THIS IS THE EDITED VERSION

# UMass BookShare

Chris Paika, Piers Calderwood, Peter Chau, Walter Doan, Amy Jiang, Bianca Tamaskar, Qiwen Wang

# Software Requirements Specification

Draft 3
April 30th, 2015

# CSRocks Inc.

| Version | Primary Author(s) | Description of Version | Date Completed |
|---|---|---|---|
| 1 | UMass BookShare | First Draft | 2/16/2015 |
| 2 | Bianca | Updates before Beta Release | 3/31/2015 |
| 3 | Walter | 1.0 Release | 4/30/2015 |

## Description

UMass Bookshare is a web application that serves as an online marketplace for primarily, but not restricted to, UMass Amherst students. Students can buy, sell, lend, and borrow textbooks from others on the app. Essentially students will begin by creating a user account by filling in the required fields such as their name and email address. The user will have options to edit their profile and even adding a profile picture as well once the account has been created. Users can act as lenders, buyers, and sellers. If a user is interested in selling a book, he or she can simply create a listing by filling out all the necessary fields in the create listing tab. Once the user successfully submits a listing, the listing will be visible on the users profile page, the home page, and it will also be search-able. If a user wants to buy a book, he or she can simply just search for it in any of the search bars throughout the web app. If there book is not available at the time, the user can has the option of adding it to their wishlist. The function of the wishlist is so users can add books they're looking for onto it. Once added, the user will be notified via email when a book they are searching for is available. After the user is contacted, he or she can simply just search for the available book and contact the owner of that book to work out a transaction. If a user is ever confused with how to navigate or use the web app there is a support page where the user can read up on how to execute common functions of the app. If the user's question has not been satisfied they can simply email our support email that is listed on the page.

UMass BookShare is a web application that will serve as an online marketplace for UMass Amherst students to buy, sell, lend, and borrow textbooks. Essentially, students will begin by creating a user account with required fields, their name and email address, and also several optional fields, such as their profile picture, phone number, and major. Users can act as both sellers and buyers. If a user is interested in selling a textbook, he or she simply creates a listing with relevant information about the book (title, author, International Standard Book Number, etc.) as well as how he or she prefers to be contacted. The seller can choose to sell, rent, or lend a book to another student for whatever price he or she feels is appropriate. When a user wants access to a textbook, he or she can search for the book by some keyword or ISBN and will then be given a listing of available books. Each listing will have details about the method of sharing the buyer is willing to do (sell, rent, lend). If the buyer is interested in getting a book, the buyer contacts the seller through the desired third-party communication type (for example: email or text message) and the two students can designate a meeting place and time to exchange the book and payment, if applicable. The application will also have a wish list feature where a user can choose to be notified if a book they want has become available, a filtering system that only shows users the sharing method they are looking for, and a rating system that allows users to assess the reliability of other buyers and sellers.

The objective of this web application is to provide a more convenient, efficient way for college students to get the books they need. Our target customer group is UMass Amherst area students who are interested in sharing books with their peers and are

concerned about sustainability and the high price of textbooks. Currently, there is no effective way for students to buy and sell books from each other. Some students use Facebook groups, but it is highly ineffective. UMass BookShare will allow students to get their books more quickly as they are using an application that is devoted to the sharing of books and because all of the sellers are located on the UMass campus. It is cost-effective because students will not have to rely on the Textbook Annex and other students may be more sympathetic to the high cost of books and may be less likely to charge exorbitant prices. The application is also unique in that it allows students to share books with each other for short periods of time, rather than an entire semester. This will be helpful to students who are reluctant to purchase textbooks because they anticipate that their limited use of the book is not worth the price. Another key benefit is that UMass BookShare is a more sustainable option as it eliminates the need for shipping and packaging.

Existing alternatives have some key strengths, but also some important weaknesses that UMass BookShare aims to fix.

- ***Facebook*:** Several students post in the UMass Class of 20XX groups about books they are trying to buy or sell. The main strength of this alternative is that these groups are huge and most students have Facebook. These posts can be viewed by a large number of people, increasing the likelihood you will get the book. However, Facebook is not built for this. Many students pay no attention to these groups and there is no way to easily search. Also buyers/sellers are restricted to the groups they themselves are members of and cannot post to other groups.
- ***Online Booksellers*:** Online booksellers, like Amazon and Chegg, offer a vast selection of books, both new and used. One can often find extremely low prices on these sites. Disadvantages of online booksellers include the time it takes for the book to actually arrive in Amherst, shipping costs, wasteful packaging, and restrictive renting periods.
- ***Textbook Annex:*** The Textbook Annex is a convenient place to purchase textbooks because it is located on-campus and is guaranteed to have custom versions of textbooks. However, a major weakness of buying books at the Textbook Annex is how expensive it is; the convenience comes with a cost. Additionally, the Textbook Annex does not always have used copies of books, so students are forced to buy brand new, expensive copies of their books. Furthermore, students rarely get a fair return when they try to sell their books back at the end of the semester. The Textbook Annex also has restrictive renting periods.

For further product information, we will have a “Help” page, which will include Frequently Asked Questions and How-To’s for tasks like creating a book listing, searching for a book, editing a post, and more.

**User Documentation: User Guide w/ Pictures**

- How to Login

    - where to login with username and password
- New User Registration Steps
    - how to create account with required fields (name, email, institution)
- Forget Password Help
    - password recovery and reset
- Update User Profile
    - make changes in profile form
- Create/Edit/Delete Posting
    - makes changes in a posting form
- Add/Remove items from WishList
    - make changes in the wishlist form
- Search/Filter/View Postings
    - how to search/filter/view postings
- Rate Users
    - how to rate and comment on users

**Admin Guide**

- Overview of System
    - Classes/Components
- Installation/Compile
    - Instructions to compile code in Github (includes Node.js installation, database setup, and instructions for running the application)

**Help Menu in Web App**

- Contact Us
    - contains email to contact or a little message box that will send email
- Support
    - for the most common issues

## Scope

The online platform is only for facilitating interaction between people wanting to buy, sell, rent, or borrow books locally. This means the service offers no support for selling anything besides books. If the item does not have an ISBN number it is not allowed on the platform. The reason for this is two fold. First, it will decrease the chance users buy the wrong books or that the wrong book is posted. Second, it is easier to find postings or match recent postings to people's wishlists, as ISBN numbers are edition specific.

The service will also not offer "middleman" services. Contact between two users is done outside of the site, there is not private messaging or public comments on posts. If a buyer wishes to contact a poster they will use the method the poster provides in the description. As a stretch goal the platform will generate a unique email for each post and any emails sent to it will be forwarded to the poster, this would protect user's anonymity however is

not a core feature. In addition no money is exchanged using the site and the website does not store or ship anything. We encourage face to face exchanges in public places to prevent a scamming attempts.

The server will be hosted on either the Edlab servers or on Amazon Web Services using node.js as our server technology. The database will run on the SQL server, most likely PostgreSQL. A users will only need a modern web browser to access the site (IE 9+, Google Chrome, Mozilla Firefox, Safari) or a mobile device. Our pages will follow the responsive template, and will scale down for mobile users to aid in the ease of use on smaller screen sizes.

## Performance Requirements:

**PER-01: Search page returns query results page in under 1 second**

The site needs to be responsive enough for the average user. If results take to long after a search they are more than likely to stop using the service.

**PER-02: No page (except search page) takes longer than 500 ms to load**

Similar to fast query results. The website needs to be seen by our users as soon as possible. This means the page is generated in 500ms of the user requesting the page, (query results above not included in this requirement).

## Reliability Requirements

**REL-1: Server can handle up to 100 concurrent users**

The service must be robust enough to meet performance requirements with 100 active users. This allows the service to be scalable for more users.

**REL-2: Server is available 99% of the year**
Like most Internet platforms the service must be accessible for users whenever they choose to use it. This requires the system to handle unexpected errors and still run.

**REL-3: User action and database state are consistent**
User actions should immediately reflect changes to their account and posts without any noticeable delay. This is to ensure users with posts that are removed or completed are not contacted afterwards.

## Use Cases:

### Formal Use Cases:

| | |
|---|---|
| **Goal** | Create a book listing (buy, sell, lend) |
| **Primary Actor** | Student |
| **Level** | User |
| **Precondition** | Student has an account |
| **Success End Condition** | Successfully creates a listing |
| **Failure End Condition** | Cannot create listing due to lack of input |
| **Trigger** | Student wants to sell/rent a book |
| **Success Scenario Steps** | 1. Visit web app<br>2. Sign in or sign up<br>3. Navigate to page for creating a listing<br>4. Fill out necessary product information<br>5. Create listing |
| **Extensions** | 2a) Wrong login info or can't confirm account<br>1. System returns to login page<br>2. System will not allow the user to continue to the next page unless required<br>2. System suggests password or username retrieval process<br>3. System suggests creating an account<br><br>4a) User fails to submit all necessary information to create a listing.<br>1. System will not continue nor return a message saying the listing has been created, instead it will wait until the user sufficiently fills out the form.<br>highlight areas that the user has not filled out sufficiently when the user tries to submit the listing.<br>2. All required information will have a black red asterisk next to it indicating that it must be filled out to post a listing.<br><br>5a) Listing cannot be posted due to a database or server error<br>1. System will not continue and will hang until the problem has been resolved by AWS, or the team members (restart server. etc) |

| | |
|---|---|
| | 1. System will return a failure message telling the user that an error has occurred and the user will be redirected to the previous page they were at (most likely the create listing page)<br>2.<br>3. System will send an email to the user indicating that the listing could not be posted due to an error and will ask the user to resubmit the listing |
| **Why is this important** | This is the fundamental function of this application that allows users to indicate their wish to sell, rent or buy a specific book. |

| | |
|---|---|
| **Goal** | Search |
| **Primary Actor** | Student |
| **Level** | User |
| **Precondition** | Student is on search screen |
| **Success End Condition** | Student finds the book |
| **Failure End Condition** | There is no matching listing |
| **Trigger** | Student needs a book |
| **Success Scenario Steps** | 1. Student visits the web app<br>2. System presents student with search screen<br>3. Student enters a search query<br>4. System matches the query and presents the matching listings<br>5. Student selects a listing<br>6. Student is presented with the seller/buyer/lender information |

| | |
|---|---|
| **Extensions** | 4a) System cannot recognize/find query<br>1. The system will not return any results which indicates that nothing has been found.<br>2. Student enters a different query<br>3. Student adds book to wishlist to be notified if the book becomes available<br>1. Student enters a different query<br>2. System returns a error message<br>3. System asks if the student mistyped the query and suggest other queries similar to the one inputted.<br>4. System automatically suggests other similar listings<br><br>5a) Listing is not what the student wanted<br>1. Student goes back to search results and tries a different listing.<br>2. Student tries a different query<br>3. System automatically suggests other similar listings<br><br>6a) There is no contact information<br>1. Student can report listing to flag it<br>2. System can send the listing owner an email if **Create a wishlist:**<br>3. The user sees a book s/he might need and clicks the "Add to wishlist" button on the listing page to add it to the wishlist connected to the user account. The user can also search for a book, and if there is no yield but the book is in the system, choose to add it to the wishlist and receive messages about matching listings in the future.<br>4. someone has flagged the listing. |
| **Why this is important?** | The reason why search is important is because without it the user would have to scroll through listings that does not pertain to their search query. The search function allows the user to refine the potential listings to fit the user's request. The search functionality also brings a complexity to the project as well as making the web app more user friendly. |

## Informal Use Cases:

**User creates an account:**

The user arrives at the web page and clicks on the "Create an account" link on the homepage. The page redirects to the "Create an account" page where the user must fill out an account creation form with required fields like name and email address, and institution. There are optional fields such as phone number and institution.

There are also optional fields like profile picture, phone number, and academic major.

**Edit or delete a posting:**

The user logs into their account and will be redirected to their profile page. The user can then scroll down to the section where it says 'Selling' and click 'Edit Listing' on the respective listing that they want alter. Once the user is redirected to the next page, the user has the option to either edit their listing or just delete the listing by simply clicking 'Delete Listing.' When the user searches the book or goes to their profile page again, the changes will be visible to other users and themselves. If the user does not confirm the changes, the original post will be left unchanged. If the user submits the changes or click 'Delete Listing' the listing will be changed in the database or deleted depending on which the user chooses to do.

The user logs into their account and clicks on the 'Current Listings' tab on the logged in homepage. The presented page shows the listings the user has made and provides options with editing, viewing, or deleting the listing. If the user chooses to edit the listing, they will be presented with the listing with ways to edit the post. Once the user finishes editing the listing, they will simply click a finish button to confirm the changes. The updated listing will now be available for others to view. If the user does not confirm the changes, the server will not update any changes and the original post will be left unchanged. If the user wants to delete the listing then a window will show up asking if the user is certain that he/she wants to delete it. Once confirmed, the listing will be deleted from the database.

**Rate a seller/buyer/lender:**

Once a user has finished an interaction with the post owner, the user has the option of rating that owner. The user has the option of giving the other person a rating between 1-5 stars and a written statement on their experience with the owner or the user can neglect to provide feedback. Once the user submits a rating the system will have a record of the submitted rating and average it out for the recipient of the rating. Once the rating is averaged out with the other ratings, the user will have an updated amount of stars shaded in on their profile. The written statement (if there was one) will also show up on the recipient's feedback page that is public to other users.

Why rating seller/buyer/lender is removed:

The reason why we decided against rating a user is because we could not find an area in the web app where the functionality could be implemented. UMass Bookshare does not do transactions but rather acts as a medium to connect students with each other. We believed that if we were to even add ratings then it would be based off an honor code which ultimately weakens the whole rating system anyways. A user cannot prove that a transaction happened so anyone could really just make false ratings/reviews.

**User adds book to wishlist:**

The user cannot find the book they are looking for on the web app so they decide to add that book by ISBN into their wishlist. The purpose of the wishlist is to have a list

of books that the user is currently looking for. Other users are able view the user's wishlist if they are able to find that users profile page. If a book becomes available in the system's database, the user who has that book on their wishlist will be notified that the book is now available. The user will then search for the book and will then contact the seller if they want to.

## **Feature List** - Expected delivery dates

| **Beta Release** | **April 1st, 2015** |
|---|---|
| Create Account | Create log-in and fill out an account information form (name, email, etc. with optional fields as well). |
| Create Listing | Filled out textbook and contact information for selling/renting |
| Search | Query by ISBN or match by title of book |
| Display Listing | Return a list matching the inputted query |

| **1.0 Release** | **April 29th, 2015** |
|---|---|
| Wishlist | Users can add books by ISBN onto their wishlist |
| Email Notifications | Users will receive an email when a book that is on their wishlist has become available. |
| View/Edit Profile | User can edit/view information on their account and other users can view a profile of another |
| Edit/Delete Post | Users can make changes to listings or delete them if needed. |
| Support Page | Support page will aid users if they are confused about any functions of the app |
| Encrypted Passwords | Passwords are not encrypted to aid in security |
| Book Images | When adding a book by isbn, the listing will now have an associated cover with it if it's available |

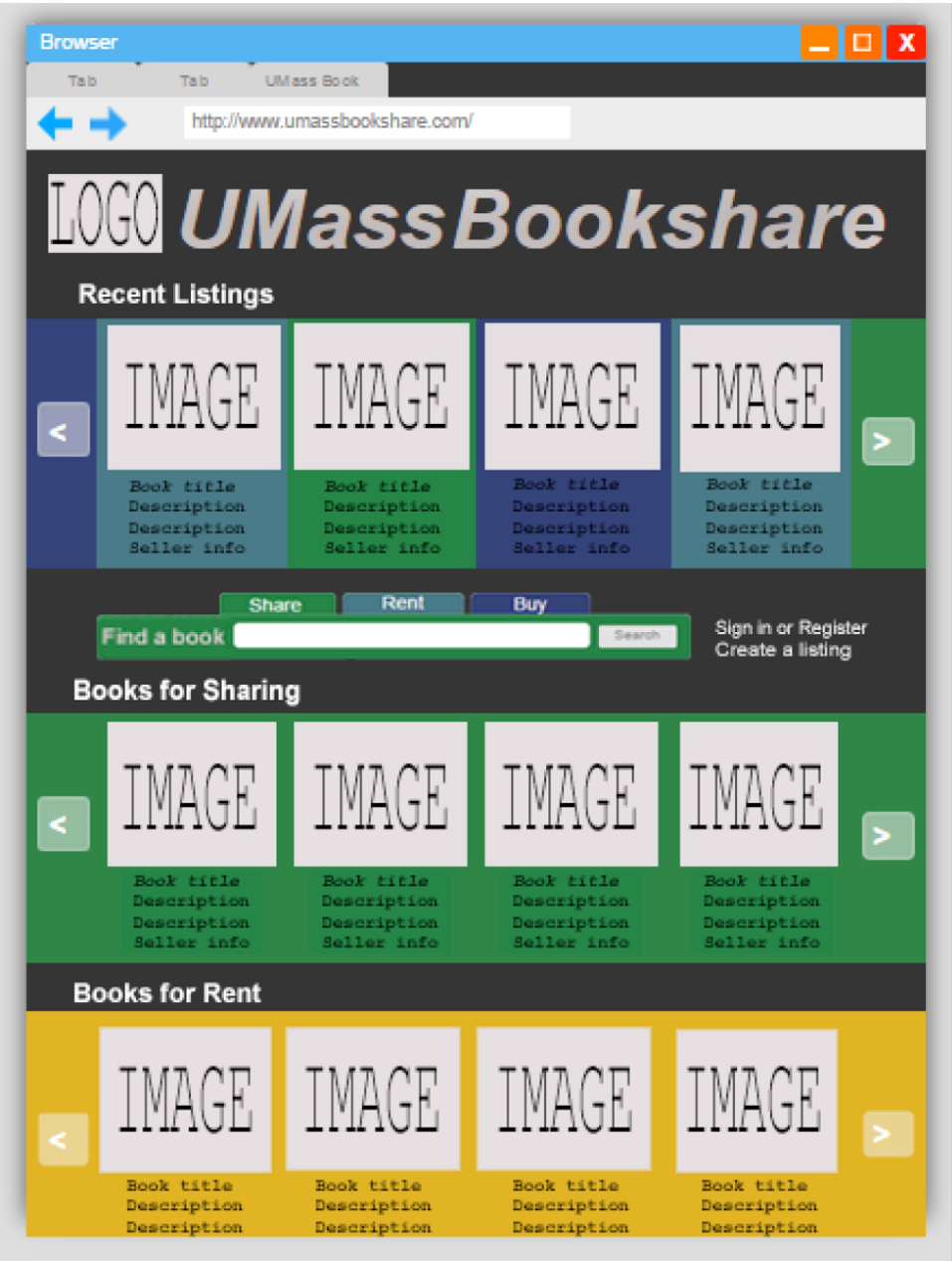
| | |
|---|---|
| Profile Picture | Users can add a profile picture by editing their profile |
| User History | View past books user has rented or bought |
| Display Recent Listing | New listings shown/updated on front of web app. |
| Notifications | Notifies users if posting has been flagged or book is available that is on their wish-list |
| Rate Users | Able to rate sellers, lenders, and buyers with a written statement. |

Why user history has been removed:

The reason why we are removing user history is because we did not find a purpose for it. Because UMass-Bookshare isn't meant to be a app that allows users to act as their own extensive book store, there is no reason for other users to view the user history. We do not expect users to make multiple transactions throughout a long period with the same user often so we believed that a user history would not be useful because interacting with a user is more of a one time event.

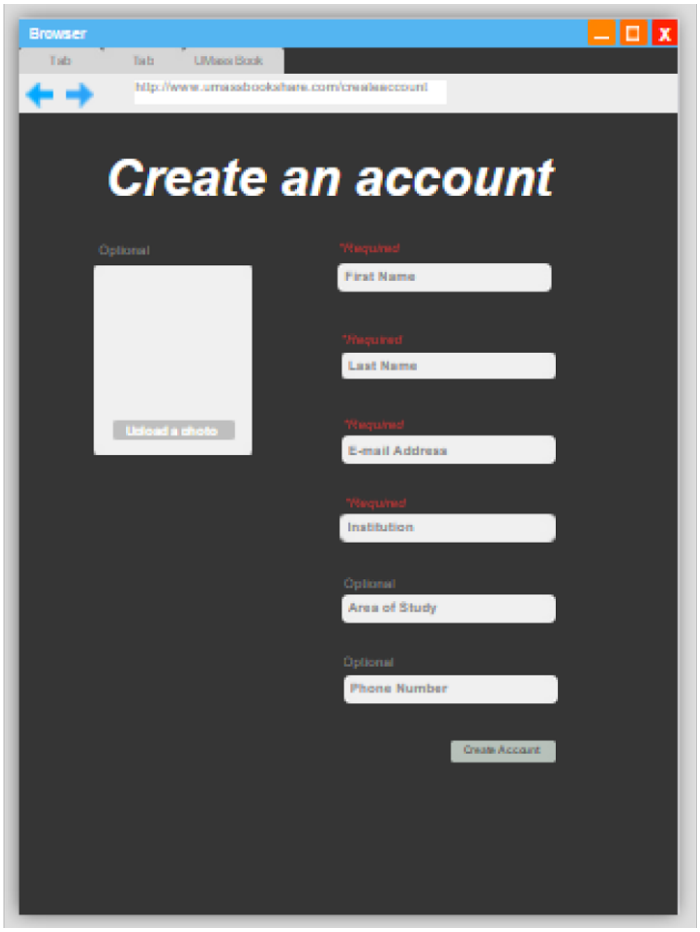
| **Stretch Goals** | |
|---|---|
| Category/Tags | Users can add categories/tags to books that are search-able or automatically given categories/tags from existing sources. Categories would be subject or major. |
| Link Classes | Textbooks are linked to classes, so by searching for a class or adding what classes you are taking, you be shown what textbooks are needed for those classes. |

**Mockups:**


Browser
Tab
Tab
UMass Book
http://www.umassbookshare.com/
LOGO UMassBookshare
Recent Listings
IMAGE
IMAGE
IMAGE
IMAGE
Book title
Description
Description
Seller info
Share
Rent
Buy
Find a book
Search
Sign in or Register
Create a listing
Books for Sharing
Books for Rent
Recent listings from all categories
Three rows of of listings for sharing, rent, and buy

Search detail panel
(Price range,
book category, etc.)


Listings with
image and
summary




Optional profile picture

Required fields (name, e-mail, institution)

More optional fields

# Use Case Diagram