Deian St. Mishev
№: F84396
Course №: INFM313

Information System Development in .Net

FM84396_LibraryCMS

Documentation

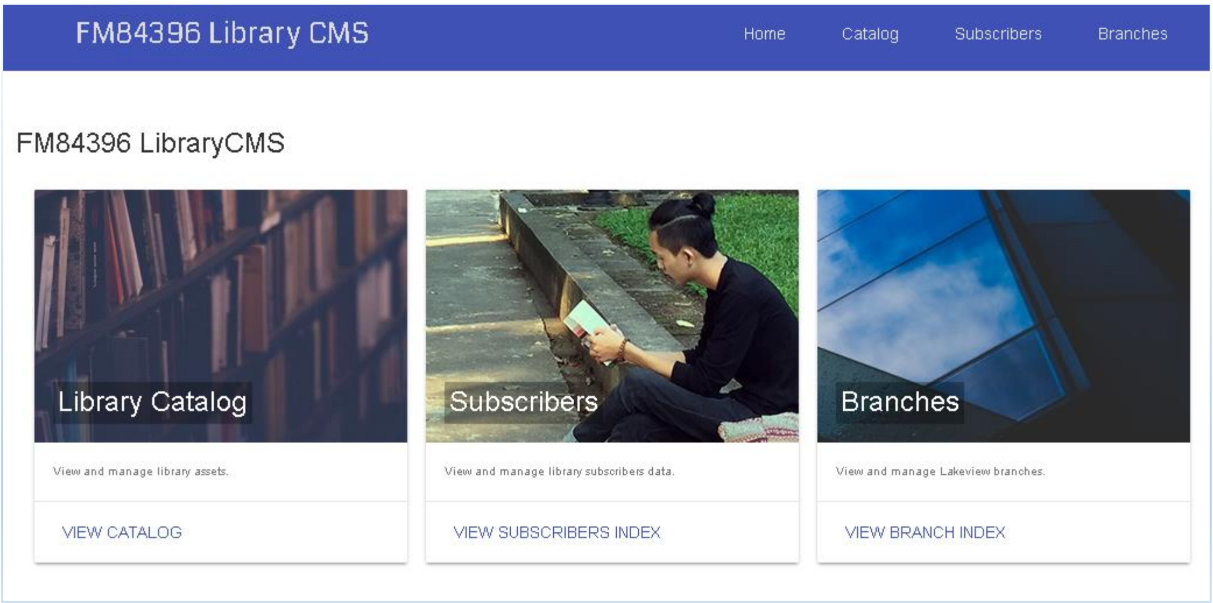
Prerequisite.

The following project is a lightweight Library MVC Management System written with .ASP .NET Core 1.0. The following document outlines the Functionality, Database and User Workflow for the application. The documentation for the code itself is written inline in the code.

Functionality and User Workflow:

The application functions as a basic Library Management System which stores assets (namely Books, Videos, Magazines and Newspapers) to lend to the patrons of the Library. The practical functionality handles most of the aspect of the frontend (user side, not admin side) of a production grade application besides the ability to persist the user through a login / subscription system. De facto the developed application both functionality wise and code-wise can be separated into the following components:

Home:



FM84396 LibraryCMS

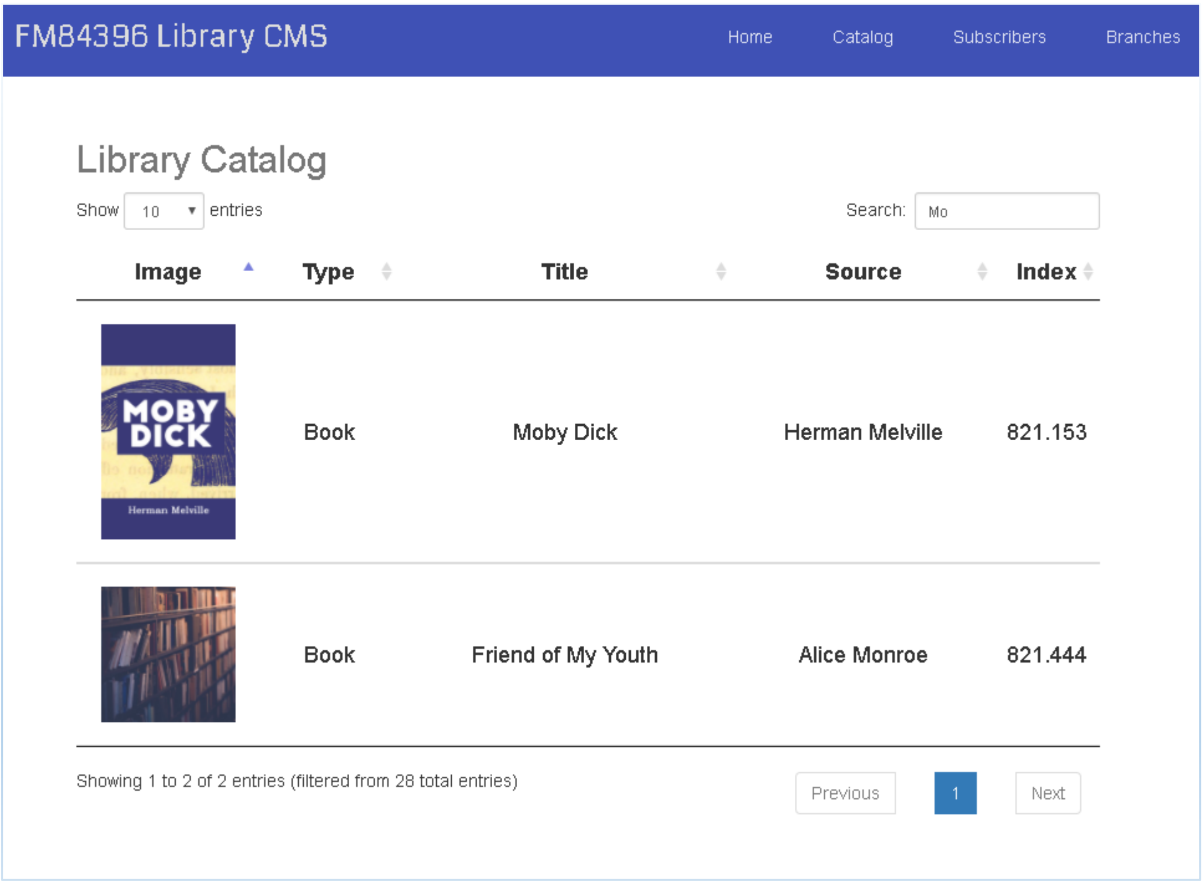
View and manage library assets.

View and manage library subscribers data.

View and manage Lakeview branches.

This component covers navigation between the other components of the application. It introduces the user to both the content of the application as well as with the navigation bar on top, which also covers nav. to other components.

Catalog:



Library Catalog

Show 10 entries

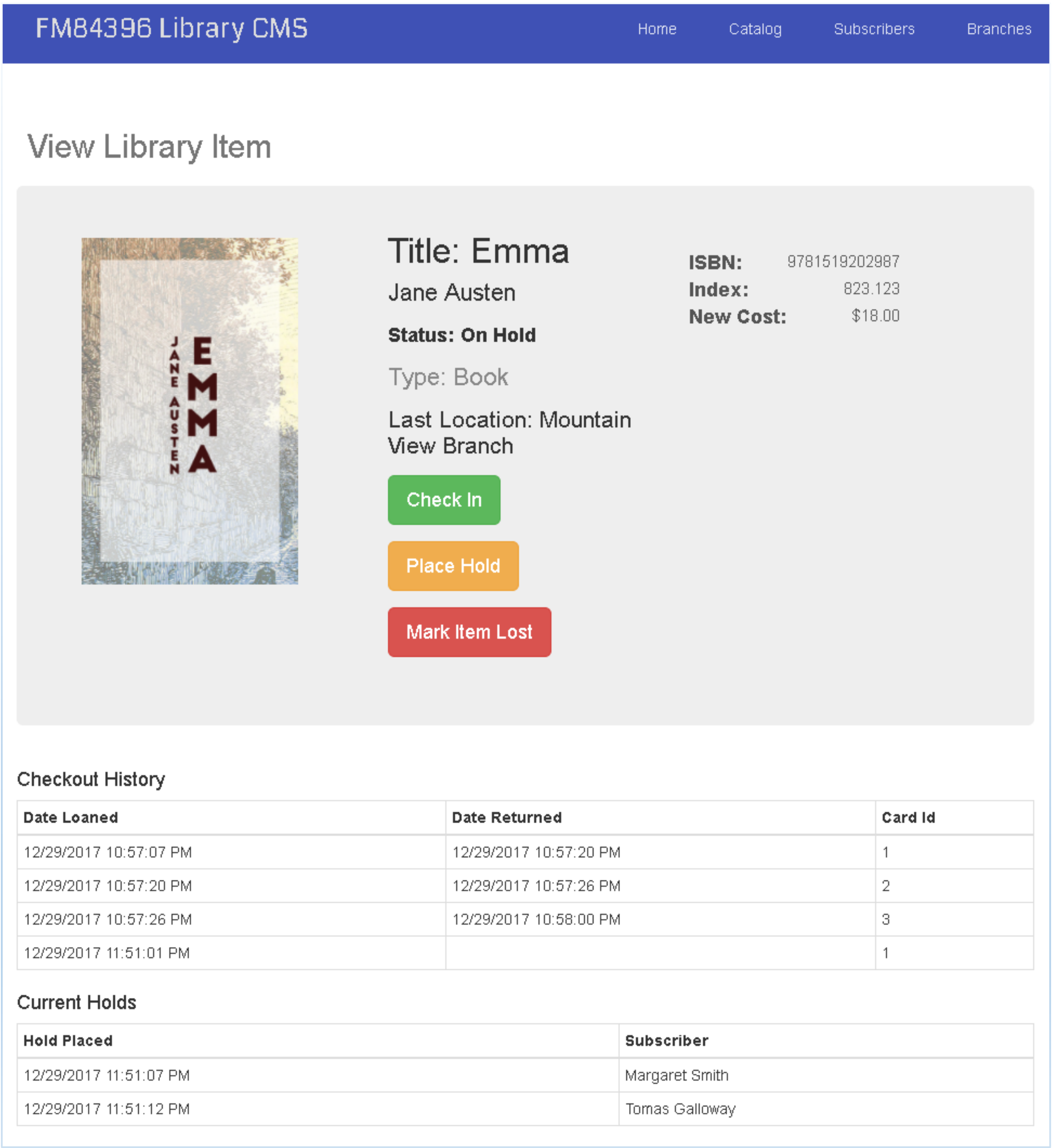
Search: Mo

Image	Type	Title	Source	Index
	Book	Moby Dick	Herman Melville	821.153
	Book	Friend of My Youth	Alice Monroe	821.444

Showing 1 to 2 of 2 entries (filtered from 28 total entries)

The catalogue view allows the user to browse between all the entries in the database of the library. This is where the user can quickly scan for the item he requires and also (basically) sort through the proposition. Sorting is achieved with Jquery Datatables as well as the choice of how many the maximum displayed entries can be. For the assets themselves here you see the Cover, Type, Title, Source (author, director, publisher, agency all depending on type), and the Library Index of the item. In future, the search and list functionality should both be better handled. A fitted solution is required in order to not alienate people who are not used to a “grab and go” mentality. If the user presses on any entry in the list he is redirected to the Assets detail page:

Assets detail page:



View Library Item

Title: Emma

Jane Austen

Status: On Hold

Type: Book

Last Location: Mountain View Branch

ISBN:	9781519202987
Index:	823.123
New Cost:	$18.00

Checkout History

Date Loaned	Date Returned	Card Id
12/29/2017 10:57:07 PM	12/29/2017 10:57:20 PM	1
12/29/2017 10:57:20 PM	12/29/2017 10:57:26 PM	2
12/29/2017 10:57:26 PM	12/29/2017 10:58:00 PM	3
12/29/2017 11:51:01 PM		1

Current Holds

Hold Placed	Subscriber
12/29/2017 11:51:07 PM	Margaret Smith
12/29/2017 11:51:12 PM	Tomas Galloway

The assets detail page consists of the Assets Status, Assets Checkout History and Assets Current Holds.

The **Assets Checkout History** is simply a way to keep track of the checkout history for the item. In the case that the last Date Returned is not filled (like in the figure above) the element is not yet checked in.

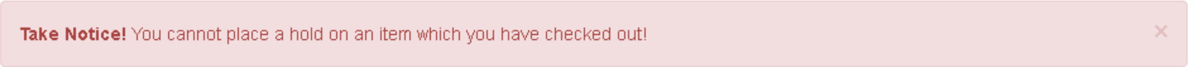
The **Assets Current Holds** is the view where the currently waiting to check out the item users are listed. The order of the subscribers listed is the same as the order in which the asset will be checked out for after returned by the current subscriber. This happens automatically on check in.

The **Assets Status** view shown here is the view with the most functionality here. This is where the Title, ISBN(if a book), Index, Cost to Replace, Last Location, Type and Author/Director…are displayed.

This view also allows to aider check in, check out, place hold (in the queue) or mark the element as lost (currently a hack to remove all Checkout and Holds history).

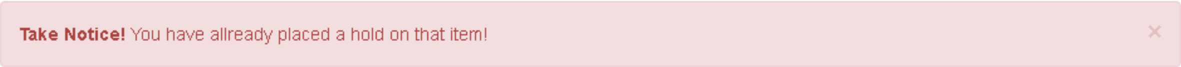
To better understand the above mentioned functionality I we will focus on the status of the assets (On Hold in the figure). When an element is not Checked it's status is "Available" and it can be only checked out (and saying this I mean there is only the "checkout" button in the UI). When a user wants to check out an item he can do that by providing his library card number. After checked out and the entry is put as "open ended" in the checkout history and the status of the item is changed to "Checked Out".

Now if another user comes by the item and he sees that it is checked out he can place a hold with his Library Card Number. He will be denied one in one of the three scenarios.



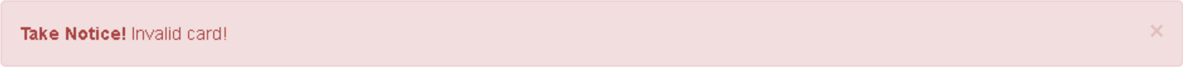
Take Notice! You cannot place a hold on an item which you have checked out!

1. He is the same user which is has the item on checkout.



Take Notice! You have allready placed a hold on that item!

2. He is coming back to place a second hold even though he has one in the queue already.

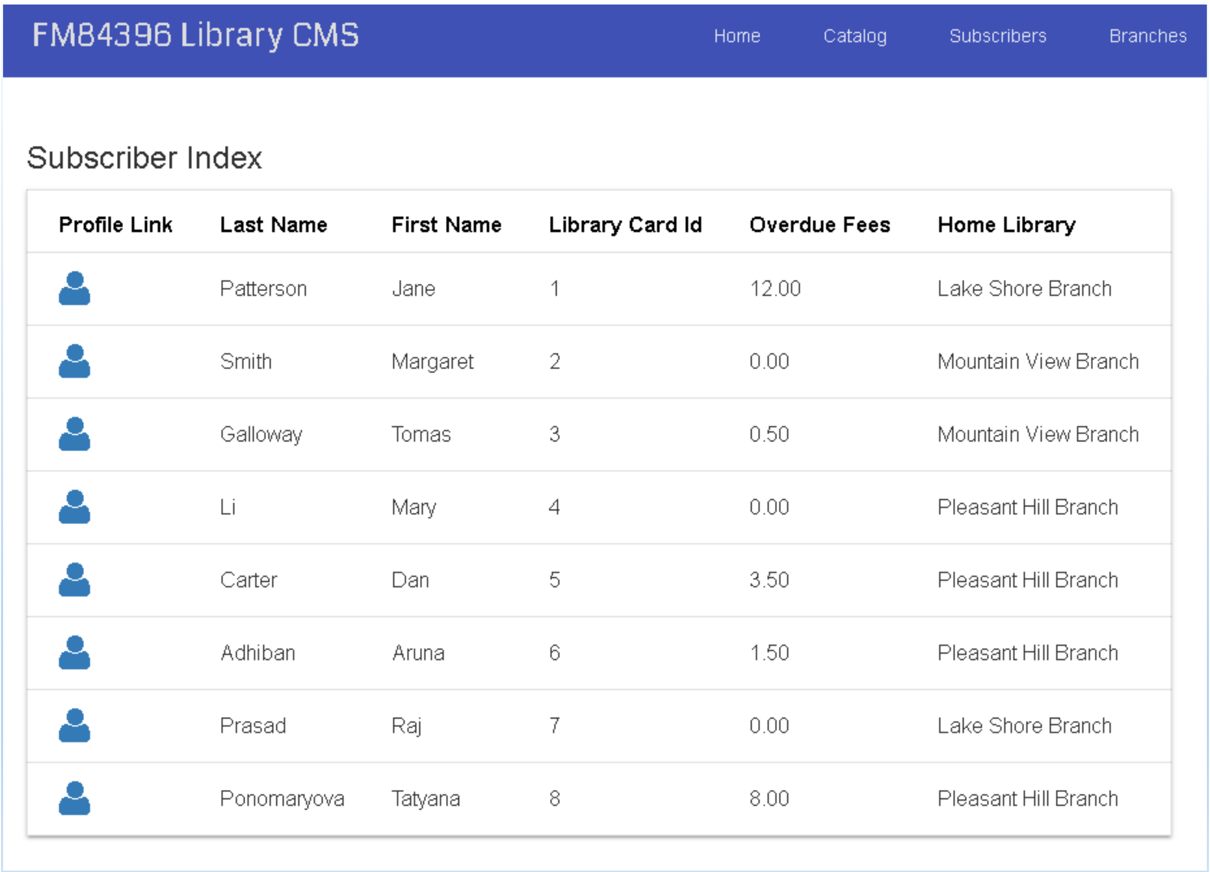


Take Notice! Invalid card!

3. His card is invalid. Wrong card number, revoked membership etc.

After placing, a hold on the item the status of the asset will be changed to "On Hold" and "Current Holds" will be updated. The user will automatically get the item on loan when the it is returned and he is the first in the holds queue. The reason for the asset to show as "On Hold" when there is somebody in the queue is so that people browsing the entries can know "straight away" that there will be some waiting time for the item to be.

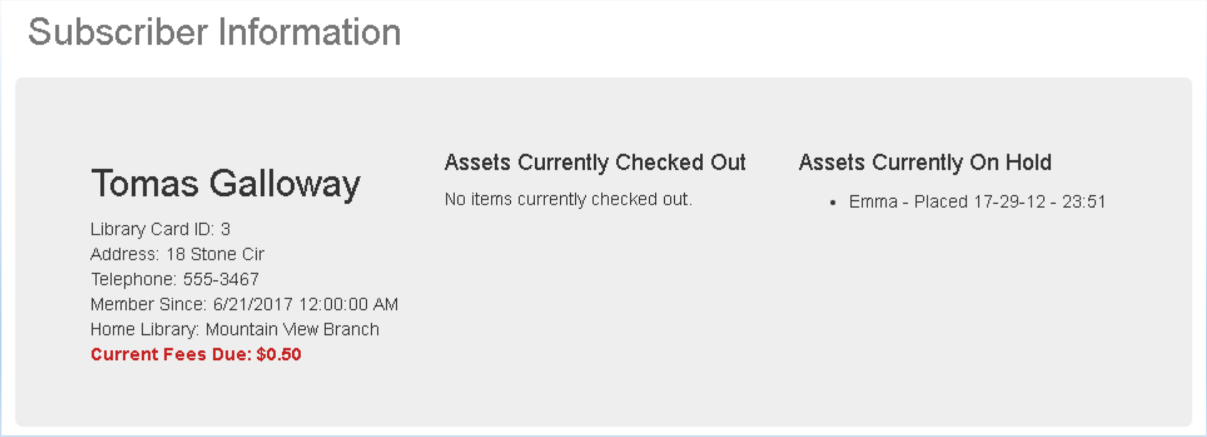
Subscribers Index Page:



Subscriber Index

Profile Link	Last Name	First Name	Library Card Id	Overdue Fees	Home Library
👤	Patterson	Jane	1	12.00	Lake Shore Branch
👤	Smith	Margaret	2	0.00	Mountain View Branch
👤	Galloway	Tomas	3	0.50	Mountain View Branch
👤	Li	Mary	4	0.00	Pleasant Hill Branch
👤	Carter	Dan	5	3.50	Pleasant Hill Branch
👤	Adhiban	Aruna	6	1.50	Pleasant Hill Branch
👤	Prasad	Raj	7	0.00	Lake Shore Branch
👤	Ponomaryova	Tatyana	8	8.00	Pleasant Hill Branch

The subscribers index page (accessible from the home page or the nav. bar) shows all the available info for the user besides their address and telephone number. The home Library Branch Id is associated through a library branch index which the user specifies himself when subscribing (subscription in todo). Here when the profile link is clicked the **Subscriber Details** are shown.



Subscriber Information

Tomas Galloway

Library Card ID: 3
Address: 18 Stone Cir
Telephone: 555-3467
Member Since: 6/21/2017 12:00:00 AM
Home Library: Mountain View Branch
Current Fees Due: $0.50

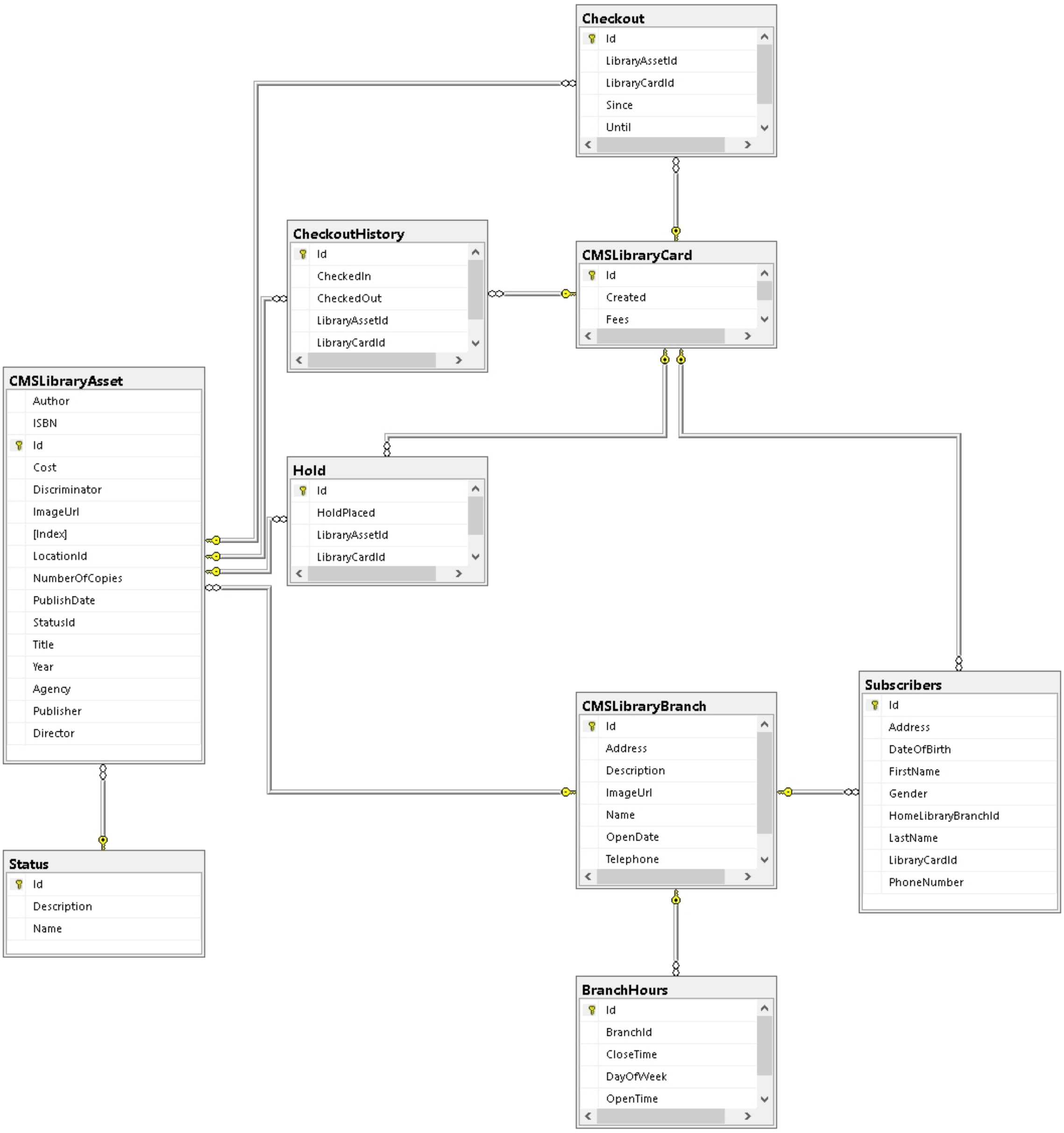
Assets Currently Checked Out

No items currently checked out.

Assets Currently On Hold

- Emma - Placed 17-29-12 - 23:51

The **Subscriber Details** view (above) includes the Address, Telephone and Checkout and the current Hold History of the subscriber.



Database Description:

Above you can see the full map of the current SQL database scheme. A more detail description of the tables is given shortly in the DB Entities in code. However, this is it:

LibraryBranch:

The LibraryBranch is the one to fill the “Branches” View in the future application. Is holds the an image, telephone, open date, address.. Etc. It is a parent branch to only the BranchHours table which is just that (branch open hours per day).

Subscribers:

The Subscribers table holds the info for the “Subscribers Detail” View and besides to the branch (by the HomeLibraryBranchId) is linked only to the LibraryCard.

LibraryCard:

The LibraryCard besides the overdue fees holds a link between the user, his current checkouts, his checkout history and his current assets on hold.

Checkout and Hold:

Checkout and Hold are both utility tables. One holds the checkout status which when finalized goes in CheckoutHistory while the other are the Holds which just get removed. The need for a Checkout Entity is made by the fact that the LibraryAssets have a specific number of copies in available for subscribers and the checkout will have to include the unique key for the copy that the user has checked out. This includes a revisit of the workflow for holds and other.

LibraryAsset:

The library assets includes all the information for the asset as well as link to a Status table featuring just a list of messages and statuses we input manually.

Still to be done:

There is much to be done on the application still. When finished the application should feature the following:

On the user side:

1. Logging Subscription and Session handling.
2. Profile Page for Subscribers where they can update their information.
3. Payment Services for subscriptions and overdue fees.
4. Updated Catalogue view.
5. A Catalogue Search View.

On the admin side:

1. A login page for Administrators.
2. A page where new Assets can be added and current assets can be updated.
3. A mechanism for dynamic loan periods (communicated with the user) for trending assets which are seldom available.
4. A mechanism to update store hours .