

Project LIB OPAC

Analysis Model

Submitted to:

Asst. Prof. Ma. Rowena C. Solamo
Faculty Member
Department of Computer Science
College of Engineering
University of the Philippines, Diliman

Submitted by:
dela Cruz, Francis Zac
Jurquina, Angelo
Pangarungan, Nicko

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2017-2018

Revision Control

History Revision:

<i>Revision Date</i>	<i>Person Responsible</i>	<i>Version Number</i>	<i>Modification</i>
10/27/17	Nicko Pangarungan	1.0	Initial Document

Purpose:

This document defines the informational, functional, and behavioral domain of the system (Solamo, 2006).

Audience:

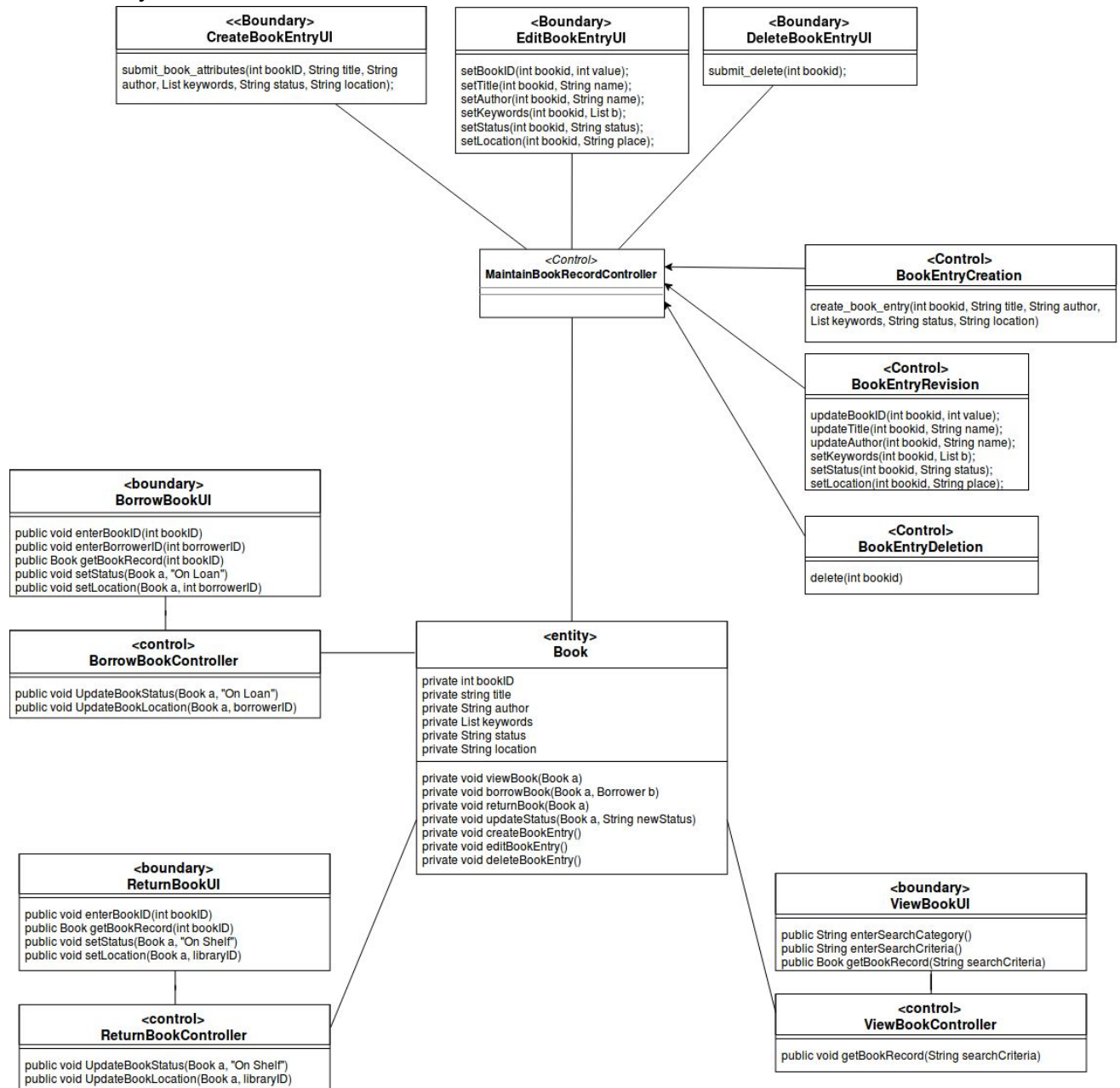
Project developers, clients, system users.

System Name: Project ILIB OPAC

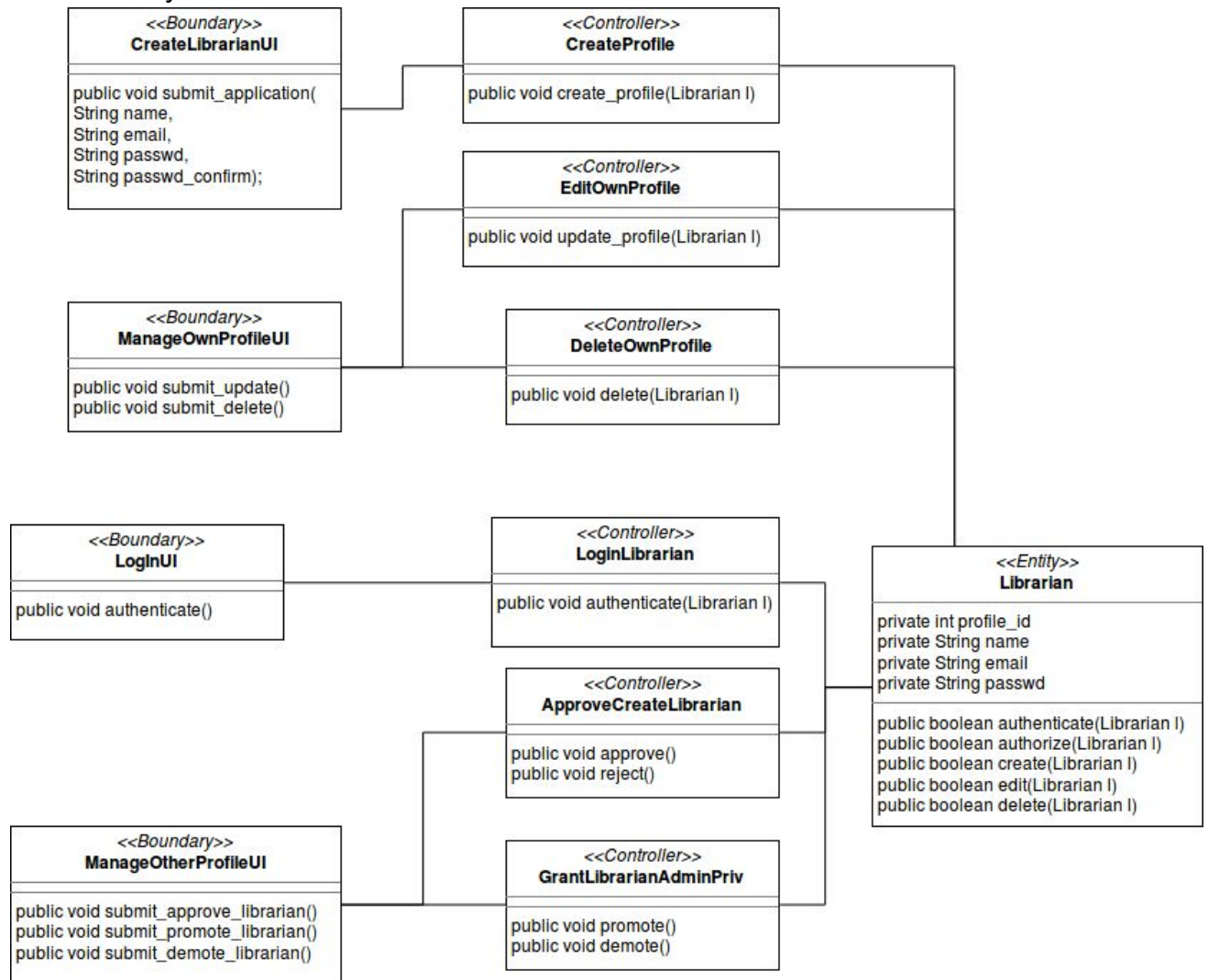
Description: This system aims to track and maintain the status of a library of books as it is borrowed from and updated.

Class Diagram:

Book Subsystem



Profile Subsystem



Boundary Classes:

Class Name	Description
CreateBookEntryUI	User interface used by system administrators to build the book database. Let's the admin create an entry in the database by providing book details such as title, author, keywords, status, location.
EditBookEntryUI	User interface used by system administrators to edit existing book entries in the database. The UI allows admins to edit any field of the book entry description.
DeleteBookEntryUI	User interface used by system administrators to delete existing book entries in the database.
ViewBookUI	User interface that lets admins, librarians, or children view book records and book entries. The interface allows users to choose a category to search books by such as title, author, or topic/keyword. The request is then passed to the appropriate controller.
BorrowBookUI	User interface that lets admins or librarians borrow books upon the request of a borrower. The user inputs identifying details of the book to be borrowed and the borrower's ID.
ReturnBookUI	User interface that lets admins or librarians return books upon the return of a book by a borrower. The user inputs the identifying details of the book to be returned to the library and the ID of the one who borrowed the book.
CreateLibrarianUI	User interface used by applying librarians to submit their data to be accepted or rejected by the admin.
ManageOwnProfileUI	User interface used by librarians to submit changes in their profile to be made.
LogInUI	User interface used by librarians or system admins to submit their name and password to log into their account.
ManageOtherProfileUI	User interface used by system admins to approve applications, promote account to have admin privileges, or demote other admins to librarians.

Control Classes:

Class Name	Description
MaintainBookRecordController	Controller that handles which specific controller to call to handle the request made by a system admin pertaining to the book record management.
BookEntryCreation	Controller that handles the request passed by the CreateBookEntryUI. Calls the createBookEntry method of the Book class and populates the entry with the supplied book details.
BookEntryRevision	Controller that handles the request passed by the BookEntryRevisionUI. Calls the editBookEntry method of the book class.
BookEntryDeletion	Controller that deletes book entries using the book ID passed by the DeleteBookEntryUI.
ViewBookController	Controller that handles the request passed by the ViewBookUI. Calls the viewBook method of the book class passing as argument the provided search criteria input from the UI.
BorrowBookController	Controller that handles the request passed by the BorrowBookUI. Calls the borrowBook method of the book class passing as arguments the provided borrower ID in the borrowBookUI and changes the status of the book to On Loan and the location to the borrower's name.
ReturnBookController	Controller that handles the request passed by the ReturnBookUI. Calls the returnBook method of the book class passing as arguments the provided borrower ID in the borrowBookUI and changes the status of the book to On Shelf and the location to the appropriate shelf/location in the library.
CreateProfile	Controller that handles the request made by the CreateLibrarianUI. Calls the createLibrarian method of the Librarian class passing as arguments the identification details provided in the UI (name, email, password).
EditOwnProfile	Controller that handles the request made by the ManageOwnProfileUI. Calls the edit method of the librarian class passing as arguments the provided details in the UI.
DeleteOwnProfile	Controller that handles the request made by the ManageOwnProfileUI. Calls the delete method of the librarian class.
LoginLibrarian	Controller that lets an authorized user log into their account given that they enter a correct name and password and gives them
ApproveCreateLibrarian	Controller that handles requests from the admin (specifically the ManageOtherProfileUI boundary) and approves the creation of new librarian profiles. It has 2 methods, approve() and reject(), for approving and rejecting the creation request, respectively.
GrantLibrarianAdminPriv	Controller that handles requests from the admin (specifically the ManageOtherProfileUI boundary) and grants admin privileges to selected Librarian profiles. It has 2 methods, promote() and demote(), for granting and removing admin privileges to and from a Librarian, respectively.

Entity Classes:

Class Name	Description
Book	Entity class representing a book in the library. It's description is composed of the attributes: title, author, keywords about the book's content, status (on shelf, on loan, or not available for borrowing), location (where the book can be located in the library if it is on shelf or to whom the book is loaned to if it is borrowed). Methods of the class allow for instantiating book entries, editing or deleting existing book entries, viewing book records and a book's details, borrowing or returning a book which updates a book's status in the record.
Librarian	This entity represents a librarian and/or System admin who authorizes, rejects, and oversees the interactions within the system in general. It has the following attributes: profile id, name, email, and password. It allows for creation, revision, deletion, and promotion of a librarian profile.