Documentation

Library Management System

Authors
Charles Brady
Jeremy Hudson

Project name:
Library Management System

Team name:
Librarians of Alexandria

Date:
5-3-2019

Documents and Authors:
Requirements Document – Jeremy Hudson and Charles Brady
Normal use-case Diagram – Charles Brady
Error use-case Diagram – Jeremy Hudson (Diagram by Charles Brady)
Architectural Diagram – Charles Brady
UML Diagram - Jeremy Hudson

Table of Contents

1 Introduction

1.1 Purpose of the Requirements Document

The purpose of this system requirements document is to define the functionality of the Library Management System and how that functionality will be provided.

1.2 Scope of the Product

The scope of this Library Management System includes easing book and customer management for Librarians, and to create a convenient application for librarians to better manage the books within their library. The system also allows librarians to add books to the library by search terms with the Google Books API to get the book's title, author, category and an image of the book.

1.3 Definitions, Acronyms, and Abbreviations

Librarians are those with the authorization to view and modify the information contained within the Library Management System. The can add books to the library, create new librarians, search the database and check books in and out of the library for customers and other librarians

Customers are those who are registered in the Library Management System, have a library card, and can search the database.

The term "Users" refers to both Librarians and Customers.

1.4 References

Google Books API information is available at http://developers.google.com/books/.

1.5 Overview of the remainder of the document

The remainder of this document contains the following information:

- General description of the product
- Access capabilities of Librarians and customers
- Assumptions
- Functional, non-functional and interface requirements

2 General Description

2.1 Product Perspective

A library management system is used to manage a library. It uses a database to maintain book, customer, and checked out books information. This information is managed by librarians. It also calls the Google Books API to receive new book information that is then added to the local SQL database.

2.2 Product Functions

The Library Management system has functionality for Customers and Librarians.

The customer should be able to register to use the system, print their library card and search for books in the library database.

The librarian should be able to obtain access to customer contact information by a uniquely generated serial number, register a new librarian, check books out for a customer, check books back into the library, and add books to the library via the Google Books API. The librarian should also have the ability to search the local SQL database for books.

2.3 User Characteristics

Customers are those who access the library to obtain books.

Librarians are staff who maintain the library.

2.4 General Constraints

Customers will not have access to Librarian-specific functions, or another customers' information. Customers cannot checkout books for themselves. Customers cannot check-in books for themselves.

2.5 Assumptions and Dependencies

- Assume, by default, the user reads English.
- Assume the user understands basic internet navigation.
- Assume there is only 1 copy of each book in the library so that the book's ISBN can be used to identify the book for check in and check out purposes.

3 Functional Requirements

3.1 Requirements

ID	Requirement Description	User	Priority
F-1.	Maintain a list of available books, including title, author, ISBN, category and image.	N/A	High
F-2.	Allows users to log into the system.	Customers Librarians	High
F-3.	Allows Customers to register for access, providing a name, userID, password, and email address.	Customers	High
F-4.	Allow Customer to print their library card	Customers	High
F-5.	Allow users to search for books in the Library by title.	Customers Librarians	High
F-6.	Allow users to search for books in the Library by author.	Customers Librarians	High
F-7.	Allow users to search for books in the Library by ISBN.	Customers Librarians	Low
F-8.	Display the results of the search, including title, author, and category.	Customers Librarians	High
F-9.	Allow Librarians to check out a book(s) to a specific Customer by scanning a barcode on the customer's library card and the book(s).	Librarian	High
F-10.	Allow Librarians to check in books that have been returned.	Librarian	High
F-11.	Allow Librarians to register another librarian for access to the system by providing a userID, password, and name.	Librarian	Med
F-12.	Allow Librarians to add a book to the system by entering the ISBN or scanning an ISBN barcode, and having relevant book information added to the library database from a Google Books lookup.	Librarian	High

3.2 External Interface Requirements

3.2.1 User Interface Requirements

ID	Requirement Description	Priority
U-1.	Web Browser	High

3.2.2 Software Interface Requirements

ID	Requirement Description	Priority
SI-1.	Google Books API	High
SI-2.	Barcode Scanner API (Barbeque)	High
SI-3.	MySQL Database API	High
SI-4.	JSON API	High

3.2.3 Hardware Interface Requirements

ID	Requirement Description	Priority
H-1.	Device that can operate a Web Browser	High
H-2.	Barcode Scanner	High
H-3.	Printer	High

3.2.4 Communication Protocols for Interfaces

ID	Requirement Description	Priority
C-1.	HTTP	High

4 Non-Functional Requirements

4.1 Infrastructure Requirements

ID	Requirement Description	Priority
I-1.	Web interface allows library customers and librarians to access the system.	High

4.2 Performance Requirements

ID	Requirement Description	Priority
P-1.	The system should respond to requests within 3 seconds	High

4.3 Security Requirements

ID	Requirement Description	Priority
S-1.	Users must log in with an UserID and password.	High
S-2.	Only Librarians have access to check out books, check in books, add books to the library, or register other librarians for access.	High

4.4 Availability Requirements

ID	Requirement Description	Priority
A-1.	The system must be available when the library is open, 8am – 10pm ET, daily.	High
A-2.	The system may be taken offline 1am-5am ET on Sundays.	Med

4.5 Reliability Requirements

ID	Requirement Description	Priority
R-1.	The system should return results that correspond to search criteria.	High

4.6 Scalability Requirements

ID	Requirement Description	Priority
SC-1.	The system must be able to support 1000 books.	High
SC-2.	The system must be able to support 100 users.	High

4.7 Maintainability Requirements

ID	Requirement Description	Priority
M-1.	The system will be developed using Java, web services, and MySQL database.	High

4.8 Backup / Disaster Recovery Requirements

ID	Requirement Description	Priority
DR-1.	GitHub provides backup and disaster recovery.	High

4.9 Demographic Requirements

ID	User Type	Device Type	Location	Usage	# Users
D-1.	Customers	Workstation or mobile device	Anywhere	Customer Features	100
D-2.	Librarians	Workstation	Library	Librarian Features	10

4.10 Training Requirements

ID	User Type	Requirements Description
T-1.	N/A	The system is intuitive and does not require training.

5 Appendix

5.1 Revision History

Version #	Date of Change	Summary of Change	Author
1.0	2/11/2019	System Overview	Jeremy Hudson
1.1	2/12/2019	Added requirements	Chip Brady
2.0	5/3/2019	Finished the first build	Jeremy Hudson

Library Management System Use Case

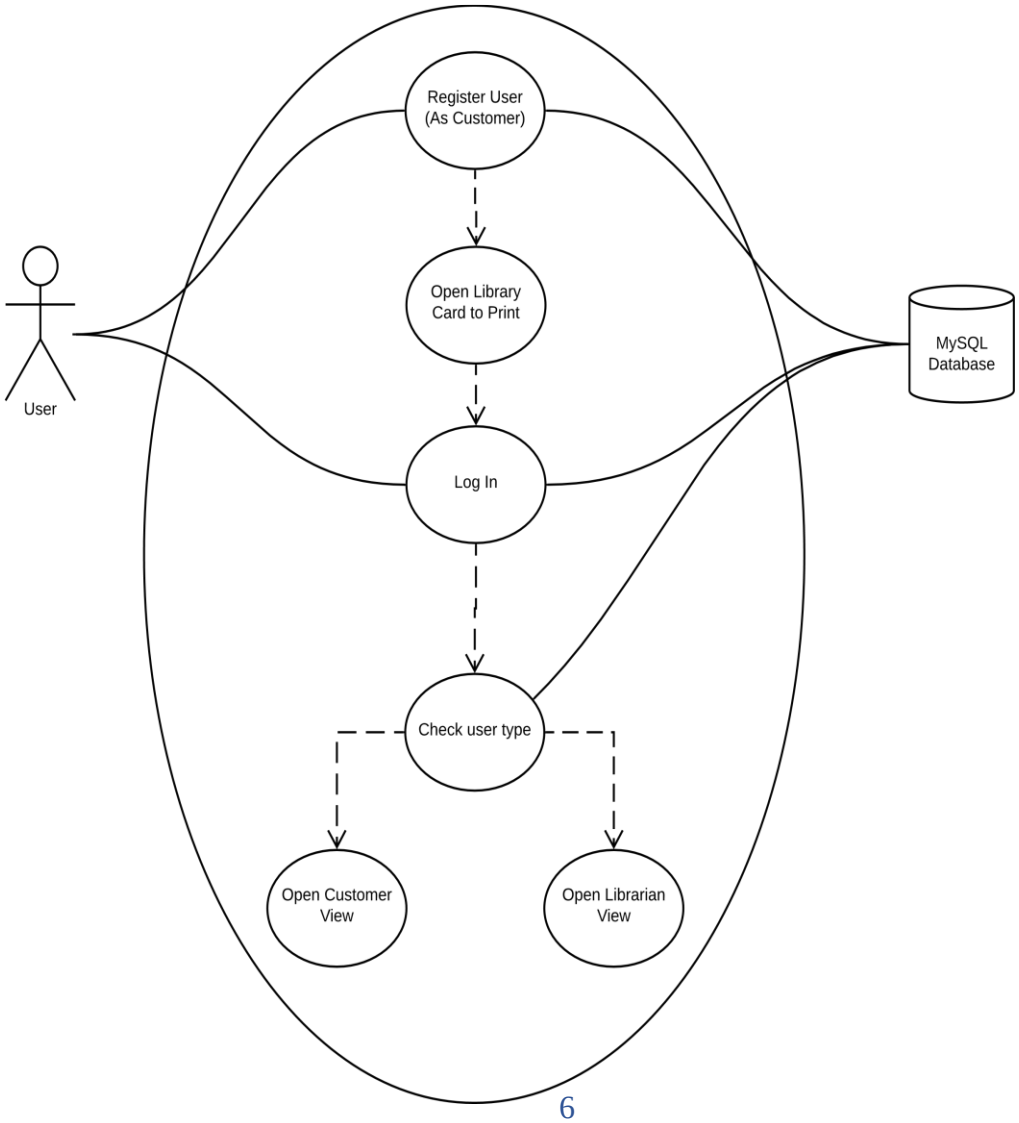
Charles Brady

Jeremy Hudson

Title: Log into Account

Actor(s): User

Purpose: Show successful login attempt.



Scenario 1

PreCondition(s): User has started the application, and is in the main menu display.

User has an existing account, but is not yet logged in.

Procedure: Click login button

Enter User Name and Password.

Click login button

Application confirms user is in the database

Application checks if the user's type is customer or librarian

Application opens the view for the corresponding user type

PostCondition: User is logged into the application.

Scenario 2

Pre Condition(s): User has started the application, and is in the main menu display

User does not have an existing account

Procedure: Click register button

Enter Name, User Name, Password, and Email.

Click register button

Application puts user information in the database with user type customer

Application opens library card

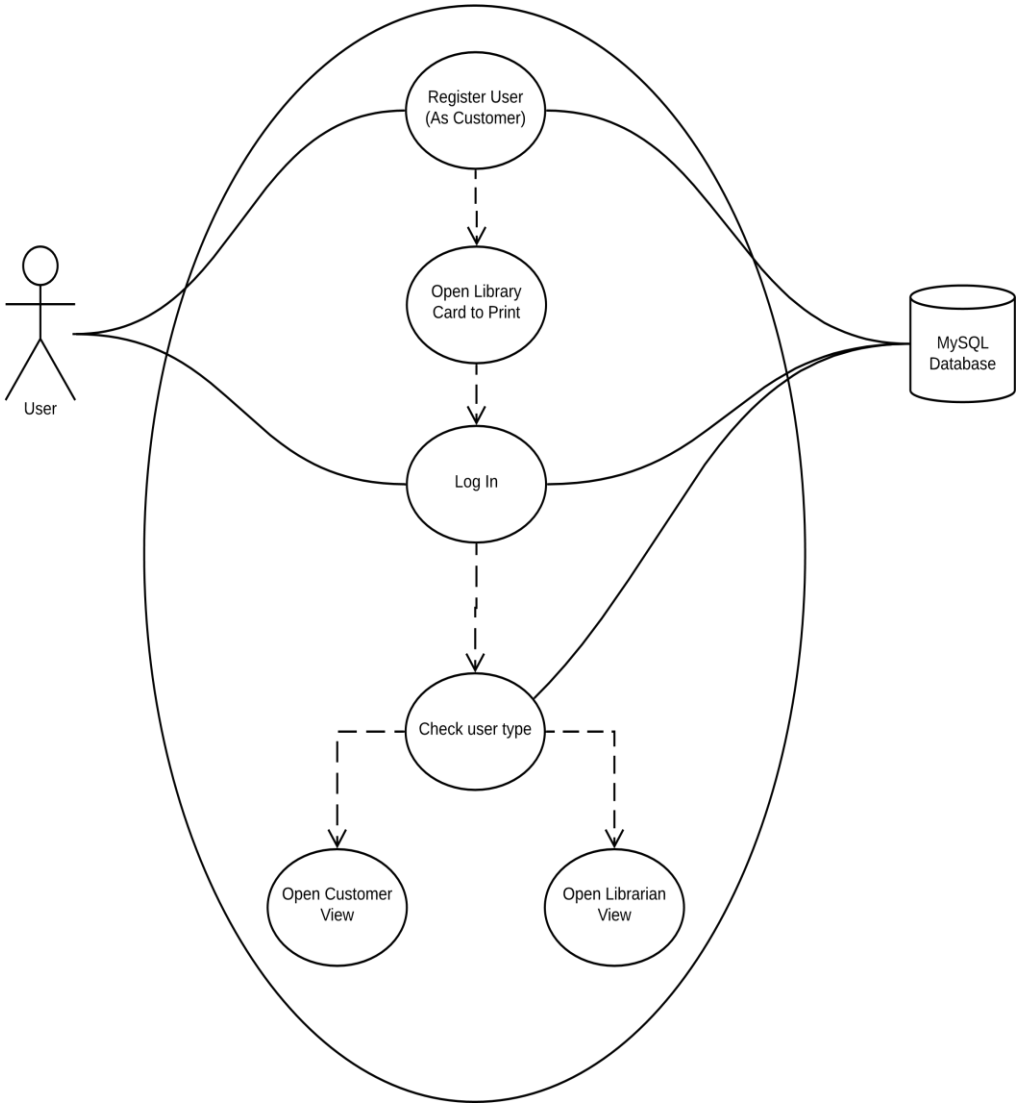
Post Condition: User is added to the database

Library Management System Error Use Case

Charles Brady

Jeremy Hudson

Title:	Register an Account
Actor(s):	User
Purpose:	Show unsuccessful registration attempt.



Scenario 1

PreCondition(s): User has started the application and is at the main menu display.

User does not have an existing account, and has not logged in.

Procedure: Click Register button

User attempts to enter an email address without an '@' symbol.

User clicks register button.

Application attempts to put user information in the database with the user type customer.

Application sees the customer has entered an invalid email address.

Application displays that it is unable to successfully create the account.

Post condition: The user is not successfully registered.

Scenario 2

Precondition(s): User has started the application, and is in the main menu display

User already has an existing account and has not logged in.

Procedure: Click register button

User enters an already existing name, username, password, and email.

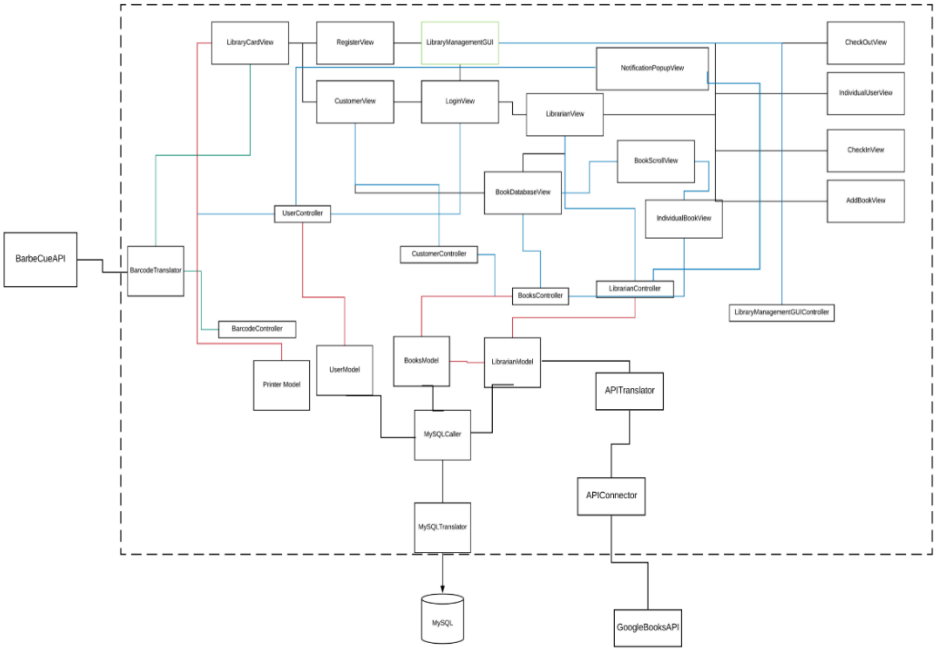
User clicks register button

Application attempts to put user information in the database with the user type customer.

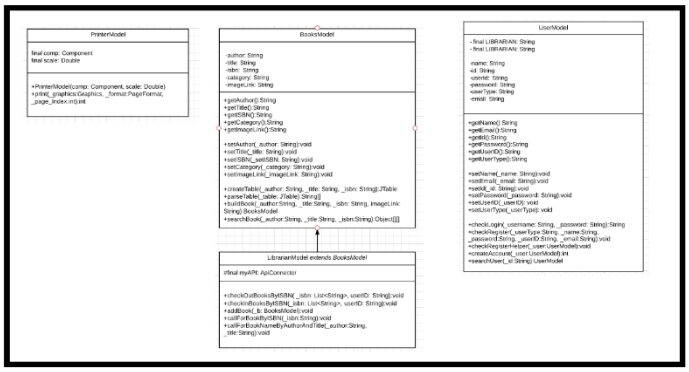
Application sees that a user with this information already exists.

Application displays an error and tells the user to try again.

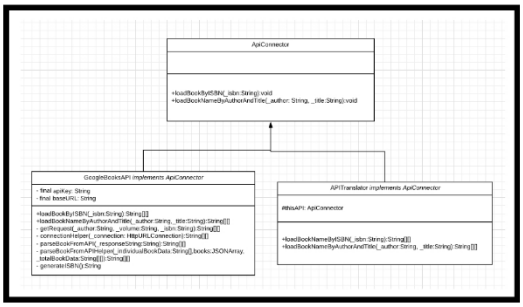
Post condition: A duplicate user is not added to the database.


LibraryCardView
RegisterView
LibraryManagementGUI
CheckOutView
NotificationPopupView
IndividualUserView
CustomerView
LoginView
LibrarianView
BookScrollView
CheckInView
BookDatabaseView
AddBookView
UserController
IndividualBookView
BarbeCueAPI
BarcodeTranslator
CustomerController
LibrarianController
BooksController
LibraryManagementGUIController
BarcodeController
BooksModel
LibrarianModel
UserModel
Printer Model
APITranslator
MySQLCaller
APIConnector
MySQLTranslator
MySQL
GoogleBooksAPI

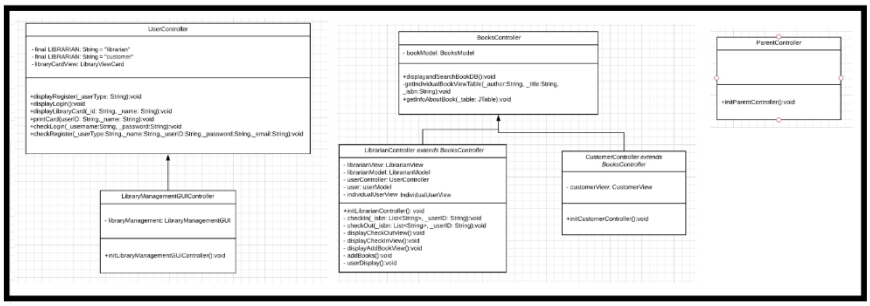
Models:



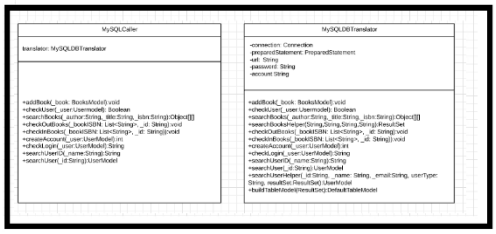
API:



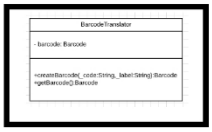
Controllers:



SQL:



Barcode:



Views: