Software Development Requirements

CookieNap

Bethany Hage, Kevin Tobias, Michael Purcell, Josh Kostal

Date	Changes	Version
9/15/2017	Initial Document	1.0
9/22/2017	Document with Group Review Changes	1.1
10/27/2017	Document with In-Progress Changes	1.2

Table of Contents

1. Introduction

1.1. Purpose

The purpose of this document is to explain the requirements and features of the CookieNap web application.

1.2. Scope

CookieNap is a mobile-first web application that will provide students at UNL with a safe and easy way to sell their unwanted textbooks. Users will be able to create and curate basic profiles and book listings. New listings will be inputted via a web form, which will have required fields for book ISBN, price, and condition. The ISBN will be used to automatically pull down the rest of the information needed for the book listing. Buyers and sellers will communicate via email. To protect our users' privacy, a user interested in a listing must request the email address of the poster.

1.3. Definitions, Acronyms, and Abbreviations

UNL- University of Nebraska-Lincoln
ISBN- system used worldwide to identify books based on a unique 10 or 13 digit code
MySQL- Relational database management system
AngularJS- Front-end web application framework
C#- Object-oriented programming language
Hash- algorithm used to turn a password into a long string of “random” characters
Salt- unique key that modifies the hashed password, different value for every user

1.4. Overview

The rest of this document will give the description and requirements of our web application, along with an appendix at the end. The description goes into detail about the intended use and purpose of the application. The requirements section covers the design and implementation of the application's structure and features. Finally, the appendix shows a few mockups of what some sections of the application will look like.

2. Overall Description

2.1. Product perspective

2.1.1. System Interfaces

2.1.1.1. CookieNap will not require any additional systems to run.

2.1.2. User Interfaces

2.1.2.1. Mobile Devices

The web application will be optimized for viewing on mobile devices.

2.1.2.2. Laptops/Desktop Computers.

CookieNap, although optimized for mobile devices, will still function well on laptops and desktop computers.

2.1.3. Hardware Interfaces

Users will be able to access the site using any internet-compatible device with a supported browser.

2.1.4. Software Interfaces

CookieNap will be available on most browsers, ideally the most up-to-date versions of Chrome, Firefox, Safari, and Microsoft Edge.
This application will use a MySQL database, provided by the CSE Department, to store our data. Our application will be hosted for free on Azure.

2.1.5. Communications Interfaces

There will be no in-app communication and no additional software required.

2.1.6. Memory Constraints

Our app will have a limited UI so it will not need much RAM.

2.1.7. Operations

Users can either login or signup for an account. They will then be able to post listings. To post these listings, users will fill out a web form with several basic fields.

2.1.8. Site Adaptation Requirements

There are no adaptations for any installations.

2.2. Product functions

The app’s only function is connecting textbook buyers with textbook sellers on the UNL campus.

2.3. User characteristics

Our users will be UNL email holders.

2.4. Constraints

One constraint is the amount of data that we can store on our MySQL database. This may result in automatic removal of the oldest listings, even if it’s before the standard removal date.

2.5. Assumptions and dependencies

Our web application depends on browser and language updates. These languages mostly refer to HTML, CSS, C#, AngularJS, JavaScript, and MySQL. In addition, we will depend on GitHub for our change management process (section 3.7).

3. Specific requirements

3.1. External interface requirements

CookieNap queries the Google Books API with the ISBN to get back additional information about a book.

3.2. Functions

3.2.1. Phase I

3.2.1.1. The interface will have three main pages: the home page, listing creation form, and the account registration/user login page.

3.2.1.1.1. The menu located at the top of each page will allow users to navigate from different pages in the app by clicking on the different options available.

3.2.1.1.2. The home page will show the active listings. Mobile and desktop models of this are included in the Appendix. Figure 4.1 is the mobile view, and Figure 4.2 is the desktop view.

3.2.1.1.2.1. Listings for users selling a textbook will contain a thumbnail of the book, it's name, condition, and asking price.

3.2.1.1.2.1.1. Listings older than 45 days will be automatically taken down.

3.2.1.1.3. The account registration and user login systems will share the same page.

3.2.1.1.3.1. The login system will require the user's Husker email and password.

3.2.1.1.3.2. The signup system will require the user's UNL email (for registration), secondary email (for communication), username, and password. The fields for first name and last name are optional.

3.2.1.2. Database

3.2.1.2.1. The database will contain two tables, User and Listing.

3.2.1.2.1.1. User will have fields for Username, FirstName, LastName, HuskerEmail, CommunicationEmail, Password, and an auto-incrementing UserId.

3.2.1.2.1.2. Listing will have fields for Price, BookISBN, Condition, LastEditedDate, IsSelling, an auto-incrementing ListingId, and foreign key User_UserId to link the tables.

3.2.2. Phase II

3.2.2.1. Email System

3.2.2.1.1. The email system will be able to send confirmation emails to newly registered users.

3.2.2.1.1.1. If confirmation is not received within 24 hours, the new account is deleted.

3.2.2.1.2. It will also send an email when a user requests the email address of a listing owner.

3.2.2.1.3. The email system will use the secondary email to communicate with the user.

3.2.2.1.4. The email system will also be used to help users reset passwords and recover their accounts.

3.2.2.2. GUI Improvements

3.2.2.2.1. We will implement a search and sorting system for better view of the available listings.

3.2.2.3. Custom Thumbnail Image

3.2.2.3.1. Sellers will be able to add their own custom thumbnail image depicting the condition of the item instead of using the default image pulled from our ISBN query.

3.2.3. Future Phases

3.2.3.1. Expansion to Other Schools

3.2.3.1.1. Eventually, CookieNap could be expanded to other schools

3.2.3.1.1.1. This would simply require the addition of each new school's email address suffix (eg. @huskers.unl.edu) to our registry.

3.2.3.1.1.2. Would have to create separate market pages for each school, as well as adding the school to the listing database table.

3.2.3.2. Online ordering

3.2.3.2.1. CookieNap could add the option to buy and sell through the site and then ship the book. Users would do all shipping themselves.

3.2.3.2.2. This could work well with the school expansion, as it would more easily allow students of different schools to buy/sell to each other.

3.3. Performance requirements

CookieNap should run smoothly when utilizing the different functions. The user should not have any problem navigating to different sections of the application.

3.4. Logical database requirements

The database of books available to buy will be updated automatically as users input data into this section. This data will be inputted using prepared statements to prevent SQL injection. User information, as well as book information, will be kept in this secure database. Refer to section 3.2.1.3. for a description of these specific tables and their relationships.

3.5. Design constraints

Because CookieNap will be a mobile-first application, the main design constraint will be a user friendly interface on a computer screen. The application will have to work on the smaller resolution of a phone as well as the larger resolution without sacrificing performance.

3.6. Software system attributes

3.6.1. Reliability

The application should be reliable enough to stay up constantly.

3.6.2. Availability

The application should be available 24/7.

3.6.3. Security

Any information provided by the user (email, name, etc.) will be kept in a secure database and will not be used or given to any outside source. Passwords will be required to have a length of at least 7 characters, and hashed and salted after being creation. The system will be secured against SQL injections using prepared statements.

3.6.4. Maintainability

Since we will follow strong coding procedures (e.g. comments and readable code), and the code base will not be large, new features and bug fixes will be simple.

3.6.5. Portability

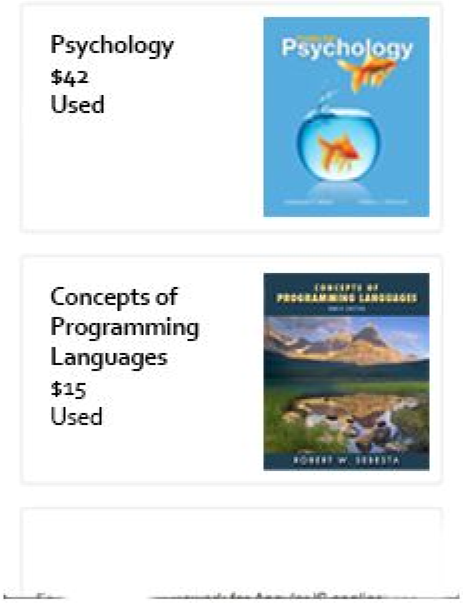
The application will be available simply through a web browser, so it will be accessible from anywhere and any internet-connected device.

3.7. Change Management Process

We will be utilizing GitHub for source control because it will be the best way for our team to collaborate and publish changes, as well as a centralized place to view our finished code.

4. Appendix

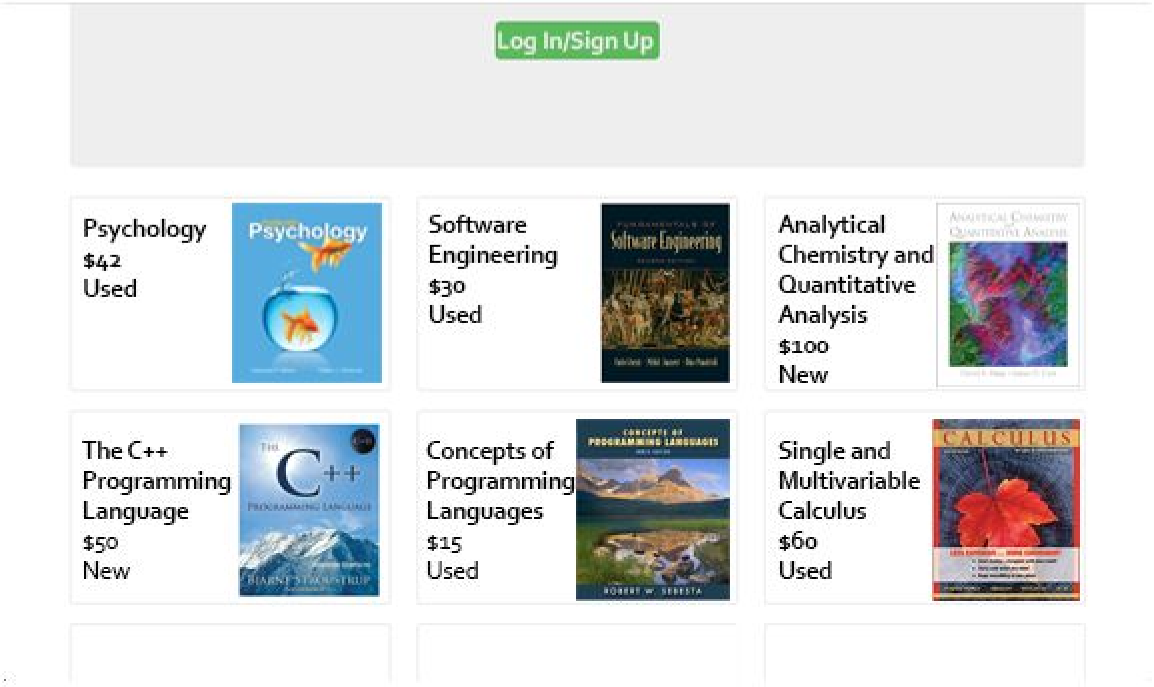
4.1. Phase 1 Home Page Layout- Mobile Version



Psychology
$42
Used

Concepts of Programming Languages
$15
Used

4.2. Phase 1 Home Page Layout- Desktop Version



Psychology
$42
Used

Software Engineering
$30
Used

Analytical Chemistry and Quantitative Analysis
$100
New

The C++ Programming Language
$50
New

Concepts of Programming Languages
$15
Used

Single and Multivariable Calculus
$60
Used