Brisk Publishing
Book Management System
Phase One

25 November 2021
pamrosel.is@gmail.com

Contents

Hardware Requirements	»	SQL PHP server
	»	2 CPU Cores (minimum)
	»	2 GB RAM (minimum)
Software Requirements	»	node.js
	»	express.js
	»	express-session
	»	Node MySQL 2
	»	node.bcrypt.js
	»	validator.js
Functional Requirements	»	Session id management
	»	Login authentication
	»	Password hashing encryption
	»	User types: admin and staff
	»	Admin can create and update user info, staff cannot
	»	Index of Books
	»	Ability to add, update, delete books
	»	Index of Authors
	»	Ability to add, update, delete authors
	»	Index of Users
	»	Ability to add, update, delete users
	»	User change log for when a book is created and last updated
	»	Sanitized form inputs plus client-side input validation
	»	Responsive design on mobile
	»	Logout at anytime

The Client

Brisk is a medium-size publishing company interested in developing a new application for their internal management of a catalog that indexes their 'most popular books of all time' and information about their related authors. They have two office/warehouse locations, one in north and south Brisbane. Admin and staff users of this system must be able to view, update and delete book records in a way that logs their last updates, and keep track of the different users using the system along with their system information and access rights.

This build marks 'phase one' of development, which focuses on establishing core functionality, with the intent to build in further functionality in future phases.

Target Audience

The main users of this build are admin and staff users at Brisk.

Target Audience Needs

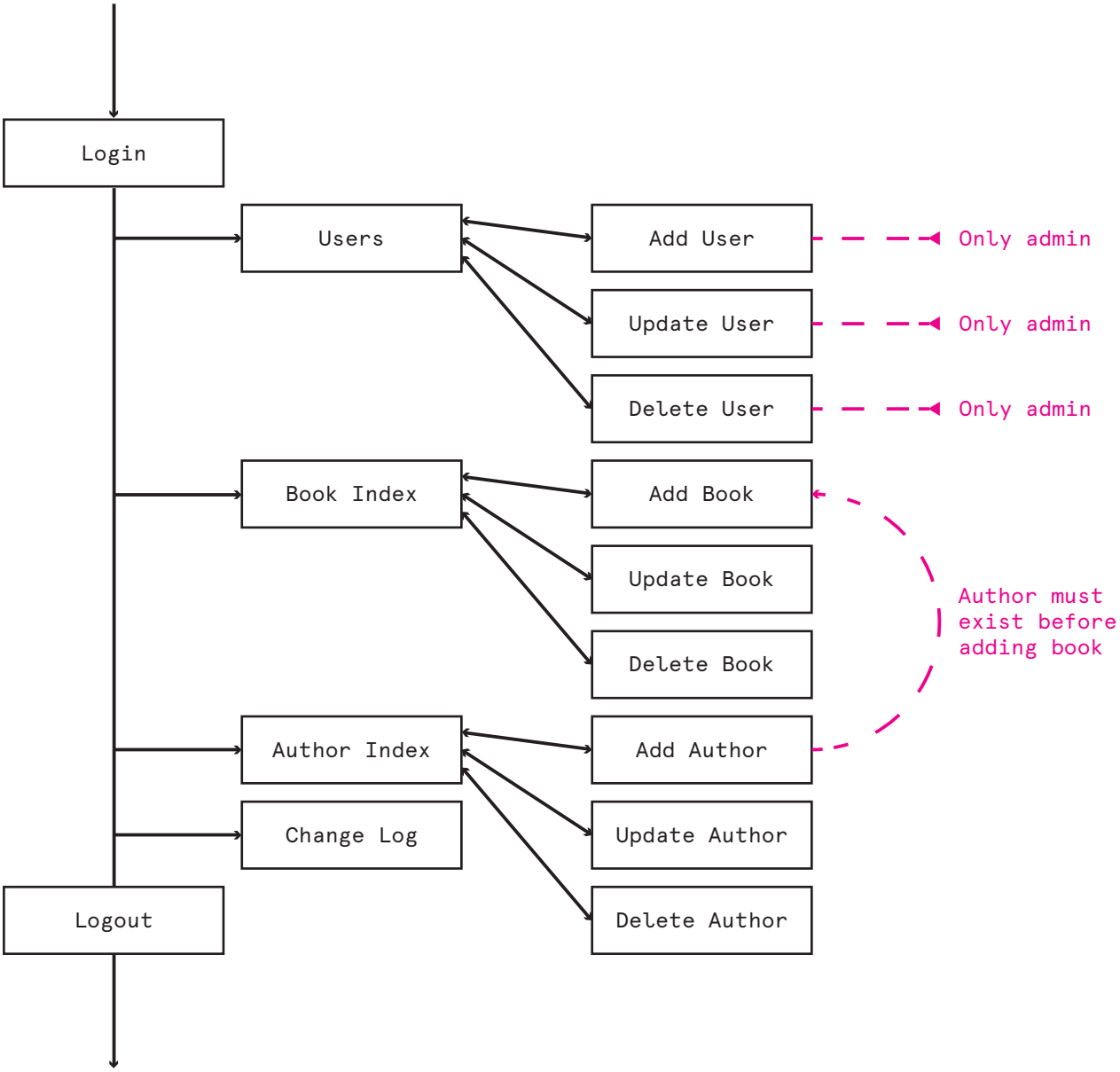
- Adheres to company branding guidelines
- Clear site navigation
- Simple, flexible UI/UX design for future development
- Use across mobile devices from any location
- Communicate clear session management (i.e. 'logged in as *username*')
- Appropriate confirm and error messaging
- User accountability
- Appropriate security measures

Technologies Used to Overcome the Stateless Nature of HTTP Protocol

This build enforces stateful sessions so that unique users can access the site to perform changes, be accountable for those changes and within the limits of what they are authorised to do. This is done using the express-session package that enables session middleware, where session data is stored server-side when a user logs in. Post login, we can request information about the session user when needed, before destroying the session upon logout.

We also utilise Fetch to grab data from the database and post it client-side by way of fulfilling request/response promises. It is able to update the DOM asyncronously, carrying JSON data to and from our database to fetch, post and update information about books, authors and users in our application.

Detailed Sitemap



Content Inventory	To view the content inventory, please visit this link to Google Sheets: *https://docs.google.com/spreadsheets/d/1bs9EZLEoZWpbcp9fdgKEqikMQ4HezXkwtCFptH1LpJs/edit?usp=sharing*
Test Reports	To view the test reports please visit this link to Google Sheets: *https://docs.google.com/spreadsheets/d/10iBRJak-pS-qx1Q4WCXI2pIMdTdDMzcomegPWuFN570/edit?usp=sharing*