

UMass BookShare

Software Design Specification

Presenters: Walter Doan, Peter Chau, Bianca Tamaskar

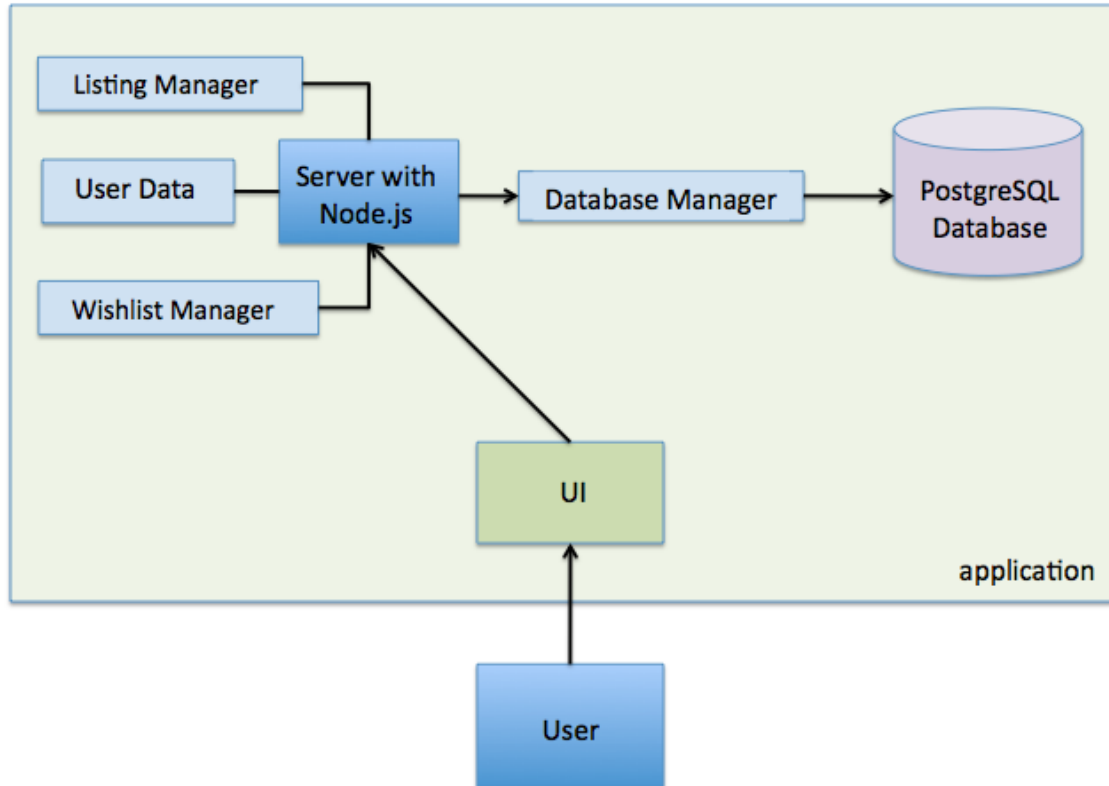
What is UMass BookShare?

- Web application
- Online marketplace for students to buy, sell, rent, or lend textbooks
- Convenient and sustainable

How will it work?

- Users can post books they are willing to share
- Users can search for books to see who has them
- Communication/payment through 3rd party
- Wishlist feature

Software Architecture

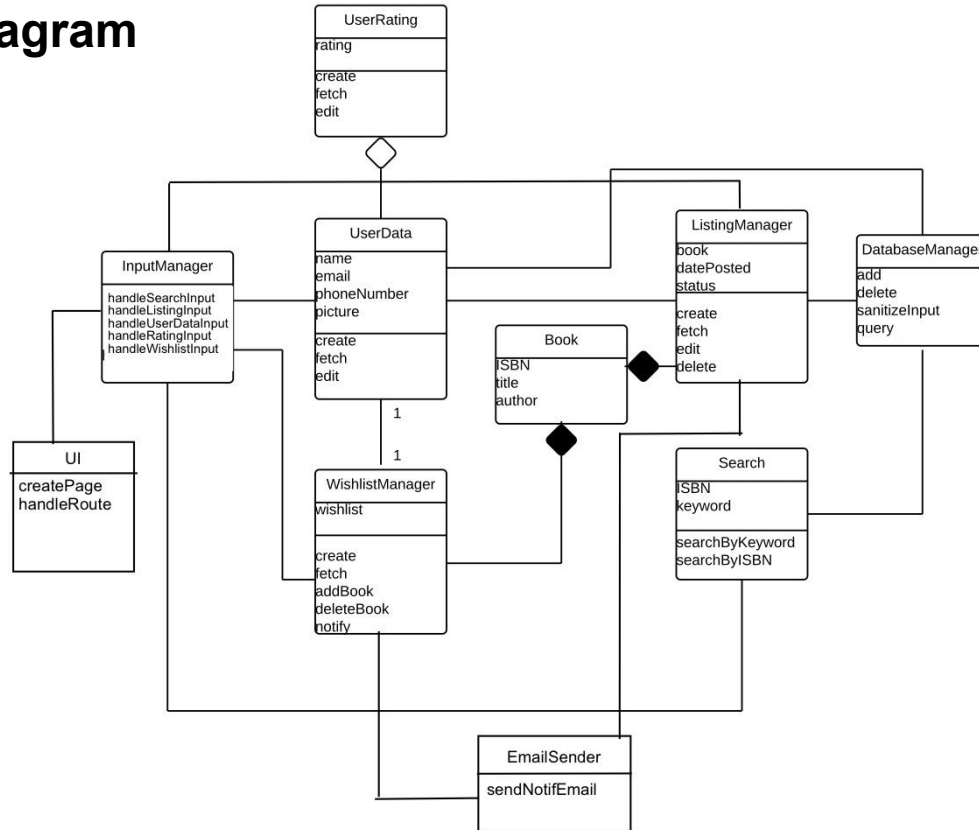


At a high level...

Model-View-Controller
Client-Server

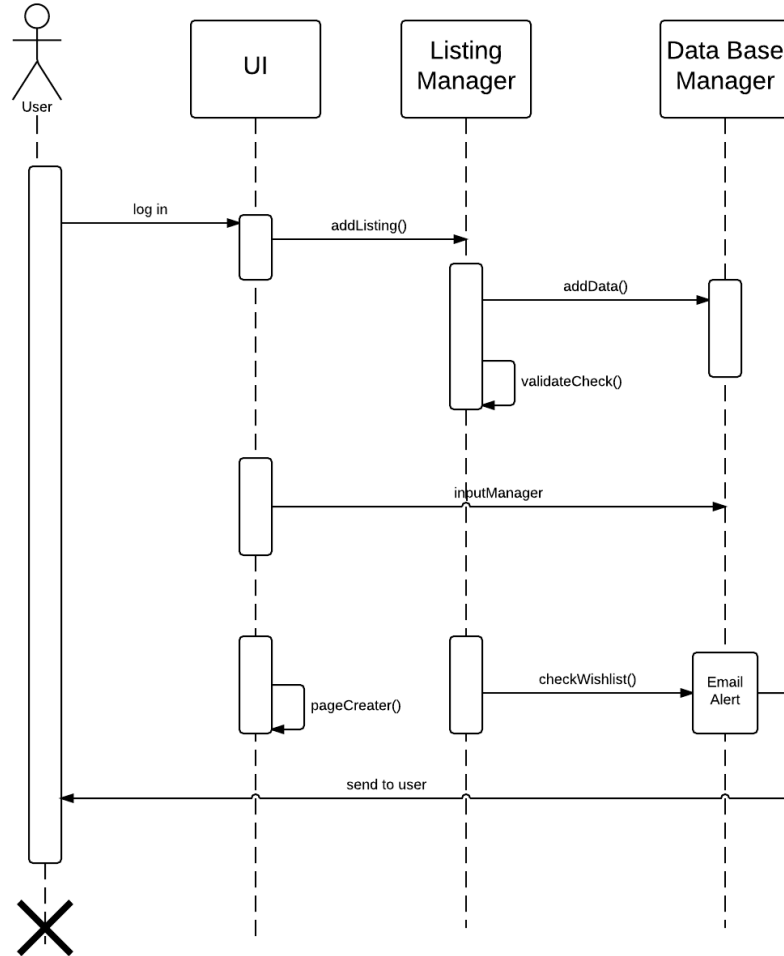
Software Architecture

UML Class Diagram

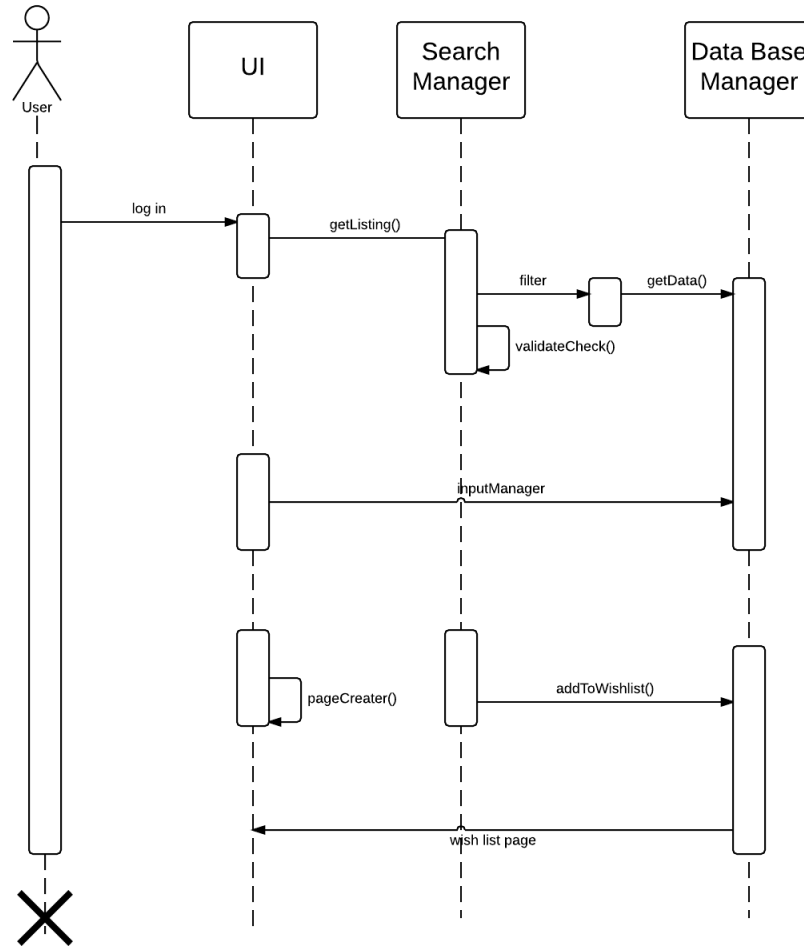


Use case sequence diagram: Listing

Creating a listing...



Use case sequence diagram: Searching



Searching for a book...

Database



- Relational database
- Main tables:
 - User (PRIMARY KEY username)
 - Book (PRIMARY KEY ISBN-13)
- All other tables references these
- Other tables: Wishlist, Rating, Listing

Design Alternatives

- We chose not to include chat or Private Messaging.
- We chose to use Node.js instead of Apache.

Team Structure/Responsibilities

- **Project manager: Chris**
 - organize meetings, reviews drafts, works with customer
- **Testing - Walter, Peter, Amy**
 - develop test strategy for automated and manual tests
- **UI - Amy**
 - mock-ups, and UI design
- **Database - Piers**
 - develop the database schema, configure development environment to work with the database correctly, and database interaction in the system architecture
- **System architecture - Qiwen, Bianca**
 - plan and design of system architecture, and server-side coding in alpha release

Communication

- **Slack**
 - documentation & resources
- **Email**
 - planning
- **Meetings**
 - USpace
- **Google Docs**
 - documentation & presentations
- **GitHub**
 - source control



Project Schedule

Alpha Release (March 9)

- Initial db
- Server
- Site structure w/ navigation flow

Beta Release (April 1)

- Initial GUI for web app
- Account Creation
- Create/Display Listing
- Search
- GUI talks to db



Project Schedule cont.

1.0 Release (April 29)

- Continued GUI development
- Filter
- Wish-list
- View/Edit Profile
- Edit/Delete Post
- User History
- Display Recent Listing
- Notifications
- Rate Users
- Testing/Debugging

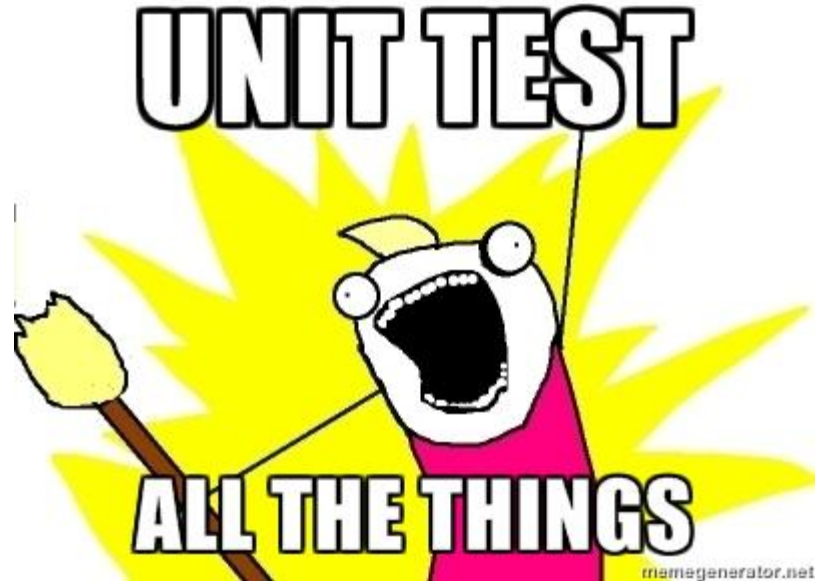


Risk Assessment

- **Project schedule inaccuracy (High)**
 - divide & conquer
 - overestimate
- **Learning curve for skills (High)**
 - ask for help
 - tutorials
- **Communication with team members (Med)**
 - meet often
- **Permissions from institution or other applicable sources (Med-Low)**
 - request permission
- **Customer satisfaction (Med-Low)**
 - customer requirements

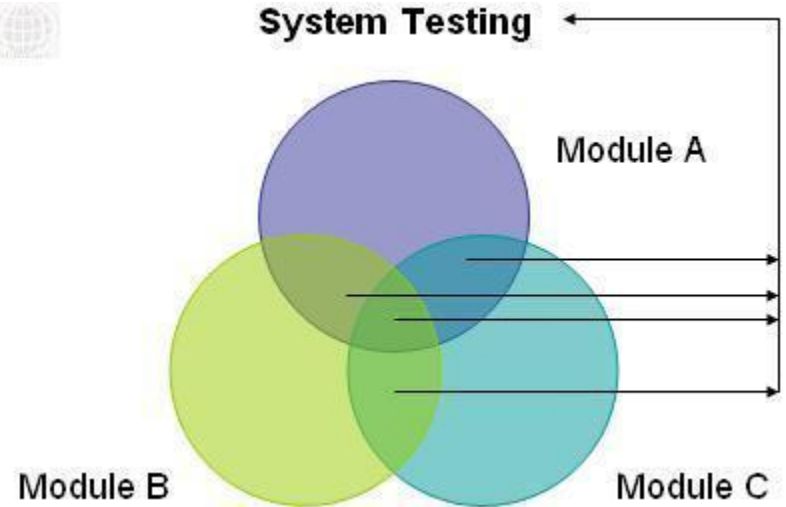
Unit Tests Strategy

- **Form Entries**
 - Username
 - Password
 - Email
- **Post Listing**
- **Wish-list**



System Test Strategy

- **Button Functionality**
 - Search Button
 - Submit Button
- **Database**
 - Search
 - Entry
- **Email Notification**
 - Item added matches wish-list
 - Verification of account creation



Usability Test Strategy

- **Browser Compatibility**
 - CSS
 - Mobile
- **Navigation**
 - Test links
- **Error Messages**
 - Page does not exist
 - Form submit fail
 - Search found no matches



Usability Test Strategy cont.

- Easy to use
 - User friendly
- Aesthetically pleasing
- Feedback time



Adequacy of test strategy

- Github Issue Tracker
 - Version control
 - Collaboration
 - Check for changes
- Trac
 - Track, report, fix bugs
 - Keeps teammates updated
 - Write and use reports
 - Tickets can be created to report bugs
 - Detailed reports on bugs

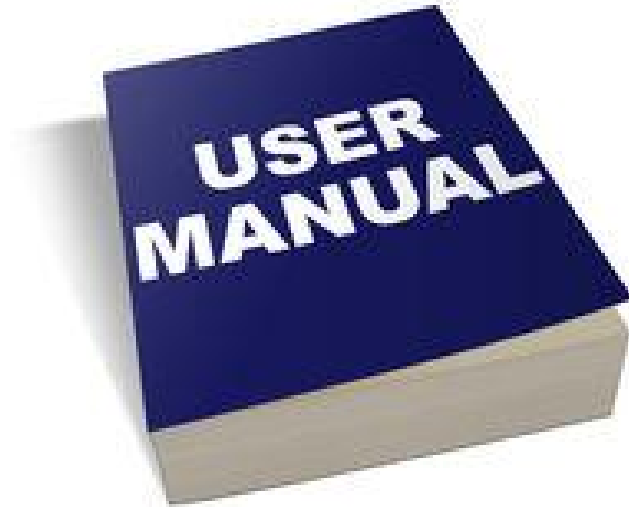
GitHub



Documentation Plan

User Guide w/ Pictures

- How to Login
- New User Registration Steps
- Forget Password Help
- Update User Profile
- Create/Edit/Delete Posting
- Add/Remove items from WishList
- Search/Filter/View Postings
- Rate Users



Documentation Plan Cont.

Admin Guide

- Overview of System
- Installation/Compile

Help Menu in Web App

- Contact Us
- FAQ

