Library Database Manageme
370 Project Phase II I

Vicky Holcomb
Nate Priddy
Joseph Fitzpatrick
Matthew Harris

Contents

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.

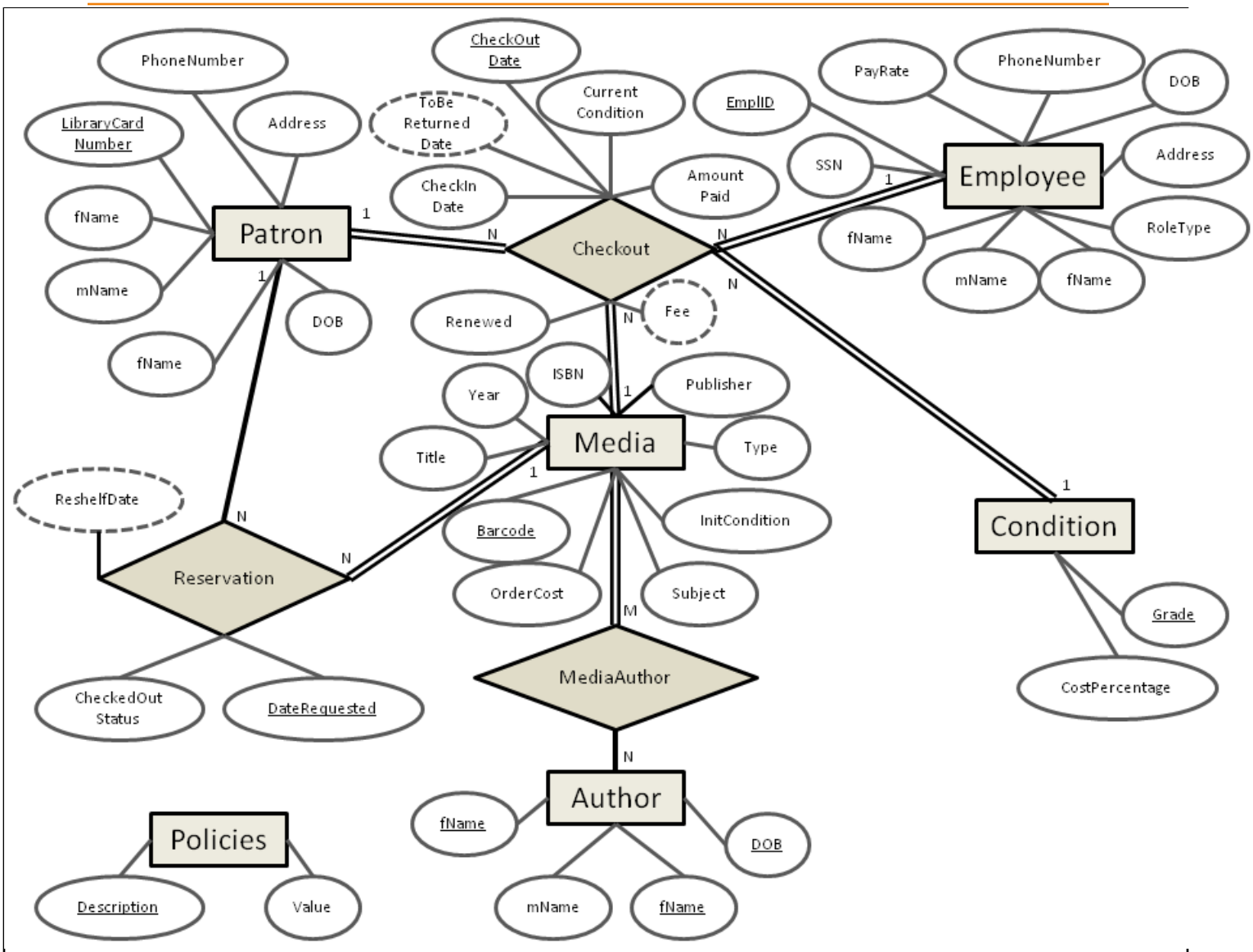
The library patrons will also have added usability. Potentially there could be computers in the library for their use to see what books they currently have checked out, and what they have checked out in the past. The information they could view would be limited, as in they would have a different set 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.

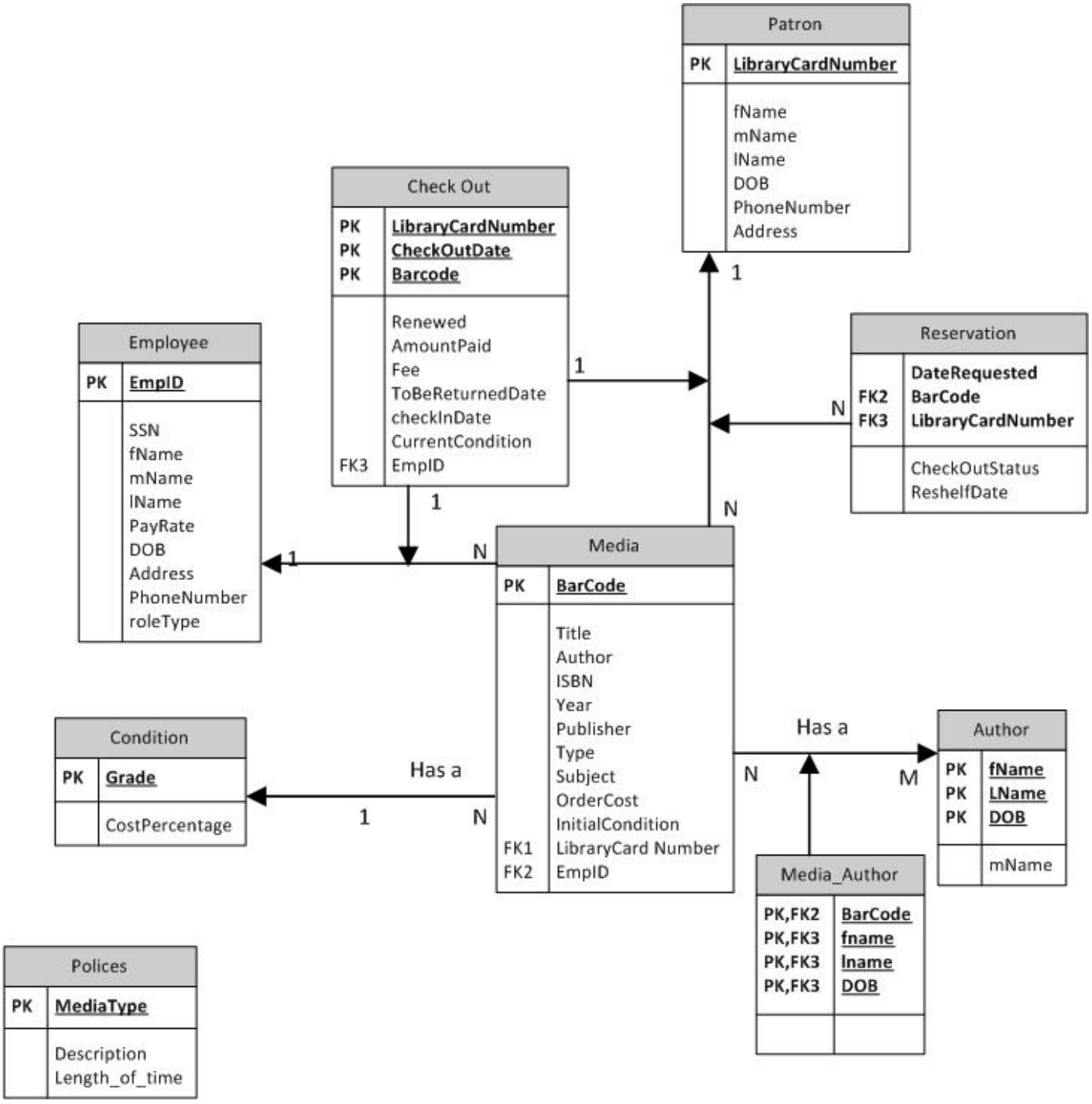
Lastly, the library manager will have the same access as the librarian as well as some added features. Employee records will all be stored in the database, replacing the paper system. He will also be able to order new media (not an actual function of the database) and add it to the system. He will then 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.

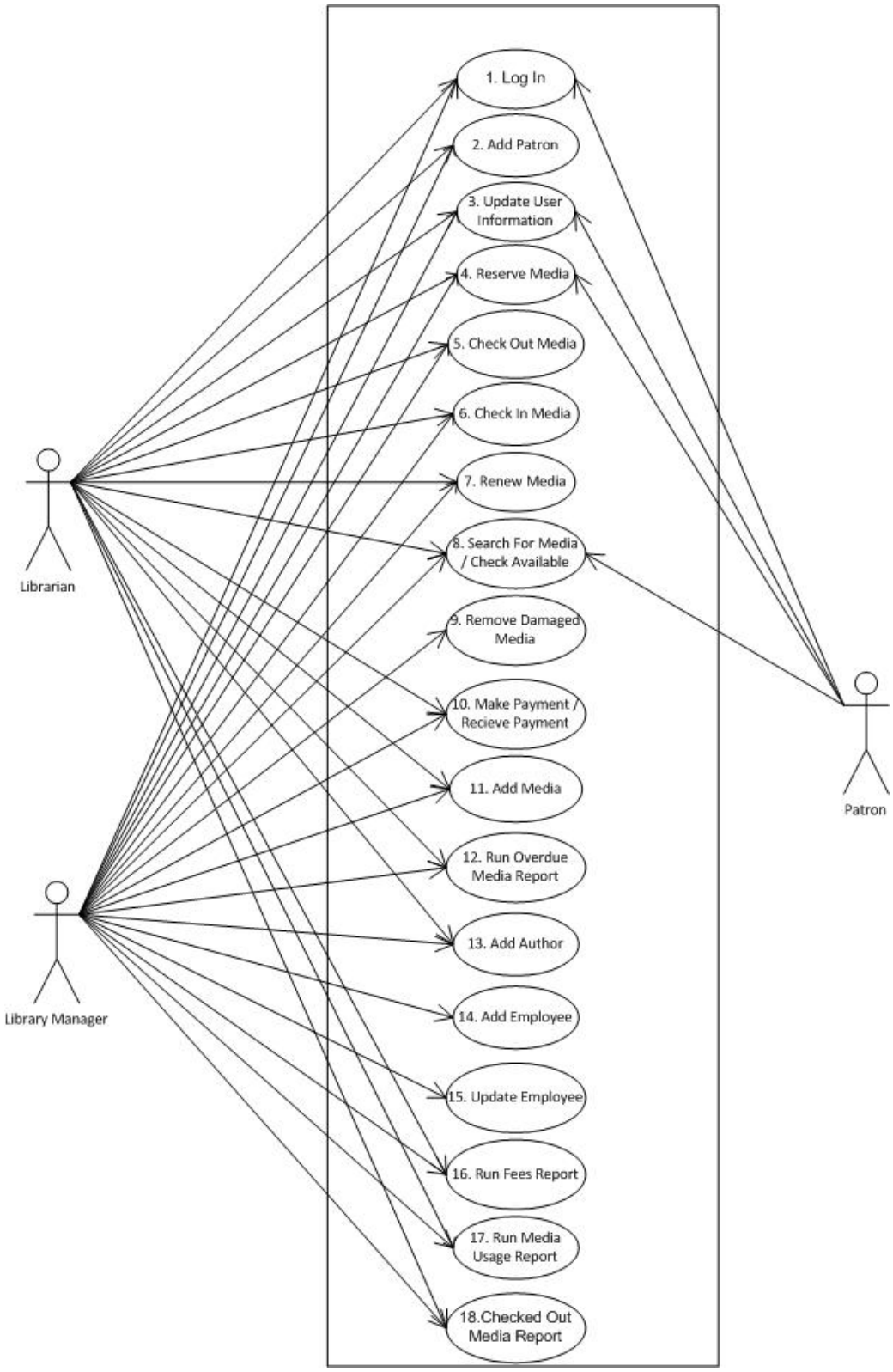
Conceptual Models:
Entity Relationship Model



Object Model Diagram



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.

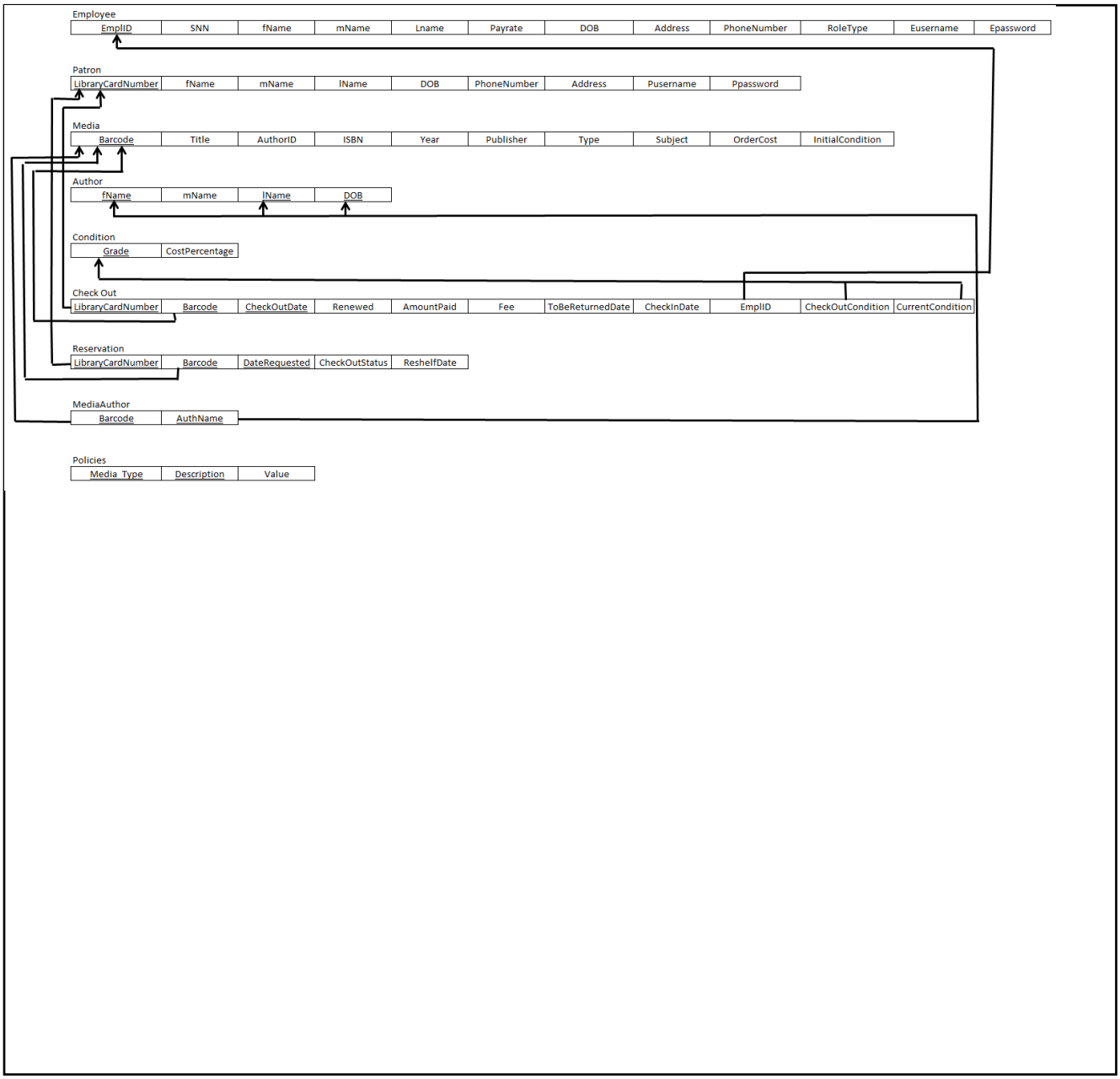
Use case 17: Run Media Usage Report
Actor(s): Library Manager

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

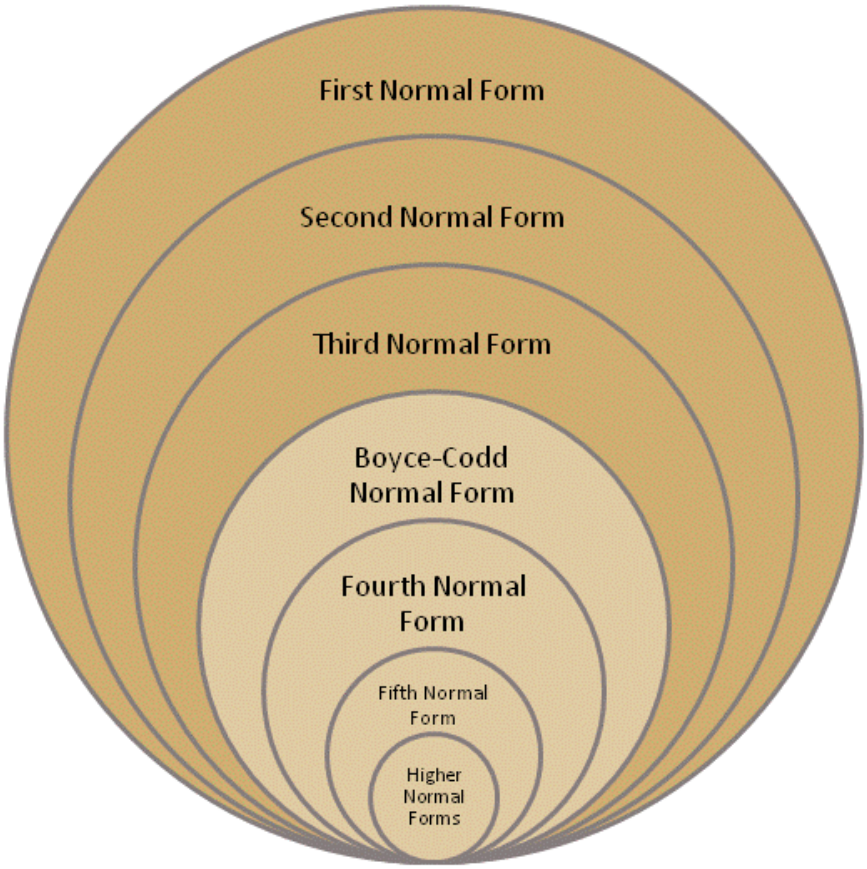


Table Title: Employee
State: Third Normal Form
Reason:
•No multi-valued attributes. Intersection of each row and column has only one value.
•No partial dependencies. One unique attribute to identify each tuple and no redundancies.
•No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.

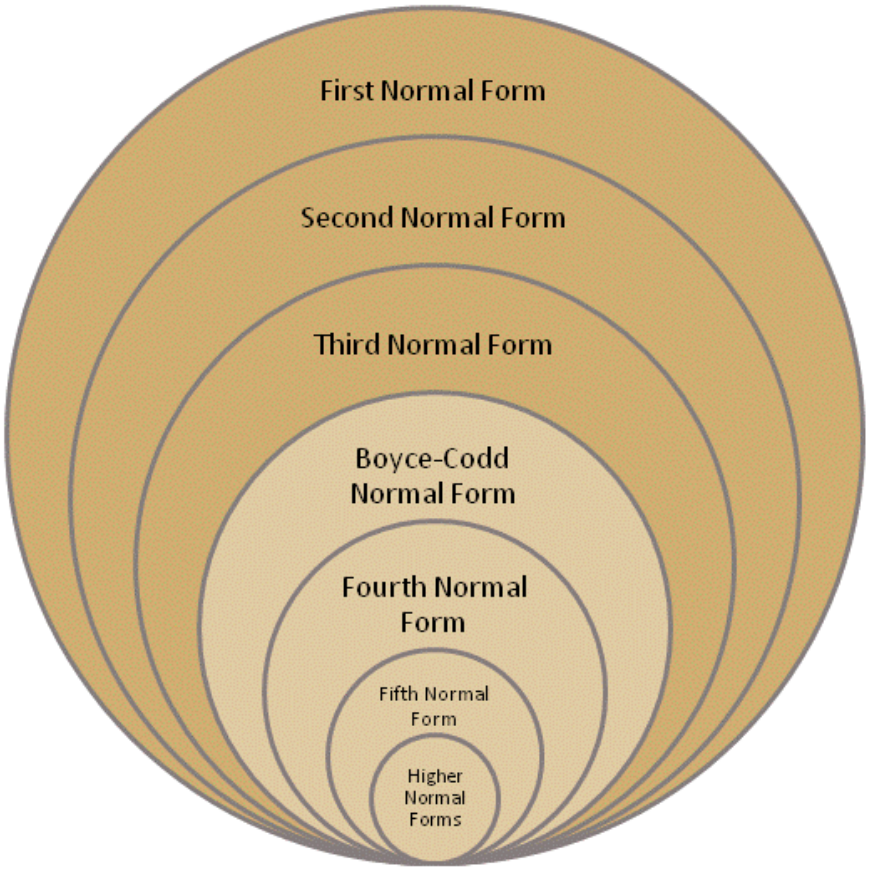


Table Title: Patron
State: Third Normal Form
Reason:
•No multi-valued attributes. Intersection of each row and column has only one value.
•No partial dependencies. One unique attribute to identify each tuple and no redundancies.
•No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.

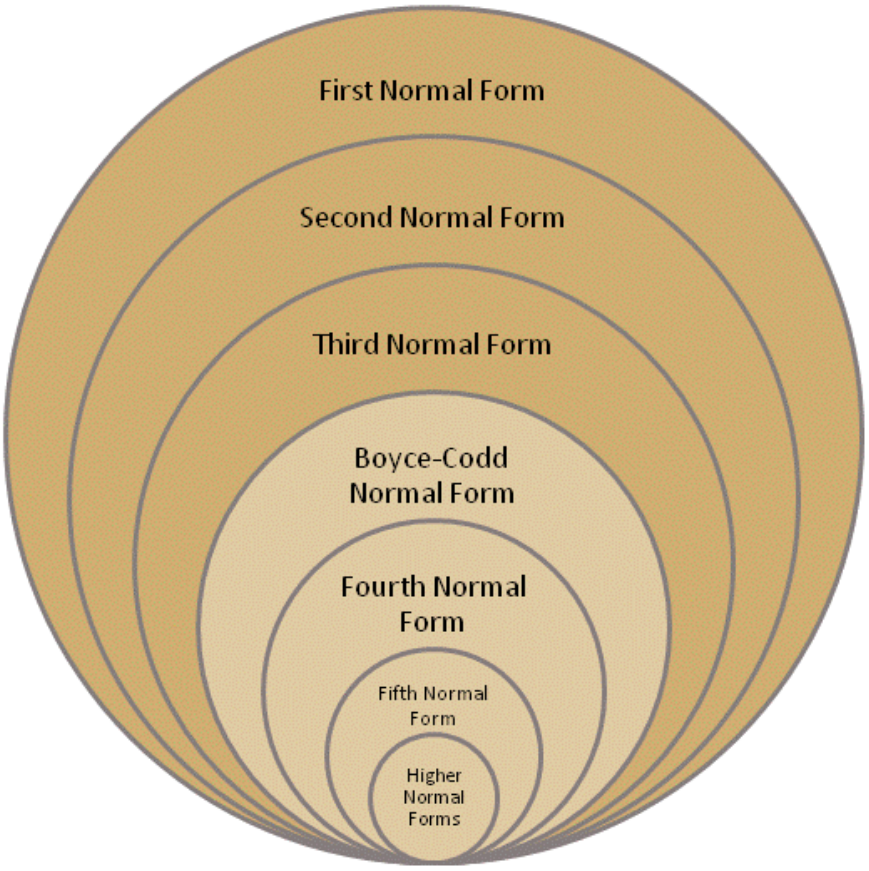


Table Title: Media
State: Third Normal Form
Reason:
•No multi-valued attributes. Intersection of each row and column has only one value.
•No partial dependencies. One unique attribute to identify each tuple and no redundancies.
•No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.

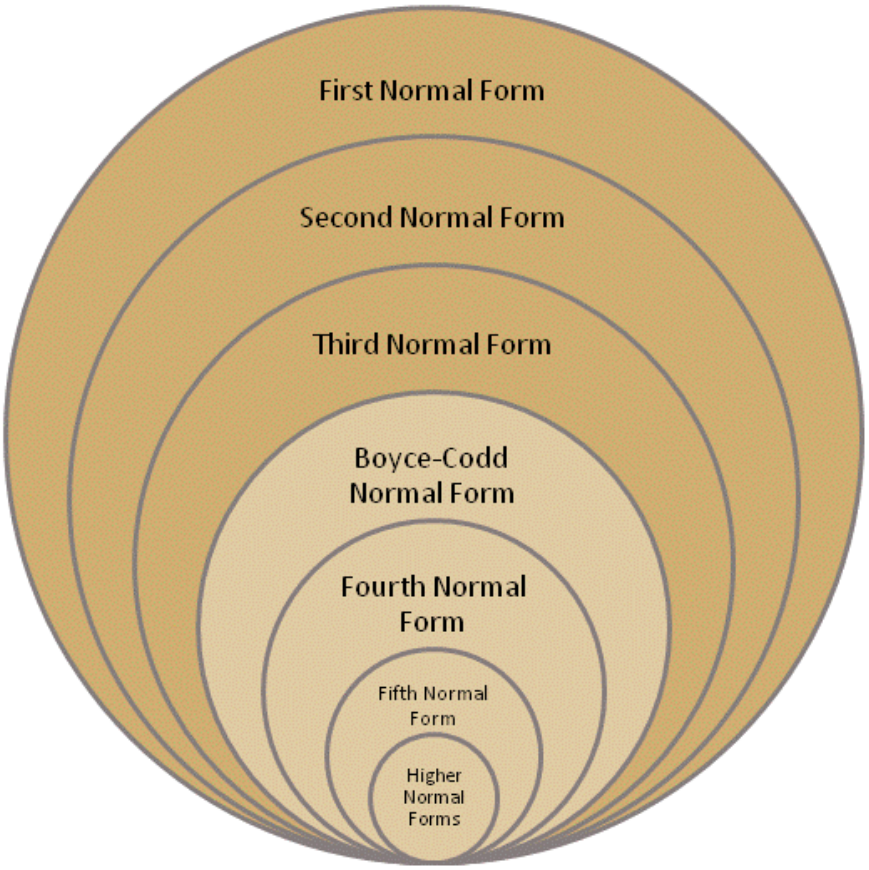


Table Title: Author
State: Third Normal Form
Reason:
•No multi-valued attributes. Intersection of each row and column has only one value.
•No partial dependencies. One unique attribute to identify each tuple and no redundancies.
•No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.

0

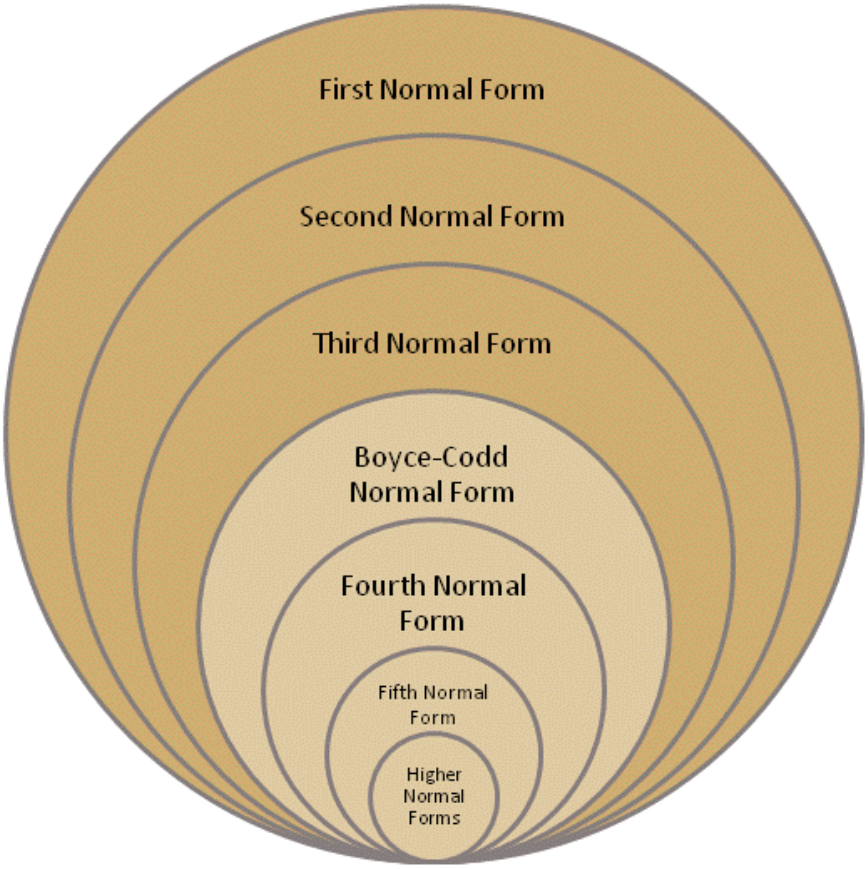


Table Title: Condition
State: Third Normal Form
Reason:
•No multi-valued attributes. Intersection of each row and column has only one value.
•No partial dependencies. One unique attribute to identify each tuple and no redundancies.
•No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.

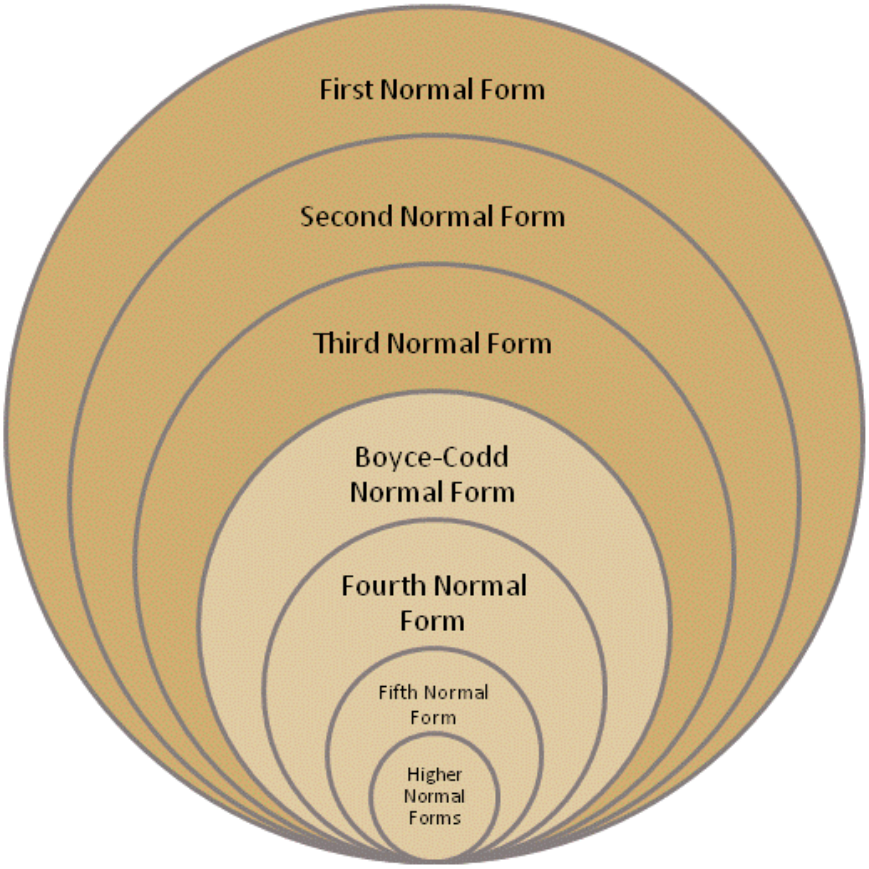


Table Title: Check Out
State: Third Normal Form
Reason:
•No multi-valued attributes. Intersection of each row and column has only one value.
•No partial dependencies. One unique attribute to identify each tuple and no redundancies.
•No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.

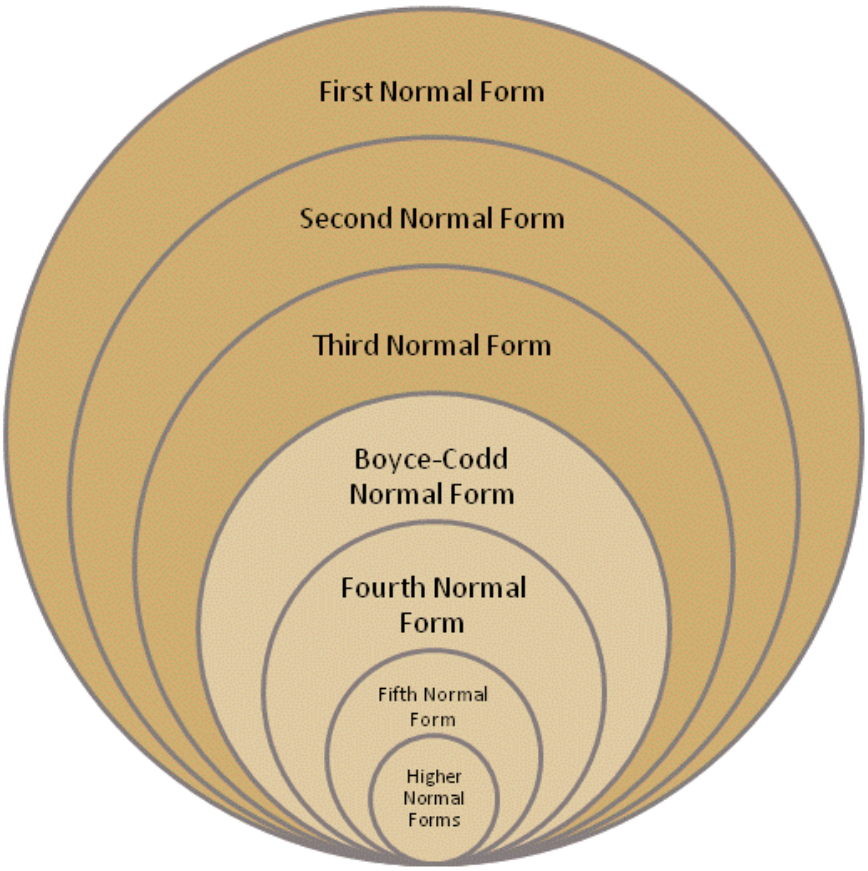


Table Title: Reservation
State: Third Normal Form
Reason:
•No multi-valued attributes. Intersection of each row and column has only one value.
•No partial dependencies. One unique attribute to identify each tuple and no redundancies.
•No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.

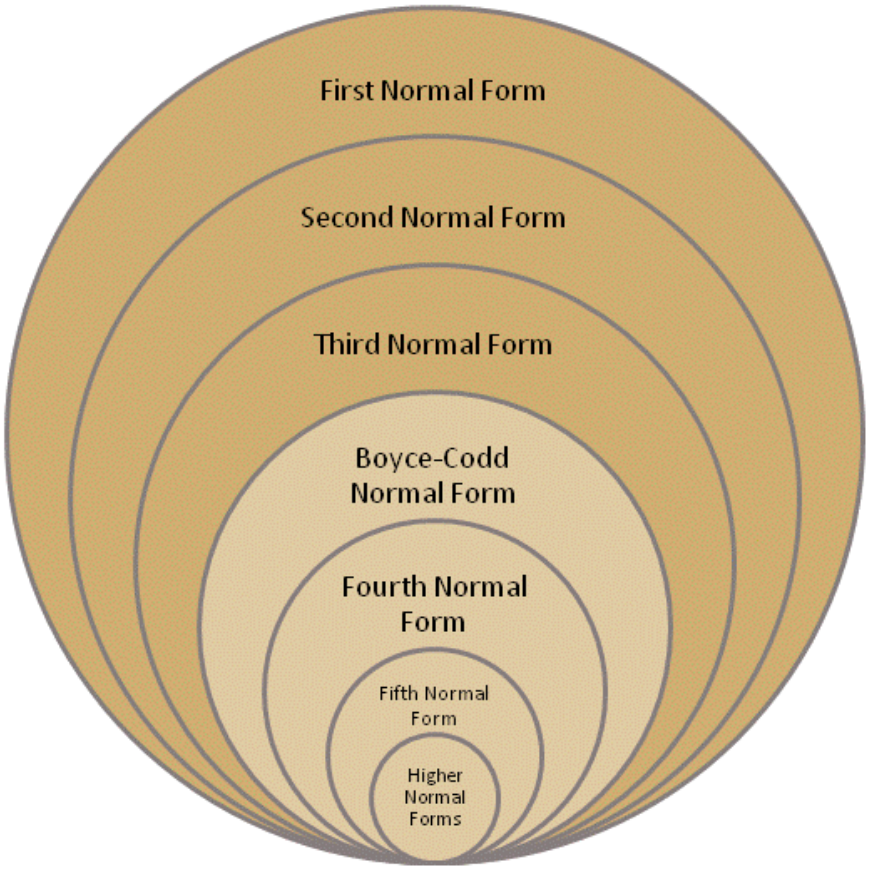


Table Title: Media-Author
State: Third Normal Form
Reason:
•No multi-valued attributes. Intersection of each row and column has only one value.
•No partial dependencies. One unique attribute to identify each tuple and no redundancies.
•No transitive dependencies. No non-primary-key attribute is transitively dependent on the primary key.

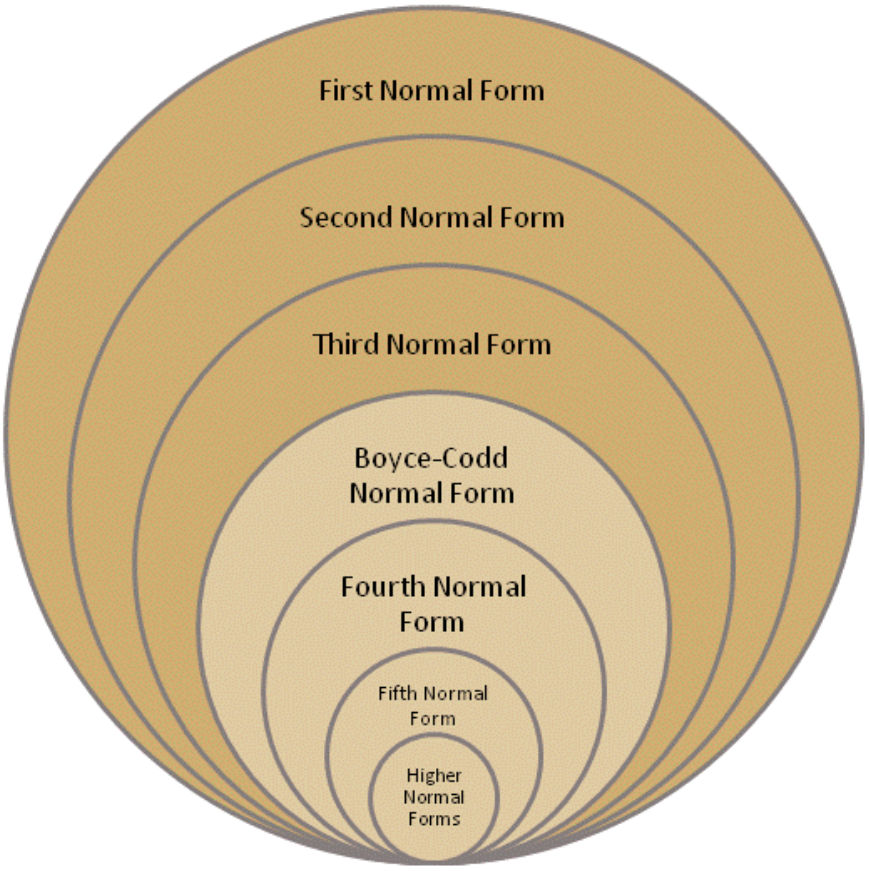
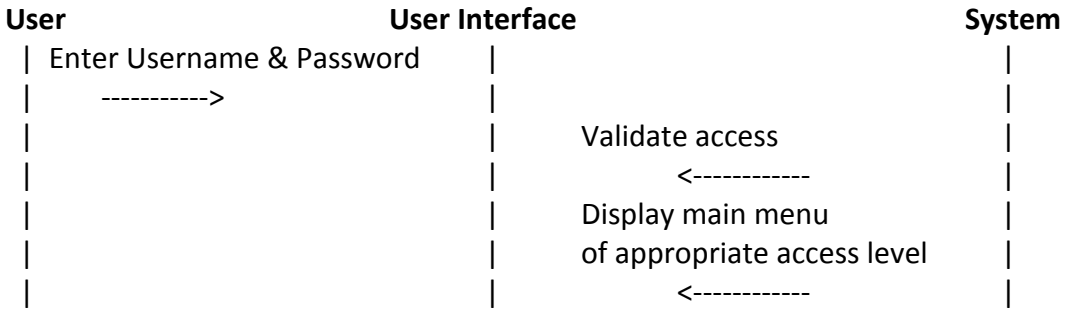


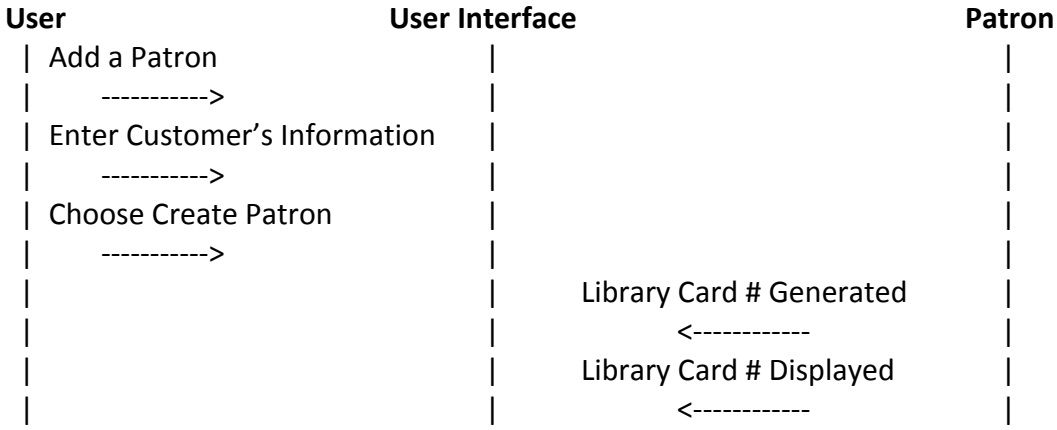
Table Title: Policies
State: Third Normal Form
Reason:
•No multi-valued attributes. Intersection of each row and column has only one value.
•No partial dependencies. One unique attribute to identify each tuple and no redundancies.
•No transitive dependencies. No non-primary-key attribute is transitively 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



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



Scenario for Update User Information

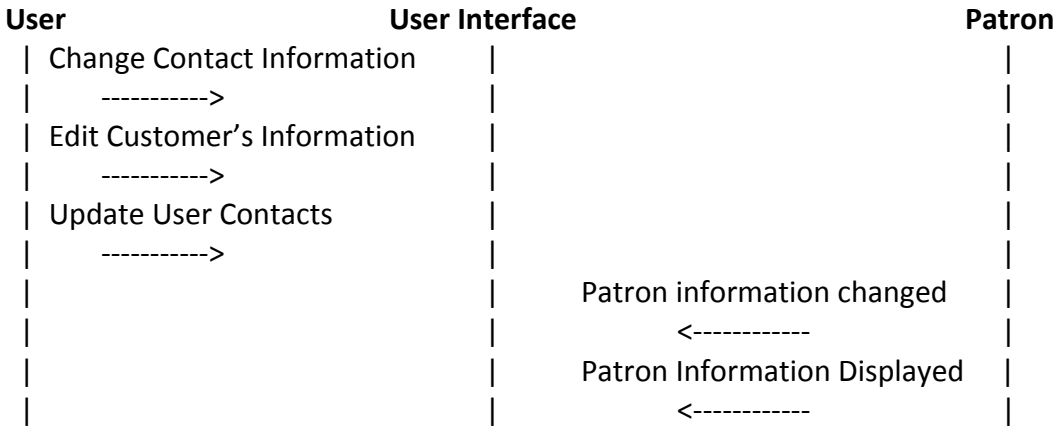
-A Librarian or Library Manager has moved or changed phone number and needs to update contact information

-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



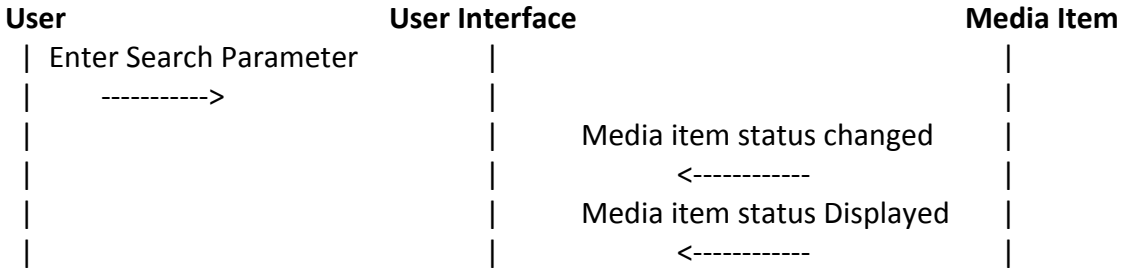
Scenario for Reserve Media

-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

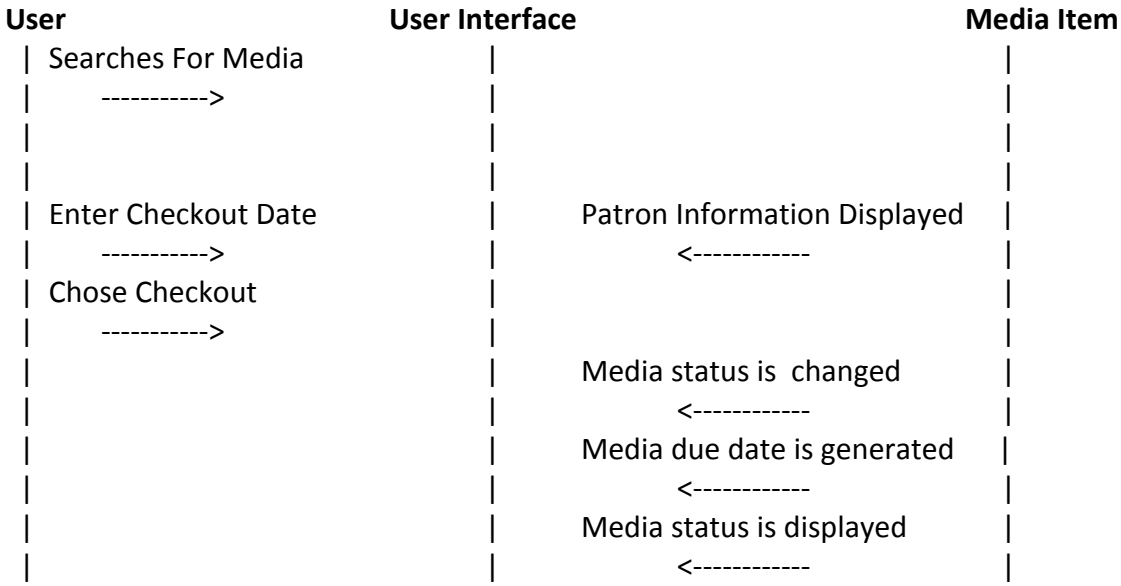
-A link is generated from a successful search for media

-Patron ID is entered



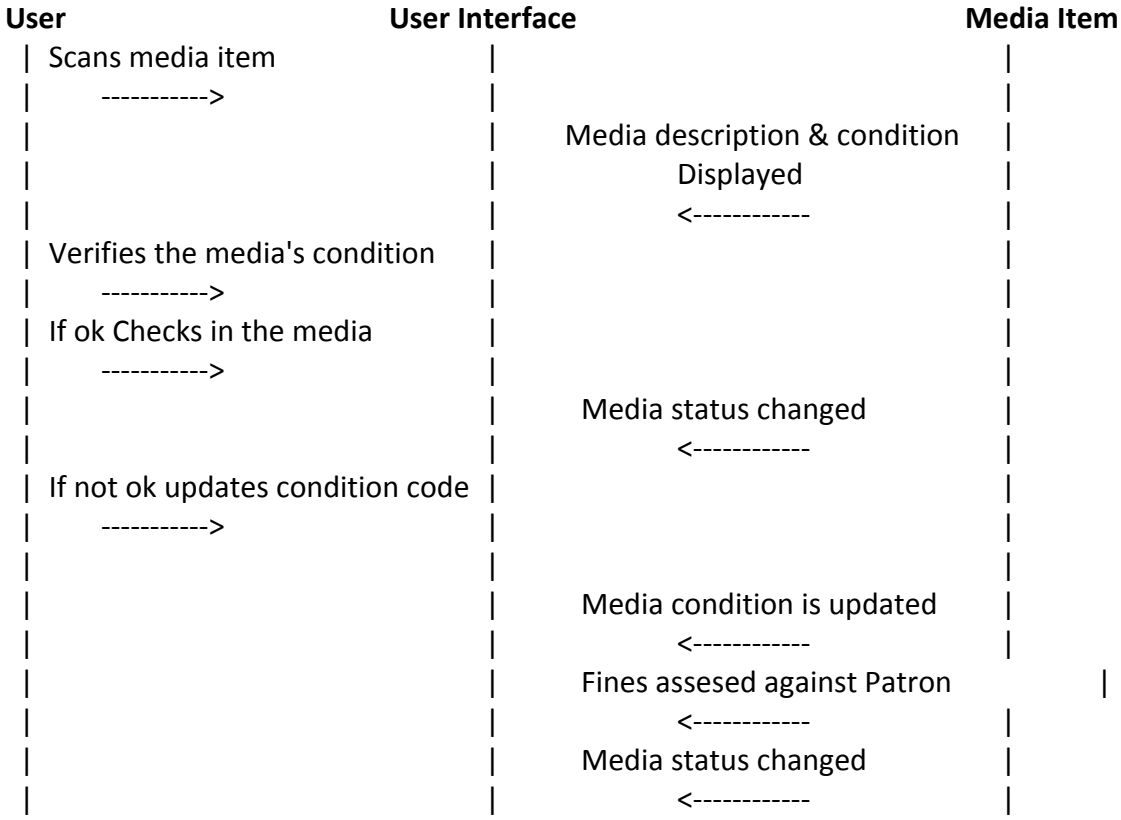
Scenario for Check out Media

-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



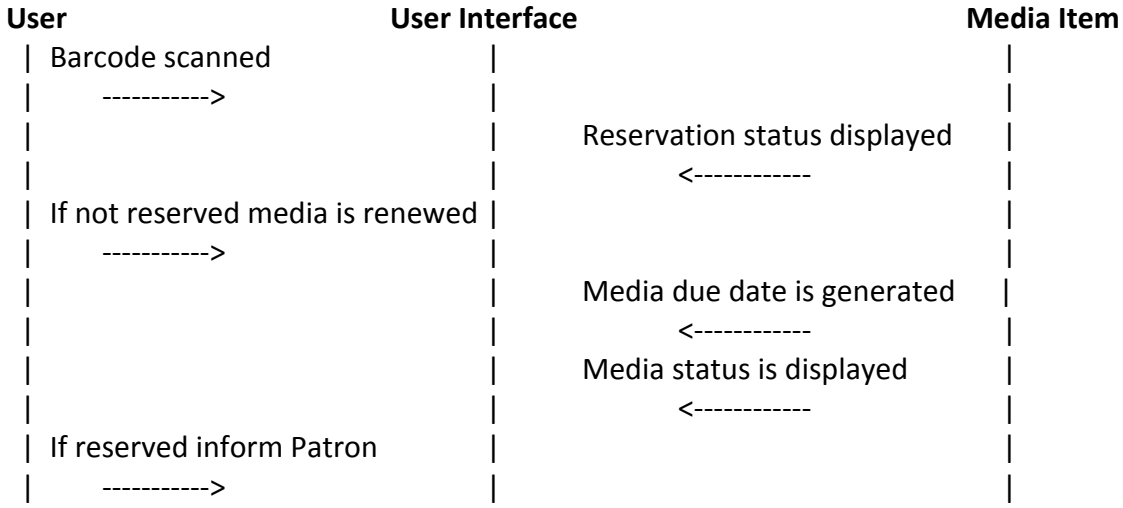
Scenario for Check in Media

-A Patron has an item they wish to check in
-A media is dropped off in the media return
-A Librarian scans the medias 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



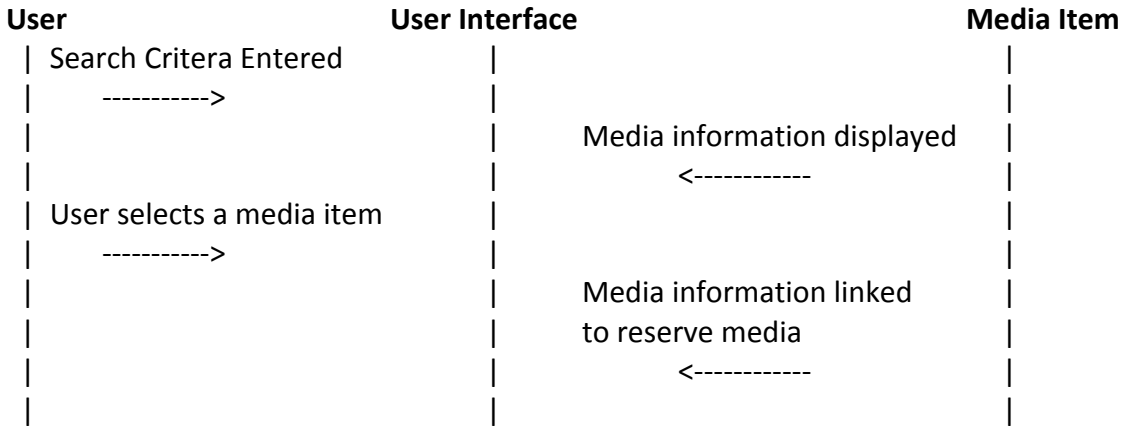
Scenario for Renew Media

-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



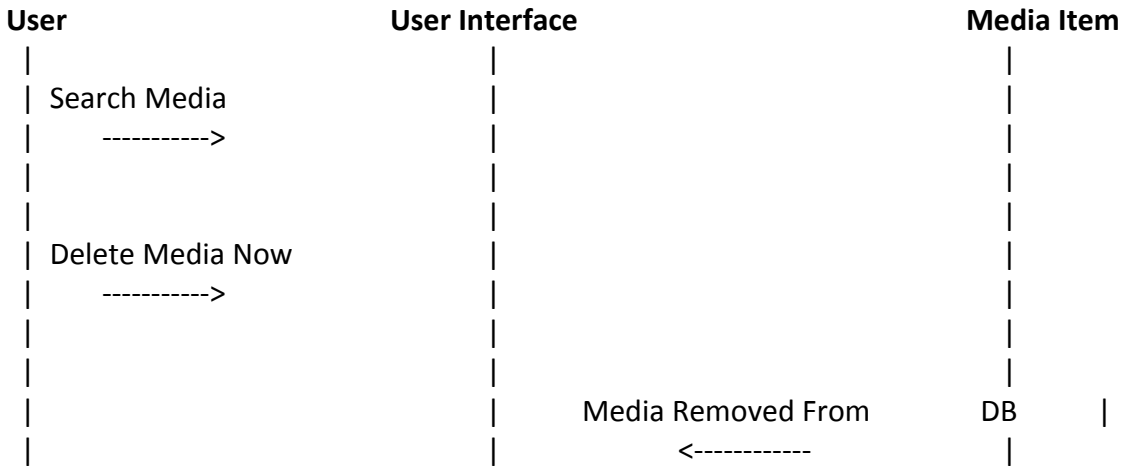
Scenario for Search for Media

-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



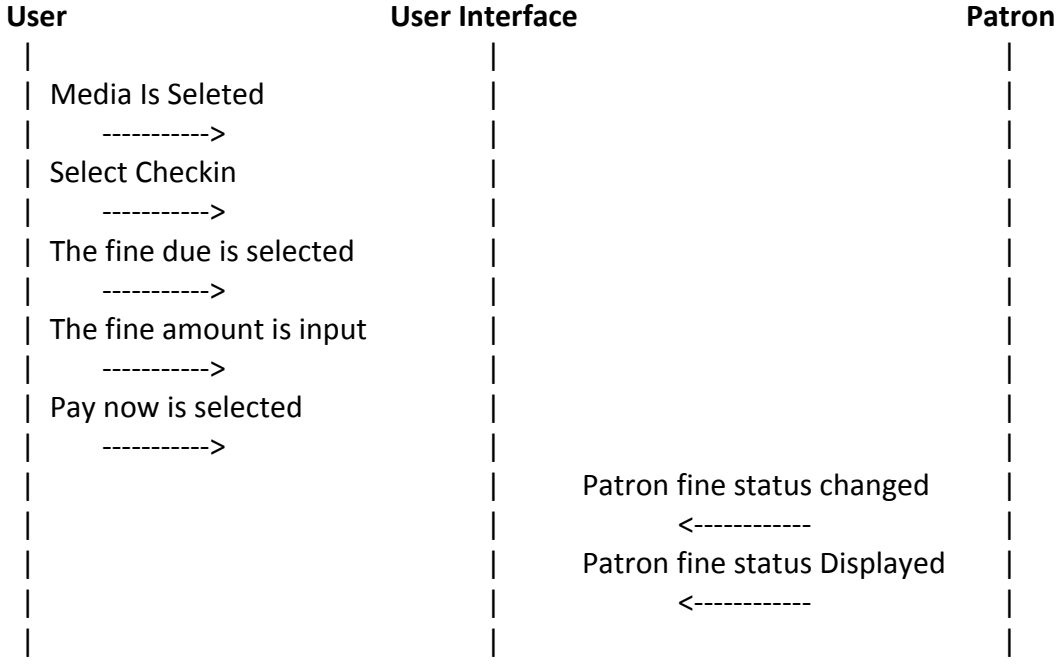
Scenario for Remove Damaged Media

-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

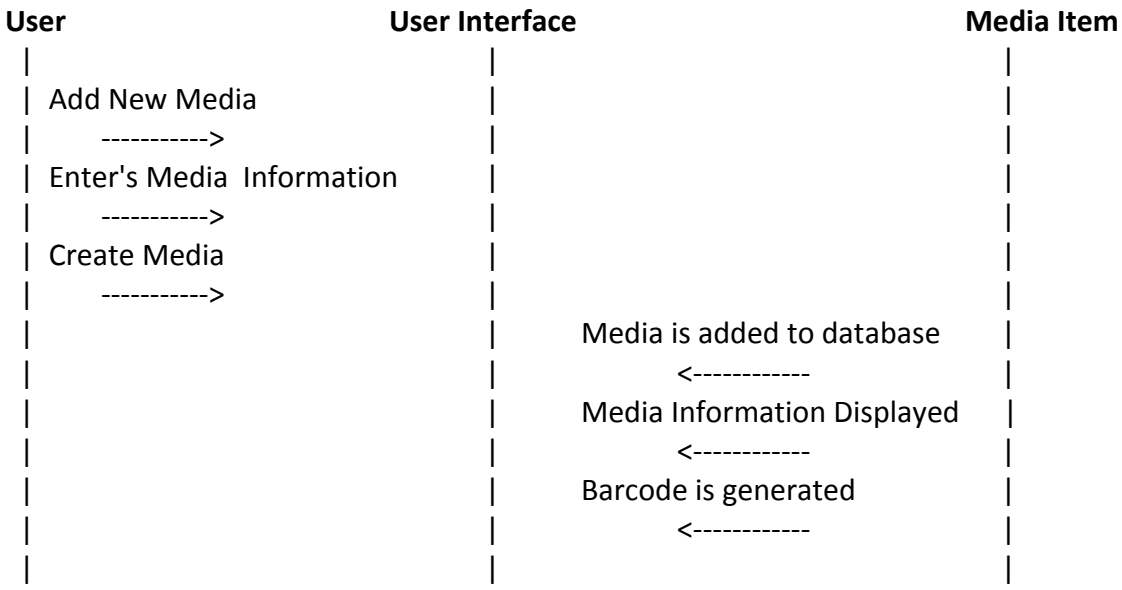


Scenario for Make a Payment

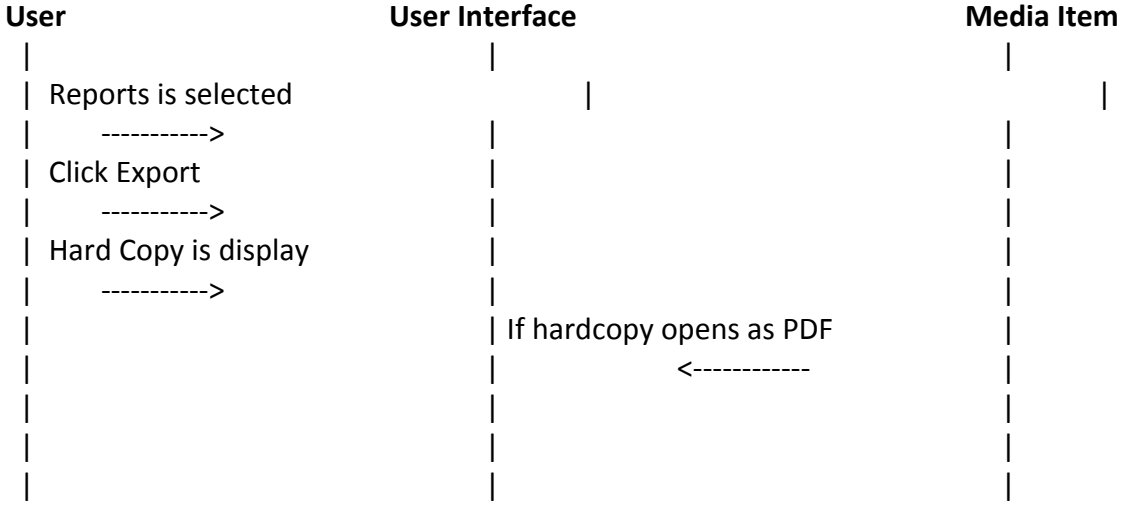
-A Patron has incurred a fine for either damage to an item or a late fee
-The Librarian or Library Manager searches for the media that is overdue.
-The Librarian or Library Manager selects the popup and enters in the payment type.
-The Librarian or Library Manager then inputs the amount to be paid
-The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



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

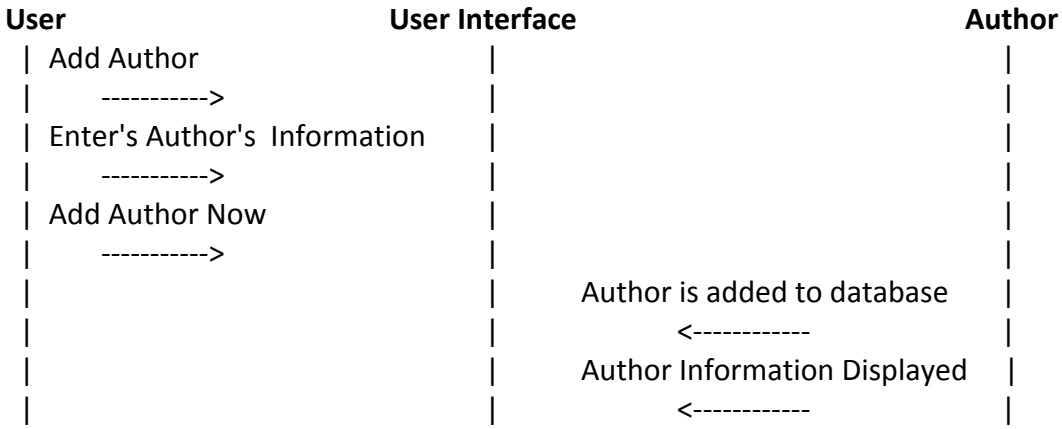


Scenario for Run Overdue Media Report
-A report is scheduled to be run over a certain time period
-A Librarian wishes to spot check with this report
-A superior wishes to have a copy of the current status



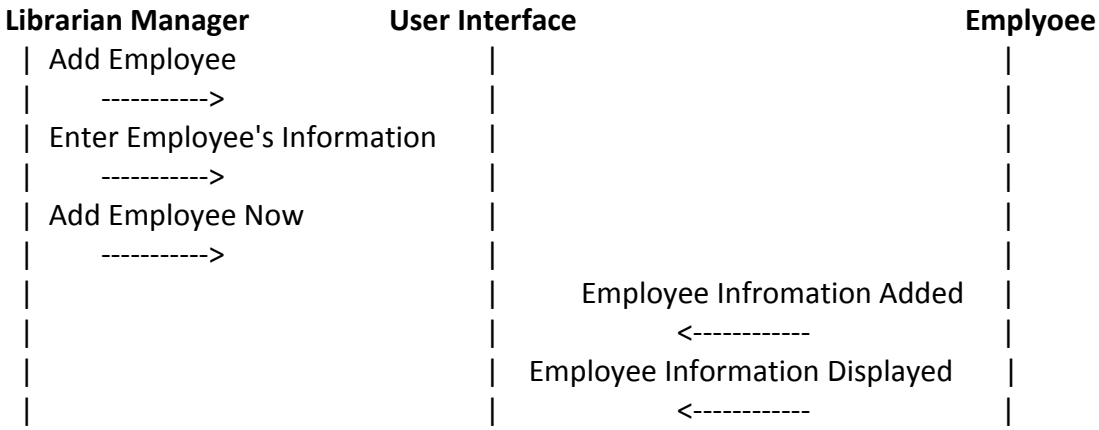
Scenario for Add Author

-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



Scenario for Add Employee

-An employee is hired

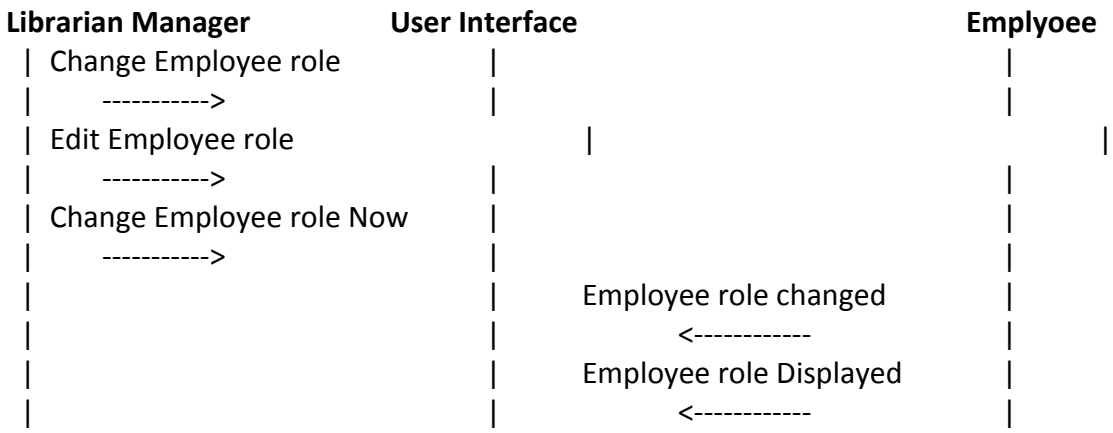


Scenario for Update Employee Role

-An emplyoee is promoted

-An emplyoee is demoted

-An emplyoee makes a lateral move in status

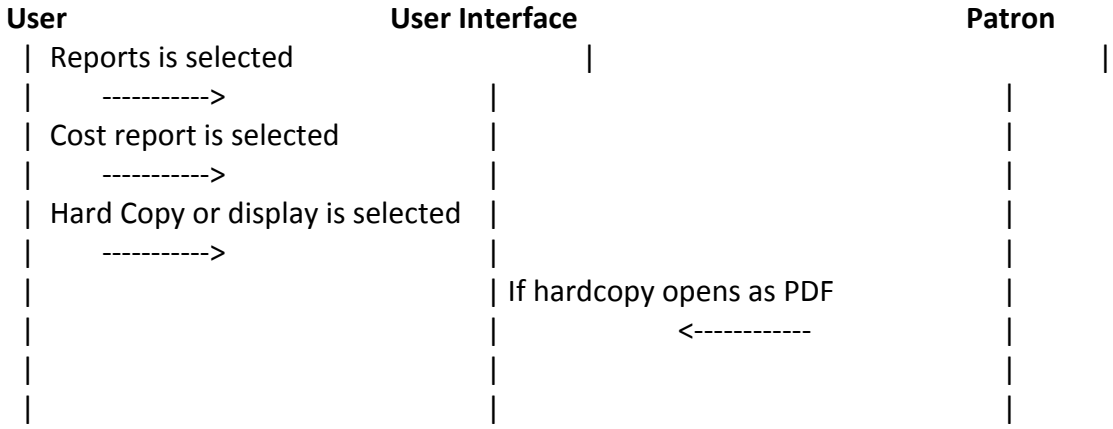


Scenario for Run Fee Report

-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



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

User	**User Interface**	**Media Item**
Reports is selected		
---------->		
Media Usage is selected		
---------->		
	If hardcopy opens as PDF	
	<----------	

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

User	**User Interface**	**Media Item**
Reports is selected		
---------->		
Checked Out is selected		
---------->		

		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 frame work.
        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

	Column Name	Data Type	Allow Nulls
PK	Fname	nvarchar(50)	☐
	Minit	nchar(1)	☐
PK	Lname	nvarchar(50)	☐
PK	DOB	date	☐

	Fname	Minit	Lname	DOB
1	Adams	Q	Douglas	1967-12-05
2	Asimov	Z	Isaac	1975-12-30
3	Card	S	Orson	1942-07-23
4	Firstname	M	Lastname	1960-04-12
5	George	J	Orwell	1930-06-05
6	Ira	T	Berkowitz	1954-04-12
7	J	J	Tolkein	1960-03-12
8	Nate	H	Priddy	1988-03-12

Check_Out

	Column Name	Data Type	Allow Nulls
PK	LibraryCardNumber	nvarchar(9)	☐
PK	BarCode	bigint	☐
PK	CheckOutDate	datetime	☐
	Renewed	bit	☑
	AmountPaid	money	☑
	Fee	money	☑
	ToBeReturnedDate	date	☑
	Checker#	nvarchar(9)	☐
	CheckedInDate	datetime	☑

	LibraryCardNumber	BarCode	CheckOutDate	Renewed	AmountPaid	Fee	ToBeReturnedDate	Checker#	CheckedInDate	CurrentCondition
1	100000001	51354873	2010-04-14 00:00:00.000	0	NULL	NULL	2010-05-06	001276439	2010-04-30 00:00:00.000	Good
2	100000001	51354873	2010-04-20 00:00:00.000	0	NULL	NULL	2010-04-30	001276439	2010-05-20 00:00:00.000	New-Good
3	100000001	51354873	2010-05-04 00:00:00.000	0	NULL	NULL	2010-05-18	001276439	NULL	Good
4	100000001	53245532	2009-11-29 00:00:00.000	0	NULL	NULL	2009-12-13	001276439	NULL	New
5	100000001	54897327	2010-04-01 00:00:00.000	0	5.50	NULL	2010-04-15	001276439	2010-05-06 00:00:00.000	Good-Pr
6	100000001	56478922	2010-04-07 00:00:00.000	0	NULL	NULL	2010-04-21	001276439	2010-05-07 00:00:00.000	New
7	100000001	59876249	2009-09-08 00:00:00.000	0	NULL	NULL	2009-09-22	001276439	2010-05-06 00:00:00.000	New-Good
8	100000005	51354873	2010-04-30 00:00:00.000	0	NULL	NULL	2010-05-14	001276439	2010-05-11 00:00:00.000	New-Good
9	100000006	51354873	2010-04-21 00:00:00.000	1	NULL	NULL	2010-05-12	001276439	2010-04-23 00:00:00.000	New
10	100000007	54879624	2010-05-05 00:00:00.000	0	NULL	NULL	2010-05-19	001276439	2010-05-27 00:00:00.000	New

Condition

	Column Name	Data Type	Allow Nulls
🔑	Grade	nvarchar(8)	☐
	Cost_Percentage	int	☐

	Grade	Cost_Percentage
1	Good	80
2	Good-Poo	70
3	New	100
4	New-Good	90
5	Poor	60
6	Replace	50

Employee

	Column Name	Data Type	Allow Nulls
🔑	EmplID	nvarchar(9)	☐
	SSN	nchar(9)	☐
	fName	nvarchar(20)	☐
	mName	nvarchar(20)	☑
	lName	nvarchar(20)	☐
	payRate	nvarchar(50)	☐
	DOB	datetime	☐
	Address	nvarchar(MAX)	☐
	phoneNumber	nvarchar(15)	☐
	roleType	nvarchar(20)	☐
	Eusername	nvarchar(20)	☐
	Epassword	nvarchar(20)	☐

	EmplID	SSN	fName	mName	lName	payRate	DOB	Address	phoneNumber	roleType	Eusername	Epassword
1	001276439	612094563	Isabella	Callie	Brady	1555	1970-04-15 00:00:00.000	5637 Erat. Av. Chico, CA 95928	5308982134	Library Manager	1_Brady	12938
2	002948173	614112819	Amy	Kristen	Swanson	8.50	1988-04-18 00:00:00.000	3521 Iaculis St. Chico, CA 95926	4355849047	Librarian	1_Swanson	23232
3	003923818	991731739	Louis	Anne	McDaniel	9.00	1967-08-19 00:00:00.000	3826 Blandit Rd. Chico, CA 95973	5305219481	Librarian	1_McDaniel	29229
4	007036383	925460932	Leon	Timothy	Maite	12.0	1986-03-07 00:00:00.000	Ap #817-7435 Massa. Avenue Chico, CA 95973	5303439182	Librarian	1_Maite	93272
5	008321876	614772341	Brady	Jay	Stevens	8.50	1989-10-03 00:00:00.000	2057 Fringilla St. Chico, CA 95926	5308987634	Librarian	1_Stevens	01938
6	009834521	983452156	Maxwell	Benjamin	Judith	10.2	1975-09-05 00:00:00.000	283-961 Ante Street Paradise, CA 95967	5303453214	Librarian	1_Judith	67890
7	009845643	994532165	Whitley	Alika	Riley	9.00	1987-07-09 00:00:00.000	Ap #346-5948 Libero Avenue Chico, CA 95928	8792005252	Librarian	1_Riley	12345

Media

	Column Name	Data Type	Allow Nulls
🔑	Barcode	bigint	☐
	Title	nvarchar(100)	☐
	ISBN	int	☐
	Year	datetime	☑
	Publisher	nvarchar(100)	☑
	Type	nvarchar(50)	☐
	Subject	nvarchar(100)	☐
	OrderCost	int	☐
	InitialCondition	nvarchar(50)	☐
	Grade	nvarchar(8)	☐

	Barcode	Title	ISBN	Year	Publisher	Type	Subject	OrderCost	InitialCondition
1	51354873	Animal Farm-Nineteen Eighty-Four	184046245	2003-01-01 00:00:00.000	Palgrave macmillan	DVD	Palgrave macmillan	27	New
2	53245532	BigFish	200993845	2010-04-05 00:00:00.000	Harding	Book	Fish	25	New
3	54879624	Ender's Game	812550706	1994-01-01 00:00:00.000	Tor Science Fiction	Book	Space	8	New
4	54897327	Foundation	553293354	1991-01-01 00:00:00.000	Speectra	Book	Foundation	8	New
5	56478922	Family Matters	932112448	2006-01-01 00:00:00.000	Justin, Charles & Co	Book	Family	16	New
6	59876249	Restaurant at the End of the Universe	345391810	1995-01-01 00:00:00.000	Ballantine Publishing Group	Book	HitchHicker	8	New

Media_Author

	Column Name	Data Type	Allow Nulls
🔑	Barcode	bigint	☐
🔑	AuthFName	nvarchar(50)	☐
🔑	AuthLName	nvarchar(50)	☐
🔑	AuthDOB	date	☐

	Barcode	AuthFName	AuthLName	AuthDOB
1	51354873	Adams	Douglas	1967-12-05
2	51354873	George	Orwell	1930-06-05
3	53245532	J	Tolkein	1960-03-12
4	54879624	Card	Orson	1942-07-23
5	54897327	Asimov	Isaac	1975-12-30
6	56478922	Ira	Berkowitz	1954-04-12
7	59876249	Adams	Douglas	1967-12-05

Patron

Column Name	Data Type	Allow Nulls
libraryCardNumber	nvarchar(9)	☐
fName	nvarchar(50)	☐
mName	nvarchar(20)	☑
lName	nvarchar(50)	☐
DOB	date	☐
phoneNumber	nvarchar(20)	☐
Address	nvarchar(MAX)	☑
Pusername	nvarchar(20)	☐
Ppassword	nvarchar(20)	☐

	libraryCardNumber	fName	mName	lName	DOB	phoneNumber	Address	Pusername	Ppassword
1	100000001	Quin	Kevin	Ellison	1989-07-08	7071241241	Ap #390-1150 Ornare. Av. Chico, CA 95973	qellison	13832
2	100000002	Amy	Kristen	Swanson	1988-04-18	4355849047	3521 Iaculis St. Chico, CA 95926	aswanson	34233
3	100000003	Susan	Eden	Bradford	1979-12-03	5303431245	9260 Lectus Rd. Chico, CA 95928	sbradford	1323v
4	100000004	Vicky	Marie	Holcomb	1988-07-07	5305210666	400 W. 1st St Chico, CA 95929	vholcomb	vhvhv
5	100000005	Matthew	NULL	Harris	1983-01-01	7071717171	400 W. 1st St Chico, CA 959529	mharris	mhmhm
6	100000006	Joseph	NULL	Fitzpatrick	1973-01-01	5301011241	400 W. 1st St Chico, CA 95929	jfitzpatrick	jfjfj
7	100000007	Nate	NULL	Priddy	1988-01-01	5301238712	400 W. 1st St Chico, CA 95929	npriddy	npnpn

Policies

Column Name	Data Type	Allow Nulls
Description	nvarchar(50)	☐
Value	decimal(18, 2)	☐

	Description	Value
1	CheckOutLength	14.00
2	FeePerDay	0.25

Reservation

Column Name	Data Type	Allow Nulls
LibraryCardNumber	nvarchar(9)	☐
BarCode	bigint	☐
DateRequested	datetime	☐
CheckOutStatus	bit	☐
ReshelfDate	datetime	☐

	LibraryCardNu...	BarCode	DateRequested	CheckOutStatus	ReshelfDate
	100000001	54879624	2010-03-12 00:...	True	2010-03-16 00:...
	100000001	54879624	2010-05-08 00:...	False	2010-05-12 00:...

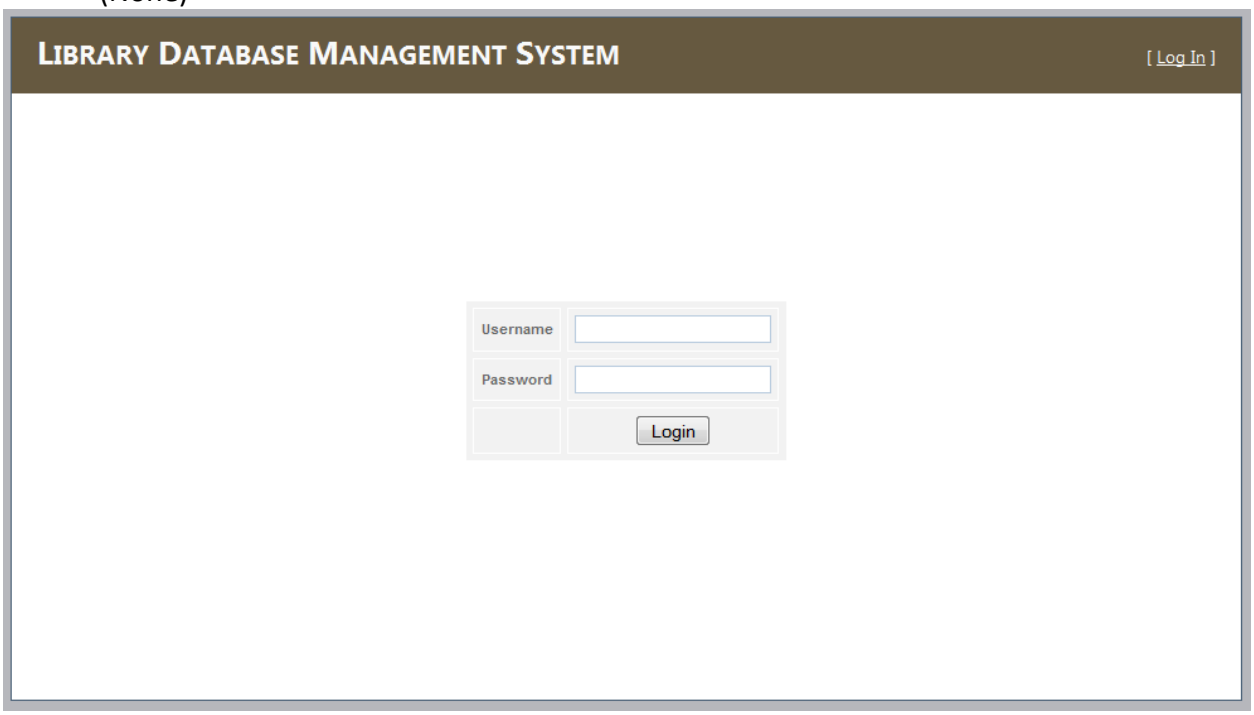
User Views - Forms

*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 npriddy. 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

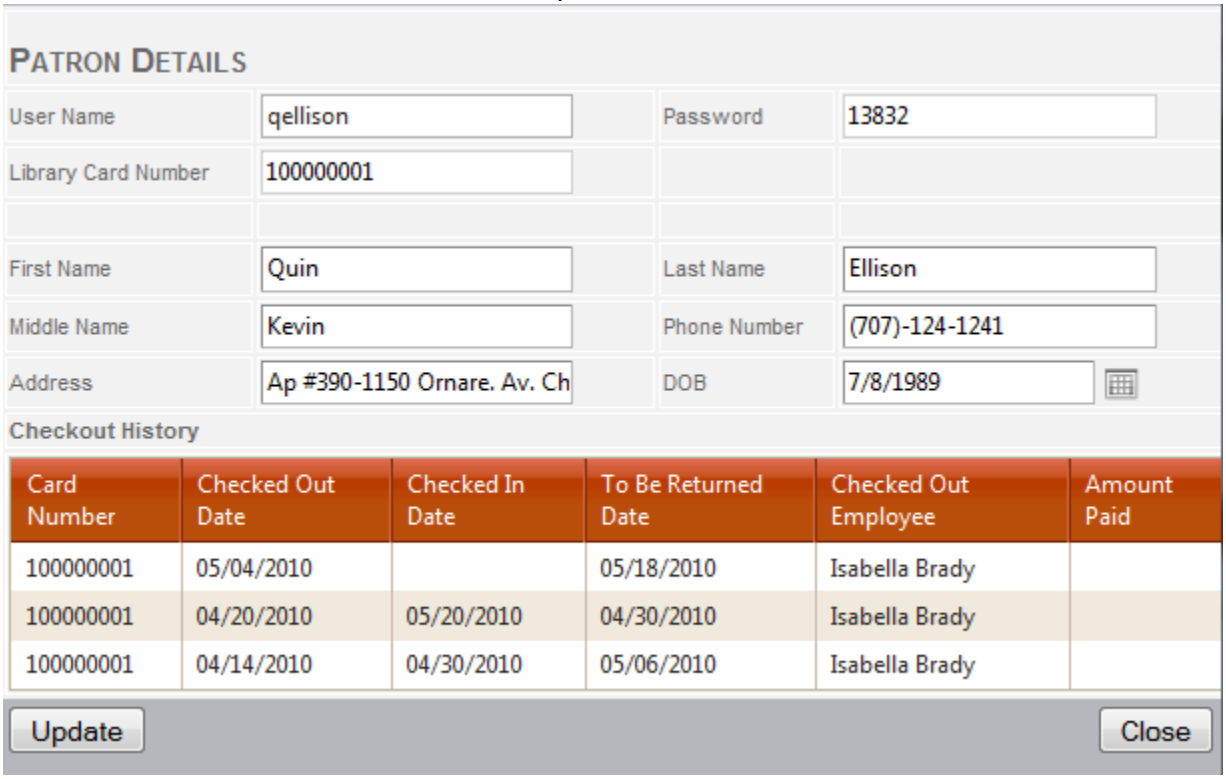
#2, 3 Use case: Add Patron/Update User Information

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

User Name	qellison	Password	13832
Library Card Number	100000001		
First Name	Quin	Last Name	Ellison
Middle Name	Kevin	Phone Number	(707)-124-1241
Address	Ap #390-1150 Ornare. Av. Ch	DOB	7/8/1989

Checkout History

Card Number	Checked Out Date	Checked In Date	To Be Returned Date	Checked Out Employee	Amount Paid
100000001	05/04/2010		05/18/2010	Isabella Brady	
100000001	04/20/2010	05/20/2010	04/30/2010	Isabella Brady	
100000001	04/14/2010	04/30/2010	05/06/2010	Isabella Brady	

Update Close

Form #3

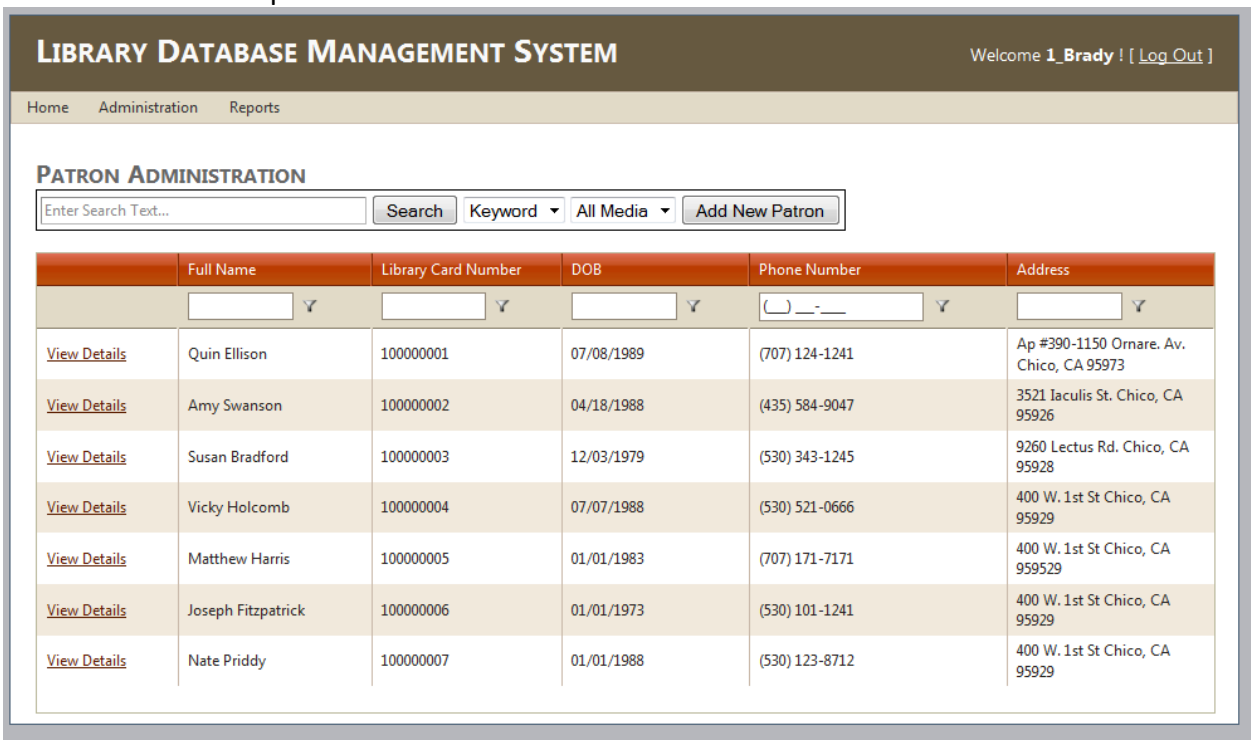
#3b Use case: View Patron Details

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.

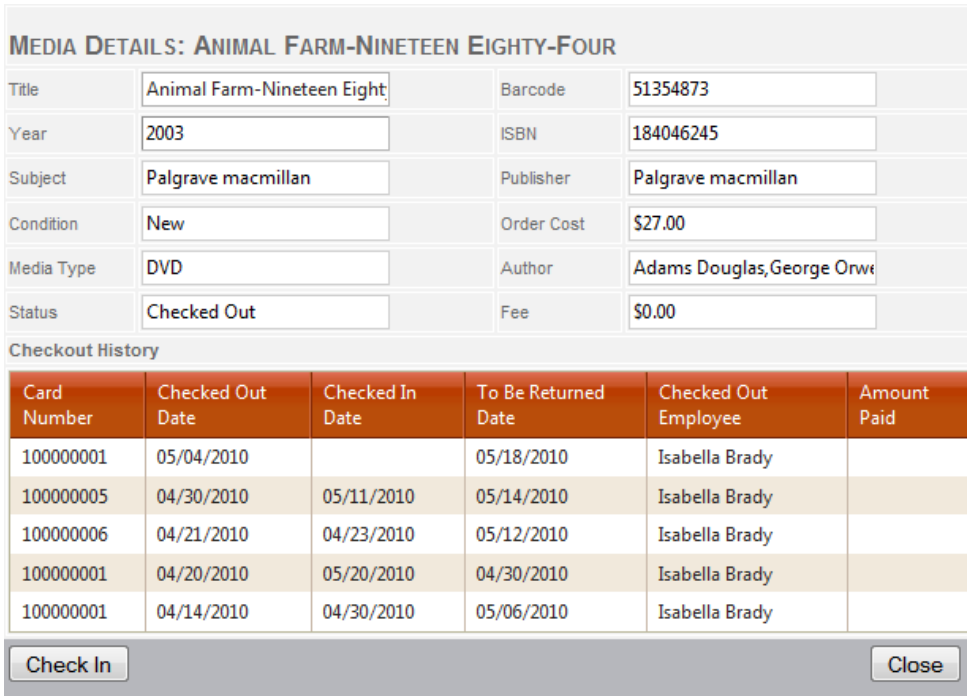


Form #4
#4 Use case: View Media Details / Renew
From use case(s) 7
Actor(s): Patron, Librarian, Library Manager

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**



Form #5

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

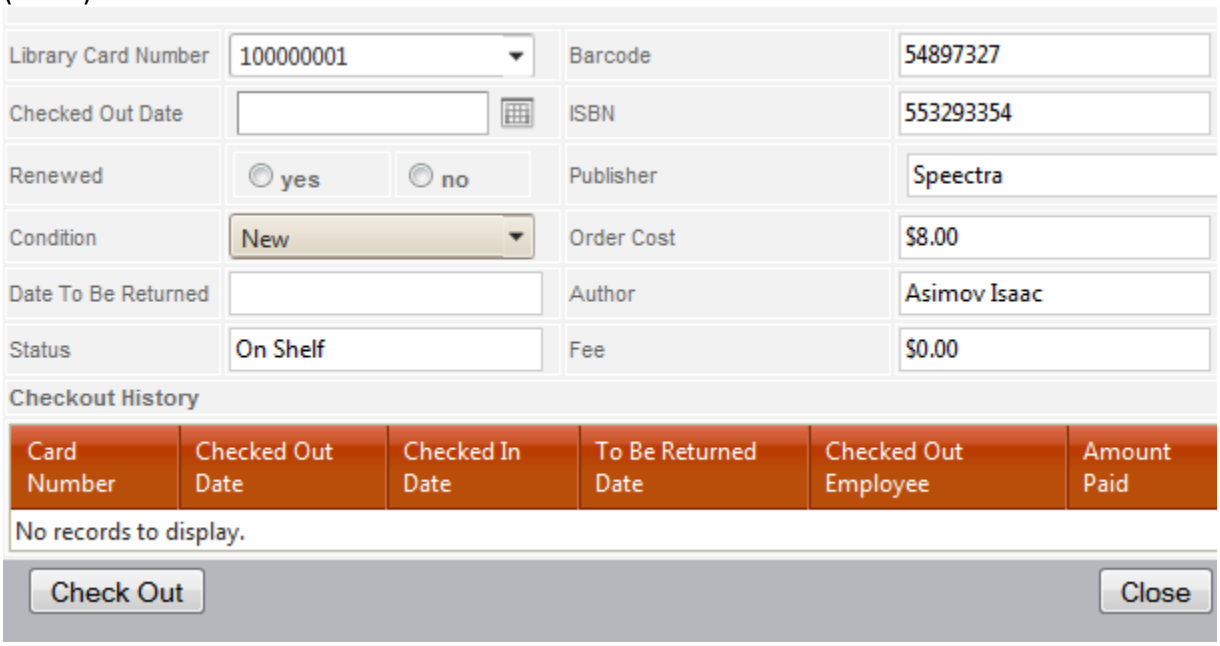
Actor(s): Librarian, Library Manager

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

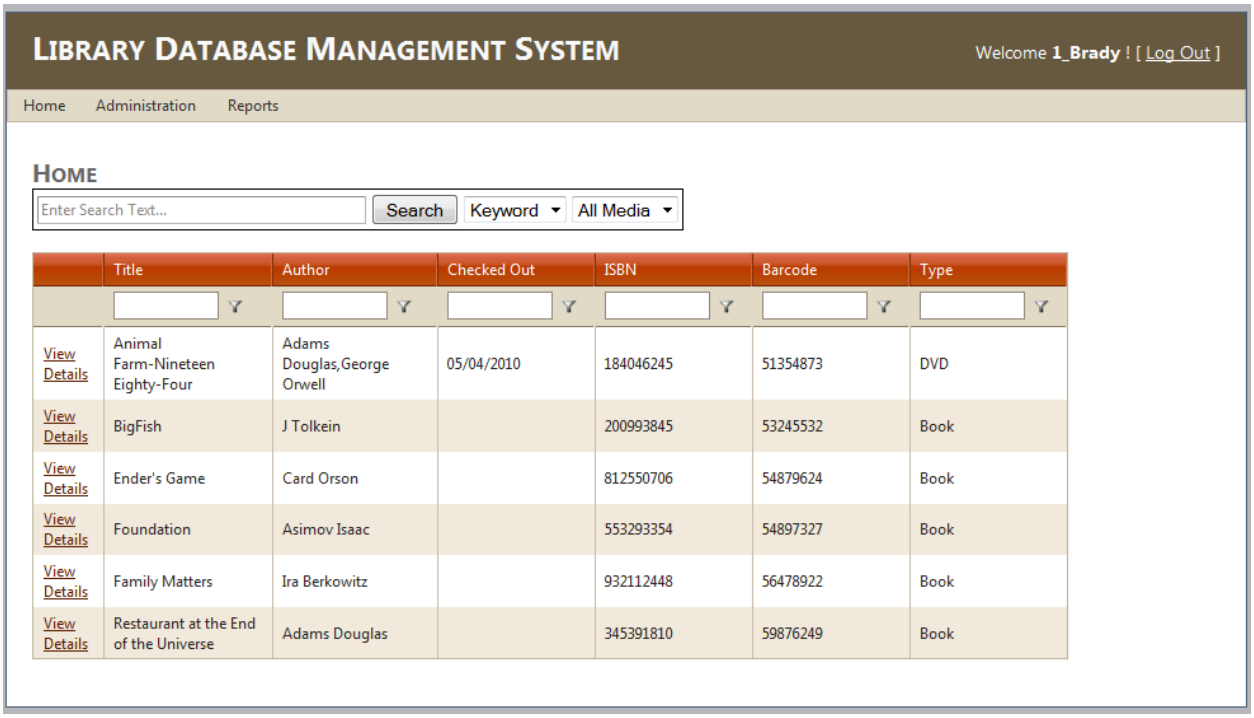
#8 Use case: Search for Media

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



Form #7

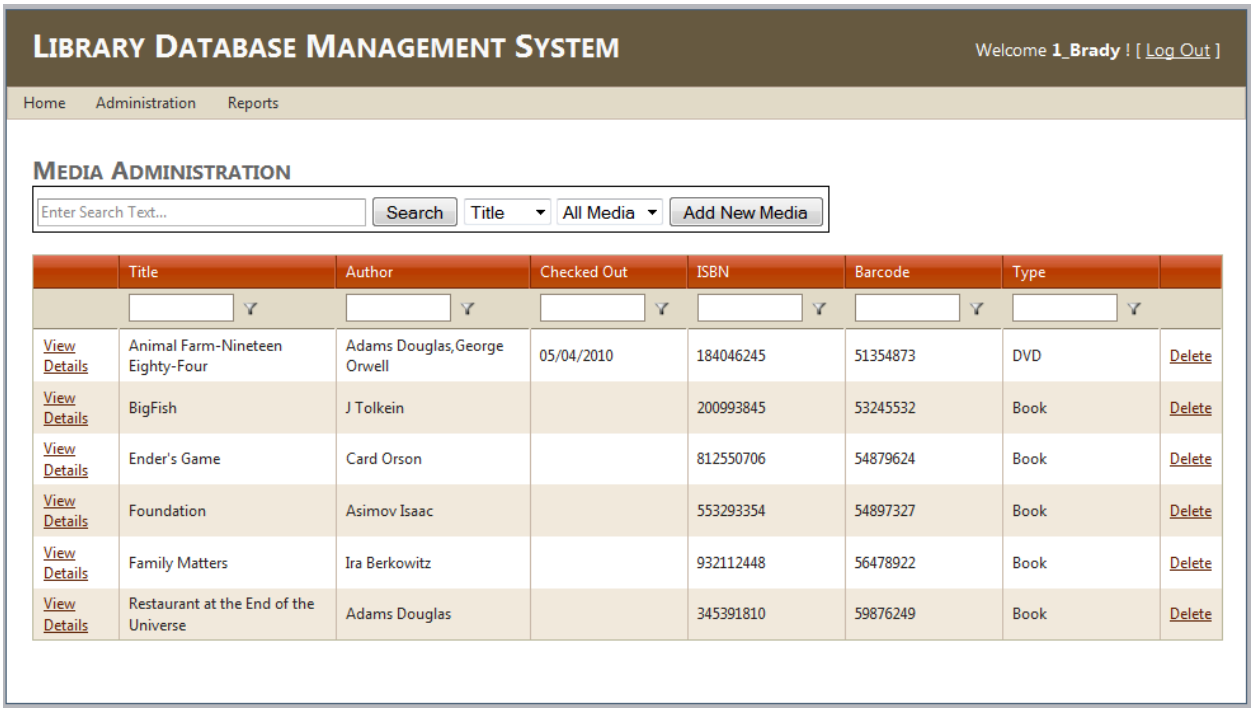
#9 Use case: Search for Media / Administration

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.



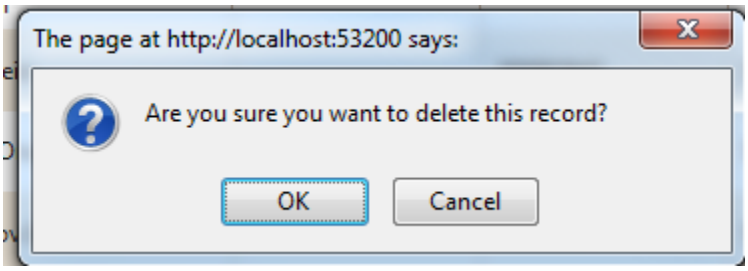
Form #8

#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

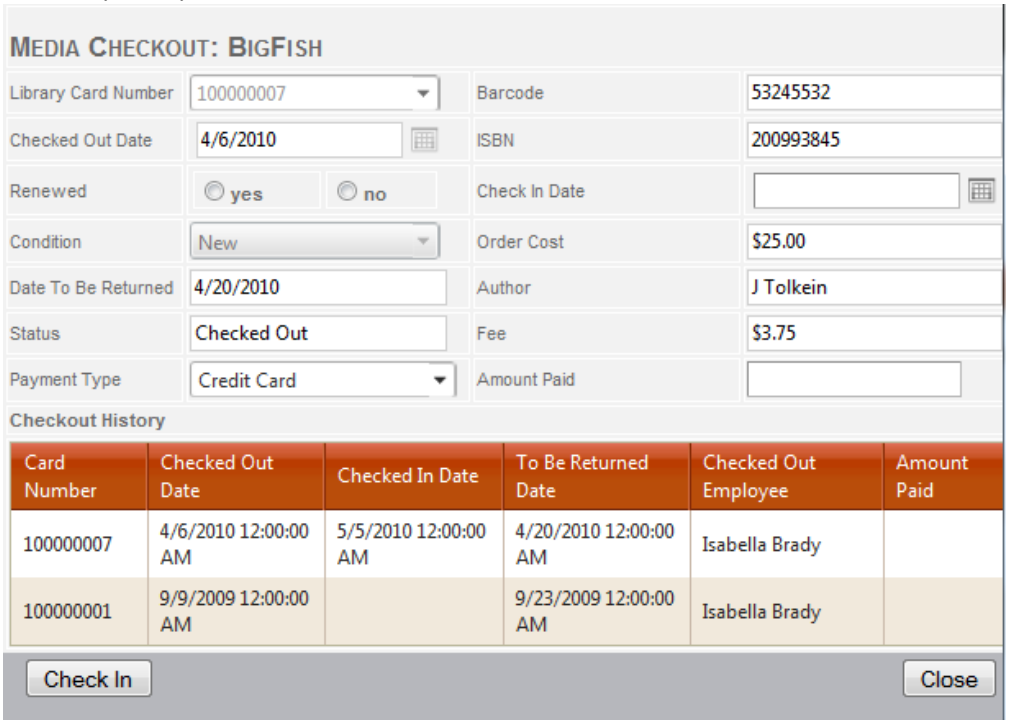
#10 Use case: Make Payment

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)

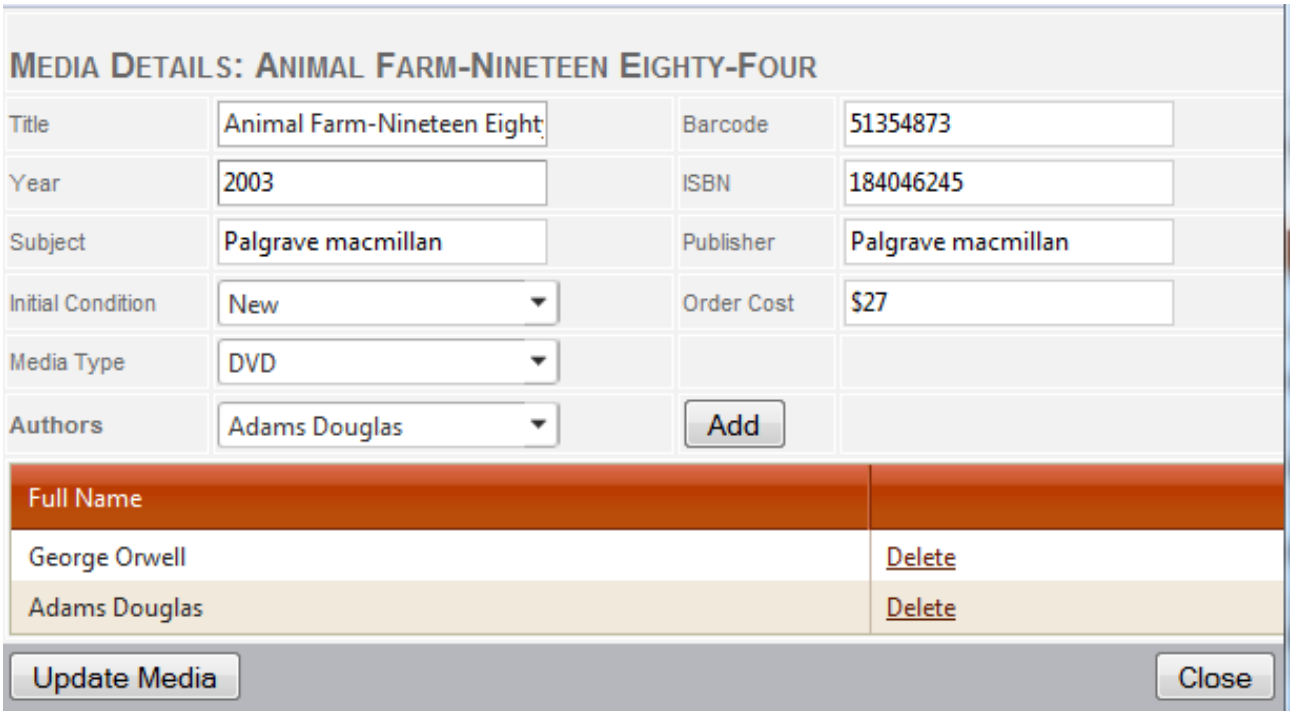


Form #10
#11 Use case: Add Media
Actor(s): Librarian, Library Manager

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.

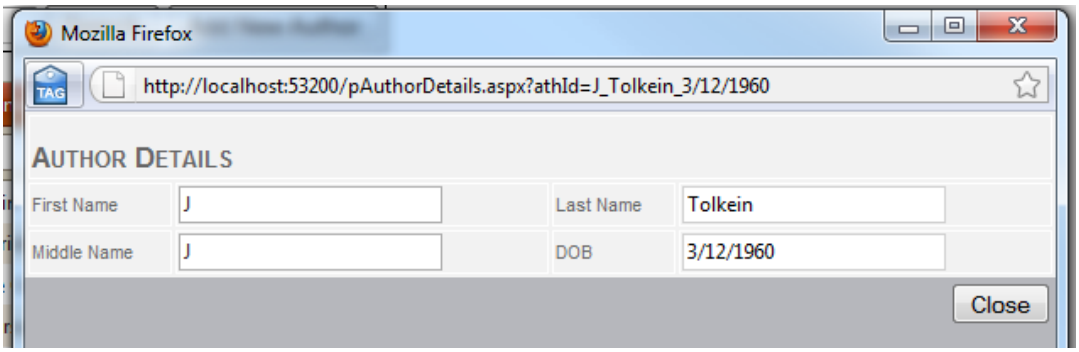


Form #11
#13 Use case: Add / Update Author
Actor(s): Librarian, Library Manager

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.

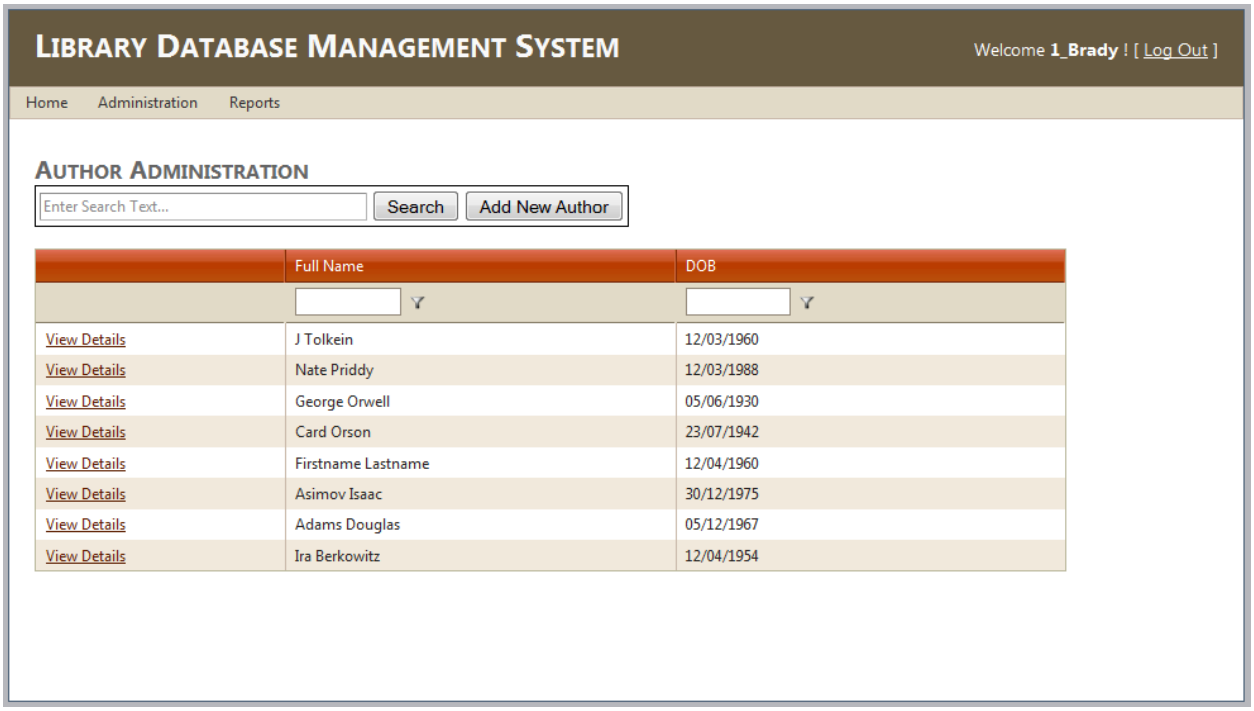


Form #12
#13b Use case: View Authors Details
Actor(s): Librarian, Library Manager

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

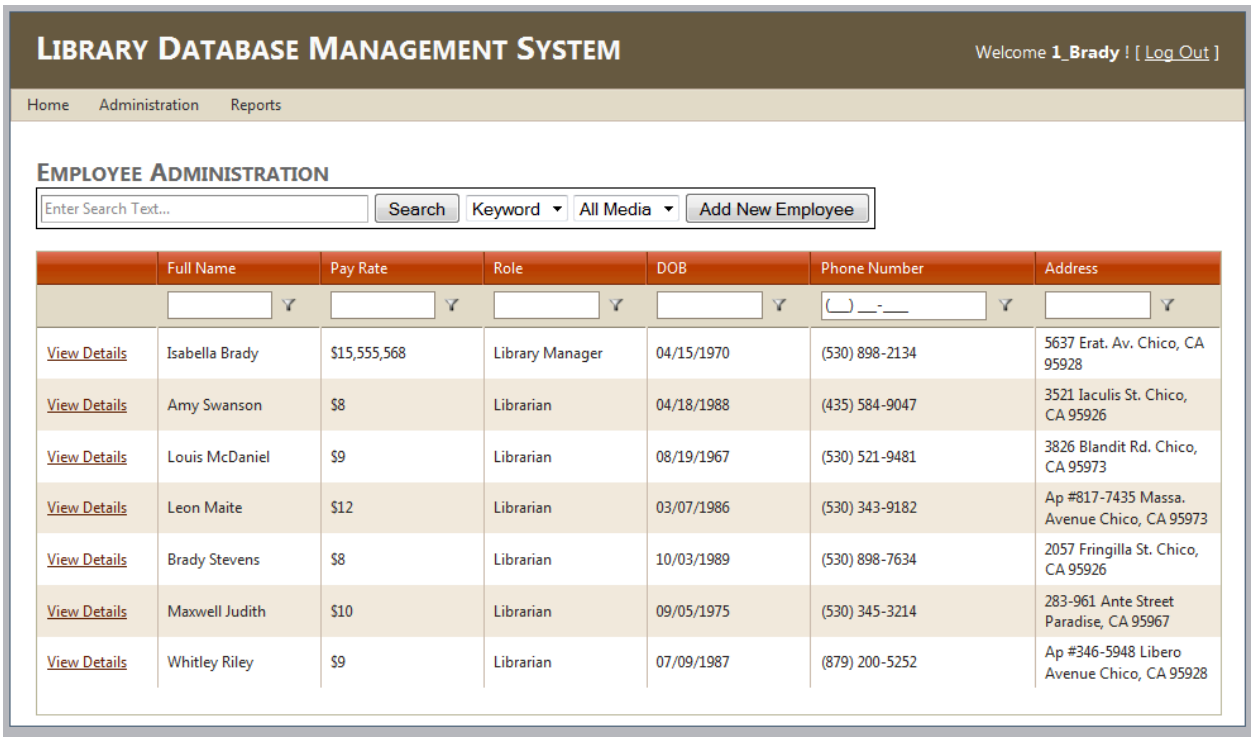
From use case(s) 7

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.

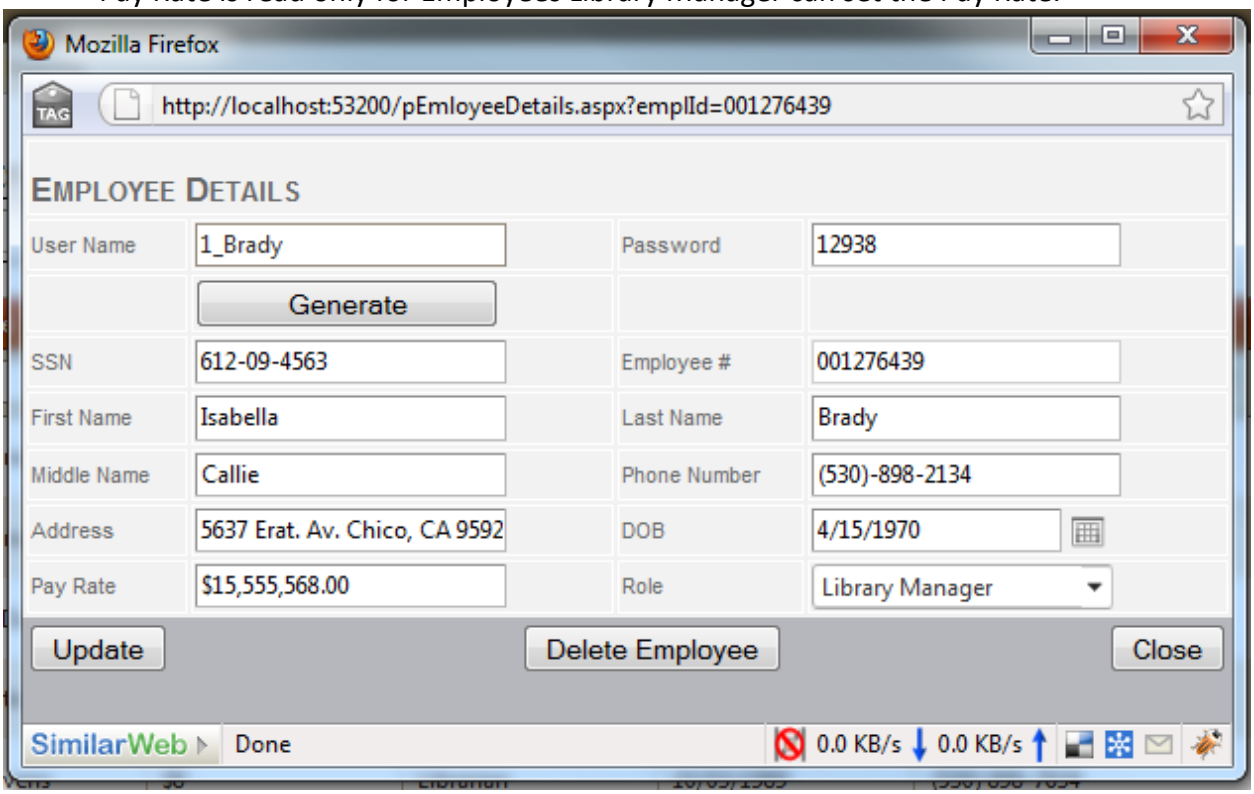


Form #14
#15 Use case: Add/Update/Remove Employee
Actor(s): Librarian, Library Manager

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

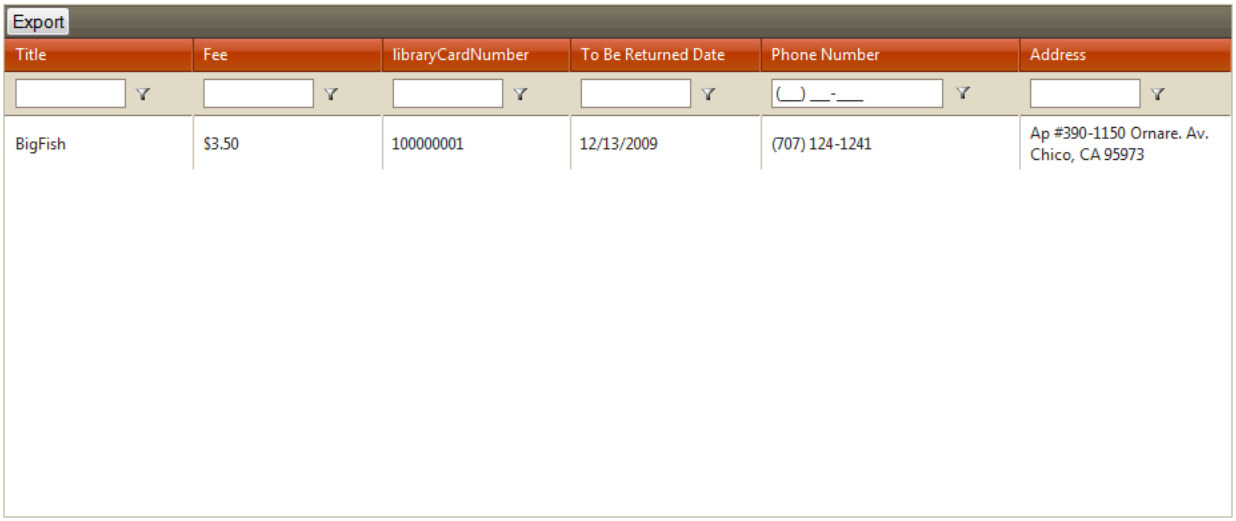
Report #1

#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.

FEE REPORT REPORT



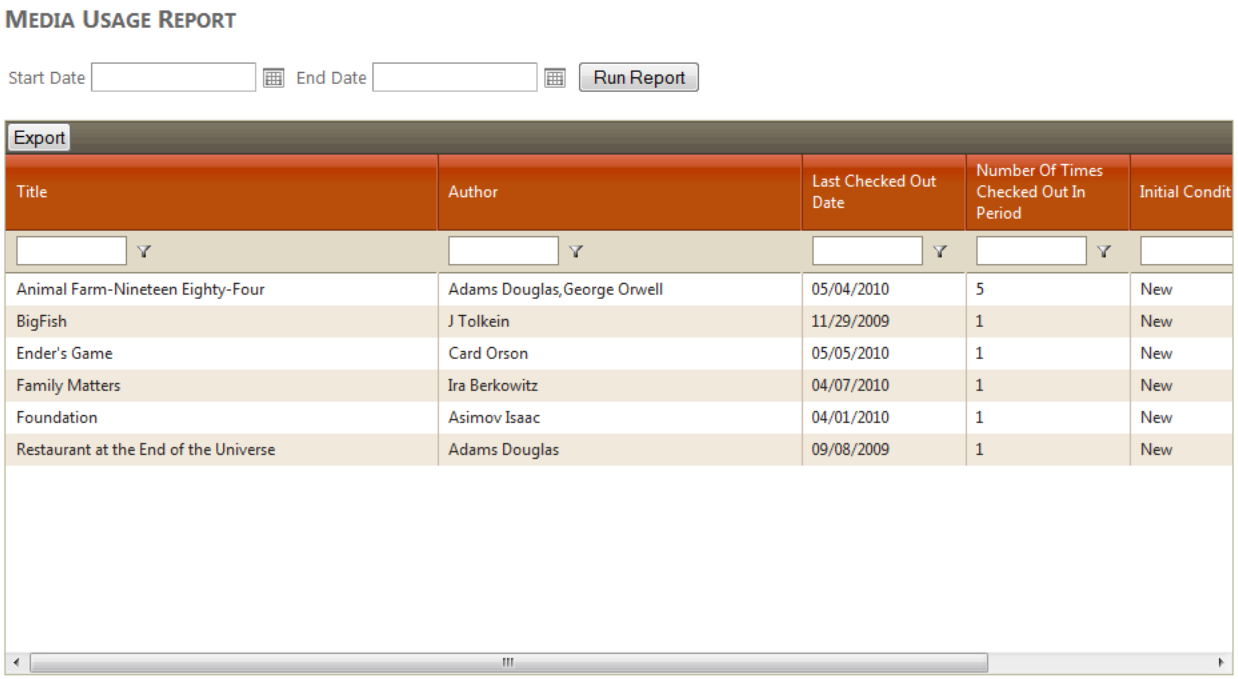
Title	Fee	libraryCardNumber	To Be Returned Date	Phone Number	Address
BigFish	$3.50	100000001	12/13/2009	(707) 124-1241	Ap #390-1150 Ornare. Av. Chico, CA 95973

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.



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

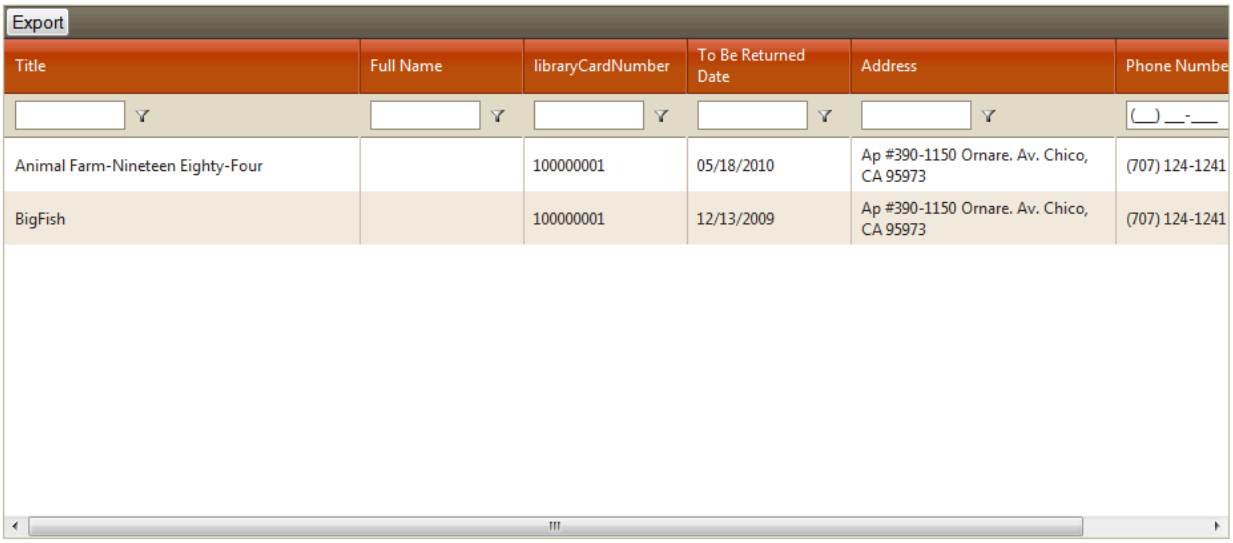
Report #3

#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.

CHECK OUT REPORT



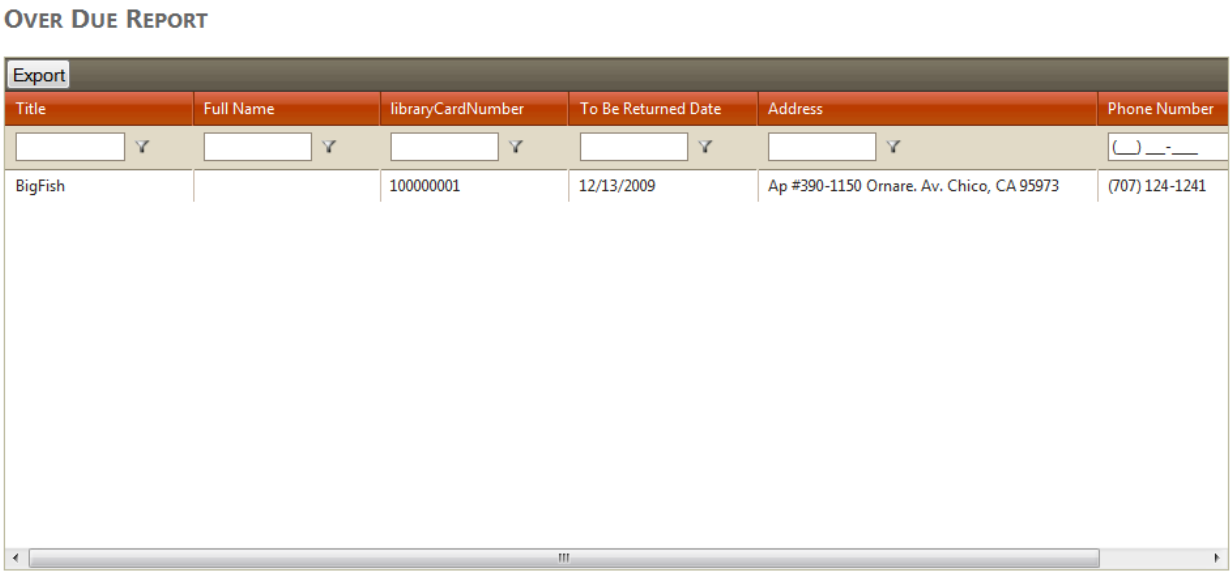
Title	Full Name	libraryCardNumber	To Be Returned Date	Address	Phone Number
Animal Farm-Nineteen Eighty-Four		100000001	05/18/2010	Ap #390-1150 Ornare. Av. Chico, CA 95973	(707) 124-1241
BigFish		100000001	12/13/2009	Ap #390-1150 Ornare. Av. Chico, CA 95973	(707) 124-1241

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.



Title	Full Name	libraryCardNumber	To Be Returned Date	Address	Phone Number
BigFish		100000001	12/13/2009	Ap #390-1150 Ornare. Av. Chico, CA 95973	(707) 124-1241

SQL Statements

Table Creation

```
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
(
	[Fname] ASC,
	[Lname] ASC,
	[DOB] 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].[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_1] PRIMARY KEY CLUSTERED
(
	[LibraryCardNumber] ASC,
	[BarCode] ASC,
	[CheckOutDate] 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].[Check_Out]  WITH CHECK ADD  CONSTRAINT [FK_Check_Out_Employee]
FOREIGN KEY([Checker#])
REFERENCES [dbo].[Employee] ([EmplID])
```

```
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,
	[lName] [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,
	[Eusername] [nvarchar](20) NOT NULL,
	[Epassword] [nvarchar](20) NOT NULL,
```

```
 CONSTRAINT [PK_Employee] PRIMARY KEY CLUSTERED
(
	[EmplID] 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,
	[AuthFName] [nvarchar](50) NOT NULL,
	[AuthLName] [nvarchar](50) NOT NULL,
	[AuthDOB] [date] NOT NULL,
 CONSTRAINT [PK_Media_Author] PRIMARY KEY CLUSTERED
(
	[Barcode] ASC,
	[AuthFName] ASC,
```

```
	[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,
	[Pusername] [nvarchar](20) NOT NULL,
	[Ppassword] [nvarchar](20) NOT NULL,
 CONSTRAINT [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 'Delete'
	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].[Employe_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)
```

```
        END) as phoneNumber



        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) + '-' + 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 + ' ' + 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
                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 + ' ' + p.lName as 'fullName', p.libraryCardNumber, c.ToBeReturnedDate,
        (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,
        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')
        END

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

        (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 Manger 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