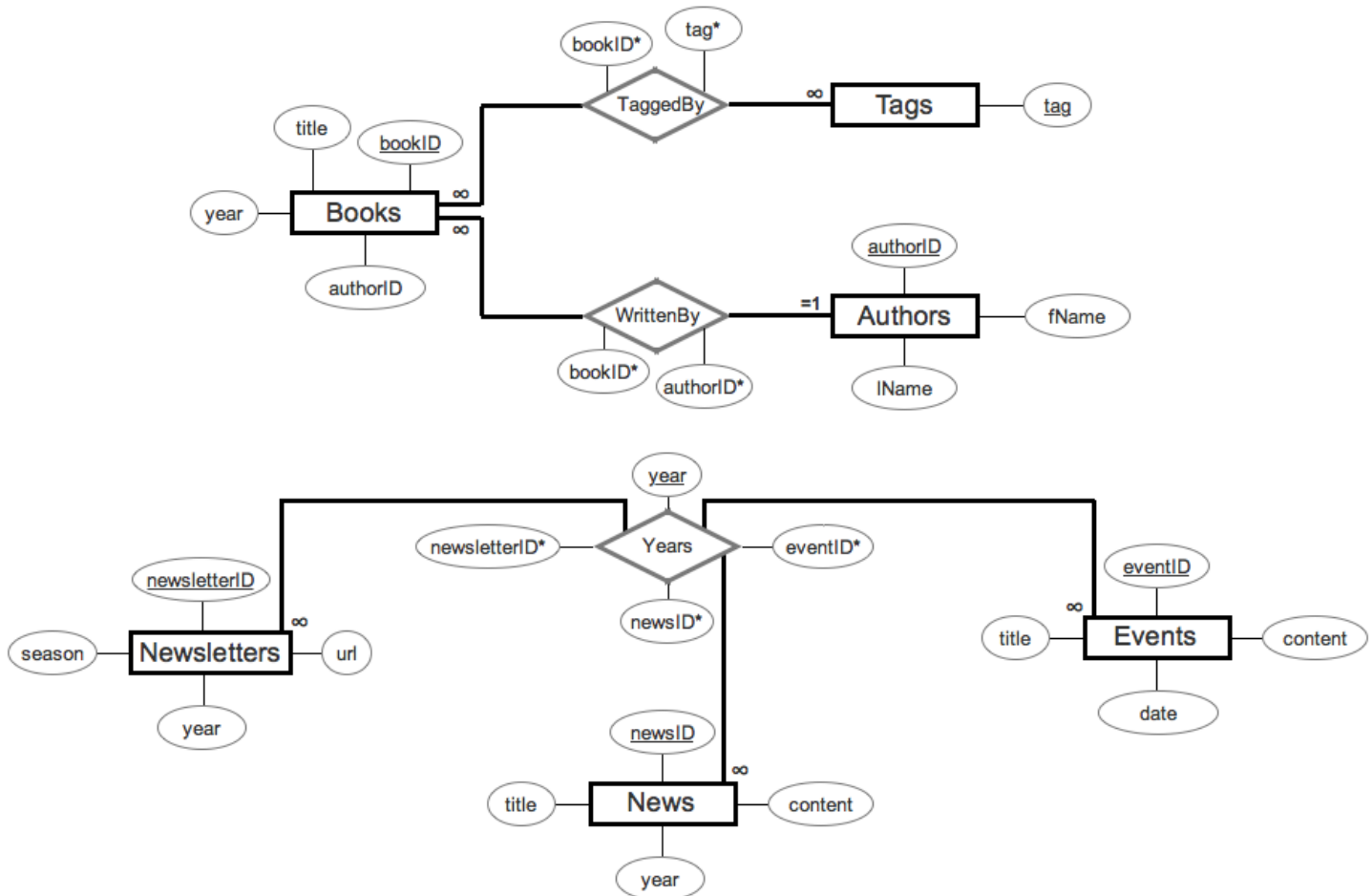


# Give Us A Minute

Noah Grossman (nbg22), Nitin Rajinikanth (ndr34), Vincent Bello (vrb25), Angel Martinez (aam255)

## Milestone 3: Draft Implementation

### Database Design



Our client wishes to have a book database implemented that stores searchable information as well as tag words that can be associated with certain books. This is so people can visit the site, browse the books onsite, and search for books by various information, including tag words. Our client also wishes to be able to add current and upcoming events to the Events page, in addition to past News and Newsletters to the News page.

The above ER Diagram accomplishes our client's wishes by breaking up the Library information into three entity sets- Books, Tags, and Authors.

- 'Books' keeps a list of book entities which contain the book's title, an authorID that points to the book's author, and a bookID for relational purposes.
- 'Tags' keeps a list of tag words that will be searchable in the site, as well as tagIDs for relational purposes.
- 'Authors' keeps a list of authors' first and last names, as well as IDs for relational purposes.
- 'TaggedBy' is a relationship set that links a book in 'Books' to zero or more tags in 'Tags'. A tag can be linked to zero or more books.
- 'WrittenBy' is a relationship set that links a book in 'Books' to exactly one author in 'Authors'. An author can be linked to zero or more books in 'Books' since an author can be the writer of multiple books.

Books(bookID : INT, title : VARCHAR(200), year : YEAR(4), authorID : INT)

Authors(authorID : INT, fName : VARCHAR(20), lName : VARCHAR(20))

Tags(tag : VARCHAR(20))

WrittenBy(bookID\* : INT, authorID\* : INT)

TaggedBy(bookID\* : INT, tag\* : VARCHAR(20))

\_\_\_ denotes primary key (underline)

\* denotes foreign key

- 'bookID' is a foreign key for WrittenBy linking to its parent table Books, and is also a foreign key for TaggedBy linking to its parent table Books; on update it cascades, and on delete it cascades.
- 'authorID' is a foreign key for WrittenBy linking to its parent table Authors; on update it cascades, and on delete it cascades.
- 'tag' is a foreign key for TaggedBy linking to its parent table Tags; on update it cascades, and on delete it cascades.

The field 'title' in Books is not unique because there can be more than one book with the same title, as long as they were published in different years.

We have also added entity sets 'Newsletters', 'News', and 'Events' to store relevant information. We are going to implement a "Year at a Glance" feature where the user can view all News, Newsletters, and Events for a given year. As a sidenote, 'News' will store information about NAMI Finger Lakes in the local news, where as 'Newsletters' will store information about the seasonal newsletters the organization publishes.

- 'Newsletters' keeps a list of the organization's seasonal newsletters which contain the url used to access the PDF file, the season it was published (Fall, Winter, Spring, or Summer), the year it was published, and a newsletterID for relational purposes.

- 'News' keeps a list of all news stories published about NAMI Finger Lakes. This includes the title of the story, the year it was published, the content in the story, and a newsID for relational purposes. Some stories are simply a title with no content, so the content field can be null.
- 'Events' keeps a list of current and upcoming events at NAMI Finger Lakes. This includes the event title, date, and content that provides more details on the event. There is also an eventID for relational purposes.
- 'Years' is a relationship set that links newsletters, events and news published in the same year. Some years may not contain a newsletter, news story, or event, so their respective paths can be null, and some can contain more than one per year.

Newsletters(newsletterID : INT, url : VARCHAR(100), year : YEAR(4), season : VARCHAR(6))

News(newsID : INT, title : VARCHAR(100), year : YEAR(4), content : VARCHAR(8000))

Events(eventID : INT, title : VARCHAR(100), date : DATE, content : VARCHAR(8000))

Years(year : YEAR(4), newsletterID\* : INT, newsID\* : INT, eventID\* : INT)

\_\_\_\_ denotes primary key (underline)

\* denotes foreign key

- 'newsletterID' is a foreign key for Years linking to its parent table Newsletters; on update it cascades, and on delete it cascades.
- 'newsID' is a foreign key for Years linking to its parent table News; on update it cascades, and on delete it cascades.
- 'eventID' is a foreign key for Years linking to its parent table Events; on update it cascades, and on delete it cascades.

## PHP Interaction

Listed below are all the files on our server and their respective functionality.

- header.php
  - Admin login form with submit action going to login.php
- login.php
  - Authorizes login by matching the credentials to the administrative credentials
- config.php
  - File containing the relevant database credentials
- adminEdit.php
  - Available if the logged\_user is the administrative user
  - Functions for adding and deleting books or current events from the database
  - Functions for uploading files such as newsletters and annual reports
- resources.php

- Contains Amazon and PayPal links
  - Contains functions for displaying list of books as well as searching for books using author, title, or tag information
- newsletters.php
  - Contains links to uploaded newsletters
- index.php
  - Displays home page with slider of images
- footer.php
  - Contains function for the footer
- events.php
  - Contains functions for displaying all the events in the database
  - For admin users, contains functions for adding and deleting events from upcoming events list
  - Displays Calendar for dynamic updating (using Google Calendar)
- aboutus.php
  - Contains information about the organization and links to annual reports
- errors.php
  - Handles errors in user friendly way, using PHP error logs
- namifl\_style.css
  - Basic style added to all files. Contains header styling, footer styling, paragraphs, etc.
- jquery.css
  - Not used yet; style relevant to JQuery plugins and could be used later.
- font-awesome.min.css
  - Basic icons, like the ones used on the navbar and in the contact info.
- bootstrap.min.css
  - Front-end framework: [www.getbootstrap.com](http://www.getbootstrap.com)
  - CSS templates for typography, forms, buttons, navigations, interface components, etc.
- bootstrap-theme.min.css
  - Customizable theme with appropriate color scheme.
- jquery.js
  - Javascript library: simplifies client-side scripting.
  - Necessary for Bootstrap to work.
- bootstrap.min.js
  - Javascript extension of Bootstrap: allows dynamic behavior of components such as the navbar or the carousel on the home page.

## User Testing

We tested the site with two different people. Tester (a) is a 21 year old male undergrad using Google Chrome and tester (b) is a 26 year old female grad student using Safari.

1. Find NAMI's contact info
  - a. Found it right away. Went to about tab and to the contact section.
  - b. Also found it quickly but first went to resources tab before looking at about tab.  
Only took a couple seconds longer.
2. Find the crisis/support information in case of emergencies
  - a. Looked for it in resources page and could find it, eventually found it in home page.  
Either it doesn't stand out enough in home page or we should also include it under resources.
  - b. Found it right away on the home page
3. Find calendar of events
  - a. Found it right away
  - b. Found it right away
4. Find and open a newsletter from 2010
  - a. Went to news page right away but didn't find the newsletters right away, he started looking at the 2010 section of NAMI in the news
  - b. Went to news page and found it right away
5. Find a news story from 2009
  - a. Found it right away
  - b. Found it right away
6. Register as a NAMI advocate
  - a. Remembered it was on the home page and found it right away
  - b. Went to get involved section under resources and found it
7. When was NAMI of the Finger Lakes founded?
  - a. Went to history section of about
  - b. Clicked directly on about page
8. How can you donate money to NAMI
  - a. Went to home page
  - b. Remembered from previous task that it is also under resources tab
9. Where would you log in if you had an account
  - a. Found it on the top right corner of the page right away
  - b. Also saw it right away
10. Find where the library book information is
  - a. Went to resources and scrolled down
  - b. Went to resources but because she was using a smaller screen library wasn't seen on the page right away, took no longer than 5 seconds and then scrolled down.