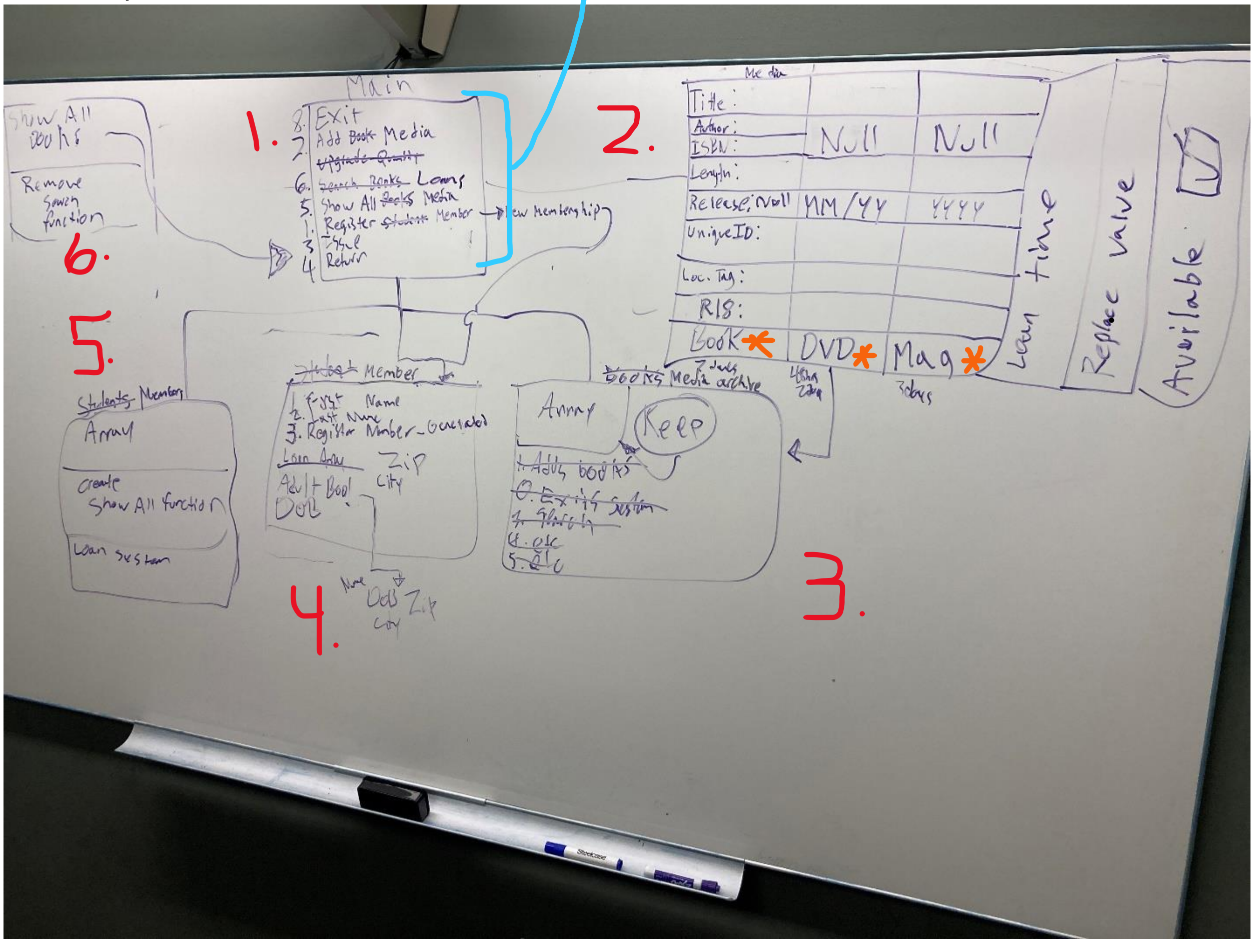
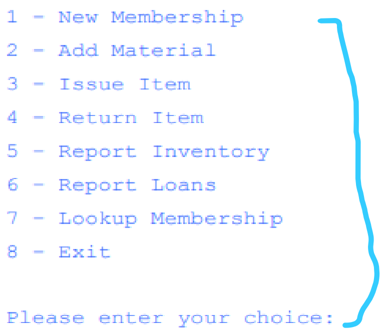
First Draft:



In our first design draft for our Library project, we wanted to have 6 separate classes. Our first class, marked with a "1." we have our main class that would hold the menus, we eventually set it up to follow the format requested of:



```
1 - New Membership
2 - Add Material
3 - Issue Item
4 - Return Item
5 - Report Inventory
6 - Report Loans
7 - Lookup Membership
8 - Exit

Please enter your choice:
```

In the second class, marked with a "2.", we debated whether we should use a book/media class utilizing inheritance to convert it into a magazine or a DVD class as marked by the orange asterisks, but decided against this as we felt the fields of magazines and books were too similar and there was only truly one uniquely attribute of DVDs.

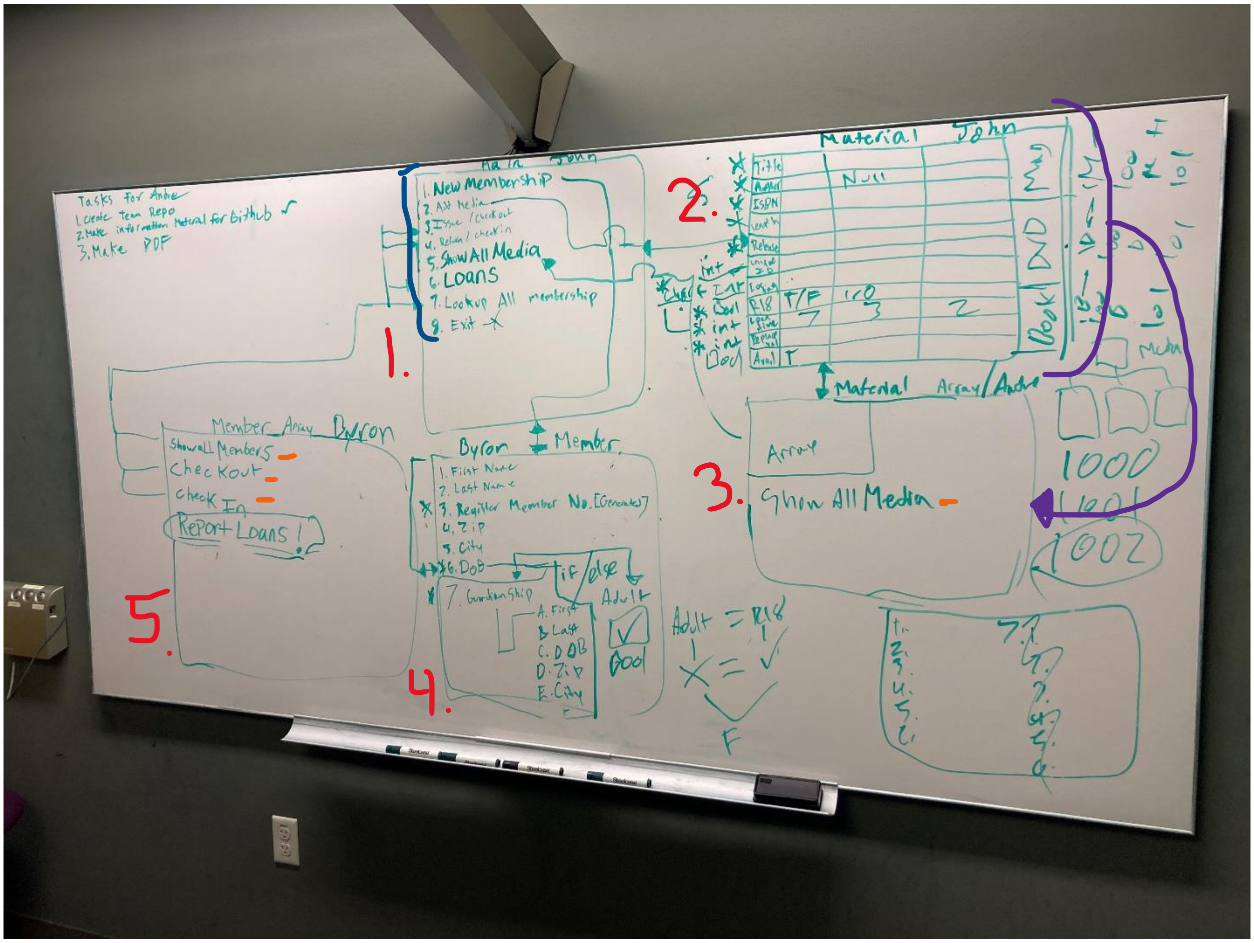
In the third class, marked with a “3.” we set up a class for an array to store the materials, we were trying to develop a search mechanism to search for books to review statistics such as looking up item 1234 to see what it’s title is, info about it, but it was cut as it was felt that the feature was excessive and outside the scope of the parameters of the project.

In the fourth class marked with a “4.” on the figure above is our class for adding members to our library, initially we were also going to use inheritance for children that signed up as members but we decided against it as it made all adults members regardless of whether or not they wanted to be one, so adult information is added to children accounts by association.

The fifth class marked with a “5.” is the array that would store the members of the library. At the time, this was the only function of the array as it did not make sense to do much with the data. At this time, we did not consider tying check-out/check-in features with members but rather, the books instead, this was mitigated to our final draft due to the number of fields already saved by the materials themselves.

The sixth class marked by a “6.” which did not make the cut to the second draft was going to be the storage of commands, such as showing all the materials within the library, showing all the members of the library, searching for specific members loans.

Second Draft:



In this similar design diagram, each class marked "1. to 5." is still mostly the same classes but our main class is now organized much closer to the menus listed in the assignment. Particularly in class 2 we chose a system that hides attributes that have nulled out properties, like DVDs that have issue numbers, or Magazines having ISBNs. In class 3 we mitigated some of the functions of the depreciated sixth class into the materials array such as the report inventory feature. In class 4 we worked farther from the original abstraction idea and moved towards the association data when a child member would create guardianship information, since it is not going to be needed much outside of material replacement issues and displaying that data when particularly looked up. In class 5, more portions of the 6th class were also mitigated into such as the show report all members and the check-in/check-out system as we decided to tie the system into the member array instead.

The loan system was the hardest part of our project, and it is unsure whether it will make it to the final stages of the project. This is largely because our first idea of having the materials record when they were checked out conflicted with how we were planning to store the loan information within our systems as they ended up being mitigated towards the member array as we felt that the material array was getting too convoluted. We wanted it to make a time stamp the day it was recorded, the material has an indicator that clarified whether or not the material was a book, magazine, or a DVD with a loan time set to days to add to the time stamp that would let the borrower know when the item was due. We used a Boolean variable to tell whether an item was logged out of the system as it was defaulted to have a true "availability" status within the system. Items that would flag as false would make them unavailable to be loaned out to anyone else as well as for them to show up within the loan system as they were considered on loan. Originally, we compared our loan system like a transaction log when this was not our interpretation, the system only needed to report items out on loan, not a transaction system that reported what had been loaned.