Learn how Amlib allows your library to make bookings through the Reservations screen for patrons. Amlib allows your library to make bookings through the Reservations screen for patrons. Bookings can be made on items within the database such as Equipment and PCs. It is generally best to have a separate form code set up for the equipment that you book to allow for easier reporting. It is also possible to set up your NetOpacs to allow patrons to place their own bookings if you choose.

AMLIBSupervisor Settings

1. Launch the Amlib client
2. Go to Main > Supervisor > Locations – the Locations table will display

3. Highlight the first circulating location and click the F9 Hours button – the Working Hours screen for that branch will display
4. Enter the opening hours for that branch and click the **F3 Save** button
5. Repeat for all locations
6. For these settings to take effect you must exit and restart the **Amlib** client

**Booking an Item**

1. Launch the **Amlib** client
2. Go to **Main > Borrowers > Borrower** – the Borrower Details screen will display
3. Search for the patron that would like to make the booking
4. From the menu, select **XReferences > ReservationsModule** – the Reservations screen will display
5. From the menu, select **Items > Item Search** – the Stock Item Enquiry screen will display.

6. Search for the equipment the patron would like to book: enter an item search term and click the **F5 Query** button.

   Please Note: This is a *browse* search, so you must search using the words that the equipment title begins with.
7. A list of equipment will display in the table

8. Highlight the exact item that is to be booked and from the menu select **Borrower > Book Marked Items**

9. This will bring up the Create Booking window
10. Enter all the booking details. The calendar can be used to select a start and end date for the booking or you are able to choose to make the booking in \( n \) Days from Today.
11. When the booking date and time has been entered, select the **Confirm Booking** button.
12. If the time slot has been used by another borrower or is not available, the following message will appear: Item No XXXX is already booked by XXXXX for the period XXXX to XXXX Proceed without this item?

13. If the booking is accepted by clicking the Yes button, you will see the following prompt: Items have been Booked

14. Click the OK button and then the Exit button on the Create Booking window
Check Existing Bookings for a Borrower

To check the booking:

1. Launch the Amlib client
2. Go to Main > Borrowers > Borrower and search for the patron
3. From the menu, select Borrower > Current Bookings – the Existing Bookings for Selected Borrower screen will display containing a list of the bookings that borrower has made.

From here you can either cancel the booking or print (via the F6 Print button) a copy of the booking for the borrower.

4. To cancel a booking: highlight the item line and select the F4 Cancel button.

5. Select the Yes button on the prompt.
View Bookings on an Item

1. Launch the Amlib client
2. Go to Main > StockItems > Stockitem and search for the item
3. From the menu, select Item > CurrentBookings – the Stockitem Bookings screen will display

Issuing a Booked Item

A booked item such as a laptop can be issued to a borrower.

1. If the item is issued before the start time for the booking has passed, the following message will appear allowing the following options:
   ◦ Continue with issue – choose the Yes button: This will issue to the patron and leave the booking on the system
   ◦ Abandon the Issue – choose the No button: this will cancel the issue
   ◦ Continue with Issue and Cancel the booking – choose the Cancel button: this is the recommended option to issue to the patron and clear the booking
2. Loan period depends on parameters for the item Form not the booked period.

NETOPACS

Set up HTML Templates

The Bookings Template

1. This requires some experience of HTML. Open an existing .htm for your style of NetOpacs. If you’re not sure which set of pages you are using, or where they are located please see the guide entitled Find Your NetOpac Templates available on the customer content portal.

2. You can essentially use any existing .htm page to create your bookings page, but a good one to start with is

https://help.oclc.org/Library_Management/Amlib/Borrower/Resource_bookings

Printed: Fri, 12 Jun 2020 19:32:28 GMT
3. For this example we will be opening the Reserve.htm in a text editor and altering the HTML of the document. If you know HTML then you should be able to follow along, but you can also do this by using an HTML editor such as EditPlus or Dreamweaver.

4. Before editing your page, make a copy and save it as Booking.htm.
5. The reason that the Reserve.htm page is a good one to start with in this instance is that it already contains some of the placeholders and form items required for the Booking.htm page. You do not need to remove any items from the current page:

```
INSERTBUTTONSHERE

INSERTDATAHERE

Library Number:

PIN:

Pickup Location:

Reserve

INSERTBUTTONSHERE
```

6. The one thing you may need to change is the text on the “Submit” button so that it says “Book Item” instead of “Reserve”.

```
</select><br />
<
nput type="submit" value="Book Item" />
</p></p>

INSERTBUTTONSHERE
```
7. Now you need to add the rest of the items required to make it a bookings page. There are six “hidden” fields required to make this page work. If you're using DreamWeaver, these are “input” items (or field items) set to type hidden.

```html
<body>
<input type="hidden" id='closeddays' value="@closeddays@" />
<input type="hidden" id='openhours' value="@openhours@" />
<input type="hidden" id='holidays' value="@holidays@" />
<input type="hidden" name="v29" />
<input type="hidden" name="v36" />
<input type="hidden" name="v37" />
</body>
```

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ID</th>
<th>NAME</th>
<th>VALUE</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>hidden</td>
<td>closeddays</td>
<td>@closeddays@</td>
<td>Stores library closed days</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>openhours</td>
<td>@openhours@</td>
<td>Stores library open hours</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>holidays</td>
<td>@holidays@</td>
<td>Stores library public holidays</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>v29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>v36</td>
<td></td>
<td>Stores “from” date</td>
<td></td>
</tr>
<tr>
<td>hidden</td>
<td>v37</td>
<td></td>
<td>Stores “to” date</td>
<td></td>
</tr>
</tbody>
</table>
8. From this table, if you are using DreamWeaver, you can insert the items. For example: In DreamWeaver, find the “Form” items:

![Image of DreamWeaver interface]

9. This is a hidden field, so select “Insert Hidden Field”.

![Image of Insert Hidden Field dialog]

10. Insert the name and the value of this field, but I’m not sure if the name is added as a “Name” or as an “ID”.

![Image of hidden field with closeddays and @closeddays@ value]

11. By clicking on the little edit icon, we can see that typing the word “closeddays” into the field above has put it as a “Name”.

```
Edit Tag: <input type="hidden"
name="closeddays"
value="@closeddays@">
```

12. This will be fine for the last three hidden fields (v29, v36, v37) but not for the first three, so manually edit the source code and change “name” to “id”.

```
Edit Tag: <input type="hidden" id="closeddays"
value="@closeddays@">
```

You can do this for the rest of the items above.

13. Before adding in the items that allow patrons to book the item, we’re going to add the placeholder that allows them to see when the item is already booked for: in a single cell, single row table below INSERTDATAHERE we’re going to add INSERTCURRENTBOOKINGSHERE

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td></td>
<td>INSERTCURRENTBOOKINGSHERE</td>
<td></td>
</tr>
</tbody>
</table>
```

If you open (or refresh) the page in a web browser, you will be able to see how it may look when compiled.
14. Next add the visible form items that allow patrons to select how long they want to book the item for. This will involve some formatting, such as maybe placing them in a table, or just ensuring there are enough line breaks in between items.

The first item is a drop down box (or “select” box) that allows patrons to select a period of time (i.e. 2 hours).

```html
<p><select id="sessionlength" onchange="JavaScript:InitTimes();">
  <option value=" "/>
  <option value="15">15</option>
  <option value="30">30</option>
  <option value="45">45</option>
  <option value="60">60</option>
  <option value="120">120</option>
</select></p>
```

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ID</th>
<th>ONCHANGE</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>select</td>
<td>sessionlength</td>
<td>JavaScript:InitTimes();</td>
<td>Booking time intervals</td>
</tr>
</tbody>
</table>

For each of the options in the drop down box, list the booking times that you would like to allow your patrons in number of minutes, i.e. 2 hours is 120 minutes. Again, if you refresh this page in your web browser, you can see how it will look when compiled.
15. The next drop down box will allow patrons to decide if they want to make the booking by entering the number of days from today’s date, or if they want to enter a date in the future.

```html
<p>
<select id="booktype" onchange="JavaScript:ToggleFields();">
  <option value="DATE">Date</option>
  <option value="DAY">Day</option>
</select>
</p>
```

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ID</th>
<th>ONCHANGE</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>select</td>
<td>booktype</td>
<td>JavaScript:ToggleFields();</td>
<td>Booking type</td>
</tr>
</tbody>
</table>

There should only be two options for this drop down box: DATE and DAY.

This next text field is used if a patron has selected “Day” from the above drop down box.
16. The next set of fields all concern the “from” time/date of the booking

<table>
<thead>
<tr>
<th>TYPE</th>
<th>ID</th>
<th>ONBLUR</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>daysfromtoday</td>
<td>JavaScript:UpdateDays();</td>
<td>Days from today</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>text</td>
<td>dayfrom</td>
<td></td>
<td>Displays day</td>
</tr>
<tr>
<td>select</td>
<td>datefrom</td>
<td>JavaScript:UpdateDate(true);</td>
<td>Displays date</td>
</tr>
<tr>
<td>select</td>
<td>monthfrom</td>
<td>JavaScript:UpdateDate(true);</td>
<td>Displays month</td>
</tr>
<tr>
<td>select</td>
<td>yearfrom</td>
<td>JavaScript:UpdateDate(true);</td>
<td>Displays year</td>
</tr>
<tr>
<td>select</td>
<td>timefrom</td>
<td></td>
<td>Displays time</td>
</tr>
</tbody>
</table>

Note: the “dayfrom” text field should be set to read-only. In HTML this is done by adding readonly="readonly".

The fields for the “to” time/date of the booking are similar.
<table>
<thead>
<tr>
<th>TYPE</th>
<th>ID</th>
<th>ONBLUR</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>dayto</td>
<td></td>
<td>Displays day</td>
</tr>
<tr>
<td>select</td>
<td>dateto</td>
<td>JavaScript:UpdateDate(false);</td>
<td>Displays date</td>
</tr>
<tr>
<td>select</td>
<td>monthto</td>
<td>JavaScript:UpdateDate(false);</td>
<td>Displays month</td>
</tr>
<tr>
<td>select</td>
<td>yearto</td>
<td>JavaScript:UpdateDate(false);</td>
<td>Displays year</td>
</tr>
<tr>
<td>select</td>
<td>timeto</td>
<td></td>
<td>Displays time</td>
</tr>
</tbody>
</table>

**Note:** the “dayto” text field should be set to read-only. In HTML this is done by adding readonly=”readonly”.

17. Your page should already have the fields for library number/barcode, PIN (if your library uses PINs) and a drop down box to select the location. But in case they are absent, they are:

```html
Library Number: <br />
<input type="text" size="40" name="v31" />
<br />
PIN: <br />
<input type="password" size="10" name="v32" />
<br />
Pickup Location: <br />
<select name="v33">
  <option value=""><br /></option>
  <option value="Branch One">Branch One</option>
  <option value="Branch Two">Branch Two</option>
</select><br />
<input type="submit" value="Book Item" />
```

<table>
<thead>
<tr>
<th>TYPE</th>
<th>NAME</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>v31</td>
<td>Barcode</td>
</tr>
<tr>
<td>password</td>
<td>v32</td>
<td>PIN</td>
</tr>
<tr>
<td>TYPE</td>
<td>NAME</td>
<td>EXPLANATION</td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>select</td>
<td>v33</td>
<td>Options should contain location codes</td>
</tr>
</tbody>
</table>

18. The select box for locations must be altered to include:
   - `onchange="JavaScript:ChangeLocation();"`

19. The submit button must also be altered to include:
   - `onclick="JavaScript:ConfirmBooking();"`

20. The final item to be added to your Booking.htm page will be the text `@bookingjs@` just before the closing body tag: `<body>`

21. Now that all of your fields are added, you can have a look at it in your web browser and customise it to suit the look your library wants. For instance, you will notice that none of the new fields have any text describing what that fields are for, so you may want to add labels next to them so they are more user friendly. You may also want to move the fields around on the page, or adjust the default size of them so they are more eye-catching.

Example end result:
22. 12. Once your template is set up, you can save and close it.

## Altering Your Item List

1. Open up your **ItemList.htm** page. You can view this in a web browser to get a good idea of where you would like your bookings link to display.

```plaintext
<table>
<thead>
<tr>
<th>Item Type</th>
<th>Call No</th>
<th>Location</th>
<th>Status</th>
<th>Reserve</th>
</tr>
</thead>
<tbody>
<tr>
<td>STKFORMSTATS</td>
<td>STK_LINES</td>
<td>STK_LOC_PERM</td>
<td>STK_IS_ON_LOAN with STK_IS_RESERVED reserves</td>
<td>STK_DATE_DUE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
2. The placeholder that you need to add to this page to create the link is INSERTBOOKINGHERE

```
<table>
<thead>
<tr>
<th>Item Type</th>
<th>Call No</th>
<th>Location</th>
<th>Status</th>
<th>Reserve</th>
<th>Booking</th>
</tr>
</thead>
<tbody>
<tr>
<td>STKFORMSTATS</td>
<td>STK_LINE1</td>
<td>STK_LOC_PERM</td>
<td>STK_IS_ON_LOAN</td>
<td>INSERTRESERVEHERE</td>
<td>INSERTBOOKINGHERE</td>
</tr>
<tr>
<td>STK_LINE2</td>
<td>STK_LINE5</td>
<td>STK_IS_RESERVED</td>
<td>reserves STK_DATE_DUE</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Editing the Amlib.ini

1. There are two items to be added to your NetOpacs Amlib.ini. This file will be located on your NetOpacs server under something like C:\Netopacs\Amlib.ini and can be opened in Notepad to be edited.
2. The two items involving bookings, which you may need to add to your Amlib.ini under the heading [NETOPACS] are:
   - Booking=*text for link to Booking.htm*
   - BookingForms=*list of Form codes that can be booked*

Compiling Your Pages

The final step in setting up bookings for your NetOpacs is to compile your pages.

1. Launch the Amlib client
2. Go to Main > Supervisor > WebParams – the Web Opacs Parameters screen will display
3. From the menu, select Html Parameters > Compile
4. The Net Opacs Compile window will display – navigate to the folder containing your web pages.
5. You will find ReserveOn.ini and ReserveOff.ini. Whichever one you use depends on whether the students/patrons are allowed to make their own reservations or not so if they are allowed to make reservations, then choose the ReserveOn.ini or vice versa. More recent Net OPAC installations may only have one .ini file called Item_Result.ini – if this is the case then select that. Then click on the Open button.
6. You will then be asked to type in a Web User Code. This code denotes your NetOpac instance. If you are running more than one set of NetOpac pages (for example: a junior and senior set of pages) then you will be using more than one code. Typically, the first instance is designated \textit{a} and the second \textit{b} and so on. If you have only one set of pages, then just type in \textit{a}.

7. Click on the \textbf{OK} button

8. \textit{Amlib} will now compile these pages (and the \textit{amlib.ini} file) – the following message will appear once complete: \textbf{All the Web Parameters have been created}.

\textbf{Restart the NetOpac WebConsole / Service Manager}

Once the recompile is complete, you will need to restart the NetOpac WebConsole or Service Manager (for example: \textit{FireDaemon})
To restart the WebConsole:

1. Now the WebConsole is probably already running, so maximise the WebConsole window.

2. You will then need to reload it using the code you entered at step 7. So in the Style box, type the code you used at step 7 (leave as ‘a’ if you compiled as ‘a’) and then press the orange ReLoad button. That will take a few seconds and then it will be ready to test.

3. The WebConsole has now been restarted.

To restart the Service Manager:

1. If you using FireDaemon or a similar Service Manager, then open the Service manager from your Programs listing.

2. Select the Net Opacs WebConsole, click on the stop button, and then the start button.

3. The service has now been restarted.
Refresh Browser Pages

1. Close any internet browser windows that are open and then reopen a new page.
2. Navigate to the URL or location or favourite that you usually go to when accessing the NetOPACS system and test the new pages.
3. If they look the same as the old pages, then press Ctrl and the Refresh/Reload button on the browser window to refresh the pages properly.
4. Now the pages should reflect the changes that were made, or you should be able to reserve items if you did the Reserveon.ini to Reserveoff.ini change. Do a few searches and ensure that the NetOPACs are functioning correctly.

**using an ‘a’ code is recommended because if the server reboots or WebConsole is restarted, ‘a’ is the default code.**

Reports

You can create lists of booked items including booking details to assist with the item collection process.

1. Launch the Amlib client
2. Go to Main > Reports > RepReservation – the Reservation Reports table will display
3. Click the F1 New button – the Select Report Format table will display
4. Highlight the Booking list by Item template (&BKGLSTK.QRP) and click the Select button
5. Give the report a name (for example: Today’s Booking by Item)

6. Click the **F3 Save** button

7. Highlight the report line and select the **F7 Where** button – the Reservation Reports – Where screen will open:

### Select Report Format

<table>
<thead>
<tr>
<th>Description</th>
<th>Default Stats Code</th>
<th>Default Sub-Stats Code</th>
<th>Update History</th>
<th>Update Entity</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Booking List by Borrower – shows it</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>&amp;BKGLIST.QRP</td>
</tr>
<tr>
<td>Booking List by Class – shows It</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>&amp;BKGLSTC.QRF</td>
</tr>
<tr>
<td>Booking List by Item - shows Item</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>&amp;BKGLSTK.QRI</td>
</tr>
<tr>
<td>Booking list</td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>&amp;BOOKLIST.QR</td>
</tr>
<tr>
<td>Cancel non collected res.</td>
<td>18</td>
<td></td>
<td>Y</td>
<td>Y</td>
<td>&amp;resexpb.qrp</td>
</tr>
<tr>
<td>Email notification of bookings</td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>&amp;BKGEML.QRP</td>
</tr>
<tr>
<td>Email Reservation Notice</td>
<td>320</td>
<td>321</td>
<td>Y</td>
<td>Y</td>
<td>&amp;RESEML.qrp</td>
</tr>
<tr>
<td>Inverell Available Reservations</td>
<td>320</td>
<td>321</td>
<td>Y</td>
<td>Y</td>
<td>&amp;INVRESA.QRF</td>
</tr>
<tr>
<td>Inverell Unavailable Reservations</td>
<td>322</td>
<td>323</td>
<td>Y</td>
<td>N</td>
<td>&amp;INVRESN.QRF</td>
</tr>
<tr>
<td>Kingston Reservation Notice</td>
<td>320</td>
<td>321</td>
<td>Y</td>
<td>Y</td>
<td>&amp;KINRES1.QRF</td>
</tr>
<tr>
<td>List of Reservations</td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>&amp;ResList.qrp</td>
</tr>
<tr>
<td>Name &amp; Phone Format (no Hist)</td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>&amp;RA4000.qrp</td>
</tr>
<tr>
<td>PostCard Format (keeps history)</td>
<td></td>
<td></td>
<td>N</td>
<td>N</td>
<td>&amp;RA4000p.qrp</td>
</tr>
</tbody>
</table>

---

https://help.oclc.org/Library_Management/Amlib/Borrower/Resource_bookings

Printed: Fri, 12 Jun 2020 19:32:28 GMT
8. Enter the following Where statement (which will give you a complete list bookings for collection today only):
   1. **Bookings Only** (Y/N) = Y
   2. (Optional) **Only Show First in Queue** (Y/N) = Y (this would only show the first booking on each item – if you would like to print all bookings on an item for that day, then omit)
   3. **Booking From Date >= Current Date** (select via **Special** button)
   4. **Booking From Date < Current Date + 1 Day** (select via **Special** button)

9. Select **F3 Save** button when complete

10. Click the **F9 Order** button – the Reservation Reports – Order By screen will display:
11. Highlight the following parameters in the Columns table and use the arrow keys to insert them in the Order By column:
   1. Item Barcode
   2. Booking From Date

12. Click the F3 Save button when complete

13. Highlight the Report line then select F8 View – the following prompt will appear: **This type of report may use a Saved File as the Source. Do you want to select the file now?**

14. Select No button

15. You will then see the bookings list for today. This list is a snapshot of bookings at the time the report is run and you may need to repeat step 13 onwards to view the report again at regular intervals throughout the day, depending on your workflow
### Today's Bookings by item

**Tripod: Video/ Camera Tripod**

<table>
<thead>
<tr>
<th>ID</th>
<th>Item Perm Loc</th>
<th>Perm Loc</th>
<th>Date</th>
<th>Time</th>
<th>Call</th>
<th>Equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td>12574</td>
<td></td>
<td>Chelsea Library</td>
<td>Feb 28, 2011 10:15 AM</td>
<td>to Feb 28, 2011 10:30 AM</td>
<td>B237</td>
<td>EQUIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Camera: Olympus DX3600 Digital Camera**

<table>
<thead>
<tr>
<th>ID</th>
<th>Item Perm Loc</th>
<th>Perm Loc</th>
<th>Date</th>
<th>Time</th>
<th>Call</th>
<th>Equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1750</td>
<td></td>
<td>Chelsea Library</td>
<td>Feb 28, 2011 11:15 AM</td>
<td>to Feb 28, 2011 11:30 AM</td>
<td>B301</td>
<td>EQUIP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Camera: Sony DCRTRV340 Digital Video Camera**

<table>
<thead>
<tr>
<th>ID</th>
<th>Item Perm Loc</th>
<th>Perm Loc</th>
<th>Date</th>
<th>Time</th>
<th>Call</th>
<th>Equipment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Chelsea Library</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Author:**

- **MS ALANA ANDERSON**
- **MRS MELISSA F CHRISTENSEN**
- **MRS MELISSA F CHRISTENSEN**