|  |  |  |
| --- | --- | --- |
| UnitecHorizontalLogo | Assignment Library Database Project  Design and planning stage  ISCG 5421 Programming Principles and Practice | |
| Members  Serman Perry  Natalia Cardillo  Aubrey Fernandes  Christain Hayward |  |  |

Contents

* Meetings 1-3 minutes so far including division of work load
* What is required-- a built point version of scenario and specifications
* Flow charts-- separating users, dealing with fines
* UML Sequence Diagrams
* Class diagrams
* SQL diagrams
* UML Use Case Diagram

Meeting 1

Date 15/04/15 Time 2pm-3pm

Topics covered

* Discussed overall plan
* Broke assignment down into six stages parts :

1. GUI—Frontend e.g. Where user **inputs** searches for a book
2. Program-- Middle where that search is turned into a method
3. Database –-Backend where the method is turned into a and a table is queried.
4. Results of that query stored
5. Query results returned to a method
6. Method **outputs** to

Important decisions made

* To use SQL lite and to link the data base to Java rather than using binary files
* We need to research how these queries work with Java

Meeting 2

Date 22/04/15 Time 1pm-4pm

Topics covered

* Discussed what exactly we need (re wrote what is required as built points)
* Problem solving (reports and passwords) through SQL
* Flow charts
* Methods classes objects instance variables and how they will all interact

Problem solving topics discussed

* How do we separate users? i.e receptionist, admin, customer?
* How do we structure out SQL tables?
* How do we deal with lost books?
* How do we deal with overdue books?
* Do we have just a user class?
* What common functions will each user use? Ways to reduce code duplication.
* Do we have class for each user?
* What methods do we need and what will they look like?
* Do we need a password for each user?
* Each user will be a class which will have a username as an instance variable
* How can we each work on code separately – Git Hub? Google docs?
* What has been done before that we can base our design on what has been done previously?
* Can receptionists and admin also get books out? How would we do that?

Meeting 2

Date 11/05/2015 Time 12pm- 2pm

Topics covered

* Who is doing what?

Outcome

* Sherman is building database
* Christian is building GUI
* Natalia is building reports
* Albury is

Overview -- What Is Required

**Receptionist -- Has the power to:**

* Register customer
* Search customer table
* Enter the name
* Enter address
* Enter contact number
* Change the status of borrowed books to returned books

**Customer – Has power to:**

* Search books by title topic, author publisher
* Borrow books

**Administrator – Has power to:**

* Search books by title topic, author publisher
* Add book from system
* Delete book from system
* Modify/Update the details of the book from system
* Add supplier to database
* Update supplier to database
* Delete supplier to database
* Add book to database
* Delete book to database
* Create username and password for the receptionist.

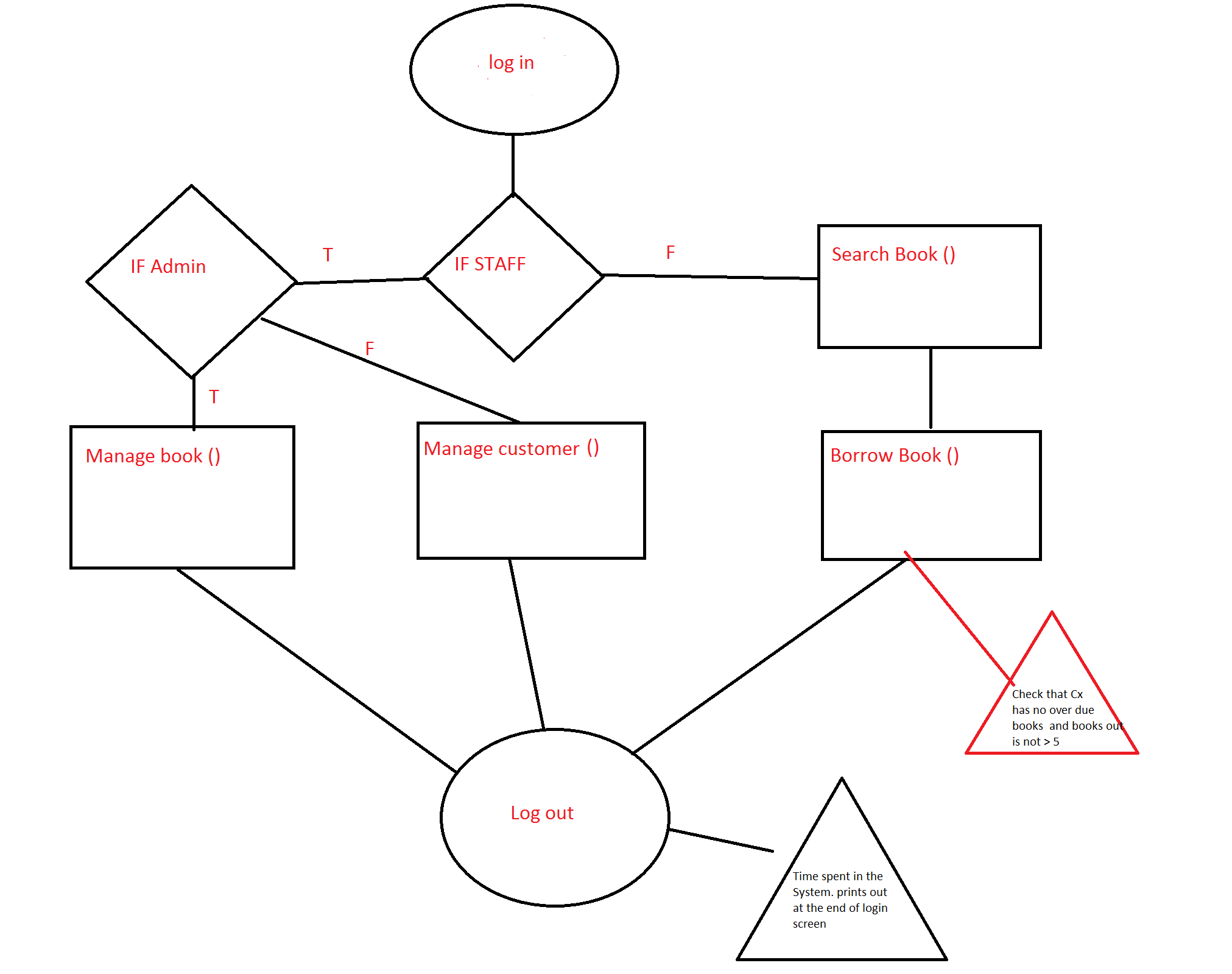
**System – things we need the system to be able to do**

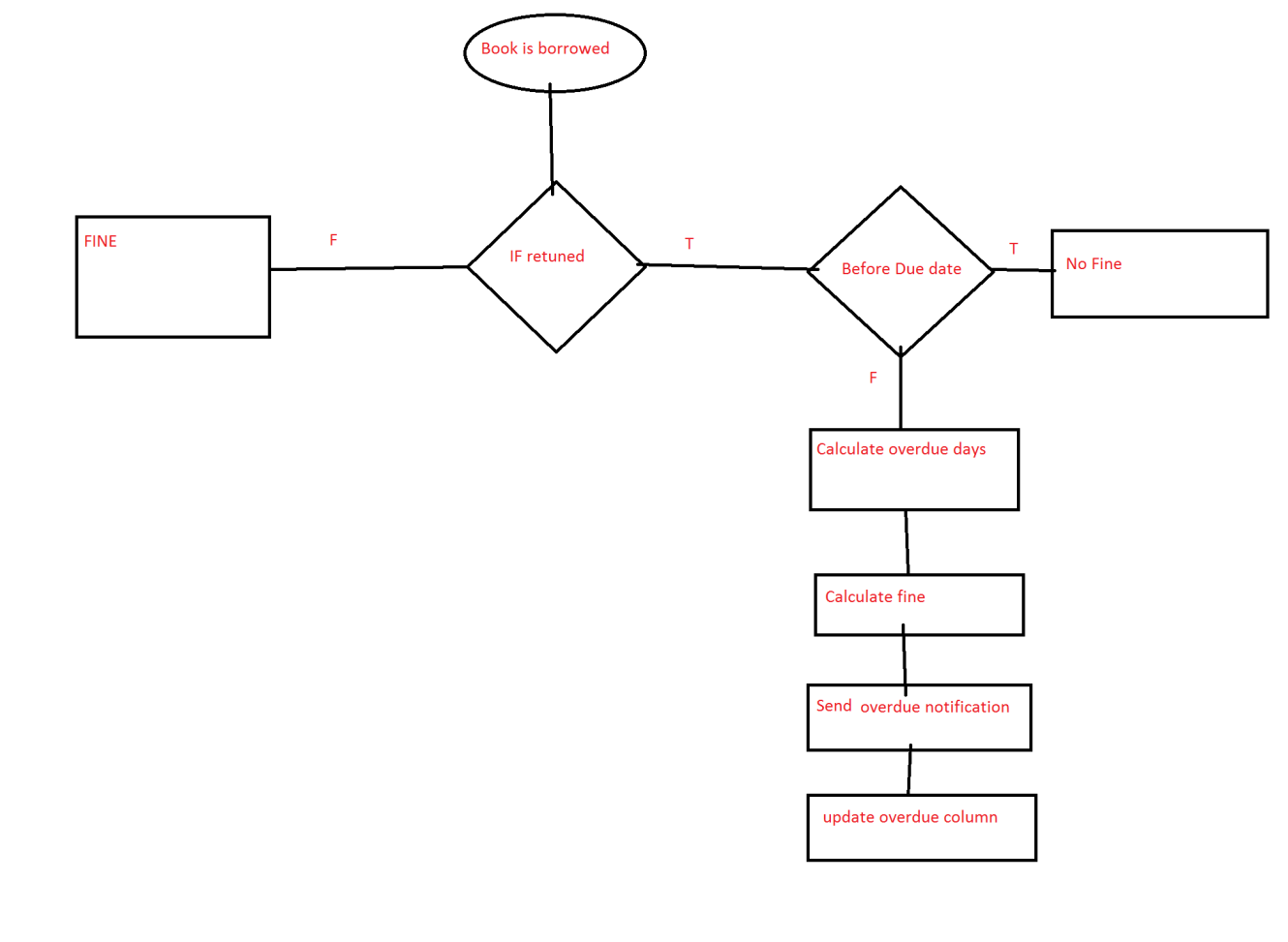
* Track Customers’ entry and exit time to the library system
* Restrict the amount of books a customer can borrow (max 5)
* Track length that customer is borrowing book for
* Fine 60cents per day that book is borrowed past due date
* Notification sent to customer about the fine
* If customer lose book a fine equals book’s cost will be added

**Reports – Different reports we will need to run**

* Book lending report (Daily, Monthly, Annually),
* fine report (book lost, late return with fine),
* book purchased report
* Optional extra reports-- Popular books report Resources Collection Report

Separating Users--Defining What They Can Do

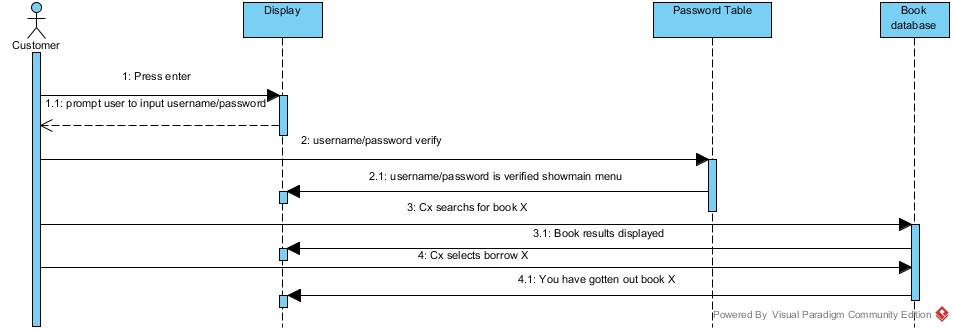


Calculating Fine 

UML Sequence Diagrams

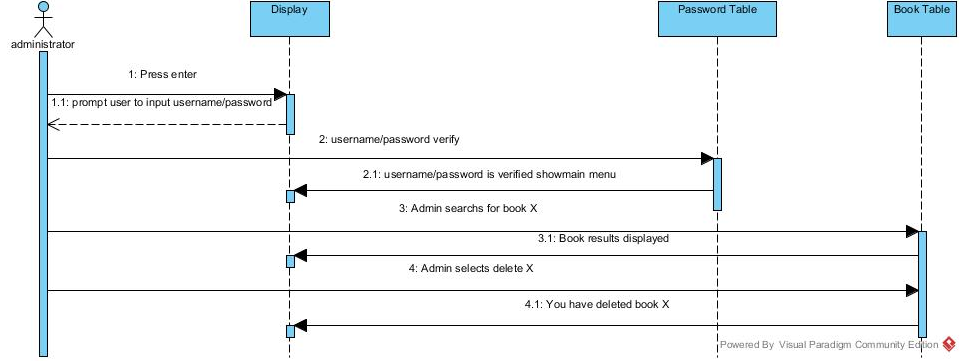
**Customer**

Example of customer searching and borrowing book



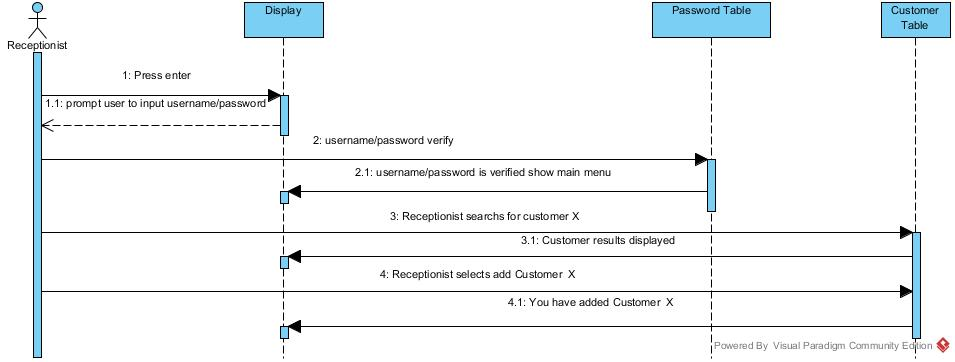
**Admin**

Example of admin searching and deleting book



**Receptionist**

Example of receptionist adding new customer

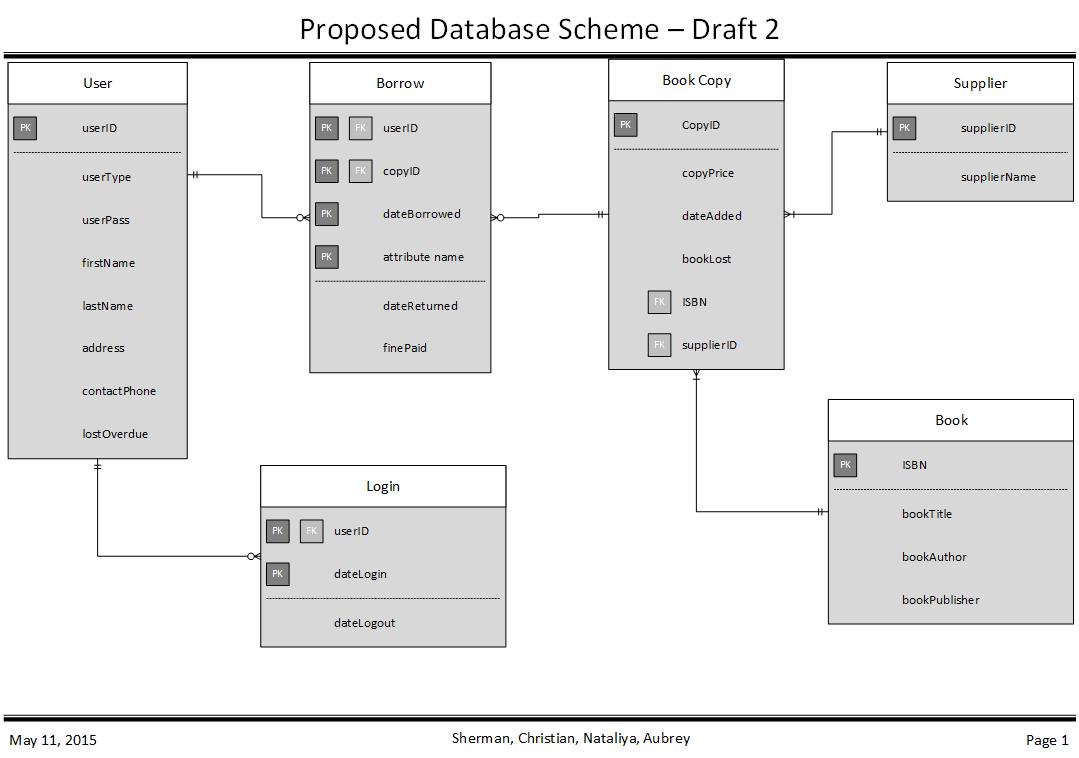


Class Diagram

|  |
| --- |
| **Customer** |
| +Title – String |
| +Author – String |
| +Publisher - String |
| +SearchBooks()- void |
| +BorrowBooks() - void |

|  |
| --- |
| **Admin** |
| +Title – String |
| +Author –String |
| +Publisher - String |
| +SearchBooks() - void |
| -AddDatabase - add |
| -DeleteDatabase() - boolean |
| -AddSupplier - add |

|  |
| --- |
| **Receptionist** |
| -CustomerName - String |
| -CustomerAddress - String |
| -CustomerNumber - int |
| -DeleteDatabase() - boolean |
| -AddDatabase - add |



|  |  |  |
| --- | --- | --- |
| **USER** | |  |
| userId | PK int | 12345 |
| userType | Char | Admin |
| UserPassword | Variable character |  |
| firstName | Variable character |  |
| lastName | Variable character |  |
| Address | Variable character |  |
| contactPhone | int |  |
| lostOverdue | int |  |

|  |  |
| --- | --- |
| **Borrow** | |
| userID | PK, FK int |
| CopyID | PK, FK int |
| dateBorrowed | PK date |
| dateReturned | Date |
| finePaid | money |

|  |  |
| --- | --- |
| **Book Copy** | |
| CopyID | PK, int |
| Copyprice | Money |
| DateAdded | Date |
| BookLost | Boolean |
| ISBN | Int |
| SupplierID | FK, Variable character |

|  |  |
| --- | --- |
| **Supplier** | |
| SupplierID | PK, Variable character |
| SupplierName | Variable character |

|  |  |
| --- | --- |
| **Book** | |
| ISBN | PK, Int |
| bookTitle | Variable character |
| BookAuthor | Variable character |
| BookPublisher | Variable character |

|  |  |
| --- | --- |
| **Login** | |
| userID | userID PK, FK |
| dateLogin | Date, PK |
| dateLogout | date |

