Library Database Management
370 Project Phase II

Vicky Holcomb
Nate Priddy
Joseph Fitzpatrick
Matthew Harris
## Contents

3 Problem Statement & Requirements Definition ................................................................. 3

Conceptual Models: ............................................................................................................. 7
   Entity Relationship Model ............................................................................................... 7
   Object Model Diagram .................................................................................................... 8
   Use-Case Diagram ........................................................................................................... 9
   Use-Cases ....................................................................................................................... 10

Logical Models: .................................................................................................................... 12
   Relational Database Schema .......................................................................................... 12
   Normalization Diagrams .................................................................................................. 14
   Scenarios with Event Trace Diagrams ............................................................................ 23

Data Dictionary .................................................................................................................... 34

Physical Models: ................................................................................................................ 37
   Implementation Narrative ............................................................................................... 37
   Features Summary Table .............................................................................................. 39
   Menu Hierarchy .............................................................................................................. 40

Deployment .......................................................................................................................... 41

Testing .................................................................................................................................. 41

Final Remarks .................................................................................................................... 42

Appendices: ........................................................................................................................ 43
   Table Definitions and Data Contents ............................................................................ 43
   User Views - Forms ........................................................................................................ 47
   User Views - Reports ...................................................................................................... 59
   SQL Statements ............................................................................................................. 64

Table Creation .................................................................................................................... 64

Stored Procedures .............................................................................................................. 68

Installation Guide .............................................................................................................. 75

User Manual ........................................................................................................................ 76
Team Members and their Responsibilities

Team Members: Vicky Holcomb, Nate Priddy, Joseph Fitzpatrick, Matthew Harris

Team Leader: Vicky Holcomb

Responsibilities (Proposal):
All: General project design including deciding what tables are going to be necessary in the database, their relationships, requirements as far as what will be enforced and data types, and the reports that will be generated.

Vicky: Background, requirements, problem statement, responsibilities, and overall polishing of data presented.

Nate: Creation of the ERD diagram after the team decided on what the database should include. He also created the use case diagram.

Joseph: Use Cases and a narrative description of each.

Matthew: Use Cases and a narrative description of each.

Responsibilities (Phase I):
Vicky: Data dictionary and putting everything together, including proofing.

Nate: User views.

Joseph: Scenarios and sequence diagrams.

Matthew: Object model diagram and relational database schema.

Responsibilities (Phase I Revisions):
Vicky: Update table of contents, proof program statement and requirements definition, update ER model, update data dictionary, update use case diagram and use case descriptions as necessary, and overall finishing of the project.

Nate: Update user views forms and reports.

Joseph: Update and proof the scenarios and event trace diagrams.

Mathew: Update the object model diagram and relational database schema.
Responsibilities (Phase II - Implementation):

Vicky: Create tables in database, add data to database, update entity relationship model, update object model, update schema, update data dictionary, update use case diagram and statements, normalization diagrams, implementation narrative, features summary table, menu hierarchy, table definitions and data contents, testing, final remarks, some form design, and overall organization of final document.

Nate: Create tables in database, add data to database, update/add forms and reports, implementation narrative, features summary table, menu hierarchy, table definitions and data contents, screen-shots of web forms and reports, code for website and documentation for it, installation guide, testing, and final remarks.

Joseph: Create tables in database, add data to database, review problem statement/requirements definition, update scenarios and event trace diagrams and add form/report/use case numbers to them, normalization diagrams, implementation narrative, features summary table, testing and testing narrative, and final remarks.

Matthew: Create tables in database, add data to database, PowerPoint presentation, general proofreading of entire document, implementation narrative, testing, and final remarks.
Problem Statement & Requirements Definition

We decided to create a database that can help organize a library. It is useful for any type of library, but would work best for an extensive lending library with various types of media including books, magazines, journals, videos, books on tape and more. It would be ideal for a library upgrading from a card or paper system where they have a card for each piece of media and for each customer or patron. It would also benefit a library where all employee records are still on a paper basis as well including all hire paperwork and payroll information such as salary. As you can imagine for a paper system, it makes figuring out how many books are overdue, or how many people owe fees, take quite a while to figure out for the employees. Patrons also have no control over the system. The only thing they can do is use the card catalog to find media.

The purpose of this database is to automate and replace the current card and paper system. All tasks previously recorded on paper or cards will be integrated into the new system. The most important change will be to the checkout process. Theoretically, computers with barcode scanners will be available to the librarians or employees (we use the term librarian and employee interchangeably). The librarians will be able to issue new library cards that come with a unique barcode and number for easy checkout to patrons. All media would be receiving a barcode with their individual unique identifying number as well. With the barcodes, the librarians will simply be able to scan a membership card and the media's barcode to check it out to the patron, thereby eliminating the need for cards. Since we do not have scanners to use, for this project, we will be hand entering library card numbers and barcode numbers. The database will be able to keep track of who has what checked out. The ability to keep track of that information will also provide a variety of new reporting capabilities. For example, based on due dates, librarians can run reports to see who has late books (checked out media report), who owes fees for late books or damaged books (cost report) and much more. It will only take a few seconds to run the report as opposed to going through all of the cards by hand, saving the librarians hours a week.

The library patrons will also have added usability. Potentially there could be computers in the library for their use to see what books they currently have checked out, and what they have checked out in the past. The information they could view would be limited, as in they would have a different set of permissions than the employees. They would need their unique identification number to use the system. They would also be able to search the library's media by author, title, and subject. Because the system is linked to the one that the librarians will be using, they will also be able to see if the media is currently available for checkout. The last main feature that would be new to patrons is that they would have web access to the library to
reserve books. It will be a simple form with the same search capabilities that allows them to submit a request. After the request, the librarians could then pull the books and place them on hold on a first come first served basis. Other capabilities include being able to renew a book online. Each book can only be renewed once before it is considered to be late.

Lastly, the library manager will have the same access as the librarian as well as some added features. Employee records will all be stored in the database, replacing the paper system. He will also be able to order new media (not an actual function of the database) and add it to the system. He will then have more reporting capabilities including a usage report that shows how often something is used to help determine what types of media should be ordered and in what quantities.

Hopefully with all of the changes, it would make the library run much more smoothly. With less time spent filling out paperwork, more time can be devoted to serving the customers and maintaining the potentially vast amounts of media held in the library. The system will hopefully also be more reliable and user friendly to everyone. While it will require some training for employees, once the system is in place, the benefits will greatly outweigh the costs of implementing the system since using the computer to do most daily tasks will be much faster than a paper based system.
Conceptual Models: Entity Relationship Model
Use-Case Diagram
Use-Cases

Use case 1: Log in to system
Actor(s): Patron, Librarian, Library Manager
   Description: First screen anyone sees is the logon screen. At this screen they will enter their username and password and click "Login". After the system has verified the user they will be taken to the main menu screen appropriate to their access level.

Use case 2: Add a Patron
Actor(s): Librarian, Library Manager
   Description: Employee will be prompted with the menu screen. After choosing “Add a Patron”, they will enter in the first and last name, address, city, state, zip code, and phone number. When the user clicks Generate the patron will be given a login, password, and a Library Card Number. After clicking create they will be entered into the database.

Use case 3: Update User Information
Actor(s): Patron, Librarian, Library Manager
   Description: They will be prompted with the menu screen. After choosing “Administration/Patron”, they will search for the patron by the Patron's Library Card number or name. From here they can view patron details. In the popup they will then enter in the new changes for the customer. After clicking button “update”, the patron's information will be updated.

Use case 4: Reserve Media
Actor(s): Patron, Librarian, Library Manager
   Description: In home the user will search for the book. When they click view details a popup window will open allowing them to view information about the book. From here the user clicks “reserve”. If the user is a patron it will insert the reservation into the database and close the popup window. Otherwise the user will select the library card number from a list and then click “reserve”.

Use case 5: Check out Media
Actor(s): Librarian, Library Manager
   Description: Librarian or Library Manager on the home page will search for the media by title, author, or barcode. From here they click “view details” this will show relevant information about the media. At the bottom of the popup the user can click “check out”. This redirects the user to the checkout screen where they select the checkout date. To be returned date is automatically calculated based on the policies table at this point. The user then clicks checkout and the media is now checked out.
Use case 6: Check in Media
Actor(s): Librarian, Library Manager
   Description: Employee at home searches for the media to be checked in, clicks “view details” and then “check in”.

Use case 7: Renew Media
Actor(s): Librarian, Library Manager
   Description: Employee will search for media on the homepage. From here the employee can click “View Details” which will pop open details about the media. Then the user clicks “Check In”. On the next page the user will select a check-in date and select “yes” under the renewed radio box. When the user clicks “check in” the next to be returned date will be calculated and the book will be renewed.

Use case 8: Search for Media
Actor(s): Patron, Librarian, Library Manager
   Description: Employees and Patron home allows for searching by title, author, ISBN, and media type. From here the user can click “View Details” and view the media information.

Use case 9: Remove Damaged Media
Actor(s): Library Manager
   Description: Library Manager will select from the menu Administration/Media. Then the user will search for the media. Then the user can click delete which will remove the damaged media.

Use case 10: Make a Payment
Actor(s): Librarian, Library Manager
   Description: Employee will search for media/view details and click “check in”. If there is a fee calculated the screen will display a drop down with payment type and payment amount and check-in date. After the employee enters the payment amount they can click “Pay Fine” and the media will be checked in and paid for.

Use case 11: Add Media
Actor(s): Library Manager, Librarians
   Description: Librarian or Library Manager accesses the area from Administration/Media they then click “Add New Media”. The media details will popup allowing the user to enter Title, Barcode, Year, ISBN, Subject, Publisher, Initial Condition, Media Type, Order Cost, and the First Author. Once they click create the media will be entered into the database.

Use case 12: Run Overdue Media Report
Actor(s): Librarian, Library Manager
   Description: User will access report by going to Reports/ Overdue Media Report. From here the User clicks “Export” and the report will be generated in a PDF form to the user.
Use case 13: Add Author
Actor(s): Librarian, Library Manager
Description: Librarian or Library Manager will be accessed from Administration/Author. After choosing “Add Author”, they will enter the Authors first and last names and date of birth.

Use case 14: Add Employee
Actor(s): Library Manager
Description: Library Manager will access Administration/Employee. After choosing “Add Employee”, they will enter the new Employee’s first and last names, address, phone, and other contact information. The user will click “Generate” and the Employee ID, username, password. Finally they click “Create” and the employee will be entered into the database.

Use case 15: Update Employee
Actor(s): Library Manager
Description: Library Manager will access Administration/Employee. After searching for the Employee they will click “View Details”. From here the user can change Employee information and click “Update”.

Use case 16: Run Fee Report
Actor(s): Librarian, Library Manager
Description: User will access report by going to Reports/ Fee Report. From here the User clicks “Export” and the report will be generated in a PDF form to the user.

Use case 17: Run Media Usage Report
Actor(s): Library Manager
Description: User will access report by going to Reports/ Run Media Usage Report. From here the User clicks “Export” and the report will be generated in a PDF form to the user.

Use case 18: Run Checked out Media Report
Actor(s): Librarian, Library Manager
Description: User will access report by going to Reports/ Run Checked Media Report. From here the User clicks “Export” and the report will be generated in a PDF form to the user.
Logical Models:
Relational Database Schema
Normalization Diagrams

Table Title: Employee
State: Third Normal Form
Reason:

- No multi-valued attributes. Intersection of each row and column has only one value.
- No partial dependencies. One unique attribute to identify each tuple and no redundancies.
- No transitive dependencies. No non-primary-key attribute is transiively dependent on the primary key.
Table Title: Patron
State: Third Normal Form
Reason:
• No multi-valued attributes. Intersection of each row and column has only one value.
• No partial dependencies. One unique attribute to identify each tuple and no redundancies.
• No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.
Table Title: Media
State: Third Normal Form
Reason:
• No multi-valued attributes. Intersection of each row and column has only one value.
• No partial dependencies. One unique attribute to identify each tuple and no redundancies.
• No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.
Table Title: Author
State: Third Normal Form
Reason:
• No multi-valued attributes. Intersection of each row and column has only one value.
• No partial dependencies. One unique attribute to identify each tuple and no redundancies.
• No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.
Table Title: Condition
State: Third Normal Form
Reason:
• No multi-valued attributes. Intersection of each row and column has only one value.
• No partial dependencies. One unique attribute to identify each tuple and no redundancies.
• No transitive dependencies. No non-primary-key attribute is transitive dependent on the primary key.
Table Title: Check Out  
State: Third Normal Form  
Reason:  
• No multi-valued attributes. Intersection of each row and column has only one value.  
• No partial dependencies. One unique attribute to identify each tuple and no redundancies. 
• No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.
Table Title: Reservation
State: Third Normal Form
Reason:
• No multi-valued attributes. Intersection of each row and column has only one value.
• No partial dependencies. One unique attribute to identify each tuple and no redundancies.
• No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.
**Table Title:** Media-Author  
**State:** Third Normal Form  
**Reason:**  
- No multi-valued attributes. Intersection of each row and column has only one value.  
- No partial dependencies. One unique attribute to identify each tuple and no redundancies.  
- No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.
Table Title: Policies
State: Third Normal Form
Reason:
• No multi-valued attributes. Intersection of each row and column has only one value.
• No partial dependencies. One unique attribute to identify each tuple and no redundancies.
• No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.
Scenarios with Event Trace Diagrams

**Scenario for Log On**
-A user wishes to access the database
-All users will be prompted with this event

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter Username &amp; Password</td>
<td>&lt;--------&gt;</td>
<td>Validate access</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;--------&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Scenario for Add a Patron**
-The Librarian or Library Manager enters the customer’s first and last name, address, city, state, zip code, and phone number
-The system then generates a unique number to be used as a library card number

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Patron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add a Patron</td>
<td>&lt;--------&gt;</td>
<td>Library Card # Generated</td>
</tr>
<tr>
<td>Enter Customer’s Information</td>
<td>&lt;--------&gt;</td>
<td>&lt;--------&gt;</td>
</tr>
<tr>
<td>Choose Create Patron</td>
<td>&lt;--------&gt;</td>
<td>Library Card # Displayed</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;--------&gt;</td>
</tr>
</tbody>
</table>
**Scenario for Update User Information**
-A Librarian or Library Manager has moved or changed phone number and needs to update contact information
-The Patron can change information from their account
-The Patron can also go into the library and show a librarian or librarian manager their card
-The card is scanned in or entered in manually
-After the information is entered it is displayed so that it can be verified

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Patron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Contact Information</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>Edit Customer’s Information</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>Update User Contacts</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Patent Information changed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Patent Information Displayed</td>
</tr>
</tbody>
</table>

**Scenario for Reserve Media**
-A Patron wishes to reserve an item they log in or call a librarian
-A librarian is contacted by a patron to reserve an item
-A link is generated from a successful search for media
-Patron ID is entered

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Media Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter Search Parameter</td>
<td>---------------</td>
<td>Media item status changed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Media item status Displayed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Scenario for Check out Media**
- A Patron has an item they wish to checkout
- A Patron comes in for a reserved book
- A Librarian searches for the Media by ISBN, title, or Author
- If the Patron's information is correct the Librarian selects check out
- A due date is generated from the check out date
- The media's status is changed to checked out
- The due date is displayed

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Media Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Searches For Media</td>
<td>&lt;---------&gt;</td>
<td></td>
</tr>
<tr>
<td>Enter Checkout Date</td>
<td>&lt;---------&gt;</td>
<td>Patron Information Displayed &lt;--------&gt;</td>
</tr>
<tr>
<td>Chose Checkout</td>
<td>&lt;---------&gt;</td>
<td>Media status is changed &lt;--------&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Media due date is generated &lt;--------&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Media status is displayed &lt;--------&gt;</td>
</tr>
</tbody>
</table>
**Scenario for Check in Media**

- A Patron has an item they wish to check in
- A media is dropped off in the media return
- A Librarian scans the media's barcode
- A Librarian can view the media's information and has the option to update book's condition status
- If the media's status is unchanged the Librarian selects check in
- If the media's status needs to be updated the Librarian selected the current condition and check in
- If the media's condition is changed a fine is generated and placed in the Patron's account
- If the due date was superseded a fine is generated and placed in the Patron's account

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Media Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scans media item</td>
<td>Media description &amp; condition Displayed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------</td>
<td></td>
</tr>
<tr>
<td>Verifies the media's condition</td>
<td>Media status changed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------</td>
<td></td>
</tr>
<tr>
<td>If ok Checks in the media</td>
<td>Media condition is updated</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------</td>
<td></td>
</tr>
<tr>
<td>If not ok updates condition code</td>
<td>Fines assessed against Patron</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Media status changed</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;----------</td>
<td></td>
</tr>
</tbody>
</table>
**Scenario for Renew Media**
- A Patron wishes to extend their borrowing of an item
- A Librarian will enter the renew menu screen
- A Librarian will scan in the Patron’s ID or scan in the barcode of the item
- If there is no reservation hold the Librarian will have the option of renew now

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Media Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barcode scanned</td>
<td>Reservation status displayed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If not reserved media is renewed</td>
<td>Media due date is generated</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Media status is displayed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If reserved inform Patron</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Scenario for Search for Media**
- A Patron wishes to borrow an item but is unsure if it is carried
- A Patron wishes to know more about a subject
- A Patron wishes to read from a certain Author
- A Patron is unsure of the exact title
- A Patron asks a Librarian the above
- A user then enters the search terms into the search menu
- A list of matches is generated and displayed to the user
- The user then can select to reserve a specific item

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Media Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Criteria Entered</td>
<td>Media information displayed</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User selects a media item</td>
<td>Media information linked to reserve media</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scenario for Remove Damaged Media
- Media item is too damaged to maintain in the inventory
- Librarian Manager decides to remove media item
- Librarian Manager will choose remove media from the menu
- The media is removed
- The Librarian removes the media item

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Media Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search Media</td>
<td>----------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete Media Now</td>
<td>----------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Media Removed From DB</td>
<td>&lt;--------</td>
</tr>
</tbody>
</table>

Scenario for Make a Payment
- A Patron has incurred a fine for either damage to an item or a late fee
- The Librarian or Library Manager searches for the media that is overdue.
- The Librarian or Library Manager selects the popup and enters in the payment type.
- The Librarian or Library Manager then inputs the amount to be paid
- The Patron's overdue and damaged media status are updated

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Patron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media Is Selected</td>
<td>----------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Select Checkin</td>
<td>----------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The fine due is selected</td>
<td>----------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The fine amount is input</td>
<td>----------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pay now is selected</td>
<td>----------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patron fine status changed</td>
<td>&lt;--------</td>
<td></td>
</tr>
<tr>
<td>Patron fine status Displayed</td>
<td>&lt;--------</td>
<td></td>
</tr>
</tbody>
</table>
Scenario for Add Media
- Media is requested to be added to the inventory
- Librarian Manager decides to carry certain media item
- Librarian will choose add media from the menu
- The media information is added
- The Librarian adds the media item
- Barcode is generated

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Media Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add New Media</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>Enter’s Media Information</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>Create Media</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Media is added to database</td>
<td>&lt;---------------</td>
</tr>
<tr>
<td></td>
<td>Media Information Displayed</td>
<td>&lt;---------------</td>
</tr>
<tr>
<td></td>
<td>Barcode is generated</td>
<td>&lt;---------------</td>
</tr>
</tbody>
</table>

Scenario for Run Overdue Media Report
- A report is scheduled to be run over a certain time period
- A Librarian wishes to spot check with this report
- A superior wishes to have a copy of the current status

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Media Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reports is selected</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>Click Export</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>Hard Copy is display</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If hardcopy opens as PDF</td>
<td>&lt;---------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scenario for Add Author
-An author is requested to be added to the inventory
-Librarian Manager decides to carry media from a certain Author
-Librarian will choose add Author from the menu
-The Author is added

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Author</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;--------------</td>
<td></td>
</tr>
<tr>
<td>Enter's Author's Information</td>
<td>&lt;------------</td>
<td></td>
</tr>
<tr>
<td>Add Author Now</td>
<td>&lt;--------------</td>
<td>Author is added to database</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Author Information Displayed</td>
</tr>
</tbody>
</table>

Scenario for Add Employee
-An employee is hired

<table>
<thead>
<tr>
<th>Librarian Manager</th>
<th>User Interface</th>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Employee</td>
<td>---------------</td>
<td>Employee Information Added</td>
</tr>
<tr>
<td></td>
<td>&lt;--------------</td>
<td>Employee Information Displayed</td>
</tr>
<tr>
<td>Enter Employee's Information</td>
<td>&lt;------------</td>
<td></td>
</tr>
<tr>
<td>Add Employee Now</td>
<td>&lt;--------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Scenario for Update Employee Role
-An employee is promoted
-An employee is demoted
-An employee makes a lateral move in status

<table>
<thead>
<tr>
<th>Librarian Manager</th>
<th>User Interface</th>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change Employee role</td>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>Edit Employee role</td>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>Change Employee role Now</td>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>Employee role changed</td>
<td>-------</td>
</tr>
<tr>
<td></td>
<td>Employee role Displayed</td>
<td>-------</td>
</tr>
</tbody>
</table>

Scenario for Run Fee Report
-A report is scheduled to be run over a certain time period
-A Librarian wishes to spot check with this report
-A superior wishes to have a copy of the current status

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Patron</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reports is selected</td>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>Cost report is selected</td>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>Hard Copy or display is selected</td>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td>If hardcopy opens as PDF</td>
<td>--------------</td>
<td>-------</td>
</tr>
</tbody>
</table>
**Scenario for Run Media Usage Report**

- A report is scheduled to be run over a certain time period
- A Librarian wishes to spot check with this report
- A superior wishes to have a copy of the current status

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Media Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reports is selected</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-----------------</td>
<td>------------</td>
</tr>
<tr>
<td>Media Usage is selected</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-----------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>If hardcopy opens as PDF</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;-------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Scenario for Run Checked Out Media Report**

- A report is scheduled to be run over a certain time period
- A Librarian wishes to spot check with this report
- A superior wishes to have a copy of the current status

<table>
<thead>
<tr>
<th>User</th>
<th>User Interface</th>
<th>Media Item</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reports is selected</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-----------------</td>
<td>------------</td>
</tr>
<tr>
<td>Checked Out is selected</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-----------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>&lt;-------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
If hardcopy opens as PDF

<--------------
Data Dictionary
Physical Models:
Implementation Narrative

* describe platform/tools choices - justification, issues and limitations.

1. .net
   a. Justifications
      i. Previous Experience
      ii. Powerful framework.
      iii. Majority of needed libraries are included in the standard framework.
      iv. Easy to deploy to other servers.
   b. Issues
      i. Harder to develop on non-windows based machines.
   c. Limitations
      i. Will not run natively on UNIX based systems.

2. Visual Studio
   a. Justifications
      i. Official Microsoft tools for developing in C# and .net
   b. Issues
      i. Projects needed to be placed in a repository otherwise changing the location of the project causes issues.
   c. Limitations
      i. Will not run on UNIX based systems.

3. SQL Server 2008
   a. Justifications
      i. Previous experience
      ii. Compatible with visual studio 2010 and supports LINQ to SQL generation.
   b. Issues
      i. Connection string needs to be changed when the project is deployed.
   c. Limitations
      i. Will not run on UNIX based systems.
      ii. Does not contain equal join.

4. LINQ to SQL
   a. Justifications
      i. Very Easy to convert SQL tables directly to objects.
ii. Inserts/Updates/Deletes are handled back end.
iii. Only executes sql commands when it detects changes have been made.

b. Issues
i. None

c. Limitations
i. Microsoft Only Product
ii. The generated objects base is an Entity class which makes it impossible to store the actual object in session

5. AJAX
   a. Justifications
      i. Helps to link the client side Java script with the Server side C#
   b. Issues
      i. If users have JavaScript disabled this will not work.
   c. Limitations
      i. Not all old browsers fully support this technology.

6. C#
   a. Justifications
      i. Previous experience
      ii. Is built upon C++ and the ideas of object oriented programming.
   b. Issues
      i. None
   c. Limitations
      i. Microsoft product

* describe problem scoping choices - justification, issues and limitations.

* describe modifications to prior phases - justification, issues and limitations

* clearly indicate successes, failures, and status of project (which features work, which don't, etc.)

1. Reports export locally but due to hosting our physical website does not allow the exporting at this time.
Features Summary Table
Menu Hierarchy
Deployment

- describe installation process and usage (reference appendices)

1. The Deployment process is a onetime setup run by either our staff or the sever admin at the place of business where this software will be deployed.
2. Updates for bug fixes will be applied with an update script that can either be sent via email or applied directly by our staff.

Testing

Reference appendices
Final Remarks

Any additional detail about your project that has not been addressed in prior sections (special features or implementations, problems, ideas for future enhancement, etc.).

1. We store derived attributes because if we change the numbers we use to drive, such as CheckOutLength, we lose the history for calculations.
   a. If we changed CheckOutLength from 14 days to 21
Appendices:
Table Definitions and Data Contents

Author

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fname</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
<tr>
<td>Minit</td>
<td>nchar(1)</td>
<td></td>
</tr>
<tr>
<td>Lname</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
<tr>
<td>DOB</td>
<td>date</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fname</th>
<th>Minit</th>
<th>Lname</th>
<th>DOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adams</td>
<td>Q</td>
<td>Douglas</td>
<td>1967-12-05</td>
</tr>
<tr>
<td>Asimov</td>
<td>Z</td>
<td>Isaac</td>
<td>1975-12-30</td>
</tr>
<tr>
<td>Card</td>
<td>S</td>
<td>Orson</td>
<td>1942-07-23</td>
</tr>
<tr>
<td>Firstname</td>
<td>M</td>
<td>Lastname</td>
<td>1960-04-12</td>
</tr>
<tr>
<td>George</td>
<td>J</td>
<td>Orwell</td>
<td>1930-06-05</td>
</tr>
<tr>
<td>Ira</td>
<td>T</td>
<td>Berkowitz</td>
<td>1954-04-12</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>Tolkien</td>
<td>1980-03-12</td>
</tr>
<tr>
<td>Nate</td>
<td>H</td>
<td>Piddy</td>
<td>1988-03-12</td>
</tr>
</tbody>
</table>

Check_Out

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>LibraryCardNumber</td>
<td>nvarchar(9)</td>
<td></td>
</tr>
<tr>
<td>BarCode</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>CheckOutDate</td>
<td>datetime</td>
<td></td>
</tr>
<tr>
<td>Renewed</td>
<td>bit</td>
<td></td>
</tr>
<tr>
<td>AmountPaid</td>
<td>money</td>
<td></td>
</tr>
<tr>
<td>Fee</td>
<td>money</td>
<td></td>
</tr>
<tr>
<td>ToBeReturnedDate</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>Checker#</td>
<td>nvarchar(9)</td>
<td></td>
</tr>
<tr>
<td>CheckInDate</td>
<td>datetime</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LibraryCardNumber</th>
<th>BarCode</th>
<th>CheckOutDate</th>
<th>Renewed</th>
<th>AmountPaid</th>
<th>Fee</th>
<th>ToBeReturnedDate</th>
<th>Checker#</th>
<th>CheckInDate</th>
<th>CurrentCondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10000001</td>
<td>2010-04-14 00:00:00.000</td>
<td>0</td>
<td>NULL</td>
<td>NULL</td>
<td>2010-05-08 00:00:00.000</td>
<td>NULL</td>
<td>NULL</td>
<td>Good</td>
</tr>
<tr>
<td>2</td>
<td>10000001</td>
<td>2010-04-20 00:00:00.000</td>
<td>0</td>
<td>NULL</td>
<td>NULL</td>
<td>2010-04-30 00:00:00.000</td>
<td>NULL</td>
<td>NULL</td>
<td>New-Good</td>
</tr>
<tr>
<td>3</td>
<td>10000001</td>
<td>2010-05-04 00:00:00.000</td>
<td>0</td>
<td>NULL</td>
<td>NULL</td>
<td>2010-05-18 00:00:00.000</td>
<td>NULL</td>
<td>NULL</td>
<td>Good</td>
</tr>
<tr>
<td>4</td>
<td>10000001</td>
<td>2010-11-20 00:00:00.000</td>
<td>0</td>
<td>NULL</td>
<td>NULL</td>
<td>2005-12-13 00:00:00.000</td>
<td>NULL</td>
<td>NULL</td>
<td>New</td>
</tr>
<tr>
<td>5</td>
<td>10000005</td>
<td>2010-04-01 00:00:00.000</td>
<td>0</td>
<td>5.50</td>
<td>NULL</td>
<td>2010-04-15 00:00:00.000</td>
<td>NULL</td>
<td>NULL</td>
<td>Good-Fin</td>
</tr>
<tr>
<td>6</td>
<td>10000001</td>
<td>2009-08-07 08:00:00.000</td>
<td>0</td>
<td>NULL</td>
<td>NULL</td>
<td>2010-04-21 00:00:00.000</td>
<td>NULL</td>
<td>NULL</td>
<td>New</td>
</tr>
<tr>
<td>7</td>
<td>10000001</td>
<td>2009-10-20 00:00:00.000</td>
<td>0</td>
<td>NULL</td>
<td>NULL</td>
<td>2009-12-22 00:00:00.000</td>
<td>NULL</td>
<td>NULL</td>
<td>New-Good</td>
</tr>
<tr>
<td>8</td>
<td>10000006</td>
<td>2010-04-06 00:00:00.000</td>
<td>0</td>
<td>NULL</td>
<td>NULL</td>
<td>2010-05-14 00:00:00.000</td>
<td>NULL</td>
<td>NULL</td>
<td>New-Good</td>
</tr>
<tr>
<td>9</td>
<td>10000006</td>
<td>2010-04-21 00:00:00.000</td>
<td>1</td>
<td>NULL</td>
<td>NULL</td>
<td>2010-04-30 00:00:00.000</td>
<td>NULL</td>
<td>NULL</td>
<td>New</td>
</tr>
<tr>
<td>10</td>
<td>10000007</td>
<td>2010-05-05 00:00:00.000</td>
<td>0</td>
<td>NULL</td>
<td>NULL</td>
<td>2010-05-13 00:00:00.000</td>
<td>NULL</td>
<td>NULL</td>
<td>New</td>
</tr>
</tbody>
</table>
### Condition

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grade</td>
<td>nvarchar(8)</td>
<td></td>
</tr>
<tr>
<td>Cost_Percentage</td>
<td>int</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grade</th>
<th>Cost_Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td>80</td>
</tr>
<tr>
<td>Good-Poo</td>
<td>70</td>
</tr>
<tr>
<td>New</td>
<td>100</td>
</tr>
<tr>
<td>New-Good</td>
<td>90</td>
</tr>
<tr>
<td>Poor</td>
<td>60</td>
</tr>
<tr>
<td>Replace</td>
<td>50</td>
</tr>
</tbody>
</table>

### Employee

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>EmpID</td>
<td>nvarchar(9)</td>
<td></td>
</tr>
<tr>
<td>SSN</td>
<td>nchar(9)</td>
<td></td>
</tr>
<tr>
<td>FName</td>
<td>nvarchar(20)</td>
<td></td>
</tr>
<tr>
<td>mName</td>
<td>nvarchar(20)</td>
<td></td>
</tr>
<tr>
<td>lName</td>
<td>nvarchar(20)</td>
<td></td>
</tr>
<tr>
<td>payRate</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
<tr>
<td>DOB</td>
<td>datetime</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>nvarchar(MAX)</td>
<td></td>
</tr>
<tr>
<td>phoneNumber</td>
<td>nvarchar(15)</td>
<td></td>
</tr>
<tr>
<td>roleType</td>
<td>nvarchar(20)</td>
<td></td>
</tr>
<tr>
<td>Eusername</td>
<td>nvarchar(20)</td>
<td></td>
</tr>
<tr>
<td>Epassword</td>
<td>nvarchar(20)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EmpID</th>
<th>SSN</th>
<th>FName</th>
<th>mName</th>
<th>lName</th>
<th>payRate</th>
<th>DOB</th>
<th>Address</th>
<th>phoneNumber</th>
<th>roleType</th>
<th>Eusername</th>
<th>Epassword</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0001276425</td>
<td>Isabel</td>
<td>Collie</td>
<td>Emily</td>
<td>1995</td>
<td>1970-01-15</td>
<td>00:00:00:00:00:00 5027 East Ave. Chicago, CA 60628</td>
<td>57009801234</td>
<td>Librarian</td>
<td>1_Body</td>
<td>27360</td>
</tr>
<tr>
<td>2</td>
<td>0025434173</td>
<td>Amy</td>
<td>Holton</td>
<td>Swanson</td>
<td>8500</td>
<td>1960-04-18</td>
<td>00:00:00:00:00:00 3201 Race St. Chicago, CA 60626</td>
<td>41232489047</td>
<td>Librarian</td>
<td>1_Swanson</td>
<td>273232</td>
</tr>
<tr>
<td>3</td>
<td>0038232818</td>
<td>Louis</td>
<td>Greene</td>
<td>McDaniel</td>
<td>5000</td>
<td>1965-09-19</td>
<td>00:00:00:00:00:00 3306 Burnell Rd. Chicago, CA 60603</td>
<td>500052341981</td>
<td>Librarian</td>
<td>1_McDaniel</td>
<td>27323</td>
</tr>
<tr>
<td>4</td>
<td>0070728303</td>
<td>Leon</td>
<td>Timothy</td>
<td>Blake</td>
<td>1200</td>
<td>1986-10-17</td>
<td>00:00:00:00:00:00 Apartment 1723 Monroe Avenue, Chicago, CA 60613</td>
<td>51003429982</td>
<td>Librarian</td>
<td>1_Blake</td>
<td>273272</td>
</tr>
<tr>
<td>5</td>
<td>0080217676</td>
<td>Brady</td>
<td>Jody</td>
<td>Stevens</td>
<td>8000</td>
<td>1989-10-02</td>
<td>00:00:00:00:00:00 3402 Knights Rd. Chicago, CA 60606</td>
<td>50006897034</td>
<td>Librarian</td>
<td>1_Stevens</td>
<td>273260</td>
</tr>
<tr>
<td>6</td>
<td>0090349451</td>
<td>Maxwell</td>
<td>Benjamin</td>
<td>Justin</td>
<td>1020</td>
<td>1995-04-05</td>
<td>00:00:00:00:00:00 203-06 Hyde Street, Paradise, CA 95067</td>
<td>50003450214</td>
<td>Librarian</td>
<td>1_Justin</td>
<td>273200</td>
</tr>
<tr>
<td>7</td>
<td>0100401643</td>
<td>Whitney</td>
<td>Allie</td>
<td>Riley</td>
<td>9000</td>
<td>1987-07-09</td>
<td>00:00:00:00:00:00 Appartment 17235450 Libero Avenue, Chicago, CA 60628</td>
<td>67502020526</td>
<td>Librarian</td>
<td>1_Riley</td>
<td>273243</td>
</tr>
</tbody>
</table>
### Media

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barcode</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td>nvarchar(100)</td>
<td></td>
</tr>
<tr>
<td>ISBN</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td>datetime</td>
<td></td>
</tr>
<tr>
<td>Publisher</td>
<td>nvarchar(100)</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
<tr>
<td>Subject</td>
<td>nvarchar(100)</td>
<td></td>
</tr>
<tr>
<td>OrderCost</td>
<td>int</td>
<td></td>
</tr>
<tr>
<td>InitialCondition</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
<tr>
<td>Grade</td>
<td>nvarchar(8)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Barcode</th>
<th>Title</th>
<th>ISBN</th>
<th>Year</th>
<th>Publisher</th>
<th>Type</th>
<th>Subject</th>
<th>OrderCost</th>
<th>InitialCondition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Animal Farm: Nineteen Eighty-Four</td>
<td>184046245</td>
<td>2000-01-01 03:00:00.000</td>
<td>Palgrave macmillan</td>
<td>DVD</td>
<td>Palgrave macmillan</td>
<td>27</td>
<td>New</td>
</tr>
<tr>
<td>2</td>
<td>Big Fish</td>
<td>200993345</td>
<td>2016-04-05 06:00:00.000</td>
<td>Harcourt</td>
<td>Book</td>
<td>Fish</td>
<td>25</td>
<td>New</td>
</tr>
<tr>
<td>3</td>
<td>Ender’s Game</td>
<td>012550706</td>
<td>1984-01-01 00:00:00.000</td>
<td>Tor Science Fiction</td>
<td>Book</td>
<td>Space</td>
<td>8</td>
<td>New</td>
</tr>
<tr>
<td>4</td>
<td>Foundation</td>
<td>0352832564</td>
<td>1981-01-01 00:00:00.000</td>
<td>Spectra</td>
<td>Book</td>
<td>Foundation</td>
<td>8</td>
<td>New</td>
</tr>
<tr>
<td>5</td>
<td>Family Matters</td>
<td>3021221248</td>
<td>2006-01-01 00:00:00.000</td>
<td>Justin Charles &amp; Co</td>
<td>Book</td>
<td>Family</td>
<td>16</td>
<td>New</td>
</tr>
<tr>
<td>6</td>
<td>Restaurant at the End of the Universe</td>
<td>315233910</td>
<td>1955-01-01 00:00:00.000</td>
<td>Ballantine Publishing Group</td>
<td>Book</td>
<td>Hitchcock</td>
<td>0</td>
<td>New</td>
</tr>
</tbody>
</table>

### Media_Author

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barcode</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>AuthFName</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
<tr>
<td>AuthLName</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
<tr>
<td>AuthDOB</td>
<td>date</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Barcode</th>
<th>AuthFName</th>
<th>AuthLName</th>
<th>AuthDOB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Adams</td>
<td>Douglas</td>
<td>1967-12-05</td>
</tr>
<tr>
<td>2</td>
<td>George</td>
<td>Orwell</td>
<td>1930-06-05</td>
</tr>
<tr>
<td>3</td>
<td>J</td>
<td>Tolkien</td>
<td>1960-03-12</td>
</tr>
<tr>
<td>4</td>
<td>Card</td>
<td>Orson</td>
<td>1942-07-23</td>
</tr>
<tr>
<td>5</td>
<td>Asimov</td>
<td>Isaac</td>
<td>1975-12-30</td>
</tr>
<tr>
<td>6</td>
<td>Ira</td>
<td>Berkowitz</td>
<td>1954-04-12</td>
</tr>
<tr>
<td>7</td>
<td>Adams</td>
<td>Douglas</td>
<td>1967-12-05</td>
</tr>
</tbody>
</table>
### Patron

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>libraryCardNumber</td>
<td>nvarchar(9)</td>
<td></td>
</tr>
<tr>
<td>fName</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
<tr>
<td>mName</td>
<td>nvarchar(20)</td>
<td></td>
</tr>
<tr>
<td>lName</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
<tr>
<td>DOB</td>
<td>date</td>
<td></td>
</tr>
<tr>
<td>phoneNumber</td>
<td>nvarchar(20)</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>nvarchar(MAX)</td>
<td></td>
</tr>
<tr>
<td>Username</td>
<td>nvarchar(20)</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>nvarchar(20)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>libraryCardNumber</th>
<th>Name</th>
<th>mName</th>
<th>Address</th>
<th>DOB</th>
<th>phoneNumber</th>
<th>Username</th>
<th>Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000000001</td>
<td>Dan</td>
<td>Kevin</td>
<td>7301 E 10th St, Chico, CA 95926</td>
<td>1990-07-01</td>
<td>7071241241</td>
<td>Selkon</td>
<td>10328</td>
</tr>
<tr>
<td>1000000002</td>
<td>Amy</td>
<td>Kristen</td>
<td>3109 E 10th St, Chico, CA 95926</td>
<td>1990-04-18</td>
<td>8560490047</td>
<td>Swanson</td>
<td>34233</td>
</tr>
<tr>
<td>1000000003</td>
<td>Susan</td>
<td>Edson</td>
<td>5301 E 10th St, Chico, CA 95926</td>
<td>1975-12-02</td>
<td>530343245</td>
<td>Bradford</td>
<td>1325v</td>
</tr>
<tr>
<td>1000000004</td>
<td>Vicky</td>
<td>Marie</td>
<td>5302 E 10th St, Chico, CA 95926</td>
<td>1950-07-07</td>
<td>5305321066</td>
<td>Holcomb</td>
<td>1900V</td>
</tr>
<tr>
<td>1000000005</td>
<td>Matthew</td>
<td>NULL</td>
<td>7071717171</td>
<td>1940-01-01</td>
<td>7071717171</td>
<td>Harris</td>
<td>NULL</td>
</tr>
<tr>
<td>1000000006</td>
<td>Joseph</td>
<td>NULL</td>
<td>530101241</td>
<td>1930-01-01</td>
<td>5301012412</td>
<td>Fitzgerald</td>
<td>NULL</td>
</tr>
<tr>
<td>1000000007</td>
<td>Nate</td>
<td>NULL</td>
<td>530833712</td>
<td>1920-01-01</td>
<td>5302328712</td>
<td>Paddy</td>
<td>NULL</td>
</tr>
</tbody>
</table>

### Policies

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>nvarchar(50)</td>
<td></td>
</tr>
<tr>
<td>Value</td>
<td>decimal(18, 2)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CheckOutLength</td>
<td>14.00</td>
</tr>
<tr>
<td>FeePerDay</td>
<td>0.25</td>
</tr>
</tbody>
</table>

### Reservation

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Data Type</th>
<th>Allow Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>libraryCardNumber</td>
<td>nvarchar(9)</td>
<td></td>
</tr>
<tr>
<td>BarCode</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>DateRequested</td>
<td>datetime</td>
<td></td>
</tr>
<tr>
<td>CheckOutStatus</td>
<td>bit</td>
<td></td>
</tr>
<tr>
<td>ReshelfDate</td>
<td>datetime</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>libraryCardNumber</th>
<th>BarCode</th>
<th>DateRequested</th>
<th>CheckOutStatus</th>
<th>ReshelfDate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000000001</td>
<td>54879624</td>
<td>2010-03-12 00:00:00</td>
<td>True</td>
<td>2010-03-16 00:00:00</td>
</tr>
<tr>
<td>1000000001</td>
<td>54879624</td>
<td>2010-03-12 00:00:00</td>
<td>False</td>
<td>2010-03-16 00:00:00</td>
</tr>
</tbody>
</table>
User Views - Forms

*User signifies a person of any role

From #1
#1 Use case: Login
Actor(s): Patron, Librarian, Library Manager
    Description: The employee or patron enters their username and password we
differentiate between patron and employee entities. Patron’s user name is formatted as the
first initial of the first name and their last name, for example npreddy. An employee has a
number for the first part and their last name as the second part, for example 1_john. In the
case that there are multiple people with the same last name it will increase the number by
one till there is no longer a duplicate. The library manager has a distinct type as well, since
she has some different capabilities than normal employees. The format for library manager is
all of the first name with the first initial of the last name, for example VickyH. Once this has
been determined the user is directed to their correct homepage.

Items Hidden/Changed for Patrons:
    1. (None)
Form #2

#2, 3 Use case: Add Patron/Update User Information

Actor(s): Patron, Librarian, Library Manager

Description: Allows a librarian or the library manager to create a new patron. This will generate a library card number along with a user name and password. Patrons are allowed to access this from the menu item “User Information” and allowed to update their own information. They are not allowed to see anyone's information but their own. Librarians and the library manager can see everyone's information. This view

Items Hidden/Changed for Patrons:
1. Only Update and Close buttons will be visible to the patron.
2. Delete Patron button is for the library manager and librarian and is meant to be used rarely.

**PATRON DETAILS**

<table>
<thead>
<tr>
<th>User Name</th>
<th>Password</th>
<th>Library Card Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>gellison</td>
<td>13832</td>
<td>100000001</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Middle Name</th>
<th>Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quin</td>
<td>Ellison</td>
<td>Kevin</td>
<td>(707)-124-1241</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Address</th>
<th>DOB</th>
<th></th>
</tr>
</thead>
</table>

**Check out History**

<table>
<thead>
<tr>
<th>Card Number</th>
<th>Checked Out Date</th>
<th>Checked In Date</th>
<th>To Be Returned Date</th>
<th>Checked Out Employee</th>
<th>Amount Paid</th>
</tr>
</thead>
<tbody>
<tr>
<td>100000001</td>
<td>05/04/2010</td>
<td></td>
<td>05/13/2010</td>
<td>Isabella Brady</td>
<td></td>
</tr>
<tr>
<td>100000001</td>
<td>04/20/2010</td>
<td>05/20/2010</td>
<td>04/30/2010</td>
<td>Isabella Brady</td>
<td></td>
</tr>
<tr>
<td>100000001</td>
<td>04/14/2010</td>
<td>04/30/2010</td>
<td>05/06/2010</td>
<td>Isabella Brady</td>
<td></td>
</tr>
</tbody>
</table>

[Update] [Close]
Form #3
#3b Use case: View Patron Details
Actor(s): Librarian, Library Manager

Description: Details View shows the list of patrons in the library. This page allows you to remove media and open a new window to enter or change patron data. Here you have the option to search by Library Card Number or Name.

Items Hidden/Changed for Patrons:
1. View is hidden from patrons.
Form #4
#4 Use case: View Media Details / Renew
From use case(s) 7
Actor(s): Patron, Librarian, Library Manager

Description: Details View shows the list of media in the library. Here you have the option to renew media, check it in or out, reserve it, and pay fines.

Items Hidden/Changed for Patrons:
1. Check out History will only show history related to the currently logged in Patron.
2. Barcode will be hidden.
3. Order Cost will be hidden and drop downs will be turned into read only text boxes.
4. Condition

![Media Details: Animal Farm-Nineteen Eighty-Four](image)
Form #5

#5,6, 7 Use case: View Media Details

Actor(s): Librarian, Library Manager

Description: This view fulfills the use cases for checking in, checking out, and renewing media. If the media is not checked out the button will say “Check Out” rather than “Check In”. In all of the following cases the system will check if the user has any outstanding fees and will disable the ability to check out or renew media until the patron has paid the fee.

1. Case # 4 (Check Out Media) – Librarian, or Library Manager, selects the patrons Library Card Number from a drop down, fills in Check Out Date and clicks the “Check Out Button”
2. Case # 5 (Check in Media) - Librarian, or Library Manager, fills in the Checked in Date and Clicks on Check In.
3. Case # 6 (Renew Media) - Librarian, or Library Manager, selects renewed radio button fills in the checked in date and clicks Check In. If the renewed radio button is selected it will use the Checked in Date as the new checkout date and calculate the new due date.

Items Hidden/Changed for Patrons:

1. (None)
Form #6

#8 Use case: Search for Media

Actor(s): Patron, Librarian, Library Manager

Description: Homepage for all users. User may select to search by "Media type" and will have the option to search by "Title", "Author", or "Subject". After they click search the results will be displayed with the table. From here a user can view media details which are covered in other use cases.

Items Hidden/Changed for Patrons:
1. Checked Out will be specified by “yes” or “no” instead of a date.
2. Barcode will be hidden.
3. Patron Menu

![Library Database Management System](Image)
Form #7

#9 Use case: Search for Media / Administration
Actor(s): Library Manager
Description: This is the same as the homepage view however it allows the Library Manager to delete media. Otherwise the functionality is the same as the homepage view. Condition will be added to the details allowing the Librarian or Librarian Manager to delete the damaged media.

Items Hidden/Changed for Patrons:
1. View is hidden from Patrons.

![Library Database Management System](image-url)
Form #8
#9b Verify Delete
Description: Warning to user when they click delete media to verify that they didn’t accidentally click on the delete link.

Items Hidden/Changed for Patrons:
1. View is hidden from Patrons.

Form #9
#10 Use case: Make Payment
Actor(s): Patron, Librarian, Library Manager
Description: Initiated when a patron is required to make a payment. The Librarian or Library Manager Selects Credit Card, Check, Or Cash enters the amount the patron is paying and submits the payment. If the fee is fully paid then the patron can now check out media.

Items Hidden/Changed for Patrons:
1. (None)
Form #10
#11 Use case: Add Media
Actor(s): Librarian, Library Manager
Description: Enter in the appropriate data for the new media. User uses the drop down to select the author they wish to add and then clicking the “add” button to apply the author to this media.

Items Hidden/Changed for Patrons:
1. View is hidden from Patrons.

Form #11
#13 Use case: Add / Update Author
Actor(s): Librarian, Library Manager
Description: This view gives the Library Manager and the ability to add and update author information.

Items Hidden/Changed for Patrons/Employees:
This View is hidden from Patrons.
Form #12
#13b Use case: View Authors Details
Actor(s): Librarian, Library Manager

Description: This view gives the Library Manager and Librarian the list of Authors and their information. This will allow you to search by Authors name and open a details window to edit or add new Authors.

Items Hidden/Changed for Patrons/Employees:
1. This View is hidden from Patrons.
Form #13
#14 Use case: View Employee Details
From use case(s) 7
Actor(s): Library Manager

Description: This view gives the Library Manager the list of patrons and their information. It allows you to search by Library Card Number, Name. From this view you can view employee details in which you can add/update/remove the employee.

Items Hidden/Changed for Patrons/Employees:
1. This View is hidden from Both Employees and Patrons.
Form #14

#15 Use case: Add/Update/Remove Employee
Actor(s): Librarian, Library Manager

Description: Allows the Library Manager to create new employees. The user name and password will be generated from the employee’s first and last name. Patrons are allowed to access this from the menu item “User Information” and allowed to update their information only.

Items Hidden/Changed for Patrons/Employees:
1. Hidden from Patrons, Employees are only allowed to update their information.
2. Pay Rate is read only for Employees Library Manager can set the Pay Rate.
User Views - Reports

Report #1
#16 Use case: Free Report
Actor(s): Library Manager

Description: The Library Manager selects this from the report menu item and lists all the fees that are due.

<table>
<thead>
<tr>
<th>Title</th>
<th>Fee</th>
<th>LibraryCardNumber</th>
<th>To Be Returned Date</th>
<th>Phone Number</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigFish</td>
<td>$3.50</td>
<td>100000001</td>
<td>12/13/2009</td>
<td>(707) 124-1241</td>
<td>Ap #390-1150 Ornare. Av. Chico, CA 95973</td>
</tr>
</tbody>
</table>
**Report #2**

**#17 Use case: Media Usage Report**

**Actor(s): Library Manager**

Description: The Library Manager selects this from the report menu item and lists the media usage report between the selected Start and End date.

**MEDIA USAGE REPORT**

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Last Checked Out Date</th>
<th>Number Of Times Checked Out In Period</th>
<th>Initial Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animal Farm-Nineteen Eighty-Four</td>
<td>Adams Douglas, George Orwell</td>
<td>05/04/2010</td>
<td>5</td>
<td>New</td>
</tr>
<tr>
<td>BigFish</td>
<td>J Tolkein</td>
<td>11/29/2009</td>
<td>1</td>
<td>New</td>
</tr>
<tr>
<td>Ender's Game</td>
<td>Card Orson</td>
<td>05/05/2010</td>
<td>1</td>
<td>New</td>
</tr>
<tr>
<td>Family Matters</td>
<td>Ira Berkowitz</td>
<td>04/07/2010</td>
<td>1</td>
<td>New</td>
</tr>
<tr>
<td>Foundation</td>
<td>Asimov Isaac</td>
<td>04/01/2010</td>
<td>1</td>
<td>New</td>
</tr>
<tr>
<td>Restaurant at the End of the Universe</td>
<td>Adams Douglas</td>
<td>09/08/2009</td>
<td>1</td>
<td>New</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Last Checked Out Date</th>
<th>Number Of Times Checked Out In Period</th>
<th>Initial Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animal Farm-Nineteen Eighty-Four</td>
<td>Adams Douglas, George Orwell</td>
<td>05/04/2010</td>
<td>5</td>
<td>Good</td>
</tr>
<tr>
<td>BigFish</td>
<td>J Tolkein</td>
<td>11/29/2009</td>
<td>1</td>
<td>New</td>
</tr>
<tr>
<td>Ender's Game</td>
<td>Card Orson</td>
<td>05/05/2010</td>
<td>1</td>
<td>New</td>
</tr>
<tr>
<td>Family Matters</td>
<td>Ira Berkowitz</td>
<td>04/07/2010</td>
<td>1</td>
<td>New</td>
</tr>
<tr>
<td>Foundation</td>
<td>Asimov Isaac</td>
<td>04/01/2010</td>
<td>1</td>
<td>New-Pr</td>
</tr>
<tr>
<td>Restaurant at the End of the Universe</td>
<td>Adams Douglas</td>
<td>09/08/2009</td>
<td>1</td>
<td>New-Good</td>
</tr>
</tbody>
</table>
Report #3

#18 Use case: Checked out Media Report

Actor(s): Library Manager

Description: The Library Manager selects this from the report menu item and lists all the Media that is currently checked out.

<table>
<thead>
<tr>
<th>Title</th>
<th>Full Name</th>
<th>libraryCardNumber</th>
<th>To Be Returned Date</th>
<th>Address</th>
<th>Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animal Farm-Nineteen Eighty-Four</td>
<td></td>
<td>100000001</td>
<td>05/18/2010</td>
<td>Ap #390-1150 Ornare. Av. Chico, CA 95973</td>
<td>(707) 124-1241</td>
</tr>
<tr>
<td>BigFish</td>
<td></td>
<td>100000001</td>
<td>12/13/2009</td>
<td>Ap #390-1150 Ornare. Av. Chico, CA 95973</td>
<td>(707) 124-1241</td>
</tr>
</tbody>
</table>
**Report #3**

**#12 Use case: Over Due Report**

**Actor(s):** Librarian, Library Manager

**Description:** The employee selects this from the report menu item and lists all the Media that is overdue to be returned.

### OVER DUE REPORT

<table>
<thead>
<tr>
<th>Title</th>
<th>Full Name</th>
<th>LibraryCardNumber</th>
<th>To Be Returned Date</th>
<th>Address</th>
<th>Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigFish</td>
<td></td>
<td>100000001</td>
<td>12/13/2009</td>
<td>Ap #390-1150 Ornare. Av. Chico, CA 95973</td>
<td>(707) 124-1241</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Title</th>
<th>Full Name</th>
<th>LibraryCardNumber</th>
<th>To Be Returned Date</th>
<th>Address</th>
<th>Phone Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigFish</td>
<td></td>
<td>100000001</td>
<td>12/13/2009</td>
<td>Ap #390-1150 Ornare. Av. Chico, CA 95973</td>
<td>124-1241</td>
</tr>
</tbody>
</table>
SQL Statements

Table Creation

```sql
CREATE TABLE [dbo].[Author](
    [Fname] [nvarchar](50) NOT NULL,
    [Minit] [nchar](1) NOT NULL,
    [Lname] [nvarchar](50) NOT NULL,
    [DOB] [date] NOT NULL,
CONSTRAINT [PK__Author__M__EFC884A033D4B598] PRIMARY KEY CLUSTERED
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
)
```

```sql
CREATE TABLE [dbo].[Check_Out](
    [LibraryCardNumber] [nvarchar](9) NOT NULL,
    [BarCode] [bigint] NOT NULL,
    [CheckOutDate] [datetime] NOT NULL,
    [Renewed] [bit] NULL,
    [AmountPaid] [money] NULL,
    [Fee] [money] NULL,
    [ToBeReturnedDate] [date] NULL,
    [Checker#] [nvarchar](9) NOT NULL,
    [CheckedInDate] [datetime] NULL,
CONSTRAINT [PK_Check_Out] PRIMARY KEY CLUSTERED
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
)
```

```sql
GO

ALTER TABLE [dbo].[Check_Out]  WITH CHECK ADD  CONSTRAINT [FK_Check_Out_Employee]
FOREIGN KEY([Checker#])
REFERENCES [dbo].[Employee] ([EmpID])
```
GO

ALTER TABLE [dbo].[Check_Out] CHECK CONSTRAINT [FK_Check_Out_Employee]
GO

ALTER TABLE [dbo].[Check_Out] WITH CHECK ADD CONSTRAINT [FK_Check_Out_Media]
FOREIGN KEY ([BarCode])
REFERENCES [dbo].[Media] ([Barcode])
GO

ALTER TABLE [dbo].[Check_Out] CHECK CONSTRAINT [FK_Check_Out_Media]
GO

ALTER TABLE [dbo].[Check_Out] WITH CHECK ADD CONSTRAINT [FK_Check_Out_Patron]
FOREIGN KEY ([LibraryCardNumber])
REFERENCES [dbo].[Patron] ([LibraryCardNumber])
GO

ALTER TABLE [dbo].[Check_Out] CHECK CONSTRAINT [FK_Check_Out_Patron]
GO

CREATE TABLE [dbo].[Condition](
    [Grade] [nvarchar](8) NOT NULL,
    [Cost_Percentage] [int] NOT NULL,
 CONSTRAINT [PK.Condition] PRIMARY KEY CLUSTERED
 (    [Grade] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
 ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

CREATE TABLE [dbo].[Employee](
    [EmplID] [nvarchar](9) NOT NULL,
    [SSN] [nchar](9) NOT NULL,
    [fName] [nvarchar](20) NOT NULL,
    [mName] [nvarchar](20) NULL,
    [IName] [nvarchar](20) NOT NULL,
    [payRate] [nvarchar](50) NOT NULL,
    [DOB] [datetime] NOT NULL,
    [Address] [nvarchar](max) NOT NULL,
    [phoneNumber] [nvarchar](15) NOT NULL,
    [roleType] [nvarchar](20) NOT NULL,
    [username] [nvarchar](20) NOT NULL,
    [Epassword] [nvarchar](20) NOT NULL,
CONSTRAINT [PK_Employee] PRIMARY KEY CLUSTERED
(
    [EmpID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

CREATE TABLE [dbo].[Media](
    [Barcode] [bigint] NOT NULL,
    [Title] [nvarchar](100) NOT NULL,
    [ISBN] [int] NOT NULL,
    [Year] [datetime] NULL,
    [Publisher] [nvarchar](100) NULL,
    [Type] [nvarchar](50) NOT NULL,
    [Subject] [nvarchar](100) NOT NULL,
    [OrderCost] [int] NOT NULL,
    [InitialCondition] [nvarchar](50) NOT NULL,
    [MediaCondition] [nvarchar](50) NOT NULL,
    [Grade] [nvarchar](8) NOT NULL,
CONSTRAINT [PK_Media] PRIMARY KEY CLUSTERED
(
    [Barcode] 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].[Media] WITH CHECK ADD CONSTRAINT [CK_Media] CHECK (((ISBN)>0))
GO

ALTER TABLE [dbo].[Media] CHECK CONSTRAINT [CK_Media]
GO

CREATE TABLE [dbo].[Media_Author](
    [Barcode] [bigint] NOT NULL,
    [Auth FName] [nvarchar](50) NOT NULL,
    [Auth LName] [nvarchar](50) NOT NULL,
    [Auth DOB] [date] NOT NULL,
CONSTRAINT [PK_Media_Author] PRIMARY KEY CLUSTERED
(
    [Barcode] ASC,
    [Auth FName] ASC,
    [Auth LName] 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].[Media_Author] WITH CHECK ADD CONSTRAINT [CK_Media_Author] CHECK (((ISBN)>0))
GO

ALTER TABLE [dbo].[Media_Author] CHECK CONSTRAINT [CK_Media_Author]
GO
"[AuthLName] ASC,
[AuthDOB] 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].[Media_Author] WITH CHECK ADD CONSTRAINT
[FK_Media_Author_Author] FOREIGN KEY([AuthFName], [AuthLName], [AuthDOB])
REFERENCES [dbo].[Author] ([Fname], [Lname], [DOB])
GO

ALTER TABLE [dbo].[Media_Author] CHECK CONSTRAINT [FK_Media_Author_Author]
GO

ALTER TABLE [dbo].[Media_Author] WITH CHECK ADD CONSTRAINT
[FK_Media_Author_Media] FOREIGN KEY([Barcode])
REFERENCES [dbo].[Media] ([Barcode])
GO

ALTER TABLE [dbo].[Media_Author] CHECK CONSTRAINT [FK_Media_Author_Media]
GO

CREATE TABLE [dbo].[Patron] (
    [libraryCardNumber] [nvarchar](9) NOT NULL,
    [fName] [nvarchar](50) NOT NULL,
    [mName] [nvarchar](20) NULL,
    [lName] [nvarchar](50) NOT NULL,
    [DOB] [date] NOT NULL,
    [phoneNumber] [nvarchar](20) NOT NULL,
    [Address] [nvarchar](max) NULL,
    [Username] [nvarchar](20) NOT NULL,
    [password] [nvarchar](20) NOT NULL,
CONSTRANT [PK_Patron] PRIMARY KEY CLUSTERED
(
    [libraryCardNumber] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

CREATE TABLE [dbo].[Policies](
    [Description] [nvarchar](50) NOT NULL,
(Value) [decimal](18, 2) NOT NULL)
) ON [PRIMARY]

CREATE TABLE [dbo].[Reservation](
    [LibraryCardNumber] [nvarchar](9) NOT NULL,
    [BarCode] [bigint] NOT NULL,
    [DateRequested] [datetime] NOT NULL,
    [CheckOutStatus] [bit] NOT NULL,
    [ReshelfDate] [datetime] NOT NULL,
    CONSTRAINT [PK_Reservation] PRIMARY KEY CLUSTERED
    (]
        [LibraryCardNumber] ASC,
        [BarCode] ASC,
        [DateRequested] 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].[Reservation] WITH CHECK ADD CONSTRAINT [FK_Reservation_Media]
FOREIGN KEY([BarCode])
REFERENCES [dbo] [Media] ([Barcode])
GO

ALTER TABLE [dbo].[Reservation] CHECK CONSTRAINT [FK_Reservation_Media]
GO

ALTER TABLE [dbo].[Reservation] WITH CHECK ADD CONSTRAINT [FK_Reservation_Patron]
FOREIGN KEY([LibraryCardNumber])
REFERENCES [dbo] [Patron] ([libraryCardNumber])
GO

ALTER TABLE [dbo].[Reservation] CHECK CONSTRAINT [FK_Reservation_Patron]
GO
Stored Procedures

Create PROCEDURE [dbo].[Author_GetByBarCode]
    @Barcode nvarchar(50)
    -- Add the parameters for the stored procedure here
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    select *, Fname + ' ' + Lname as 'fullName', Fname + '_' + Lname + '_' +
    CONVERT(VARCHAR(10), a.DOB ,111) as 'DOB'
    from Author a
    where Exists(Select * from Media_Author ma where ma.Barcode = @Barcode and
    ma.AuthFName = a.Fname and ma.AuthLName = a.Lname and ma.AuthDOB = a.DOB)
    order by Fname DESC
END

CREATE PROCEDURE [dbo].[Author_GetGridView]
    -- Add the parameters for the stored procedure here
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    select *, Fname + ' ' + Lname as 'fullName'
    from Author
    order by Lname DESC
END

CREATE PROCEDURE [dbo].[Check_Out_GetByBarcode]
    @Barcode nvarchar(50)
    -- Add the parameters for the stored procedure here
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
select c.LibraryCardNumber, c.CheckOutDate, c.CheckedInDate, c.ToBeReturnedDate, c.AmountPaid, e.fName + ' ' + e.lName as 'CheckedOutEmployee'
from Check_Out c
left join Employee e on e.EmplID = c.Checker#
where c.BarCode = @Barcode
order by c.CheckOutDate DESC
END

CREATE PROCEDURE [dbo].[Check_Out_GetByLibraryCardNumber]
@LibraryCardNumber nvarchar(50)
-- Add the parameters for the stored procedure here
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
select c.LibraryCardNumber, c.CheckOutDate, c.CheckedInDate, c.ToBeReturnedDate, c.AmountPaid, e.fName + ' ' + e.lName as 'CheckedOutEmployee'
from Check_Out c
left join Employee e on e.EmplID = c.Checker#
where c.LibraryCardNumber = @LibraryCardNumber
order by c.CheckOutDate DESC
END

CREATE PROCEDURE [dbo].[Employee_GetGridViewAll]
-- Add the parameters for the stored procedure here
AS
BEGIN
-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

-- Insert statements for procedure here
select e.fName + ' ' + e.lName as 'fullName', PARSENAME(@$+Convert(varchar,Convert(money,e.payRate),1),2) as payRate, e.roleType, e.DOB, e.Address, e.EmplID,
(CASE
    when e.phoneNumber = ' ' then ''
    when e.phoneNumber is not null then '(' + LEFT(e.phoneNumber, 3) + ') ' + SUBSTRING(e.phoneNumber, 4,3) + ' ' + RIGHT(e.phoneNumber,4)
from Employee e
END

CREATE PROCEDURE [dbo].[Media_GetGridViewAll]
-- Add the parameters for the stored procedure here
AS
BEGIN
  -- SET NOCOUNT ON added to prevent extra result sets from
  -- interfering with SELECT statements.
  SET NOCOUNT ON;

  -- Insert statements for procedure here
  select m.Title,
    SUBSTRING((SELECT DISTINCT ' ' + (a.FName + ' ' + a.Lname) FROM Media_Author ma
      Left Outer JOIN Author a ON a.Fname = ma.AuthFName
      Where ma.Barcode = m.Barcode and a.Lname = ma.AuthLName and a.DOB =
    ma.AuthDOB
    FOR XML PATH('') ) , 2, 8000) as Author,

    (SELECT DISTINCT c.CheckOutDate FROM Check_Out c WHERE c.BarCode = m.Barcode and
    c.CheckedInDate is null ) as checkOutDate,
    m.ISBN, m.Barcode, m.Type

  from Media m
END

AS
BEGIN
  -- SET NOCOUNT ON added to prevent extra result sets from
  -- interfering with SELECT statements.
  SET NOCOUNT ON;

  -- Insert statements for procedure here
  select p.fName + ' ' + p.lName as 'fullName', p.libraryCardNumber, p.DOB, p.Address,

  (CASE
    when p.phoneNumber = '' then ''
    when p.phoneNumber is not null then '(' + LEFT(p.phoneNumber, 3) + ') ' +
SUBSTRING(p.phoneNumber, 4,3) + 'l' + RIGHT(p.phoneNumber,4)
END) as phoneNumber

from Patron p
END

CREATE PROCEDURE [dbo].[Report_CheckedOut]
-- Add the parameters for the stored procedure here
AS
BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

select m.Title, p.fName + 'l' + p.lName, p.libraryCardNumber, c.ToBeReturnedDate, p.Address,
(CASE
  when p.phoneNumber = '' then ''
  when p.phoneNumber is not null then '(' + LEFT(p.phoneNumber, 3) + ')l' +
  SUBSTRING(p.phoneNumber, 4,3) + ')' + RIGHT(p.phoneNumber,4)
END) as phoneNumber

from Check_Out c
left join Patron p on p.libraryCardNumber = c.LibraryCardNumber
left join Media m on m.Barcode = c.BarCode
where c.CheckedInDate is null
END

CREATE PROCEDURE [dbo].[Report_FeeReport]
-- Add the parameters for the stored procedure here
AS
BEGIN

-- SET NOCOUNT ON added to prevent extra result sets from
-- interfering with SELECT statements.
SET NOCOUNT ON;

select DISTINCT m.Title, DATEDIFF(day, c.CheckOutDate, c.ToBeReturnedDate) * (Select top 1 p.Value from Policies p where p.Description = 'FeePerDay') as 'Fee',
p.fName + 'l' + p.lName as 'fullName', p.libraryCardNumber, c.ToBeReturnedDate,
(CASE
  when p.phoneNumber = '' then ''
  when p.phoneNumber is not null then '(' + LEFT(p.phoneNumber, 3) + ')l' +
SUBSTRING(p.phoneNumber, 4, 3) + 'l' + RIGHT(p.phoneNumber, 4)
   END) as phoneNumber,
p.Address,

DATEDIFF(day, c.CheckOutDate, c.ToBeReturnedDate),
(Select top 1 p.Value from Policies p where p.Description = 'CheckOutLength')

from Check_Out c
left join Media m on m.Barcode = c.BarCode
left join Patron p on p.libraryCardNumber = c.LibraryCardNumber
where c.CheckedInDate is null and DATEDIFF(day, c.CheckOutDate, GETDATE()) > (Select top 1 p.Value from Policies p where p.Description = 'CheckOutLength')

CREATE PROCEDURE [dbo].[Report_MediaUsageReport]
   -- Add the parameters for the stored procedure here
   @startdate datetime,
   @enddate datetime
AS
BEGIN

   -- SET NOCOUNT ON added to prevent extra result sets from
   -- interfering with SELECT statements.
   SET NOCOUNT ON;

   select DISTINCT m.Title ,
      
SUBSTRING((SELECT DISTINCT ',', (a.FirstName + ' ' + a.LastName) FROM Media_Author ma
       Left Outer JOIN Author a ON a.FirstName = ma.AuthorName
       Where ma.Barcode = m.Barcode and a.LastName = ma.AuthorName and a.DOB =
ma.AuthorDOB
FOR XML PATH('"'), 2, 8000) as Author,

(Select MAX(c.CheckOutDate) from Check_Out c where c.BarCode = m.Barcode and
   c.CheckOutDate >= @startdate and c.CheckOutDate <= @enddate) as LastCheckOutDate,

(Select COUNT(*) from Check_Out c where c.BarCode = m.Barcode and
   c.CheckOutDate >= @startdate and c.CheckOutDate <= @enddate) as NumberOfTimes,
   m.InitialCondition,

(Select Top 1 c.CurrentCondition from Check_Out c where c.BarCode = m.Barcode and
   c.CheckOutDate >= @startdate and c.CheckOutDate <= @enddate ) as CurrentCondition
   from Media m
   where exists(Select * from Check_Out c where c.CheckOutDate >= @startdate and
c.CheckOutDate <= @enddate)
  END
CREATE PROCEDURE [dbo].[Report_Overdue]
  -- Add the parameters for the stored procedure here
AS
BEGIN
  -- SET NOCOUNT ON added to prevent extra result sets from
  -- interfering with SELECT statements.
  SET NOCOUNT ON;

  select m.Title, p.fName + ' ' + p.lName, p.libraryCardNumber, c.ToBeReturnedDate, p.Address,
    (CASE
       when p.phoneNumber = '' then ''
       when p.phoneNumber is not null then '(' + LEFT(p.phoneNumber, 3) + ') ' +
    SUBSTRING(p.phoneNumber, 4,3) + '-' + RIGHT(p.phoneNumber,4)
    END) as phoneNumber

  from Check_Out c
  left join Patron p on p.libraryCardNumber = c.LibraryCardNumber
  left join Media m on m.Barcode = c.BarCode
  where c.CheckedInDate is null and c.ToBeReturnedDate < GETDATE()
END
Installation Guide

1. Requires SQL server 2008 installed on the local host and IIS6-8 needs to be installed and normal ports open.
2. Run our install.bat file and enter the username and password for the sql server at the prompt.
   a. This will copy over all required site files and append the login information into the web.config
3. A default Library Manager will be entered into the database as username: 1_admin password 55555 by default.
4. Direct your browser to localhost and you may now begin testing your installation.
User Manual