Team Dec 14-13
CySwap
Design Document

Client: Merry Rankin
Adviser: Professor Mitra

Team Members:
Fabian Briesmoore
Jared Cook
Kyle Johnson
Adam Sunderman

Table of Contents

Functional Requirements

FR-1	The system shall allow users to login and logout via Iowa State's NetID system.
FR-2	The system shall allow users to post items for sale and edit a previously post.
FR-3	The system shall allow users to search for posts based on specified criteria.
FR-4	The system shall allow users to view item postings.
FR-5	The system shall allow users to post comments on any posting.
FR-6	The system shall allow sellers to mark transactions as completed.
FR-7	The system shall allow users to contact sellers to initiate discussion on completing the transaction.
FR-8	After a transaction has been completed, the system shall allow users (buyer and seller in a specific transaction) to leave positive or negative feedback on one another.
FR-9	The system shall allow users to report posts that violate the posting code of conduct.
FR-10	The system shall allow moderators to view reported postings in a F.I.F.O. ordering.
FR-11	The system shall allow moderators to remove postings from the list of reported postings.
FR-12	The system shall allow moderators to edit or delete postings that violate the posting code of conduct.
FR-13	The system shall allow moderators to suspend account privileges for users who abuse the website's posting code of conduct.
FR-14	The system shall allow admin to assign and remove moderator privileges from NetID accounts.
FR-15	The system shall allow admin assign and remove moderator privileges from NetID accounts.
FR-16	The system shall allow admin to view usage statistics.
FR-17	The system shall allow admin to edit content in information sections of the website (i.e. posting code of conduct).
FR-18	The system shall allow admin to configure which site features are enabled and disabled.
FR-19	The system shall remove posts that are 60 days old unless they are "reposted" by a user.

|	
FR-20	The system shall display a section of safety tips, including appropriate locations and police contact information.

Non-Functional Requirements

NFR-1	The system should be seen as visually appealing by a number > 75% of surveyed students.
NFR-2	During high traffic timeframes (beginning of semester) response times should be < 1 second on average. (assuming no network delay on client side)
NFR-3	The site should not be able to have its information maliciously accessed through user input, direct server requests and cookie jacking.
NFR-4	Must be easily maintainable after project completion. Proposed maintenance features include: o The ability to update static content on the website. o Turn on and off certain features o Changing styling such as logo or color scheme. o Change the number of reports it takes to be added to a list of moderator action required reports.

System Block Diagram
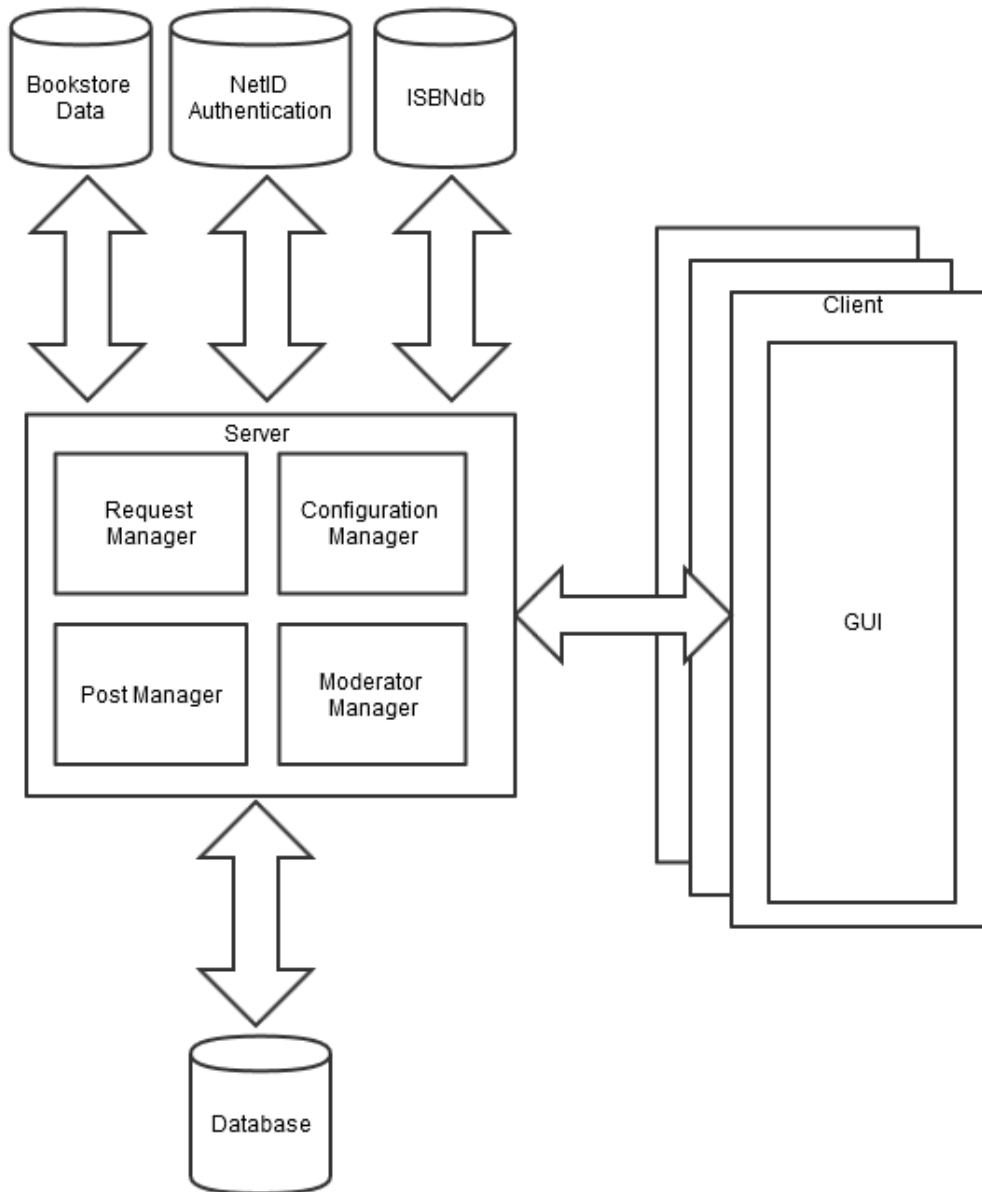
The server communicates with the database to get user data (such as account details and settings) and other necessary data. This data, along with other data (possibly from other databases) is used to generate a page. The pages will be generated using php. The system will communicate ISU servers for authenticating users. The client will communicate to the server through a session. The system will pull information about books given an isbn number using ISBNDB. The system will pull information for a course's textbooks from a dump of the bookstore's database.

System Analysis

The system is designed to allow Iowa State students and faculty to be able to sell and trade books and other commodities with other users. Account details, system settings, posted data, and other necessary data is stored in the mySQL database and accessed by the server. Managing users and other system maintenance will be managed by GSB (Government of the Student Body) representatives. The system will likely be managed by individuals with little technical experience. Consequently, intuitive and simple administrator options will be needed for at least one administrator user. Different levels of administrators will likely be necessary. Administrators will be able to block users, manually adjust user data and postings made by users. In addition to being easily configurable by one or more administrators, the system will need to implement searching and filtering for different items, will be compatible with Iowa State netids, will need to be easily expandable for new categories, will have a feedback system, will need to be secure, and may need to communicate with other databases to get data for different categories.

Though this system is responsible for connecting users who want to sell and buy a common item, it is not responsible for dealing with financial transactions. Users will communicate to each other through email after clicking on a link from the post of the item the buyer is interested in. This is possible because all users that are able to connect to other users will be logged in using their Iowa State email accounts. Viewers of a page that are not logged in will be able to view postings but not be able to see specific information about items, including information necessary to contact the seller. Though the system will be secure with user data, it will not be able to ensure that users are always honest. For this reason, a feedback system will be implemented. Additionally, information advising users to be cautious when making transactions will posted on the website.

Though this is currently how the system is going to operate, the features that GSB wants could change and better implementations might replace the current system. Because of this, meetings with them are ongoing and system design is still in progress.

Input/Output Module Specification

Request Manager

The request manager will contain the code to accept URL requests and delegate the type of tasks to the different request manager classes (Page Fetcher, Configuration Manager, Moderator Manager, etc.).

Module Function	Input	Output
Accept URLs from clients' browsers	Requested destination URL.	html and javascript for client to render requested page

Test Case	Success	Fail
User clicks a link on current html page.	link exists in routing php. Page loads successfully.	link does not exist in routing php. Page doesn't load successfully.

Configuration Manager

The Configuration Manager will contain the code to manage the configuration of the website from enabled and disabled features to the content displayed on the information pages (such as the posting code of conduct).

Module Function	Input	Output
Modify information content on static pages.	html/javascript object to modify, new content	html and javascript for client to render updated page.
Enable Feature	featureID	html for client to render "success" notification.
Disable Feature	featureID	html for client to render "success" notification

Test Case	Success	Fail
User (admin) modifies static content	Content is in valid format for page object that user is trying to modify (string for text modifications)	Content isn't in valid format for the page object that is being modified.
User (admin) enables feature	Feature becomes enabled on the site.	Feature does not become enabled on the site.
User (admin) disables feature	Feature becomes disabled on the site.	Feature is not disabled on the site.

Verification/Authentication

The Verification/Authentication class will contain the code to interact with Iowa State's NetID login system to authenticate users.

Module Function	Input	Output
User login	email/password	Verification that a user is logged in (Session)

Test Case	Success	Fail
User logs in for the first time (logs in with NetID and our system creates a user with that NetID and stores necessary information on top of that profile.	NetID added to our database. Session is successfully created.	NetID added to our database. Session is not created.
User logs in after their first time (uses profile already in our database).	Session is successfully created.	Session is not created.

Post Manager

The post manager will contain the code for users to create and update information on their posts.

Module Function	Input	Output
Create Post	Post type, post sellerID, Posting fields (varies by type).	HTML and Javascript for client to render posting.
Update Post	Posting ID, modified fields	HTML and Javascript for client to render updated posting.
Get Post	Posting ID	HTML and Javascript for client to render posting.

Test Case	Success	Fail
User creates posting	Associated information is stored in the database.	Associated information is not stored in the database
User updates posting.	Database fields are updated to match the user's update.	Database fields are not updated to match the user's update.
Home page, category page or view posting page requests a posting's information.	Posting information is correctly returned.	Posting information is not returned or is incorrect.

Moderator Manager

The moderator manager will contain the code for moderator functions like report handling and blocking users

Module Function	Input	Output
Block a user.	User's netid	Confirmation that the user has been blocked.
Unblock a user.	User's netid	Confirmation that the user has been unblocked.
Close a reporting case.	A report's status.	Confirmation of the report's new status.

Test Case	Success	Fail
User (admin) blocks a user.	User is blocked from the system.	User can still gain access to the system.
User (admin) unblocks a user.	User is not blocked from the system.	User cannot gain access to the site.
User closes a report.	Report is marked as closed and is archived.	Report is not successfully closed.

ISBNDB

The ISBNDB connection will be responsible for the retrieval of textbook information provided an isbn.

Module Function	Input	Output
Retrieve information about a textbook	ISBN	Information about the book.

Test Case	Success	Fail
Request to retrieve information from the database given an isbn.	Correct information is returned.	No information is returned or incorrect information is returned.

Bookstore

The bookstore connection will contain code responsible for determining which books are currently required for a course.
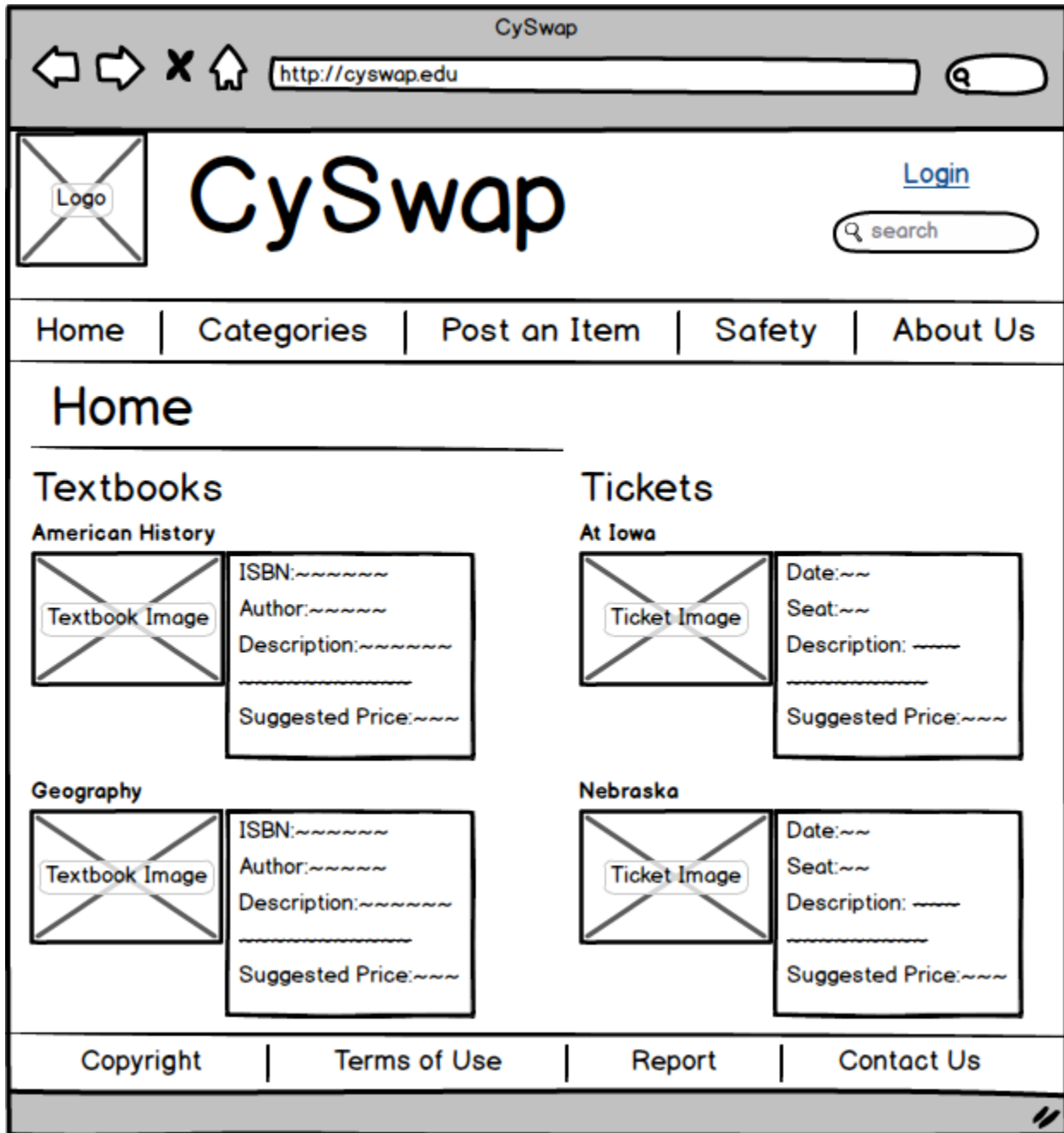
Module Function	Input	Output
Determine which textbooks a course requires.	Bookstore database dump.	Information linking current textbook postings to a course.

Test Case	Success	Fail
Create posting that matches a required course.	Website displays the course information on the book's posting.	Website doesn't displays the course information on the book's posting or the course information is incorrect.

User Interface Specification

This section will show all the different mockups for each screen.

Home Screen

Category Screen

CySwap

http://cyswap.edu

CySwap

Logo

Login

🔍 search

Home | Categories | Post an Item | Safety | About Us

Textbooks

Sort by: [Most Recent ▼]

Posts per page: [10 Posts per page ▼]

American History

Textbook Image

ISBN:~~~~~~
Author:~~~~~
Course (if applicable): ---------
Description:~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Suggested Price:~~~

Geography

Textbook Image

ISBN:~~~~~~
Author:~~~~~
Course (if applicable): ---------
Description:~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Suggested Price:~~~

Page: (Previous) [1] [2] [3] [4] [5] ... [20] (Next)

Copyright | Terms of Use | Report | Contact Us

Shows brief descriptions of several postings within a certain category.

Posting Screen

CySwap

http://cyswap.edu

CySwap

Login

search

Home | Categories | Post an Item | Safety | About Us

American History, the Early Years

Seller: kabernsj

👍 3   👎 1

Textbook Image

P1
P2
P3

Suggested Price:
$37

Contact Seller

Title: American History The Early Years, Student Edition
ISBN: 9780078216114
Author:Glencoe
Edition: 3rd
Course: History 999
Description: This textbook is in great shape; it is like new.
It has only been used a few times. There is no writing or
highlighting in the textbook.

Copyright | Terms of Use | Report | Contact Us

Shows an individual posting.

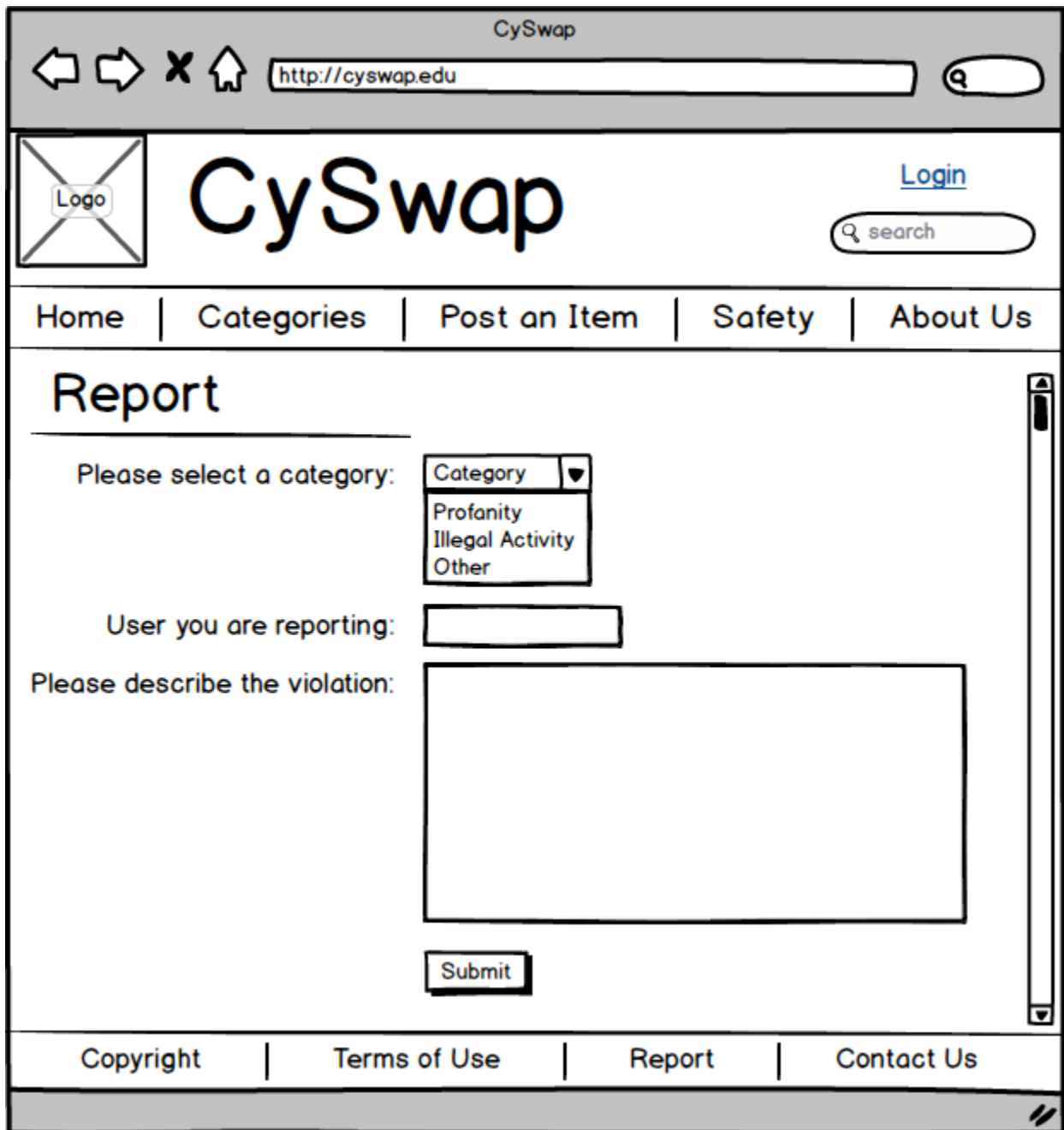## Contact Seller Popup

Contact Seller

To: kabernsj

Subject: American History, The Early Years on CySwap

Send   Cancel

Pop-up that will appear when the 'Contact Seller' button from the previous screen is clicked.

Report Screen

CySwap

http://cyswap.edu

Logo CySwap

Login

🔍 search

Home | Categories | Post an Item | Safety | About Us

Report

Please select a category:

| Category ▼ |
| Profanity |
| Illegal Activity |
| Other |

User you are reporting: [_____]

Please describe the violation:

[   ]

Submit

Copyright | Terms of Use | Report | Contact Us

Screen where a user can report behavior that is a violation of the site's terms of use.

Create a Post Screen

CySwap

http://cyswap.edu

Logo **CySwap**

Login

Q search

Home | Categories | Post an Item | Safety | About Us

Post an Item

Please select a category: Category ▼
Textbook
Miscellaneous

ISBN:

Description:

Submit

Copyright | Terms of Use | Report | Contact Us

Screen users will see when they are posting an item.

Software Specification

Languages: PHP, JavaScript, HTML, CSS, and mySQL.
Libraries: Laravel (PHP framework)
Database Server: mySQL
Development Tools: Atlassian Products: Bamboo, Confluence, and Jira.

Design Decisions

We determined to use PHP after examining the alternative option of using Java with Spring MVC. Our deciding factor was that PHP would have a smaller learning curve.
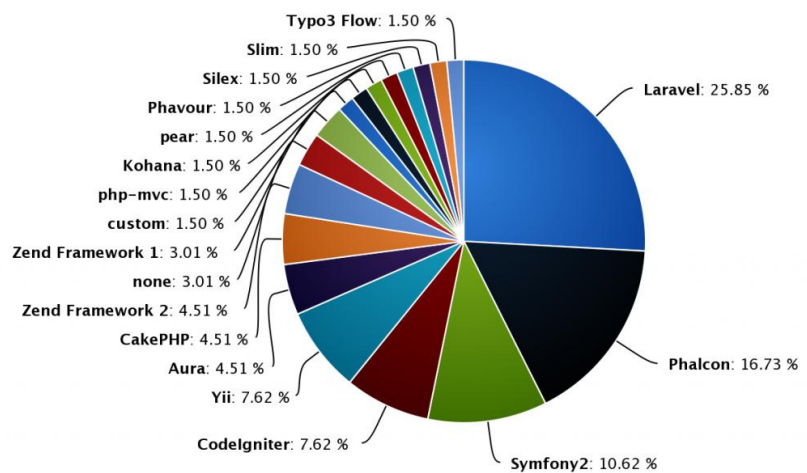We determined that MySQL was the best choice for our database because of our prior knowledge with the system.
We determined that Bootstrap would be used for our web site because of Jared's previous experience with Bootstrap and its popularity and documentation. It was also chosen because it would lead to faster site development.
Laravel our PHP framework was chosen based on our research on PHP frameworks. It was found to have the smallest learning curve. In addition, Laravel was found to be the most popular PHP framework in 2013. The main article we used when making this decision can be found here: http://www.sitepoint.com/best-php-frameworks-2014/

Framework popularity, end of 2013; SitePoint

Typo3 Flow: 1.50 %
Slim: 1.50 %
Silex: 1.50 %
Phavour: 1.50 %
pear: 1.50 %
Kohana: 1.50 %
php-mvc: 1.50 %
custom: 1.50 %
Zend Framework 1: 3.01 %
none: 3.01 %
Zend Framework 2: 4.51 %
CakePHP: 4.51 %
Aura: 4.51 %
Yii: 7.62 %
CodeIgniter: 7.62 %
Laravel: 25.85 %
Phalcon: 16.73 %
Symfony2: 10.62 %

Highcharts.com