

DESIGN SPECIFICATION DOCUMENT FOR HIGH SCHOOL BOOK TRACKING APPLICATION

TEAM 13

Sandeep Inampudi	A20379958
Poojitha Vagale Duggappa	A20370777
Vaibhav Hongal	A20378220
Divija Sai Konanki	A20377662

TABLE OF CONTENTS

1. Introduction	2
1.1 Goals and objectives	2
1.2 Statement of Scope	2
1.3 Software context	3
1.4 Major Constraints	3
2. Data design	3
2.1 Internal software data structure	3
2.2 Global data structure	4
2.3 Temporary data structure	4
2.4 Database Description	4
3. Architectural and Component Level Design	6
3.1 System Architecture	6
3.2 Description of components	7
3.3 Dynamic Behaviour	10
4. User interface Design	13
4.1 Description of user interface	13
4.2 Design interface rules	17
4.3 Components available	18
5. Restrictions, Limitations and Constraints	18
6. Testing issues	18
6.1 Classes of tests	18
6.2 Expected software response	18
6.3 Performance bounds	19
6.4 Identification of critical components	19

1. Introduction

The main objective of this document is to illustrate the requirements for the project High School Book Tracking system.

1.1 Goals and Objectives

The overall goal of this project is to develop a Web-based application for bookkeeping in a high school library. It is used for controlling and monitoring transactions in the library.

This document describes all the data, functional and behavioral requirements of the system. This project is developed in Java and mainly focuses on basic operations in library like adding new members (students and teachers), adding new books, searching for books and facilitate students and teachers to access the library books and use them on a rental basis for a specified amount of time.

It also does a reporting job for producing the following reports.

1. List of all the students with books checked out showing due date and date checked out
2. List of all authorized users and their access level
3. List of all books in the system with search capabilities
4. List of all books checked out by class or book name/number
5. List of students and parents' info with amount due

This report generation functionality of the system helps to get a good idea of which books are borrowed by the members and makes users generate the reports' hard copy.

1.2 Scope of Project

The High School Book Tracking system basically upgrades the manual library system into an internet-based application so that the users can know the details of their accounts, availability of books, maximum borrowing capacity etc. This system can be used in any existing library or new library to manage the books and transactions. The project can easily be implemented for various situations. We can add new features as and when we require. The language used for development is Java which make the application platform independent and can be used on any system.

The application will be accessible over internet on any computer and provides the following capabilities.

1. The system provides login facility to the users [students/parents/teachers/admin].
2. The system provides the library staff with the ability to manage the library user accounts including add, edit or remove accounts.
3. The system lets the library staff to manage book inventory including add, edit or remove book information.
4. The system provides the members with the option to check their account and/or change their options like password of the account whenever needed.

5. The system lets the library staff to check which all members have blocked the books and whether they can borrow any more books or not. The application will track books, classes and students who check out books.
6. The system provides its registered users with an access to the system before they can use it, based on their level (students/teachers).
7. The application will generate reports for administrative purposes.
8. The application will provide a search function on books based on their ISBNs, school generated ids, subjects, author names etc.

The success criteria for the system is based in the level up to which the features described in this document are implemented in the system.

1.3 Software context

The application we are going to design will be web based and hence the clients shall use a browser to access the application. The application server (mostly tomcat) will handle the total application. The application will communicate to the database (SQL Server) to retrieve data about students and teachers and Google Books API as well to retrieve information about the books.

The application helps libraries of any size to track its books efficiently. It shall allow teachers to check out books to students, as per their registered courses. It shall also let the students track their due dates, and the late fee incurred if any.

1.4 Major constraints

This system is web-based and hence there will be a need to provide a PC hardware connected to the internet. It should be easy to use, providing help instructions and appropriate error messages for invalid operations. System should be reliable and should not have any unscheduled down time during high school library operational hours.

2.0 Data design

2.1 Internal Software Data structure

We are using Linked lists and hash maps as our data structures, to be passed among different components. Linked Lists are used when we need to add or delete a book from the system, and hash maps are used as a dict, to store the name of the books along with their ISBN's for easier access.

Eg : If the end user is the admin, and he needs to add a new book into the system, the details of the books are passed as the parameters within a method, and are added into the database with the help of linked list, similarly the books are deleted with the help of linked list.

Also, if a student has borrowed a particular book, his information is stored corresponding to the book that is borrowed within a hashmap, everytime the admin wants to get student info, the student ID is matched within the hash table and the student info is retrieved.

2.2 Global data structure

We have used dicts, as global data structures, in order to be able to access student info by all the components, the staff, students, parents and the admin. Also used if any of the components want to check the availability of certain books.

2.3 Temporary Data Structure

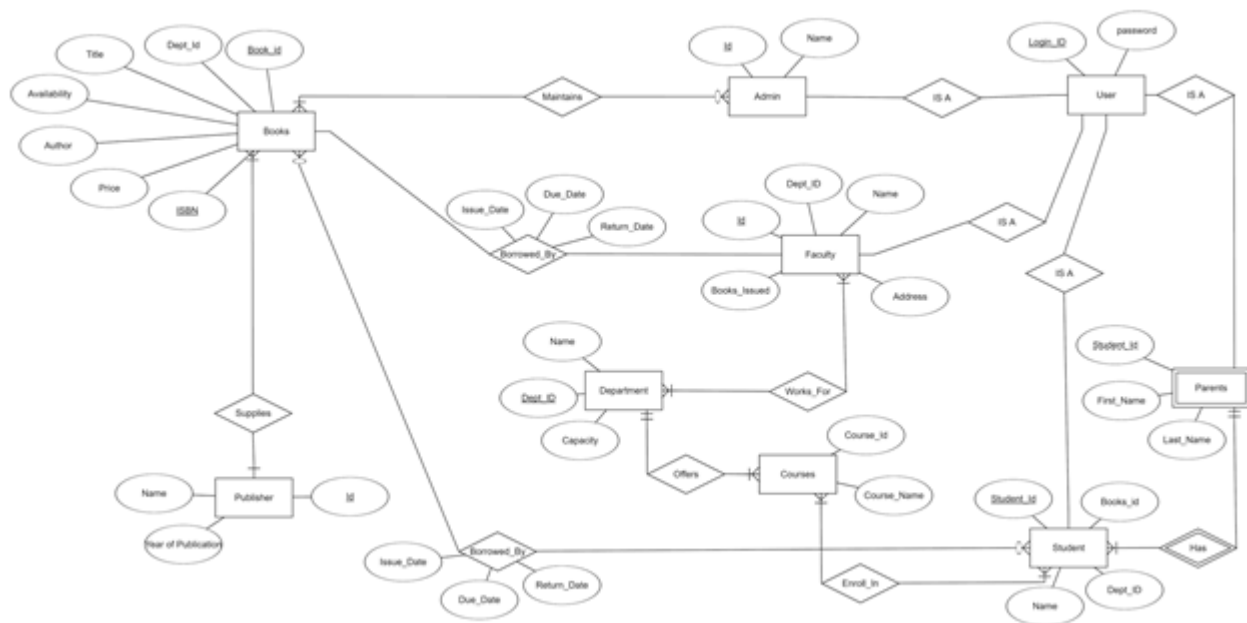
We have used files as temporary data structure, which stores a list of all the books that are rented by the students or teachers, it is stored in a temp file because once the books are returned, the main database which handled book info has to be updated and that is the primary source of book info.

2.4 Database Description

Our database consists of 4 separate components, which is the books, the list of students, teachers, and the admin. There is restricted access from one component to another. Eg : The parents can only access their children's history and do not have access to the book database.

Database design mainly included requirement analysis, concept structure design stage, the logic structure design stage, physical structure design stage, database implementation stage, database operation and maintenance stage, there're six steps altogether.

Among them; Conceptual structure design was the user's data needs which was got in demand analysis stage to be synthesized and abstracted, then turn them into a unified conceptual model. It was mainly expressed by the Entity-Relationship model; Logical structure design was to make sure the logical structure of database, it was to turn E-R model which was got in the concept structure design stage into relation model; Physical structure design stage was to choose a logical data model which was the most suitable for applications environment for physical structure (including storage structure and access method).

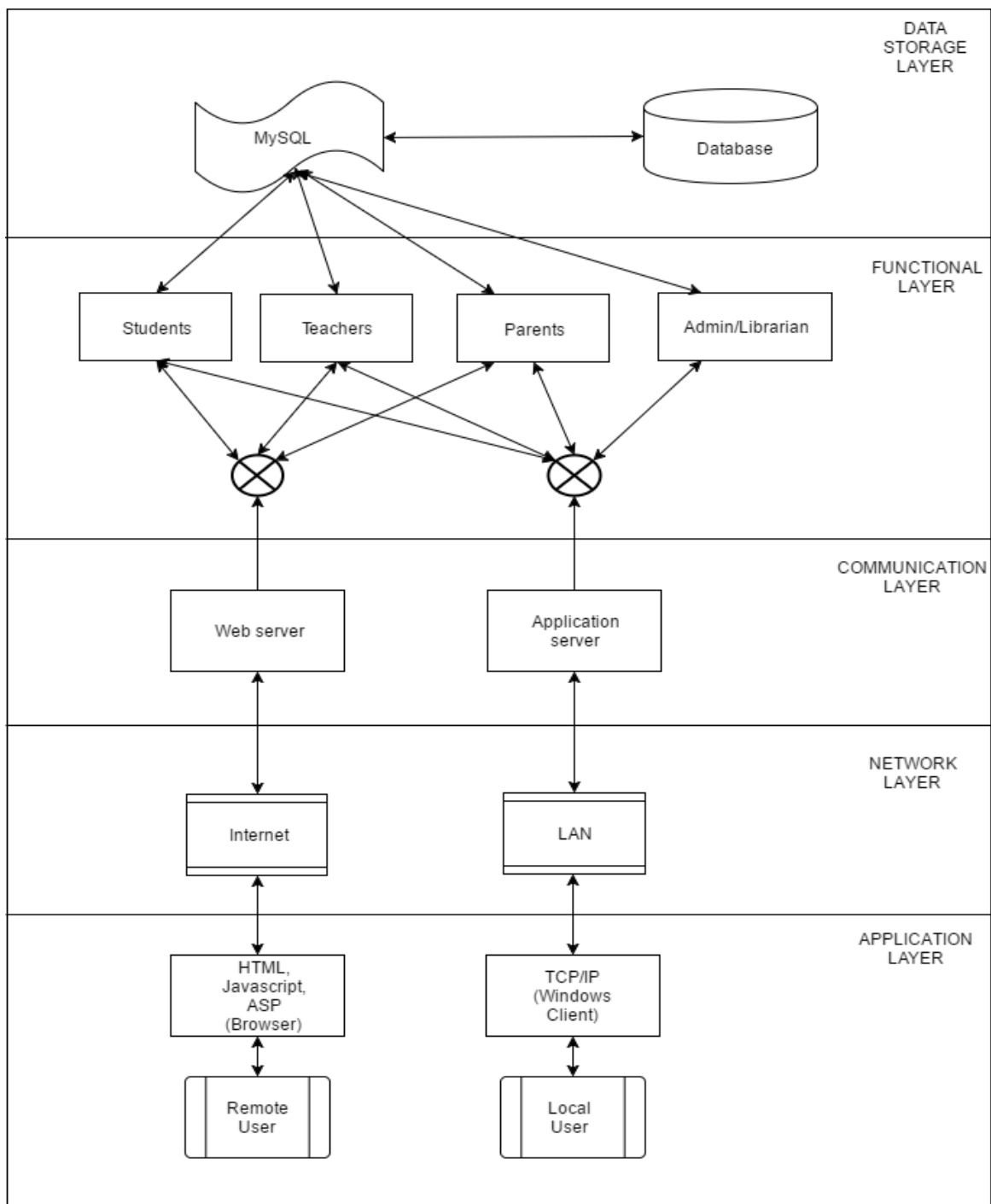
The Entity Relationship Diagram of our database:

3. Architectural and Component Level Design

3.1 System Architecture

3.1.1 Architecture Diagram

The main purpose of the architectural design is to determine the structure of the software. It is important because considering it the entire system will be designed. Our system architecture is shown below using the diagram.



3.2 Description for Components

1. Data Storage Component

3.2.1 Processing narrative

This component provides the functionality of the data storage. We intend to use MySQL database and SQL queries to query the database. The database provides the storage for keeping the information of students, their parents, teachers and all the available books in the library. the users will interact with the database via the SQL server.

3.2.2 Interface description

Input: Queries from different system users, in the form of SQL commands.

Output: Resultset from database according to the query.

3.2.3 Processing detail

We use mainly these commands in the MySQL code, they are insert, delete, add, update, select. During the registration, we use insert to add login credentials and other info to store into the Database. We use Add/delete/update to work on the pending book requests, students lists etc. we use select to choose or issue a certain book to a certain student along with other commands stated above.

3.2.3.1 Restrictions/limitations

The restrictions we face here mainly come from the standard restrictions of the MySQL ecosystem like read and write speed and limitations etc.

2. Functional layer Component

3.2.1 Processing narrative

This component is the backbone of the whole system. It gives the architecture for the users of the system like teachers, students, parents and admins. Different users have different functions allocated to them in addition to all of them having some common functionalities like login, registration. After registration/login, as one of the roles stated above, the users will be directed to their respective landing pages, where they can carry out their specified functions like working on the book requests for teachers, adding students for admin, searching and requesting books for students, information about their children for parents.

3.2.2 Interface description

Input: user id, role and password were the primary inputs, then depending on the role, inputs may range from button clicks to searches on the GUI, which will be then transferred to the data storage component to either read or write data from the databases.

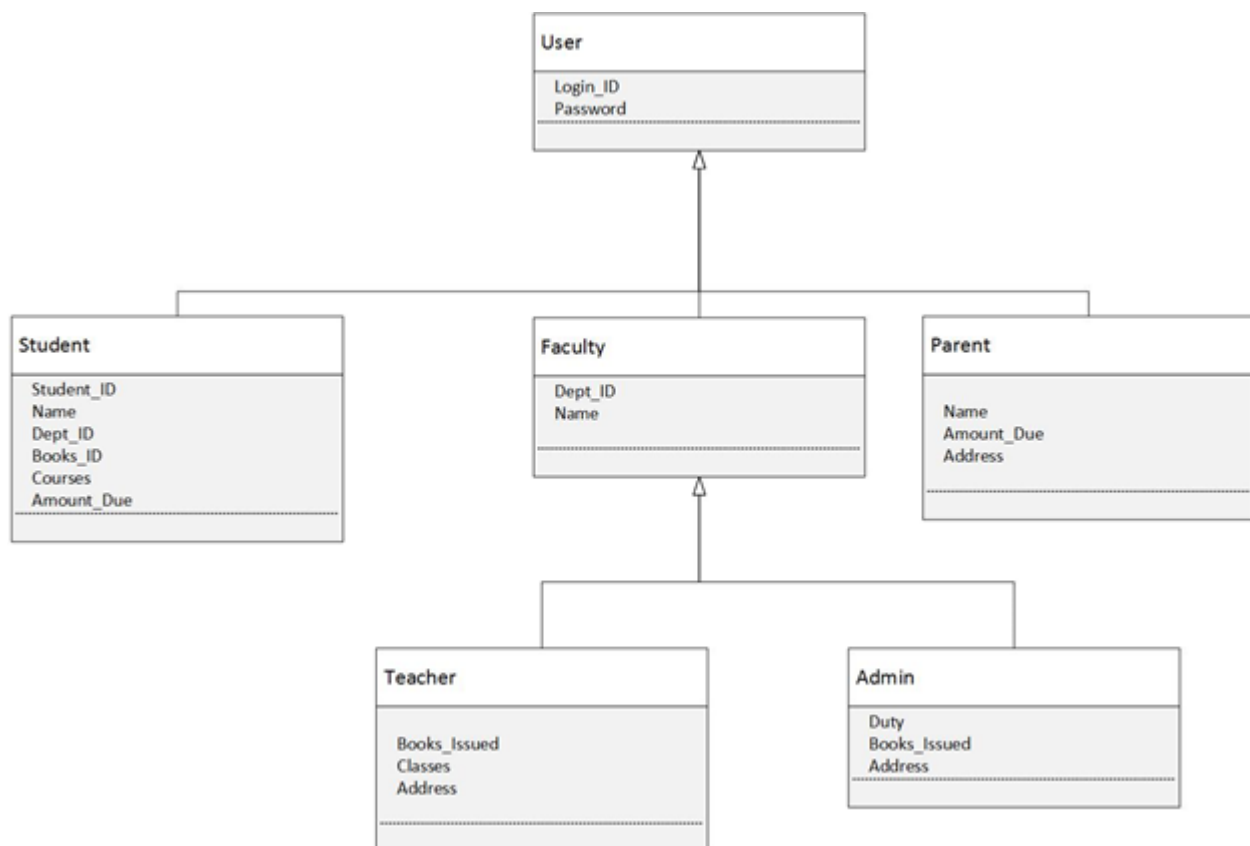
Output: The outputs of this component mainly depends on the role of the user. Student users can get outputs like search results of books, their fee due etc. Teacher user's outputs may vary from their class roster, books checked out by

students etc. admin user's outputs may vary from pending student approvals to adding new books etc.

3.2.3 Processing Detail

A detailed algorithmic approach of the component is presented below

3.2.3.1 Design class hierarchy



3.2.3.2 Restrictions/limitations

Some of the users shall have their user ids and passwords generated by the admins before the installation of the system to start working initially.

3.2.3.3 Performance issues

Downtime or any issues with google servers may affect the performance of the search function as we are using google books API as the books database.

3.2.3.4 Processing detail

Name	description
Add	Allows the admin to add new students
Search	Allows the users search for books
Login	Processes the entered credentials at the login phase
Book request	Allows the students to request for a book from the search results
Book issue	Allows the teachers issue books to students
Book return	Allows the students to return the book at a pickup location
add/delete/update	Allows the admins to update or delete info on books

3. Communication component

3.2.1 Processing narrative

The communication layer mainly consists of two components, web server and application server. The web server loads all the web pages we designed and the application server does all the application part to generate data to populate the SQL tables, search results, and retrieve information from the tables etc. so that the web server will present this data to the user.

3.2.2 Interface description

Input: The inputs to this component are the user requests and data from like searches, logins registrations

Output: the outputs of this component is mainly web pages we designed and data the user requested pulled from the database or google API

3.2.3 processing detail

As soon as the user's request the first login page, their entered info will be cross checked with the info in our database and will give the next landing page according to the correctness of the credentials and the role selected. The functions in the landing page will lead to different following webpages or different search results according to the function selected.

3.2.3.1 Restrictions/limitations

The limitations of this component arises from the bandwidth problems on the user side or from the Google API side. And also our server may not perform very well at peak load situations.

4. Network component

3.2.1 Processing narrative

The network component mainly consists of two parts, viz. internet and LAN. Most of our traffic will come from remote users via internet. We are using HTTP POST method while developing the web application, so our transmitted data through the network will be secure.

3.2.2 Restrictions/limitations

The limitations of this component will come mostly from the server downtime, heavy traffic or requests, any other connection problems that hinders proper functioning of the web application.

3.2.3.3 Performance issues

Again the performance of the whole web application depends on many factors like amount of traffic, Server downtime, Google API snags etc. All these issues may cause the performance of the whole application go down.

5. Application component

3.2.1 Processing narrative

We will develop the whole web application using HTML5, CSS, JavaScript, XML, JSP. All these together will be bundled and loaded in a Apache server. The HTML will display the webpage beautified by custom cascading style sheets developed and tailored for a typical book tracking application. JSP and JavaScript code will do all the processing required for the functionality, data manipulation and connectivity requirements of the application.

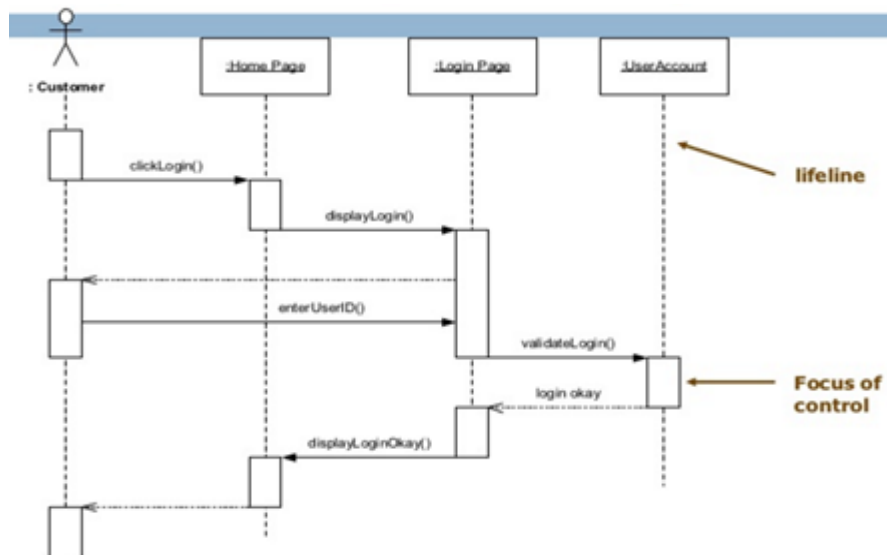
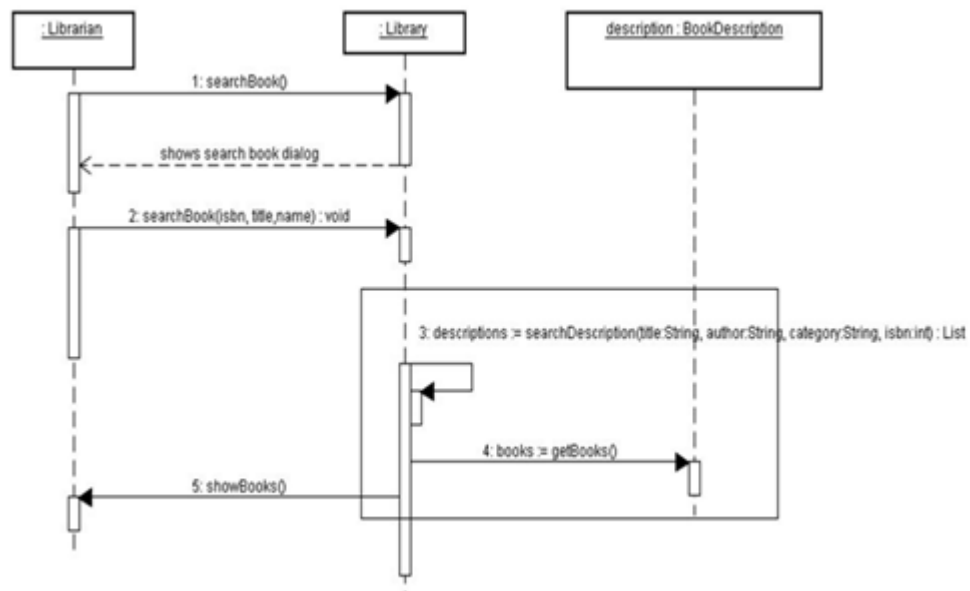
3.3 Dynamic Behavior

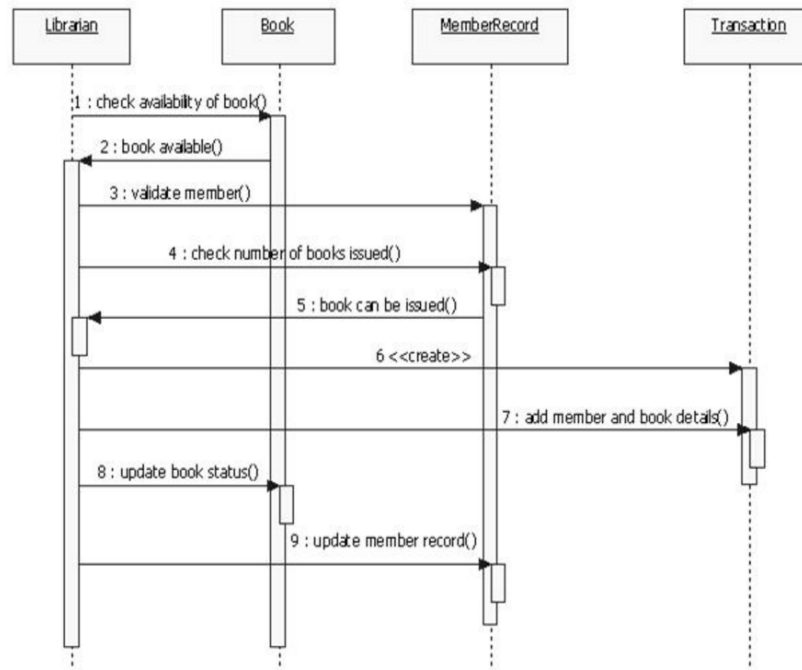
A description of the interaction of the classes is presented below.

3.3.1 Interaction Diagrams

A sequence diagram, for each use case the component realizes, is presented.

Phases for the sequence diagram are Registration, login, Book search, transaction, View/edit information and report generation.

Login:**Search book:**

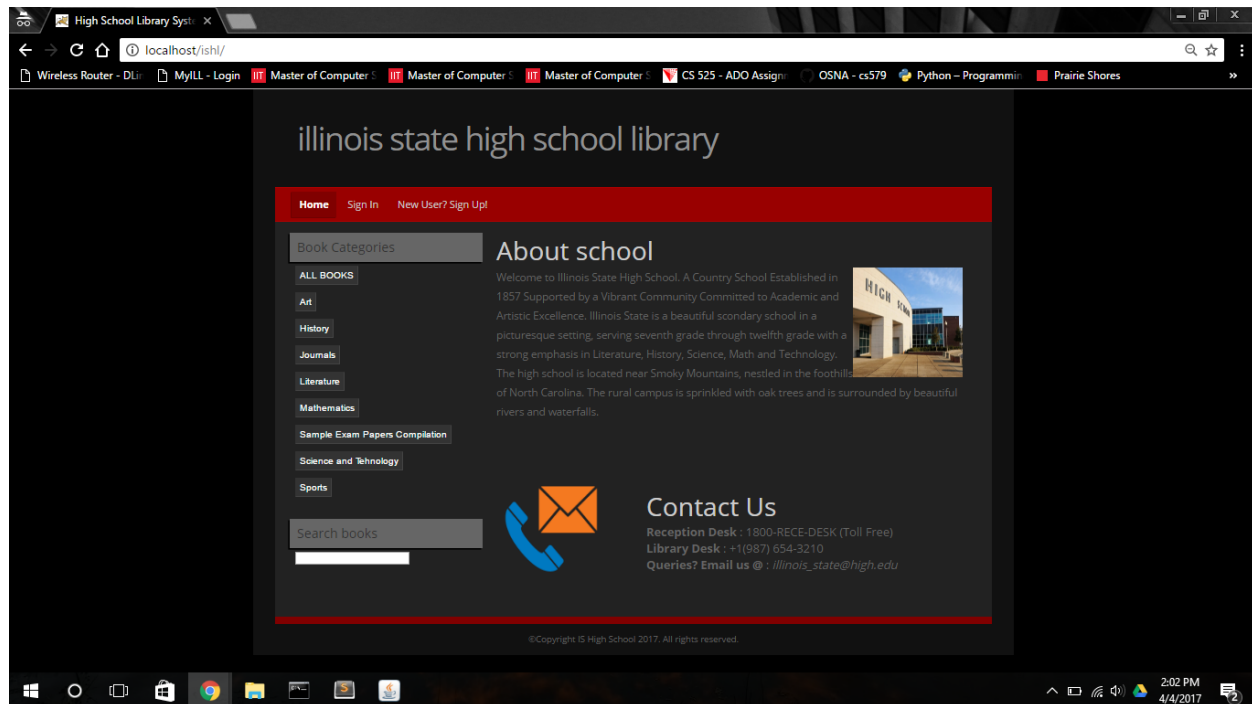
Transactions:

4. User Interface Design

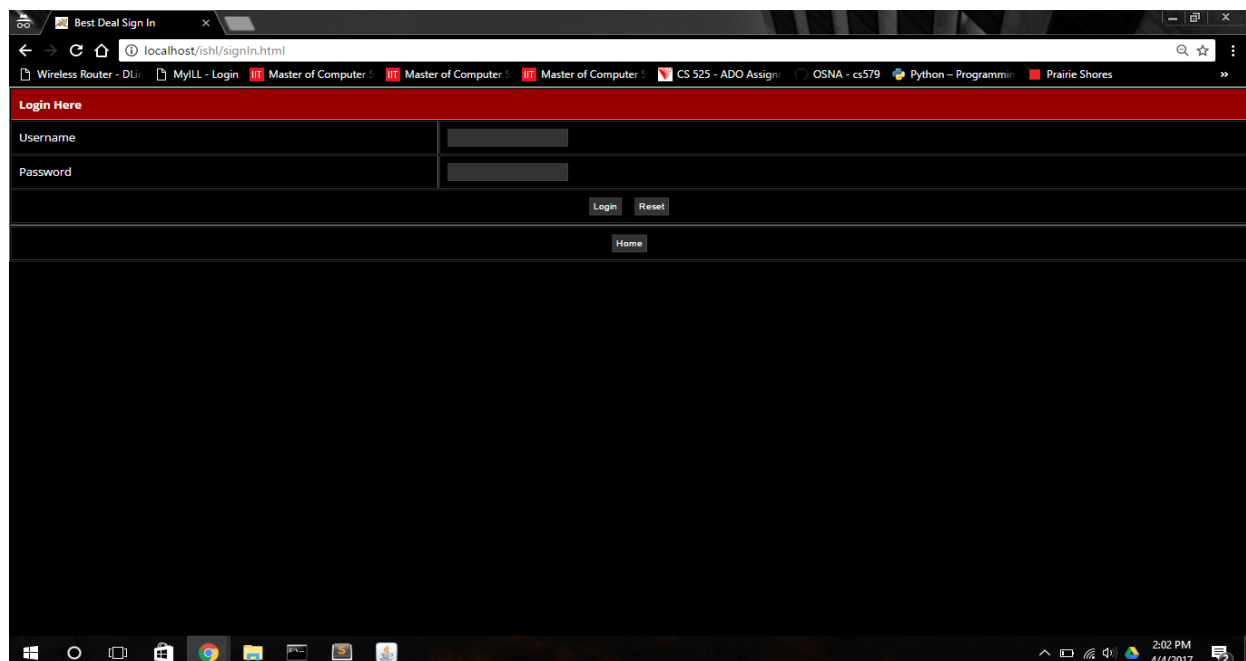
4.1 Description of the user interface

4.1.1 Screen images

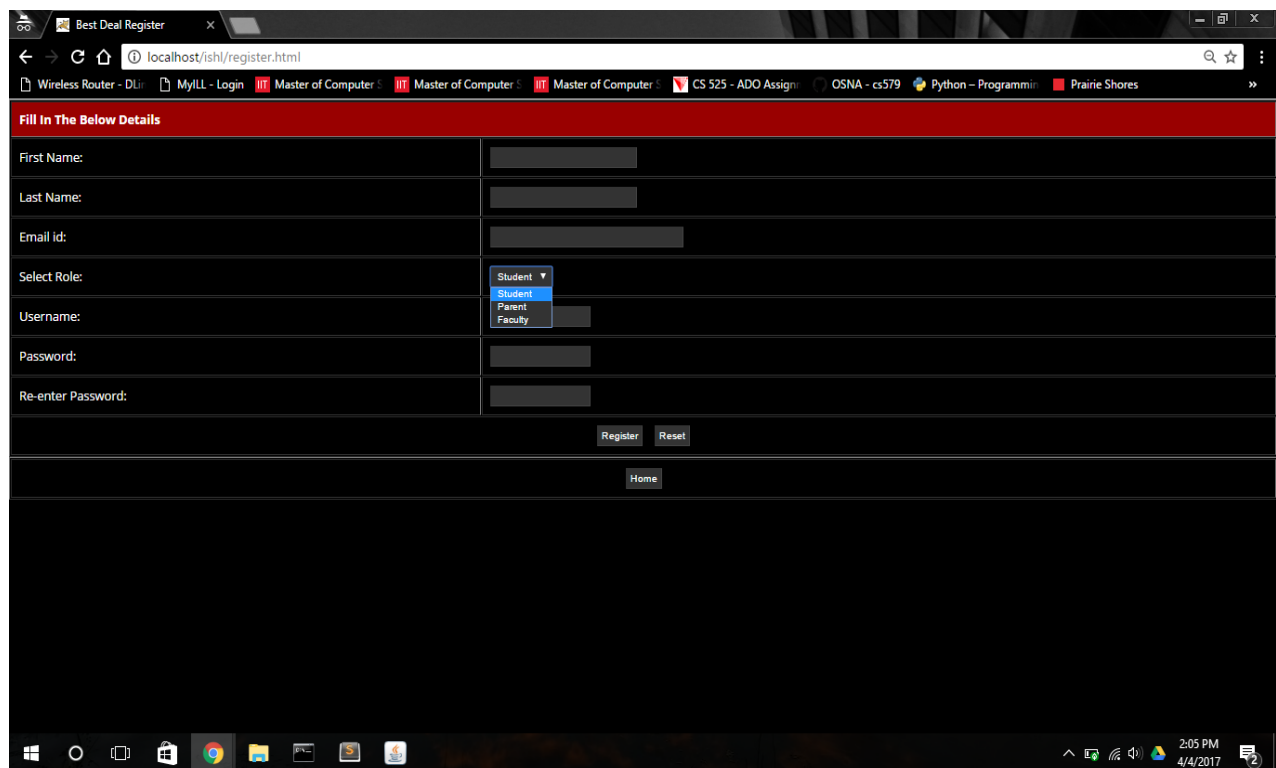
Home Page



User Login page

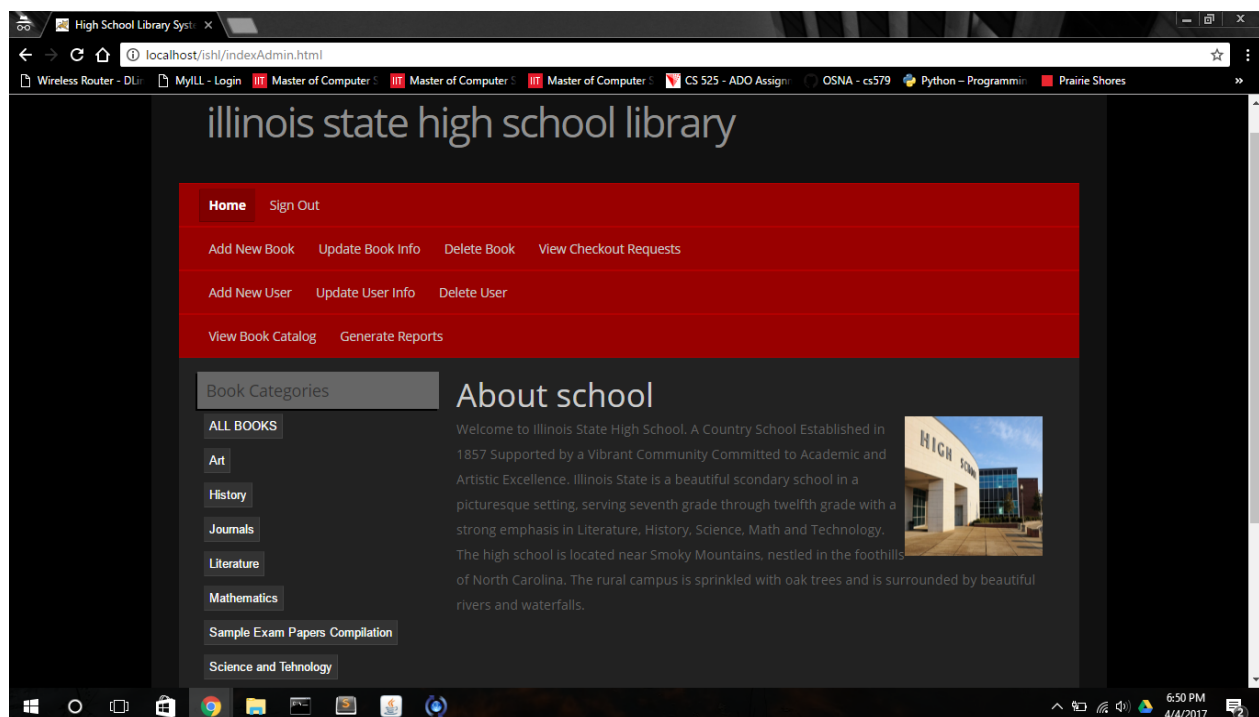


New User Registration page



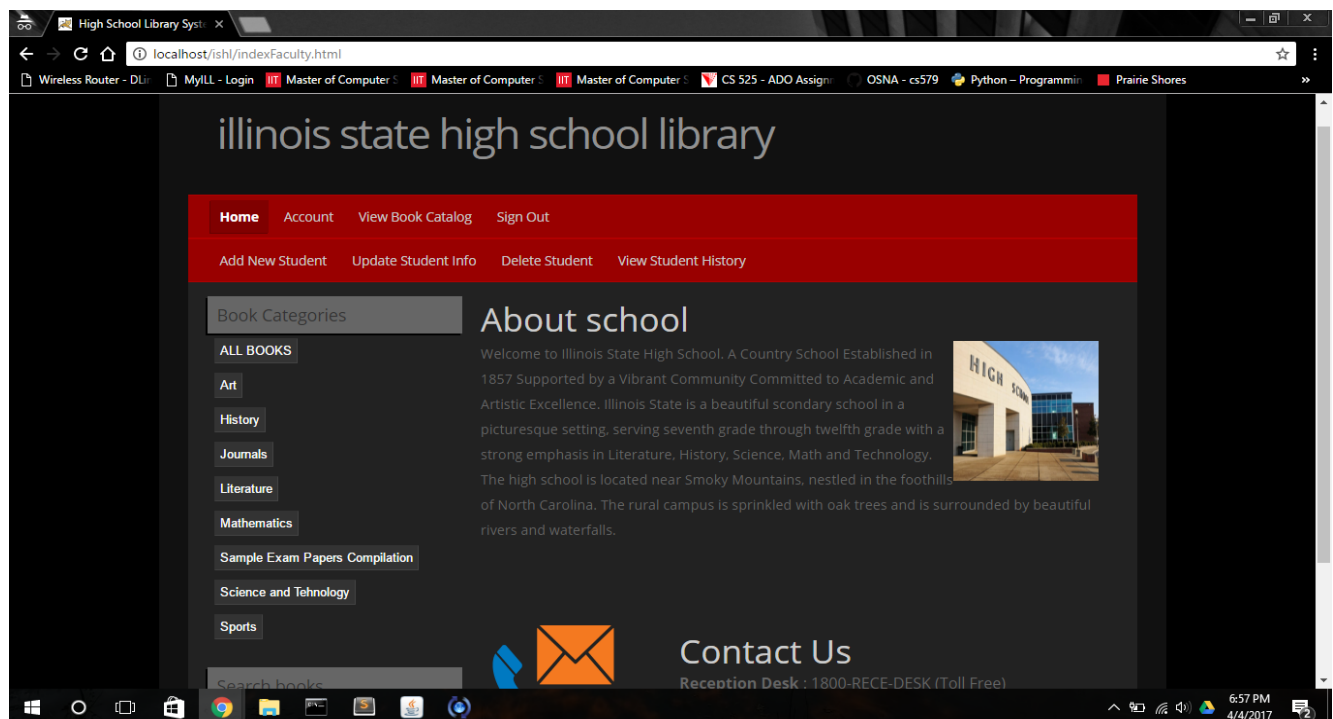
The screenshot shows a web browser window with the address bar displaying 'localhost/ish/register.html'. The page has a dark theme with a red header bar containing the text 'Fill In The Below Details'. Below the header, there is a registration form with the following fields: First Name, Last Name, Email id, Select Role (a dropdown menu with options: Student, Parent, Faculty), Username, Password, and Re-enter Password. At the bottom of the form, there are 'Register' and 'Reset' buttons. Below the form, there is a 'Home' button. The browser's taskbar at the bottom shows various icons and the system clock indicating 2:05 PM on 4/4/2017.

Admin Login and Functionalities

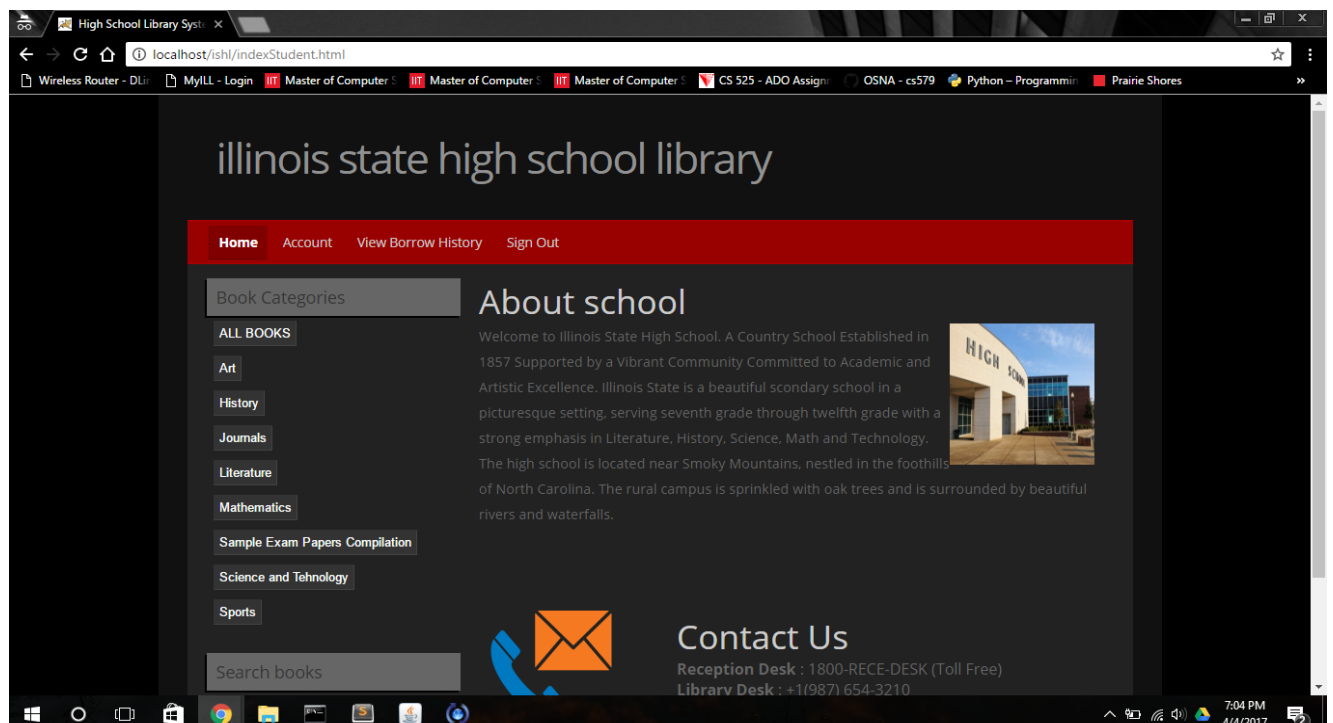


The screenshot shows a web browser window with the address bar displaying 'localhost/ish/indexAdmin.html'. The page has a dark theme with a red header bar containing the text 'illinois state high school library'. Below the header, there is a navigation menu with links: Home, Sign Out, Add New Book, Update Book Info, Delete Book, View Checkout Requests, Add New User, Update User Info, Delete User, View Book Catalog, and Generate Reports. Below the navigation menu, there is a section titled 'Book Categories' with a list of categories: ALL BOOKS, Art, History, Journals, Literature, Mathematics, Sample Exam Papers Compilation, and Science and Tehnology. To the right of the book categories, there is a section titled 'About school' with a paragraph of text and a small image of a school building. The browser's taskbar at the bottom shows various icons and the system clock indicating 6:50 PM on 4/4/2017.

Faculty Login and Functionalities



Student Login and Functionalities



4.1.2 Objects and actions

Home Page

The home page is the home of the application. It displays the general and contact information about the school. It also provides the functionality of a search bar where the student can enter the name of the books and retrieve the book directly. Also, different categories of books available in library are shown on the left, using which the user can view the books in different categories. The following links are also used in the Home page:

1. Home: It takes the user to the home page of the application.
2. Sign In: This is used by different users for signing in to the system, only after which they can perform their tasks. It redirects to a separate sign in page.
3. New User? Sign Up! : This is a registration link for any new user (student/faculty/parents). It is used for account creation for accessing the bookstore. It redirects to a separate registration page, with a registration form.

User Login Page

This page provides a form with username and password fields for a registered user to login to the system and search the library. It loads up when Sign In link is clicked on Home page. It provides the following buttons:

1. Submit - for submitting the form
2. Reset - for clearing the text fields
3. Home - for redirecting to home page

New User Registration Page

This page provides a form to be filled by the first time users of the library system, so as to register them on to the system. This page provides with

1. Text Fields for inputting the user information such as first name, last name, role of the user, preferred username etc.
2. Register button - for submitting the registration form
3. Reset button - for clearing the text information entered
4. Home button - for redirecting to home page.

Admin, Faculty and Student Login and Functionalities Page

These page are for the Admin, Faculty (teachers) and Students respectively. They provide:

1. Admin functionalities like adding new users to the system, adding new books to the system, updating the users' information/books' information, managing users, generating various reports etc.,
2. Faculty functionalities such as new student adding, updating student info, viewing the student history etc.

3. Student Functionalities like editing account info, adding books to cart, searching for books etc.

4.2 Interface design rules

To improve the usability of an application, it is important to have a well-designed interface. Shneiderman's "Eight Golden Rules of Interface Design" are a guide to good interaction design. The rules are as follows:

1. Strive for consistency
Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.
2. Enable frequent users to use shortcuts
As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.
3. Offer informative feedback
For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.
4. Design dialog to yield closure
Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.
5. Offer simple error handling
As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.
6. Permit easy reversal of actions
This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.
7. Support internal locus of control
Experienced operators strongly desire the sense that they are in-charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.
8. Reduce short-term memory load
The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

4.3 Components available

All the below components are used in the High School Book Tracking Application at different stages.

4.3.1 Input Components

1. Text box
2. Text area
3. Buttons
4. Drop down list

4.3.2 Navigation Components

1. Search fields
2. Buttons

4.3.3 Information Components

1. Forms

5. Restrictions, limitations, and constraints

- Some of the users shall have their user ids and passwords generated by the admins before the installation of the system.
- Downtime of Google servers may affect the system as we are using google books API.
- The school's billing system must be properly integrated by the Admins.

6. Testing Issues

6.1 Classes of tests

- Testing process ensures that all the requirements have been tested. It also ensures that the given input produces actual results that agree with the required results. In this phase, we shall consider all the likely variations to which it will be suggested and push the system to limits.
- Two major types of testing are performed. They are:
 - 1) White-Box Testing: It is also called as "Glass Box Testing". All logical decisions can be checked. Bugs prevailing in any part of the code can be fixed
 - 2) Black Box Testing: This testing focuses on the functional requirements of the software. This is a complementary approach to white box testing that uncovers different class of errors- Interface errors, Performance errors, Initialization and termination errors

6.2 Expected software response

- Test cases will be generated for all the modules and analysis will be done to check the results.
- Validation of username/password combination.

- Check for response time.
- Checking if the book information and availability status is correctly displayed or not.
- Check for maximum number of books to be issued.

6.3 Performance bounds

- Time taken to search books, access information and view issue history should not be more than 5sec. System shall take as less time as possible.
- Fetching data from server or searching data should not make the user wait too long.

6.4 Identification of critical components

- Components that are critical and demand attention during testing are identified.
- Server should be adoptive against increasing number of users. For example, response time for 1000 users is 1 sec; for the next user, it should increase response time for acceptable.