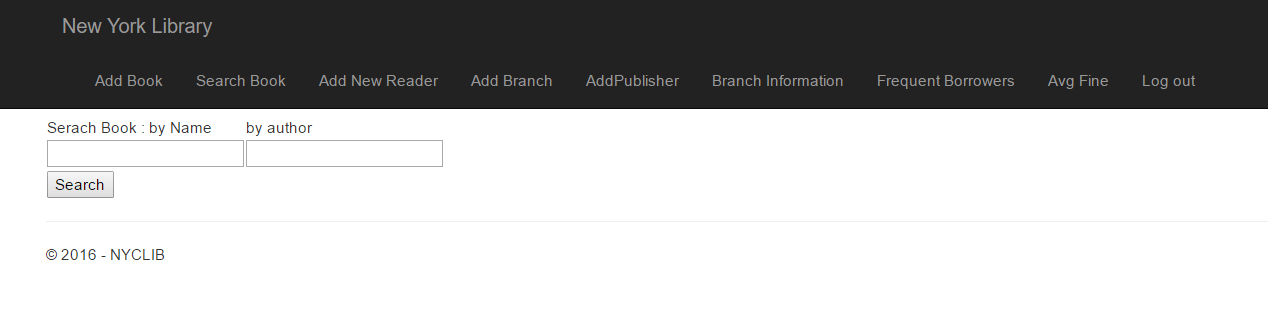
**2. Admin Page**



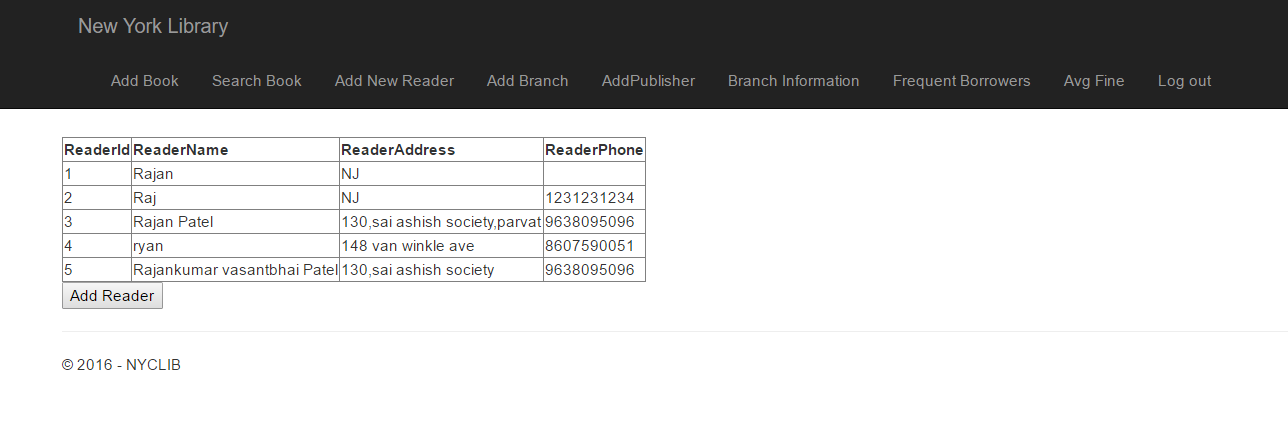
**3. Add a new book(Admin)**



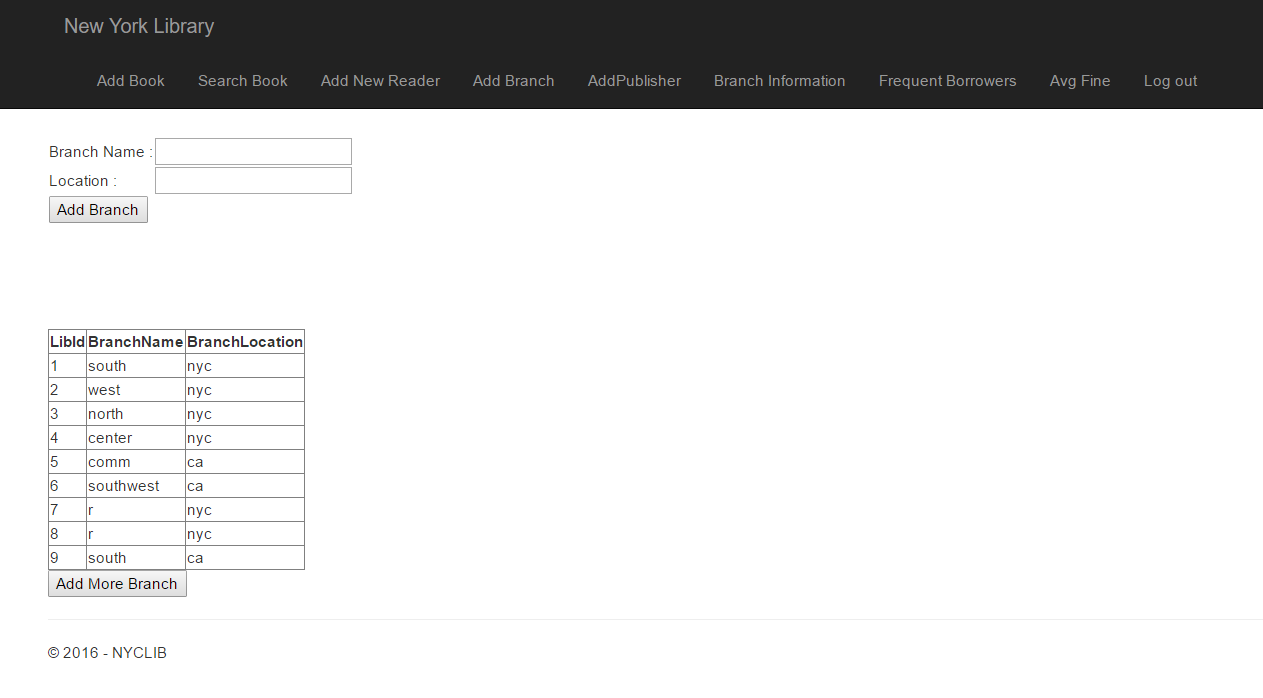
**4. Search book(Admin)**



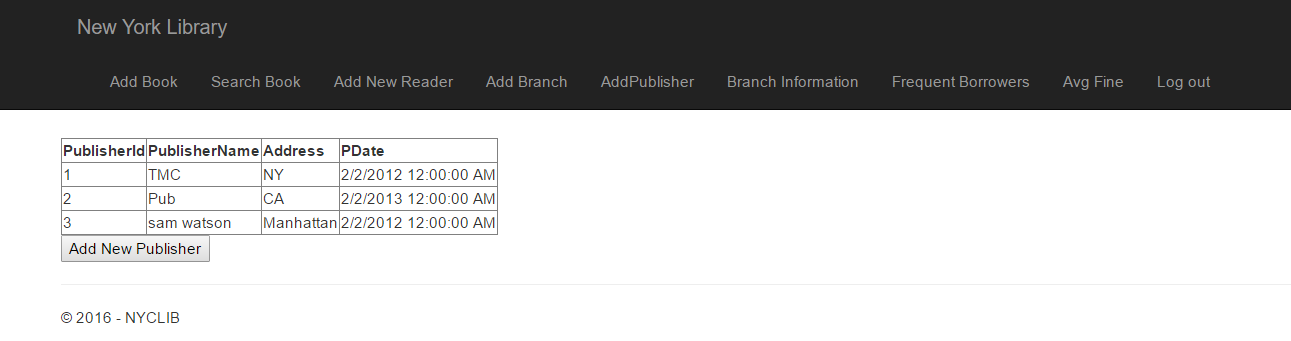
**5. Add a new reader(Admin)**



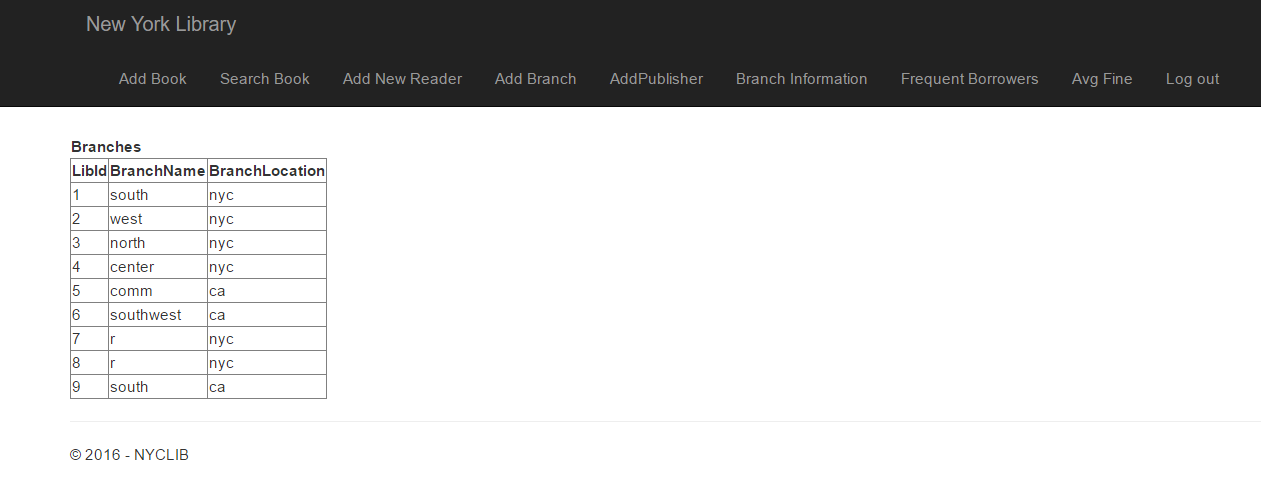
**6. Add a new branch(Admin)**



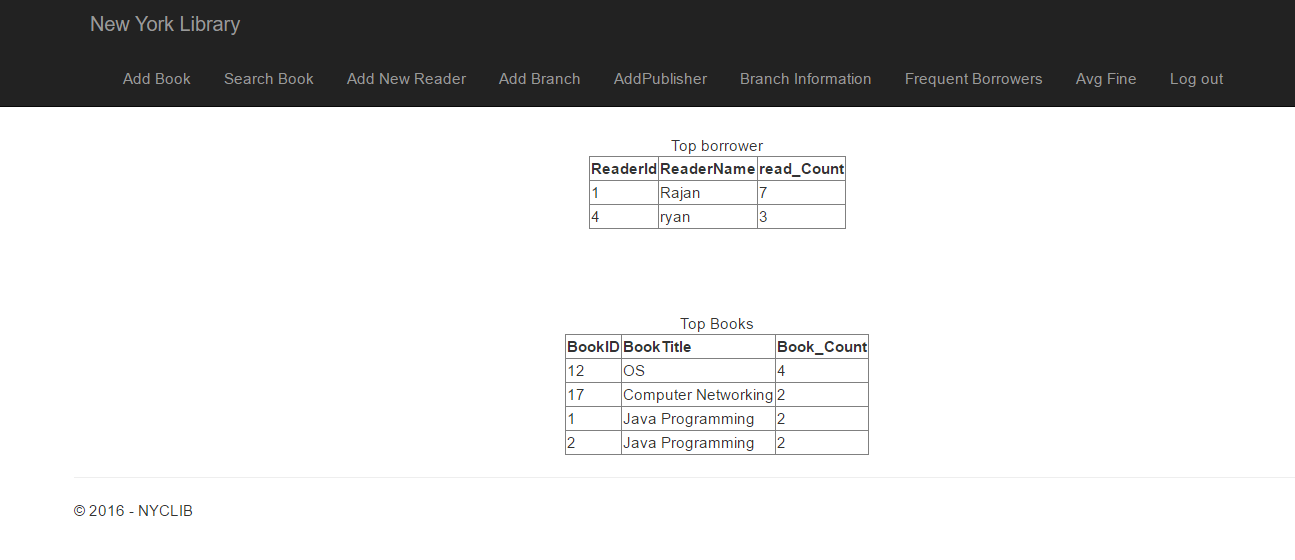
**7. Add a new publisher(Admin)**



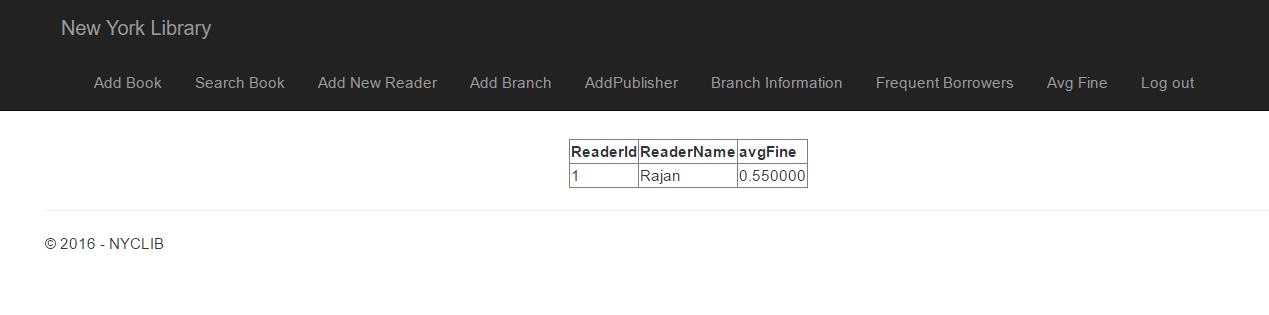
**8. Add a new branch(Admin)**



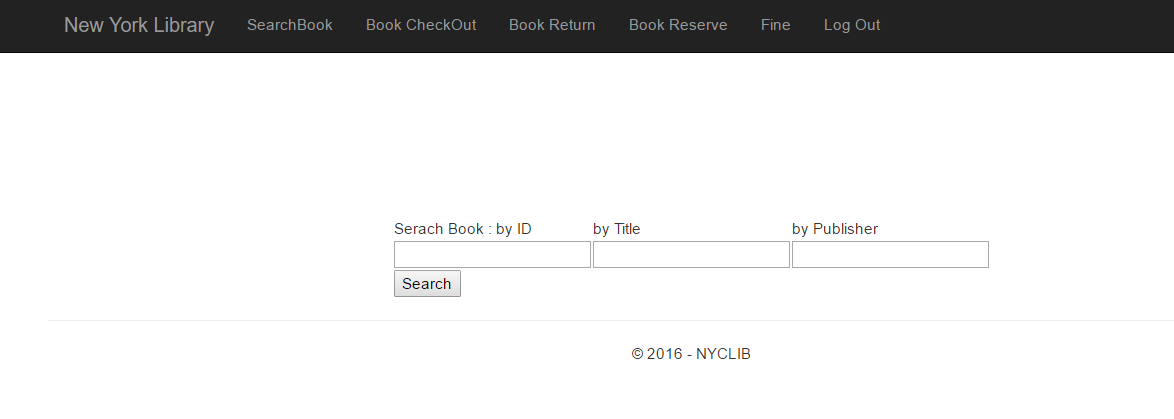
**9. Frequent borrowers(Admin)**



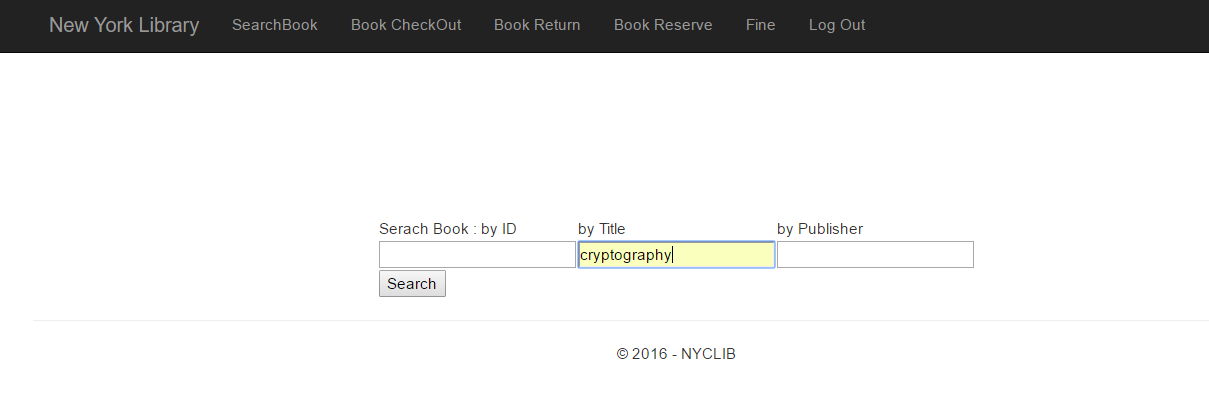
**10. Average Fine for a individual reader(Admin)**

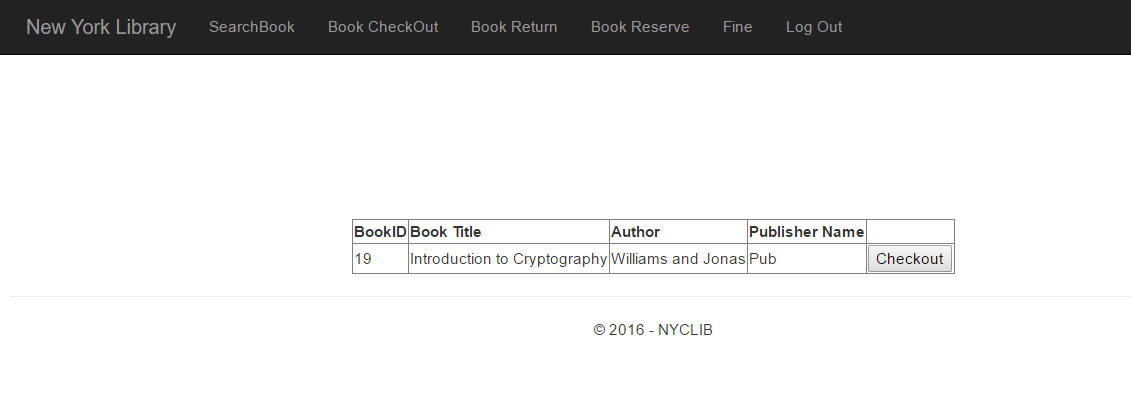


**11. Reader(User) page**

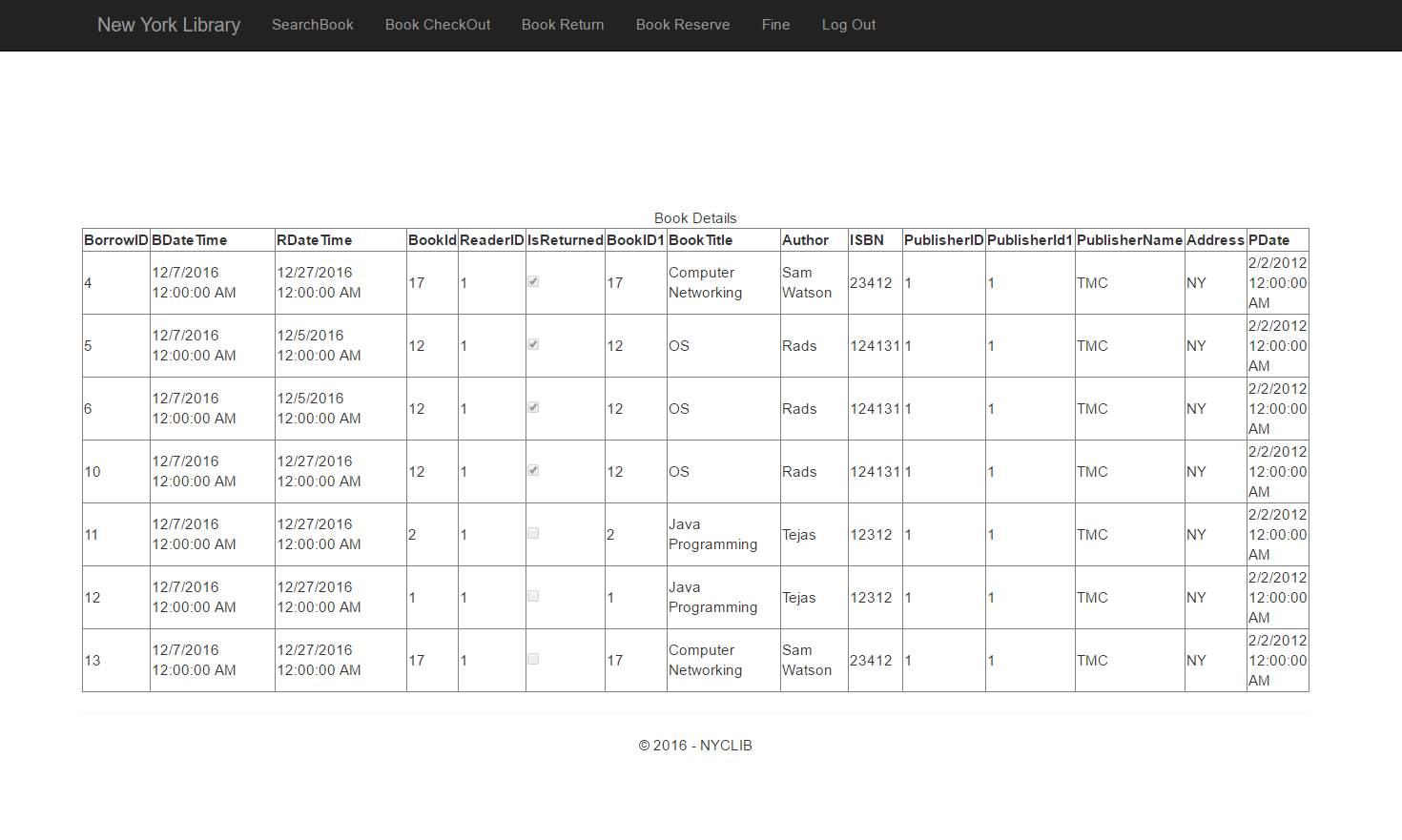


**12. Search book(User)**

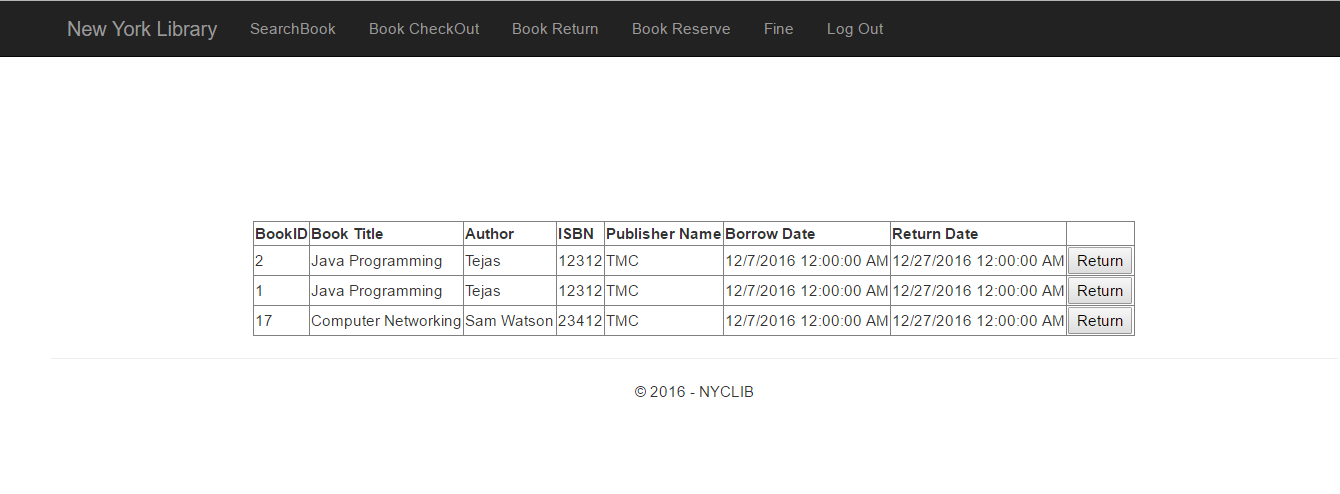




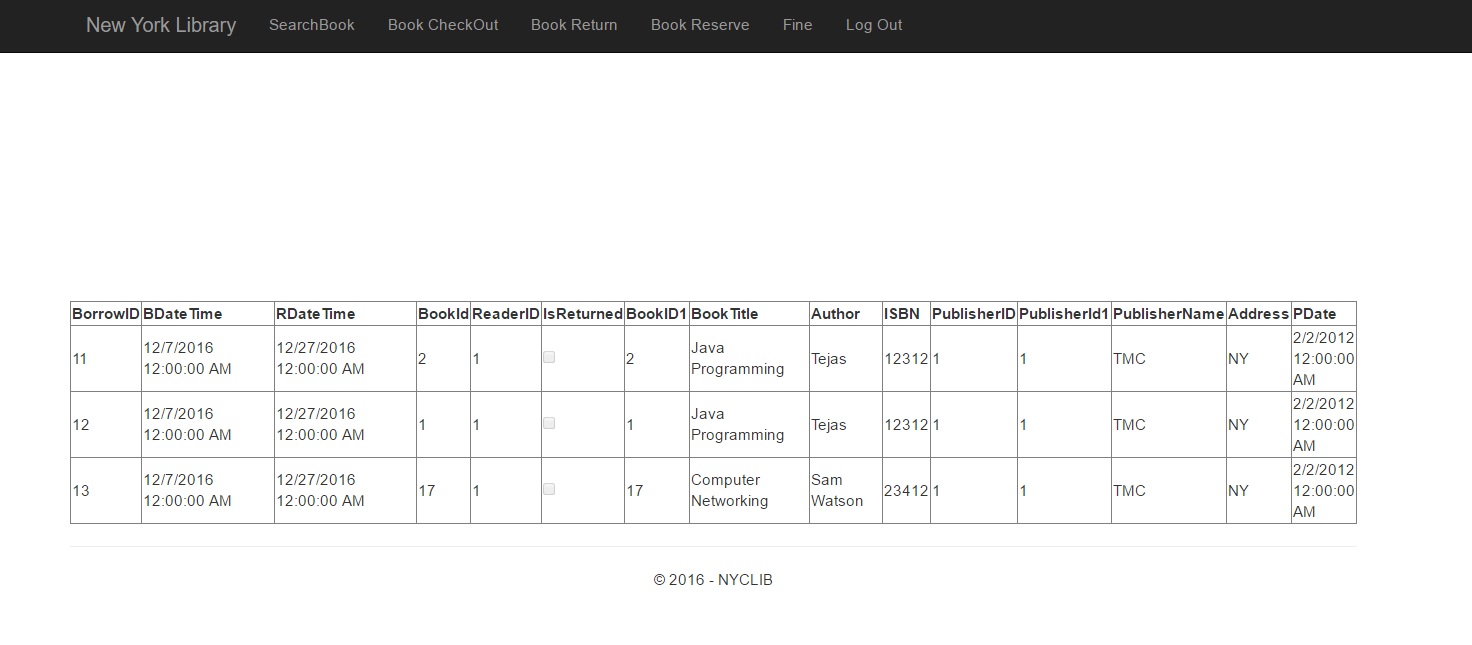
**13. User Log**



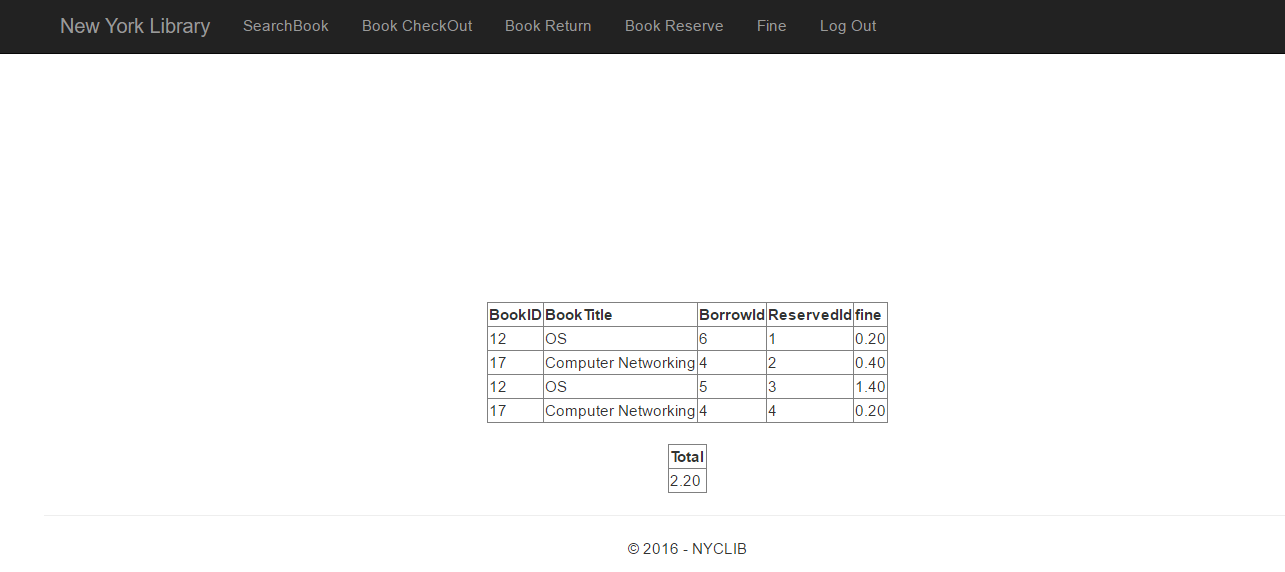
**14. Return Book(User)**



**15. Book reserved(User)**



**16. User Fine(User)**



**Ch 9. Specify in detail every team member’s work.**

**Rajankumar Patel(1115905) :**

Implementation of system in ASP.net

Mapping relation Diagram to ER Diagram

UI HTML5

Testing

**Nachiket Patel(1058527) :**

Data base Design

ER diagram

Queries

Else report

Reference:

<https://docs.google.com/document/d/1PI7KEkbqgs97PXHZ84QhmwHYebF1Vy-Ll4aEE7GyKW8/edit>

<https://msdn.microsoft.com/en-us/library/>

<http://www.w3schools.com/sql/>